A Broadcom Company

technologies

DX NetOps - 20.2

Table of Contents

Getting Started

DX NetOps improves your time to value through advanced AI capabilities and unified network visibility. It delivers simplified NetOps intelligence with a user experience that traverses modern architectures through high scale, unified network monitoring. This unified monitoring enables full-stack analytics for assuring traditional and modern architectures.

DX NetOps converts inventory, topology, device metrics, faults, flow and packet analysis into actionable intelligence for network operations teams. Complimented by our AIOps solution, DX NetOps, enables IT teams to establish proactive, autonomous remediation capabilities across the applications, infrastructure, and networks that fuel superior user experiences.

The following video shows DX NetOps in action:

Key Solutions

DX NetOps includes the following integrated solutions:

- Performance Monitoring with DX NetOps Performance Management
  Performance Monitoring: Monitors, stores, analyzes, and displays a massive amount of information for assuring service quality across large, complex, multi-technology, multi-vendor network infrastructure.
- Modern Network Monitoring with DX NetOps Virtual Network Assurance
  Modern Network Monitoring: Extends traditional network monitoring to the virtual network. Provides modern network monitoring for software-defined architectures and hybrid cloud platforms. Correlates logical network entities with physical resources.
- Fault Monitoring with DX NetOps Spectrum
  Fault Monitoring: Map network topology and create alarms from performance events.
- Non-SNMP Monitoring with DX NetOps Mediation Manager
  Non-SNMP Monitoring: Integrate non-SNMP monitoring in your DX NetOps environment.
- Flow Monitoring with Network Flow Analysis
  Flow Monitoring: Monitor interface performance and bandwidth utilization with insight into the traffic going through your network.

Key Features

- **Cross domain AIOps insights**
  Delivers algorithmic noise reduction, anomaly detection, and application-centric root cause topologies.
- **Correlated, normalized, and reliable network data**
  Provides network- aware service, alarm, capacity and topology analytics.
- **Vendor agnostic modern architecture support**
  Enables project monitoring from pilot to global deployment.
- **Optimized application policies**
  Balances the end user experience and cost.
- **Operational workflows**
  Simplifies large and complex SDN technologies.
- **Unified device availability, flow, faults and packet analysis**
  Enables granular visibility.
- **Full stack, contextual, end-to-end workflows**
  Provides for different roles in a single tool.

NetOps Architecture

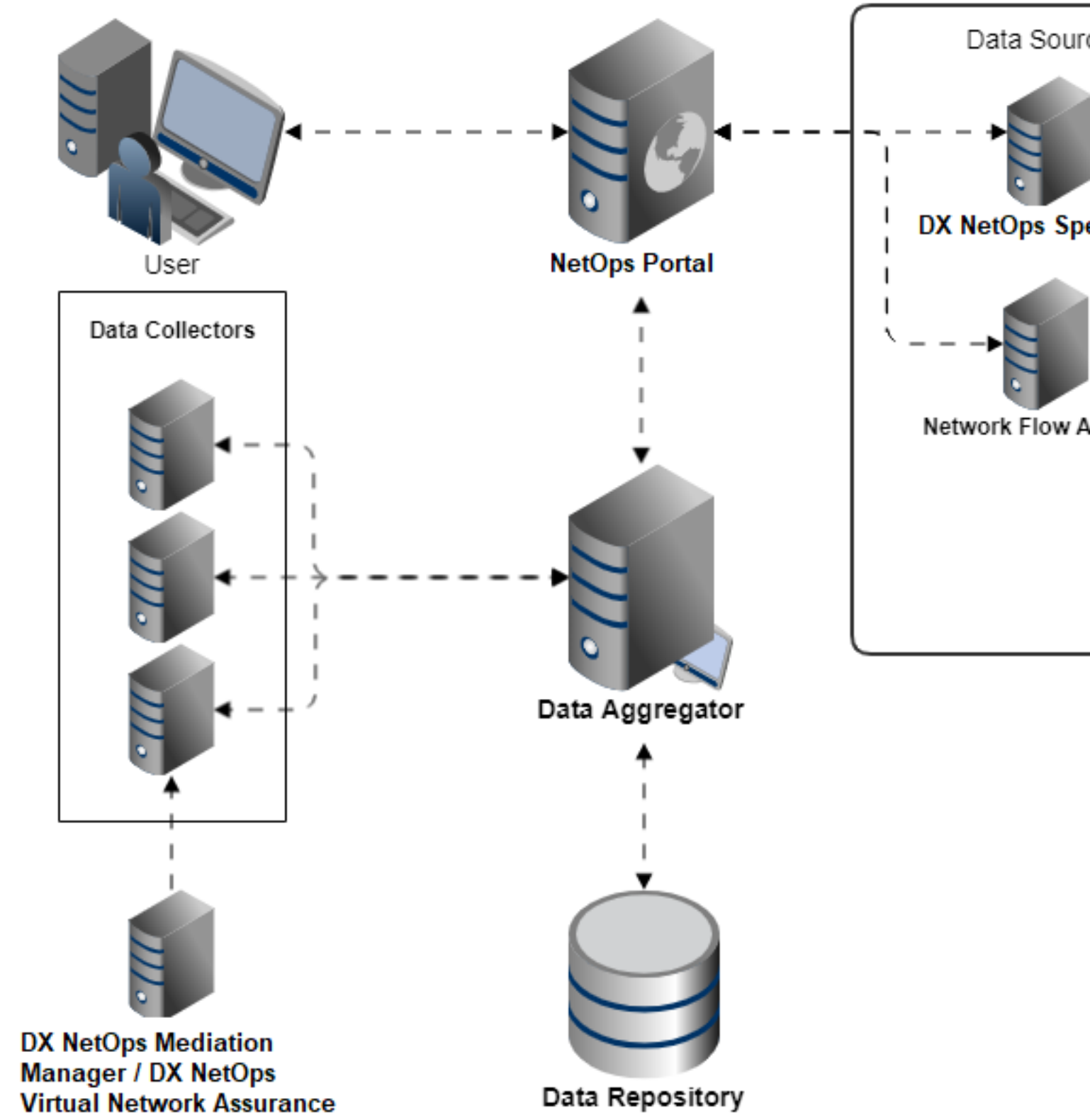
The architecture of DX NetOps supports the following NetOps monitoring:

- Performance Monitoring with DX NetOps Performance Management
- Modern Network Monitoring with DX NetOps Virtual Network Assurance
- Fault Monitoring with DX NetOps Spectrum
- Non-SNMP Monitoring with DX NetOps Mediation Manager
- Flow Monitoring with Network Flow Analysis

The following image shows the basic system architecture of DX NetOps:

NOTE
Other integrations are supported. For more information, see each monitoring section.

Figure 1: Basic System Architecture



The following video walks through the overall DX NetOps installation, which is covered in more detail in each section:

Release Notes

The Release Notes section outlines the new features and current and fixed issues for DX NetOps Performance Management. In addition, the release notes contain the Third-Party Software License Agreements, which detail the terms and conditions of using third-party software in the creation of DX NetOps Performance Management. Use the release notes as a starting point when you first install, or upgrade to a new version of DX NetOps Performance Management.

Release Comparison

This table compares the key features in recent releases for DX NetOps Performance Management:

Key Features	Release 20.2	Release 3.7	Release 3.6	Release 3.5	Release 3.2
New Group Editor	**Yes**	No	No	No	No
Flow Administration	**Yes**	No	No	No	No
Separate Health Dashboards	**Yes**	No	No	No	No
128T Plug-in	**Yes**	No	No	No	No
Broadcom Broadview Plug-in	**Yes**	No	No	No	No
Silver Peak Plug-in	**Yes**	No	No	No	No
Poll Time Configuration	**Yes**	No	No	No	No
Nuage Network Device Support	**Yes**	No	No	No	No
Spectrum Performance View (Beta) Improvements	**Yes**	No	No	No	No
Launch OneClick Clients with Context	**Yes**	No	No	No	No
Spectrum TrapInsight Dashboard View	**Yes**	No	No	No	No
CASBC Devicepack	**Yes**	No	No	No	No
CA5620SamY1731 Devicepack	**Yes**	No	No	No	No
CAPaloAltoPanOS Devicepack	**Yes**	No	No	No	No
Manage Address-Hostname	**Yes**	No	No	No	No
NFA ODataAPI QueryBuilder	**Yes**	No	No	No	No

New Features and Enhancements

The following new features have been added in this release. Some of the existing features have been improved, and are listed under the Enhancements section.

For more comprehensive and detailed information about the new features and enhancements in this release, see the Release Notes in each of the following monitoring sections:

- Performance Monitoring with DX NetOps Performance Management
- Modern Network Monitoring with DX NetOps Virtual Network Assurance
- Fault Monitoring with DX NetOps Spectrum
- Non-SNMP Monitoring with DX NetOps Mediation Manager
- Flow Monitoring with Network Flow Analysis

New Features

New Group Editor

The new and improved group editor no longer uses Adobe Flash.

The following out-of-the box group names changed:

- "Defined Tenants" changed to "Tenants"
- "Collections" changed to "Custom Collections"
- "Domains" changed to "IP Domains"
- "Service Provider Global Groups" changed to "Global Tenant Groups"
- "Service Provider Defined Groups" changed to "Global Tenant Groups"
- "Service Provider Items" changed to "Global Tenant Items"

> **NOTE**
> These name changes impact any existing custom scripts using the previous naming. You must update your scripts accordingly. The DA REST paths have changed.

For more information, see the Groups section.

Network Flow Analysis Administration

This release introduces the ability to manage network flow processing and view the overall heath status of your Network Flow Analysis reporters, harvesters, and interfaces.

For more information, see Flow Administration, Manage Network Flow Processing, View the Health of the System.

Separate Health Dashboards

In previous releases, the Data Aggregator Health Dashboard and Data Collector Health Dashboard were combined into a single dashboard. For clarity, they now appear as separate dashboards.

For more information, see Data Aggregator / Data Collector Health Dashboards.

New Plug-ins

The following plug-ins are newly supported:

- 128T SD-WAN
- Broadcom BroadView
- Silver Peak

Poll Time Configuration

The default timeout value for a poll in the engine of any plug-in is 30 minutes. If a poll requires more than the default value, you can configure the `sdn_poll_timeout_minutes` parameter in the `/etc/VNA.cfg` file. The unit for the parameter value is minutes. After you update this parameter, you must restartDX NetOps Virtual Network Assurance.

Support for Nuage Network Devices

From the current release, DX NetOps Spectrum supports monitoring of Nuage network devices through DX NetOps VNA integration. This functionality allows you to use the SD-WAN solution that is provided by Nuage. SD-WAN stands for Software-Defined Wide Area Networking. It is a combination of Software Defined Networking (SDN) and Wide Area Networking (WAN).

For more information, see Monitoring SD-WAN for Nuage.

Spectrum Performance View (Beta) Improvements

From the current release, DX NetOps Spectrum installer configures the Influx server and automatically creates the user *spectrum*. A new page named **InsideView Configuration(beta)** is added under the OneClick Administration page to configure InsideView, which has options to save, start, and stop InsideView.

For more information and to configure the influxd using https, see InsideView Configuration (beta) in OneClick Administration Pages and Spectrum Performance View (Beta).

Launch OneClick Clients with Context

Launch the OneClick WebApp in-context to open the alarm, explorer, and topology.

For more information, see Launch OneClick Clients with Context.

Spectrum TrapInsight Dashboard View

From this release, DX NetOps Spectrum TrapInsight provides a real-time trap trend analysis dashboard for the distributed DX NetOps Spectrum environment. The administrator can run this tool to get the trap analysis trend; it is disabled by default. This feature is part of SDC with TrapX installation. When a new trap is received, TrapX forwards the trap to Logstash. Logstash processes and sends the trap to the Influx database using the logstash-influx-output plugin.

For more information, see Spectrum TrapInsight Dashboard View.

CASBC Devicepack

This device pack collects performance metrics of:

- CallRecord
- SummaryRecord
- VoiceQualitySummaryRecord

CA5620SamY1731

This device pack collects cfmTwoWayDealy SAS metrics of SAM 5620 (NFM-P) through FindToFile and LogToFile by enabling JMS subscription. It collects the following performance metrics.

- MinimumResponse
- MaximumResponse
- SuccessfulAttempts
- MaxOneWayDelaySrcDest
- MinOneWayDelayDestSrc
- MaxOneWayDelayDestSrc
- MinOneWayDelaySrcDest
- PacketsLost
- PacketsSent
- PacketsArrivingAfterTimeout
- AverageJitter
- AvgResponseTime
- JitterOut
- JitterIn
- Jitter
- DelayVariation
- AvgOneWayDelaySrcDest
- AvgOneWayDelayDestSrc
- Latency
- AveragePercentPacketLoss

CAPaloAltoPanOS

This device pack collects data for the PaloAlto devices from the api and shows the performance stats for:

- Utilization
- SessionAverage
- SuccessfulAttempts
- SessionMaximum
- PacketBufferAverage
- PacketBufferMaximum
- PacketDescriptorAverage
- PacketDescriptorMaximum
- PacketDescriptorOnchipAverage
- PacketsArrivingAfterTimeout
- PacketDescriptorOnchipMaximum

Manage Address-Hostname

The NFA maintains only the latest resolved hostname against the IP address previously. From NFA 10.0.3, we maintain the historical data of the IP address and hostname association.

NOTE

More Information:

Manage Address-Hostname

IP Address and Hostname Historical Data

IP Address and Hostname API

NFA ODataAPI QueryBuilder

The ODataAPI is a flexible tool that lets users easily extract data from the DX NetOps database. The ODataAPI enables integration between DX NetOps data and external applications. The ODataAPI is a public API that uses the QueryBuilder GUI. The QueryBuilder is a guided URL builder that lets you create custom Query URLs to extract and explore performance data. The URLs return customized data in the specified format. You can view the data in a browser or process the data in a custom web application.

More Information: NFA OData QueryBuilder

NetOps Compatibility

NetOps Portal can use the following CA Technologies products as registered data sources:

- DX NetOps Virtual Network Assurance
- DX NetOps Spectrum
- Network Flow Analysis and Anomaly Detector
- CA Application Delivery Analysis
- CA Unified Communications Monitor
- CA Application Performance Management
- CA Business Intelligence

For supported releases, see the DX NetOps Interoperability.

Some components function as data sources. For these components, use only the release that is delivered as part of DX NetOps Performance Management:

- Data Aggregator
- Event Manager

Performance Monitoring

DX NetOps Performance Management monitors, stores, analyzes, and displays a massive amount of information for assuring service quality across large, complex, multi-technology, multi-vendor network infrastructure. The solution helps the largest networks successfully monetize service offerings while lowering the cost and complexity of service.

This section contains everything you need for performance monitoring from getting started to troubleshooting information.

Getting Started

DX NetOps Performance Management monitors, stores, analyzes, and displays a massive amount of information for assuring service quality across large, complex, multi-technology, multi-vendor network infrastructure. The solution helps the largest networks successfully monetize service offerings while lowering the cost and complexity of service.

Communications service providers can use DX NetOps Performance Management to improve network monitoring and delivery of revenue-generating services, such as 4G LTE, Voice over LTE, Mobile Backhaul, Metro Ethernet and more. Enterprises can use DX NetOps Performance Management to assure underlying network services for applications that drive their internal business processes and revenue-generating customer interactions.

Key Features

Very high-scale monitoring architecture on a platform that scales efficiently.

The system architecture provides scale that supports the largest networks. To understand the system requirements to support your scale, see the DX NetOps Performance Management Sizing Tool.

Unified multi-technology, multi-vendor device monitoring. Certifications for classic network devices and specialized carrier Ethernet, WiFi offloading, and mobile wireless equipment.

DX NetOps Performance Management supports the common vendors, metrics, and components in your network infrastructure. The Technology Certification Portal lists out-of-the-box certifications by Data Aggregator version, vendor certification, and metric families: http://serviceassurance.ca.com/im

The extensible certification model enables users to customize communication with monitored devices. For more information, see Self-Certification.

Intelligent analytics, high-scale visualization, and fast processing for instant reporting. Flexible, easily customizable dashboards and reports.

The customizable dashboards and views provide flexible visualization. For example, a regional manager uses a dashboard that pins views to each site group in that region and systems administrator uses a dashboard to monitor all servers. Dashboards show information that is scoped to customizable groups. Context pages show information that is related to a specific item in the system. Views provide visualization options.

Extensible architecture for easy integration and automation.

DX NetOps Performance Management supports downstream integration through the customer facing OpenAPI. For more information, see Integrating.

Predictive analytics to give a complete, unencumbered view of the network, and business key performance indicators.

Configurable and dynamic projections provide insight into capacity planning and situations to watch. For more information, see Metric Projection.

Support for modern network monitoring.

DX NetOps Virtual Network Assurance integrates with DX NetOps Performance Management and provides dashboards and views designed for modern network monitoring. For information, see Modern Network Monitoring.

Key Solutions

DX NetOps Performance Management includes the following integrated solutions:

- DX NetOps Mediation Manager Documentation
  Integrate non-SNMP monitoring in your CAPM environment.
- Network Flow Analysis Documentation
  Monitor interface performance and bandwidth utilization with insight into the traffic going through your network.

The following solutions contribute key features to DX NetOps Performance Management:

- DX NetOps Spectrum Documentation
  Map network topology and create alarms from performance events.
- DX NetOps Virtual Network Assurance Documentation
  Extends traditional network monitoring to the virtual network. Provides modern network monitoring for software-defined architectures and hybrid cloud platforms. Correlates logical network entities with physical resources.

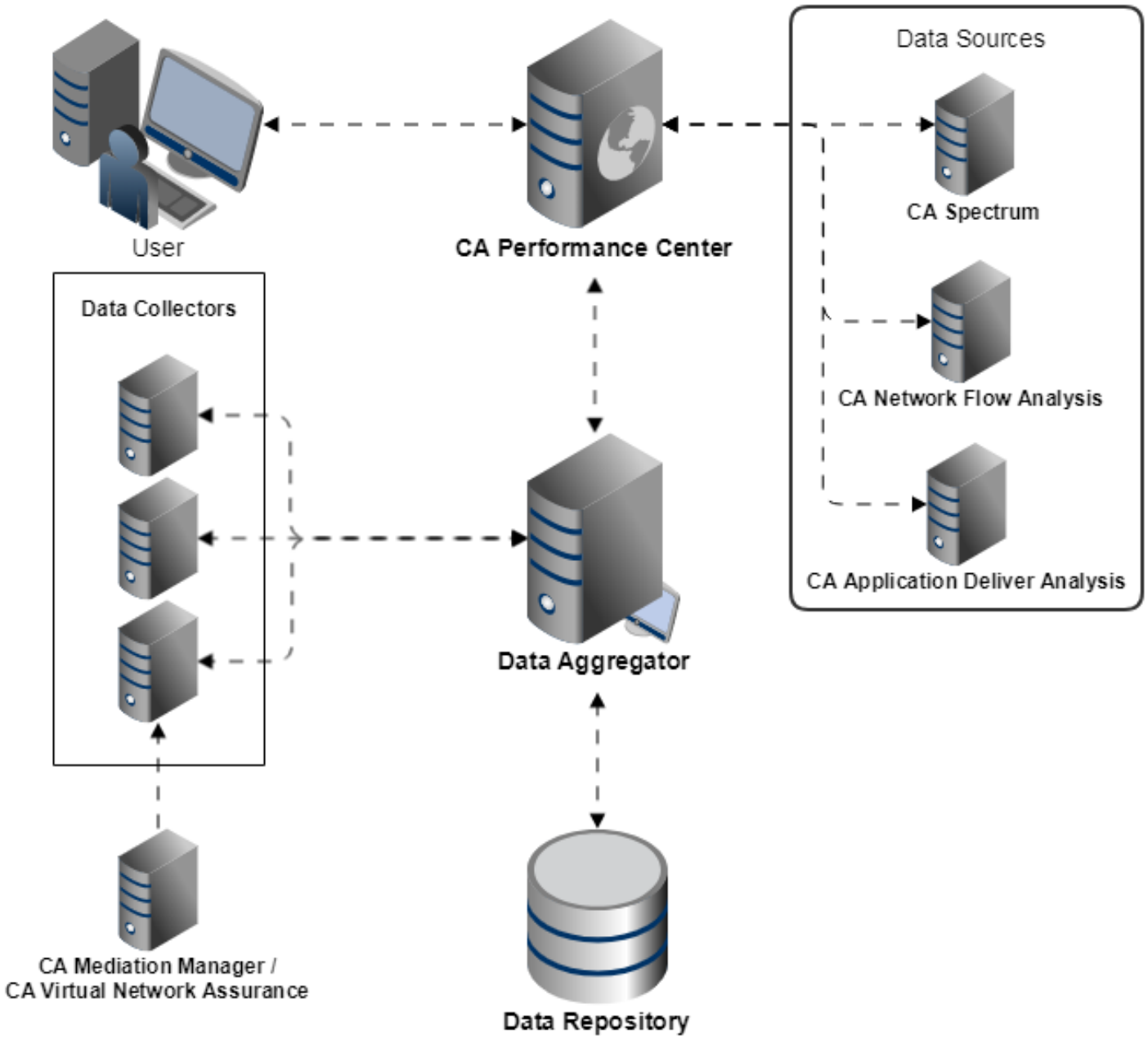
Product Architecture

DX NetOps Performance Management uses an extensible multiserver architecture to support monitoring the largest networks. DX NetOps Performance Management collects performance data and integrates with other data sources, such as CA Application Delivery Analysis and Network Flow Analysis, to provide more information about your network. This page covers the basic system architecture. Fault tolerance enables your DX NetOps Performance Management environment to continue operating properly when a hardware failure or network issue occurs.

The following diagram shows the basic system architecture:

> **NOTE**
> For DX NetOps Performance Management to work properly in a firewall-protected environment, certain ports must be open.

Figure 2: Basic System Architecture



Components

DX NetOps Performance Management includes four primary components:

- **NetOps Portal**
  - Front-end client for the console, dashboards, and reporting
  - Administration and event management
  - Integrates with other data sources
- **Data Aggregator**

  - Data loading and normalization
  - Threshold monitoring
  - OpenAPI
- **Data Repository**
  - Vertica database for Data Aggregator performance data
  - Multi-node (3-5) cluster for large networks
  - Single-node for small networks
- **Data Collector**
  - Simple Network Management Protocol (SNMP) and Internet Control Message Protocol (ICMP) discovery and data collection
  - Integration point for DX NetOps Virtual Network Assurance and DX NetOps Mediation Manager
  - Each Data Collector supports monitoring up to 500,000 monitored items
  - Deploy multiple Data Collectors to support large or geographically dispersed networks
- **DX NetOps Mediation Manager**
  - Collects non-SNMP inventory and performance data
  - Loads data to the Data Aggregator through a dedicated Data Collector
- **DX NetOps Virtual Network Assurance**
  - Collects inventory and performance data from software-defined networking (SDN) and network functions virtualization (NFV) environments
  - Loads data to the Data Aggregator through a Data Collector

Other Architectures

The following architectures reduce data loss:

- Disaster Recovery
- Fault Tolerance

A low-scale architecture is also available. A low-scale architecture matches the basic architecture, but the Data Aggregator and Data Collector are on a single node. If the sizing tool recommends a low-scale deployment, see Install a Low-Scale System.

Videos

The following videos are available for quick overviews about DX NetOps Performance Management. These videos are also embedded throughout the content.

Access Video Transcripts

If you would like to access a video transcript for accessibility reasons, you can access them from YouTube.

Follow these steps:

1. Click the video title to open it in YouTube.
2. Click the **More (...)** icon.
3. Click **Open transcript**.
   The transcript appears in the Transcript pane.

Integrations

The following video examines a DX NetOps Spectrum integration:

For more information, see Integrate CA Spectrum.

The following video examines a CABI integration:

For more information, see Integrate CA Business Intelligence.

The following video examines CA Business Intelligence reports:

For more information, see CABI Reports and Dashboards.

The following video examines the CAMM Netrounds Device Pack:

For more information, see Generate CAMM Device Packs.

Context Pages and Dashboards

The following video examines data sharing:

For more information, see Share Data with Other Users.

The following video examines custom dashboards:

For more information, see Manage Dashboards.

The following video introduces dashboards and context pages:

For more information, see Dashboards and Context Pages.

The following video examines how to set a default dashboard.

For more information, see Customize Your User Settings.

Discovery

The following video examines device discovery:

For more information, see Discovery.

The following video examines more about device discovery:

For more information, see Discovery.

Events and Notifications

The following video examines threshold events and notifications:

For more information, see Events and Configure Notifications.

The following video examines threshold profiles:

For more information, see Configure Threshold Profiles.

The following video examines event processing:

For more information, see Events.

Groups

The following video examines custom groups:

For more information, see Groups.

The following video examines grouping managed items:

For more information, see Groups.

Install and Upgrade

The following video examines the Data Collector upgrade:

For more information, see Upgrade the Data Collectors.

The following video examines the Data Repository installation:

For more information, see Install the Data Repository.

The following video examines a disaster recovery setup:

For more information, see Disaster Recovery.

Manage Devices

The following video examines device types:

For more information, see Override Device Types

The following video examines device deduplication:

For more information, see Device Deduplication.

Monitoring

The following video examines alarm monitoring:

For more information, see Alarms View.

The following video examines SD-WAN monitoring:

For more information, see Monitor SD-WAN.

The following video examines monitoring configuration:

For more information, see Configure Monitoring.

The following video examines Cisco ACI monitoring:

For more information, see Monitor Cisco ACI.

OpenAPI

The following video examines OpenAPI Apps:

For more information, see OpenAPI Apps.

The following video examines the OpenAPI and QueryBuilder:

For more information, see Use the OpenAPI QueryBuilder.

NetOps Portal User Interface

The following video introduces the basic UI features:

The following video examines how to edit user settings:

For more information, see Customize Your User Settings.

The following video examines how to log in:

Network Discovery and Monitoring

DX NetOps Performance Management uses ICMP and SNMP to discover your network and uses SNMP to collect performance data. For information about configuring monitoring, see Configure Monitoring in a New Environment.

Configurable items in the system control discovery and monitoring:

- **Collections** are groups of devices that share monitoring behavior through association with monitoring profiles.
- **Discovery profiles** identify IP address ranges for the Data Collectors to attempt to discover devices. The discovery profile defines which SNMP profiles to use to contact devices in the defined IP ranges.
- **Metric families** are related sets of metrics collected across several technologies. The metric definitions determine how to report the values for the metrics. Metric families normalize performance data from different devices and device types.
- **Monitoring profiles** define the monitoring behavior, poll cycle duration and which metrics to collect, for the associated device collections.
- **SNMP profile** include access credentials to use for discovery and monitoring.
- **Vendor certifications** map the MIB attributes for a particular vendor device to the metrics in supported metric families.

Discovery

During discovery, DX NetOps Performance Management identifies devices in your network.

1. The Data Aggregator assigns IP addresses in the discovery profile IP ranges to Data Collectors.
2. The Data Collectors determines whether devices respond to ICMP or SNMP
3. For devices that respond to SNMP, the Data Collector associates the device with a vendor certification and determines configuration data, which includes the following information:
   - The classification of the device, such as router or switch
   - The device vendor, such as Cisco or Juniper
   - The device type, such as 7700 or 8200
4. Group rules add the devices to collections.
5. The Data Collector uses information in the vendor certifications to identify components for each device.

Monitoring

During operation, the Data Collectors query devices using SNMP MIB requests.

1. For each device, the Data Collector consolidates information from each monitoring profile that is associated with each collection that the device belongs to.

2. The Data Collector requests each supported metric in each monitoring profile at the fastest rate among all monitoring profiles for the device.
3. The Data Collector staggers SNMP requests within the time of the poll cycle.
4. The Data Collector batches poll responses and send the messages to the Data Aggregator.
5. The Data Aggregator loads the performance data to the Data Repository.

Non-SNMP Inventory and Performance Data

DX NetOps Mediation Manager and DX NetOps Virtual Network Assurance collect inventory and performance data and load that data to DX NetOps Performance Management through the Data Collectors. If DX NetOps Performance Management discovers and monitors the same devices through SNMP, the Data Aggregator deduplicates devices with the inventory collected from DX NetOps Mediation Manager and DX NetOps Virtual Network Assurance.

DX NetOps Mediation Manager is the standard non-SNMP data collection tool. For more information, see the DX NetOps Mediation Manager documentation.

DX NetOps Virtual Network Assurance is the data collection and normalization tool for SDN and NFV controllers and orchestrators. For more information, see the DX NetOps Virtual Network Assurance documentation.

Determine Monitoring Requirements

Before you configure monitoring in a new environment, use the following guidelines to determine the monitoring requirements of the environment.

To determine the monitoring requirements of the environment, complete the following steps:

After you determine the monitoring requirements, configure monitoring.

Identify Data Sources

During the installation, you register the Data Aggregator as a data source. Many implementations of DX NetOps Performance Management include other data sources, such as DX NetOps Spectrum and CA Application Delivery Analysis. To import inventory from other data sources, you must register those data sources.

Plan Discovery

Discovery is the process that DX NetOps Performance Management uses to build an inventory of devices in the network. You can define SNMP profiles with the authentication credentials to communicate with devices in the network and use discovery profiles to limit discovery.

Plan Collections and Monitoring Profiles

Collections are system groups that group devices for monitoring. Monitoring profiles control how often to poll devices and which metrics to collect. Associating a collection with a monitoring profile causes DX NetOps Performance Management to monitor the devices according to the parameters in that profile. You can identify a basic set of monitoring profiles and collections up front to reduce maintenance overhead.

Plan Metrics

You can assign metric families to each monitoring profile to specify the metrics that the system should collect. Consider the metrics that are most useful for monitoring the environment.

Example: You might plan to collect the following metric families:

- Interface
- CPU
- Memory
- Disk
- IPSLA
- QoS

Plan Poll Rates

In each monitoring profile, you can specify how frequently to poll for specific metrics.

Example: You might plan to collect metrics at the following poll rates:

- Interface (1 minute)
- Interface (5 minutes)
- CPU (5 minutes)
- Memory (5 minutes)
- Disk (5 minutes)
- IPSLA (5 minutes)
- QoS (5 minutes)

Plan Monitoring Profiles

You can create a matrix like the following example to determine the necessary monitoring profiles:

Polling Rate	Interface	CPU	Memory	Disk	IPSLA	QoS
1 minute	yes	no	no	no	no	no
5 minutes	**yes**	**yes**	**yes**	**yes**	yes	yes
15 minutes	no	no	no	no	no	no

You can then use the planned metrics and poll rates to plan the monitoring profiles.

In this example, the administrator plans the following monitoring profiles:

- Fast Interfaces (includes Interface metrics that are polled at 1 minute)
- Standard (includes Interface, CPU, Memory, and Disk metrics that are polled at 5 minutes)
- IPSLA (includes IPSLA metrics that are polled at 5 minutes)
- QoS (includes QoS metrics that are polled at 5 minutes)

You can use monitoring profile filters to refine which managed items the monitoring profile applies.

Plan Collections

Consider each of the managed items as you plan the collections.

In this example, the administrator plans the following collections:

- Fast Interfaces (including WAN links) for association with the Fast Interfaces monitoring profile
- Active Devices (including all active devices with some exceptions) for association with the Standard monitoring profile
- Core Routers (including routers in core) for association with the IPSLA monitoring profile
- Distributed Routers (including routers in the distributed network) for association with the QoS monitoring profile

You can use group rules to refine which managed items the collection applies.

Plan Monitoring Profile Filters

Monitoring profile filters specify criteria that governs which components are monitored. Only the component items that match the filter criteria are polled for the associated metric family. Filtering limits SNMP traffic and ensures that the system monitors only relevant components. The filters of each monitoring profile are assessed independently.

Plan Group Rules for Collections

Use rules to keep the collections up-to-date when systems and networks change. Newly discovered items that meet rule specifications are added to collections. If existing items no longer meet rule requirements, they are removed from collections. After you create a rule, you can modify it by deleting filters or adding subrules.

Plan Events

An event is a message that provides information about what is happening in DX NetOps Performance Management. Events provide information for monitoring the health and status of your system and your environment. All events include basic information, such as related devices and the time of the occurrence that triggered the event.

For more information, see Events.

Plan Threshold Profiles

Threshold profiles trigger events when specified conditions occur in associated groups. Event rules define the conditions that trigger events. Each event rule is set to a single metric family, and determines the conditions that cause or clear a violation. Each threshold profile requires at least one event rule.

Plan Monitoring Profile Event Rules

For thresholds that apply broadly to devices in the network, you can add event rules at the monitoring profile level. For example, a rule that creates an event whenever CPU utilization exceeds 95 percent could apply to any device.

Plan Dashboards

Dashboards contain sets of views that show you polled data as meaningful information. DX NetOps Performance Management includes several out-of-the-box dashboards that provide basic information about your infrastructure. To set up dashboards that match your specific monitoring requirements, create a custom dashboard or edit an out-of-the-box dashboard.

Plan Reports

You can access and share On-Demand reports, which dynamically retrieve the most recent data sets from specific sets of items or groups. You can also access and share dashboards and views.

Configure Monitoring in a New Environment

After you install DX NetOps Performance Management, determine the monitoring requirements of your environment, and configure the system to monitor your network. For information about determining the monitoring requirements of your environment, see Determine Monitoring Requirements.

This article discusses the basic procedures and best practices for monitoring. These procedures represent the simplest methods to begin monitoring. Each procedure includes links to pages that provide more information and details about complex configurations. Many of these procedures require the Administrator role.

Register Data Sources

During the installation, you registered the Data Aggregator as a data source. Many implementations of DX NetOps Performance Management include other data sources, such as CA Spectrum and CA Network Flow Analysis. To import inventory from other data sources, register those data sources. If this installation includes only the Data Aggregator, skip this procedure.

For more information about data sources and synchronization, see Manage Data Sources.

Follow these steps:

1. Log in to NetOps Portal as an administrator user. Access NetOps Portal at the following URL:
   `PC_host:8181/pc/desktop/page`
2. Hover over **Administration**, and click **Data Sources: Data Sources**.
3. Click **Add**.
4. Select the Source Type, specify the required information, and click **Save**.
   DX NetOps Performance Management synchronizes with the data sources and adds the relevant devices.

Quickly Discover SNMP Devices

Discovery is the process that DX NetOps Performance Management uses to build an inventory of devices in your network. You can quickly discover SNMP devices without having to configure SNMP profiles and discovery profiles manually.

For more information, see Discovery.

Follow these steps:

1. Hover over **Administration**, and click **Monitored Items Management: Quick Device Discovery**.
2. Select **SNMP devices**.
3. Select the SNMP protocol to use.
4. Complete the following fields:
   - **IP Domain**
     If you have multiple IP domains, select the IP domain for discovery.
   - **Community Name** (SNMPv1/v2c Only)
     Specify a secure string that lets the data source query the MIB of the associated device. The community that you supply must provide read access to the device MIB.
   - **Verify Community Name**
     Specify the Community Name again for verification.
   - **Port**
     Specify the port that is used to make SNMP connections to your devices.
     Default: 161

     NOTE

     This port can also be used to send SNMP traps to trap receivers associated with this profile through notifications. In this scenario, use 162 by default. For more information, see Configure Notifications.
   - **IPs/Host**
     Specify the IP address ranges that you want to discover for IPv4. Range discovery is not supported for IPv6 addresses.
5. Click **Discover**.
   DX NetOps Performance Management discovers the devices within the specified IP addresses and hostnames, and adds the devices to the system inventory.

Quickly Discover Virtual Network Devices

Discovery is the process that DX NetOps Performance Management uses to build an inventory of devices in your network. You can quickly discover Virtual Network devices without having to configure discovery profiles manually.

For more information, see Discovery.

Follow these steps:

1. Hover over **Administration**, and click **Monitored Items Management: Quick Device Discovery**.
2. Select **Virtual Network Devices**.
3. Select a technology.
4. Complete the following fields for the selected technology:
   - **Viptela SD-WAN**
     - **IP Address**
     - **Protocol**
       The HTTP security scheme for vManage
     - **Port**
     - **User Name**
     - **Administrator Password**
     - **Re-enter Administrator Password**
   - **Cisco ACI**
     - **APIC Host IP Address**
       The IP address of the APIC controller host
     - **Protocol**
       The communication protocol with the APIC controller
     - **Port**
     - **User Name**
     - **Administrator Password**
     - **Re-enter Administrator Password**
   - **128T**
     - **Host Conductor IP Address/Name**
       The IP address of the 128T Conductor
     - **Protocol**
       The HTTP security scheme for 128T SD-WAN
     - **Port**
     - **User Name**
     - **Administrator Password**
     - **Re-enter Administrator Password**
   - **Nuage**
     - **VSD Host IP Address**
       The Virtualized Services Directory (VSD) host
     - **Protocol**
       The communication protocol with the VSD
     - **VSD Port**
       The port that the VSD UI/API server listens on
     - **Stats Host IP Address**
       The IP address of the stats server
     - **Stats Protocol**
       The communication protocol with the stats server
     - **Stats Port**

The port the stats server is listening on for REST requests
- **Nuage Organization**
  The name of the Nuage enterprise to manage
- **Time Zone**
  The time zone of the system, which must match the VSD time zone
- **User Name**
- **Administrator Password**
- **Re-enter Administrator Password**

– **SilverPeak**
- **Host Orchestrator IP Address/Name**
  The Unity Orchestrator Management host
- **Protocol**
  The communication protocol with the Orchestrator
- **Port**
- **User Name**
- **Administrator Password**
- **Re-enter Administrator Password**

5. Click **Discover**.
   DX NetOps Performance Management discovers the devices within the specified IP addresses and hostnames, and adds the devices to the system inventory.

Discover Devices

If you do not use quick discovery, you can configure SNMP profiles and discovery profiles manually.

The following video examines the detailed steps that are required to discover devices:

For more information, see Discovery.

Configure SNMP Profiles

SNMP profiles provide authentication credentials to communicate with devices in your network.

Follow these steps:

1. Hover over **Administration**, and click **Configuration Settings: SNMP Profiles**.
2. Click **New**.
3. Complete the fields, and change any default settings. Some fields apply only to SNMPv3.
   For complete details, see SNMP Profiles.
4. Click **Save**.
   The SNMP profile is added to the system and used for discovery and polling.

Create Discovery Profiles

Discovery profiles specify which devices DX NetOps Performance Management discovers. Create granular discovery profiles for devices with different SNMP credentials or different rediscovery schedule. Granular discovery profiles reduce unnecessary SNMP requests.

For more information, see Discovery Profiles.

Follow these steps:

1. Hover over **Administration**, and click **Monitored Items Management: Discovery Profiles List**.
2. Click **New**.

3. Specify a name for the profile.
4. Specify the IP address, IP ranges, or hostnames to target for discovery.
5. (Optional) Open the **SNMP** tab, select **Use specific list of assigned SNMP profiles**, and select the SNMP profiles to include in discovery.
   Using a specific list of SNMP profiles reduces unnecessary SNMP requests.
6. (Optional) Open the Schedule tab, and define a schedule.
   During normal operation, discovery runs on a schedule basis to discover new devices in the target range.
7. Click **Save**.
   DX NetOps Performance Management uses the discovery profile to find devices in your network.

Run Discovery

To build your inventory, use the discovery profiles to run discovery.

Follow these steps:

1. Select the discovery profile.
2. Click **Run**. You can run discovery only if the State is Ready.
   DX NetOps Performance Management discovers the devices within the specified IP addresses and hostnames, and adds the devices to the system inventory.
   To view a list of the discovered devices, select the discovery profile, and click History.

Configure Monitoring Profiles

Monitoring profiles control how often to poll devices and which information to collect. Metric families are related sets of metrics that are collected across several technologies. The metric definitions determine how to report the values for the metrics. Metric families normalize performance data from different devices and device types. Assigned metric families determine which metrics the system collects. Collections are system groups that group devices for monitoring. Associating a collection with a monitoring profile causes DX NetOps Performance Management to monitor the devices according to the parameters in that profile.

For more information, see Configure Monitoring Profiles.

Configure a Collection

Consider the following best practices for organizing devices into collections for monitoring:

- Create custom collections that match the monitoring requirements in the environment.
  - Consider the different layers of the network, access, distribution, and core. Devices in different layers might require different levels of monitoring.
  - Consider which technologies and metric families are required. Metric families that would be applied to all devices, such as CPU and memory, apply to broad collections. Targeted monitoring, such as QoS and IPSLA, apply to limited collections.
- Create collections that enable the flexibility to break out monitoring.
  - Some devices are included in multiple collections so that specific metric families are polled at different rates.
  - Devices in different collections have different filtering criteria.
  - Different monitoring requirements depending on importance of device

Follow these steps:

1. Hover over **Administration**, and click **Group Settings: Groups**.
2. Select the **Collections** folder in the left pane.
3. Click **Add Group**.
4. Specify a name, and click **Save**.
   The collection is created. To add devices to the collection, add rules.

5. Click the **Rules** tab, and click **+ Add Rule**.
6. Specify a rule name, select devices for the item type, add conditions as required, and click **OK**.
7. Click **Save and Run Rules**.
   DX NetOps Performance Management adds the items to the collection.

Create a Monitoring Profile

Consider the following best practices for creating monitoring profiles:

- Copy default monitoring profiles to create customer-specific monitoring profiles. Remove the collections that are applied to the default monitoring profiles. This model makes it easier to customize and manage device polling.

  NOTE

  Do not modify the DA Health Fast, Normal, and Slow monitoring profiles. These profiles are used for self-monitoring of the DX NetOps Performance Management components.
- For monitoring flexibility, use multiple monitoring profiles. Do not add all metric families that you monitor in your network to a single monitoring profile.
- Some metric families, such as CPU and Memory, apply broadly to all devices. You can apply a monitoring profile with these metric families to all-encompassing collections, such as All Routers or All Managed Devices.
- Monitoring profiles control the poll rate. To poll the same metric family on different devices at different rates, create monitoring profiles with different poll rates. Apply the faster poll rate to a collection that includes only the devices that require fast polling.
- Monitoring profiles control filtering. If different filtering criteria is required for different sets of devices, create a monitoring profile for each set of requirements.
- Apply only metric families that are applicable to the items in the associated collections. For example, do not apply a monitoring profile that contains the VMWare metric families to a collection of routers. Unnecessary metrics add processing to the Data Aggregator at every change detection period.

Follow these steps:

1. Hover over **Administration**, and click **Monitored Items Management: Monitoring Profiles**.
2. Click **New**, or select an existing profile and click **Copy**.
3. Specify a unique profile name and description.
4. Select a poll rate.

   TIP

   Five minutes is the standard poll rate for most devices in most networks. One-minute polling adds strain to the system and increases SNMP traffic.
5. To assign the metrics that the system collects for your devices, select metric families.
   Example: To report and view port information for your devices, select the Interface metric family.
6. Click **Save**.

Assign the Collection to the Monitoring Profile

Follow these steps:

1. Select your monitoring profile from the list.
2. Click the **Collections** tab.
3. Click **Manage**.
4. Select your collection, and click **Save**.
   DX NetOps Performance Management uses the metric families in the monitoring profile to poll the items in the collection.

After you configure monitoring, configure reporting. For more information, see Configure Reporting in a New Environment.

Configure Reporting in a New Environment

After you configure monitoring, configure reporting. Reporting in DX NetOps Performance Management includes dashboards and threshold alerts. Thresholds and dashboards scope reporting to items in groups.

Group Polled Items

Groups organize items logically for reporting and thresholding. When you view a dashboard, the data is scoped to your selected group. Individual user profiles are also scoped to particular groups. Each profile is mapped to an initial group that is the selected context when that user logs in.

> **TIP**
>
> Do not assign All Groups as the default for any users. Because this group includes all items in the system, this context causes dashboards to load slowly.

Organize your group structure according to business and reporting needs. To create a regional structure that represents regions, countries, and locations, use site groups. Use custom groups for other types of organizations, such as customers, services, or technologies.

Threshold profiles apply threshold rules to all items in a group. The group hierarchy requirements for thresholding are probably different from the requirements for reporting. Create separate groups that address both sets of requirements. Consider the different layers of the network and how to create thresholds for components in those layers. For example, you might threshold on CPU, memory, and interface metrics on the core network differently to the distribution layer. Create multiple groups to apply threshold rules appropriately.

For more information, see Manage Groups.

Create a Group

Follow these steps:

1. Hover over **Administration**, and click **Group Settings: Groups**.
2. Select a location in the group hierarchy.
3. Click **Add Group**.
4. Specify values for the following parameters:
   - **Group Name**
   - Select **Custom** from the **Group Type** list.
     If you are creating a Site group, select **Site** from the **Group Type** list.
     To create a Site group, specify the following parameters:
     - Latitude, Longitude, and Elevation
     - Name of the geographical location
     - Time Zone
     - Business Hours
   - **Description**
5. Confirm the setting for the following parameter:
   - **Include the children of managed items**
     Adds the children of managed items automatically when the items are added to this group. If you disable this option and you add a router to the group, the interfaces on that router are not included. Therefore, their data is not visible in drilldown views.
6. Click **Save**.
   The new group appears in the Groups tree.

Configure Group Rules

To populate the groups, configure group rules.

Follow these steps:

1. Select the group that you want to add rules to.
2. Click **New Rule**.
3. Select the item type to add to the group.
4. Click **Add Condition**, and define the condition.
   - (Optional) To add 'OR' matches, click + at the end of the condition.
   - (Optional) To add 'AND' matches, click **Add Condition**.
5. (Optional) Add more rules. Each item type requires a separate rule.
6. Click **Save**.
7. (Optional) To confirm that the new rule includes the correct items, click **Preview Results**.
   The results are shown in the Group Rules Preview window. You can expand each item type to view the items that were added.
8. Click **Save** and **Run Rules**.
   DX NetOps Performance Management saves the rules and populates the group with the specified items.

Build Dashboards

Dashboards contain sets of views that show you polled data as meaningful information. DX NetOps Performance Management includes several out-of-the-box dashboards that provide basic information about your infrastructure. To set up dashboards that match your specific monitoring requirements, create a custom dashboard or edit an out-of-the-box dashboard.

The following video shows how you can customize or create dashboards and context pages to meet your requirements:

Create a Dashboard

If an out-of-the-box dashboard is close to your requirements for a custom dashboard, copy the dashboard as a template for the new dashboard.

Follow these steps:

1. Open the dashboard that you want to copy.
2. Click **More** in the upper right corner, and click **Copy Dashboard**.
3. Select the dashboard menu where you want the copied dashboard to appear.
4. Specify the name for the dashboard in the dashboard menu.
5. Specify the title that appears at the top of the dashboard page.
6. Click **Save**.
   A copy of the dashboard is created. The new dashboard opens.
   Click **More** in the upper right corner, and click **Edit Dashboard**.

If none of the existing dashboards are a good template for your requirements, create a dashboard.

Follow these steps:

1. Click the **Dashboards** tab.
   A list of available dashboards appears. Each pane on the page corresponds to a menu.
2. Click **Add Dashboard** next to the menu where you want the new dashboard to appear.

Customize the Dashboard

Customize the new dashboard to provide the required monitoring information.

Follow these steps:

1. Complete the following fields:
   - **Dashboard Menu**
     The menu where the dashboard appears
   - **Menu Item**
     The name of the dashboard in the menu
   - **Dashboard Title**
   - The name that appears at the top of the dashboard
2. Select a layout template for the dashboard.

   TIP

   Some views, such as scorecard views and MultiTrend views, include a lot of detail and require more screen space. These views do not render well in layouts with more than one column.
3. Click and drag views to the page layout. The maximum number of views per dashboard is 25.
4. To customize the view settings, click the **Edit** (gear) icon for the view.
   For more information about configuring views, see Customize Views.

   TIP

   To limit the list of views, click **Select Context**, and select a group, item, or device. Views that you add to the layout are pinned to the selected context.
5. Click **Save**.
   The dashboard is saved, and is added to the selected menu.

Create Threshold Profiles

Threshold profiles trigger events when specified conditions occur in associated groups. Event rules define the conditions that trigger events. Each event rule is set to a single metric family, and determines the conditions that cause or clear a violation. Each threshold profile requires at least one event rule.

The following video shows the threshold profile configuration process:

For more information, see Configure Threshold Profiles.

NOTE

For thresholds that apply broadly to devices in your network, you can add event rules at the monitoring profile level. For example, a rule that creates an event whenever CPU utilization exceeds 95 percent could apply to any device. For more information, see Configure Monitoring Profiles.

Create a Threshold Profile

Follow these steps:

1. Hover over **Administration**, and click **Monitored Items Management: Threshold Profiles**.
2. Create a folder, or select an existing folder.
3. Click **New Profile**.
4. Specify the required information.
5. Add event rules to the profile.
   a. In the Event Rules pane, click **New**.
   b. Specify the required information for the event rule. The following fields require explanation:
      - **Duration**

Specifies the total amount of time a given condition must be true within the specified Window to generate an event. The poll cycles that trigger the condition do not need to be consecutive.

- **Window**
  Specifies the overall range of time to evaluate the rule condition.
- **Aggregation**
  Specifies whether the threshold applies to an aggregate value of all components for the device. This field appears only when you select a supported metric family.

  NOTE

  Currently, only the Utilization (%) Metric for the CPU and Memory metric families are supported for aggregation. When you select this option, the event rule must use Fixed Value for the Condition Type.

c. Save the event rule.

6. Click **Save**.
   The threshold profile is added to the system. To generate events, assign the profile to a group.

Assign Groups to a Threshold Profile

Follow these steps:

1. Click the **Groups** tab in the right-hand pane.
2. Click **Manage** at the bottom of the screen.
3. Select the groups from the **Available Groups** tree, and click the right arrow to add it to the Selected list.
4. Click OK.
   The groups in the Selected list are assigned to the threshold profile. DX NetOps Performance Management applies the threshold rules to items in the assigned groups and creates events.

Get Started as a New User

NetOps Portal is a web-based interface that helps you manage your physical and virtual networks, applications, and devices. The NetOps Portal context pages, dashboards, and reports show performance data from your network and systems-monitoring products (data sources). You can compare large amounts of statistical data from multiple sources in one web page.

NetOps Portal takes a "performance-first" approach to application service delivery. This approach places end users in the primary role. To understand how well an IT organization supports application delivery, use DX NetOps Performance Management to capture and analyze data from applications, devices, and the network.

NetOps Portal offers role-specific views of application response times, traffic composition, infrastructure health, and flow-based diagnostics.

Also, modern network monitoring is available from NetOps Portal with DX NetOps Virtual Network Assurance. NetOps Portal with DX NetOps Virtual Network Assurance enables comprehensive coverage with monitoring that is scalable and heterogeneous across the greatest number of technology stacks in the following architectures:

- Traditional
- SDN
- SDDC
- SD-WAN
- NFV
- Hybrid-cloud

The following video highlights several key features of the NetOps Portal UI:

In this article:

Customize Your User Settings

Each user account provides customization options for your personal settings, such as your preferred language for the NetOps Portal UI.

Follow these steps:

1. Click the link for your user account in the upper-right corner and click **User Settings**.
2. Modify the following user settings and click **Save**:
   - **Preferred Language**
     Specify a language for the NetOps Portal user interface. NetOps Portal displays the selected language regardless of the language that is selected for the operating system or for the browser language.

     NOTE
     For a language to display appropriately, the relevant fonts must be installed.
   - **Email Address**
   - **Time Zone**
     The default time zone is UTC (Coordinated Universal Time).

     NOTE
     Changing the time zone after email schedules are set up in NetOps Portal can cause incorrect times to appear in the Scheduled Emails UI.
   - **Time Display Format**
     Select the default time format, either 12 hours or 24 hours.
   - **Default Group**
     This group is the default context when you log in. The list only includes groups from your permission groups.
   - Select one of the following options from the **View Suppression** drop-down:
     - **Suppress Views**
       View suppression is enabled and views are hidden.
     - **Display All Views**
       View suppression is disabled and all views appear.
   - Select one of the following options from the **Item Name Display Setting** drop-down:
     - **Use Display Name**
     - **Use Item Name Alias**

For more information, see Customize Your User Settings.

Explore Managed Items

Data sources discover and monitor your managed items (for example, applications, devices, or interfaces). After monitoring is configured, you can explore your managed items using the Inventory pages or search as a launch point. For more information, see Synchronize Data Sources and Configure Monitoring in a New Environment.

Navigate the Inventory

All the managed items that you have permission to view are available from the Inventory. From the Inventory, you can navigate by item type category (for example, Devices) to lists of those managed items. You can also drill down to the context page of an individual item for more details.

Follow these steps:

1. Hover over **Inventory**, and then click the item type category to view.
   The Inventory for the selected item type category provides minimal information for each item, such as device hostnames or IP addresses.

NOTE
The list for the selected item type category contains a maximum of 5,000 items. If the number of managed items for the selected item type category exceeds 5,000, use the search filter in the lower-left corner.

2. Explore the Inventory list:
   – To sort by a column, click the column heading.
   – To add or remove columns, hover over a column heading, and then click the gear icon. Hover over **Columns**, and then select or clear columns.
3. To drill down to the context page of an individual item, click the item in the list.

For more information, see Inventory Pages and Views.

Search for Managed Items

The global-level search box lets you search for text that is contained in an item string. The search returns inventory lists of all the managed items that match your search, which are sorted by item type category.

NOTE
If your search string includes the equal sign (=), you must begin your search string with an equal sign (=).

Example:`=StringWith=Sign`

Follow these steps:

1. Enter a search string in the global search box in the upper-right corner of NetOps Portal. Press **Enter**:
   The search results categorize the items by type.

   TIP
   To narrow or broaden your search, add the asterisk (*) wildcard character to the search field. The asterisk is the only supported wildcard character. Using asterisks at the beginning and end of a keyword work like quotation marks in the search. You can add multiple search words to narrow the search. If you search for devices using the string “server 192.168*”, the search returns all servers on the 192.168.0.0/16 network.
2. To drill down to the context page for an item, click the item in the list.

For more information, see Search and Filter in CA Performance Center.

View Performance Data

You can see performance data on dashboards and context pages.

- **Dashboard Pages**
  Provide performance and status data that is scoped to a group. For example, a dashboard page can provide the average performance of monitored items in a group. Dashboards often provide a drill-down path to more detailed, related pages from a selected context.
- **Context Pages**
  Provide focused performance and status data that is scoped to a specific managed item, such as a single router or server. These pages are available as drill-down links or tabs from dashboard pages.

Dashboards and context pages render views, which report collected data in a chart or a table format. Depending on the view, the data comes from the various registered data sources. Views that show data for a group contain collated and aggregated data from data sources. Views that show data for individual items provide a drill-down path to the context page for the item.
For more information, see Views.

The following video examines dashboards and context pages:

View a Dashboard

You can view and filter the performance data on a dashboard with group contexts and time ranges.

The group context lets you filter the data that appears in views on the dashboard. When you select a group for a standard dashboard, you apply a filter to all views on the page. When you select a group context, items from all subgroups that are available to you appear in views on the dashboard.

To filter data based on specific time periods, specify time ranges for your dashboards. Changing the time range is useful for troubleshooting performance issues. For example, you can change the time range to show data from the last seven days. In this case, the time range helps you to determine whether an issue is occurring regularly.

Follow these steps:

1. To view a dashboard, hover over **Dashboards**, and then click the dashboard.
2. To filter on the group context, do the following tasks:
   a. Click the **[change]** link under the title of the dashboard.
   b. Select a group from the group hierarchy.
   c. Click **OK**.
   All views on the page with dynamic context are refreshed to reflect the new data context. The change applies until you log out.
3. To filter on a specific time period, do the following tasks:
   a. Click the **[change]** link in the upper-right corner of the dashboard page.
   b. Select a default time period from the list or specify a custom time range.
   The selected time range is applied to the dashboard.

For more information, see Dashboards.

View a Context Page

You can access context pages from an inventory list or a dashboard. Unlike standard dashboard pages, item context pages are clustered in sets of tabbed pages. You can edit the predefined tabs and can change the views that are displayed on those tabs. You can add tabbed pages. You can also rearrange the tabs in an item context to change their order.

Follow these steps:

1. To view a context page, from an inventory list or dashboard, do one of the following tasks:
   – Right-click a hyperlink on an item and select a context page tab.
   – Click a hyperlink on an item to open the default context page tab.
2. To manage tabs, click the **Edit** icon in a tab, select one of the menu options, and then edit, delete, add, reorder, or restore tabs as desired.

For more information, Context Pages.

Access and Share Reports

You can access and share On-Demand reports, which dynamically retrieve the most recent data sets from specific sets of items or groups. You can also access and share dashboards and views.

For more information, see On-Demand Reports and Share Data with Other Users.

Access and Share an On-Demand Report or Dashboard

You can create or access reusable On-Demand report templates, and download a report for sharing. You can also access a dashboard and download it for sharing.

Follow these steps:

1. Do one of the following tasks:
   – Hover over **Reports**, click **On-Demand Report Templates**, and then run an existing report template.
   – Hover over **Dashboards**, and then select a dashboard.
2. Select one of the following sharing options:
   – Click **Print**, select one of the following options, and then download and share the file:
      • **Print PDF**
        Select **Portrait** or **Landscape** to specify the page layout of the PDF document.
      • **Print CSV**
        Select **Scaled** or **Unscaled** to specify whether the values in the exported dashboards are scaled. Scaled values appear with larger units, for example, 1 KB. Unscaled values appear in the raw form for the metric, for example, 1000 bytes.
   – If an email server is configured, click **Email/Schedule Report**, complete the email parameters, and then click **OK**.

Access and Share a View

You can download and share individual views.

Follow these steps:

1. Click the **Gear** icon in the upper-right corner of a view.
2. Select one of the following options:
   – **Export CSV (scaled)**
     Scaled values appear with larger units, for example, 1 KB.
   – **Export CSV (unscaled)**
     Unscaled values appear in the raw form for the metric, for example, 1000 bytes.
   – **Generate URL**
     Generate a URL for the view and complete the associated parameters.
3. Do one of the following tasks:
   – Download and share the file.
   – Copy and share the URL.

Organize Managed Items

A group is a filter definition that functions as a container for managed items. Groups let you logically organize managed items in a hierarchical tree structure, with each group containing subgroups or managed items. The structure is propagated to the data sources, where it enables drill down from top-level groups into data from an increasingly narrow but related context.

If the My Custom Groups functionality is enabled for your user account, you can create custom groups. You can define relationships, policies, and dependencies among services, devices, applications, locations, and users within your organization using the Groups tree. Then you can organize your managed items by creating group rules. The groups that appear in your **My Custom Groups** area are visible to only you.

Follow these steps:

1. Do one of the following tasks:
   – Hover over **Administration**, and then click **Group Settings: Groups**.
   – Click the name of your user account in the upper-right corner, and then click **My Custom Groups**.
2. Select a location for the new group in the **Groups** tree, and then click **Add Group**.
3. Specify the parameters for the group, and then click **Save**.
   The new group appears in the Groups tree.
4. Select the group to which you want to add managed items.

Items that have already been added to this group appear in the right pane.
5. Click **New Rule**.
6. Type a name for the rule.
7. Select the type of managed item that you would like to add to the group.
8. Specify the conditions for the rule, and then click **Save**.
9. To confirm that the new rule includes the correct items, click **Preview Results**.
10. Click the following options:
    - **Save**
      Saves the rules without running them. The group is populated during the next global synchronization, which occurs approximately every 5 minutes.
    - **Run Rules**
      Populates the group immediately.

For more information, see Groups.

Customize Your Experience

You can customize your display settings, dashboards, and views.

The following video shows how you can customize or create dashboards and context pages to meet your requirements:

Customize a Dashboard and its Views

You can edit dashboards to add, remove, rearrange, or customize views. You can also click the Gear icon in the upper-right corner of any view to edit its settings.

Follow these steps:

1. Click the **Dashboards** tab, and then select a dashboard to edit.
2. In the upper-right corner, click **More**, and then select **Edit Dashboard**.
3. Change the layout and add or remove views to your layout as desired.
4. To customize a view, click the **Gear** icon to the right of the view name in the layout.
5. Update the view settings and click **Save**.

For more information, see Manage Dashboards, and Customize Views.

Manage Events

An event is a message that provides information about what is happening in DX NetOps Performance Management. Events provide information for monitoring the health and status of your system and your environment. All events include basic information, such as related devices and the time of the occurrence that triggered the event.

To view events, access or add one of the following views:

- **Events View**
  This view displays all the events that occurred in the selected time range for the dashboard. This view can be filtered for a specific group. This view is the default view in the Events Display dashboard.
- **Filtered Event Views**
  This view includes filters for data source, severity, event type, event subtype, and threshold profile.

You can configure notifications for events that come from a data source to the Event Manager. The incoming events are evaluated against the conditions that you configure for the notification criteria. Only when the criteria are met does Event Manager take a notification action. If an event does not trigger a notification, the event can still be displayed in the Event List.

The following video examines events and notifications:

Follow these steps:

1. Do one of the following tasks:
   - Hover over **Administration**, and then click **Configuration Settings: Notifications.**
   - Click the name of your user account in the upper-right corner, and then click **Manage Notifications**.
2. Click **New**, step through the **Create Notification** wizard, and then click **Save**.

For more information, see Events and Configure Notifications.

Telemetry

Telemetry is a capability that is integrated into DX NetOps Performance Management to send product usage and system configuration data to CA Technologies, a Broadcom Company (CA). This data helps CA gain insights into customers' product usage, understand their software needs, and focus on the features and platforms that are most commonly used.

By default, the data is collected and stored in the database if the Opt-in field is set to Yes on the Edit Usage Data Sharing Settings screen or Opt-In is set to true through REST. Users can access the collected telemetry data on the System Health page. The uploaded metrics are also stored in a file that is located under <*PC Install Directory*>/PC/bin/esdplatelemetry/.

WARNING

Telemetry does not collect any personally identifiable information (PII) or sensitive information.

What Data Do We Collect?

The following table describes the data that is provided by customers:

Data	Description
Opt-In (Yes/No)	Send diagnostic and usage data to Broadcom. If you have a Portfolio License Agreement in your contract with Broadcom, specify Yes.
PLA Agreement (Yes/No)	If you have a Portfolio License Agreement in your contract with Broadcom, specify Yes.
Domain name	Your company's domain name (the last part of your company's email address). For example, in xyz@company.com, the domain name is company.com.
Enterprise site ID	Your company's site ID that is listed on the CA Support portal.
Description	(Optional) Specify a department or cost center that you use for internal tracking. For example: IT-Sales-1234.

Data	Description
Use Proxy (Yes/No)	(Optional) Details of the proxy server and user credentials to access the proxy server. The user credentials are required only if your proxy requires it. If api.segment.io is not reachable through port 443 from this machine, you can configure a proxy server to transmit anonymous usage data.Provide these details only when the telemetry service cannot be reached through the outgoing port 443 from the DX NetOps Performance Management. Selecting Yes provides you the following fields to enter proxy server credentials: **Proxy URL** - (Optional)The URL proxy address. We recommend that you use a secure protocol (https) For example: https://myproxy.company.com. **Proxy Username** - (Optional) Specify the username on the proxy to direct usage data through. **Proxy Password** - (Optional) Specify the password on the proxy to direct usage data through.

The following table describes the data that is generated by DX NetOps Performance Management:

Data	Description
version	Specifies the data source version
tenant_count	Specifies the number of tenants that are associated with this data source
domain_count	Specifies the number of IP domains that are associated with this data source
active	Specifies if the data source is enabled
source_id	Specifies the unique identifier for the represented data source
active_device_count	Specifies the number of active devices that are associated with this data source
retired_device_count	Specifies the number of retired devices that are associated with this data source
monitored_item_count	Specifies the count of monitored items that are associated with this data source
consumed_license_count	Specifies the count of the items consuming the licenses
ha_da_enabled	Indicates whether the Data Aggregator is configured for a fault tolerant environment
data_collector_count	Specifies the count of data collectors that are associated with this data source

NOTE

NetOps Telemetry takes into account only the items synced to Performance Center. It does take into account the 200 ports per licensed device coming from DX NetOps Spectrum or the Data Aggregator. CA Application Delivery Analysis, Network Flow Analysis, and DX NetOps Virtual Network Assurance use different licensing count logic because the licensing agreements for those products count differently.

How Frequently Do We Collect the Telemetry Data?

By default, telemetry collects and stores the data daily at 12.00 a.m. If the scheduler is not active at 12.00 a.m., the data is collected only in the next day run. The data is collected only once per day.

How to Configure DX NetOps Performance Management to Collect and Send Telemetry Data to CA?

As an administrator, you can configure DX NetOps Performance Management to collect and send telemetry data to CA either in the DX NetOps Performance Management user interface or using the REST API.

From the DX NetOps Performance Management user interface, perform the following steps:

1. Select Administration, Configuration Settings, Usage Data.
2. Click Change Configuration.
3. Fill out the form.
4. Click Save.

To configure telemetry through REST, post the following to http://<*PCHOST*>:<*Port*>/pc/center/webservice/insights/config

```
<Configure>
<CompanyDomain>ca.com</CompanyDomain>
<EnterpriseSiteID>105246</EnterpriseSiteID>
<InternalIdentifier> IT Sales</InternalIdentifier>
<Opt-In>true</Opt-In>
<PLAAgreement>true</PLAAgreement>
<UseProxy>false</UseProxy>
 </Configure>
```

NOTE

Specify a department or cost center that you use for internal tracking in the `InternalIdentifier` parameter. For example, "IT Sales."

More fields for using a proxy are `ProxyURI` , `ProxyUsername` , and `ProxyPassword` .

How Do You Know if Telemetry Failed to Collect or Send Data?

The `TELEMETRY_FAILURE (553)` event is generated to indicate that telemetry failed to collect or send data.

The `trapTelemetryFail (553)` trap is generated to indicate that telemetry failed to complete successfully.

Transition to Performance Management

If you are transitioning off CA eHealth, consider DX NetOps Performance Management. DX NetOps Performance Management prepares network operations teams to embrace the latest, scalable technology. Transitioning helps to ensure that the capabilities your organization relies on are met. Transitioning also helps to ensure that more value from new enhancements is realized.

The top five reasons to transition to DX NetOps Performance Management include the following items:

1. Visibility into traditional and SDNs for operational simplicity
   DX NetOps Virtual Network Assurance seamlessly integrates with DX NetOps Performance Management to provide the most comprehensive network monitoring solution for traditional, SDN, and cloud networks.
2. Big data architecture at low cost
   Flexible data acquisition and storage controls provide lowered operational costs across network environments of any size.
3. Prescriptive analytics for improved mean time to repair (MTTR)
   Real-time data analysis, baselining and alerting, with guided workflows, provide meaningful and proactive actions to network performance triage.
4. Innovative visualizations for a customized experience
   DX NetOps Performance Management is built on an open API architecture to ensure feature velocity for maximum data value and a truly customized network troubleshooting experience.

5. Open and extensible for cross-domain clarity
   Protocol-agnostic data acquisition helps eliminate silos for cross-domain visibility to deliver a one-pane-of-glass network operations experience.

DX NetOps Performance Management includes advanced network performance monitoring, relationship mapping, and advanced visualizations for improved operational assurance. The following video explains how DX NetOps Performance Management reduces complexity inherent in legacy and modern networks that are built across numerous technology stacks:

The following sections describe the transition process:

For information about transition services, see the Webcast Replay: Finding an easier and faster way to transition from CA eHealth to CA PM.

Evaluate License Entitlements

CA eHealth customers are entitled to 1:1 device licenses for DX NetOps Performance Management.

The following conditions apply:

- Maintenance must be current or brought up to date.
- You must agree to a $0.00 contract with an amendment referencing dual entitlement.
- You can maintain dual licenses for up to 12 months.
- CA eHealth support expires after a period up to 12 months (no later than the end of life date).
- DX NetOps Performance Management should move from test to production within 12 months.
- You must convert old licenses that were purchased by suite or by element to by device.

If you have any questions, reach out to your account team for assistance.

Assess the Current Environment

Before you transition, examine how you are using CA eHealth. Compare the business requirements of CA eHealth to DX NetOps Performance Management. Evaluate the existing reports, performance data collection, and business processes associated with CA eHealth:

- Identify Live Exception monitors configured in your current tool and determine which ones to transition to DX NetOps Performance Management as performance thresholds.
- Identify missing device monitoring certifications.

  NOTE
  DX NetOps Performance Management does not require some of the MIBs that CA eHealth uses for monitoring the same devices.
- Identify and prioritize business, operational, and technical risks, with risk owners and a mitigation plan.
- Identify any product integrations with your current tool.

Assess the Configured Statistics Rollup Schedule

The statistics rollup schedule parameter determines how long raw data, hourly data, and daily data is maintained in CA eHealth. You can find this parameter in CA eHealth OneClick. You can also use this method to obtain this information.

Follow these steps:

1. Start SQL*Plus:
   Windows:
   ```
   sqlplus %NH_USER%/%NH_USER%@%NH_DB_CONNECT_STRING%
   ```

Solaris/Linux:

```
sqlplus $NH_USER/$NH_USER@$NH_DB_CONNECT_STRING
```

2. Enter the following query to retrieve the rollup schedule:

```
SELECT DECODE (RLP_STAGE_NMBR, 0, 'raw data', 1, 'hourly data', 2, 'daily data')
AS DATA_TYPE, DURATION_SIZE/86400 days
FROM nh_rlp_plan
WHERE RLP_TYPE = 'ST';
```

Example Output:

```
DATA_TYPE DAYS
---------- -----------------
raw data 2
hourly data 42
daily data 490
```

The output shows that CA eHealth keeps two days of raw data (also called as-polled data), 42 days of hourly data, and 490 days of daily data.

Obtain Details for Input into the Sizing Tool

To verify that all your servers meet the minimum requirements and sizing guidelines, use the DX NetOps Performance Management Sizing Tool. You can use the following commands to obtain the following details for input into the DX NetOps Performance Management Sizing Tool.

Tips:

- If the CA eHealth systems are in a single cluster, run the following commands from the console for a total count.
- If the CA eHealth systems are in multiple clusters, run the following commands on a console in each cluster and sum the results.
- If one or more standalone CA eHealth systems exist, run the following commands on each standalone system and sum the results.
- If one or more standalone systems exist with a cluster, run the following commands on each standalone system and the cluster console. Sum the results.

Follow these steps:

1. Review the necessary input for the DX NetOps Performance ManagementSizing Tool.
2. Verify that the cluster is in good health and all systems are functioning properly.
3. Run the following commands:

   Number of Devices:

   ```
   nhListElements -elemType device | wc -l > ca-device-elements-count.txt
   ```

   Number of Interfaces:

   ```
   nhListElements -elemType interface | wc -l > ca-interface-elements-count.txt
   ```

4. To get the average number of interfaces per device, divide the number of interfaces by the number of devices.
5. To account for growth, multiply the number of devices by 1.1 and use the result in the DX NetOps Performance Management Sizing Tool.
6. Use the summarized information that is collected for devices in the DX NetOps Performance Management Sizing Tool.

Assess the Certification Coverage

Determine the types of devices that CA eHealth manages and how they map to DX NetOps Performance Management. Work with CA Services to use the `CertAnalyzerPM32_20170731.zip` package.

The tool requires the following input:

- **eHealth-DCI-Location**
  Specify the location of the DCI file. The DCI file is available from the CA eHealth poller directory. If you have multiple DCIs, put them in the same location to run the tool against all of them. To run the tool against only some of your DCIs, put the DCIs into separate folders and specify the folder to run the tool against. The DCI files must have the `.dci` file extension (for example, `elementCfgBackup.dci`). The output files are generated in this location.
- **eHealth-Poller-Version**
  The CA eHealth version number
  Default: 6.3.3.01
- **Other-Version-eHealth-Poller-Location**
  `Unzipped_CertAnalyzer_location``\CertAnalyzer\plugins\capm\def\cert_analyzer.xpp_tmp\EH-POLLER\`*`eHealth_Version`*
- **PM-Certification-Version**
  Specify the DX NetOps Performance Management version number.
- **PM-Certification-Location**
  `Unzipped_CertAnalyzer_location``\CertAnalyzer\plugins\capm\def\cert_analyzer.xpp_tmp\PM-CERT\`*`PM_Version`*
- **Ignore-Aggregation**
  Default: false
- **Ignore-IMP**
  Default: false
- **Ignore-MIB2-Variance**
  Default: true
- **Ignore-Unpolled**
  Default: true

The following results are provided:

- **Summary**
  View the percentage of coverage in DX NetOps Performance Management.
- **Details**
  View each CA eHealth MTF and the DX NetOps Performance Management certification it maps to.
- **Device List Info**
  View each device and its mappings.

The results are also exported into .xls files in the location that is specified in the eHealth-DCI-Location field. Provide the results to your account manager or pre-sales consultant for a summary analysis of your results.

Transition to DX NetOps Performance Management

Before implementation, review the key features, integrations, and architecture. For more information, see Network Discovery and Monitoring.

Execute the transition to DX NetOps Performance Management. We recommend that you take a phased approach. Plan out your group structure and test with a subset first. For more information, see Groups.

Follow these steps:

1. Deploy and configure DX NetOps Performance Management. For more information, see Installing.
2. Execute targeted monitoring setup and configuration migration. For more information, see Configure Monitoring in a New Environment.
   a. To import inventory from other data sources, register those data sources.
   b. Configure SNMP profiles, which provide authentication credentials to communicate with devices in your network.
   c. Configure discovery profiles, which specify the devices DX NetOps Performance Management discovers.
   d. To build your inventory, run discovery.

e. Configure monitoring profiles, which control how often to poll devices and which information to collect.

3. Deploy out-of-the-box reports and in-scope tailored reports. For more information, see Configure Reporting in a New Environment.
   a. Group polled items.
   b. Build dashboards.
   c. Configure performance thresholds in DX NetOps Performance Management that were identified as required from the Live Exception profiles.
4. You might retain CA eHealth to run in parallel with DX NetOps Performance Management. If so, review and approve a cut-over plan with a timeline and project milestones.
5. Get Started as a New User.

Release Notes

The Release Notes section outlines the new features and current and fixed issues for DX NetOps Performance Management. In addition, the release notes contain the Third-Party Software License Agreements, which detail the terms and conditions of using third-party software in the creation of DX NetOps Performance Management. Use the release notes as a starting point when you first install, or upgrade to a new version of DX NetOps Performance Management.

Release Comparison

This table compares the key features in recent releases for DX NetOps Performance Management:

Key Features	Release 20.2	Release 3.7	Release 3.6	Release 3.5	Release 3.2
New Group Editor	**yes**	no	no	no	no
Network Flow Analysis Administration	**yes**	no	no	no	no
Separate Health Dashboard	**yes**	no	no	no	no
Telemetry	**yes**	yes	no	no	no
Extended Flow Views	**yes**	yes	no	no	no
Administer Business Hours Role Right	**yes**	**yes**	yes	no	no
Encrypt Data Aggregator Communication	**yes**	**yes**	yes	no	no
Full PDF and CSV Generation	**yes**	**yes**	yes	no	no
Script Notification Action	**yes**	**yes**	yes	no	no
Support for Oracle Linux (OL) and SUSE Linux Enterprise Server	**yes**	**yes**	yes	no	no
Polling Control	**yes**	**yes**	yes	no	no
Alarms View	**yes**	**yes**	**yes**	yes	no
Customizable Themes	**yes**	**yes**	**yes**	yes	no

Data Aggregator Fault Tolerance	**yes**	**yes**	**yes**	yes	no
Simplified Upgrade for Data Collectors	**yes**	**yes**	**yes**	yes	no
Group Trend Reports	**yes**	**yes**	**yes**	yes	no
System Status Page	**yes**	**yes**	**yes**	yes	no
Red Hat Enterprise Linux (RHEL) 7.4 support	**yes**	**yes**	**yes**	yes	no

New Features and Enhancements

The following new features have been added in this release. Some of the existing features have been improved, and are listed under the Enhancements section.

New Features

New Group Editor

The new and improved group editor no longer uses Adobe Flash.

The following out-of-the box group names changed:

- "Defined Tenants" changed to "Tenants"
- "Collections" changed to "Custom Collections"
- "Domains" changed to "IP Domains"
- "Service Provider Global Groups" changed to "Global Tenant Groups"
- "Service Provider Defined Groups" changed to "Global Tenant Groups"
- "Service Provider Items" changed to "Global Tenant Items"

> **NOTE**
> These name changes impact any existing custom scripts using the previous naming. You must update your scripts accordingly. The DA REST paths have changed.

For more information, see the Groups section.

Network Flow Analysis Administration

This release introduces the ability to manage network flow processing and view the overall heath status of your Network Flow Analysis reporters, harvesters, and interfaces.

For more information, see Flow Administration, Manage Network Flow Processing, View the Health of the System.

Separate Health Dashboards

In previous releases, the Data Aggregator Health Dashboard and Data Collector Health Dashboard were combined into a single dashboard. For clarity, they now appear as separate dashboards.

For more information, see Data Aggregator / Data Collector Health Dashboards.

Enhancements

Data Aggregator Cleanup (20.2.3 and Higher Only)

A cleanup now runs daily by default to remove deleted items from the Data Aggregator. You can configure whether the cleanup runs. You can also disable the cleanup for attribute instance tables or relationships.

For more information, see Configure the Data Aggregator Cleanup.

Data Aggregator Upgrade (20.2.3 and Higher Only)

The `etlHealth.sh` script is now included with the Data Repository installation. We recommend you run this script well in advance of the upgrade and again directly before the upgrade.

For more information, see Upgrade the Data Aggregator.

Password Synchronization (20.2.3 and Higher Only)

NetOps Portal and Network Flow Analysis SSO cookie encryption key changes are now automatically synchronized. You no longer have to restart Network Flow Analysis for the changes to reflect. This enhancement requires you to upgrade Network Flow Analysis to 10.0.4 or higher. This enhancement requires the following patch for Network Flow Analysis 10.0.4: `NFA_10.0.4_PTF_003.`

Access Point Discovery

Previously, the associated wireless controller discovered access points and DX NetOps Performance Management managed them as device components. Access points are now managed as devices available under the device inventory. New metric families are available for Cisco wireless setup. Existing access points remain device components upon upgrade. Make sure to add new metric families for Cisco certification after the upgrade before running any metric family discovery.

If access points restart and their IP addresses change, access points are reconciled. Going forward, access points with matching MAC addresses are reconciled.

In addition, we now support the high availability Cisco setup. If DX NetOps Performance Management discovered both a primary and secondary controller, the access points are moved between the primary and secondary controller when failover occurs.

Lastly, Cisco access points are also reconciled between DX NetOps Spectrum and DX NetOps Performance Management.

Baseline Metrics for SD-WAN Monitoring

Baseline metrics are now available for SD-WAN tunnels and application/SLA paths.

Baseline Rendering

Baseline metrics are processed up to hourly and daily increments. Previously, when the resolution was less than an hour, single dots appeared for the known data points. Now, when the resolution is less than an hour, data points are interpolated between known hourly data points.

For more information, see Trend Views.

Data Collector Statuses

The System Status page now includes new status columns for your Data Collectors. The Configuration Status column indicates whether the associated Data Aggregator can process distributed item repository information from the Data

Collector. The Polling Status column indicates whether the the associated Data Aggregator is receiving poll responses from the Data Collector.

For more information, see View the Health of the System.

Email Extended Flow Views

You can now send or schedule extended flow views in a report by email. By default, the first row is selected in the Top table and the data for the following views are of the first-row selection.

For more information, see Share Data with Other Users.

Enhanced CSV Output

When you generate CSV output, the following leading characters are now removed from cells:

- Plus sign (+)
- Dash (-)
- At symbol (@)
- Equal sign (=)

This enhancement prevents characters that can be interpreted as formula injection.

For more information, see Share Data with Other Users.

Enhanced MySQL Security

As a step toward enhanced security, the Performance Center installation now allows you to set a custom MySQL password.

For more information, see Upgrade Performance Center.

Increased Maximum Alarm Filter Condition Values

You can now have a maximum of 50 overall conditions and 100 alarm filter condition values per condition. For all conditions that support multiple values, you can enable Advanced mode. In Advanced mode, you can enter a comma separated list of values. You can use "and" for all the values or you can select the checkbox to use "or". For more information, see Alarm Views.

Increased Maximum Threshold Value Digits

Threshold values for views now support up to 15 characters including decimal points.

LDAP Authentication for CA Business Intelligence Reports and Dashboards

When you integrate CA Business Intelligence 7.1.1 and higher with DX NetOps Performance Management 3.7.5 and higher, you can enable LDAP authentication for CA Business Intelligence reports and dashboards. For more information, see Enable LDAPS Authentication. You also need to enable LDAP authentication in CA Business Intelligence. For more information, see the CA Business Intelligence JasperReports Server documentation.

Map Views Include All Sites

Sites with no data now appear as gray icons in Map views. In previous versions, sites with no data were excluded from the Map view. For more information, see Map Views.

Meraki Access Point Device Reconciliation

Meraki access points are now reconciled in DX NetOps Performance Management. If the IP address of an access point changes, it is reconciled with the existing access point device in DX NetOps Performance Management. Discovery is initiated on the new IP address of the access point.

Multiple Metric Selection for MultiViews

You can now select multiple metrics within a single metric family for MultiViews. For more information, see Trend Views.

New Custom Gauge View Options

If you are configuring a custom gauge view, you can now select **Scaled** or **Unscaled** to specify whether the values in the view are scaled. Scaled values appear with larger units, for example, 1 KB. Unscaled values appear in the raw form for the metric, for example, 1000 bytes. The new options are unavailable for all existing custom gauge views, which now have "Deprecated" appended to the title.

Non-Root Performance Center Install and Upgrade

When you install and upgrade NetOps Portal, you can now specify a non-root user as the install owner.

For more information, see Install Performance Center and Upgrade Performance Center.

Search Within Specific Columns

Global searches in large-scale environments can impact performance. You can now use the following filter expressions to refine your search results and minimize the impact on performance.

ViewName1:*Column1*;*ColumnN*&*ViewName2*:*ColumnN*=*value*

> **NOTE**
> If you exclude the *ViewName* , the search applies to all views.

```
Groups:Name;Owner&Interfaces:Name;Description=cisco
```

For more information, see Search and Filter in CA Performance Center.

Single Sign-On URL Spoofing Protection

For enhanced security, this release provides enhanced Single Sign-On URL spoofing protection. Administrators can now enable Single Sign-On URL spoofing protection. When this protection is enabled, users can log in using only authorized URLs. The Log In button is disabled for all other URLs. Otherwise, users now get a warning regarding suspicious URLs.

The list of available authorized URLs is populated based on successful log in attempts before protection is enabled.

For more information, see Enable Single Sign-On Spoofing Protection.

Quickly Discover Virtual Network Devices

You can quickly discover Viptela SD-WAN, Cisco ACI, 128T, Nuage, and Silver Peak devices.

For more information, see Discovery.

Vendo Certification Import UI

The user interface for importing custom vendor certifications can now handle extensions, install, and update.

For more information, see Create or Extend Vendor Certifications.

Platform Updates

Jetty

This release upgrades Jetty to release 9.4.15.v20190215

Supported Browsers

The following browsers are supported for this release:

- Microsoft Internet Explorer version 11
- Microsoft Edge version 48.x and later
- Google Chrome 48.x and later
- Mozilla Firefox 42.x and later

New and Updated Technology Certifications

The following certifications are new or updated in this release:

For a complete list of certifications, see Technology Certification Portal.

20.2.5 Certifications

Vendor	Model	Version
CA Technologies	SystemEDGE	
Check Point Software Technologies	Check Point 23500	2.6.18
Cisco Systems	Cisco Catalyst 3850 Stack	3.7.2
Cisco Systems	Cisco UCS 6332-16UP Fabric Interconnect	5.0(3)
Cisco Systems	UCS C240 M5 Rack Server	4.0(4h)
Juniper	Juniper SRX 345	
Juniper	MX 10003	18.2
Netscaler	Citrix NetScaler	11.1
PALO ALTO NETWORKS	PA-3260	
PALO ALTO NETWORKS	PA-5220	
PALO ALTO NETWORKS	PA-7050	
PALO ALTO NETWORKS	PA-850	
Viptela, Inc.	Viptela vEdge2000	

20.2.4 Certifications

Vendor	Model	Version
Check Point Software Technologies	Check Point 4800, 21700	2.6.18
Check Point Software Technologies	Check Point Gaia OS VSX R77.20	2.6.18
Cisco Systems	CAT 6509	
Cisco Systems	Cisco CBS 3110	15.0(2)

Cisco Systems	Cisco FirePower 2110	15.2(4)
Cisco Systems	Cisco ISR4431	16.3.4
Cisco Systems	Cisco NCS55A1-36H-SE Router	6.6.25
Cisco Systems	Cisco3620	15.2.2
F5 Networks Inc	BIG-IPi4600	
Fortinet	FortiGate 100, 60E, 301E, 601E	
Futurex	Futurex Guardian 9000	
Juniper	Juniper SRX 320	
Juniper	Juniper SRX 340	
Juniper	Juniper SRX 345	
Juniper	Juniper SRX 1500	
Netscaler	Citrix NetScaler	63.13
net-snmp	Linux	3.2.32
PALO ALTO NETWORKS	PA-5260	
Raksha Networks Inc.	Thunder 1030S	4.1.0
Versa Networks, Inc	Versa 1000 FWA-5020U-00A1R	

20.2.3 Certifications

Vendor	Model
Check Point Software Technologies	Check Point Gaia OS VSX R77.20
Cisco Systems	CAT 6509
Cisco Systems	Cisco CBS 3110
Cisco Systems	Cisco FirePower 2110
Cisco Systems	Cisco ISR4431
Cisco Systems	Cisco NCS55A1-36H-SE Router
Cisco Systems	Cisco3620
Netscaler	Citrix NetScaler
net-snmp	Linux
PALO ALTO NETWORKS	PA-5260
Raksha Networks Inc.	Thunder 1030S

20.2.2 Certifications

Vendor	Model
Check Point Software Technologies	Check Point 12400
Check Point Software Technologies	Check Point Security Management Server
Cisco Systems	Cisco ASA5585 SSP40
Cisco Systems	Cisco ISR4321
Cisco Systems	Cisco NCS 540

Cisco Systems	CiscoIE200016TCGEP
Juniper	Juniper SRX 320
Silver Peak Systems	SD-WAN System

20.2 Certifications

Vendor	Model
3Com	3Com Switch 4200G 48-Port
Acme Packet	Net-Net OS
Adtran	Netvanta 818
ADVA AG Optical Networking	ADVA FSP150CC-GE112
APCON	4K 4040
Arbor Networks	APS-2600-10G
Arista Networks, Inc. (previous was 'Arastra, Inc.')	Arista DCS-7280SE-64
Aruba Networks/Aruba	8400
Aruba Networks/Aruba	Aruba 6000
Aruba Networks/Aruba	ClearPass
Aruba Networks/Aruba	ClearPass
AudioCodes LTD	M 500 MSBR, MG 800C MSER
AXGATE	AXGATE
Big Switch Networks	Big Switch Controller
Blue Coat Systems (Cache Flow)	Blue Coat Packetshaper
Blue Coat Systems (Cache Flow)	Blue Coat Management Center
Blue Coat Systems (Cache Flow)	CAS S500
Blue Coat Systems (Cache Flow)	SG9000
Blue Coat Systems (Cache Flow)	SG-S200
Blue Coat Systems (Cache Flow)	SG-S200
Blue Coat Systems (Cache Flow)	SG-S400
Blue Coat Systems (Cache Flow)	SG-S400
Broadcom	Accton-AS7326-56X
Brocade Communications Systems	SilkWorm 6510
Brocade Communications Systems	SilkWorm Switch
CA Technologies	SystemEDGE
Check Point Software Technologies	Check Point 12600
Check Point Software Technologies	Check Point 13500
Check Point Software Technologies	Check Point 15400
Check Point Software Technologies	Check Point 21700
Check Point Software Technologies	Check Point 5800A
Check Point Software Technologies	Firewall 64000

Check Point Software Technologies	Security Management Server
Cisco Systems	Cisco 5672UP
Cisco Systems	Cisco ASA5555
Cisco Systems	Cisco ASR 1002-X Router
Cisco Systems	Cisco IronPort Mail Gateway Family
Cisco Systems	Cisco ISR4351
Cisco Systems	Cisco Nexus 5696Q
Cisco Systems/Cisco	3925 K9
Cisco Systems/Cisco	5500 WLC
Cisco Systems/Cisco	8500WLC
Cisco Systems/Cisco	897VAG-LTE-GA-K9
Cisco Systems/Cisco	9508
Cisco Systems/Cisco	9800 Wireless Controller
Cisco Systems/Cisco	ASA5525
Cisco Systems/Cisco	ASR 1001
Cisco Systems/Cisco	ASR 1002
Cisco Systems/Cisco	ASR 1002-X Router
Cisco Systems/Cisco	ASR 9912
Cisco Systems/Cisco	ASR1009-X
Cisco Systems/Cisco	ASR901
Cisco Systems/Cisco	C250 UCS
Cisco Systems/Cisco	Cat 2960 48TTS
Cisco Systems/Cisco	Catalyst 3560CX-12PD-S Switch, Catalyst 3560CX-8PC-S Switch, Catalyst 3560CX Stack
Cisco Systems/Cisco	Catalyst 3850 Stack
Cisco Systems/Cisco	IronPort Mail Gateway Family
Cisco Systems/Cisco	Nexus 7702
Cisco Systems/Cisco	Nexus 7706
Cisco Systems/Cisco	Telepresence IX5000
Cisco Systems/Cisco	TSPri
Cisco Systems/Cisco	TSPriG2
Cisco Systems/Cisco	UCS C220
Clickarrray Networks	ClickArray
Crossbeam Systems, Inc.	Crossbeam C2
Crossbeam Systems, Inc.	Crossbeam C25
Crossbeam Systems, Inc.	Crossbeam C6
Data Security Systems Solutions Pte Ltd	Ezio server
Dell	DRAC 5
F5 Networks Inc	BigIP 2200

F5 Networks Inc	BigIP VCMPGuest
F5 Networks Inc	BIG-IPi4600
F5 Networks/F5	BigIP VCMPGuest
F5 Networks/F5	BIG-IPi4600
FireEye	FireEye Security Applications
ForeScout Technologies, Inc	CounterACT Appliance
Fortinet	FortiGate 12000D
Fortinet	FortiGate 1000D
Fortinet	FortiGate 100D
Fortinet	FortiGate 100E
Fortinet	FortiGate 12000D
Fortinet	FortiGate 1500D
Fortinet	FortiGate 311B
Fortinet	FortiGate 3700D
Fortinet	FortiManager FAZ-VM
Fortinet	Fortimanager FMG-VM-BASE
Fortinet	Fortinet fgtVM64
Fraunhofer FOKUS	NCM-pfSense
Frontier Software Development	NetScout Infinistream 2410H
Frontier Software Development	NetScout Infinistream 4795H
Frontier Software Development	Netscout nGeniusONE Collector Model VI3300
Futurex, LLC	Futurex Guardian 9000
Huawei	AR1220EV
Huawei	AR161F
Huawei	AR6120
Huawei	MA5600T
Huawei	NE 40E-X16A
Huawei	NE9000
Juniper	EX4300
Juniper	Juniper QFX5100-48T-6Q
Juniper	MX10003
Juniper	MX960
Juniper	PTX 3000
Juniper	SRX5800
Juniper	SRX5800
Juniper Networks	IVE PSA5000
Juniper Networks	IVE PSA7000C
Juniper Networks	NScreenSSG550

Juniper Networks	SRX4600
LigoWave	LigoWave
Mediatrix Telecom Inc	Mediatrix C715
Meraki Networks, Inc.	MR42 Cloud Managed AP
MerakiMeraki Networks, Inc.	MX65 Cloud Managed Router
Microsoft	Windows Server
Netscaler	Citrix Netscaler
Netscaler	Citrix NetScaler
net-snmp	Linux
net-snmp	Linux
net-snmp	linux
net-snmp	linux
net-snmp	Linux
net-snmp	linux
net-snmp	linux
Niagara Networks	N2804
Nortel Networks	Alteon Application Switch 5224XL (vADC)
Nortel Networks	Alteon Application Switch 6420
Nortel Networks	Alteon Application Switch VA
PALO ALTO NETWORKS	PA-3060
Palo Alto Networks	PA-800
PALO ALTO NETWORKS	VM series
Panthera Networks, Inc	Alcatel NUAGE-VSC
Panthera Networks, Inc.	Alcatel 7750 SR-12
Raksha Networks Inc.	A10 TH5440S
Riverbed Technologies	Steelhead Appliance
RND	Radware AppDirector 1000 AS2
RND	Radware DefensePro
Ruckus Wireless, Inc.	Ruckus R300
Ruckus Wireless, Inc.	Ruckus R310
Ruckus Wireless, Inc.	Ruckus R500
Silver Peak	NX11700
Silver Peak	Edge
Solera Networks	MUMDCSOLERA
Stonesoft Corp	Stonesoft NGFW Firewall
TELDAT	TELDAT TV BASE VDSL2/ADSL
TippingPoint Technologies	TippingPoint 330
TrendPoint	TrendPoint Systems

Versa Networks, Inc	Versa 1000 FWA-5020U-00A1R
Versa-networks	FlexVNF
ViaVideo Communications, Inc.	Polycom-HDX 8000
Vormetric, Inc.	Vormetric Security Server

New Look and Feel

The Performance Center user interface has a new look and feel for CA Performance Management versions 3.7 and higher.

You might notice the following changes:

Page Header

Search

Global search now appears as an icon with a tooltip. Click the **Search** icon to open a full-width input box with the focus set in it. An **X** icon appears next to the Search box. To search, press Enter. To clear the search, click the **X** icon.

Search Results

Global search results now appear in an overlay below the header instead of on a separate page. The page that you were on before searching is available underneath the overlay. If you click the **X** icon in upper-right, the overlay with the search results closes. The containing group that the search is performed in is shown in the title along with the search term. Individual views do not have locked group indicators.

User Settings

The link to the user settings for the logged in user account is now an avatar badge with the first two letters of the user name.

Notifications

A notification icon now appears in the page header.

Filter Bar

Time Context Picker

The **Live Update Auto-Refresh** option now appears in the time context picker. If the **Live Update Auto-Refresh** option is selected, the end date appears as **Now** and the custom time range options are invalid. If the **Live Update Auto-Refresh** option is unselected, custom time range options and historic options are available.

Group Context Picker

The group context picker has changed slightly. The change link is now a down arrow.

Color

The color palette of the user interface has changed slightly. Status colors remain mostly the same. Chart colors are new and updated.

Typography

The typography has changed slightly. The font color is similar to what appeared previously. The font weights are bolder and the font sizes are larger.

Iconography

The user interface features new and updated iconography. The icons are more rounded, two-colored, and lighter gray.

Grids

In addition to minor changes to the look and feel of grids, quick filters now appear above the grid.

Fixed Issues

This release provides fixes and enhancements to pre-existing functionality:

20.2.5 Fixes

- Symptom: Under specific conditions some newly created Performance Center items at the end of current sync cycle like tenants, groups, domains might not be synced to the data source. Workaround is to perform a full sync.
  Resolution: With the fix, newly created Performance Center items at the end of current sync cycle would be synced to the data source on the next sync cycle.
  (20.2.5, DE348850, 01352680)
- Symptom: DX NetOps Virtual Network Assurance can create a flood of both email and trap notifications after email service in configuration had been unreachable or down but returned to the normal operation state.
  Resolution: With the fix, DX NetOps Virtual Network Assurance should not create a flood of notifications as send email failures should be processed gracefully.
  (20.2.5, DE439091)
- Symptom: Vulnerability tests against "https://<DAHost>:8582" show vulnerabilities for "LUCKY13" and "Secure Client Renegotiation".
  Resolution: "LUCKY13" and "Secure Client Renegotiation" vulnerabilities on data aggregator host when in https mode are resolved.
  (20.2.5, DE451369)
- Symptom: Description and error messages for a couple of OOB groups still use old names.
  Resolution: Updated description and error messages for these OOB groups to use new names.
  (20.2.5, DE463889)
- Symptom: One28T VNA plugin is slow to gather performance data.
  Resolution: Updated the One28T VNA plugin to use GraphQL interface to query the data source for poll data. It allows us to limit the request to the metrics VNA needs vs all the metrics.
  (20.2.5, DE474270)
- Symptom: Wrong number of active interfaces are seen for a router.
  Resolution: Earlier implementation used 'Router Address' to filter the active interfaces causing some other router addresses to match the filter which led to interfaces of all the matching routers to be listed. 'Router Address' was

used because NFA OData API did not support filtering on 'Router Id' for 'Active Interfaces. NFA OData API has been enhanced to support 'router id' and the corresponding changes to use 'Router Id' for filtering has been made in NetOps Portal.
(20.2.5, DE474276, 32162682)

- Symptom: VNA installs on unsupported Java version.
  Resolution: A warning message which alerts while installing a Java version which is not openjdk or Oracle has been added.
  (20.2.5, DE478257)
- Symptom: Unhanded Exceptions on the `server.log` file.
  Resolution: Handled the Exceptions gracefully.
  (20.2.5, DE478725)
- Symptom: CTRL+clicking a menu item in the navigation menu did not open the page in a new browser window or tab.
  Resolution: CTRL+clicking a menu item in the navigation menu now opens the page in a new browser window or tab.
  (20.2.5, DE479025)
- Symptom: On a Performance Center that has multiple IP addresses, Performance Center selects the first interface and first non-loopback IP for the Performance Center Service default Web Site Host value. This can lead to unreachable Performance Center IP used by Event Manger and users/services that contact the Performance Center service.
  Resolution: Update Performance Center to set the default Web Site Host by trying to determine the fully qualified hostname of Performance Center. If Performance Center cannot determine an FQHN, it falls back to first interface/first non-loopback IP. You can override by way of SsoConfig to set Remote Value for Performance Center Web Site Host.
  (20.2.5, DE479930)
- Symptom: DC out of memory due to excessive in-memory logging of poll errors.
  Resolution: Fixed maximum size calculation for polled errors log. Added options to persist all dcdebug logs to disk.
  (20.2.5, DE480361)
- Symptom: In the upgrade scenario, --no-prompt was not handled that results in posting VNA auth credentials again that leads to validation failure and results in warning message.
  Resolution: Upgrade scenario of VNA with --no-prompt handled and VNA existing credentials checked before posting new ones.
  (20.2.5, DE481089)
- Symptom: Admin group UI does allow to modify locked group by adding or removing items.
  Resolution: With the fix, Admin group UI would not allow to modify locked group by adding or removing items.
  (20.2.5, DE481382)
- Symptom: When new items come in to poll, if there is a bad polled item ID (NULL) in the request, a NullPointerException is thrown and it fails to start polling the additional item ID(s) after the NULL item ID. It does not schedule those polled item IDs.
  Resolution: Updated start polling code to handle if NULL is passed it, and print a WARN message and continue processing the non-NULL item IDs.
  (20.2.5, DE482423)
- Symptom: Rule Editor column selector has empty checkbox fields.
  Resolution: Removed gear for column selection/sorting in the Rule Editor grid. Named last two columns in Rule Editor.
  (20.2.5, DE482432)
- Symptom: Users that did not have permission to administrate groups can run rules on those groups.
  Resolution: Fixed code to grey out the **Run Rules** button. Users that cannot administrate groups cannot run rules on those groups.
  (20.2.5, DE482434)
- Symptom: When the not present items process ran, a recurring exception was appearing in the logs for items that are marked as not present and do not have a metric table because there was never any polled data.
  Resolution: When putting together a set of metric tables to check for metric data on not present items, a query is run to filter out any tables that are not in the database.
  (20.2.5, DE482445)
- Symptom: NFA admin pages are not getting loaded with NFA 10.0.5.

Resolution: Performance Center now looks for the new property (ODataApiUrl) to read OData URL from NFA, and uses it if available. If not, it looks for older property (ODataURL) and uses it. Falling back to older property lets Performance Center continue to work with older versions of NFA.
(20.2.5, DE482726)

- Symptom: Running the `testssl.sh` script against the (Fault Tolerant) data aggregator Proxy server while configured for HTTPS shows vulnerabilities for cipher order, SWEET32 and LUCKY13.
  Resolution: The data aggregator proxy's cipher suites, when configured for HTTPS, have been adjusted so that running `testssl.sh` no longer shows vulnerabilities.
  (20.2.5, DE482977)
- Symptom: On Fault Tolerant system using RHEL6, the daproxy service fails to start and shows "traefik error: error while building entryPoint http" in log.
  Resolution: On RHEL6, the `daproxy-init.sh` script has been updated to correctly reference the `daproxy.toml` file.
  (20.2.5, DE483212)
- Symptom: Custom Horizontal Bar chart when configured for SD-Wan metrics, metric families Tunnel or SLA Path, hyperlink for chart redirects to home page.
  Resolution: Addressed issue with hyperlink that redirected to home page instead of context page. The custom Horizontal Bar chart when configured for SD-Wan metrics, metric families Tunnel or SLA Path, correctly redirects user to context page.
  (20.2.5, DE483606)

20.2.4 Fixes

- Symptom: Python 2 is end-of-life (EOL), and there are a few admin/debug Python 2 scripts shipped with the data aggregator/repository.
  Resolution: Updated the Python scripts to run against Python 3. You must now install Python 3 in Linux to run these scripts.
  (20.2.4, DE440532)
- Symptom: The DX NetOps Virtual Network Assurance Viptela plugin was throwing database-related exceptions in the `Server.log` file.
  Resolution: The DX NetOps Virtual Network Assurance Viptela plugin does not throw database-related exceptions in the `Server.log` file anymore.
  (20.2.4, DE460271)
- Symptom: The procedure for configuring DAProxy and Consul services as HTTPS on a fault-tolerant DX NetOps Performance Management is incomplete and confusing. Configuring DAProxy service from the proxy server to data aggregators yields certificate_unknown errors.
  Resolution: The Traefik service has been upgraded to version 2.2.8 to resolve certificate errors between the DAProxy service and the backing data aggregators. The procedures for configuring the Traefik connection as HTTPS have been rewritten and clarified.
  For more information, see Configure the Traefik Connection as HTTPS from Proxy Server to Data Aggregator.
  (20.2.4, DE460814)
- Symptom: Items sent over from DX NetOps Virtual Network Assurance of type MANAGEMENT_SYSTEM are treated as groups instead of devices.
  Resolution: Update the NetOps mapped type for MANAGEMENT_SYSTEMs sent from DX NetOps Virtual Network Assurance.
  (20.2.4, DE463352)
- Symptom: When a Data Collector upgrade that has been initiated from the UI has failed, the Data Collector remains upgrading.
  Resolution: Update Data Collector status to failed upgrade if a registration message is not received within 10 minutes of initiating an upgrade through the UI process.

(20.2.4, DE470798)

- Symptom: When a poll value delta is calculated to be greater than max positive 32-bit or 64-bit integer, the poll response is dropped only if it happens during a rollover. This happens when an SNMP agent returns a nearly impossible delta that could happen within the poll cycle.
  Resolution: Changed the SNMP delta calculation code to handle these very large deltas even when not in a rollover situation. The max delta is controlled via configuration.
  For more information about how to increase or decrease the large 32-bit or 64-bit delta value used in this check, see Configure Counter Behavior.
  (20.2.4, DE472708)
- Symptom: The DX NetOps Virtual Network Assurance Viptela plugin is not closing the vManage connections properly after use.
  Resolution: With this fix, the DX NetOps Virtual Network Assurance Viptela plugin now closes the vManage connections properly after use.
  (20.2.4, DE472718)
- Symptom: The following behaviors are seen while using the ACI Console dashboard. 1. Application profile topology incorrectly shows endpoint groups that are associated with other application profiles; 2. Application profile topology shows more endpoint groups than the application profile hierarchy; 3. When navigating between an application profile and its endpoint groups in the topology, the Health Score on the application profile item disappears; 4. The topology fails to load initially when navigating to the ACI Console from the context of an ACI leaf switch.
  Resolution: The following changes have been implemented to address the identified issues:
  - Endpoint Groups are now distinguished between normal and L2/L3 External. This change now allows the distinction between L2/L3 External Endpoint groups in the ACI Console topology and has resolved the issue where unrelated Endpoint Groups were being incorrectly pulled into the application profile topology.
  - The Health Scores associated with topology parents are now preserved when navigating in the ACI Console. Using the ACI Console link on the Context page of an ACI Leaf/Spine switch no longer fails to load the topology for the selected item. After upgrading from a release prior to 20.2.3, perform the following on your ACI DX NetOps Virtual Network Assurance Gateways to fully take advantage of the new features.

    NOTE
    These steps cause the inventory of your DX NetOps Virtual Network Assurance Gateways to be re-evaluated in the data aggregator.

    a. Navigate to Administration, Monitored Items Management, DX NetOps Virtual Network Assurance Gateways Page.
    b. For each of your ACI DX NetOps Virtual Network Assurance Gateways, select **Edit**, set the **Administration Status** to **Down**, and then save the changes.

  (20.2.4, DE473532)
- Symptom: userName of DX NetOps Virtual Network Assurance UI accepting spaces.
  Resolution: Added validation to restrict spaces in userName.
  (20.2.4, DE475474)
- Symptom: SNMP discovered Network Interfaces that have been reconciled with DX NetOps Virtual Network Assurance discovered interfaces are deleted when the associated DX NetOps Virtual Network Assurance discovered item is deleted.
  Resolution: The software has been updated to prevent SNMP discovered Network Interfaces from being removed when aj associated DX NetOps Virtual Network Assurance discovered Interface is deleted.
  (20.2.4, DE476082)
- Symptom: "No data to display" after upgrade to 3.7.14.
  Resolution: Added ETL health check to the `dr_validate.sh` file to detect ETL issues before upgrade.
  (20.2.4, DE476111)
- Symptom: Users are not able use the Type Catalog web service to import Metric Family definitions which use the `Pollable.Indexes` attribute instead of `DeviceComponent.IndexList` . It is rejected with a parsing error.
  Resolution: Modified the Type Catalog web service to support the `Pollable.Indexes` attribute.

(20.2.4, DE476319)

- Symptom: DX NetOps Virtual Network Assurance orgVsConnectedDevicesFile during Versa Inventory poll does not get updated.
  Resolution: With this fix, the file is now updated during the inventory poll.
  (20.2.4, DE476346)
- Symptom: Non-application users are able to login to the Performance Center MySql database without providing a password.
  Resolution: Update Performance Center install script to remove anonymous user from the MySql database.
  (20.2.4, DE476790)
- Symptom: OOB groups under all old tenants displayed old group names.
  Resolution: Added upgrade script to update OOB group names of groups that still use old group names.
  (20.2.4, DE477368)
- Symptom: In alarm details panels, links are not shown as operable controls.
  Resolution: In alarm details panels, links are now shown as operable controls.
  (20.2.4, DE477700)
- Symptom: ETL failures might occur if multiple Data Aggregators are or previously have been running simultaneously while using the same Data Repository schema instance. This results in NetOps Portal dashboards showing no data or not showing data for recently discovered items. This might occur by accidentally using the same Data Repository information when installing two data aggregators (that are not intended for Disaster Recovery) or when migrating the Data Aggregator to a new system and both the old and new data aggregator are running and the Data Repository is not blocking access to the old Data Aggregator.
  Resolution: To protect the Data Repository, the data aggregator now verifies that no other data aggregator is attached to the same Data Repository schema instance during the data aggregator startup process. If a data aggregator detects that another data aggregator is actively using the Data Repository, it shuts down immediately and log information to the `shutdown.log` file as to why it has shut down. A long-running data aggregator takes precedence over a newly started data aggregator, and the newly started data aggregator shuts down while the long-running data aggregator remains running.
  (20.2.4, DE477749)
- Symptom: OC is skipping processing of few files on failure scenario.
  Resolution: Fixed issue in OC file processing.
  (20.2.4, DE477816)
- Symptom: After customizing a view, any attempts re-edit the view does not launch view settings dialogue until after the page is refreshed.
  Resolution: Addressed issue that after editing a view, the user is unable make subsequent edited to view configuration until after the page is refreshed.
  (20.2.4, DE477881)
- Symptom: The `dr_validate.sh` script requests linux user password during database connectivity testing.
  Resolution: Modified database connectivity check to use existing passwordless ssh.
  (20.2.4, DE477938)
- Symptom: Packet loss value greater than sum of packets.
  Resolution: Added a condition in the DX NetOps Virtual Network Assurance plugin. If the traffic is less than 100, packet loss displays "0".
  (20.2.4, DE478053)
- Symptom: Some rate records have incorrect tstamp value in the rate tables in the Data Repository (for example, 2 records for same item has the same tstamp). This incorrect tstamp value is sent in error by the Data Collector.
  Resolution: Made Data Collector variable that sets the cycle timestamp on poll responses volatile so it is read correctly on all threads.
  (20.2.4, DE478242)
- Symptom: Ctrl+Click on chart items (for example, entries in the chart legend) that open the clicked item opens the link in the same window (replaces the page). Ctrl+Click should open the link in a new browser window (or tab).

Resolution: Ctrl+Click on chart items (for example, entries in the chart legend) now opens the link in a new browser window (or tab).
(20.2.4, DE478277)

- Symptom: ArrayIndexOutOfBounds exception while processing the tunnel response.
  Resolution: OK Tunnel name: to_CO-GBINTX1-WOAGG03_APACBBL2-APACBBL2BAD Tunnel name: to_JPNI418BRWOBR001_APAC_WAN-APAC_WAN (the extra _'s are causing an issue with our parsing as we get back APAC and try to split it on "-").
  (20.2.4, DE478839)
- Symptom: Output of some Open API queries in CSV format might include extra columns, and device information might be missing in some rows.
  Resolution: With this fix, output of Open API queries do not have extra empty columns, and device information is correctly repeated whenever it is needed.
  (20.2.4, DE478981)
- Symptom: vEdge Interfaces that have non-unique private IP addresses are being incorrectly reconciled within DX NetOps Virtual Network Assurance. This leads to missing interface and tunnel items and also interfaces and tunnels being associated with the wrong devices.
  Resolution: The DX NetOps Virtual Network Assurance Viptela plugin has been updated to no longer use the private IP of an interface as a means to uniquely identify it. This allows the missing interfaces and tunnels to be created, and resolves the issue with items being associated with the wrong devices.
  (20.2.4, DE479272)
- Symptom: DX NetOps Virtual Network Assurance sets the Unique ID to the PK ID, but this is not unique across deployments (with multiple Orchestrators) so the IP address matching is bypassed.
  Resolution: The serial number is appended to the instance ID.
  (20.2.4, DE479296)
- Symptom: Unable to identify a user's permission group through REST to be able to set the user's default group to **My Assigned Groups**.
  Resolution: Add permission group information to user REST endpoint.
  (20.2.4, DE479547)
- Symptom: Cannot add items to groups within the User Defined Groups under the NFA data source inventory in Performance Center.
  Resolution: With this fix, users can add items to groups within the User Defined Groups under the NFA data source inventory in Performance Center.
  (20.2.4, DE479574)
- Symptom: When SSO is running on https/443 and SAML2 is being used, SSO fails to validate the SAML2 Assertion, and fail to log in. Also, if user overrides SSO Virtual Directory, some SSO URLs are made with wrong path, as they were hard coded to use /sso.
  Resolution: Updated the validation code to handle when the assertion is coming in with https/443 or https/no port. Both are acceptable. Also, updated the various places SSO URLs are generated to now use the SSO Virtual Directory setting in SsoConfig.
  (20.2.4, DE479656)
- Symptom: Sometimes there are duplicate entries in the Help drop-down menu.
  Resolution: Validate that the help menu has been created. If it exists, do not re-create the Help menu and drop-down items.
  (20.2.4, DE479752)
- Symptom: The ACI Console fails to load the topology when selecting certain items.
  Resolution: The ACI Console logic has been updated to more gracefully handle incorrectly formatted Open API responses.
  (20.2.4, DE480372)

20.2.3 Fixes

- Symptom: Progress bars and loading indicators in the user interface did not have aria attributes set on them to make them visible to screen readers.
  Resolution: Progress bars and loading indicators in the user interface now have aria attributes set on them to make them visible to screen readers.
  (20.2.3, DE434593)
- Symptom: Disclosures (blocks of text that hide/show) in the user interface did not have aria attributes set on them to make them visible to screen readers.
  Resolution: Disclosures (blocks of text that hide/show) in the user interface now have aria attributes set on them to make them visible to screen readers.
  (20.2.3, DE434594)
- Symptom: "VNA Domains" is not translated for non-EN users.
  Resolution: With this fix, 'VNA Domains' gets correctly translated in French and Japanese.
  (20.2.3, DE463156)
- Symptom: When navigating to the Context Page of a Service Chain Item, the Service Chain Topology graph is left blank.
  Resolution: When navigating to the Context Page of a Service Chain Item, the Service Chain Topology graph has the correct diagram with all the correct images.
  (20.2.3, DE465512)
- Symptom: When the user sorts on the 'Last Flow' column in 'All Interfaces' page, the sorting happens on string format of the date rather than on epoch time.
  Resolution: To avoid sorting on String format of date, an invisible column is created holding the epoch time in long format which is sorted when the user sorts on the last flow column.
  (20.2.3, DE470887)
- Symptom: The DX NetOps Virtual Network Assurance Installer does not maintain an `install.history` file that can be helpful to understand when upgrades occurred, and which versions have been installed.
  Resolution: The installer has been updated to create and maintain a history file to track DX NetOps Virtual Network Assurance installations/upgrades. The `install.history` file exists in the base product directory (for example, /opt/CA/VNA/install.history).
  (20.2.3, DE471049)
- Symptom: The System Status in Performance Center UI does not show the backend DA's - only shows daproxy - on a system configured for fault Tolerance, when the consul service on DAproxy is configured for HTTPS.
  Resolution: Performance Center was not recognizing the consul service is configured for HTTPS when calling consul rest services. Performance Center and data aggregator have been updated to recognize when FT's consul is in HTTPS mode, so DA's can be shown in System Status.
  (20.2.3, DE471182)
- Symptom: On a system configured for fault tolerance, /odataquery points to the active data aggregator instead of DAproxy.
  Resolution: With this fix, on a system configured for fault tolerance, /odataquery now points to the DAproxy.
  (20.2.3, DE471203)
- Symptom: Removing Proxy port field from Viptela Config is causing NPE in the `Server.log` file.
  Resolution: Added additional checks for NPE in the DX NetOps Virtual Network Assurance `server.log` file.
  (20.2.3, DE473370)
- Symptom: Active access points are reported incorrectly.
  Resolution: Active access points filed in broker code are no longer behaving like static.
  (20.2.3, DE473478)
- Symptom: End of DX NetOps Virtual Network Assurance installation, the ReadMe is still pointing to docops.ca.com.
  Resolution: Updated to point to the correct location of the documentation on techdocs.broadcom.com.
  (20.2.3, DE473504)
- Symptom: Observed exceptions in oc logs while query for events in network where the specific deviceType does not exists.

Resolution: Before query for events in network for any productType its existence is checked.
(20.2.3, DE473799)

- Symptom: When we click on an Item in the ACI console topology view that does not have a topology (eg. Vswitch, Contracts, etc), the loading symbol remains indefinitely, even though nothing is loading.
  Resolution: Updated ACI Console to have only a few items be clickable.
  (20.2.3, DE473842)
- Symptom: When configured to use fixed context items, device or interface, with On-Demand Report or Dynamic Trend View if the items profile cannot be determined, as in retired device, configuration warning message is shown when there is no data to display.
  Resolution: No data to display is displayed when there is no metric data to report with with fixed context items with On-Demand Report and Dynamic Trend View. Previously when configured to use fixed context items, device or interface, if the item metric profile cannot be determined view showed configuration warning.
  (20.2.3, DE474023)
- Symptom: In the data aggregator `karaf.log` file, there can be massive warning messages generated from the Event Manager on the CPU and Memory component items due to Syncable facet being mistakenly attached to them.
  Resolution: Improved the component discovery not to create Syncable facet on the CPU and Memory components.
  (20.2.3, DE474142)
- Symptom: An exception is thrown for a Threshold Profile ending with a percent (ie. "95%") when you click on it in the Threshold Profiles tab on the Monitored Devices page.
  Resolution: Fixed issue when a Threshold Profile ending with a percent (ie. "95%") threw an exception.
  (20.2.3, DE474143)
- Symptom: While installing DX NetOps Virtual Network Assurance the end timestamp is displaying as dd_mm_yyyy_hh_mm_ss instead of actual timestamp.
  Resolution: Display the actual end timestamp.
  (20.2.3, DE474261)
- Symptom: CARE files for the Data Aggregator and Data Collector no longer include log files and other important information after installing version 20.2.x.
  Resolution: Update the CARE script to look for the updated product name.
  (20.2.3, DE474321)
- Symptom: An insufficiently sized thread pool leads to a WARN message to be logged for a mis-fired Poll Item or Group ETL job every hour when the DIM Item ETL is running at the same time.
  Resolution: The ETL scheduler's thread pool size was increased and made configurable.
  (20.2.3, DE474462)
- Symptom: "No data to display" after upgrade to 3.7.14.
  Resolution: Added new `etlHealth.sh` script to the upgrade process to detect ETL issues before upgrade.
  (20.2.3, DE474520)
- Symptom: There is no indication of what version of CE is installed.
  Resolution: Update the CE installer to write the version when installing.
  (20.2.3, DE474889)
- Symptom: Dynamic Trend View is not properly suppressed on Device Context Page when configured within the Dashboard Editor.
  Resolution: Addressed issue that the Dynamic Trend View is not properly suppressed on Device Context Page when configured within the Dashboard Editor.
  (20.2.3, DE474985)
- Symptom: The Add Data Source dialog showed incorrect options when adding a CA Business Intelligence Data Source.
  Resolution: The Add Data Source dialog shows the correct options when adding a CA Business Intelligence Data Source.

(20.2.3, DE475210)

- Symptom: Multiple records with the same item_id and facet_id value pairs are present in the item_facet table in the Data Repository. There are internal workflows in the Data Aggregator that can result in the same facet being added to the same item in a single database request.
  Resolution: The item repository data base layer now removes duplicate item_facet records when inserting a single batch.
  (20.2.3, DE475453)
- Symptom: After running the Performance Center disaster recovery script, the new Event Manager IP might not be sent to Data Aggregator due to timing between EM synchronization and event poller threads.
  Resolution: Updated Performance Center disaster recovery script to update the DB field, used for the EM URL sent to the Data Aggregator during re-registration, with the new EM IP address. This allows the event poller thread to see it on startup and send the correct EM URL to Data Aggregator.
  (20.2.3, DE475891)
- Symptom: Odata count query on the metricfamilyhistory entity with a filter might return incorrect count of entities.
  Resolution: With this fix, odata count query on the metricfamilyhistory entity with a filter now returns correct count of entities.
  (20.2.3, DE476454)
- Symptom: The "Détection rapide des unités" ("Quick Discovery") page is blank for French users.
  Resolution: The "Détection rapide des unités" ("Quick Discovery") page displays correctly for French users.
  (20.2.3, DE476675)

20.2.2 Fixes

- Symptom: In a trend view that supports "zoom", there was no way to "zoom" in using the keyboard (it could only be done with the mouse).
  Resolution: In a trend view that supports zoom, you can now zoom into the chart using the keyboard. Pressing space bar while an element in the chart has focus marks the start of the zoom. After using the left/right arrow keys to move focus to another part of the chart, pressing space bar again zooms into the selection.
  (20.2.2, DE406325)
- Symptom: Grids with editable fields in them such as the final pane of the user definition wizard do not support keyboard access or work with screen readers.
  Resolution: Grids with editable fields in them such as the final pane of the user definition wizard now can be used via the keyboard and work with screen readers.
  (20.2.2, DE406328)
- Symptom: The refresh icon is missing from some data aggregator Admin pages grids.
  Resolution: Add the code to display the missing refresh icon in all data aggregator Admin pages that require this grid refresh.
  (20.2.2, DE416639)
- Symptom: The Neighbor Topology in the Alarm Console has some nodes cut off in Firefox. logy tab. The nodes drawn are over to the left and cut off.
  Resolution: Modified the d3 code to arrange the nodes slightly so that they are not cutoff in Firefox.
  (20.2.2, DE431384)
- Symptom: Traps generated for events on Tunnel components do not contain all event properties.
  Resolution: Improved logging and error checking in trap sender.
  (20.2.2, DE431603)
- Symptom: The Create/Edit User and Create/Edit Notification wizards are missing some aria attributes that identify them as wizards to screen readers.
  Resolution: The Create/Edit User and Create/Edit Notification wizards now have additional information in them to make them easier to use with screen readers.
  (20.2.2, DE434592)
- Symptom: If you can capture the cookies from a Performance Center session, then you can use the cookies to impersonate that user even after they have logged out.

Resolution: When the user logs out, the cookies are invalidated and so can't be used to impersonate that user.
(20.2.2, DE438456)

- Symptom: OData API is fetching old metric data when start time and end time equals ZERO.
  Resolution: OData API now returns empty data when start time and end time equals ZERO.
  (20.2.2, DE442822)
- Symptom: Data aggregator failed over to the standby data aggregator with no clear indicator in the logs about why the decision was made to fail the over to the standby.
  Resolution: Connection and read timeouts have been added to the data aggregator ActiveMQ and REST status checks. Additionally, the Consul Extension logging has been updated to be more verbose when a system that is currently the active data aggregator goes down and a fail-over occurs.
  (20.2.2, DE448360, 31918553)
- Symptom: When the browser window width is less than 1024 pixels, the user interface is clipped on the left and none of the controls (including a scroll bar) can be manipulated.
  Resolution: When the browser window width is less than 1024 pixels, the user interface now displays a horizontal scroll bar at the bottom of the window which allows all of the interface to be accessed. Elements within the user interface (such as dialog boxes) also now support horizontal scroll bars if elements are clipped.
  (20.2.2, DE450132)
- Symptom: Old RIB document files can build up in the Performance Center /tmp/ribcache.
  Resolution: Performance Center now cleans up the old version(s) of a RIB document file when a new version is available.
  (20.2.2, DE452506)
- Symptom: When TransientDBConnection logging is set to debug, SQL statements with passwords are logged.
  Resolution: Update logger to sift out SQL with passwords.
  (20.2.2, DE456484)
- Symptom: The REST endpoint requesting the SNMPv3 EngineID every time returns a new SNMPEngineID that is one that is not much used in traps.
  Resolution: With this fix, the REST endpoint requesting the SNMPv3 EngineID returns the correct SNMPEngineID that is one that is much used in traps.
  (20.2.2, DE457958)
- Symptom: The Group Editor's Group Description gets stuck on the invalid description if over 255 chars long.
  Resolution: The Group Editor allows the user to modify a description that is over 255 characters.
  (20.2.2, DE458149)
- Symptom: When discovering Check Point firewall virtual systems with context support, many virtual system items are duplicated with empty context names.
  Resolution: Enhanced the virtual system validation process. The virtual systems that are improperly configured with empty context names are no longer discovered.
  (20.2.2, DE461334)
- Symptom: 8443 HTTPS is enabled by default in DX NetOps Virtual Network Assurance, though HTTPS is not supported.
  Resolution: Disabled HTTPS port on wildfly.
  (20.2.2, DE462717)
- Symptom: On-Demand, view option **Chart per item with multiple metrics** when applying Metric Calculate level by Device the sort order is not being honored.
  Resolution: Addressed issue with On-Demand, view option **Chart per item with multiple metrics** when applying Metric Calculate level by Device the sort order is not being honored.
  (20.2.2, DE464964)
- Symptom: Data Aggregator is unresponsive to RIB, REST, or any other request, but java memory usage/garbage collection is fine. Could be a deadlock situation between a couple tasks the data aggregator is running. There is a deadlock possibility when doing a REST/UI delete, change detection, discovery profile running, and an OData call that filters on groups.
  Resolution: Updated the OData filter code to negate the possibility of the deadlock situation.

(20.2.2, DE464968)

- Symptom: BST Feature update task cannot complete due to incorrect logic in BST Feature Update Task Builder (race condition). 2020-06-03 20:20:03,614 WARN (EE-ManagedThreadFactory-default-Thread-112) [OC_BROADVIEW_PLUGIN] BSTFeatureTaskBuilder$SimpleBSTFeatureUpdateTask 105 Timeout occurs for BST Feature Configuration Update thread, timeout value = 90000.
  Resolution: ManagedScheduledExecutorService was replaced by ScheduledThreadPoolExecutor with a fixed coreThread number. It allows to control thread pool size for scheduled tasks and can help avoid potential thread problems (unexpected growth. etc.) 1. Timeouts for tasks were changed according to plugin config variables; 2. Optimized AlarnClearTaskBuilder logic - removed supervisor thread such as plugin always used only one thread to clear all alarms; added CountDownLatch's logic to control alarm clear thread by a timeout; 3.Implemented new logic for BST Feature Update task (for example, task builder).
  (20.2.2, DE465379)
- Symptom: An OData query might not return device information in expand when it starts from metric family/component, for example:
  ```
  portmfs?$expand=device
  ```
  Resolution: An OData query now returns device information in expand when it starts from metric family/component for example:
  ```
  portmfs?$expand=device
  ```
  (20.2.2, DE465743)
- Symptom: Open API in some circumstances might not honor tenants boundaries.
  Resolution: With this fix, Open API now always honors tenant boundaries.
  (20.2.2, DE466422)
- Symptom: When discovering Cisco switch's Environment sensors, only one of the relevent metric families is discovered; or, all of the environmental sensors metric families, but some have no components.
  Resolution: Improved discovery logic for Cisco Environmental Sensor vendor certs. Added new MVEL method that allows discovery expressions to scan all table entries for an OID, instead of only the first entry.
  (20.2.2, DE466858)
- Symptom: On the System Health page, when you page to the next page in a table the section is collapsed each time.
  Resolution: Instead of setting the collapsed property to be purely based on the status of the section, we're also taking into account the current collapsed state.
  (20.2.2, DE467043)
- Symptom: When proxying as a user who uses a different Locale, the Group Rule Editor shows incorrect translations in its group selector.
  Resolution: Modified group selector to get locale from active user.
  (20.2.2, DE467052)
- Symptom: Scheduling "All Pages of multiple page views" for On-Demand report option "Per metric by single item" the PDF output combines multiple metrics the trend charts.
  Resolution: Fixed issue when scheduling On-Demand report option "Per metric by single item" using "All Pages of multiple page views" report the PDF output combines multiple metrics the trend charts.
  (20.2.2, DE467175)
- Symptom: The responsiveness of the ACI Console dashboard has been improved by updating the ACI Console to request data for multiple items at a time. This should prevent the dashboard from taking excessive time to load when a large number of switches are part of the topology.
  Resolution: The ACI Console can take up to a minute to respond after clicking on one of the icons in the hierarchy. The extended time that it takes to load the topology is ultimately based on number of switches that show up in the topology view on the right.
  (20.2.2, DE467188)
- Symptom: The ACI console displays non-ACI User Domains in the hierarchy.
  Resolution: The ACI console has been updated to display only user domains that contain ACI technology group. By having the user domains in data aggregator store the information about the technology groups, Performance Center can directly fetch user domains that contain ACI.

(20.2.2, DE467199)

- Symptom: When installing/upgrading the Performance Center, Data Aggregator or DAProxy, and there is 1 private IP address and 1 public IP address on the machine, the installer doesn't prompt for which IP address to use for consul to bind and listen on. It instead just uses the private IP address.
  Resolution: Updated the installers to ask if there is 1 private and any number of public IP addresses.
  (20.2.2, DE467204)
- Symptom: The initial state of the sections displayed in the new Administration Group Editor was confusing as some sections were initially uncollapsed and some sections were collapsed.
  Resolution: The initial state of the sections displayed in the new Administration Group Editor is now consistent with all sections except Properties initially collapsed. As before, excepting the Items section, as you move between groups (or pages of the UI), your configuration of what is expanded and collapsed is maintained throughout the user session.
  (20.2.2, DE467211)
- Symptom: Two jre directories are created during installation of Performance Center.
  Resolution: Update the installer sub-module to no longer install the extra JRE.
  (20.2.2, DE467362)
- Symptom: When selecting SD-Wan Tunnel/App Path metrics By Component level with the Scorecard Trend might be missing component items.
  Resolution: When selecting SD-Wan Tunnel/App Path metrics By Component level with the Scorecard Trend always render component items.
  (20.2.2, DE467610)
- Symptom: SDWan Performance and Baseline charts show spikes on trend lines, chart renders with clutter when resolution is under 30 minutes.
  Resolution: With SDWan trend charts are forced to use 30 minute resolution when time range is two more hours to eliminate spikes on trend lines that that clutters rendered charts.
  (20.2.2, DE467612)
- Symptom: DX NetOps Virtual Network Assurance is sending multiple IP addresses as the outofbandmanagment IP address on VM.
  Resolution: DX NetOps Virtual Network Assurance now sends the correct IP as the outofbandManagment IP address. .
  (20.2.2, DE467747)
- Symptom: Zooming did not work in trend charts (when it was allowed).
  Resolution: Zooming now works in trend charts that support it.
  (20.2.2, DE468020, 32162368)
- Symptom: When running Alarm Console against Spectrum 10.4.x+ (20.2.x), it might time out or throw an error when not using a custom group. This is because the landscape group synchronized from DX NetOps Spectrum to Performance Center changed and Alarm Console did not take that into account. It uses global collection groups, and if no global collections are defined, it uses all synced DX NetOps Spectrum items. DX NetOps Spectrum also has a limit on 150 comparisons in the webservice used for Alarm Console, so if too many global collections are passed, it fails.
  Resolution: Updated the logic to check for the old and new DX NetOps Spectrum landscape group IDs. By using landscape group, DX NetOps Spectrum can return all events for the landscape group members quicker than specifying all items in groups. In addition, to resolve the issue with 150 comparisons, up to 150 comparisons are added to the webservice request, and a message is printed to the `PCService.log` file when more than 150 comparisons is hit.
  (20.2.2, DE468346)
- Symptom: During installation of the data repository, the I/O scheduler is always set to the same value regardless of type of storage.
  Resolution: Update the data repository installation process to allow the I/O scheduler to be updated to a different value based on storage type.
  (20.2.2, DE468411)
- Symptom: When device is discovered that are related to F5 metric the context tabs for F5, available on Switch Context page, are not available on the Device Context page. When preference is to not associate the device as a switch device, requires manual creating F5 Tabs on the Device Context page.

Resolution: When F5 related devices are discovered as generic device and prefer to not associate the device as a switch. Provide an automated script to import context tabs for F5, available on Switch Context page, to the Device Context page.
(20.2.2, DE468840)

- Symptom: During Performance Center and DR install, the disaster recovery scripts that might have been customized are overwritten without a backup.
  Resolution: Update the Performance Center and DR installers to make backups of the disaster recovery scripts before setting down updated versions.
  (20.2.2, DE468863)
- Symptom: Group Admin Item section could become non responsive after it shows "no items were added to the group" message.
  Resolution: Group Admin Item section continues to work as expected after it shows "no items were added to the group" message.
  (20.2.2, DE469448)
- Symptom: The Group Editor's rule editor does not allow you to edit or create new rules.
  Resolution: Fixed missing comma in `.jsp` file.
  (20.2.2, DE469478)
- Symptom: Manage Groups admin page is missing Device Component Items section.
  Resolution: With this fix, the Manage Groups admin page properly displays Device Component Items section.
  (20.2.2, DE469491)
- Symptom: The DX NetOps Virtual Network Assurance SilverPeak Plugin - 401 Invalid username or password specified.
  Resolution: The logintype has been fixed - value 2 for TACACS authentication for Silver Peak.
  (20.2.2, DE469607)
- Symptom: Card view on SD-Wan Tunnel Statistics dashboard page does not align with site count being reported with the Geo-Map View.
  Resolution: Addressed issue with the number of Site being reported by the Card view on SD-Wan Tunnel Statistics dashboard page, not aligning with site count in Geo-Map View.
  (20.2.2, DE469805)
- Symptom: The Open API might not return all groups in expand clause for queries starting from configuration entity types (devices,interfaces,etc.) if the total number of groups is more then a hundred.
  Resolution: With this fix, the Open API returns all groups in expand clause for queries starting from configuration entity types (devices,interfaces,etc.) regardless of the number of groups.
  (20.2.2, DE470008)
- Symptom: Multi-View Trend for Interface Utilization, Discards and Errors when configured to use bi-directional metrics, In and Out, might render with 'No Data to Display'.
  Resolution: Fixed the 'No Data to Display' problem with Multi-View Trend for Interface Utilization, Discards and Errors when configured to use bi-directional metrics, In and Out.
  (20.2.2, DE470045, 32099937, 32154068)
- Symptom: With multiple columns trend chart views, the Y-axis title is partially hidden when metric unit is scaled to numeric values that forces axis height to 1000 units.
  Resolution: Addressed the partially visible Y-axis title when numeric values forces axis height on trend chart to 1000 units seen with multiple columns trend chart views.
  (20.2.2, DE470115)
- Symptom: A debug logging message is shown when running the Performance Center installer.
  Resolution: The debugging message has been removed from the installer.
  (20.2.2, DE470139)
- Symptom: Items cannot be manually added to the group(s) in the Admin Group Editor.
  Resolution: With this fix, the administrator can now manually add items to the group(s) using Admin Group Editor.

(20.2.2, DE470634)

- Symptom: Read-only checkbox in the group editor triggers a "form is dirty" state (Save button is enabled) if they were clicked on or given focus.
  Resolution: Read-only checkbox fields in the group editor no longer causes the Save button to be enabled (the form becoming dirty) if clicked on or given focus.
  (20.2.2, DE471045)
- Symptom: In 20.2.1, when logging into odataquery, the login box shows a Host field. The default was changed by mistake.
  Resolution: Updated the sign in page to only show Host field when it should, like in 3.7.x.
  (20.2.2, DE471202)
- Symptom: Site Context Pages rendering business hours in subtitle when business hour filter is disabled.
  Resolution: Fixed issue seen on Site Context Pages with business hours rendered in subtitle when business hour filter is disabled.
  (20.2.2, DE471214)
- Symptom: The data aggregator proxy is unable to write to the `daproxy.log` file.
  Resolution: Update the owning group of the log directory for data aggregator proxy.
  (20.2.2, DE471368)
- Symptom: After upgrading Performance Center to 20.2, event data appears in Chinese until the data aggregator is upgraded to 20.2. Problem occurs after Performance Center upgrade, and ends after data aggregator upgrade.
  Resolution: Modified the Event Manager sync code to ignore translations for languages that are no longer supported.
  (20.2.2, DE471566)
- Symptom: DAProxy installer fails due to missing jre folder.
  Resolution: Update consul server installer to not cleanup extra jre folder on upgrade.
  (20.2.2, DE471583)
- Symptom: A few icons displayed in scorecards were not drawing correctly when using the White theme.
  Resolution: All the icons used in scorecards draw correctly when using the White theme.
  (20.2.2, DE471739)
- Symptom: When device is discovered that are related to F5 metric the context tabs for F5, available on Switch Context page, are not available on the Device Context page. When preference is to not associate the device as a switch device, requires manual creating F5 Tabs on Device Context page.
  Resolution: When F5 related devices are discovered as generic device and prefer to not associate the device as a switch. Provide an automated script to import context tabs for F5, available on Switch Context page, to the Device Context page. Updated to better handle view suppression within the Device Context Page.
  (20.2.2, DE471952)
- Symptom: Custom Site Group are not able to reported as sub-group sites with out of the box SD-WAN Tunnel/App Path Scorecard view.
  Resolution: Enhanced the out of the box SD-WAN Tunnel/App Path Scorecard view to support Custom Site Group when used a sub-group sites.
  (20.2.2, DE471965)
- Symptom: When a role for a user with My Custom Groups was changed via the REST API, the My Custom Groups group was deleted.
  Resolution: Update the role REST API call to pass down the permission group ID for the user. A mismatch for the new permission group was causing the deletion.
  (20.2.2, DE472070)
- Symptom: When selecting SD-Wan Tunnel/App Path metrics with mix of core and baseline metrics at times is missing both metric types.
  Resolution: When selecting SD-Wan Tunnel/App Path metrics with mix of core and baseline metrics always render item to process both metric types.
  (20.2.2, DE472250)
- Symptom: In the Rule Editor, you cannot add multiple values for each condition or sub-rule.
  Resolution: With this fix, the Rule Editor's OR condition icon (plus sign) is always displayed when it is required.

(20.2.2, DE472271)

- Symptom: Sub-Rule's "type" and "condition" columns in the Group Editor's Rule grid are not localized for French and Japanese users.
  Resolution: Use Culture instead of Locale so the Sub-Rule's "type" and "condition" columns in the Group Editor's Rule grid are localized for French and Japanese users.
  (20.2.2, DE472464)
- Symptom: The Vsphere plugin sends all the IPs discovered as outofband Management IP address.
  Resolution: The Vsphere plugin now only sends the management IP only in the outofband management IP address set.
  (20.2.2, DE472565)
- Symptom: When creating notification emails, the subject and body fields do not behave correctly (they take forucs from other fields, do not correctly respond to clicking the Insert button, and do not maintain changes made to their content).
  Resolution: When creating notification emails, the subject and body fields now behave correctly.
  (20.2.2, DE472636)
- Symptom: Cisco ISPLA polling can fail with NullPointerException if component IndexList has unexpected format for a given AttributeGroup.
  Resolution: Cisco IPSLA poller gracefully handles the unexpected component IndexList format, log an appropriate message, and ensure data from the other valid AttributeGroups can be collected.
  (20.2.2, DE472770)
- Symptom: InnoDB fails to initialize during install. A common symptom is that functions that should have installed into MySql did not. Confirmation can be found in the MySql log.
  Resolution: Update the installer to check the fs.aio-max-nr setting and update it to a higher recommended value.
  (20.2.2, DE472860)
- Symptom: When generating CSV reports for trend chart views the CSV Export of data does not reflect Business Hour periods in both raw and scaled format.
  Resolution: Export of data when generating CSV reports for trend chart views data dis not reflect Business Hour periods in both raw and scaled format.
  (20.2.2, DE473237)
- Symptom: ClassCastException in the `expression.log` file.
  Resolution: Improved error handling in mvel snmpProtectedDiv function, and improved expression logging.
  (20.2.2, DE473263)
- Symptom: Some tooltips displayed poorly (the contents line wraps after individual characters or pairs of characters) in some browsers.
  Resolution: Tooltips now display correctly in all browsers.
  (20.2.2, DE473508)
- Symptom: When selecting multiple groups within the On-Demand report to be aggregated at group level not all groups are included in generated report by view type 'Chart per Metric by Single Item'.
  Resolution: Address issue with On-Demand report for view type 'Chart per Metric by Single Item' selecting multiple groups to be aggregated at group level that not all groups are included in generated report.
  (20.2.2, DE473710)

20.2.1 Fixes

- Symptom: Notification traps might show the following error in logs: Operation not permitted
  (sendto failed).
  Resolution: Updated notification trap sender to retry up to two additional times, when receiving this error from the operating system. The error is relogged if retry attempts also fail.
  (20.2.1, DE384374, 01175358, 01261904)
- Symptom: Availability stats were missing from the DX NetOps Virtual Network Assurance Versa plugin in some cases.
  Resolution: Availability stats are now available from the DX NetOps Virtual Network Assurance Versa plugin.

(20.2.1, DE402507)

- Symptom: Items appear in Excluded Items tab for a group after a rule is removed from a group. And can only be added back into the group after removing them from Excluded Items list.
  Resolution: Updated SQL to clear the ByRule flag on items removed from a group due to the rule being removed. Items do not appear in the Excluded Items list, and can be added back via a new rule.
  (20.2.1, DE403297, 01287437)
- Symptom: A number of icons in dialog boxes and message boxes did not have the correct aria attributes set on them. As a result screen readers were unable to read the purpose of the user interface component.
  Resolution: All icons in dialog boxes and message boxes have the correct aria attributes assigned to them so that screen readers can read the purpose of the user interface component.
  (20.2.1, DE406123)
- Symptom: User inteface components used to search and filter did not have the correct aria attributes set on them so that screen readers could provide information about how to use the control and it's components.
  Resolution: All user inteface components used to search and filter now have the correct aria attributes set on them so that screen readers can provide information about how to use the control and it's components.
  (20.2.1, DE406124)
- Symptom: Addressed the missing SD-Wan Application Path metric family when customizing a view with context value that is an DX NetOps Virtual Network Assurance Site.
  Resolution: When building custom view with SD-Wan metric if the context selected is a DX NetOps Virtual Network Assurance Site the Application Path metric family is not available in the metric family selector.
  (20.2.1, DE407396)
- Symptom: During install of the cabi_reports zip file from Performance Center on to the CABI server, if user specified HTTPS scheme, it saves with HTTP scheme.
  Resolution: Fixed the installer to correctly save the scheme for Performance Center as HTTPS when specified during install.
  (20.2.1, DE407715)
- Symptom: Editing a dashboard brings up a blank page and cannot be edited. The `PCService.log` file shows a message like "Caused by: java.util.NoSuchElementException: Property missing: Panel/Pane".
  Resolution: If Pane/Row properties are missing from a view, we now default them to 0 and post a message to user, allowing the dashboard to be recovered.
  (20.2.1, DE408142, 01309964,01328507)
- Symptom: Selecting metrics with different units does not get correctly render with when configuring custom Horizontal Bar Chart views.
  Resolution: Alter and validate unit type when selecting metrics when configuring custom Horizontal Bar Chart views.
  (20.2.1, DE409154)
- Symptom: If the ActiveMQ process on the data aggregator or Data Collector runs out of memory, it is not restarted.
  Resolution: Added '-XX:OnOutOfMemoryError=/opt/IMDataAggregator/scripts/activemq stop' as JVM parameter to data aggregator and Data Collector broker processes.
  (20.2.1, DE409714, 01294980)
- Symptom: IM Card View does not support Threshold values more than 10 digits long.
  Resolution: IM Card View now supports Threshold values up to 12 digits long.
  (20.2.1, DE410339, 01324894)
- Symptom: Sync intermittently fails with high heap usage.
  Resolution: Improved the group path caching mechanism to reduce the memory used by the cache.
  (20.2.1, DE411395, 01331371)
- Symptom: Availability Poll is not closing the session properly and throwing exception while closing the session.
  Resolution: Delivered a code fix to close the availability poll as expected.
  (20.2.1, DE413247)
- Symptom: During install or upgrade, the installation process might fail during a check to see if MySql is running. This is particularly true when using an externalized MySql as we are looking for the running process.

Resolution: The installation process has been improved to check if MySql is running by testing the MySql connection port instead of the mysql process. This allows for the check to validate that MySql is available regardless of whether is it running on the same machine or a different machine.
(20.2.1, DE413399, 01340717,01337090)

- Symptom: Customer is not able to integrate DX NetOps Virtual Network Assurance with one vCenter server.
  Resolution: Provided a code fix after adding a null check for dvsPortStatus.
  (20.2.1, DE413974, 01339886)
- Symptom: Custom PE Interfaces context page suppressed CBQOS Context Tab after upgrading to 3.6 service pack.
  Resolution: Resolved the Custom PE Interfaces context page being suppressed CBQOS Context Tab after upgrading to 3.6 service pack.
  (20.2.1, DE414383, 01317817)
- Symptom: Several combo-boxes used to select time values were missing labels which made them difficult to use with screen readers.
  Resolution: Combo-boxes used to select time values are now labelled with either label tags or aria-label attriibutes so that screen readers can provide context.
  (20.2.1, DE414787)
- Symptom: Edit User wizard shows a password error when saving changes to a user using LDAP authentication.
  Resolution: Disabled password strength checking when the password fields are disabled.
  (20.2.1, DE415377, 01350459,01349270)
- Symptom: At times, clicking a group in the group tree on the Manage Groups page takes a large amount of time and the group is not accessible.
  Resolution: With this fix, clicking a group in the group tree on the Manage Groups page now takes a reasonable amount of time.
  (20.2.1, DE415423, 01345758)
- Symptom: If the `Drilldown` or `DetailedLogging` parameters are not included in a generated URL in DX NetOps Performance Management, (such as when a URL from pre-3.6 is used on 3.6+ DX NetOps Performance Management), the rendered view shows the gear icon and allows editing the view settings.
  Resolution: If the `Drilldown` or `DetailedLogging` parameters are not included in the URL, DX NetOps Performance Management defaults to `Drilldown` and/or `DetailedLogging` , and the view cannot be edited.
  (20.2.1, DE415632, 01351680)
- Symptom: Customers want the option to have the system automatically discover devices contributed through DX NetOps Virtual Network Assurance as SNMP manageable.
  Resolution: There is now a **DiscoverVNADeviceAsSnmpManageable** option under the discoverydefaultconfig data aggregator REST endpoint. Setting this option to `true` causes the data aggregator to create a discovery profile for the IP Domain with any new DX NetOps Virtual Network Assurance IPs.
  (20.2.1, DE415805, 01349698)
- Symptom: Exception on looking up IP address for vm in OC Engine for vsphere plugin.
  Resolution: Handled the scenario where IP address was not available for Virtual machine in vsphere plugin.
  (20.2.1, DE416260, 01354780)
- Symptom: Operation status of the DX NetOps Virtual Network Assurance gateway is shown as "NoSuchMessage" on Performance Center when the Data Collector was not reachable.
  Resolution: Performance Center shows correct operation status for the DX NetOps Virtual Network Assurance gateway when the Data Collector is not reachable.
  (20.2.1, DE416277)
- Symptom: Virtual Machines ,Clusters and Hosts were not discovered which were in Folders.
  Resolution: Fixed the issue , added logic to discover Inventory in folders.
  (20.2.1, DE416658, 01353535)
- Symptom: When loading the ACI Console at larger scales, it takes the tree a significant time to load the health scores.
  Resolution: Updated the ACI Console to load the health scores in a more efficient manner, so the tree renders significantly faster.

(20.2.1, DE417378, 01356232)

- Symptom: SslConfig could not import key or certificate from a PKCS12 file.
  Resolution: With this fix, SslConfig does allow to import key and certificate from a PKCS12 file.
  (20.2.1, DE417649, 01358678)
- Symptom: Access Point is shown as Other device instead of manageable device.
  Resolution: The Access Point is now shown as manageable device from other devices.
  (20.2.1, DE417857)
- Symptom: Exceptions in the performance and inventory capture causing the inventory and performance not to not populate for the customer.
  Resolution: Handled the exceptions in the code, Now the data is populated.
  (20.2.1, DE417918, 01339886)
- Symptom: When discovering the Performance Center device via SNMP in the data aggregator, you cannot view SNMP metrics on the Performance Center item.
  Resolution: Updated the Performance Center item to have same default SNMP metric context tabs and ability to add a custom tab of SNMP metrics, when discovered from Data Aggregator.
  (20.2.1, DE417930, 01356609)
- Symptom: On any request OData service responds with "Invalid entity collection or function name 'odata'.".
  Resolution: With this fix, if the issue is seen again, the OData server now logs full trace back of the exception.
  (20.2.1, DE418095)
- Symptom: When the Data Collector runs as a non-root user, the user cannot re-assign the Data Collector to a different tenant and/or IP domain due to permissions to restart the Data Collector from inside the Data Collector.
  Resolution: Updated the restart logic to call sudo to restart the Data Collector if it's running as non-root user. Be sure to check the documentation for an additional sudo command to allow the Data Collector to run as the install user.
  (20.2.1, DE418500, 01363625)
- Symptom: A new vendor set "AWS_IPSEC_TUNNEL" created for aws tunnel metrics and mapped to SDN Tunnel MF this is turned out to be a default VC for SDN Tunnel MF, which should not be the case.
  Resolution: Added <Vendor>Amazon</Vendor> in AWS_IPSEC_TUNNEL Vendor Cert and updated the version of the SDN Tunnel MF which was missing in earlier delivery.
  (20.2.1, DE418710)
- Symptom: RIB queries that target a Data Aggregator and do not include any GROUP BY, ORDER BY, or LIMIT clause started to fail due to parsing errors on the Data Aggregator after a recent Data Aggregator upgrade.
  Resolution: Fixed the Data Aggregator's RIB query parsing logic for this scenario.
  (20.2.1, DE418891, 01367393)
- Symptom: When the system Performance Center is set to have localization other than en_US.UTF-8, the caperfcenter_console service fails to start.
  Resolution: In the `caperfcenter_console` script, locally set the localization to en_US.UTF-8. This allows the expected error checking to proceed and let the process start up.
  (20.2.1, DE418968, 01366004)
- Symptom: When rendering baseline and core metrics on same trend chart when resolution is less than an hour trend line for baseline metric appears as dots that are not connected due to gap data points.
  Resolution: Addressed issue with baseline metric appearing as dots, not connected when rendering baseline and core metrics on same trend chart when resolution being applied is less than an hour.
  (20.2.1, DE419012)
- Symptom: When I select a global collection or enter a string filter value with an ampersand, for example, a global collection named "Demo & Development Systems", I get an error on the alarm console.
  Resolution: Modified the code that generates the built in filters, and user defined filters for the alarm console to properly escape values when generating XML requests for alarm data.
  (20.2.1, DE419546, 01368197)
- Symptom: When doing an OData query that had multiple filter groups, it was incorrectly combining the checks to return the wrong items.
  Resolution: Updated the OData filter group handling to correctly group the checks for items to find.

(20.2.1, DE419659, 01366295)

- Symptom: DVS elements were not collected when the they are under multiple network folder structure levels.
  Resolution: Fixed the code to support multiple folder structure for DVS.
  (20.2.1, DE419699, 01354247)
- Symptom: Wildfly reload fails on initial Installation.
  Resolution: Using Synchronized Version of reload method for Wildfly.
  (20.2.1, DE419709)
- Symptom: When running Remote Engineer on Performance Center, it complains that it cannot find em.properties.
  Resolution: Updated the `re.sh` script to use the correct path to `em.properties` .
  (20.2.1, DE419850)
- Symptom: The `commons-fileupload-1.2.2.jar` file is present in the `PCInstallDirectory/PerformanceCenter/PC/webapps/pc/WEB-INF/lib/` directory. It has been highlighted as a security vulnerability by some sites.
  Resolution: Upgrade the `commons-fileupload` file to version 1.4.
  (20.2.1, DE419870, 01372590)
- Symptom: NFA "Top ..." Pie Chart with Table views intermittently take a long time to render. Sometimes they are quick, other times they take minutes.
  Resolution: NFA "Top ..." Pie Chart with Table views now render in one pass rather than repeatedly updating the Table and Pie sections of the view.
  (20.2.1, DE419970, 01342689,20020017)
- Symptom: When upgrading to 3.7, it might fail to run the mysql_upgrade program because the password for the account used was not being passed. Check `/opt/CA/MySql/mysql_upgrade_results.txt` to see if it complained about bad password or other error.
  Resolution: Updated Performance Center installer to pass the password to the mysql_upgrade program during upgrade.
  (20.2.1, DE420025)
- Symptom: On a scale system Global Search could become too slow since the search criteria is applying a LIKE to every column for each view.
  Resolution: With this fix, the search criteria is applying to a given subset of columns for each view.
  (20.2.1, DE420103)
- Symptom: On Migration from Wildfly 10 to Wildfly 13 & 17, the `JBOSS_CONSOLE_LOG` property changed in the wildfly initialization Script.
  Resolution: Added the `JBOSS_CONSOLE_LOG` property to `/dev/null` in Wildfly 13 and 17.
  (20.2.1, DE420361)
- Symptom: DX NetOps Virtual Network Assurance gateway runs in TLSv1.0 or TLSv1.1.
  Resolution: Updated DX NetOps Virtual Network Assurance gateway to run using TLSv1.2 only.
  (20.2.1, DE420450)
- Symptom: When discovering existing devices and an abort discovery happens before it can complete device discovery and could set the flag that the device was existing, it deleted the existing device.
  Resolution: Updated the discovery abort logic to not delete any existing devices.
  (20.2.1, DE420542)
- Symptom: The contrast of link text
  (a shade of blue) on a selected row in a grid
  (which has a background color of a pale gray or blue), was insufficient. On certain monitors, the background color wasn't visible
  (appeared to be white). The shade of blue used for links had a contrast against the background color that was insufficient to meet Accessibility text on background contrast ratio requirements.
  Resolution: The background color for selected rows in grids and lists has been darkened and the color used for link text has been adjusted so that it meets accessibility contrast requirements when the link is drawn on the new background color.

(20.2.1, DE420630)

- Symptom: For Viptela environments with a large number of devices/tunnels, the performance polling results in high memory usage on the DX NetOps Virtual Network Assurance. This causes DX NetOps Virtual Network Assurance to run out of memory and/or slow down.
  Resolution: The DX NetOps Virtual Network Assurance Viptela plugin polling logic has been updated to more efficiently process lists of polled data.
  (20.2.1, DE420663)
- Symptom: Threshold values in Group Scorecard Trend are not reversed (so that low values are Red/Critical) when only Critical and Major values are provided, and Minor is set to 0 (zero - disabled) in the view settings.
  Resolution: Threshold handling has been updated so that forward or reverse direction is correctly detected when any two threshold values are set, and the third is set to zero / disabled in the view settings. If only one threshold value is provided (others are zero), the direction is set to forward (High values are Red/Critical).
  (20.2.1, DE420784, 20003177)
- Symptom: Geo location attributes cleared in Performance Center group editor for the Site groups are not cleared on the data aggregator, so the OpenAPI shows the last numerical value before they were changed to blank.
  Resolution: With this fix, the geo location attributes cleared in Performance Center group editor for the Site groups is now cleared on the data aggregator and OpenAPI-provided values now match Performance Center values.
  (20.2.1, DE420818)
- Symptom: DX NetOps Virtual Network Assurance reports Virtual Machine and Host Metrics to wrong metric family in data aggregator.
  Resolution: Fixed and Mapped Virtual Machine and Host metrics to the correct Metric family in the data aggregator.
  (20.2.1, DE421149)
- Symptom: The Data Collector can stop receiving inventory and performance data from DX NetOps Virtual Network Assurance under the following scenario: (1) In Performance Center, go to DX NetOps Virtual Network Assurance Gateways admin page, set the Administrative State to **Down** and then to **Up**. (2) Later on, Data Collector loses connection to DX NetOps Virtual Network Assurance because of network issue or if DX NetOps Virtual Network Assurance itself goes down.
  Resolution: Fixed logic in the Data Collector to ensure that is always restores the connection to DX NetOps Virtual Network Assurance when it becomes available.
  (20.2.1, DE421399)
- Symptom: When group memberships in Performance Center are removed during global synchronization (and moved to deleted table), due to item being deleted, it does not remove the BY_RULE flag from the relationship. This causes the relationship to appear in Excluded Items.
  Resolution: When removing a group relationship and the relationship is moved to the deleted table, the BY_RULE flag is removed. Now, the relationship is not considered for Excluded Items list, and if the item comes back, the group rules can re-add the relationship.
  (20.2.1, DE422002, 01287437)
- Symptom: When using the DX NetOps Virtual Network Assurance VMware plugin and the data aggregator is on a virtual machine managed by the plugin, removing the DX NetOps Virtual Network Assurance gateway could cause the data aggregator item to be deleted and re-created on startup.
  Resolution: Added check to the DX NetOps Virtual Network Assurance gateway removal code to not remove any data aggregator or Data Collector device items, and just remove the DX NetOps Virtual Network Assurance facets off them instead.
  (20.2.1, DE422039)
- Symptom: During Global Synchronization, if a deadlock or cannot acquire lock issue arises, temporary tables are dropped and the connection is remade. This results when the command is re-run, it could not find the temporary table.
  Resolution: Updated the retry query handler to not remake the connection on deadlock or cannot acquire lock errors. So when command is re-tried, the temporary tables are still there.
  (20.2.1, DE422072)
- Symptom: When a user with only access to dashboards and **Drill into View** role right, the menu bar disappears when drilling into an item due to NullPointerException in _topbar.jsp.

Resolution: Resolved the issue causing NullPointerException, when drilling into a context page from a dashboard without **View Inventory and Search** role right.
(20.2.1, DE422134, 20022262)

- Symptom: A DX NetOps Virtual Network Assurance device could be mis-reconciled to an SNMP device if the DX NetOps Virtual Network Assurance device's primary is found in the SNMP device's secondary IP list.
  Resolution: Modified the DX NetOps Virtual Network Assurance device reconciliation algorithm to have the Unique ID have a higher precedence than the IP list match.
  (20.2.1, DE422172)
- Symptom: OpenAPI does not show all paths to a group, as a result, you cannot to create a filter to get groups that belong to a path show in Performance Center.
  Resolution: With this fix, OpenAPI now shows all paths to a group separated by ";".
  (20.2.1, DE422272, 20025914,20025914)
- Symptom: After a long period of operation, MySQL process eventually consumes a large percentage of overall system memory even though the process memory settings are properly sized. This is due to MySQL's performance_schema, which is an in-memory schema and not used for Performance Center, being enabled by default.
  Resolution: The MySQL performance_schema has been disabled in the `my.cnf` file installed with Performance Center.
  (20.2.1, DE422458)
- Symptom: When Data Aggregator root logger is changed to ERROR, an erroneous ERROR message was printed during schema validation when there were no differences found.
  Resolution: Updated to check that there are differences before printing the ERROR message.
  (20.2.1, DE422638)
- Symptom: Export the Global Search results to CSV produce a file that is virtually empty.
  Resolution: With this fix, export of the Global Search results now produces a CSV file containing the search results.
  (20.2.1, DE422647)
- Symptom: After an Event Manager full synchronization is requested, EM spawns its inventory synchronization. During inventory sync, the group path cache is refreshed. If another EM synchronization is kicked off during this time, it could throw an error that group path cache is being rebuilt.
  Resolution: Removed the unneeded check during EM PULL to determine if group path cache was currently being rebuilt. EM synchronization should succeed.
  (20.2.1, DE422795)
- Symptom: Inventory updates result in processing exceptions and ultimately fail to be processed. This results in Persistence exceptions filling up the DX NetOps Virtual Network Assurance log files and might result in missing inventory in DX NetOps Performance Management and DX NetOps Spectrum.
  Resolution: The DX NetOps Virtual Network Assurance datamodel has now been updated to cleanup orphan Subnetted IP Addresses when persisting NetworkInterfaces. The upgrade process now also identifies and cleans up any duplicated Subnetted IP Address objects in the database.
  (20.2.1, DE423138)
- Symptom: Inventory might not update. During upgrade, an exception is thrown in log, containing the following: "Caused by: org.hibernate.NonUniqueResultException: query did not return a unique result: 3".
  Resolution: Updated inventory processing logic to delete multiple instances of cache key entries with the same keystring.
  (20.2.1, DE423273)
- Symptom: The Jetty server might expose its details on error pages.
  Resolution: With this fix, the Jetty server does not expose details on error pages.
  (20.2.1, DE423311)
- Symptom: In a multiple data sources environment, any device changes from other data source could trigger the data aggregator to run the inventory discovery during the Performance Center and data aggregator sync process.
  Resolution: The sync process has been improved so that only if a new device IP is pushed down from Performance Center can the inventory discovery be run in the data aggregator.

(20.2.1, DE423494)

- Symptom: Devices with multiple poll rates for same metric family lose polling after data aggregator restart.
  Resolution: Fixed a flaw in the polling configuration recalculation logic that runs during data aggregator startup that causes a loss of polling.
  (20.2.1, DE423637, 20015980)
- Symptom: When re-connecting to a DX NetOps Virtual Network Assurance Gateway, the data aggregator deletes relationships that should still exist. This ultimately leads to group memberships disappearing in the DX NetOps Virtual Network Assurance Domains hierarchy in Performance Center.
  Resolution: Updated the data aggregator logic around determining which relations should be removed when re-connecting to a DX NetOps Virtual Network Assurance Gateway. The data aggregator no longer deletes relationships that should still exist.
  (20.2.1, DE423878)
- Symptom: Scheduled Reports List and Edit dialog intermittently show incorrect time for the schedule run, and are off by the number of hours between the user's time zone and the system time zone. If saved while in this state, the scheduled report now runs at the incorrectly offset time. If not saved, the scheduled email still runs at the originally defined time.
  Resolution: Certain dashboard views were modifying the thread's TimeZone to the user's TimeZone and not restoring it, which injected an incorrect time zone offset in the Scheduled Reports UI. This has been corrected.
  (20.2.1, DE423920)
- Symptom: Tenant themes might switch back to CA-Blue after changing them to another theme. This is because the data aggregator tenant entry in Performance Center data source tenant table still had CA-Blue.
  Resolution: Updated Data Aggregator pull sync code to send up tenant and ip domain changes sent during previous push sync, so data aggregator data source table entries are updated with new theme.
  (20.2.1, DE424092)
- Symptom: Performance Center takes a very long time possibly to consolidate SDWAN tunnels and application paths at large scale.
  Resolution: Rewrote SQL queries to consolidate SDWAN tunnels and application paths in an acceptable time frame.
  (20.2.1, DE424244)
- Symptom: User's default group is not accessible via REST.
  Resolution: Added defaultGroupId tag to user REST service. Added new REST endpoint to set default group: http://{PC}:8181/pc/center/webservice/users/userId/{User ID}/defaultGroup/{group ID}.
  (20.2.1, DE424967, 20028161,20030925)
- Symptom: When device's Reachability is not backed by the ICMP vendor cert and the device is down, the device's status in the context page are shown as Unknown, instead of Down.
  Resolution: Improved the polling process to better handle the SNMP rechability so that the device status can be shown correctly.
  (20.2.1, DE425145)
- Symptom: When doing a Data Aggregator synchronization, it might take a long time during PUSH group stage, when there are many site groups using business hours at very large pollable item scale. Checking kill -3 on the data aggregator process during PUSH shows getBusinessHoursID in one of the stacks and repeating after 5 mins and seeing the same stack on the same thread confirms.
  Resolution: Sped up the getBusinessHoursID search greatly by just looking at items with business hour tag on it, and not also checking all items with polllable tag.
  (20.2.1, DE425157)
- Symptom: When using the Performance Center REST service to clone a user who uses external (LDAP) authentication, the new user cannot log in.
  Resolution: For users with external authentication, modified the clone operation to set the correct authentication flags. For users with internal authentication, modified the clone operation to expire the password immediately; new users must change their password on login.
  (20.2.1, DE425283)
- Symptom: The parseSyncTimes.pl script provided with the Performance Center product does not include sync stage timings for data sources that use the HTTPS protocol.

Resolution: The parseSyncTimes.pl script has been modified to report sync stage timings for data sources that use the HTTPS protocol.
(20.2.1, DE425408)

- Symptom: Inventory discovery hangs when the Data Collector does not return a response to the Data Aggregator. This is due to the Data Collector throwing an ArrayIndexOutOfBoundsException when it receives invalid SNMP GetNext response, causing it to stop processing instead of returning a response.
  Resolution: Fixed the logic to ensure an SNMP Get next response contains the appropriate number of var binds based on the request. If this not the case, the Data Collector sends a failure response to the data aggregator.
  (20.2.1, DE425425)
- Symptom: Parsing the SSO token could result in retrieving wrong field for version.
  Resolution: Updated the SSO token parsing code to correctly retrieve version from the token.
  (20.2.1, DE425577)
- Symptom: If a sync push or pull phase for a data source fails due to an exception, there is no END PULL or END PUSH message logged for the sync cycle for that data source. This can make it difficult to understand when each sync phase completes when reviewing the logs.
  Resolution: The END PULL and END PUSH messages are logged even if there is a failure during the cycle. The error that caused the failure is logged shortly after then END PULL or END PUSH log.
  (20.2.1, DE425597)
- Symptom: DX NetOps Virtual Network Assurance and CAMM metrics for devices and their components are dropped after using the Stop and then Start Polling actions. .
  Resolution: The SDN and EMS Polling Configurations are now correctly recreated when the Start Polling action is sent to a device.
  (20.2.1, DE426041)
- Symptom: A small number of UI elements that need labels in order for screen readers to successfully explain the purpose of the element were missing associated label tags or were missing aria-label attributes.
  Resolution: All UI elements that need labels in order for screen readers to successfully explain the purpose of the element now have either explicit label tags associated with them or have aria-label attributes associated with them.
  (20.2.1, DE426089)
- Symptom: Disabled buttons are present in the keyboard tab order even though they cannot be operated and the focus indicator is barely visible.
  Resolution: Disabled buttons are no longer in the keyboard tab order.
  (20.2.1, DE426090)
- Symptom: Labels in some types of charts are drawn in a color that does not have sufficient contrast against the background color.
  Resolution: Labels in some types of charts are now drawn in a color that has sufficient contrast against the background color.
  (20.2.1, DE426091)
- Symptom: Open API does not allow to filter on null values, so it is not possible to create a filter expression find all for example device that have Description property value as null .
  Resolution: With this fix, OpenAPI now allows filtering on null value for String property types. Filtering on other property type values, such as Double and Integer, is not possible due to OData2 framework limitation. The issue is fixed in Odata4 framework.
  (20.2.1, DE426201)
- Symptom: The Data Collector does not send a poll request if it is determined to be "late". This can happen if the Data Collector system time moves ahead by at least 90 seconds (if Data Collector is polling items at 1 minute) or by at least 450 seconds (if Data Collector is polling items at 5 minutes).
  Resolution: The Data Aggregator generates events indicating that a Data Collector is having this issue, and also generate events on devices for which poll requests are dropped.
  (20.2.1, DE426204)
- Symptom: The Edit Metric Expression dialog did not display its contents correctly.
  Resolution: The Edit Metric Expression dialog now displays correctly.

(20.2.1, DE426590)

- Symptom: Time-selection combo-boxes did not work correctly with screen readers.
  Resolution: Time-selection combo-boxes work correctly with screen readers.
  (20.2.1, DE426904)
- Symptom: Fields associated with picking dates or times do not all have either label tags associated with them or aria-label attributes associated with them so that screen readers can identify the purpose of the field.
  Resolution: Fields associated with picking dates or times now have either label tags associated with them or aria-label attributes associated with them so that screen readers can identify the purpose of the field.
  (20.2.1, DE426917)
- Symptom: Description Field for Notifications in MySql is only 255 Char , So when the String to be added is more than 255 char it is throwing an persistence Exception.
  Resolution: Increased the Description Field for Notifications to 1024 Char.
  (20.2.1, DE427528, 20030542)
- Symptom: Inventory not detected by DX NetOps Spectrum in case of Aggregator scenario.
  Resolution: We Improved the logging to stop such issues in the future.
  (20.2.1, DE427652)
- Symptom: After a DX NetOps Virtual Network Assurance gateway has been removed from the Data Aggregator, temporary DX NetOps Virtual Network Assurance items and relationships are not cleaned up.
  Resolution: Update the delete DX NetOps Virtual Network Assurance Gateway logic to also clean up temporary SDN items.
  (20.2.1, DE427863)
- Symptom: When the Performance Center MySQL instance is installed on a host that is remote to the Performance Center services, firewall rules might close idle DB connections which leads the SSO process to exhaust its connection pool, and eventually hang while processing login requests. This problem should not be present if MySQL is co-located with the Performance Center services.
  Resolution: Fixed a defect related to persistent DB connections that "leaked" connections if the connection was closed by an external agent, such as firewalls do with idle connections. Reconfigured DB connection pools used by all Performance Center services to periodically validate idle connections to keep them from being closed by firewalls.
  (20.2.1, DE427880)
- Symptom: DX NetOps Virtual Network Assurance Notifications are dropped if the value of the string is more than the mysql database char size.
  Resolution: The Description field and the cause field are auto -truncated if the size of the chars is more than 1024 and 255 chars respectively.
  (20.2.1, DE428342)
- Symptom: The data sources with host name length greater than 50 fail in data source registration.
  Resolution: The max length of data source host name has been increased to 255.
  (20.2.1, DE428397)
- Symptom: Performance Center's console service is unable to start on a system that doesn't have wget installed.
  Resolution: In the Performance Center console start-up script, check if wget is installed and report an error to the user if it isn't.
  (20.2.1, DE428440)
- Symptom: The ipadress from Vcenter is sent with invalid spaces at the end of the ip address , which causing error on adding the device to the database.
  Resolution: The Ipaddress from Vcenter is trimmed before adding to the database.
  (20.2.1, DE428464)
- Symptom: Component name aliases in data aggregator were being synced to Performance Center, and if longer than 255 characters, throws SQL error about data truncation. This causes data aggregator PULL phase of synchronization to fail.
  Resolution: Updated data aggregator synchronization code to no longer send component name aliases to Performance Center. The data aggregator should never have been synced to Performance Center. Performance Center should only be pushing the name alias to data aggregator.

(20.2.1, DE428603)

- Symptom: Some separators (horizontal rules) in pages and in drop-down lists were in the tab order of the page (and had nothing associated with the that a screen reader could use). This issue manifested itself only in Firefox.
  Resolution: Separators in pages and drop-down lists are no longer tab stops at any time or in any browser.
  (20.2.1, DE428706)
- Symptom: Our enhanced keyboard navigation didn't have protection from overloading the tab contents loading queue. So, when "zipping" through the tabs with the keyboard, a user might see a dialog pop up with a `500 Server Error.` Hitting refresh in the browser cleared this error, but it was annoying.
  Resolution: We added protection so that navigating with the keyboard was protected, like navigating with the mouse.
  (20.2.1, DE429272)
- Symptom: Trend/Table views with Business Hours applied shows no shading in Trend for time ranges longer than 14 days.
  Resolution: The Trend in Trend/Table view has been corrected to request hourly data instead of daily for longer time ranges when Business Hours filtering is applied. This allows the shading to be rendered correctly.
  (20.2.1, DE429361)
- Symptom: After proxying to a user and loading user's default dashboard, if the user then brings up the group selection dialog, it hangs when trying to close the dialog.
  Resolution: Proxying takes you to the user's default dashboard. The url contains a "pg" parameter with no value. That was causing a problem when bringing up the group context selector dialog. We added protective code to handle this case properly.
  (20.2.1, DE429375, 20044304)
- Symptom: Global synchronization for devices could throw error about "Data too long for column 'IPs'" if there were too many devices with same common IP.
  Resolution: Removed the function that contained the offending SQL, as it was no longer needed.
  (20.2.1, DE429382)
- Symptom: When a Data Collector goes down and back up during a discovery, the discovery could get into a state of trying to be aborted by the hang detector and not fully get aborted and cleaned up. Resulting in the number of running discoveries to be clogged up by discoveries not running.
  Resolution: Added additional info/debug to discovery code around aborting discovery to help track these what is happening.
  (20.2.1, DE430011)
- Symptom: Items are missing from DX NetOps Virtual Network Assurance due to persistence failures while processing messages. .
  Resolution: Updated the DX NetOps Virtual Network Assurance persistence error handling to prevent failures from rolling back message processing. Additionally, updated the persistence logic to correctly persist objects that were previously leading to persistence failures.
  (20.2.1, DE430044)
- Symptom: Upgrading Performance Center after previously upgrading to a recent monthly update appears to be successful, but the `dbmigrate` log file in the `InstallLogs` directory shows errors similar to the following:

```
        Sep 04, 2019 11:20:11 AM com.ca.im.installanywhere.util.DbMigrateApp
  main SEVERE: java.lang.IllegalArgumentException: Expected scheme name at index
  0: :mysql\\://localhost\\:3306/netqosportal?characterEncoding\\=UTF-8&useSSL\
 \=true&verifyServerCertificate\\=false at java.net.URI.create(URI.java:852) at
  com.ca.im.installanywhere.util.DbMigrateApp.runMigrateMode(DbMigrateApp.java:286)
  at com.ca.im.installanywhere.util.DbMigrateApp.run(DbMigrateApp.java:164) at
  com.ca.im.installanywhere.util.DbMigrateApp.main(DbMigrateApp.java:54)
```

  This problem requires two Performance Center upgrades to be exhibited.
  Resolution: The Performance Center installer adds escape characters to the database URLs in the various properties files, which were not expected by the DB migration utilities, and were causing errors in parsing such URLs. The DB migration utilities have been updated to properly handle URLs if they have escape characters.

(20.2.1, DE430601)

- Symptom: The overall performance of Performance Center becomes very slow in a high scale environment when telemetry data is being collected.
  Resolution: Update the telemetry database queries to optimize performance.
  (20.2.1, DE430708)
- Symptom: Inventory discovery could hang and then abort when discovering a device that has class D or E IP addresses.
  Resolution: Enhanced the device discovery to properly handle IP class D and E IP addresses.
  (20.2.1, DE430883)
- Symptom: DX NetOps Virtual Network Assurance device items were not showing up within the DX NetOps Virtual Network Assurance group hierarchy.
  Resolution: Updated the Data Collector to correctly handle situations where DX NetOps Virtual Network Assurance sends more than one type of relation for an entity.
  (20.2.1, DE430894)
- Symptom: When threshold events are forwarded to DX NetOps Spectrum, the quotes around the item are not displayed correctly.
  Resolution: Updated threshold event message to use ASCII single quote and double-quotes instead which are displayed correctly.
  (20.2.1, DE431054)
- Symptom: Certain special characters within a custom database password were causing database setup to fail for Performance Center.
  Resolution: Updated the installer logic to better handle when special characters are included in the database password.
  (20.2.1, DE431277)
- Symptom: Reconciliation of entities between Performance Center and DX NetOps Spectrum is not happening in aggregator scenario of DX NetOps Virtual Network Assurance, due to mismatch in Entity IDs.
  Resolution: Introduced Sub Entity ID which is sent in aggregator scenario which DX NetOps Spectrum uses for reconciliation in case of aggregator scenario.
  (20.2.1, DE431335)
- Symptom: When trying to override the device type for a Meraki device, discovered via SNMP, it shows Other.
  Resolution: Updated the CiscoMerakiManagementMib vendor certification to pass the SysObjectID read from the device to the device type MVEL function. Now it can use the custom `DeviceTypes.xml` to override the type.
  (20.2.1, DE431612)
- Symptom: Inventory view for "Virtual Interfaces" was incorrectly labeled as an Event List view. The incorrectly label view confused user when configuring dashboard page.
  Resolution: Correct the name used for a Virtual Interfaces inventory view that was labeled as an Event List view. .
  (20.2.1, DE431708)
- Symptom: Fips is unable to be configured with an encrypted password in the `sso.properties` configuration file.
  Resolution: Update Fips configuration to be able to handle encrypted passwords.
  (20.2.1, DE431709)
- Symptom: Emails could not be sent securely if email server supported STARTTLS.
  Resolution: Updated Performance Center email properties to enable STARTTLS on all emails being sent. Removed Use SSL button, as Performance Center tries and sends all emails securely if email server supports STARTTLS.

  NOTE
  This now requires the email server certificate and any intermediate/root certificates to be imported into the `/opt/CA/jre/lib/security/cacerts` directory, so the email server certificate is trusted on connection. For more information, see Set the Email Server.

  (20.2.1, DE432418)
- Symptom: Configuring custom Trend Chart to generate comma-separated, CSV, report non-baseline metrics were missing when combined with baseline metrics as primary sort column.

Resolution: Addressed issue when generating a comma-separated, CSV, report was missing non-baseline metrics when combined with baseline metric as primary sort column on custom Trend Chart views. .
(20.2.1, DE432700)

- Symptom: The adminStatus and operationalStatus of the OpenDayLight controller in DX NetOps Virtual Network Assurance were not captured correctly.
  Resolution: The adminStatus and operationalStatus of the OpenDayLight controller in DX NetOps Virtual Network Assurance are now captured correctly.
  (20.2.1, DE432780)
- Symptom: The time displayed in the time picker
  (upper right corner of dashboard) shows incorrect/next year for the last week of the year. December 30, 2019 Displays as December 30, 2020.
  Resolution: Modified the date formatter to use yyyy instead of YYYY which formats the last week of the year correctly. The time displayed in the time picker now shows the correct year for the last week of the year, for example, December 30, 2019.
  (20.2.1, DE433614, 20197781)
- Symptom: Vendor cert "Cisco IPSLA Ethernet DMM and IP Statistics" and "Cisco IPSLA Ethernet SLM (Synthetic Loss Measurement) Statistics" were coupled with vendor cert "Cisco IPSLA Ethernet Jitter Precision Statistics". If "Cisco IPSLA Ethernet Jitter Precision Statistics" is not supported, the other two vendor certs are not discovered as supported even though they were configured on the devices.
  Resolution: Decoupled the three vendor certs so that they can be discovered properly and independently. .
  (20.2.1, DE433784)
- Symptom: Metric family "Response Path Test Ethernet Jitter" missed two metrics: FrameLossRatioSrcDest and FrameLossRatioDestSrc.
  Resolution: Added two metrics FrameLossRatioSrcDest and FrameLossRatioDestSrc to metric family "Response Path Test Ethernet Jitter".
  (20.2.1, DE433788)
- Symptom: When installing the consul service for Fault Tolerance in the proxy and Data Aggregator, the installer isn't asking which IP address to bind to, if there are multiple IPs on the machine.
  Resolution: Updated the proxy and data aggregator installers to prompt for which IP address to bind consul service to, if there are multiple IPs on the machine.
  (20.2.1, DE434128)
- Symptom: A recurring message of dropped relations appears in the data aggregator whenever it receives an inventory update from the DX NetOps Virtual Network Assurance.
  Resolution: Change the logging level of the message to debug so we aren't always seeing the message, just for debugging.
  (20.2.1, DE434146)
- Symptom: When DX NetOps Spectrum pulls events from Event Manager, you could see duplicate events. This is due to the item being contributed by multiple data sources, and not using DISTINCT in EM DB queries.
  Resolution: Updated the EM DB query to use DISTINCT when getting events to send to DX NetOps Spectrum.
  (20.2.1, DE434444)
- Symptom: The data aggregator might fail to start due to an ArrayIndexOutOfBoundsException when initializing the RelationshipImporter. This is due to inconsistent values in the attributes used to persist previously imported relationships. The inconsistency could have been caused by a bug in the item repository persistence layer where attribute writes that should have been committed in a single transaction were committed individually.
  Resolution: Fixed a bug in the item repository persistence layer's transaction management. Fixed the relationship importer's initialization process to re-initialize the history of previously imported relationships if the history attributes are inconsistent. If the attributes are found to be inconsistent, an error is logged and any automatically generated relationships that had previously been removed are returned and must be removed again via the UI.
  (20.2.1, DE434597)
- Symptom: Items are missing from DX NetOps Virtual Network Assurance due to persistence failures while processing messages. .

Resolution: Updated the DX NetOps Virtual Network Assurance persistence error handling to prevent failures from rolling back message processing. Additionally, updated the persistence logic to correctly persist additional objects that were previously leading to persistence failures.
(20.2.1, DE435483, 20099081,20093248)

- Symptom: Percentage of Actively Memory used was not calculated for ESX hosts.
  Resolution: Updated the vendor cert to support the percentage of Active memory used.
  (20.2.1, DE435721)
- Symptom: After upgrade to 3.7.5, the Dashboards menu might contain duplicate entries. Upgrade was calling create_dashboards, when it should only be calling create_new_dashboards.
  Resolution: Updated installer to not call create_dashboards on upgrade.
  (20.2.1, DE435768)
- Symptom: Users are unable to access Performance Center from the DX SaaS tile page.
  Resolution: Users can now access Performance Center after successfully logging into DX SaaS.
  (20.2.1, DE435872)
- Symptom: Group Scorecards when combining total rate and percentiles metrics throws query processing error.
  Resolution: Addressed error when combining total rate and percentile metrics on the Group Scorecards view.
  (20.2.1, DE435901)
- Symptom: When setting the NetOPs group context to be a spectrum global collection, you only see device level alarms, and not interface level alarms. One Click shows both.
  Resolution: Modified the way that the global collection filter is sent to DX NetOps Spectrum, so that the alarm view now shows device and interface level alarms when the alarm view context is set to a synchronized spectrum global collection.
  (20.2.1, DE436170)
- Symptom: If you select an Interface in Inventory, and then select the IP Performance Tab, and then zoom into one of the charts. After clicking Apply To Dashboard for that chart, the time does not change in the Time Display for the dashboard.
  Resolution: The dashboard's time range field was updated to reflect the "Apply to Zoom" feature.
  (20.2.1, DE436188)
- Symptom: Dateparser was not handling few cases of date format during alarm processing in Versa.
  Resolution: Added the possible date formats.
  (20.2.1, DE436625)
- Symptom: Performance poll was running sequentially for various device types , which is consuming time thereby missing every other poll.
  Resolution: Added Multi Threading for processing performance poll.
  (20.2.1, DE436698)
- Symptom: There is a spike while calculating Delta for DX NetOps Virtual Network Assurance metrics on the reports when the engine or DX NetOps Virtual Network Assurance is down for a particular time.
  Resolution: Data Collector now discards first poll after the DX NetOps Virtual Network Assurance or Engine is down, if the value is more than 2 polls.
  (20.2.1, DE436702)
- Symptom: NetOps user interface experiencing performance issue when attempting to render Trend Chart with Events reporting when greater than monthly period.
  Resolution: Added safeguard to limit the time range to 31 days when attempting to render Trend Chart with Events reporting to address performance impact./ .
  (20.2.1, DE436873)
- Symptom: Unable to Print "SDN/NFV vSwitch Performance Overview" page to PDF, yields "Property missing: Title" exception in log.
  Resolution: This could occur on systems that were upgraded from a pre-3.0.0 release due to a view removed from this page in 3.0.0. The properties of the deprecated view are now restored and the page can now be printed or edited without "Property Missing" exception.

(20.2.1, DE436879)

- Symptom: When an error happens in the middle of a Full Pull data source stage, it can result in all items not sent by the data source to be removed from the Performance Center inventory. This could result in new item ids being assigned to them if they only come from a single data source like spectrum global collections.
  Resolution: Updated the Performance Center Full Pull synchronization logic to only mark items, not sent by the data source, after the stage is complete. For example, this should stop spectrum global collections from getting assigned a new Performance Center item id, if there is an error during Full pull for groups.
  (20.2.1, DE437034, 01142425)
- Symptom: After upgrade of a very high scale system, Performance Center user REST services are significantly slower.
  Resolution: Optimized telemetry queries.
  (20.2.1, DE437335, 20103983)
- Symptom: OK and Cancel buttons on On Demand's Add/Remove Item/Groups dialogs were missing the text (OK, Cancel). They were blank buttons. Also, Add/Remove Group for the On Demand from Search Results emitted a 500 Server error.
  Resolution: Modified the properties files to reflect new page (18 for nested dialog). Also changed add/remove groups to nested dialog.
  (20.2.1, DE437351)
- Symptom: Engine which is deleted an not cleaned up properly is remaining in the collector folder. On Wildfly restart, DX NetOps Virtual Network Assurance is considering it a running engine and trying to start the same.
  Resolution: On Restart of DX NetOps Virtual Network Assurance, now it checks if the engines exist in the DB. All engines which don't exist are deleted.
  (20.2.1, DE437603)
- Symptom: When data aggregator thought Data Collectors were disconnected, it aborts all currently running discoveries. Due to a un-handled exception in the aborting process, the counter of the running discoveries was not decreased. When the number of running discoveries reached to the maximum number of concurrent discoveries, all discoveries are stuck on Performance Center.
  Resolution: Improved the discovery abort process and handled all the exceptions so that the discovery counter can work properly.
  (20.2.1, DE437748)
- Symptom: With On-Demand report view option Chart per Item with Multiple Metrics when reporting at the component level item titles on chart can mismatched with graph data.
  Resolution: Addressed titles on chart mismatched with graph data with On-Demand report view option Chart per Item with Multiple Metrics when reporting at the component level.
  (20.2.1, DE437961, 20108850)
- Symptom: Data Collector could fail to start due to a NULL pointer exception when it tries to sync items from the Data Aggregator.
  Resolution: Improved the sync process between data aggregator and Data Collector to handle all possible exceptions properly.
  (20.2.1, DE438315)
- Symptom: Some fields in out product weren't adequate protection against HTML/JavaScript injection. .
  Resolution: We added the injection protection to all the identified fields.
  (20.2.1, DE438451)
- Symptom: One of the services displays a stack trace to the client when an exception was thrown. This potentially gives and attacker information about the underlying code.
  Resolution: The stacktrace is now logged and sent back a standard server error code.
  (20.2.1, DE438461)
- Symptom: When DX NetOps Virtual Network Assurance receives a time filter larger than 24hrs, it throws an error, which results in client not able to get further updates.
  Resolution: DX NetOps Virtual Network Assurance now sends a FULL update on receiving a time filter larger than 24hrs, which helps auto renew connections with the client.

(20.2.1, DE438556)

- Symptom: In a system with DX NetOps Virtual Network Assurance installed, the data aggregator/Performance Center sync process can be very expensive. It checks the item creation time on all items in the system to find out the newly created DX NetOps Virtual Network Assurance domain group items.
  Resolution: Improved performance of DA/Performance Center sync process by dramatically narrowing down the search scope.
  (20.2.1, DE438808)
- Symptom: Tunnels from the router(customer gateay) which was discovered through snmp are missing./Its not the issue of snmp reconciliation.
  Resolution: There was an aws technology specific code written in data aggregator based on the vendor attribute, due to which further code that builds tunnels getting restricted. Upon discussion with Abe/Li this code cleaned-up, that fixed the issue.
  (20.2.1, DE438825)
- Symptom: If any ec2 is in terminated state in AWS console and its ip's are deleted and empty/null ip not handled, due to which entire inventory failed.
  Resolution: Terminated ec2's are handled so that upon null/empty ip, the entire inventory poll shall not fail.
  (20.2.1, DE439129)
- Symptom: When rendering shade area on trend chart related to standard deviation the processing performance impacted memory resource usage.
  Resolution: Address the processing performance impact when rendering shade area on trend chart related to standard deviation.
  (20.2.1, DE439284)
- Symptom: 1. The data aggregator falls behind in processing DX NetOps Virtual Network Assurance changes. This results in models missing from existing DX NetOps Virtual Network Assurance Domains or else DX NetOps Virtual Network Assurance Domains not being created after configuring an additional SDN Gateway.2. All of the Groups under a DX NetOps Virtual Network Assurance Domain disappear after toggling the Administration Status of the applicable SDN Gateway.
  Resolution: Enhancements were made to the DX NetOps Virtual Network Assurance integration to allow inventory updates to be processed much faster. These enhancements also added more robust logic to avoid exceptions previously seen during response processing and also to prevent Items and Relationships from incorrectly being deleted when toggling the Administration Status of a DX NetOps Virtual Network Assurance.
  (20.2.1, DE439424)
- Symptom: Distributed Item Repository communication between Data Aggregator and all Data Collectors can be disrupted if one Data Collector gets into a bad state. An example is seen when a Data Collector broker stops consuming messages from the DA, causing a back up in its DIM.request queue. Once the data aggregator broker detects this queue is 100% full, all other data aggregator/Data Collector communication is disrupted.
  Resolution: Reworked data aggregator side Distributed Item Repository processing to isolate workloads per Data Collector so that when one Data Collector gets into a bad state, communication is not disrupted for all Data Collectors.
  (20.2.1, DE439788)
- Symptom: Time bar view does not sort as expected when initially rendered on the page, required column selection to sort correctly.
  Resolution: Addressed issue where time bar view does not sort as expected when initially rendered on the page, required column selection to sort correctly.
  (20.2.1, DE440194)
- Symptom: Running the `testssl.sh` script against Performance Center shows vulnerabilities for "Secure Client-Initiated Renegotiation", "LOGJAM
  (CVE-2015-4000)", and "LUCKY13
  (CVE-2013-0169)".
  Resolution: Vulnerabilities for "Secure Client-Initiated Renegotiation", "LOGJAM
  (CVE-2015-4000)", and "LUCKY13
  (CVE-2013-0169)" have been resolved in Performance Center, and are no longer flagged by the `testssl.sh` script.

(20.2.1, DE440319)

- Symptom: The Data Aggregator misinterprets an exception from the database as an outage and shuts down.
  Resolution: Updated the regular expression for expected exceptions from the database that should not be interpreted as an outage.
  (20.2.1, DE440348)
- Symptom: Report Analyzer view, Interfaces over Threshold, renders missing image icon on interface name column with Firefox.
  Resolution: Addressed the missing image icon with Firefox appearing on interface name column when rendering Report Analyzer view, Interfaces over Threshold.
  (20.2.1, DE440550)
- Symptom: Cannot delete group using Performance Center REST service, when the user has "My Assigned Groups".
  Resolution: Fixed permission checking code in Performance Center REST API.
  (20.2.1, DE440557)
- Symptom: When upgrading Performance Center, the service (Performance Center, DM, EM, sso) properties files, that store settings, loses any user added entries.
  Resolution: Modified install to preserve any user added settings in the service properties files.
  (20.2.1, DE440676)
- Symptom: Some of the metrics were missing and for those, broker code throwing Null Pointer exception.
  Resolution: Null check added for the metrics, so that it avoids the null pointer exceptions.
  (20.2.1, DE441345)
- Symptom: CSV report export only includes first page for legacy systems that upgraded to Service Pack 3.7.4 or above.
  Resolution: Addressed upgrade Service Pack 3.7.4 or above issue that CSV report export only includes first page.
  (20.2.1, DE441538)
- Symptom: When Domain is a selected inventory column rendering No Data when applying filter to table view search.
  Resolution: Addressed issue when applying filtered search to table with Domain inventory column that sometimes renders No Data to Display.
  (20.2.1, DE441661)
- Symptom: After configuring a Scorecard table that contains percentile metric; when removing percentile metric from list metrics with hierarchy option still renders percentile entry.
  Resolution: Addressed issue when removing percentile metric from Scorecard Table with hierarchy option still renders percentile entry in view.
  (20.2.1, DE441793)
- Symptom: Certain DX NetOps Virtual Network Assurance entities are deleted and recreated every poll cycle even though there are no changes occurring in the environment.
  Resolution: The Persistence logic in DX NetOps Virtual Network Assurance has been updated. With the new implementation, if keys are to be removed from the ID service, the key is checked to make sure it associated with the entity's ID. If it is associated with a different ID, a warning is printed and it is no longer removed from the ID service.
  (20.2.1, DE442571)
- Symptom: The global search can hang on a search string that contains '=' character.
  Resolution: With this fix, the global search works, however if the search string contains '=' character, add a new '=' character to the beginning of the search string.
  (20.2.1, DE442778, 20140853)
- Symptom: SslConfig on Performance Center and `sslConfig.sh` on data aggregator uses SHA1WithRSA when generating private key for HTTPS use.
  Resolution: Updated both tools to use SHA256WithRSA when generating private key for HTTPS.
  (20.2.1, DE442855)
- Symptom: When Spectrum system time is behind Event Manager system time, events might not be getting created in DX NetOps Spectrum when pulled from Event Manager.
  Resolution: Updated Event Manager to only handle when current time on Event Manager is older than the end timestamp being requested by DX NetOps Spectrum. It time shifts the start/end request timestamps to EM local

timestamp. Event Manager no longer time shifts forward Spectrum start/end request timestamps, if older than Event Manager local timestamp.
(20.2.1, DE443851)

- Symptom: When business hour filter is applied to context page incorrectly shown on view subtitle for data source not related to Data Aggregator.
  Resolution: Addressed view subtitle showing business hour filter on context page for data sources that are not related to Data Aggregator.
  (20.2.1, DE444080)
- Symptom: Generate URL not always initially rendering correctly when when view container is enabled on URL Generate dialog.
  Resolution: Addressed problem when generate URL that view is collapsed if view container is enabled on URL Generate dialog.
  (20.2.1, DE444200)
- Symptom: If the user's time zone causes "last 8 hours" to shift into the prior day, then the dashboard's time display incorrectly says Today instead of yesterday's date.
  Resolution: With this fix, the timezone is now included when checking to see if the date is today's date.
  (20.2.1, DE444215)
- Symptom: Calendar Heat charts were not accessible.
  Resolution: Calendar Heat charts are now fully accessible and meet Section 508 accessibility requirements.
  (20.2.1, DE444413)
- Symptom: Reports using custom group of SDWAN tunnels shows no data. Custom groups of devices shows data for SDWAN tunnels they are a participant in.
  Resolution: Updated Data Aggregator SQL view to check if the SDWAN tunnel is in the group, in addition to either device of the tunnel.
  (20.2.1, DE444464)
- Symptom: Vulnerable version of jQuery.
  Resolution: Updated to latest version of jQuery.
  (20.2.1, DE444735)
- Symptom: When Performance Center restarts, if MySql does not start in 6 seconds, then the Performance Center services fails to start.
  Resolution: Fixed comparison in MySql check script to do integer comparison, not string compare that resulted in the check exiting earlier than expected. Also increased check to 120 seconds.
  (20.2.1, DE445174)
- Symptom: NFA Hierarchy sub-table for By Host and By Protocol intermittently does not display the interface data, as item are selected in top table the lower table is still blank.
  Resolution: Addressed intermittent issue when selecting item in top table the lower table does not render interface data with NFA Hierarchy sub-table for By Host and By Protocol view.
  (20.2.1, DE446176)
- Symptom: Vcenter is sending IP address in invalid format, which is causing Persistence failures.
  Resolution: Added check for Invalid IP addresses coming from the Vcenter. The plugin lookups for IP address for the VM, if the IP address is invalid.
  (20.2.1, DE446231)
- Symptom: During global synchronization stage on Performance Center, a SQL exception might be thrown during tenant or IP domain stage about truncating LocalID in ds_items table. Large model handles from DX NetOps Spectrum were causing the SQL exception.
  Resolution: Updated the SQL call to do string comparison instead of numeric. The LocalID column is a string and should've been comparing as a string but was using numeric comparison that could result in truncation issues.
  (20.2.1, DE447006)
- Symptom: After configuring Scorecard Trend view with baseline metric renders 'No Data to Display'.
  Resolution: Addressed issue when selecting baseline metric with Scorecard Trend view renders 'No Data to Display'.

(20.2.1, DE447990)

- Symptom: When selecting SD-Wan Tunnel/App Path metrics By Component level with the Scorecard Trend might be missing component items.
  Resolution: When selecting SD-Wan Tunnel/App Path metrics By Component level with the Scorecard Trend always render component items.
  (20.2.1, DE448032)
- Symptom: NFA Stacked Protocol Trend views rendered No Data to Display when time range is greater than 2 hours.
  Resolution: Addressed No Data to Display rendered on NFA Stacked Protocol Trend views when time range is greater than 2 hours.
  (20.2.1, DE448349, 31842884,31828116)
- Symptom: New NFA views on interface context page has default row value of 1000 that creates long running view processing.
  Resolution: New NFA views on interface context page modified row items in view selection and default row value to avoid long running view processing, altered views include; IP Performance Protocol- Protocol List, IP Performance ToS - ToS List, IP Performance Host - Hosts List, IP Performance Conversation - Conversation List.
  (20.2.1, DE449080)
- Symptom: Installers complain about SLES 12 SP4+ not passing kernel check.
  Resolution: Updated installers to pass kernel check with SLES 12 SP4, and SP5.
  (20.2.1, DE449181, 20310176)
- Symptom: When doing a new data aggregator fault tolerant silent install, the data aggregator data directory uses default instead of the `DA_DATA_HOME` value in properties file provided.
  Resolution: Updated silent installer logic to not set the `DA_DATA_HOME` to a default if already set via properties file.
  (20.2.1, DE449615)
- Symptom: DX NetOps Virtual Network Assurance was not sending the correct out of band management IP address for some Versa devices.
  Resolution: A new configuration has been added to DX NetOps Virtual Network Assurance that enables finding and sending the correct out of band management IP address for Versa devices.
  (20.2.1, DE450038)
- Symptom: When a device has more than 150 interfaces the Alarms View on Alarms tab of device context page fails with error, "An unexpected error has occurred".
  Resolution: Addressed issue that Alarms View on Alarms tab of device context page fails with error when a device has more than 150 interfaces.
  (20.2.1, DE450059)
- Symptom: A cross-site scripting vulnerability exists for a url passing javascript to the "pg=" parameter,
  like this example which raises an alert dialog: https://your_pc_host:8182/pc/desktop/page?timeRange=0&endTime=2020-01-28+14%3A25&startTime=2020-01-28+13%3A25&mn=5&pg='%2Balert(15)%2B'&parentid=13
  Resolution: The vulnerability for the "pg=" parameter has been resolved - the alert in the example url no longer executes./ .
  (20.2.1, DE451032)
- Symptom: Plugin logger exposing passwords in plaintext when category OC_ACQUISITION is set to level DEBUG. Sample message: oc.log:2020-03-11 13:53:43,328 DEBUG (EE-ManagedThreadFactory-default-Thread-3) Viptela_17664d4c-c80b-44b1-8722-aa768696e4e9=Inventory Poll [OC_ACQUISITION] TimReflectionHelper 213 inject vman#2837 to method setPassword.
  Resolution: Added logic to the OC_ACQUISITION plugin logger to suppress debugging of password values.
  (20.2.1, DE451580)
- Symptom: When SNMP MIB table read encountered PARTIAL_FAILURE error during a metric family change detection process, it could create some incomplete component items. Next time when the MIB table is read successfully, the incomplete items are marked as Not Present. If a device's SNMP agent frequently returns PARTIAL_FAILURE for its MIB table reads, a lot of Not Present components is created.
  Resolution: Improved the metric family discovery not to create any component items when a SNMP PARTIAL_FAILURE is encountered. .

(20.2.1, DE451685)

- Symptom: When sending email via STARTTLS, Performance Center only allows TLSv1.0. If email server is TLSv1.1 or TLSv1.2, the email server rejects the email connection.
  Resolution: Updated Performance Center to support TLSv1.2, TLSv1.1 and TLSv1.0 when email server supports STARTTLS.
  (20.2.1, DE451724)
- Symptom: When LogoutService is called during SAML logout, it throws an exception "net.shibboleth.utilities.java.support.component.UninitializedComponentException: Component has not yet been initialized and cannot be used." while trying to decode the HTTP request.
  Resolution: Resolved the exception by initializing the decoder code before using it. This exception was hit due to admins trying to not have Performance Center logout of SAML server when logging out of Performance Center. So this fix also includes code to allow admins to not provide a SingleLogoutService in the metadata XML from the IDP. When Performance Center/sso sees there is no SingleLogoutService, it logs the user out of Performance Center. The user can still remain logged into SAML.
  (20.2.1, DE452302)
- Symptom: ACI filter Configuration is reset to default after the upgrade and all customer has to re-configure the same .
  Resolution: It is a code fix to take backup of existing filter configurations and retain the same after upgrading DX NetOps Virtual Network Assurance.
  (20.2.1, DE452507)
- Symptom: Daily scheduled email jobs are sent multiple times during Spring DST transition when email TZ is UTC and scheduled time is between 00:00 - 01:00.
  Resolution: Some combinations of Performance Center host TZ
  (a DST TZ like ESTEDT) and scheduled email TZ
  (non-DST TZ like UTC), when scheduled time is between midnight and 1AM were leading to incorrect calculation of next run time, causing run time to be repeated until after 1AM. The date math has been corrected to better account for the email timezone when calculating next run time.
  (20.2.1, DE453071)
- Symptom: The DX NetOps Virtual Network Assurance Meraki plugin was fetching the client performance data for only a few of the access points.
  Resolution: The DX NetOps Virtual Network Assurance Meraki plugin now fetches the client performance data for all the access points.
  (20.2.1, DE453478)
- Symptom: The DX NetOps Virtual Network Assurance Meraki plugin was not discovering any access point without a lan ip address.
  Resolution: The DX NetOps Virtual Network Assurance Meraki plugin now discovers access points without a lan ip address.
  (20.2.1, DE453486)
- Symptom: If other (1) and softwareLoopback (24) are re-enabled in Interface or High Speed Interface vendor certifications, users cannot select them in monitoring profile filters for Interface metric family under Type.
  Resolution: Updated monitoring profiles Interface metric family filter support for Type to show other and softwareLoopback values, and treat them as supported.
  (20.2.1, DE453520)
- Symptom: Custom Gauge View automatically scales the gauge needle when using the fixed end point combined with percent option.
  Resolution: Custom Gauge View allows for option whether to scale the gauge needle when using the fixed end point combined with percent option.
  (20.2.1, DE453522)
- Symptom: Interfaces were not processed when an HTTP Request fails ,which is creating a null or empty file.
  Resolution: Added null check for content in the files when HttpRequestFailure occurs.
  (20.2.1, DE454311)
- Symptom: Suborgs were not processed earlier in versa plugin.
  Resolution: Added support for processing SubOrgs.

(20.2.1, DE454336, 31812847)

- Symptom: DX NetOps Virtual Network Assurance installer is allowing duplicate domain names across multiple DX NetOps Virtual Network Assurances.
  Resolution: Updated installer banner text and DX NetOps Virtual Network Assurance Swagger update domain with a caution message "Domain name should be unique across VNAs".
  (20.2.1, DE455141)
- Symptom: After upgrading to 3.7.10, the following symptoms were seen on certain data aggregators that were monitoring SDWAN environments with DX NetOps Virtual Network Assurance: 1. High CPU Usage after upgrade caused Performance Center to lose contact with the data aggregator (see: \u003Ca class="cke-link-popover-active" href="https://knowledge.broadcom.com/external/article?articleId=188059"https://knowledge.broadcom.com/external/article?articleId=188059\u003C/a); 2. Changes in the monitored DX NetOps Virtual Network Assurance environment were not being reflected in Performance Center. These include, but might not be limited to: Sites, Tunnels and SLAPaths; 3. Prior to and after upgrading to 3.7.10 "duplicate" / "stale" items and SDN Domains were seen in Performance Center under the SDN Domains group.
  Resolution: The data aggregator's processing of DX NetOps Virtual Network Assurance data has been updated to be more efficient during the processing of data from SDWAN environments. The changes include: 1. A reduction in the upper limit of the number threads available for SDN processing to 1/3 of the CPUs on the Data Aggregator; 2. General improvements to the DX NetOps Virtual Network Assurance processing to improve the efficiency of processing DX NetOps Virtual Network Assurance updates from SDWAN environments; 3. Additional protections have been added to the DX NetOps Virtual Network Assurance processing to resolve missing items seen after upgrading; 4. SDN processing in the data aggregator has been updated to prevent Items from being created for SDN Gateways that had already been deleted from the DA.
  (20.2.1, DE455285)
- Symptom: In the CSV output OpenAPI might not correctly match expand metric family data with its corresponding configuration entity for example a query like the following `devices?$expand=interfaces,portmfs` might return metric data in the line for a different interface.
  Resolution: With this fix, OpenAPI now matches all expand data correctly.
  (20.2.1, DE455825)
- Symptom: When using DX NetOps Virtual Network Assurance as an aggregator inventory items/keys are removed after upgrade. This is especially applicable to DX NetOps Spectrum. After upgrade, it was unable to reconcile the changes which has the effect of the system creating duplicate items.
  Resolution: Migration script modified to only modify ID cache if DX NetOps Virtual Network Assurance is not operating in "aggregator" mode.
  (20.2.1, DE456076)
- Symptom: For consolidated devices, Performance Center maintained IP address of the device that was sync'ed up first (Data aggregator device IP was not given priority over Spectrum).
  Resolution: Updated Global sync to give data aggregator device IP priority over Spectrum device IP during device consolidation based on primary-secondary IP matching.
  (20.2.1, DE456272)
- Symptom: Reporting views using business hours, that have multiple timezones to consider, can take minutes to run the query in the database. This is partly due to the amount of memory the database reserves to run the query, which is based on 65k characters for timezone values.
  Resolution: Updated the sql view, for comparing timezone, to select only 255 characters from the timezone attribute. This results in far less memory being needed to run the query. It could result in steps of the query now being multi-threaded, where before it couldn't due to memory requirements. This could result in some business hour queries running much faster than before.
  (20.2.1, DE456460)
- Symptom: When TransientDBConnection logging is set to debug, SQL statements with passwords are logged.
  Resolution: Update logger to sift out SQL with passwords.
  (20.2.1, DE456484)
- Symptom: Web service calls to DX NetOps Spectrum for Alarm Console or the subscription alarm service might timeout after 100 seconds. This usually is due to DX NetOps Spectrum's response time to the request being made.

Resolution: Updated the timeout to 120 secs, but also made it configurable via the netqosportal.general table or Performance Center admin debug global attributes page. To increase the Alarm Console timeout, run: REPLACE INTO netqosportal.general VALUES('AlarmView_WebService_Timeout', '<timeout in seconds>');
To increase the Spectrum alarm status subscription timeout, run: REPLACE INTO netqosportal.general VALUES('SpectrumStatus_WebService_Timeout', '<timeout in seconds>');.
(20.2.1, DE456697, 20294595)

- Symptom: Performance Center global synchronization fails if duplicate tunnels and/or SLA paths are synchronized from any data source. Error similar to the following is found in the `DMService.log` file upon failure: "MySQLIntegrityConstraintViolationException: Duplicate entry '5868174-14' for key 'PRIMARY".
  Resolution: Updated the global sync SQL calls to not error when duplicate tunnels/SLA paths are sent from data source(s).
  (20.2.1, DE456711)
- Symptom: Data Aggregator operations that require internal attribute reads such as data aggregator REST calls, DX NetOps Virtual Network Assurance inventory processing, etc are subject to slowness at high concurrency due to excessive use of java locking objects.
  Resolution: Refactored Item Repository read functionality to more efficiently use java locking objects.
  (20.2.1, DE456931)
- Symptom: On Deletion of Engine, a few Entities are not deleted.
  Resolution: The issue is caused due to inconsistent Hashcode on the Properties. Updated the hashcode.
  (20.2.1, DE458021)
- Symptom: After applying edit changes context page with F5 device there is an error when attempting to save context tab, requiring name entry. Workaround required add entry to context tab prior to being able to save modifications.
  Resolution: Address error that required F5 context tab manually add title entry to context tab prior to being to save modifications on context page.
  (20.2.1, DE458272)
- Symptom: Various security CVEs reported against the version of Consul used by Performance Management.
  Resolution: Upgraded the version of Consul to 1.7.2. For the new version of Consul to be enabled after Performance Center is upgraded, the user must run the following commands. If these commands are skipped, the old version of Consul remains running after the install.
  Run the following commands:
  a. `service capc-consul stop`
  b. `rm -rf <CA_Performance_Center_Directory>/consul/data/*` **Example:** `rm -rf /opt/CA/PerformanceCenter/consul/data/* )`
  c. `service capc-consul start`
  (20.2.1, DE458789)
- Symptom: The DX NetOps Virtual Network Assurance Viptela plugin is sending data with many disparate sample time values, resulting in many timestamps with few items in the sdn_tunnel_rate table in the DR. This is inefficient for threshold evaluation because each timestamp results in a threshold evaluation query in the DR. This causes long running threshold evaluation times for the NormalizedSDNTunnelInfo metric family and causes the System Health to indicate a degraded state for Data Aggregator Threshold evaluation.
  Resolution: Provided an option in the Data Collector to modify sample times for each incoming DX NetOps Virtual Network Assurance metric so that they appear on a boundary (for example, samples collection at 1:00, 1:02, 1:03, et al) has a sample time of 1:00. This greatly reduces the number of timestamps that need to be evaluated for threshold evaluation, thus keeping the evaluation time for the NormalizedSDNTunnelInfo metric family to comparable values as other metric families.
  (20.2.1, DE459147)
- Symptom: After an upgrade of Performance Center, entries for service properties files add backslash before colons and equals, causing some of the scripts to not run successfully.
  Resolution: Updated the upgrade custom code to not add backslashs before colon and equals characters in service properties files.

(20.2.1, DE459183)

- Symptom: The ACI Console is very unresponsive to mouse clicks in environments that contain a large number of groups.
  Resolution: The ACI Console was updated to use more efficient OpenAPI queries for the performance metrics displayed in the dashboard.
  (20.2.1, DE459510)
- Symptom: If a Calix chassis has multiple blades that shares the same IP address of the chassis, only one of them can be discovered properly.
  Resolution: Improved the Calix blade discovery to properly handle the shared IP on the blades.
  (20.2.1, DE459921)
- Symptom: When not providing a SsoProductCode to the sign-out.jsp, a NullPointerException are shown.
  Resolution: Updated the logout code to return an informative error message, if no SsoProduceCode is provided.
  (20.2.1, DE460006)
- Symptom: Alarms are generated with unknown severity.
  Resolution: Made changes in AlarmNotifications to not change the severity to unknown when the alarms are cleared.
  (20.2.1, DE460254)
- Symptom: When using a combo-box, the pop-up menu associated with the control always "opens downwards". If the menu is positioned near the bottom of the browser window, it is clipped such that some or all of the menu is not available.
  Resolution: When using a combo-box, the pop-up menu associated with the control now opens "upward" if it is clipped by the bottom of the browser window.
  (20.2.1, DE460315)
- Symptom: When global synchronization runs, it might throw a SQL exception
  like: com.mysql.jdbc.MysqlDataTruncation: Data truncation: Data too long for column 'SourceIds' at row 672.
  Resolution: Fixed the SQL statement so it won't throw a data truncation error causing global sync to fail.
  (20.2.1, DE460948)
- Symptom: VM and Host metrics were polled on the wrong values on the vsphere client.
  Resolution: VM and Host Metrics are polled now on the correct values from the Vsphere client.
  (20.2.1, DE461311)
- Symptom: The value of Added By column of group items created by a rule could be overwritten if a new item is added to the group manually.
  Resolution: With this fix, a new item added to a group manually does not override Added By column value of the items created by a rule.
  (20.2.1, DE463375)
- Symptom: Some telemetry SQL queries in Performance Center could still run a long time at scale, due to not using available indexes. They could interfere with queries run for UI tasks, making the UI unresponsive.
  Resolution: Updated the remaining telemetry SQL queries that needed additional comparisons to cause them to use indexes to speed up queries.
  (20.2.1, DE464384)
- Symptom: The product key used for on-prem telemetry is invalid.
  Resolution: Update the product key for on-prem telemetry to the correct value.
  (20.2.1, DE464806)
- Symptom: odata query might not return device information in expand when it starts from metric family/component, for example portmfs?$expand=device.
  Resolution: With this fix, odata query now returns device information in expand when it starts from metric family/component, for example portmfs?$expand=device.
  (20.2.1, DE465743)
- Symptom: The Data Collector might not start polling all components if many calls are made to data aggregator REST webservice to start/stop polling on an individual component.
  Resolution: Fixed NullPointerException in the Data Collector so we can better handle polling change requests from the Data Aggregator.

(20.2.1, DE466104)

- Symptom: Running a Discovery Profile for ranges that were discovered by a DX NetOps Virtual Network Assurance Gateway can lead to items being deleted if the IP addresses are pingable but not SNMP contactable.
  Resolution: Updates were made to the discovery process which now prevents items from being deleted when they have been previously discovered by a DX NetOps Virtual Network Assurance Gateway.
  (20.2.1, DE466358)
- Symptom: Open API in some circumstances might not honor tenants boundaries.
  Resolution: With this fix, Open API should honor tenant boundaries all the tine.
  (20.2.1, DE466422)
- Symptom: On the System Health page, when you page to the next page in a table the section is collapsed each time.
  Resolution: Instead of setting the collapsed property to be purely based on the status of the section, the product also takes into account the current collapsed state.
  (20.2.1, DE467043)

Known Limitations

The following limitations have been identified in this release of DX NetOps Performance Management:

Alarms Tab on Customized Context Pages

The Alarms tab is automatically added to most device-level context pages. For customized device-level context pages, you must manually add the Alarms tab. If you have many tenants with customized device-level context pages, see the ReadMe file in the following location to streamline the addition of the Alarms tab:

`/PC_install_directory/PerformanceCenter/SQL/plugins/custom_context_spectrum`

Apostrophes in Custom Attribute Descriptions

When a custom attribute description contains an apostrophe, the tooltip for the attribute does not render correctly in some situations.

Blank CA Business Intelligence Dashboard Output

In some cases, synchronized users (created in NetOps Portal) see blank PDF output from scheduled reports.

In some cases, users see blank PDF output because of a misconfiguration in the `js.config.properties` file.

For more information, see the Jaspersoft Community.

Broken Links in a DX Spectrum Integration

In an integration with DX NetOps Spectrum, you might find broken links from DX NetOps Spectrum OneClick to NetOps Portal. For the linking to work correctly, the fully qualified domain name is required.

To prevent this issue, complete the following steps:

1. Log in to the Performance Center host.
2. Navigate to the Performance Center directory:
   `cd PC_Install_Directory/PerformanceCenter`
3. Launch the SSO Configuration utility:
   `./SsoConfig`
4. Select and run option 1: CA Performance Center.
5. Select and run option 3: Performance Center.

6. Select and run option 1: Remote Value.
7. Select and run option 7: Web Site Host.
8. Enter `u` to update.
9. Specify the fully qualified domain name of the Perfomance Center host.
10. Sync the DX NetOps Spectrum data source.

Bulk Data Export Unsupported in Fault Tolerant Environments

The bulk data export feature is unsupported in environments with fault tolerant Data Aggregators.

CA Business Intelligence Chart Errors in Scheduled Jobs

When a scheduled CA Business Intelligence (CABI) report or dashboard is executed, pie charts may appear incomplete or may be missing a color.

This issue occurs in the CA Business Intelligence JasperReports Server.

To prevent this issue, complete the following steps:

1. Open <*CABI_INSTALL*>/WEB-INF/js.quartz.base.properties.
2. Change the default values as follows:

```
org.quartz.jobStore.clusterCheckinInterval = 3600000
org.quartz.threadPool.threadCount=9
org.quartz.threadPool.threadPriority=9
org.quartz.jobStore.misfireThreshold=3600000
```

CA Business Intelligence Failed Export for Scheduled Jobs

If you export everything for a scheduled job from the CABI user interface, the dashboards are not exported.

To prevent this issue, use the user interface, js-export, or the REST API to export each dashboard individually.

CA Business Intelligence Report and Input Control Issues

Scheduled CABI reports and saved input controls for reports might work incorrectly after upgrade. This issue occurs because this release introduces changes to the report input controls. Some input controls were added, some were removed, and some were changed. For example, if a TopN report was scheduled for two metrics of Interface metric family, after you upgrade, the scheduled report shows four variables. We recommend that your recreate scheduled reports and report templates after upgrade.

cEdge Device and Tunnel Data Unavailable

DX NetOps Virtual Network Assurance collects inventory and performance metrics from Viptela to support DX NetOps Performance Management SD-WAN monitoring. While the plug-in collects inventory for vEdge routers, vEdge interfaces, tunnels, and application/SLA paths, it cannot collect data for cEdge devices due to a Cisco bug. This limitation results in missing cEdge devices in the interface inventory and missing tunnels that originate or terminate from or on cEdge interfaces.

For more information, see the Cisco documentation.

Data Collector Polling Status Issue

Before you upgrade the Data Collectors, after you upgrade the Data Aggregator to the 20.2 GA release, the Polling Status column on the System Status page inaccurately says "Not Connected" for each Data Collector. However, a connection

remains. No data loss should occur. After you upgrade the Data Collectors, the correct status appears. This issue will be fixed in releases after the 20.2 GA release.

Deprecated Top Performance by Application View

A change starting with CA Application Delivery Analysis 10.5 deprecated the Top Performance By Application view.
For this version of DX NetOps Performance Management, the new Port List field replaces the Begin Port and End Port fields.
The deprecated view is now the Top Performance By Application - Deprecated view.
The new Top Performance By Application view was created to replace the deprecated view.
If all versions of CA Application Delivery Analysis have been upgraded to at least version 10.5, administrators can edit the dashboards and can replace the deprecated view with the new view.
If not all Application Delivery Analysis data sources are above the specified version, administrators can edit the view title to remove the "Deprecated" text.

Disabled Trend Charts with Events

Trend charts with events are disabled when the time range is greater than 3 months.

Device Context Page:

- Availability Trend with Events
- Average CPU Utilization Trend with Events
- Average Memory Utilization Trend with Events

Interface Context Page:

- Interface Utilization/Discard Out Trend with Events
- Interface Utilization/Discard In Trend with Events
- Interface Utilization Out Trend/Baseline Detail with Events
- Interface Utilization In Trend/Baseline Detail with Events

Empty PDFs After Scheduled Job Executions for CABI Dashboards

The PDFs of DX NetOps Performance Management Monitoring Status and DX NetOps Performance Management Overview – Product Usage dashboard reports are not generated after a scheduled job execution. There is no workaround to generate the PDFs for the two dashboard reports as we do not have a menu item to save the reports in the PDF format.

Extended Flow Views

You can drill down into Network Flow Analysis data for an interface within NetOps Portal without having to navigate to Network Flow Analysis.

Extended flow views have the following known limitations:

- The search box on the related table views does not filter the content in the data results.
- Extended flow views on the interface context page cannot be exported in a scheduled report.
- Extended flow views on the dashboard page cannot be exported in a scheduled report containing all pages.
- The maximum rows are limited for performance. By default, the Top Selection table views are 1000 rows and the Trend Table views are 25 rows.
- For upgrades only, you must manually add extended flow views to customized interface context pages. If you have many tenants with customized interface context pages, do the following post-installation steps:
  a. On the NetOps Portal server, change to the following directory:
     `cd PC_install_directory/SQL/plugins/reporter`
  b. Follow the steps described in the ReadMe file.

NOTE

Even for environments with Network Flow Analysis 9.x and earlier, we recommend that you run the post-installation script. Running the post-installation script helps to ensure compatibility with the latest version when it is enabled.

- If the data source is Network Flow Analysis 10.x and higher, the links in the charts and tables in the IP Performance page open the corresponding network flow pages within DX NetOps Performance Management. If the data source is Network Flow Analysis 9.x and earlier, the links open the corresponding pages in the Network Flow Analysis console.

For more information, see Context Pages.

Flow Administration Issues with IPv6

If the Flow Administration page is empty, check `PCService.log` .

If the following message appears, the Network Flow Analysis Console server might be configured with an IPv6 address:

```
Failed in sending REST: http://[0000:0000:0000:0000:0000:0000:0000:0001]:8981/odata/api/AuthToken
 java.net.ConnectException
```

Follow these steps:

1. Disable the IPv6 address.
   For more information, see Windows Support.
2. Make sure only one Network Interface Card (NIC) is present on the Network Flow Analysis Console server.
3. Run the following command and verify that there are no IPv6 addresses:
   ```
   ipconfig
   ```

Future Web Services Deprecation

When you use the NetOps Portal UI to export vendor certifications, the **genericWS** format is used. The **genericWS** format will be deprecated in a future release. Therefore, we recommend that you use the **typecatalog** web services instead of the NetOps Portal UI.

For more information about using the **typecatalog** web services, see Create or Extend Vendor Certifications.

Group Scorecard Table Metric Fields

The first time you edit the view settings for a group scorecard table view you might encounter an issue with the selected metrics. If you change between Hierarchy Calculate Levels and Metric Calculate Levels, your selected metric fields might clear. For example, if you select several metric fields, and then change the selection from **Device Hierarcy** to **by Device**, your selected metric fields clear.

Input Parameter for Schema During CABI Installation Not Updated

Issue:
During the CABI content installation when you provide the input parameter for schema as "https", the DX NetOps Performance Management Data Source page schema field displays "http".

Workaround:

After you install CABI content, login as administrator in the CABI server.

Open View, Repository page and extract "root-Public-ca-Performance Management - datasources" folder.

Edit the CA PM datasource to change Schema parameter to "https" and click Save.

You need not restart the CABI service.

Invalid Certificates

- For the first discovery, invalid certificates might cause devices, interfaces, and tunnels to be missing.
- After discovery, invalid certificates remove all items that belong to any invalidated devices (routers, interfaces, tunnels, and possibly sites).

Keys for ActiveMQ Communication

If you have authenticated and encrypted activeMQ communication, you must regenerate the keys before this upgrade. For more information, see Authenticate and Encrypt ActiveMQ Communication.

> **NOTE**
> For the first and last name prompt, you must enter the host name of the system where you are creating the certification.

Without new keys, the Data Aggregator and Data Collector cannot communicate.

MultiViews

You can now select multiple metrics for MultiViews. However, existing legacy MultiViews remain the same during upgrade and do not support multiple metrics. If you want an existing view to include multiple metrics, configure a new view in the dashboard builder.

Screen Reader Limitations

When you use a screen reader with the Chrome browser, the contents of charts (for example, points, legend entries, and so on) are not read aloud. This issue is cause by a limitation with the Chrome browser. When you use a screen reader with other browsers including Firefox and Safari, the contents of charts are read alound.

When you use a screen reader to read the contents of charts, the up and down arrow keys are not available. Usually, you must use the screen reader key in conjunction with the left and right arrow keys to move focus in the chart.

When you use a screen reader to read the contents of grids using Internet Explorer, the tooltips associated with icons in the grids are not always read aloud due to a limitation in Internet Explorer.

When use a web browser with the zoom setting of the browser set to greater than 100%, the minimum supported browser window size increases by the zoom factor. For example, if the zoom setting is set to 200%, the minimum required browser window width is 2048 pixels (1024 pixels times 2).

SDN/NFV Dashboard Limitations

The VNF Count by Type Stacked Chart in the SDN/NFV Virtual Inventory Overview dashboard does not show data for the last period. The data for the same period appears when you export the data to a CSV file.

SD-WAN Monitoring

The following limitations apply to SD-WAN monitoring:

- All SD-WAN tunnel reporting is processed at the device component level. Parent devices do not appear in the reported raw data. The edge device source, and destination, can be manually added to the tabular related reports. Pick the inventory columns on the rendered grid.
- The SD-WAN tunnel metrics (latency, packet loss, and jitter) do not support minimum, maximum, or baselines variants. Support for percentile metrics is available, but the support is not provided out-of-the-box and must be configured. For more information, see Edit a Metric.
- The Data Aggregator Administration UI lets you configure projection metrics for the SD-WAN tunnel metrics (latency, packet loss, and jitter). However, these metrics do not have baselines. Therefore, projection metrics are unsupported for the SD-WAN tunnel metrics. For more information, see Edit a Metric.
- No direct reconciliation of the SD-WAN edge devices to SNMP physical devices occurs. Therefore, reporting for the CPU and Memory utilization of edge devices is done from the virtual host metric family.
- No direct reconciliation of virtual interfaces to SNMP physical interfaces occurs.
- The parented edge device of the tunnel must report the virtual interface of the SD-WAN tunnels. Direct queries by the tunnel item are unsupported.
- SD-WAN tunnels do not have values in the description fields. The description values are always blank.
- The alias setting for the SD-WAN tunnel items is unsupported.
- The VNA Domain Sites are initially shown as numeric values. The site group can be altered with the Group Admin UI. Other alterations are not recommended under the SD-WAN Sites group.
- The SD-WAN Tunnel metric family is unsupported for On-Demand reports.
- If you set a custom time range to 30 minutes on an SD-WAN dashboard, the time bar charts and trend views have data gaps.

SD-WAN Tunnel and Application Paths Dashboard Limitations

In DX NetOps Performance Management releases before 3.6, the site selected in the Map view filtered the Time Bar view. In DX NetOps Performance Management release after 3.6, the selected site in the Map view does *not* filter this view. However, you can use the search bar to filter the view by site name. For more information, see Monitor SD-WAN.

Search for NFA Drill-Down Pages

- Search must be on name or description fields.
- Protocol search does not work on the group protocols(ip,ipv6,tcp,udp) as they are not single protocols.
- When searching the Hosts, you must provide the host name without braces. For example, if the host name is (10.10.1.2), the host name you must search will be 10.10.1.2.
- When searching a conversation, you must use the Server IP or the Client IP. For example, when searching for the conversation "10.19.19.179 - 10.20.20.179", the search value must be 10.19.19.179 or 10.20.20.179. "10.19.19.179 - 10.20.20.179" is not allowed.
- For ToS search, search without braces '()' of ToS description. For example, if the ToS description is ToS 20 (ToS 20), the search value must be ToS20. For a ToS format like AF12 (DSCP12) ECT=0;CE=1 (ToS 49), the search value must be AF12 (DSCP12) ECT=0.
- Search does not work on 'other' as it is group of items.

Theme Deployment Hangs

Theme deployment occasionally hangs when you use the Chrome browser. If this known limitation occurs, deploy the theme using Firefox or Internet Explorer as your browser.

Time Zone Limitation in CA Business Intelligence Reports

When you change a time zone for a CA Business Intelligence report, you can select only a time zone with a different offset. If the time zone is **(UTC-05:00) Bogota, Lima, Quito**, the system does not register a selection with the same

offset, such as **(UTC-05:00) Eastern Time**. To work around this issue, select a time zone with a different offset before selecting the desired time zone.

Top Enterprise Flow Views

If you have CA Network Flow Analysis 10.0 or higher, the links in the Top Enterprise Hosts by Volume view and the Top Enterprise Protocols by Volume view do not work. If you have view suppression enabled, these views do not appear in the relevant dashboards (for example, the Infrastructure Overview dashboard) or context pages. You can find useful information in the new Network Interface Performance dashboard. These views perform as expected with CA Network Flow Analysis 9.5 and lower.

Use Defaults Button Limitations

A view cannot be converted to report at a group level under the following circumstance:

- A context item filter is applied when adding on-demand reports or dynamic trend views to a page within Dashboard Builder.

Avoid reverting the view settings with the Use Defaults button at the All Tenant Users level under the following circumstance:

- You are editing a view that is locked to a device or interface that no longer exists in the system.

If the Use Defaults button is applied, do the following tasks:

- Edit the view twice to restore the metric selection.
- Add a new device or interface to the view before rendering data.

Viptela Inventory Includes Only Devices with Valid Certificates

The Viptela plug-in accurately interprets vEdge devices with "valid" certificate states. The Viptela plug-in cannot accurately interpret vEdge devices with "invalid" certificate states. If the certificate of a vEdge device is invalidated, the vEdge device is no longer discovered in the inventory. Associated interfaces are no longer reported, however tunnels are still reported and reference the interface underlay.

A vEdge device must have a valid security certificate to participate in a Viptela network. You can configure the certificate state from the vManage Certificates administration page.

To administer a certificate for a vEdge device:

1. Log in to vManage.
2. Select **Certificates** from the **Configuration** sidebar drop-down.
3. Select the desired administrative state for the device certificate in the **Validate** column (Invalid, Staging, Valid).

For more information, see the Viptela product documentation. Viptela online documentation is available by clicking the question mark (?) icon in the top menu bar in vManage.

Viptela Scale Boundary

Viptela has an upper boundary for the number of managed items. For example, depending on the size of your physical system, more than 1000 vEdge devices might cause issues.

The boundary depends on the following factors:

- Network bandwidth between DX NetOps Virtual Network Assurance and vManage
- Number of vEdge devices
- Tunnel topology (for example, hub-and-spoke versus full mesh)
- Number of interfaces
- Number of SLA classes/policies

If the upper boundary is hit, you might notice the following issues:

- 24 hours after a fresh install, the inventory is incomplete.
- Managed items are missing.
- Performance data is missing.
- The performance cycle does not complete within the allotted time (10 minutes for inventory, 30 minutes for performance).

If you suspect an issue, go to the log in the following directory and look for any warnings or errors indicating a failure to receive responses:

VNA_install_directory/VNA/standalone/log

Virtual and SNMP Interface Reconciliation

Virtual interfaces from DX NetOps Virtual Network Assurance are now reconciled with existing SNMP interfaces. When the VNA Gateway restarts (for example, during upgrade), the virtual interfaces are reconciled with existing SNMP interfaces, and the virtual interface is deactivated. Currently, the performance data on the virtual interface is not migrated over to the SNMP interface. All new performance data goes to the SNMP interface. The performance data on the virtual interface is not easily accessible.

Interface Utilization

For interface utilization, Cisco recommends looking at the higher of utilization in and utilization out. DX NetOps Performance Management uses this method. The advantage of this method is that if either utilization in or out is high, utilization shows the problem.

The legacy monitoring solution CA eHealth shows the average of the two values. The advantage of this method is that high utilization indicates a problem with utilization in *and* utilization out. The disadvantage is that if either utilization in or utilization out is high, the utilization metric alone does not reflect the severity of the problem. As a result, the DX NetOps Performance Management method might show higher overall utilization values than CA eHealth.

To apply the CA eHealth method to the utilization metric in DX NetOps Performance Management, apply the extension to the relevant vendor certification:

ifXUtilizationTableMIB

```
<DataModel namespace="http://im.ca.com/certifications/snmp" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance"
 xsi:noNamespaceSchemaLocation="SNMPCertificationFacet.xsd">
    <Author>CA</Author>
    <Version>2.1</Version>
    <FacetType name="IfXTableMib"
 descriptorClass="com.ca.im.core.datamodel.certs.CertificationFacetDescriptorImpl">
       <FacetOf namespace="http://im.ca.com/core" name="Item" />
       <Expressions>
```

```
            <ExpressionGroup destCert="{http://im.ca.com/normalizer}NormalizedPortInfo"
 name="PortNRMDS">
                <Expression
 destAttr="Utilization">CalculatedUtilization=(CalculatedSpeedIn
 > 20000000 &amp;amp;&amp;amp; isdef(CalculatedIfHCInOctets)) ?
 (isdef(CalculatedDuplexStatus) ? (CalculatedDuplexStatus==3 ?
(snmpProtectedDiv(((CalculatedIfHCInOctets+CalculatedIfHCOutOctets)*8),
 ((CalculatedSpeedIn
+CalculatedSpeedOut)*_rspDuration))*100):snmpProtectedDiv(((CalculatedIfHCInOctets
+CalculatedIfHCOutOctets)*8),
 (CalculatedSpeedIn*_rspDuration))*100):snmpProtectedDiv(((CalculatedIfHCInOctets
+CalculatedIfHCOutOctets)*8), (CalculatedSpeedIn*_rspDuration))*100) :
 (isdef(CalculatedDuplexStatus) ? (CalculatedDuplexStatus==3?
snmpProtectedDiv(((CalculatedIfInOctets+CalculatedIfOutOctets)*8), ((CalculatedSpeedIn
+CalculatedSpeedOut)*_rspDuration))*100:(snmpProtectedDiv(((CalculatedIfInOctets
+CalculatedIfOutOctets)*8),(CalculatedSpeedIn*_rspDuration)))*100) :
(snmpProtectedDiv(((CalculatedIfInOctets+CalculatedIfOutOctets)*8),
(CalculatedSpeedIn*_rspDuration)))*100);
                    CalculatedUtilization &gt; UtilizationMaxPercent ?
                        null : CalculatedUtilization</Expression>
            </ExpressionGroup>
        </Expressions>
    </FacetType>
</DataModel>
```

JuniperifXUtilizationTableMIB

```
<DataModel namespace="http://im.ca.com/certifications/snmp" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance"
 xsi:noNamespaceSchemaLocation="SNMPCertificationFacet.xsd">
    <Author>CA</Author>
    <Version>2.1</Version>
    <FacetType name="IfXTableMib"
 descriptorClass="com.ca.im.core.datamodel.certs.CertificationFacetDescriptorImpl">
        <FacetOf namespace="http://im.ca.com/core" name="Item" />
        <Expressions>
            <ExpressionGroup destCert="{http://im.ca.com/normalizer}NormalizedPortInfo"
 name="PortNRMDS">
                <Expression
 destAttr="Utilization">CalculatedUtilization=(CalculatedSpeedIn &gt; 20000000
                    &amp;&amp; isdef(CalculatedIfHCInOctets))
                    ?(isdef(dot3StatsDuplexStatus)
                        ? (dot3StatsDuplexStatus==3
                            ?(snmpProtectedDiv((CalculatedIfHCInOctets
+CalculatedIfHCOutOctets)*8,
```

```
 (CalculatedSpeedIn*_rspDuration))*100):snmpProtectedDiv(((CalculatedIfHCInOctets
+CalculatedIfHCOutOctets)*8),

 (CalculatedSpeedIn*_rspDuration))*100):snmpProtectedDiv(((CalculatedIfHCInOctets
+CalculatedIfHCOutOctets)*8),
                            (CalculatedSpeedIn*_rspDuration))*100)
                    :
                    (isdef(dot3StatsDuplexStatus)
                        ?(dot3StatsDuplexStatus==3
                            ?snmpProtectedDiv((CalculatedIfInOctets
+CalculatedIfOutOctets)*8,
                            (CalculatedSpeedIn*_rspDuration))*100:
(snmpProtectedDiv(((CalculatedIfInOctets+CalculatedIfOutOctets)*8),
(CalculatedSpeedIn*_rspDuration)))*100)
                            :(snmpProtectedDiv(((CalculatedIfInOctets
+CalculatedIfOutOctets)*8),(CalculatedSpeedIn*_rspDuration)))*100);
                    CalculatedUtilization &gt; UtilizationMaxPercent ?
                        null : CalculatedUtilization</Expression>
            </ExpressionGroup>
        </Expressions>
    </FacetType>
</DataModel>
```

Hide Gaps in Trend Views

By default, DX NetOps Performance Management shows gaps in all trend type charts when no data exists for that data point. Gaps can occur when DX NetOps Performance Management does not have data for the item for a timeframe. The following circumstances cause gaps in the data:

- The device does not respond to a poll request.
- The device is down.
- A network problem causes the poll response to be lost.
- The Data Collector is down.
- For as polled data, a counter rollover or bad poll occurred.
  For more information about counters, see Configure Counter Behavior.

DX NetOps Performance Management includes a mechanism to hide gaps. The chart draws a straight line from the last data point to the next data point.

WARNING

When gaps are hidden, trend charts have no indication that a gap exists in the data. The connected line *does not* reflect the real behavior for the device during the period with missing data. Gaps are visible in exported data.

Follow these steps:

1. Log in to the Data Aggregator host.
2. Create the following file:
   `apache-karaf-2.4.3/etc/com.ca.im.dm.ribsource.RIBSourceWsImpl.cfg`
3. Add the following line to that file:
   `gapShowingOn=false`

All trend type views hide gaps. To apply the new behavior, refresh the view.

Metric Family Changes

DX NetOps Performance Management introduces the following changes to metrics and metric families:

Metric Name Changes

Change occurred in release 3.5

Old Metric Display Name	New Metric Display Name
Mobile WLAN Access Point	Wireless Access Point
Mobile Stations	Wireless Stations

Change occurred in release 2.6.

To ensure ease of use, long metric display names have been shortened, as follows:

Old Metric Display Name	New Metric Display Name
Fail Fast Chan Change Requests Rcvd from Std Def Channels	Std Def Channel Change Fail
Ttl Aggregate BW of The Currently Running Egress Streams	Running Egress Aggregate BW
Number of Currently Active Connections Getting Optimized	Active Connections Optimizing
IPv6 Prefixes from Peer Which Were Suppressed By Damping	Peer IPv6 Prefix Damping Suppressed
VPN IPv6 Prefixes from Peer Which Were Suppressed By Damping	Peer VPN IPv6 Prefix Damping Suppressed
Next Sequence No Identifier to Be Sent In A Linktrace Msg	Next Sequence No ID LinkTrace
Next Sequence No Identifier to Be Sent In A Loopback Msg	Next Sequence No ID LoopBack
Overall CPU busy percentage in last cpmCPUMonInterval period	CPU Busy Last cpmCPUMonInterval
The overall CPU busy percentage in the last 5 second period	CPU Busy Last 5 Second
Max outstanding snmp requests reached in last poll cycle	Max Outstanding SNMP Requests
CPU Capacity Priority Weight Units Currently Unallocated	CPU prority weight unallocated
Average One Way Delay Destination to Source with Precision	Avg 1-way delay dest to src w/precision
Average One Way Delay Source to Destination with Precision	Avg 1-way delay src to dest w/precision
Maximum One Way Delay Destination to Source with Precision	Max 1-way delay dest to src w/precision
Maximum One Way Delay Source to Destination with Precision	Max 1-way delay src to dest w/precision
Minimum One Way Delay Destination to Source with Precision	Min 1-way delay dest to src w/precision
Minimum One Way Delay Source to Destination with Precision	Min 1-way delay src to dest w/precision
Percent of Time in Off State Critical Central Office Relay	Pct Time Off State Critical
Percent of Time in On State Critical Central Office Relay	Pct Time On State Critical
Percent of Time in Unknown State - Chassis Audible Alarm	Pct Time Unknown Chassis Alarm
Percent of Time in Unknown State Major Central office Relay	Pct Time Unknown State Major
Percent of Time in Unknown State Minor Central office Relay	Pct Time Unknown State Minor
Mobile IP Registration Denied PDSN - Missing Home Address	Mobile IP Reg Denied Missing Home Address
Number of Registration Requests Discarded - Mobile IP HA	Registration Requests Discarded - Mobile IP HA

Number of Registration Requests - FA Authentication Failure	Registration Requests FA Auth Failure
Number of Registration Requests - Authentication Failure	Registration Requests Auth Failure
Number of L3 Egress Classifications Matching a Drop Policy	L3 Egress Class Drop Policy Count
Number of L3 Ingress Classifications Matching a Drop Policy	L3 Ingress Class Drop Policy Count
The total number of Double Data Rate channel sync errors	DDR Channel Sync Errors
Num of Active Conn Going Through TCP Plus Other Optimization	Active TCP Connections Plus Optimization
Connections Declined due to Non-availability of Resources	Connections Declined - Non Availability
URL Requests Resource Dropped per second (last 5 minutes)	URL Resource Dropped per second (last 5 min)
Number of Instances Received Packet Matches The Filter Rule	Instances Rcvd Packet Matching Filter Rule
Reserved Memory Used for Running VM in this Resource Pool	VM Resource Pool Reserved Memory
Count Successful location report between 50 and 150 meters	Successful Location Report 50-150m
Count SUPL Transactions Using LLPW with MSA And OTDO Method	SUPL Transactions Using LLPW w/MSA &OTDO
Lower Threshold on Total Number of Instances of Running Process	Running Process Instances Lower Threshold
Upper Threshold on Total Number of Instances of Running Process	Running Process Instances Upper Threshold
Number of Poll Groups Stopped Due to Prior Timeouts (internal)	Poll Groups Stopped Due to Prior Timeouts (internal)
Pct Tx Frames Arrived with Delay Variation Greater Than Thresh	Pct Tx Delay Variance Over Threshold
Total Number of Cache Misses Bytes Excluding Uncacheable Data	Cache Misses Bytes Excluding Uncacheable Data
Number of Frame Relay Link Q921 Call Control Subsystem Errors	Q921 Call control subsystem errors
Mobile IP Registration Requests Denied PDSN - Authentication Failure	Mobile IP Reg Req Denied PDSN Auth Failure
Number of Handoff Registration Requests Accepted - Mobile IP HA	Mobile IP HA Handoff Reg Requests Accepted
Number of Handoff Registration Requests Denied - Mobile IP HA	Mobile IP HA Handoff Reg Requests Denied
Number of Sessions Released which Failed the IP Control Protocol	Sessions Released Failed IP Control
The Total Number of Cells Input Packet and Cells Output Packet	Total Cells I/O Packets
Limit Buffer To High Priority Traffic When List 1 Is Below Threshold	Buffer High Priority List 1 Below Thresh
Limit Buffer To High Priority Traffic When List 2 Is Below Threshold	Buffer High Priority List 2 Below Thresh
Limit Buffer To Low Priority Traffic When List 1 Is Below Threshold	Buffer Low Priority List 1 Below Thresh
Limit Buffer To Low Priority Traffic When List 2 Is Below Threshold	Buffer Low Priority List 2 Below Thresh
Limit Buffer To Medium Priority Traffic When List 1 Is Below Threshold	Buffer Med Priority List 1 Below Thresh
Limit Buffer To Medium Priority Traffic When List 2 Is Below Threshold	Buffer Med Priority List 2 Below Thresh
Limit Buffer To Urgent Priority Traffic When List 1 Is Below Threshold	Buffer Urgent Priority List 1 Below Thresh
Limit Buffer To Urgent Priority Traffic When List 2 Is Below Threshold	Buffer Urgent Priority List 2 Below Thresh

Interface Metric Family

Change occurred in release 2.5.

The Interface Metric Family now includes new metrics:

- Bits Per Second
- Bits Per Second In
- Bits Per Second Out

The following metrics have been disabled in this metric family:

- Bytes
- Bytes In
- Bytes Out

This change disrupts existing thresholds or views that use Bytes, Bytes In, or Bytes Out.

NOTE

the Starent device pack for CA Mediation Manager uses the Bytes metrics. If you use this device pack, you may want to enable Bytes metrics for this vendor certification. The XML to extend this vendor certification is located in the following directory:

```
/opt/IMDataAggregator/examples/vendorCertification/
```

Enable Bytes Metrics

WARNING

Enabling Bytes metrics for the interface metric family may affect system performance.

To enable the Bytes, Bytes In, and Bytes Out metrics for individual vendor certifications, extend the certification. The XML to enable the metric is included in the following directory:

```
/opt/IMDataAggregator/examples/vendorCertification/
```

Merge the example XML with your extension XML. For information about how to extend the vendor certification, see Create or Extend Vendor Certifications.

Mobile WLAN Access Point (certified since 2.3) – renamed into **Wireless Access Point**

OpenAPI Changes

DX NetOps Performance Management introduces the following changes to the OpenAPI in release 2.8:

Removed Entities

The `pollgroup` entity was removed, which included the following attributes:

- `NormalizedFacetType`
- `CertificationFacetType`
- `PollingInterval`

Removed Relationships

The following table lists the relationships that were removed:

`interface`	`qosclassmap`
`qosclassmap`	`qospolicer`

qosclassmap	qoscontract
qosclassmap	qosqueuing
qosclassmap	qosred
qosclassmap	qostrafficshaping

Impact:

The QoS elements hierarchy is no longer supported.

Examples:

```
odata/api/interfaces?&$expand=qosclassmaps
odata/api/qosclassmaps?&$expand= qosreds
```

Updated Entities

Component

The following attributes were removed for the `component` entity type:

- `IsAlso N/A`
- `MDRItemID`
- `IsFiltered`
- `IndexList`
- `SourceFacetTypes`
- `IsExcludedFromReports`

The `IsAlso` attribute was removed.

Impact:

No information is available on the entity type when querying for devices and components.

Examples:

```
odata/api/devices?$top=10&$filter=substringof('router', IsAlso) eq true and
 substringof('switch', IsAlso) eq true

odata/api/components?$top=10&$filter=substringof('cpu', IsAlso) eq true or
 substringof('memory', IsAlso) eq true
```

Cpu

The following attributes were removed for the `cpu` entity type:

- `MDRItemID`
- `IsFiltered`
- `IndexList`
- `SourceFacetTypes`
- `IsExcludedFromReports`

Memory

The following attributes were removed for the `memory` entity type:

- `MDRItemID`
- `IsFiltered`
- `IndexList`
- `SourceFacetTypes`
- `IsExcludedFromReports`
- `CertificationFacetTypes`
- `EMSEngineId`
- `DiscoverTime`
- `EMSSourceFacetType`

Interface

The following attributes were removed for the `interface` entity type:

- `MDRItemID`
- `IsFiltered`
- `IndexList`
- `SourceFacetTypes`
- `IsExcludedFromReports`
- `ItemSubType`
- `ItemType`
- `SpeedOutOverride`
- `SpeedInOverride`

Monitoringprofiles

The following attributes were removed for the `monitoringprofiles` entity type:

- `PollGroupIDs`

Mplsinterface

The following attributes were removed for the `mplsinterface` entity type:

- `MDRItemID`
- `IsFiltered`
- `IndexList`
- `SourceFacetTypes`
- `IsExcludedFromReports`
- `CertificationFacetTypes`
- `EMSEngineId`
- `DiscoverTime`
- `EMSSourceFacetType`

Mplsinsegment

The following attributes were removed for the `mplsinsegment` entity type:

- MDRItemID
- IsFiltered
- IndexList
- SourceFacetTypes
- IsExcludedFromReports
- CertificationFacetTypes
- EMSEngineId
- DiscoverTime
- EMSSourceFacetType

Mplsoutsegment

The following attributes were removed for the `mplsoutsegment` entity type:

- MDRItemID
- IsFiltered
- IndexList
- SourceFacetTypes
- IsExcludedFromReports
- CertificationFacetTypes
- EMSEngineId
- DiscoverTime
- EMSSourceFacetType

Qosclassmap

The following attributes were removed for the `qosclassmap` entity type:

- MDRItemID
- IsFiltered
- IndexList
- SourceFacetTypes
- IsExcludedFromReports
- ParentItemID
- ChildItemIDs

Qospolicer

The following attributes were removed for the `qospolicer` entity type:

- MDRItemID
- IsFiltered
- IndexList
- SourceFacetTypes
- IsExcludedFromReports
- ParentItemID
- ChildItemIDs

Qoscontract

The following attributes were removed for the `qoscontract` entity type:

- MDRItemID
- IsFiltered
- IndexList
- SourceFacetTypes
- IsExcludedFromReports
- ParentItemID
- ChildItemIDs

Qosqueuing

The following attributes were removed for the `qosqueuing` entity type:

- MDRItemID
- IsFiltered
- IndexList
- SourceFacetTypes
- IsExcludedFromReports
- ParentItemID
- ChildItemIDs

Qosred

The following attributes were removed for the `qosred` entity type:

- MDRItemID
- IsFiltered
- IndexList
- SourceFacetTypes
- IsExcludedFromReports
- ParentItemID
- ChildItemIDs

Qostrafficshaping

The following attributes were removed for the `qostrafficshaping` entity type:

- MDRItemID
- IsFiltered
- IndexList
- SourceFacetTypes
- IsExcludedFromReports
- ParentItemID
- ChildItemIDs

Device

The following attributes were removed for the `device` entity type:

- MDRItemID
- ItemSubType
- ItemType
- ConsolidatedMonitoringProfile
- SupportsOnDemandMFDiscovery

The IsAlso attribute was removed.

Impact:

No information is available on the entity type when querying for devices and components.

Examples:

```
odata/api/devices?$top=10&$filter=substringof('router', IsAlso) eq true and
 substringof('switch', IsAlso) eq true

odata/api/components?$top=10&$filter=substringof('cpu', IsAlso) eq true or
 substringof('memory', IsAlso) eq true
```

The LifeCycleState column was added.

Router

The following attributes were removed for the router entity type:

- MDRItemID
- ItemSubType
- ItemType
- ConsolidatedMonitoringProfile
- SupportsOnDemandMFDiscovery
- MACAddresses

The State attribute was renamed LifeCycleState.

Switch

The following attributes were removed for the switch entity type:

- MDRItemID
- ItemSubType
- ItemType
- ConsolidatedMonitoringProfile
- SupportsOnDemandMFDiscovery
- MACAddresses

The State attribute was renamed LifeCycleState.

Server

The following attributes were removed for the server entity type:

- `MDRItemID`
- `ItemSubType`
- `ItemType`
- `ConsolidatedMonitoringProfile`
- `SupportsOnDemandMFDiscovery`
- `MACAddresses`
- `FirstMDRItemID`
- `StatusEvaluationType`
- `ChangeDetectionDisabledMFs`
- `CalculatedContactStatus`

The `State` attribute was renamed `LifeCycleState`.

Data Source Compatibility

NetOps Portal can use the following CA Technologies products as registered data sources:

- DX NetOps Virtual Network Assurance
- DX NetOps Spectrum
- Network Flow Analysis and Anomaly Detector
- CA Application Delivery Analysis
- CA Unified Communications Monitor
- CA Application Performance Management
- CA Business Intelligence

For supported releases, see the DX NetOps Interoperability.

Some components function as data sources. For these components, use only the release that is delivered as part of DX NetOps Performance Management:

- Data Aggregator
- Event Manager

For more information, see Manage Data Sources.

Language Support

DX NetOps Performance Management supports the following locales:

- English (US)
- French (France)
- Japanese

> **NOTE**
> DX NetOps Performance Management 3.7 and earlier support Simplified Chinese and Traditional Chinese. These languages have been deprecated in this release.
> For more information:
>
> - About this feature deprecation, see Deprecated Features.
> - About how to customize the language for your user account, see Customize Your User Settings.

The Help icons in the Simplified Chinese UI and Traditional Chinese UI link to the English documentation.

Additional languages might be supported in future releases. The following items are not translated:

Untranslated Items in DX NetOps Performance Management

The following items are not localized in DX NetOps Performance Management:

Component and Device Type Names

Component types are not localized, and the following device type names are not localized when you view them in report dashboards, groups, and inventory views:

- **Device types**
  - Other Devices
  - Pingable Devices
  - Call Server

The Device Components label and description in the Group Rule dialog are not localized.

Custom Context Types from Data Sources

Custom context types synchronized by data sources, such as the context types display in the Dashboard Editor, are not localized.

Data Collector Installer Download Page

The Data Collector installer download page (`http://data_aggregator:port/dcm/install.htm`) is not localized. For more information about how to download the Data Collector installer without accessing the webpage, see Installing.

Data Collector Names

The names of data collectors are not localized.

Data Source Localization Limitations

The names, descriptions, and other aspects of the predefined monitoring profiles and threshold event profiles that you can view in DX NetOps Performance Management administration are not localized.

Network Flow Analysis is only translated into French, Simplified Chinese, and Japanese. The Network Flow Analysis bookshelf displays in English if the user language preference is set to Traditional Chinese.

Event information from CA Application Delivery Analysis, Network Flow Analysis, and CA Unified Communications Monitor data sources, such as event descriptions and event types, is not localized.

DX NetOps Performance Management supports integration with CA eHealth. However, integration with NetOps Portal is provided only in English versions of CA eHealth. The Japanese and French versions of CA eHealth do not support the NetOps Portal integration.

Custom Item Types from Data Sources

Some data sources support unique managed item types. For example, the DX NetOps Performance Management Data Aggregator data source supports a "Device Component" managed item type.

When data sources synchronize managed items with unique item types, the items appear in the Inventory, but their names and types are not localized.

Data Source Role Rights

Some of the supported data sources have their own sets of role rights to let selected users access product features. In some cases, the data source user interface is not localized into all of the supported languages. Role rights that are synchronized from such data sources that appear in the Edit Role Rights dialog are not localized.

Additional data sources like CA Application Delivery Analysis and CA Unified Communications Monitor also have limitations.

Role rights that are synchronized from CA Application Delivery Analysis, CA Unified Communications Monitor, and Network Flow Analysis are not localized.

Direction Terms

The Direction terms "In" and "Out" are not localized in the Interfaces Over Threshold view.

DNS Names

DNS names are not localized.

English String in NetOps Portal Installer

When you run the installer on a server or in command line with a locale set to a language other than English, the string "DEFAULT:" appears in English. This string is not localized.

How To Videos

Videos that supplement the documentation are not translated.

Installation Scripts

The following Data Repository scripts are not localized: `dr_validate.sh` and `dr_install.sh` .

Limitations on Custom Strings

You cannot provide multiple translations for strings that you customize, such as the following strings:

- Group Name
- Tenant Name
- Domain Name
- Role Name
- View Title
- Monitoring Profile Name
- Threshold Event Profile Name
- Discovery Profile Name

Monitoring Profiles

The names and descriptions of the product default monitoring profiles are not localized.

MIB Compiler Errors

MIB compiler errors are not localized.

Overview Tab in NetOps Portal

CA Unified Communications Monitor uses some tabs on the top-level dashboards. The Overview tab is not localized. The workaround is to manually create a new dashboard that contains the CA Unified Communications Monitor views.

Theme Names

Theme names are not localized.

XML Tag Names

The XML tag names for vendor certification and metric family files are not localized.

Third-Party Information

The following third-party information is not localized:

License Agreements for Third-Party Products

The license agreements for third-party products are not localized.

Third-Party Scripts

Any third-party scripts included in the product, such as the capabilities of a script, are not localized.

Known Issues

The following localization issues currently exist in DX NetOps Performance Management:

Date Format Incorrect on Heat Charts

The date format tooltip on heat charts in Traditional Chinese is incorrect. The correct date format is yyyy/mm/dd.

Date Format Incorrect on Performance Tab

The date format is incorrect in Asian languages on the Performance tab of the Device context pages.

Third Party Agreements

DX NetOps Performance Management

All third-party software has been used in accordance with the terms and conditions for use, reproduction, and distribution as defined by the applicable license agreements. The following file contains the license agreements: TPSAs.

The file contains the following license agreements. Where applicable, the license agreements are also available in the following directories:

- DA/DC Installation
  `<Root installation folder>\NOTICE.txt`
- PC Installation
  `<Root NetOps Portal folder>\NOTICE.txt`

- Ace Editor 1.2.0
- Activation-api-1.1 2.5.0
- ActiveMQ 5.15.8
- Adobe Flex SDK 3.6
- AdobeOpen JDK 1.8_222-b10
- AntiSamy 1.5.3
- Antlr 4.5.1
- Apache Aries Blueprint - API 1.0.1
- Apache Aries - Blueprint - CM 1.0.6
- Apache Aries - Blueprint - Core 1.4.3
- Apache Aries - Proxy API 1.0.1
- Apache Aries - Proxy Implementation 1.0.4
- Apache Aries - Utils 1.1.0
- Apache Commons CSV 1.4
- Apache Commons FileUpload 1.3.1
- Apache CXF 2.3.11
- Apache CXF 2.4.10
- Apache CXF 2.7.11
- Apache FOP 2.2
- Apache ServiceMix Bundles Spring Framework 3.2.11 Release 1
- Apache SSHD 0.4.0
- Apache Tomcat JDBC Connection Pool 7.0.32
- ASM 3.3.1
- ASM 5.0.3
- Avalon Framework 4.1.3
- Batik 1.7
- Batik 1.8
- Beanshell 2.0b6
- Bnd 2.4.0
- Bootstrap 3.2.0
- Bootstrap 4.0.0
- Bouncy Castle Java FIPS 1.0.0
- Cglib 2.2.2
- Commons beanutils 1.8.3
- Commons beanutils 1.9.2
- Commons beanutils 1.9.3
- Commons Cli 1.4
- Commons Codec 1.4
- Commons Codec 1.7
- Commons Codec 1.9
- Commons Collections 3.2.2
- Commons Collections 4.1
- Commons Configuration 1.6
- Commons-DBCP 1.4
- Commons-DBCP 2.1.1
- Commons Digester 1.8.1
- Commons Digester 2.0
- Commons FileUpload 1.4
- Commons IO 2.0.1
- Commons IO 2.4
- Commons JXPath 1.3
- Commons Lang 2.5
- Commons Lang3 3.4
- Commons Lang3 3.5

OI Connector for DX NetOps Performance Management

All third-party software has been used in accordance with the terms and conditions for use, reproduction, and distribution as defined by the applicable license agreements. The following file contains the license agreements: OI_Connector_TPSAs.

The file contains the following license agreements:

- Antisamy 1.5.3
- Apache Commons CSV 1.4
- Apache Commons FileUpload 1.3.1
- Apache CXF 2.4.10
- ASM 3.3.1
- Batik 1.8
- Beanshell 2.0b6
- Bouncy Castle Java FIPS 1.0.0
- Commons beanutils 1.9.2
- Commons Codec 1.9
- Commons Collections 3.2.2
- Commons dbcp 1.4
- Commons dbcp 2.1.1
- Commons Digester 1.8.1
- Commons IO 2.0.1
- Commons JXPath 1.3
- Commons Lang 2.5
- Commons Logging 1.0.4
- Commons Net 3.5
- Commons Validator 1.5.1
- ESAPI 2.1.0.1
- Geronimo Servlet 3.0 Spec 1.0
- Geronimo-javamail 1.4 spec 1.7.1
- Guava 21.0
- H2 1.4.193
- HttpClient 4.5.3
- HttpCore 4.4.6
- Jackson 2.9.4
- Java Deep-Cloning Library 1.7.4
- Javax Javaee API 7.0
- javax servlet api 3.1.0
- JBoss RESTEasy 3.1.3
- JBoss LogManager 2.0.4 Final
- Jettison 1.3.2
- Jetty 9.3.20.v20170531
- Jetty 9.3.8.v20160314
- JSON 20170516
- jsr311-api 1.1.1
- Java Service Wrapper (JSW) 3.5.27
- Log4j 1.2.17
- Log4j-jboss-logmanager 1.1.2 Final
- myfaces 1.1.4
- neethi 3.0.2
- nekohtml 1.9.16
- Objenesis 1.2
- openpojo 0.8.4
- Oracle Java Runtime Environment (JRE) 1.8.0_192
- OWASP Java Encoder 1.2.1
- slf4j 1.7.21
- Spring Framework 4.3.2
- Stax2-api 3.1.4
- Tomahawk 1.1.5
- WildFly Swarm 2017.7.0
- Woodstox 5.0.3

Deprecated Features

DX NetOps Performance Management 3.7 and earlier support Simplified Chinese and Traditional Chinese. These languages have been deprecated in this release. These language options are not available from the User Settings menu.

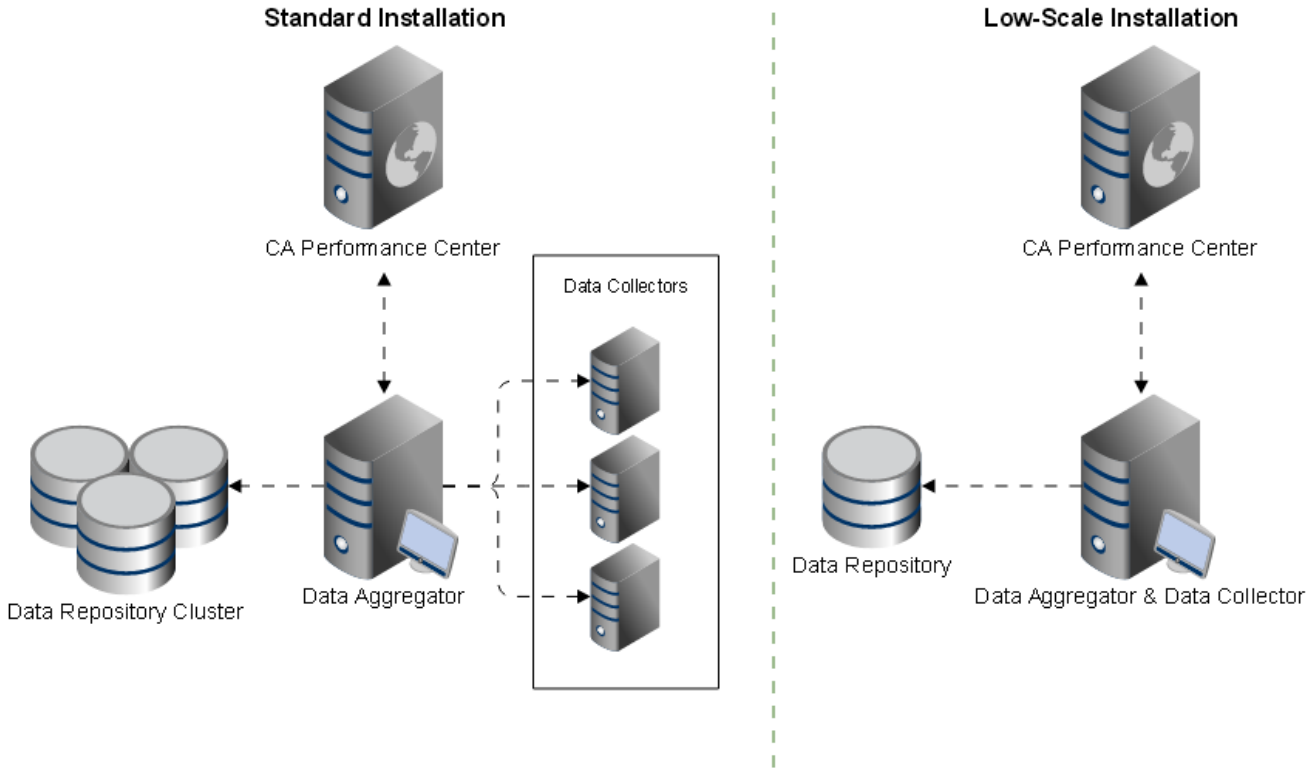
Installing

DX NetOps Performance Management is a distributed application that includes multiple components across several servers. A successful deployment includes installing these components in the following order:

1. NetOps Portal
2. Data Repository
3. Data Aggregator
4. Data Collectors

Your deployment strategy depends on the number of devices, location of these devices, and which metrics you want to monitor. The following diagram shows the installation options:

Figure 3: CA Performance Management Component Installation Options



Install DX NetOps Performance Management

NOTE

This process describes a standard installation. For low-scale systems, see Install or Uninstall the Proxy Server.

DX NetOps Performance Management requires the installation of linked components. To install the product, use the following installation order:

1. Review Installation Requirements and Considerations
2. Prepare to Install Performance Center
3. Install Performance Center
4. Prepare to Install the Data Repository
5. Install the Data Repository
6. Prepare to Install the Data Aggregator
7. Install the Data Aggregator
8. Prepare to Install the Data Collectors
9. Install the Data Collectors
10. Complete the Post-Installation Configuration
11. (Optional) Install CA Mediation Manager

NOTE

You can configure DX NetOps Performance Management to use FIPS-compliant encryption and hashing algorithms (where applicable). For more information, see FIPS-Compliant Encryption.

Review Installation Requirements and Considerations

Review the following information before you install the product:

Operating System Requirements

DX NetOps Performance Management supports the latest version of Red Hat 7.x unless otherwise specified.

The following operating systems have been verified:

- SUSE Linux Enterprise Server (SLES) 12 SP2
- Oracle Linux (OL) 7.3 (Red Hat compatible kernel only)

The following Red Hat Enterprise Linux (RHEL), SLES, and OL versions were verified:

NOTE
A RHEL installation offers packages and add-ons. DX NetOps Performance Management supports a minimal install environment for RHEL.

Release	Kernel
SLES 12.2	4.4.103
OL 7.3	3.10.0-514
RHEL 7.5	3.10.0-862
RHEL 7.4	3.10.0-693
RHEL 7.3	3.10.0-514
RHEL 6.9	2.6.32-696
RHEL 6.8	2.6.32-642
RHEL 6.7	2.6.32-573
RHEL 6.6	2.6.32-504

NOTE
RHEL 6.8, 6.9, 7.3, or 7.4 are recommended for all DX NetOps Performance Management components.

DX NetOps Performance Management does not support the following RHEL versions:

- RHEL 5.x
- RHEL 6.5 and lower
- RHEL 7.0 - 7.2
- RHEL kernel versions before 2.6.32.504

Root or Sudo User Access

Administrative privileges are required to install the software. Typically, the root users installs the software. In some environments, unrestricted root user access is not available.

If root user access is not available, configure a sudo user with access to a limited set of commands.

If you install the components with a sudo user account, add the 'sudo' prefix to commands that require the same user as the service owner, such as restart commands and SSL set up.

User Interface Access Requirements

Supported Browsers

The following browsers are supported:

- Microsoft Internet Explorer version 11
- Microsoft Edge version 42.x and later
- Google Chrome 70.x and later
- Mozilla Firefox 60.x and later

Other Requirements

- The minimum supported screen resolution is 1280x1024.

Virtual and SAN Environment Requirements

Review the CA Support Statement for Running CA Infrastructure Management Products in Virtualization and SAN Environments document. This document discusses CA policies for installing and operating Infrastructure Management products on virtualized servers or Storage Array Networks (SAN).

> **NOTE**
> You must be logged in as a registered user to view documentation on CA Support Online.

Package Requirements

The installer for each component requires the following packages:

Components	Packages
All (SLES)	• `dialog` • `mcelog` • `zip` • `unzip`
All (RHEL 7.x, OL)	• `dialog` • `mcelog` • `zip` • `unzip` • `chrony`

All (RHEL 6.x)	• dialog • mcelog • zip • unzip • glibc
NetOps Portal (SLES)	• fontconfig • libaio1 • libnuma1 • wget
NetOps Portal (RHEL 6.x)	• fontconfig • libaio • libaio-devel • numactl • wget
NetOps Portal (RHEL 7.x, OL)	• fontconfig • libaio • libaio-devel • numactl-libs • wget
Data Repository (RHEL 6.x, RHEL 7.x, SLES, OL)	• bc • pstack • gstack For RHEL 7.x, the pstack and gstack packages are included in the gdb package.
Data Collectors (RHEL 6.x, RHEL 7.x, SLES, OL)	• at

Common Considerations

- Install each component on a separate system.
- Verify that all your servers meet the minimum requirements and sizing guidelines.

  TIP
  To provide high availability for your data, future scalability, and best end-user experience, deploy your Data Repository as a cluster.

  For information about the sizing requirements, see the DX NetOps Performance Management Sizing Tool.

  NOTE
  If the sizing tool recommends a low-scale deployment, see Install a Low-Scale System.
- If you plan to stand up DX NetOps Performance Management in the cloud, see Review Cloud Sizing Guidelines.
- Time synchronization using NTP is required. Start the NTP daemon on Linux if it is not running. All machines must use the same NTP server.

  NOTE
  Except for anti-virus, system management, and time-synchronization software, do not install third-party software, especially third-party network monitoring software, on the same server as DX NetOps Performance Management components. Third-party software can interfere with the monitoring abilities of the CA system, and could void the warranty.

  If you install third-party software on a CA system, CA Support might ask you to uninstall this software before troubleshooting an issue on the server.

Multi-tenant Deployment Considerations

In a multi-tenant deployment, note the following information:

- The Data Aggregator is shared between tenants. The information for each tenant is secure and other tenants cannot view this information.
- In a standard tenant deployment, each tenant has a dedicated Data Collector. A tenant can have more than one Data Collector. For multiple tenants that reside in the same IP routing space, DX NetOps Performance Management can be configured to use fewer Data Collectors. For more information, see Tenant-Agnostic Data Collectors.
- Where a managed service provider is monitoring devices for multiple tenants, you can install Data Collector at the MSP site.

  NOTE
  This setup requires the Data Collector to gain access through a tenant firewall to poll the devices that are being managed.

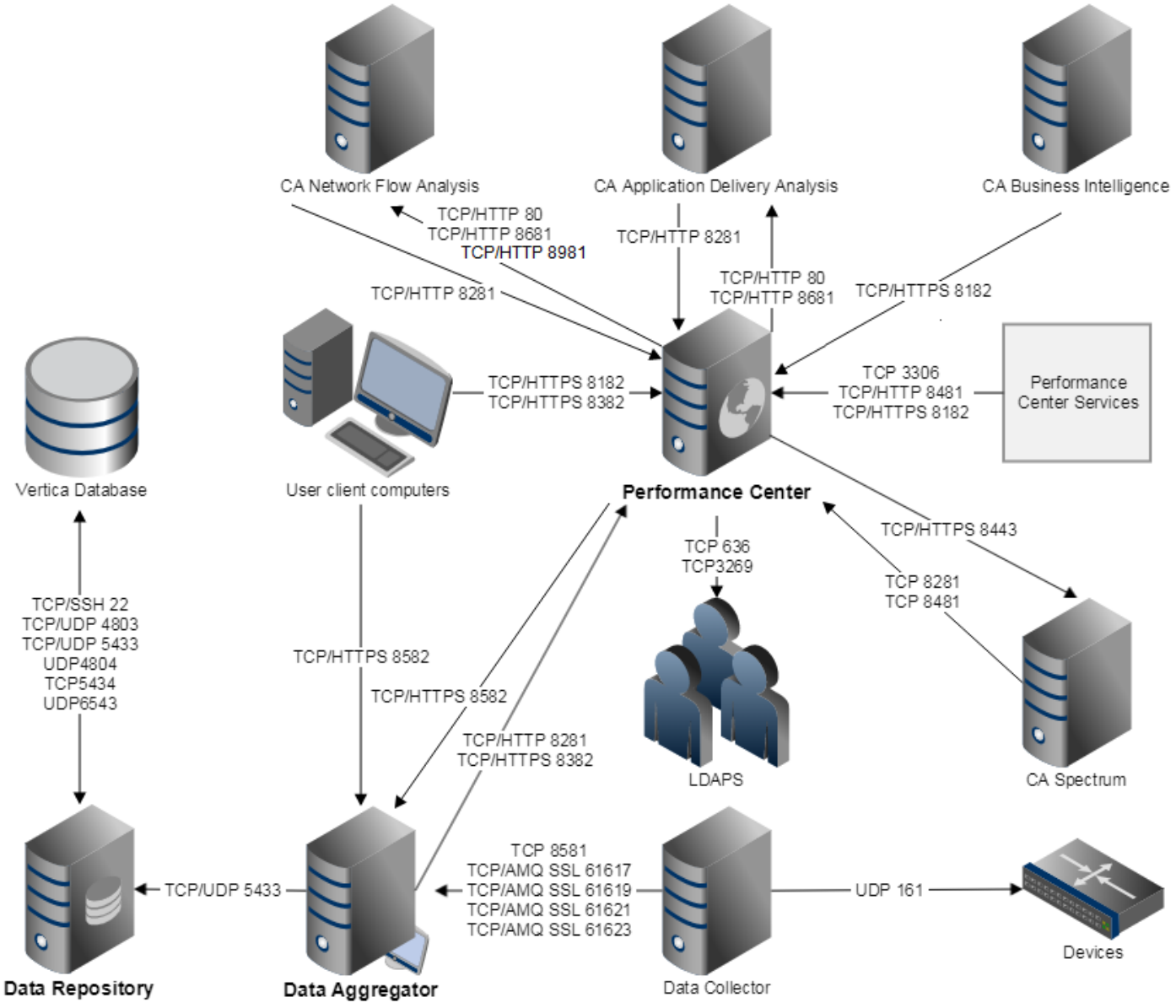
Firewall and Connectivity Considerations

For DX NetOps Performance Management to work correctly in a firewall-protected environment, certain ports must be open.

The following diagram shows the required ports for a hardened environment with a single data aggregator:

NOTE
If you changed the Network Flow Analysis OData API port and you have integrated Network Flow Analysis with NetOps Portal, the port value is synced in the next poll cycle of NetOps Portal or you can perform a manual resync with the Network Flow Analysis data source.

Figure 4: Port_Diagram



Open the following ports to allow DX NetOps Performance Management communications to function properly. For more information, see Prepare to Install Performance Center.

NOTE
Throughout the documentation 8182, 8382, 8582, 61617, 61619, 61621, and 61623 appear as suggested port numbers for secured communications. In the instances where these ports appear, you are free to use any value you want as long as no other processes are using it.

For more information about individual data sources, see the appropriate product documentation.

From	To	Port [Function]
NetOps Portal services	NetOps Portal	• **TCP 3306** Enables communications to the MySQL database (inbound) from the NetOps Portal services. • **TCP/HTTP 8481** Enables communications between the Device Manager and Console services. • **TCP/HTTPS 8182** This port is the default port for NetOps Portal if configured using the HTTPS documentation. For more information, see Configure the Port and Website for HTTPS.
User client computers	NetOps Portal	If you put the application behind a firewall, and you want customers to access only the user interface, open the following ports to the world: • **TCP/HTTP 8181** Enables communications between client computers and the NetOps Portal server. • **TCP/HTTP 8381** Enables communications between client computers and the NetOps Portal server. Also enables login using the single sign-on authentication component. For secured communications, use the following ports instead of 8181 and 8381: • **TCP/HTTPS 8182** This port is the default port for NetOps Portal if configured using the HTTPS documentation. For more information, see Configure the Port and Website for HTTPS. • **TCP/HTTPS 8382** This port is the default port for Single Sign-On if configured using the HTTPS documentation. For more information, see Configure the Port and Website for HTTPS.
User client computers	Data Aggregator	• **TCP/HTTP 8581** Allows for OpenAPI access. Opening this port exposes the rest of the Data Aggregator services. • **TCP/HTTPS 8582** Allows for secure OpenAPI access. Opening this port exposes the Data Aggregator REST services. Open only for clients that require direct access to the Data Aggregator services for administrative and automation purposes.

User client computers	Proxy Server	• **TCP/HTTP 8581** Allows for OpenAPI access in a fault tolerant environment. Opening this port exposes the Data Aggregator REST services. Open only for clients that require direct access to the Data Aggregator services for administrative and automation purposes. • **TCP/HTTP 8581** Allows for secure OpenAPI access in a fault tolerant environment. Opening this port exposes the Data Aggregator REST services. Open only for clients that require direct access to the Data Aggregator services for administrative and automation purposes.
Proxy Server	Data Aggregator	• **TCP/HTTP 8581** Allows for OpenAPI access in a fault tolerant environment. Opening this port exposes the Data Aggregator REST services. Open only for clients that require direct access to the Data Aggregator services for administrative and automation purposes. • **TCP/HTTP 8581** Allows for secure OpenAPI access in a fault tolerant environment. Opening this port exposes the Data Aggregator REST services. Open only for clients that require direct access to the Data Aggregator services for administrative and automation purposes.

Data Aggregator	NetOps Portal	• **TCP/HTTP 8281** Enables communications between the Event Manager, which is installed automatically with the NetOps Portal software, and the data aggregator. The data aggregator initiates communication and pushes data through this port. • **TCP/HTTP 8381** Enables communication between the Data Aggregator and NetOps Portal for direct authentication of OpenAPI queries. For secured communications, use the following ports: • **TCP/HTTPS 8382**If NetOps Portal is configured to use HTTPS, this port enables secured communication between the data aggregator and NetOps Portal for direct authentication of OpenAPI queries. For more information, see Configure the Port and Website for HTTPS.
NetOps Portal	Data Aggregator	• **TCP/HTTP 8581** Enables synchronization with DX NetOps Performance Management for the Data Aggregator. NetOps Portal initiates communication and pulls data through this port. For secured communications, use the following ports: • **TCP/HTTPS 8582** If you have configured the data aggregator to use HTTPS, this port enables secured synchronization with DX NetOps Performance Management for the Data Aggregator. NetOps Portal initiates communication and pulls data through this port. For more information, see Configure the Port and Website for HTTPS.

NetOps Portal	Proxy Server	• **TCP/HTTP 8581** In a fault tolerant environment, enables synchronization with DX NetOps Performance Management for the Data Aggregator. NetOps Portal initiates communication and pulls data through this port. For secured communications, use the following ports: • **TCP/HTTPS 8582** If you have configured the data aggregator to use HTTPS, this port enables secured synchronization with DX NetOps Performance Management for the Data Aggregator in a fault tolerant environment. NetOps Portal initiates communication and pulls data through this port. For more information, see Configure the Port and Website for HTTPS.
NetOps Portal	Network Flow Analysis	• **TCP/HTTP 80** Enables synchronization with Network Flow Analysis to retrieve configuration data. • **TCP/HTTP 8681** Enables synchronization with Network Flow Analysis to retrieve device data. • **TCP/HTTP 8981** Enables communication between Network Flow Analysis and NetOps Portal.
NetOps Portal	CA Application Delivery Analysis	• **TCP/HTTP 80** Enables synchronization with CA Application Delivery Analysis to retrieve configuration data. • **TCP/HTTP 8681** Enables synchronization with CA Application Delivery Analysis to retrieve device data.
CA Business Intelligence	NetOps Portal	• **TCP/HTTP 8181** Enables communications between CA Business Intelligence and the NetOps Portal server. For secured communications, use the following port instead of 8181: • **TCP/HTTPS 8182** This port is the default port for NetOps Portal if configured using the HTTPS documentation. For more information, see Configure the Port and Website for HTTPS.

Data Collector	Data Aggregator	• **TCP 8581** Enables the simplified upgrade for Data Collectors. For more information, see Upgrade the Data Collectors. • **TCP/AMQ 61616** Enables only ActiveMQ traffic between the Data Collector and Data Aggregator. • **TCP/AMQ 61618** Enables poll response delivery traffic between the Data Collector and Data Aggregator. • **TCP/AMQ 61620** Enables distributed IREP traffic between the Data Collector and Data Aggregator. • **TCP/AMQ 61622**Enables large data transfers between the Data Collector and Data Aggregator. This port also enables the simplified upgrade for Data Collectors. For more information, see Upgrade the Data Collectors. For secured communications, use the following ports instead of 61616, 61618, 61620, 61622: The following ports are the default ports for Secure ActiveMQ communication if configured using the AMQ SSL documentation. For more information, see Authenticate and Encrypt ActiveMQ Communication. • **TCP/AMQ SSL 61617** Enables only ActiveMQ secured communications between the Data Collector and Data Aggregator. • **TCP/AMQ SSL 61619** Enables poll response delivery secured communications between the Data Collector and Data Aggregator. • **TCP/AMQ SSL 61621** Enables distributed IREP secured communications between the Data Collector and Data Aggregator. • **TCP/AMQ SSL 61623** Enables secured large data transfers between the Data Collector and Data Aggregator.
Data Collectors	Devices	• **UDP 161** Enables SNMP and ICMP connections to devices. To enable ping during discovery and reachability checks, ICMP must be enabled on the devices and the network.

Data Aggregator	Data Repository	• **TCP/UDP 5433** Enables communication between the Data Aggregator and the Data Repository for Java Database Connectivity.
Data Repository	Data Repository	• **TCP/SSH 22** Enables Vertica administration tools and backup to run between nodes. • **TCP/UDP 4803** Enables spread communication between nodes. • **TCP/UDP 5433** Enables communication between the Data Aggregator and the Data Repository for Java Database Connectivity. Open the following ports for the Vertica database: • **UDP 4804** • **TCP 5434** • **UDP 6543**
Data Repository	Backup Hosts	• **TCP 50000** Enables the Data Repository host to access the custom rsync/ssh on the backup hosts.
Data Repository	Disaster Recovery Hosts	• **TCP 50000** Enables the Data Repository host to access the custom rsync/ssh on the disaster recovery hosts.
DX NetOps Spectrum	NetOps Portal	• **TCP 8281** For event integration, enables the DX NetOps Spectrum OneClick server to communicate to the NetOps Portal host. • **TCP 8481** Enables the DX NetOps Spectrum OneClick server to communicate to the Device Manager.
NetOps Portal	LDAP	• **TCP 389** Enables Clear Text communication from the client to the LDAP server. • **TCP 3268** If you are using the global catalog for searches, enables communication from the client to the LDAP server.
NetOps Portal	LDAPS	• **TCP 636** Enables encrypted and secure communication from the client to the Secure LDAP server. • **TCP 3269** If you are using the global catalog for searches, enables communication from the client to the Secure LDAP server.

Consul Servers (the proxy server, active Data Aggregator, inactive Data Aggregator)	Consul Servers (the proxy server, active Data Aggregator, inactive Data Aggregator)	• **TCP 8300** In a fault tolerant environment, enables communication between the proxy server and the Data Aggregators. • **TCP/UDP 8301** In a fault tolerant environment, enables LAN communication between the proxy server and the Data Aggregators. • **TCP 8500** In a fault tolerant environment, enables communication between the proxy server and the Data Aggregators to the HTTP API.

Review Cloud Sizing Guidelines

If you plan to stand up CA Performance Management in the cloud, review the following cloud sizing guidelines. You can choose from a variety of cloud platforms including Amazon Web Services (AWS), Google Cloud Platform (GCP), and so on. Our current guidelines focus on AWS.

Follow these steps:

1. Complete the CA Performance Management sizing tool to determine the estimated requirements of your environment.
   For more information, see the CA Performance Management Sizing Tool.
2. Review the estimated disk requirements from the sizer and consider your data retention rates. Your data retention rates impact disk requirements greatly.
   For more information, see Configure Data Retention Rates.
3. Review the other entries in the sizer, which impact the requirements of your environment.
4. Adjust the entries in the sizer until they most accurately reflect the needs of your environment.
5. Review the guidelines for your cloud platform.

AWS

Components	Instance Types	Descriptions	Added Storage	Notes	Examples
Data Repository / Vertica	r5.4, r5.8, r5.12	r5 instance types are memory optimized instances. These instances are designed to deliver fast performance for workloads that process large data sets in memory. We recommend r5 instance types for the Data Repository because Vertica supports them. For more information, see the Vertica on Amazon Web Services documentation. Also, the core to memory ratio best matches what our sizer recommends. Storage does not factor into this recommendation. Use one of the recommended added storage options from EBS to cover the recommended disk size from the sizer.	IOPS SSD, Thoughput HDD	To select the instance type best suited for your Data Repository, review the estimated CPU requirements in the sizing tool. The Data Repository requires storage beyond the boot disk. Amazon EBS allows you to create storage volumes and attach them to Amazon EC2 instances. Use EBS for the catalog directory and the data directory.	Using the sizer with the default values and 50,000 devices, the expected Data Repository sizing would be 14 cores and 224 GiB RAM. The r5.8 has 32 hyperthreaded cores and 256 GiB RAM. You can deselect the hyperthreaded option to run at 16 cores and 256 GiB RAM, which gives the expected ratio by the sizer. You are not required to deselect the hyperthreaded option, as the benefit can vary from deployment to deployment.
Data Aggregator	m5.1, m5.2, m5.4	m5 instance types provide a balance of compute, memory, and networking resources. The 1:4 CPU to memory ratio best matches the recommendations from our sizer.	No	To select the instance type best suited for your Data Aggregator, review the estimated memory requirements in the sizing tool.	Using the sizer with the default values and 50,000 devices, the expected Data Aggregator sizing is 18 cores and 48 GiB RAM. The m5.4 is the closest fitting instance type.

Data Collector	c5d.1, c5d.2, c5d4	c5 instance types are compute optimized instances. These instances are designed for compute bound applications that benefit from high performance processors. The 1:2 CPU to memory ratio best matches the recommendations from our sizer.	No	To select the instance type best suited for your Data Collectors, consider the following guidance: • Select c5d.1 in a small environment monitoring around 100K items. • Select c5d.2 in a medium environment monitoring around 500K items. • Select c5d.4 in a large environment monitoring around 1000K items or an environment with intensive metrics or polling rates.	Using the sizer with the default values and 50,000 devices, the expected Data Colector sizing would be 8 cores and 16 GiB RAM. The c5d.2 fits the sizer recommendations exactly.
Performance Center MySQL Database Node	m5.1, m5.2, c5	m5 instance types provide a balance of compute, memory, and networking resources. c5 instance types are compute optimized instances. These instances are designed for compute bound applications that benefit from high performance processors.	IOPS SSD	To select the instance type best suited for your Performance Center MySQL Database Node, review the estimated memory requirements in the sizing tool. The MySQL Database Node requires storage beyond the boot disk. Amazon EBS allows you to create storage volumes and attach them to Amazon EC2 instances. Use EBS for the MySQL data directory.	Using the sizer with the default values and 50,000 devices, the expected Performance Center MySQL node sizing would be 12 cores and 32 GiB RAM. In this example, we recommend the m5.2. If you have a high user load, it might be best to use a c5.4 to help with concurrent requests made by users.

Performance Center Core Services Node	c5.4	c5 instance types are compute optimized instances. These instances are designed for compute bound applications that benefit from high performance processors.	No	To select the instance type best suited for your Performance Center Core Services Node, review the estimated CPU requirements in the sizing tool.	Using the sizer with the default values and 50,000 devices, the expected Performance Center MySQL node sizing would be 12 cores and 32 GiB RAM. In this example, we recommend the m5.2. If you have a high user load, it might be best to use a c5.4 to help with concurrent requests made by users.
CA Virtual Network Assurance	r5.2	r5 instance types are memory optimized instances. These instances are designed to deliver fast performance for workloads that process large data sets in memory. CA Virtual Network Assurance, varying based on the plugins deployed, is a memory intensive application. The r5 instance type scales best for both smaller and larger VNA deployments.	No	r5.2 is the only recommended instance type for CA Virtual Network Assurance.	
CA Spectrum SS Node	c5.2, c.5.4	c5 instance types are compute optimized instances. These instances are designed for compute bound applications that benefit from high performance processors. The 1:2 CPU to memory ratio best matches the recommendations from our sizer.	No	To select the instance type best suited for your CA Spectrum SS Node, review the estimated memory requirements in the CA Spectrum Sizing Tool.	

CA Spectrum OC Node	c5.1, c5.2, c5.4	c5 instance types are compute optimized instances. These instances are designed for compute bound applications that benefit from high performance processors. The 1:2 CPU to memory ratio best matches the recommendations from our sizer.	No	To select the instance type best suited for your CA Spectrum OC Node, review the estimated memory requirements in the CA Spectrum Sizing Tool.	

Prepare to Install Performance Center

To ensure that your NetOps Portal installation is successful, complete the requirements before you install NetOps Portal:

Verify the Prerequisites

- Review Installation Requirements and Considerations.
- The software can be installed in any filesystem to which the root user has write access. The default installation directory is /opt/CA/PerformanceCenter. The setup program lets you select another location.
- Verify that Security Enhanced Linux (SELinux) is disabled on the server where you plan to install NetOps Portal. By default, some Linux distributions enable this feature, which does not allow the product to function properly.

  NOTE
  For information about configuring an SELinux security policy, see the Red Hat documentation.
- By default, the MySQL database is installed to /opt/CA/MySql/, but you can select another location during the installation. Verify that the selected filesystem has enough allocated disk space to support a database.
- To avoid database corruption, exclude the installation directory, and all its subdirectories, from antivirus scans. Prevent scanning by a local instance of an antivirus client and scanning by a remote antivirus instance. Exclude the following directories:
  - `/opt/CA/MySql/`
  - `/opt/CA/MySql/bin`
  - `/opt/CA/MySql/data`
  - `/opt/CA/MySql/tmp`
  - `/opt/CA/PerformanceCenter`
- NetOps Portal requires DNS resolution. If DNS is not configured, add system entries to the /etc/hosts file on your server manually.
- Verify that your /tmp location has at least 4 GB of space available.
- The installer requires the zip and unzip packages. If these packages are not installed, use one of the following commands to install them:

  ```
  yum -y install zip unzip
  ```

  SLES:

  ```
  zypper install -y zip unzip
  ```
- NetOps Portal requires the wget package. If this package is not installed, use one of the following commands to install it:

  ```
  yum -y install wget
  ```

SLES:

```
zypper install -y wget
```

- Perl is required to run some of the available scripts.

Set the Limit on the Number of Open Files

Verify that the user account that is installing NetOps Portal has a value of at least 65536 on the number of open files. Set this value permanently.

> **NOTE**
>
> For systems where a sudo user installs NetOps Portal, the installation user might not have the required permissions to complete this procedure. Work with the system administrator to configure the limit.

Follow these steps:

1. Log in to the NetOps Portal host.
2. Edit the following file:
   `/etc/security/limits.conf`
3. Add the following lines to the file:

```
# Added by Performance Center
* soft nofile 65536
# Added by Performance Center
* hard nofile 65536
```

4. Restart the login session.
5. Verify that the number of open files is set properly:

```
ulimit -n
```

   The command returns the limit that you have specified.

Verify Communication Ports

NetOps Portal uses multiple ports to communicate with various components, particularly data sources. In addition, some of the products and components that integrate with NetOps Portal have specific port requirements. Consult the documentation of each data source for the list of required ports.

> **WARNING**
> For any firewall that protects this server, open the required ports and protocols for the data sources you are deploying. The product documentation for each data source provides a list of required ports and protocols.

Each data source uses unique ports.

The following communication ports allow NetOps Portal services to communicate with NetOps Portal:

- **TCP 3306**
  Enables communications to the MySQL database (inbound) from the NetOps Portal services.
- **TCP/HTTP 8481**
  Enables communications between the Device Manager and Console services.

The following communication ports allow users to contact NetOps Portal:

- **TCP/HTTP 8181**
  Enables communications between client computers and the NetOps Portal server. Enables console communications with data sources.
- **TCP/HTTP 8381**
  Enables communications between client computers and the NetOps Portal server. Also enables login using the single sign-on authentication component.

The following communication ports allow other data sources to contact NetOps Portal for eventing and OpenAPI single sign-on:

- **TCP/HTTP 8281**
  Enables communications between the Event Manager, which is installed automatically with the NetOps Portal software, and the data sources. The Data Aggregator initiates communication and pushes data through this port.
- **TCP/HTTP 8381**
  Enables communication between the Data Aggregator and NetOps Portal for direct authentication of OpenAPI queries.

The following communication ports must be open on the other data sources:

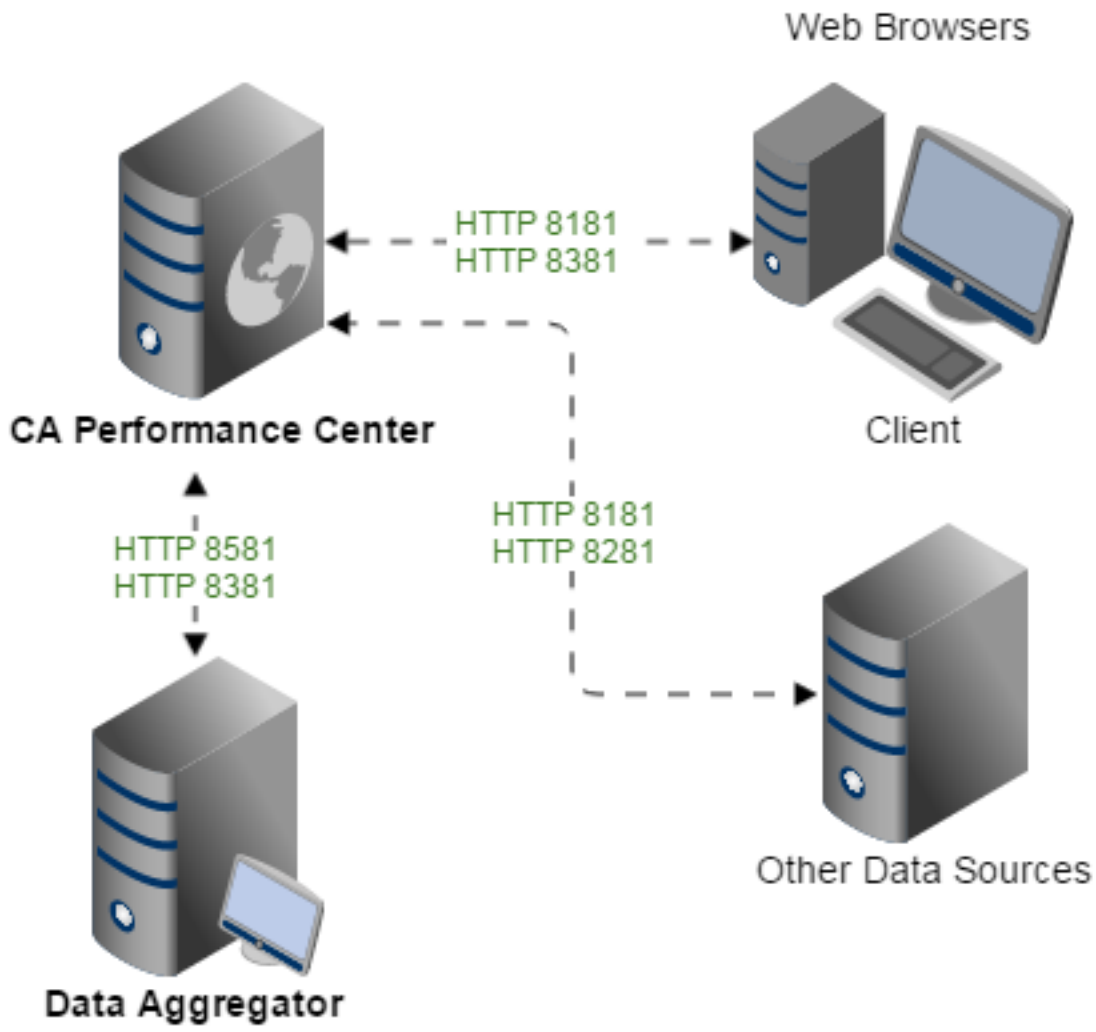
- **TCP/HTTP 80**
  Enables synchronization with CA Network Flow Analysis to retrieve configuration data.
- **TCP/HTTP 8681**
  Enables synchronization with CA Network Flow Analysis to retrieve device data.
- **TCP/HTTP 8581**
  Enables synchronization with DX NetOps Performance Management. For the Data Aggregator, NetOps Portal initiates communication and pulls data through this port.

The following diagram illustrates the default port usage:

NOTE

For DX NetOps Performance Management to work properly in a firewall-protected environment, certain ports must be open. Throughout the documentation 8182, 8382, and 61617 appear as suggested port numbers for secured communications. In the instances where these appear, you are free to use any value you want as long as no other processes are using it. For more information about firewall and connectivity considerations, see Review Installation Requirements and Considerations

Figure 5: Performance Management Ports



Verify Time Synchronization

Time synchronization using the Network Time Protocol (NTP) daemon is required for NetOps Portal and is recommended for all data source consoles. On Linux servers, the NTP daemon ensures that the clocks on the hosts are synchronized for timing purposes. Verify that the daemon is running on the NetOps Portal host server.

SLES System

Use the following procedure to verify time synchronization on an SLES system.

Follow these steps:

1. Open a console and type the following command:
   ```
   $ systemctl status ntpd
   ```
2. Verify that the NTP daemon is in an active (running) state.

3. Type the following command to start and enable the NTP daemon manually:
   ```
   $ systemctl start ntpd
   $ systemctl enable ntpd
   ```
   The daemon is started.

RHEL 7.x or OL 7.x System

RHEL 7.x and OL 7.x. run NTP with `chronyd` . Use the following procedure to verify time synchronization on a RHEL 7.x or OL 7.x system.

Follow these steps:

1. Open a console and type the following command:
   ```
   $ systemctl status chronyd
   ```
2. Verify that the `chrony` daemon is in an active (running) state.
3. Type the following command to start and enable the `chrony` daemon manually:
   ```
   $ systemctl start chronyd
   $ systemctl enable chronyd
   ```
   The daemon is started.

RHEL 6.x System

Use the following procedure to verify time synchronization on a RHEL 6.x system.

Follow these steps:

1. Open a console and type the following command:
   ```
   $ chkconfig --list ntpd
   ```
   If the NTP daemon is installed, the output resembles the following example:
   ```
   ntp 0:off 1:off 2:on 3:on 4:off 5:on 6:off
   ```
   The output indicates the runlevels where the daemon runs.
2. Verify that the current runlevel of the system (usually 3 or 5) has the NTP daemon set to 'on'. If you do not know the current runlevel, type the following commands to find it:
   ```
   $ runlevel
   N 3
   ```
   If the current runlevel does not have the NTP daemon enabled, enable it by typing the following command:
   ```
   $ chkconfig --level current_runlevel ntpd on
   ```
   Example:
   ```
   $ chkconfig --level 3 ntpd on
   ```
3. Type the following command to start the NTP daemon manually:
   ```
   $ service ntpd start
   ```
   The daemon is started.

(Optional) Configure the Sudo User Account for NetOps Portal

If you do not have root access to install and run NetOps Portal, configure the sudo user account.

Follow these steps:

1. Locate the following file on the NetOps Portal host:
   `/etc/sudoers`
2. Add a command alias with the following permissions to the file:

- /tmp/CAPerfCenterSetup.bin
- /etc/init.d/caperfcenter_console **(for RHEL 6.x)**
- /etc/init.d/caperfcenter_devicemanager **(for RHEL 6.x)**
- /etc/init.d/caperfcenter_eventmanager **(for RHEL 6.x)**
- /etc/init.d/caperfcenter_sso **(for RHEL 6.x)**
- /etc/init.d/mysql
- /opt/CA/PerformanceCenter/PC/bin/caperfcenter_console **(for RHEL 7.x, SLES, OL)**
- /opt/CA/PerformanceCenter/DM/bin/caperfcenter_devicemanager **(for RHEL 7.x, SLES, OL)**
- /opt/CA/PerformanceCenter/EM/bin/caperfcenter_eventmanager **(for RHEL 7.x, SLES, OL)**
- /opt/CA/PerformanceCenter/sso/bin/caperfcenter_sso **(for RHEL 7.x, SLES, OL)**
- /opt/CA/PerformanceCenter/Tools/bin/npcshell.sh
- /opt/CA/PerformanceCenter/SsoConfig
- /opt/CA/PerformanceCenter/Uninstall_MySql
- /opt/CA/PerformanceCenter/Uninstall_PerformanceCenter
- /opt/CA/PerformanceCenter/Uninstall_SSO
- /sbin/service
- /opt/CA/MySql/bin/mysql
- /opt/CA/MySql/bin/mysqldump
- /opt/CA/PerformanceCenter/sso
- /opt/CA/PerformanceCenter/PC
- /opt/CA/PerformanceCenter/PC/webapps/pc/apps
- /opt/CA/PerformanceCenter/PC/webapps/pc/css/CA-Blue/images
- /opt/CA/PerformanceCenter/PC/webapps/pc/css/CA-Gray/images
- /usr/bin/vim
- /opt/CA/jre/bin/keytool
- /opt/CA/PerformanceCenter/RemoteEngineer/re.sh
- /opt/CA/PerformanceCentre/SslConfig

Separate the permissions with commas and place all permissions on a single line.

Example:

```
Cmnd_Alias CAPERFCENTER = /tmp/CAPerfCenterSetup.bin,/opt/CA/
PerformanceCenter/PC/bin/caperfcenter_console,/opt/CA/PerformanceCenter/
DM/bin/caperfcenter_devicemanager,/opt/CA/PerformanceCenter/EM/bin/
caperfcenter_eventmanager,/opt/CA/PerformanceCenter/sso/bin/caperfcenter_sso,/
etc/init.d/mysql,/opt/CA/PerformanceCenter/Tools/bin/npcshell.sh,/opt/CA/
PerformanceCenter/SsoConfig,/opt/CA/PerformanceCenter/Uninstall_MySql,/opt/
CA/PerformanceCenter/Uninstall_PerformanceCenter,/opt/CA/PerformanceCenter/
Uninstall_SSO,/sbin/service,/opt/CA/MySql/bin/mysql,/opt/CA/MySql/bin/mysqldump,/
opt/CA/PerformanceCenter/sso,/opt/CA/PerformanceCenter/PC,/opt/CA/PerformanceCenter/
PC/webapps/pc/apps,/opt/CA/PerformanceCenter/PC/webapps/pc/css/CA-Blue/images,/opt/
CA/PerformanceCenter/PC/webapps/pc/css/CA-Gray/images,/usr/bin/vim,/opt/CA/jre/bin/
keytool,/opt/CA/PerformanceCenter/RemoteEngineer/re.sh,/opt/CA/PerformanceCentre/
SslConfig
sudouser ALL = CAPERFCENTER
```

- ***sudouser*** Specify the user who can run the sudo commands.

With the sudo user configured, add the sudo prefix to all commands to install NetOps Portal.

Example:

```
sudo ./CAPerfCenterSetup.bin
```

Configure UTF-8 Support

Configure NetOps Portal host to support UTF-8 encoding. If UTF-8 encoding is disabled, characters might not display properly during the installation.

The appropriate language packs are also required to support localized deployments.

> **NOTE**
> Some scripts that are used in the installation of selected components are not localized and run in English. For more information, see the *Localization Status Readme* file.

Follow these steps:

1. Do one of the following steps:
   a. Type the following command from a Korn or bash shell:
      `export LANG=LANG_value ; export LC_ALL=$LANG`
      LANG_value specifies the language that you want the product to support. The following variables are supported:
      - **English:** `en_US.utf8`
      - **French:** `fr_FR.utf8`
      - **Japanese**:`ja_JP.utf8`
      - **Simplified Chinese:** `zh_CN.utf8`
      - **Traditional Chinese:** `zh_TW.utf8`

      For example:
      `export LANG=zh_TW.utf8 ; export LC_ALL=$LANG`
   b. Type the following command from a Bourne Shell:
      `LANG=LANG_value ; export LANG`
      `LC_ALL=LANG_value ; export LC_ALL`
      For example:
      `LANG=zh_CN ; export LANG`
      `LC_ALL=zh_CN ; export LC_ALL`

   The language variable is set.

Install Support for Required and Non-English Fonts

For the installer to run, and for NetOps Portal to generate PDF files, follow the standard instructions for installing the necessary fonts on your operating system.

NetOps Portal and its data sources support multiple languages. The administrator can select a preferred language for each unique product operator. Language packs take advantage of operating system support for localized environments.

Product operators with some language preferences may not be able to view dashboard data in reports by default. To support these language preferences, install the fonts on the NetOps Portal host.

Follow these steps:

1. Log in to the NetOps Portal host.
2. Run one of the following commands to install the fonts:
   `yum groupinstall fonts`
   SLES:
   ```
   zypper install dejavu-sans-fonts dejavu-fonts-common arphic-ukai-fonts arphic-uming-fonts ipa-ex-mincho-fonts ipa-mincho-fonts ipa-pmincho-fonts xano-mincho-fonts baekmuk-bitmap-fonts baekmuk-ttf-fonts liberation-fonts
   ```
3. Run the following command to rebuild the font caches:

```
fc-cache –v
```

SLES:

```
fc-cache -fv
```

4. Restart NetOps Portal.

Install Performance Center

After you have met the requirements to install NetOps Portal, complete the installation as follows:

MySQL Password Requirements (3.7.3 and Higher Only)

As a step toward enhanced security, the Performance Center installation prompts you to set a custom MySQL password.

The MySQL password must meet the following requirements:

- Excludes the user names "root" or "netqos"
- Minimum length of 8 characters
- Maximum length of 30 characters
- Contains at least 3 of the following types of characters:
  - Special Characters (!#&?)
  - Uppercase
  - Lowercase
  - Numbers (0-9)
- Excludes the percentage sign (%), apostrophe ('), or quotation mark (")
- **(3.7.3 through 3.7.4 Only)** Excludes the asterisk (*), or the dollar sign ($)

Install NetOps Portal

Select one of the following options:

Install from the Command Line

Use the NetOps Portal Setup program to install and configure the database and website. The Setup program supports two types of installations. Select one of the following types during installation:

- **Complete:** Installs NetOps Portal services and the MySQL database on a single node.
- **Advanced:** Installs NetOps Portal services and the MySQL database on separate nodes for better performance. First, run the Setup program to install the database. Then, run the Setup program on a different node to install the services.

Follow these steps:

1. Log in to the NetOps Portal host as the root or the `sudo` user.
2. Copy the `CAPerfCenterSetup.bin` file to the `/tmp` directory.

   NOTE
   Verify that your `/tmp` location has at least 4 GB of space available.
3. Change to the `/tmp` directory by issuing the following command:
   ```
   cd /tmp
   ```
4. Change the permissions for the installation file by issuing the following command:
   ```
   chmod +x CAPerfCenterSetup.bin
   ```
5. Run the installation file by issuing the following command:
   ```
   ./CAPerfCenterSetup.bin -i console
   ```

6. Follow the instructions in the console.

   NOTE
   You are prompted to specify an install owner. You can specify a non-root user.

   The installation checks to see whether the partition with the MySQL data directory has enough disk space to handle storage engine upgrades. If there does not appear to be enough disk space to complete the installation successfully, exit the installer, allocate more space for the data partition, and reinstall NetOps Portal.

   (3.7.2 and Higher Only) As a step toward enhanced security, the Performance Center installation prompts you to set a custom MySQL password.

   The installation runs. The following Linux daemons are created and started during the installation:
   - **caperfcenter_console**
     The console daemon. Uses port 8181.
   - **caperfcenter_devicemanager**
     The Device Manager daemon. Uses port 8481.
   - **caperfcenter_eventmanager**
     The Event Manager daemon. Uses port 8281.
   - **caperfcenter_sso**
     The Single Sign-On daemon. Uses port 8381.
   - **mysql**
     The database daemon. Uses port 3306.

   When the installation has completed, a message states that the program has been installed successfully.

Install in Silent Mode

To install NetOps Portal without entering user inputs, install the component in silent mode.

Follow these steps:

1. Log in to the NetOps Portal host as the root or the sudo user.
2. Copy the `CAPerfCenterSetup.bin` file to the `/tmp` directory.

   NOTE

   Verify that your `/tmp` location has at least 4 GB of space available.
3. Change to the `/tmp` directory by issuing the following command:
   ```
   cd /tmp
   ```
4. Change the permissions for the installation file by issuing the following command:
   ```
   chmod +x CAPerfCenterSetup.bin
   ```
5. Run the following command on all servers where you want to install NetOps Portal:
   ```
   ./CAPerfCenterSetup.bin -r /tmp/silent.properties
   ```
6. Follow the prompts until you get to the summary, type `quit` , and then press the return key on your keyboard.
   The `silent.properties` file is created in the `/tmp` directory.
7. Review the `/tmp/silent.properties` file.
8. If present, confirm the values for the following variables in the `silent.properties` file:
   - **USER_INPUT_INSTALL_OWNER**
     Designates a user as the install owner. You can specify a non-root user.
     Default: `root`
   - **USER_INSTALL_DIR**
     Designates the directory where the application is installed.

Default: /opt/CA

- **MYSQL_DATA_FOLDER**
  Designates the location for the MySQL data directory.
  Default: /opt/CA/MySql/data
- **MYSQL_TEMP_FOLDER**
  Designates the location for the directory to store MySQL temporary files.
  Default: /opt/CA/MySql/tmp
- **DB_PASSWORD_VARIABLE**
  Designates the MySQL password.
- **DB_PASSWORD_CONFIRM**
  Confirms the MySQL password.

9. Run the following command on all servers where you want to install NetOps Portal:
   ```
   ./CAPerfCenterSetup.bin -i silent -f /tmp/silent.properties
   ```
   The installation begins.
   An empty prompt indicates that components have been installed.

Verify the Installation

NetOps Portal requires the following Linux daemons:

service_name	Description
`mysql`	Database process
`caperfcenter_devicemanager`	Device Manager process
`caperfcenter_console`	NetOps Portal console process
`caperfcenter_sso`	Single Sign-On process
`caperfcenter_eventmanager`	Event Manager process

After a successful installation, the services listed in this table are running. Check the status of a daemon by issuing the following command:

```
service service_name status
```

Access NetOps Portal at the following URL:

```
http://PC_host:8181/pc/desktop/page
```

- **PC_host** is the hostname or IP address of the NetOps Portal host.

  NOTE
  8181 is the default port. If you specified a different port during the installation, use the custom port number.

If the NetOps Portal login screen appears, Performance center has installed successfully.

IMPORTANT
At the initial login, you are required the change the **admin** and **user** passwords. We recommend that you change these passwords immediately after a fresh install.

If you have Network Flow Analysis as a data source, you must restart the CA NFA OData Service:

1. Click **Administration Services**, and then **Services**.
2. Right-click the **CA NFA OData Service**.
3. Click **Restart**.

(Optional) Review Log Files

The following log files help you track events that occur during the installation:

Installation Errors and Configuration Events

`/opt/CA/PerformanceCenter/InstallLogs`

During the installation, a history file that indicates the installed version is generated in this directory.

Device Manager Daemon

`/opt/CA/PerformanceCenter/DM/logs`

Website and Console Errors

`/opt/CA/PerformanceCenter/PC/logs`

MySQL Database Errors

`/opt/CA/MySql/data/<hostname>.err`

Event Manager (Other Events)

`/opt/CA/PerformanceCenter/EM/logs`

User Authentication (Single Sign-On)

`/opt/CA/PerformanceCenter/sso/logs`

Prepare to Install the Data Repository

To ensure that your Data Repository installation is successful, complete the requirements before you install Data Repository:

For more information about Data Repository configuration options and administration, see Data Repository Administration.

Verify the Prerequisites

Verify the following prerequisites before you install Data Repository:

- Review Installation Requirements and Considerations.
- Review the Vertica documentation.
- Verify whether the dialog and chrony packages are installed on each Data Repository host:

  NOTE

  The chrony package is required only for RHEL 7.x and OL 7.x.

  ```
  rpm -qa | grep ^dialog
  rpm -qa | grep ^chrony
  ```

  If either command does not return results, install the package:

  NOTE

  If you are not the root user, use the sudo prefix.

  ```
  yum install dialog
  yum install chrony
  ```

  If this package is not installed, the validation and installation scripts fail.
- The installer requires the zip and unzip packages. If these packages are not installed, use the following command to install them:

```
yum -y install zip unzip
```

- Verify that you have at least 2 GB of swap space on Data Repository host.
- Verify that the Data Repository hosts use the ext4 file system. Vertica does not support XFS or btrfs. All disks with Vertica should use ext4.

  WARNING

  The default file system for RHEL 7.x and OL 7.x is the XFS file system. The default file system for SLES is btrfs. Vertica does not support XFS or btrfs. The database performs best with the ext4 file system.
- Verify that the following ports are open on the Data Repository systems:
  - Port 22 (TCP protocol)
  - Port 4803 (TCP and UDP protocol)
  - Port 4804 (UDP protocol)
  - Port 5433 (TCP protocol)
    Remote access is required to this port.
  - Port 5434 (TCP protocol)
  - Port 6543 (UDP protocol)
- To avoid database corruption, exclude the installation directory, and all its subdirectories, from antivirus scans. Prevent scanning by a local instance of an antivirus client and scanning by a remote antivirus instance. Exclude the following directories:
  - /opt/vertica/*
  - /opt/vconsole/*
  - The specified data directory
    Default: /drdata/data
  - The specified catalog directory
    Default: /drdata/catalog
  - Vertica temporary files in /tmp
    - /tmp/4803
    - /tmp/vbr/*
  - The directory where you back up the Data Repository
- If a file named 'release' appears in the /etc directory, remove it. Otherwise, the Data Repository installation fails.
- Verify the access according to your installation type:
  - **Single Node:** Root access is required to install Data Repository. Determine whether you can install Data Repository as root.
  - **Cluster:** Verify that the root user or sudo user can create database administrator user accounts, or can have an administrator create these accounts.
- Verify that CPU frequency scaling is disabled. Disable CPU frequency scaling through the host system BIOS and OS settings.

  NOTE

  If CPU frequency scaling is enabled, you might experience inconsistent performance for similar queries in Vertica. CPU frequency scaling can cause observable slowness and variation in dashboard loading.
- Verify that you are not using Logical Volume Manager (LVM) for /data and /catalog directories.
- (Cluster only) Verify all the hosts in the cluster are in the same subnet.
- (Cluster only) Verify that the root user can use Secure Shell (SSH) to log in (ssh) to all the hosts in the cluster.

  NOTE

  Set up SSH for the root user for the Data Repository installation or upgrade.
- The default shell environment must be `bash` .
- (Cluster only) Select the hosts where you install Data Repository nodes.

WARNING
Warning! Database software is deployed on each participating host in a cluster. This software represents a 'node' in the cluster. A three-node cluster represents the simplest configuration that can tolerate the loss of a single node. You can, however, include more than three hosts in the cluster. If more than one node fails or shuts down, Data Repository is no longer available for use and Data Aggregator shuts down automatically.

Install the Data Repository on VMs

For best performance, install the Data Repository in a bare-metal environment. However, if you install the Data Repository in VMware virtual machines, verify the following requirements:

- Use VMware version 5.5 or greater.
- The number of VMs per host does not exceed the number of physical processors.
- Pre-allocate and reserve 4 GB of memory for each of the VMs.
- Each VM has a dedicated 10 GB NIC.
- Disable CPU frequency scaling at the host level and for each VM.
- Disable VMotion. VMotion can disrupt communication, and can cause the Data Repository to shut down.
- Set the VMware parameters for hugepages to the version 5.5 default values.
- Verify the hardware and network performance. Use the Vertica diagnostic tools described bellow to verify performance.

For more information about running Vertica on VMs, see the Vertica documentation.

Install the Data Repository on Shared Storage (SAN)

To install the Data Repository on SAN, verify the following requirements:

- The hosts have no contention for disk space or bandwidth.
- Each host has a unique catalog and data location. The hosts cannot share the location for these directories.
- The storage has enough I/O bandwidth for each node to access the storage independently. To verify the I/O bandwidth, simultaneously run vioperf from all hosts in the Data Repository cluster. For more information, see the following procedures.

Set a Unique Hostname for Each Data Repository Host

Set a unique hostname for each Data Repository host in the cluster.

Follow these steps:

1. As the root user, log in to each Data Repository host, and verify the unique hostname.
   The hostname must be associated with the IP address and *not* the loopback address of 127.0.0.1.
2. Verify that the following lines appear in the /etc/hosts file on each computer:
   ```
   Do not remove the following line, or various programs
   # that require network functionality will fail.
   127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
   ::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
   IP address of your hostYourHostName YourHostName.domain
   ```
3. If you change the file, run the following command:
   ```
   service network restart
   ```
   The /etc/hosts file is configured correctly.
   The unique host name is set.
4. (Cluster installations only) The hostnames of all hosts in the cluster must resolve correctly. If the hostname resolution is incorrect, the Data Repository cluster does not install or work properly. All participating hosts in the cluster must use

static IP or permanently leased DHCP addresses. Set up the /etc/hosts file on each of the hosts you selected for the cluster. The hosts file must contain entries for all hosts in the cluster.
Example: This example shows the /etc/hosts file for a cluster where the hosts are named host01, host02, and host03:

```
127.0.0.1 localhost.localdomain localhost
192.168.13.128 host01.domain host01
192.168.13.129 host02.domain host02
192.168.13.130 host03.domain host03
```

NOTE
Do not remove the loopback address (127.0.0.1) line. The local Data Repository hostname cannot be on the 127.0.0.1 line. Also, do not use the loopback address or localhost name when you are defining hosts in the cluster.

5. Verify that hostname resolution works for each host in the cluster.
   For example, on host01, the following syntax is correct:

```
$ /bin/hostname -f
host01
```

   Hostname resolution is configured.

(Optional) Set Up Passwordless SSH for the Root or Sudo User

The hosts in a Data Repository cluster require passwordless ssh for the root or sudo user during the Data Repository installation or upgrade. The `dr_validate.sh` script sets up passwordless SSH, but requests the password many times. To avoid repeatedly specifying the root or sudo user password, set up passwordless ssh before you run the validation script.

Repeat this procedure for each pair of hosts. If you have passwordless ssh set up for the root user, but you do not have root access to install and run the Data Repository, configure a sudo user account. You also have an alternative method to install the product without requiring to enter the root password by using the sudo user account. For more information about configuring the passwordless sudo user account for Data Repository, see the section Configure the passwordless Sudo User Account for Data Repository.

NOTE

Passwordless SSH is automatically set up for the Data Repository admin user when you install the Data Repository.

Follow these steps:

1. Open a console and log in to the Data Repository host as the root or sudo user.
2. Run the following commands:

```
ssh-keygen -N "" -t rsa -f ~/.ssh/id_rsa
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys2
chmod 644 ~/.ssh/authorized_keys2
```

3. Copy the root or sudo user public key into the list of authorized keys on the remote hosts:

```
ssh-copy-id -i user@remotehost
```

   remotehost specifies a host in the cluster where you are copying the SSH ID.
4. To verify that passwordless ssh is set up correctly, log in to the remote host from the local host:

```
ssh user@remotehost ls
```

   If the passwordless SSH has been set up successfully, you are not prompted for a password. You also see a directory listing from the ‘ls command’.

(Optional) Configure the Sudo User Account for the Data Repository

If you have passwordless ssh set up for the root user, but you do not have root access to install and run the Data Repository, configure a sudo user account.

For cluster environments, complete this procedure on each host in the cluster.

Follow these steps:

1. Locate the following file:
   `/etc/sudoers`
2. Add a command alias with the following permissions to the file:

```
Cmnd_Alias CA_DATAREP = /tmp/installDR.bin,/opt/CA/IMDataRepository_vertica9/
dr_validate.sh,/opt/CA/IMDataRepository_vertica9/dr_install.sh,/usr/bin/vim,/usr/bin/
reboot,/usr/bin/yum,/opt/CA/IMDataRespository_vertica9/RemoteEngineer/re.sh
## Allows the Data Repository user to manage the Data Repository
sudouser ALL = CA_DATAREP
```

- ***sudouser*** Specify the user who can run the sudo commands.

This command alias details the commands that the sudo user must be able to run.
With the sudo user configured, add the sudo prefix to all commands to install the Data Repository.
Example:

```
sudo ./installDR.bin
```

(Optional) Configure the Passwordless Sudo User Account for the Data Repository

Due to certain security policies, in some environments, you cannot enable passwordless SSH for the root users on the host servers. The following procedure provides you an alternative method to install the product without requiring that level of access by using the sudo user account.

> **NOTE**
>
> This functionality is not supported on RHEL 6.x

1. Locate the following file:
   /etc/sudoers
2. Add command aliases with the following permissions to the file:

- On RHEL 7

```
Cmnd_Alias CA_DATAREP=/opt/vertica/sbin/install_vertica,/tmp/installDR.bin,/opt/
CA/IMDataRepository_vertica9/dr_validate.sh,/opt/CA/IMDataRepository_vertica9/
dr_install.sh,/usr/bin/vim,/usr/bin/reboot,/opt/CA/IMDataRespository_vertica9/
RemoteEngineer/re.sh,/bin/mkdir*,/usr/bin/whoami,/bin/echo,/sbin/service,/bin/grep,/
usr/bin/test,/sbin/iptables,/opt/vertica/oss/python/bin/python,/usr/bin/tee,/usr/
sbin/ntpd,/etc/init.d/ntpd,/sbin/blockdev,/etc/init.d/sshd,/etc/sysconfig/sshd,/etc/
ssh/sshd_config,/bin/su,/usr/sbin/sshd restart,/usr/bin/ssh,/bin/df,/bin/mv,/bin/rm,/
usr/bin/install


Cmnd_Alias VERTICA = /opt/vertica/bin/,/opt/vertica/sbin/,/opt/vertica/oss/python/
bin/
```

```
Cmnd_Alias VERTICA_INSTALL = /bin/echo,/bin/ps -A,/bin/cp /opt/vertica/config/
admintools.conf /opt/vertica/config/admintools.conf.bak.*,/bin/rm -rf /tmp/
dbRPM.rpm,/bin/df --portability /tmp,/usr/bin/install --owner * --mode 700 -d *,/bin/
mv -f /tmp/vstage-*/file /tmp/*,/bin/rm -rf /tmp/vstage-*,/usr/bin/id *,/bin/cp -T /
opt/vertica/* /tmp/vstage-*,/bin/su --login dbadmin *,/bin/mkdir -p /opt/vertica/*,/
bin/touch /opt/vertica/*,/bin/rm -rf /opt/vertica/*,/bin/mv -f /tmp/vstage-* /opt/
vertica/*,/bin/mkdir -p /opt/vertica/*,/bin/touch /opt/vertica/config/users/dbadmin/
agent.conf,/bin/su dbadmin *,/bin/sh -c *,/usr/bin,/opt/vertica/share/binlib/test/*,/
usr/bin/su dbadmin,/bin/test [ -e /* ],/usr/bin/[ -e /* ]


Cmnd_Alias USEFUL = /usr/bin/lshw,/usr/bin/yum,/bin/rpm,/sbin/reboot,/sbin/shutdown,/
usr/bin/cpan,/bin/chgrp,/bin/chmod,/bin/chown,/bin/mnt,/usr/bin/test,/bin/[,/sbin/
service


## Allows the Data Repository user to manage the Data Repository
sudouser ALL = CA_DATAREP, VERTICA , VERTICA_INSTALL , USEFUL


Defaults env_keep +="VERT_DBA_USR VERT_DBA_HOME VERT_DBA_GRP VERT_DBA_DATA_DIR
 _ENV_VPWD_VAR"
```

- On SLES 12

```
Cmnd_Alias CA_DATAREP =/opt/vertica/sbin/install_vertica,/tmp/installDR.bin,/opt/
CA/IMDataRepository_vertica9/dr_validate.sh,/opt/CA/IMDataRepository_vertica9/
dr_install.sh,/usr/bin/vim,/usr/bin/reboot,/opt/CA/IMDataRespository_vertica9/
RemoteEngineer/re.sh,/usr/bin/mkdir, /sbin/SuSEfirewall2 off *,/usr/bin/whoami,/
usr/bin/echo,/usr/bin/id,/usr/bin/env,/usr/sbin/service,/usr/bin/grep,/usr/bin/
test,/sbin/iptables,/opt/vertica/oss/python/bin/python,/usr/bin/tee,/usr/sbin/
ntpd,/etc/init.d/ntpd,/sbin/blockdev,/etc/init.d/sshd,/etc/sysconfig/sshd,/etc/ssh/
sshd_config,/usr/bin/su,/usr/sbin/sshd restart,/usr/bin/ssh,/usr/bin/sh,/usr/bin/
install
Cmnd_Alias VERTICA = /opt/vertica/bin/,/opt/vertica/sbin/,/opt/vertica/oss/python/
bin/
Cmnd_Alias VERTICA_INSTALL = /usr/bin/echo,/usr/bin/ps -A,/usr/bin/cp /opt/vertica/
config/admintools.conf /opt/vertica/config/admintools.conf.bak.*,/usr/bin/rm -rf /
tmp/dbRPM.rpm,/usr/bin/df --portability /tmp,/usr/bin/install --owner * --mode 700
 -d *,/usr/bin/mv -f /tmp/vstage-*/file /tmp/*,/usr/bin/rm -rf /tmp/vstage-*,/usr/
bin/id *,/usr/bin/cp -T /opt/vertica/* /tmp/vstage-*,/usr/bin/su --login dbadmin *,/
usr/bin/mkdir -p /opt/vertica/*,/usr/bin/touch /opt/vertica/*,/usr/bin/rm -rf /opt/
vertica/*,/usr/bin/mv -f /tmp/vstage-* /opt/vertica/*,/usr/bin/mkdir -p /opt/vertica/
*,/usr/bin/touch /opt/vertica/config/users/dbadmin/agent.conf,/usr/bin/su dbadmin *,/
usr/bin/sh -c *,/opt/vertica/share/binlib/test/*,/usr/bin/su dbadmin,/usr/bin/test
 [ -e /* ],/usr/bin/[ -e /* ]
```

```
Cmnd_Alias USEFUL = /usr/bin/lshw,/usr/bin/yum,/bin/rpm,/sbin/reboot,/sbin/shutdown,/
usr/bin/cpan,/bin/chgrp,/bin/chmod,/bin/chown,/bin/mnt,/usr/bin/test,/bin/[,/sbin/
service


## Allows the Data Repository user to manage the Data Repository
sudouser ALL = CA_DATAREP, VERTICA , VERTICA_INSTALL , USEFUL


Defaults env_keep +="VERT_DBA_USR VERT_DBA_HOME VERT_DBA_GRP VERT_DBA_DATA_DIR
 _ENV_VPWD_VAR"
```

Install the Data Repository

After you meet the prerequisites described in Prepare to Install the Data Repository, complete the installation as follows:

NOTE
You can install the database with root user passwordless SSH or with sudo user passwordless SSH configured.

The following video shows the installation process:

The Data Repository installation creates two users. A third user is created when you install the data aggregator. The following table provides information about these users:

New User Example	Password Example	Operating System User Account?	Vertica Database User Account?	Notes	Permissions
dradmin	drpass	Yes	Yes	This user is the first user that you created when you installed the Data Repository. When this user is created, the verticadba group is also created. This user is added to this group.	This user can run the Data Repository processes and the Administration Tools utility. This user owns Data Repository catalog files, data files, and so on.
dauser	dbpassword	No	Yes	This useris the user that the data aggregator uses to interact with the database. The installation script creates this user during the data aggregator installation.	

NOTE
Vertica includes the verticadba group for tighter control over filesystem access in the `/opt/vertica/` directories. During the installation, the verticadba group is created, and existing users are added to the group

with permissions set to 775. This setting grants full privileges to the verticadba group and read/execute privileges to all other users. The `/opt/vertica/log` and `/opt/vertica/config` directories are the folders with the modified permissions.

Install the Database with Root User Passwordless SSH

To set up the Data Repository, you can install and configure the Vertica database as the root user or sudo user that has or will have passwordless SSH configured.

In a cluster installation, initiate the Data Repository installation from any of the hosts that participates in the cluster. The instation pushes the required software components to the additional nodes.

Follow these steps:

1. Log in to any host in the Data Repository cluster as the root user.
2. Copy the `installDR.bin` file locally.
3. Change permissions for the installation file by issuing the following command:
   ```
   chmod u+x installDR.bin
   ```
4. Extract the installation file using one of the following options:
   - As the root user by issuing the following command:
     ```
     ./installDR.bin
     ```
   - As the sudo user by issuing the following command:
     ```
     sudo ./installDR.bin
     ```

     NOTE
     The `installDR.bin` file extracts the Data Repository rpm, the license file, and the three installation scripts. You install the Data Repository later in this procedure.
5. Follow the instructions in the console.
6. When prompted, specify the installation directory to which to extract the Data Repository installation package and Vertica license file. The default installation directory is `/opt/CA/IMDataRepository_vertica`*`Version`*`/` . Press the Return key on your keybaord twice.

   NOTE
   The installation script generates WARN messages for any LVM present in the environment. For help, contact Support.

   The Data Repository installation package, license file, and associated setup scripts are extracted to the chosen directory.
7. Adjust the following parameters in the `drinstall.properties` file to reflect your installation-specific values. This file applies to `dr_validate.sh` and `dr_install.sh` . The `drinstall.properties` file exists in the installation directory that you specified previously.
   - DbAdminLinuxUser
     The Linux user that is created to serve as the Vertica database administrator.
     Default: dradmin
   - DbAdminLinuxUserHome
     The Vertica Linux database administrator user home directory.
     Default: `/export/dradmin`

     NOTE
     This directory is created if the Vertica installer creates the user. Ensure that the directory leading up to the home account already exists on the system. For example, if you are using `/export/dradmin` , be sure that `/export` exists.
   - DbDataDir
     The location of the data directory.

Default: `/data`

IMPORTANT
Do not use the Logical Volume Manager (LVM) for the data directory.

- DbCatalogDir
  The location of the catalog directory.
  Default: `/catalog`

  IMPORTANT
  Do not use the Logical Volume Manager (LVM) for the catalog directory.
- DbHostNames
  The comma-delimited list of hostnames for the Data Repository.
  Default: yourhostname1,yourhostname2,yourhostname3
- DbName
  The database name.
  Default: drdata

  NOTE
  This parameter is case-sensitive.
- DbPwd
  The database password.
  Default: dbpass

  NOTE
  The `dr_install.sh` installation script uses the database password that you define here during the installation of the data aggregator. You can use special characters (except for single quotation marks) in passwords. To use special characters, encase the password with single quotation marks (for example, `DbPwd='test$tr|ng'`). If the script does not find the `DbPwd` property or if it is blank, the script prompts for this information at runtime.

8. Run the `dr_validate.sh` validation script. This script verifies the OS settings and modifies the settings if necessary. To run the validation script as the root user, issue the following command:
   `./dr_validate.sh -p` *`properties_file`*

   The validation script establishes SSH without a password for the root user across all hosts in a cluster. If SSH without a password does not exist for the root account, you are prompted for a password. You are sometimes prompted multiple times.

   TIP
   You can use the `-l` flag to allow `localhost` as the value for the `DbHostNames` property. You can use the `-n` flag to skip database connectivity checks.
9. Review any on-screen output for failures or warnings. You can run the validation script multiple times after you fix any failures or warnings. The script automatically corrects many failures or warnings. Proceed only if the final status is “PASSED”. If the final status is not "PASSED", contact Support.
   The validation script might ask you to reboot.

   NOTE
   The validation script and the installation script generate a log file in the *`installation_directory`*`/logs` directory on the Data Repository host from which you run the scripts. These log files include the step-by-step output of the scripts. To validate successful/failed script runs, review the script output.

   The following example shows the script output and lists what settings the script verifies and changes:

```
Log File: logs/install_log_validate_10-29-2015_11-14-11.log
================================================================================
Checking Passwordless SSH to all hosts: verticahost-dr
================================================================================
```

```
Passwordless SSH from verticahost-dr to root@verticahost-dr ..................[ OK ]
================================================================================
Beginning Data Repository Prerequisite Compliance Enforcement on host verticahost-dr
================================================================================
Red Hat Enterprise Linux Major Release:  6 ..............................[ OK ]
Processor Type: Intel ...................................................[ OK ]
CPU frequency scaling not available on this system ......................[ OK ]
DR Administrative User dradmin does not exist.  It will be created during vertica
 installation. [ OK ]
Maximum number of file handles >= 65536 .................................[ OK ]
Detected incorrect maximum number of memory maps ........................[WARN]
Set maximum number of memory maps to Total Mem(KB)/16 ...................[ OK ]
Detected incorrect page reclaim threshold value .........................[WARN]
Set page reclaim threshold value to 7924 ................................[ OK ]
Disabling necessary firewall settings. ..................................[ OK ]
Enabling NTP daemon. ....................................................[ OK ]
Starting the NTP daemon. ................................................[ OK ]
Detected incorrect readahead parameter for /dev/sda ......................[WARN]
Set readahead parameter for /dev/sda to 2048 .............................[ OK ]
Block Size for /dev/sda is 4096 .........................................[ OK ]
Readahead parameter for /dev/sda1 is 2048 ...............................[ OK ]
Block Size for /dev/sda1 is 1024.  Expected value >= 4096 ................[WARN]
Readahead parameter for /dev/sda2 is 2048 ...............................[ OK ]
Block Size for /dev/sda2 is 4096 .........................................[ OK ]
Readahead parameter for /dev/sda3 is 2048 ...............................[ OK ]
Block Size for /dev/sda3 is 4096 .........................................[ OK ]
Detected incorrect swappiness setting ...................................[WARN]
Set swappiness to 0 .....................................................[ OK ]
Transparent hugepages in /sys/kernel/mm/redhat_transparent_hugepage/enabled are
 enabled [WARN]
Disabled Huge Page Compaction ...........................................[ OK ]
Huge Page Compaction Defrag in /sys/kernel/mm/redhat_transparent_hugepage/defrag is
 enabled [WARN]
Disabled Huge Page Compaction Defrag ....................................[ OK ]
Disk Scheduler for sda is not deadline ..................................[WARN]
Set Disk Scheduler for sda to deadline ..................................[ OK ]
Reloading sysctl.conf ...................................................[WARN]
SELinux is disabled .....................................................[ OK ]
Verifying Swap Space. ...................................................[ OK ]
No Logical Volumes exist. ...............................................[ OK ]
Root entry exists in /etc/sudoers file. .................................[ OK ]
Verifying ext3 or ext4 filesystem used for data directory. ...............[ OK ]
Verifying ext3 or ext4 filesystem used for catalog directory. ............[ OK ]
Fresh install of Vertica is being performed - skipping database connectivity testing.


Data Repository Prerequisite Compliance Status on host verticahost-dr -- PASSED
```

```
================================================================================
Script finished - /user/home/verticahost/dr_validate.sh
================================================================================
```

NOTE
If the `dr_install.sh` installation fails early enough in the process, the log file might be available in the home directory of the root or sudo user.

10. Run the `dr_install.sh` installation script:
    `./dr_install.sh -p properties_file`
    This script installs the data repository, creates the database, and disables unnecessary Vertica processes on all the hosts in the cluster.
    If the database administrator user does not already exist, the installation script creates the user. The script prompts you to assign a new password. If the database administrator user exists, but passwordless SSH is not set up, the script prompts for the password to set up.
    If the installation script returns a WARN message for LVM on directories that Vertica does not use, contact Support.
11. Verify that the installation script has installed the Data Repository successfully by doing the following steps:
    a. Log in to the database server as the database administrator user by issuing the following command:
       `su - dradmin`
    b. Issue the following command:
       `/opt/vertica/bin/adminTools`
       The Administration Tools dialog opens.
    c. Select **(1) View Database Cluster State**, and then select **OK** or press the return key on your keyboard.
       The database name appears and the State is reported as UP.
    d. Select **OK** to acknowledge that the database is UP.
    e. Select **(E) Exit**, and then press the return key on your keyboard.

NOTE
If the database does not start automatically, to avoid the data aggregator installation failure, start the database manually by selecting **Start DB**.

Install the Database with Sudo User Passwordless SSH Configured

To set up the Data Repository, you can install and configure the Vertica database as the sudo user.

NOTE
RHEL 6.x does not support this functionality.

Follow these steps:

1. Log in to *each* node in the data repository cluster as the sudo user.
2. Copy the `installDR.bin` file locally.
3. Change permissions for the installation file by typing the following command:
   `chmod u+x installDR.bin`
4. Extract the installation file as the sudo user by issuing the following command:
   `sudo ./installDR.bin`

   NOTE
   The `installDR.bin` file extracts the Data Repository rpm, the license file, and the three installation scripts. You install the Data Repository later in this procedure.
5. Follow the instructions in the console.
6. When prompted, specify the installation directory to extract the Data Repository installation package and Vertica license file to. When you are installing the Data Repository using the sudo user account with passwordless SSH, after

extracting, you must run the installation on *each* host in the cluster using the same location. The default installation directory is `/opt/CA/IMDataRepository_vertica*Version*/` . Press Enter twice.

NOTE

The script generates WARN messages for any LVM present in the environment. For help, contact CA Support.

The Data Repository installation package, license file, and associated setup scripts are extracted to the chosen directory.

7. Adjust the following parameters in the `drinstall.properties` file to reflect your installation-specific values. This file applies to the `dr_validate.sh` and `dr_install.sh` scripts. The `drinstall.properties` file exists in the installation directory that you specified previously.
   - DbAdminLinuxUser
     The Linux user that is created to serve as the Vertica database administrator.
     Default: dradmin
   - DbAdminLinuxUserHome=*The Vertica Linux database administrator user home directory*
     Default: /export/dradmin

     NOTE
     This directory is created if the Vertica installer creates the user. Be sure that the directory leading up to the home account already exists on the system. For example, if you are using /export/dradmin, be sure that /export exists.
   - DbDataDir=*The location of the data directory*
     Default: /data

     NOTE
     Do not use the Logical Volume Manager (LVM) for the data directory.
   - DbCatalogDir=*The location of the catalog directory*
     Default: /catalog

     NOTE
     Do not use the Logical Volume Manager (LVM) for the catalog directory.
   - DbHostNames=*The list of hostnames for Data Repository*
     Default: yourhostname1,yourhostname2,yourhostname3

     NOTE
     For this step, list the local hostname only. You add all other nodes in a later step.
   - DbName=*The database name*
     Default: drdata

     NOTE

     This parameter is case-sensitive.
   - DbPwd=*The database password*
     Default: dbpass

     NOTE
     The database password that you define here is used during the installation of the data aggregator. You can use special characters (except for single quotation marks) in passwords. To use special characters, encase the password with single quotation marks (for example, `DbPwd='test$tr|ng'`). If the `DbPwd` property is not found or blank, the script prompts for this information at runtime.
8. Run the validation script with the `-sp` command line argument on *each* node:

```
sudo ./dr_validate.sh -sp properties_file
```

TIP
You can use the `-l` flag to allow `localhost` as the value for the `DbHostNames` property. You can use the `-n` flag to skip database connectivity checks.

9. Review any on-screen output for failures or warnings. You can run this script multiple times after you fix any failures or warnings. The script automatically corrects many failures or warnings. Proceed only if the final status is “PASSED”. If the final status is not "PASSED", contact Support.
   The validation script may ask you to reboot.

   NOTE
   The validation script and the installation script generate a log file in *installation_directory*`/logs` on the Data Repository host from which you run the scripts. These log files include the step-by-step output of the scripts. To validate successful/failed script runs, review the script output.

   The following example shows the script output and lists what settings the script verifies and changes:

```
Log File: logs/install_log_validate_10-29-2015_11-14-11.log
================================================================================
Checking Passwordless SSH to all hosts: verticahost-dr
================================================================================
Passwordless SSH from verticahost-dr to root@verticahost-dr ..................[ OK ]
================================================================================
Beginning Data Repository Prerequisite Compliance Enforcement on host verticahost-dr
================================================================================
Red Hat Enterprise Linux Major Release:  6 ..............................[ OK ]
Processor Type: Intel ...................................................[ OK ]
CPU frequency scaling not available on this system ......................[ OK ]
DR Administrative User dradmin does not exist.  It will be created during vertica
 installation. [ OK ]
Maximum number of file handles >= 65536 .................................[ OK ]
Detected incorrect maximum number of memory maps ........................[WARN]
Set maximum number of memory maps to Total Mem(KB)/16 ...................[ OK ]
Detected incorrect page reclaim threshold value .........................[WARN]
Set page reclaim threshold value to 7924 ................................[ OK ]
Disabling necessary firewall settings. ..................................[ OK ]
Enabling NTP daemon. ....................................................[ OK ]
Starting the NTP daemon. ................................................[ OK ]
Detected incorrect readahead parameter for /dev/sda .....................[WARN]
Set readahead parameter for /dev/sda to 2048 ............................[ OK ]
Block Size for /dev/sda is 4096 .........................................[ OK ]
Readahead parameter for /dev/sda1 is 2048 ...............................[ OK ]
Block Size for /dev/sda1 is 1024.  Expected value >= 4096 ...............[WARN]
Readahead parameter for /dev/sda2 is 2048 ...............................[ OK ]
Block Size for /dev/sda2 is 4096 .........................................[ OK ]
Readahead parameter for /dev/sda3 is 2048 ...............................[ OK ]
Block Size for /dev/sda3 is 4096 .........................................[ OK ]
Detected incorrect swappiness setting ...................................[WARN]
Set swappiness to 0 .....................................................[ OK ]
Transparent hugepages in /sys/kernel/mm/redhat_transparent_hugepage/enabled are
 enabled [WARN]
```

```
Disabled Huge Page Compaction ..........................................[ OK ]
Huge Page Compaction Defrag in /sys/kernel/mm/redhat_transparent_hugepage/defrag is
 enabled [WARN]
Disabled Huge Page Compaction Defrag ...................................[ OK ]
Disk Scheduler for sda is not deadline .................................[WARN]
Set Disk Scheduler for sda to deadline .................................[ OK ]
Reloading sysctl.conf ..................................................[WARN]
SELinux is disabled ....................................................[ OK ]
Verifying Swap Space. ..................................................[ OK ]
No Logical Volumes exist. ..............................................[ OK ]
Root entry exists in /etc/sudoers file. ................................[ OK ]
Verifying ext3 or ext4 filesystem used for data directory. ..............[ OK ]
Verifying ext3 or ext4 filesystem used for catalog directory. ...........[ OK ]
Fresh install of Vertica is being performed - skipping database connectivity testing.


Data Repository Prerequisite Compliance Status on host verticahost-dr -- PASSED
================================================================================
Script finished - /user/home/verticahost/dr_validate.sh
================================================================================
```

NOTE
If the installation fails early enough in the process, the log file might be available in the home directory of the root or sudo user.

10. Repeat the previous steps for *each* node.
11. Go to the first node and edit the `DbHostnames` parameter in the `drinstall.properties` file to include *all* the nodes in the cluster.
12. Run the installation script with the `-sp` command line argument:
    ```
    sudo ./dr_install.sh -sp properties_file
    ```
    NOTE
    To run the script as sudo, passwordless SSH (public key) must be set up for the sudo account between the Data Repository hosts. If passwordless SSH does not exist for the sudo account, you cannot proceed. For more information, see Prepare to Install the Data Repository.

    This script installs the Data Repository, creates the database, and disables unnecessary Vertica processes on all the hosts in the cluster.

    NOTE
    If the database administrator user does not already exist, the installation script creates the user. The script prompts you to assign a new password. If the database administrator user exists, but passwordless SSH is not set up, the script prompts for the password to set up.

    If the installation script returns a WARN message for LVM on directories that Vertica does not use, contact Support.
13. Verify that Data Repository has been installed successfully by doing the following steps:
    a. To log in to the database server as the database administrator user, issue the following command:
       ```
       su - dradmin
       ```
    b. Issue the following command:
       ```
       /opt/vertica/bin/adminTools
       ```
       The Administration Tools dialog opens.
    c. Select **(1) View Database Cluster State**, and then select **OK** or press Enter.
       The database name appears and the State is reported as UP.

d. Select **OK** to acknowledge that the database is UP.
e. Select **(E) Exit**, and then press Enter.

NOTE
If the database does not start automatically, select Start DB to start the database manually. If the database is not started, the Data Aggregator installation fails.

(Optional) Secure Data Repository

To limit the users who can log in to the database to only the Data Repository administrative account and the root user, lock down the database.

Follow these steps:

1. Modify the `/etc/pam.d/sshd` file by adding the following entry, for the PAM access module, after the "account required pam_nologin.so" entry:
   ```
   account required pam_access.so accessfile=/etc/security/sshd.conf
   ```
   NOTE
   If the `/etc/security/sshd.conf` file is missing, you must create it using the SSHD documentation.
2. If the following line from the `/etc/security/access.conf` file exists, remove it:
   ```
   -:ALL EXCEPT database_admin_user root:LOCAL
   ```
   For example:
   ```
   -:ALL EXCEPT dradmin root:LOCAL
   ```

Configure Log Rotation for Data Repository

To prevent the underlying Data Repository log file (`vertica.log`) from becoming too large, configure log rotation for Data Repository. The recommended configuration for the log rotation is a daily rotation with logs retained for 21 days.

WARNING
Configuring the log rotation is required. The `vertica.log` file can grow substantially.

Follow these steps:

1. Log in to the database server for Data Repository as the database administrator user by issuing the following command:
   ```
   su - dradmin
   ```
2. Issue the following command:
   ```
   /opt/vertica/bin/admintools -t logrotate -d database_name -r frequency -k number
   ```
   – **-d** indicates the database name.

   NOTE
   This parameter is case-sensitive.
   – **-r** specifies how often to rotate the daily logs.
   Values: daily, weekly, monthly
   – **-k** specifies how many logs to keep according to the frequency. For example, if the frequency is weekly, a value of 3 keeps three weeks of daily log files.
   Example:
   ```
   /opt/vertica/bin/admintools -t logrotate -d drdata -r daily -k 14
   ```
3. (Optional) To verify that the vertica.log rotation has been configured correctly, look at the new gzipped vertica.log files in the Vertica catalog directory for previous days. The log files use the following filename format:
   `vertica.log.YYYYMMDD.gz`

Set Up Automatic Backups of Data Repository

To preserve your data against failures, set up automatic backups of the Data Repository.
For more information, see Back Up the Data Repository.

Prepare to Install the Data Aggregator

To ensure that your Data Aggregator installation is successful, complete the requirements before you install Data Aggregator:

Verify the Prerequisites

Meet the following prerequisites before installing Data Aggregator:

- Review Installation Requirements and Considerations.
- Verify that Data Repository installation is complete and the service is running.
- Verify that port numbers 8581, 61616, 61618, 61620, and 61622 are open on the Data Aggregator system. Remote access is required to these ports.

  NOTE
  Throughout the documentation 8182, 8382, 61617, 61619, 61621, and 61623 appear as suggested port numbers for secured communications. In the instances where these ports appear, you are free to use any value you want as long as no other processes are using it. You can change port 616xx to another port after you install the Data Aggregator. For more information, see Complete the Post-Installation Configuration.
- Verify that ports 1099 and 11099 are blocked from external access.
  These ports must remain open locally for internal communication.
- Verify that Security Enhanced Linux (SELinux) is disabled or permissive on the computer where you are going to install Data Aggregator. By default, some Linux distributions have this feature enabled, which does not allow Data Aggregator to function properly. Disable SELinux, set to permissive, or create a policy to exclude Data Aggregator processes from SELinux restrictions. If you would like Security Enhanced Linux (SELinux) to be enforcing, consult the Red Hat documentation.

  NOTE
  For information about configuring an SELinux security policy, see the Red Hat documentation.
- To avoid potential corruption of data, exclude the installation directory, the backup directory, and all subdirectories, from antivirus scans. Prevent scanning by a local instance of an antivirus client and scanning by a remote antivirus instance. For more information about Data Aggregator backups, see Back Up Data Aggregator.
- Verify that the directory where you are going to install has write privileges for your Data Aggregator user.
- The installer requires the zip and unzip packages. If these packages are not installed, use one of the following commands to install them:

  ```
  yum -y install zip unzip
  ```

  SLES:

  ```
  zypper -y install zip unzip
  ```
- For a fault tolerant environment, verify that proxy server is installed. For more information, see Install or Uninstall the Proxy Server.

(Optional) Configure the Sudo User Account for Data Aggregator

If you do not have root access to install and run the Data Aggregator, configure the sudo user account.

Follow these steps:

1. Locate the following file on the Data Aggregator host:

```
/etc/sudoers
```

2. Add one of the following command aliases with the following permissions to the file:

```
Cmnd_Alias CA_DATAAGG = /tmp/installDA.bin,/sbin/service dadaemon *,/opt/
IMDataAggregator/Uninstall/Uninstall, /opt/IMDataAggregator/RemoteEngineer/re.sh
## Allows the Data Aggregator user to manage the Data Aggregator
sudouser ALL = CA_DATAAGG
```

   – ***sudouser*** Specify the user who can run the sudo commands.

   This command alias details the commands that the sudo user must be able to run.

   SLES:

```
Cmnd_Alias CA_DATAAGG = /tmp/installDA.bin,/usr/sbin/service dadaemon *,/usr/sbin/
service activemq *,/opt/IMDataAggregator/Uninstall/Uninstall, /opt/IMDataAggregator/
RemoteEngineer/re.sh
## Allows the Data Aggregator user to manage the Data Aggregator
dasudouser_name ALL = CA_DATAAGG
```

   – ***sudouser*** Specify the user who can run the sudo commands.

   This command alias details the commands that the sudo user must be able to run.

3. With the sudo user configured, add the sudo prefix to all commands to install the Data Aggregator.

   Example:

```
sudo ./installDA.bin
```

Configure the Limit on the Number of Open Files on Data Aggregator

Verify that the user that is installing Data Aggregator has a limit of at least 65536 on the number of open files. Set this value permanently.

Follow these steps:

1. As the root user or a sudo user, log in to the Data Aggregator host.
2. Change the ulimit for the open files limit to at least 65536:

```
ulimit -n ulimit_number
```

   For example:

```
ulimit -n 65536
```

3. Open the following file:

```
/etc/security/limits.conf
```

4. Add the following lines:

```
# Added by Data Aggregator
* soft nofile 65536
# Added by Data Aggregator
* hard nofile 65536
```

   NOTE

   Restart Data Aggregator for these changes to take effect. If you are upgrading, the upgrade process automatically restarts Data Aggregator.

5. Verify that the number of open files is set properly:

```
ulimit -n
```

   The command returns the limit that you specified earlier.

Configure UTF-8 Support

Configure the Data Aggregator host to support UTF-8 encoding. If UTF-8 encoding is not enabled, characters might not display properly during the installation.

The appropriate language packs are also required to support localized deployments.

NOTE
Some scripts that are used in the installation of selected components are not localized and run in English. For more information, see Language Support.

Follow these steps:

1. Do one of the following steps:
   a. Type the following command from a Korn or bash shell:
      ```
      export LANG=LANG_value ; export LC_ALL=$LANG
      ```
      - **LANG_value** specifies the value of the language you want the product to support. The following variables are supported:
        English: `en_US.utf8`
        French: `fr_FR.utf8`
        Japanese: `ja_JP.utf8`
        Simplified Chinese: `zh_CN.utf8`
        Traditional Chinese: `zh_TW.utf8`

      For example:
      ```
      export LANG=zh_TW.utf8 ; export LC_ALL=$LANG
      ```
   b. Type the following command from a Bourne shell:
      ```
      LANG=LANG_value ; export LANG
      LC_ALL=LANG_value ; export LC_ALL
      ```
      For example:
      ```
      LANG=zh_CN ; export LANG
      LC_ALL=zh_CN ; export LC_ALL
      ```
   The language variable is set.

Install the Data Aggregator

Once you have met the requirements to install Data Aggregator, complete the installation as follows:

Install the Data Aggregator

To install a Data Aggregator in silent mode, run the installation and create a response file. Use the response file to run future installations in silent mode. To generate the response file, follow the installation procedure, and add the response file argument when you run the installDA.bin file.

TIP

To install the Data Aggregator, the Data Repository must be running. To verify the status of the Data Repository, run the following command on the Data Repository host:

```
su - dr_admin_user -c "/opt/vertica/bin/admintools -t show_active_db"
```

Follow these steps:

1. Log in to the Data Aggregator host as the root user or the sudo user.

2. Copy the installDA.bin file to the /tmp directory.
3. Change to the /tmp directory:

```
cd /tmp
```

4. Change permissions for the installation file:

```
chmod a+x installDA.bin
```

5. To run the console installation, do one of the following steps:
   - To run the installation as the root user, type the following command:

     ```
     ./installDA.bin -i console
     ```

   - To run the installation as the sudo user, type the following command

     ```
     sudo ./installDA.bin -i console
     ```

     NOTE

     To generate a response file for silent installation, add the following argument:

     ```
     -r response_file
     ```

     response_file specifies the directory the directory path and file name for the response file.
     Example: /temp/installer.properties

     Follow the prompts until you get to the summary, type "quit", and press Enter.

     To run the installation in silent mode, use the following command:

     ```
     ./installDA.bin -i silent -f response_file
     ```

     response_file is the directory path and file name of the previously generated response file.
6. Follow the instructions in the console.

   WARNING

   When you are prompted for the data directory, use the default directory. Do not use *DA_Install_Directory*/apache-karaf-*version*/data .
7. When prompted, specify the following parameters for Data Repository:
   - **Data Repository server hostname/IP**
     Defines either a name or an IP address for the Data Repository server host.
     For a Data Repository cluster, specify the name or the IP address of any one host in the cluster. The installer automatically determines the name and IP address of the remaining nodes.
   - **Data Repository server port**
     Defines the port number for the Data Repository server.

Default: 5433

- **Database name**
  Defines the database name of Data Repository.
- **Data Repository username**
  Specifies the username that Data Aggregator uses to connect to the database. The username and the password cannot match. This username and password combination is added to the database during the installation.
  Example: dauser
- **Data Repository admin username**
  Specify the Linux user account that was used to install Data Repository. This username is needed for administration, such as backing up and restoring Data Repository, or updating the database schema.
  Example: dradmin
- **Data Repository admin password**
  Defines the password for the Data Repository admin username.

  NOTE
  This database user account password was specified when you created the database after the Data Repository installation.

  Example: dbpassword

  TIP

  If necessary, a `doEncryption.sh` script is available to edit the `dbconnection.cfg` file. For example, you can use the script to change your Data Repository admin password.

8. (Optional) When prompted, specify the following parameters for fault tolerance. For more information, see Fault Tolerance.
   - **Configure Data Aggregator For Fault Tolerance**
     Specify 2 to configure fault tolerance.
     Default: 1

     NOTE

     The default is for a non-fault tolerant environment.
   - **Data Aggregator Proxy Host**
     Specify the host name/IP address of the proxy server.
   - **Consul HTTP port:**
     Specify the port for communication with Consul.
     Default: 8500
   - **Choose host IP address for Consul**

     NOTE

     This prompt appears only when multiple public IP addresses are configured.

     Specify the bind address that the Consul agents use to communicate with each other. The Consul agents include the proxy host and both Data Aggregators in the cluster. If prompted for an address, specify an address that the other two hosts in the Consul cluster can reach.
9. The Data Aggregator is installed and started.

Verify the Data Aggregator Installation

NOTE

For the Data Aggregator installations, we use the static location of /etc/DA.cfg to store where the Data Aggregators are installed and the chosen installation options for upgrade.

Follow these steps:

1. Verify that the Data Aggregator service is running:

   ```
   service dadaemon status
   ```

2. Review the following log file on the Data Aggregator host:

   ```
   /opt/IMDataAggregator/Logs/
   CA_Performance_Management_Data_Aggregator_Install_timestamp.log
   ```

   If the installation is successful, the log shows 0 Warnings, 0 NonFatalErrors, and 0 FatalErrors.

   NOTE

   If the installation fails early enough in the process, the log file may be available in the home directory of the root or sudo user.

3. Verify that the ActiveMQ broker is running:

   ```
   service activemq status
   ```

4. Open a browser on a computer where you have HTTP access to Data Aggregator. Navigate to the following address:

   ```
   http://da_host:8581/rest
   ```

   da_host specifies the Data Aggregator host name or IP address.
   The return is a list of hyperlinks for available web services. When you click a link, the XML content describing the selection displays.

Register Data Aggregator as a Data Source

To connect NetOps Portal to the Data Aggregator, register Data Aggregator as a data source.

Follow these steps:

1. Log in to NetOps Portal as an administrator user. Access NetOps Portal at the following URL:

   ```
   PC_host:8181/pc/desktop/page
   ```

2. Navigate to **Administration**, **Data Sources**.
3. Click **Add**.
4. Select Data Aggregator for the Source Type, specify the required information, and click **Save**.

   NOTE

   For a fault tolerant environment, specify the proxy server host name. For more information, see Fault Tolerance.

   Wait a few minutes for Data Aggregator to synchronize automatically with NetOps Portal.

Prepare to Install the Data Collectors

To ensure that your Data Collector installation is successful, complete the requirements before you install a Data Collector:

Review Data Collector Considerations

In a standard tenant deployment, each tenant has a dedicated Data Collector. For multiple tenants that reside in the same IP routing space, DX NetOps Performance Management can be configured to use fewer Data Collectors. For more information, see Tenant-Agnostic Data Collectors.

The following considerations apply to a standard tenant deployment:

- Review Installation Requirements and Considerations.
- You can install more than one Data Collector. Each Data Collector must be installed on a separate host.
- In a standard tenant deployment, a Data Collector supports only one Data Aggregator.
- In a multi-tenant environment where a managed service provider is monitoring devices for multiple tenants, you can take the following steps:
  - Install Data Collector at the MSP site.

    NOTE
    This setup requires Data Collector to gain access through a tenant firewall to poll the devices that are being managed.
  - Install Data Collector at each tenant site.
- If Data Aggregator is IPv6 only, Data Collector must support the IPv6 protocol.
  To verify that Data Collector supports IPv6, take the following steps:
  - On the Data Aggregator host, type the following command to find the IPv6 address of the computer:
    ```
    > ifconfig
    ```
  - On the Data Collector host, type the following command to ensure that Data Collector can contact Data Aggregator using its IPv6 address:
    ```
    > ping6 ipv6_address_of_Data Aggregator
    ```

Verify the Prerequisites

Meet the following prerequisites before you install Data Collector:

- On the Data Aggregator system, verify that ports 61616, 61618, 61620, and 61622 are open. These ports enable communication between the Data Collector and the Data Aggregator.
- Verify that ports 1099 and 11099 are blocked from external access.
  These ports must remain open locally for internal communication.
- Verify that Security Enhanced Linux (SELinux) is disabled on the computer where you are going to install Data Collector. By default, some Linux distributions enable this feature, which does not allow Data Collector to function properly. Disable SELinux or create a policy to exclude Data Collector processes from SELinux restrictions.

  NOTE
  For information about configuring an SELinux security policy, see the Red Hat documentation.
- To avoid database corruption of ActiveMQ broker files, exclude the installation directory, and all its subdirectories, from antivirus scans. Prevent scanning by a local instance of an antivirus client and scanning by a remote antivirus instance.
- Ensure that your desired tenant and corresponding IP domain are provisioned in NetOps Portal. While a single IP domain can be associated with more than one Data Collector, each Data Collector can have only one IP domain assigned to it.

  NOTE
  If you are not deploying multi-tenancy, use the Default Tenant and the Default Domain.
- The installer requires the zip and unzip packages. If these packages are not installed, use one of the following commands to install them:

```
yum -y install zip unzip
```

SLES:

```
zypper install -y zip unzip
```

- The Data Collector uses the "at" package to schedule the restart of the application when assigning a Tenant or IP Domain. Verify whether the "at" package is installed on each Data Collector host:

  ```
  rpm -qa | grep ^at
  ```

  If the command does not return a result, install the package:

  NOTE

  If you are not the root user, use the sudo prefix.

  ```
  yum install at
  zypper install -y at
  ```

(Optional) Configure the Sudo User Account for Data Collector

If you do not have root access to install and run the Data Collector, configure the sudo user account.

Follow these steps:

1. Locate the following file on the Data Collector host:
   `/etc/sudoers`
2. Add one of the following command aliases with the following permissions to the file:

   ```
   Cmnd_Alias CA_DATACOLL =  /tmp/install.bin,/sbin/service dcmd *, /opt/
   IMDataCollector/Uninstall/Uninstall, /opt/IMDataCollector/RemoteEngineer/re.sh, /opt/
   IMDataCollector/scripts/dcmd
   ## Allows the <caimdc> user to manage the <caimdc>
   sudouser ALL = CA_DATACOLL
   ```

   – ***sudouser***
   Specify the user who can run the sudo commands.

   This command alias details the commands that the sudo user must be able to run.

   SLES:

   ```
   Cmnd_Alias CA_DATACOLL = /tmp/install.bin,/usr/sbin/service dcmd *,/usr/sbin/
   service activemq *,/opt/IMDataCollector/Uninstall/Uninstall, /opt/IMDataCollector/
   RemoteEngineer/re.sh, /opt/IMDataCollector/scripts/dcmd
   ## Allows the <caimdc> user to manage the <caimdc>
   sudouser ALL = CA_DATACOLL
   ```

   – ***sudouser***
   Specify the user who can run the sudo commands.

   This command alias details the commands that the sudo user must be able to run.
3. With the sudo user configured, add the sudo prefix to all commands to install the Data Collector.
   Example:

   ```
   sudo ./install.bin
   ```

Configure UTF-8 Support

Configure the Data Collector host to support UTF-8 encoding. If UTF-8 encoding is not enabled, characters might not display properly during the installation.

The appropriate language packs are also required to support localized deployments.

NOTE
Some scripts that are used in the installation of selected components are not localized and run in English. For more information, see the *Localization Status Readme* file.

Follow these steps:

1. Do one of the following steps:
   a. Type the following command from a Korn or bash shell:
      ```
      export LANG=LANG_value ; export LC_ALL=$LANG
      ```
      - **LANG_value** specifies the value of the language you want the product to support. The following variables are supported:
        English: `en_US.utf8`
        French: `fr_FR.utf8`
        Japanese: `ja_JP.utf8`
        Simplified Chinese: `zh_CN.utf8`
        Traditional Chinese: `zh_TW.utf8`

      For example:
      ```
      export LANG=zh_TW.utf8 ; export LC_ALL=$LANG
      ```
   b. Type the following command from a Bourne shell:
      ```
      LANG=LANG_value ; export LANG
      LC_ALL=LANG_value ; export LC_ALL
      ```
      For example:
      ```
      LANG=zh_CN ; export LANG
      LC_ALL=zh_CN ; export LC_ALL
      ```
   The language variable is set.

Set a Unique Hostname for the Data Collector Host

Set a unique hostname for the computer where you plan to install Data Collector.

Follow these steps:

1. As the root user, log in to the Data Collector host
2. Verify the unique hostname on the computer.
   The hostname for the computer must be associated with the IP address and *not* the loopback address of 127.0.0.1.
3. Verify that the following lines appear in the /etc/hosts file on the computer:
   ```
   Do not remove the following line, or various programs
   # that require network functionality will fail.
   127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
   ::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
   IP address of your host YourHostName YourHostName.ca.com
   ```
4. If the hostname required any changes, restart the network:
   ```
   service network restart
   ```
   The /etc/hosts file is configured correctly.
   The unique host name is set.

These ports must remain open locally for communication between the Data Aggregator and Data Collectors.

Install the Data Collectors

Install Data Collectors after you install Data Aggregator.

To install a Data Collector in silent mode, run the installation and create a response file. Use the response file to run future installations in silent mode. To generate the response file, follow the installation procedure, and add the response file argument when you run the install.bin file.

If you are installing multiple Data Collectors, install each Data Collector instance on a separate host.

If you are reinstalling a Data Collector, see Update the Data Collector.

Follow these steps:

1. Log in to the Data Collector host either as the root user or the sudo user.
2. Access the Data Collector installation package by doing one of the following actions:
   - If you have HTTP access to the Data Aggregator host *and* you are running an X Window System, open a web browser on the Data Collector host. Navigate to the following address and download the installation package:
     `http://data_aggregator:port/dcm/install.htm`
     - ***data_aggregator*:*port***
       Specifies the Data Aggregator host name and the required port number.
       Default: 8581, unless you specified a nondefault value during the Data Aggregator installation.

     Save the installation package to the /tmp directory.
   - If you have HTTP access to the Data Aggregator host and you are *not* running an X Window System, open a command prompt on the Data Collector host. Download the installation package to the /tmp directory using one of the following commands:
     - `wget`
       is available:
       `wget -P /tmp -nv http://data_aggregator:port/dcm/InstData/Linux/VM/install.bin`
       - ***data_aggregator*:*port***
         Specifies the Data Aggregator host name and the required port number.
         Default: 8581, unless you specified a nondefault value during the Data Aggregator installation.
     - `wget`
       is unavailable:
       `curl -o /tmp/install.bin http://data_aggregator:port/dcm/InstData/Linux/VM/install.bin`
   - If you do *not* have HTTP access to the Data Aggregator host, open a command prompt on a computer that *does* have HTTP access.
     a. Download the installation package to your Desktop directory using one of the following commands:
        - `wget`
          is available:
          `wget -P ~/Desktop -nv http://data_aggregator:port/dcm/InstData/Linux/VM/install.bin`
          - ***data_aggregator*:*port***
            Specifies the Data Aggregator host name and the required port number.
            Default: 8581, unless you specified a nondefault value during the Data Aggregator installation.
        - `wget`
          is unavailable:
          `curl -o /tmp/install.bin http://data_aggregator:port/dcm/InstData/Linux/VM/install.bin`

     b. Transfer the install.bin file to /tmp on the Data Collector host.
3. Change to the /tmp directory:
   `cd /tmp`
4. Change the permissions for the installation file:

```
chmod a+x install.bin
```

5. To run the console installation, do one of the following steps:
   - To run the installation as the root user, type the following command:
     ```
     ./install.bin -i console
     ```
   - To run the installation as the sudo user, type the following command
     ```
     sudo ./install.bin -i console
     ```

     NOTE

     To generate a response file for silent installation, add the following argument:

     ```
     -r response_file
     ```

     response_file specifies the directory the directory path and file name for the response file.
     Example: /temp/installer.properties

     Follow the prompts until you get to the summary, type "quit", and press Enter.

     To run the installation in silent mode, use the following command:

     ```
     ./install.bin -i silent -f response_file
     ```

     response_file is the directory path and file name of the previously generated response file.
6. Follow the instructions in the console.
7. When the installer prompts you for the Data Aggregator host information, specify the IP address or the hostname for Data Aggregator.

   WARNING
   Specify the Data Aggregator host information correctly. If you specify the Data Aggregator host information incorrectly, the Data Collector shuts down after installation. An error message is logged in the *DC_installation_directory*/apache-karaf-<*vers*>/shutdown.log file. Uninstall and reinstall Data Collector.
8. When you are asked whether to associate this Data Collector with the Default Tenant, enter 'y' or 'n'.
   - If you are planning to deploy multi-tenancy, enter 'n'. You can then associate each Data Collector installation with a tenant.
   - If you are not deploying multi-tenancy, enter 'y'.
9. (Optional) When prompted, specify the following parameters for fault tolerance. For more information, see Fault Tolerance.
   - **Is the Data Aggregator configured with fault tolerance?**
     If fault tolerance was configured for the Data Aggregators, specify 2 for Yes.
     Default: 1
   - **Inactive Data Aggregator Host/IP Address**
     Specify the host name or IP address of the inactive Data Aggregator responsible for this Data Collector.

     NOTE
     To view the status of your Data Aggregators in NetOps Portal, hover over **Administration**, and click **Data Sources: System Status**.

   Data Collector is installed, started, and connects to Data Aggregator.

   NOTE
   If you restart the Data Collector host, the Data Collector service automatically restarts and connects to Data Aggregator.

Verify the Installation

NOTE

For the Data Collector installations, we use the static location of /opt/DCM.cfg to store where the Data Collectors are installed and the chosen installation options for upgrade.

Follow these steps:

1. Verify that Data Collector is running on the Data Collector host:
   ```
   service dcmd status
   ```
2. Review the the following log file on the Data Collector host:
   ```
   /opt/IMDataCollector/Logs/
   CA_Performance_Management_Data_Collector_install_timestamp.log
   ```
   If the installation is successful, the log shows 0 Warnings, 0 NonFatalErrors, and 0 FatalErrors.

   NOTE

   If the installation fails early enough in the process, the log file may be available in the home directory of the root or sudo user.
3. Verify that the Data Collector connection is successful after the installation by doing the following actions:
   a. Log in to NetOps Portal as the global administrator.
   b. Navigate to the Data Aggregator administration view and expand the System Status view.
   c. Select Data Collectors from the menu.
   d. Verify that Data Collector appears in the list.

   NOTE
   The list can take several minutes to refresh and show the new Data Collector installation.
4. Assign a tenant and IP domain to each Data Collector if the Tenant and IP Domain are blank:
   a. Select the Data Collector instance and click Assign.
   b. Select another tenant and an IP domain for this Data Collector in the Assign Data Collector dialog and click Save.

Complete the Post-Installation Configuration

Perform the following optional and recommended steps after you install DX NetOps Performance Management:

Set Up Autostart on Data Repository

You can set up autostart on Data Repository. If autostart is set up and you reboot the computer where Data Repository is installed, Data Repository starts automatically.

WARNING
This feature might not work if Data Repository did not shut down properly. If the database did not shut down properly, the database might require manual intervention during startup to restore the last good epoch. If the Vertica database does not start automatically after an improper shutdown, use admintools to start it manually.

Data Aggregator stops automatically when Data Repository becomes inaccessible. Restart Data Aggregator manually once Data Repository is online again.

Do one of the following steps:

- Start the Data Aggregator service:
  ```
  service dadaemon start
  ```

NOTE

For RHEL 7.x or OL, `service` invokes `systemctl` . You can use `systemctl` instead.

- (Fault tolerant environment) Run one the following commands to enable the fault tolerant Data Aggregator so that it can start when necessary:
  - **RHEL 6.x:**
    `service dadaemon activate`
  - **RHEL 7.x, SLES, or OL:**
    DA_Install_Directory`/scripts/dadaemon activate`

Follow these steps:

1. To become the Linux user account for the database administrator user, type the following command:
   `su` *dradmin*
2. Verify that the Linux user account for the database administrator user is set up with a passwordless ssh key:
   a. To see if a passwordless ssh key is already set up, type the following command:
      `ssh` *dr_host* `ls`
      If the passwordless ssh key is set up, you are *not* prompted for a password. You do not need to do anything further.
   b. If you *are* prompted for a password, ignore the prompt, press Ctrl+C, and complete step 5.
3. (Optional) Set up the Linux user account for the database administrator user with a passwordless ssh key.
   a. To generate a public key, type the following command. In a cluster installation, type this command on each host that is participating in the cluster:
      `ssh-keygen -N "" -t rsa -f ~/.ssh/id_rsa`
   b. Copy the contents of the public key to the authorized_keys2 file on the same computer. In a cluster installation, copy the contents of the public key to the authorized_keys2 file on each host in the cluster:
      `cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys2`
   c. (Cluster installation only) Copy the contents of the public key from each host to each of the other hosts:
      - As the database administrator user on the first host, type the following command and copy the content of the file:
        ```
        vi ~/.ssh/id_rsa.pub
        vi ~/.ssh/authorized_keys2
        vi ~/.ssh/authorized_keys2
        ```
      - As the database administrator user on the second host, type the following command:

      Paste the contents from the id_rsa.pub file on the first host to the end of the authorized_keys2 file on the second host.
   d. As the database administrator user on the third host, type the following command:
      `vi ~/.ssh/authorized_keys2`
      Paste the contents from the id_rsa.pub file on the first host to the end of the authorized_keys2 file on the third host.
4. To enable you to ssh from one host to another without being prompted for a password, repeat these steps for all hosts in the cluster:
   a. To set permissions for the authorized_keys2 file, type the following command. In a cluster environment, type these commands on each host in the cluster:
      `chmod 644 ~/.ssh/authorized_keys2`
   b. As the root user, type the following commands to restart the ssh daemon. In a cluster environment, type this command on each host in the cluster:
      ```
      su root
      service sshd restart
      ```

c. (Single-node installations only) To confirm that you are not prompted for a password, type the following commands:
```
su dradmin
ssh dradmin@hostname ls /tmp
```
d. (Cluster installations only) To confirm that you are not prompted for a password, type the following commands on the first host in the cluster:
```
su dradmin
ssh dradmin@host1 ls /tmp
ssh dradmin@host2 ls /tmp
ssh dradmin@host3 ls /tmp
```
Repeat this step on each host in the cluster.

WARNING
If you do not set up the passwordless ssh key, you cannot configure autostart on Data Repository.

5. Type the following command:
```
/opt/vertica/bin/adminTools
```
The Administration Tools dialog opens.
6. Select (6) Configuration Menu and select OK.
7. Select (4) Set Restart Policy and select OK.
The Select Database dialog opens.
8. Select the database name and select OK.
The Select policy dialog opens.
9. Select 'always' when doing a single-node Data Repository installation. Select 'ksafe' when doing a cluster installation. Select OK.

NOTE
In a single-node installation, ‘always’ means that, Data Repository automatically restarts when the system restarts. In a cluster installation, ‘ksafe’ means that, upon the system restarting, the Data Repository node automatically restarts if the database still has a status of ‘UP’.

The Restart Policy setting is saved.
10. Select OK to close the Select policy dialog.
11. Return to the (M) Main Menu.
12. Select (E) Exit.
13. (Optional) Test that Data Repository starts when you reboot the computer where Data Repository is installed:
a. Reboot the computer where Data Repository is installed.

NOTE
Log in as the root user or sudo user to reboot the computer.

b. Become the Linux user account for the database administrator user. Type the following command:
```
su dradmin
```
c. Type the following command:
```
/opt/vertica/bin/adminTools
```
The Administration Tools dialog opens.
d. Select (1) View Database Cluster State and select OK.
e. The state is "UP." Select OK.

NOTE
Data Repository can take several minutes to start up after you reboot.

Configure the Automatic Recovery for the Data Aggregator Process

Configure the automatic recovery of the Data Aggregator process.

If the database server runs out of memory, or if Data Repository is unavailable for a time, Data Aggregator shuts down automatically to ensure that data consistency is maintained. When Data Aggregator shuts down, an audit message is logged in the following log file:
`DA_installation_directory/apache-karaf-<vers>/shutdown.log`

When the Data Aggregator is unavailable, the Data Collectors continue polling. The Data Collector caches the poll responses in memory, up to a configurable limit. When the Data Aggregator host becomes available, the cached polled data is sent to Data Aggregator.

We recommend that you disable this cron job before you upgrade Data Aggregator. If you shut down Data Aggregator manually with the service dadaemon stop command, the cron job does not restart Data Aggregator automatically. Maintenance can be performed without having the cron job disrupt the system when it is expected to be down.

> **NOTE**
>
> In a fault tolerant environment, this procedure is unnecessary because Consul manages the start and stop state of the Data Aggregator. For more information, see Fault Tolerance.

Follow these steps:

1. Log in to the computer where the Data Aggregator is installed as the root user.
2. Open a console and type the following command:
   ```
   crontab  -e
   ```
   A vi session opens. If there are no cron jobs for the database administrator user, an empty file opens. Otherwise, the file contains existing cron job definitions.
3. Add the following lines to the file for the cron job:
   ```
   * * * * * /sbin/service dadaemon start > /dev/null
   ```
   The cron job issues a start command to Data Aggregator every minute. If Data Aggregator is running, the start command is ignored.

(Optional) Modify the External ActiveMQ Memory Limit

The Data Aggregator installer calculates the memory that is needed on your system to accommodate the Apache ActiveMQ process. However, you can manually modify the memory limit settings to fine tune ActiveMQ on your Data Aggregator system. For example, you can modify the settings under the following circumstances:

- When the system memory has changed.
- When the number of Data Collector systems have changed.
- To optimize the memory settings.
- When you have determined that the performance of ActiveMQ is degraded. Monitor the performance through the JConsole or the DX NetOps Performance Management custom chart with ActiveMQ metrics.

Follow these steps:

1. Calculate the amount of memory for ActiveMQ based on the following settings:
   - **Maximum java heap size**
     Default: 20%
     Minimum: 512M
   - **Initial minimum java heap size**
     50% of maximum java heap size
   - **Young generation java heap size**
     25% of the maximum java heap size
   - **Memory limit for all messages**

50% of the maximum java heap size

– **Memory limit per queue**
Calculate based on how many Data Collector installations you have.
Example: The memory per queue
(system memory for all messages)/5/(Data Collector count)

2. Log in to the computer where Data Aggregator is installed. Log in as the root user or a sudo user with access to a limited set of commands.
3. Type the following command to stop the ActiveMQ broker:
```
service activemq stop
```
4. Modify the java heap size for ActiveMQ:
   a. Access the **activemq** file under *DA_INSTALL_DIRECTORY*/broker/apache-activemq-*version*/bin.
   b. Locate the line that defines ACTIVEMQ_OPTS_MEMORY.
   c. Change - Xms to be the Initial minimum java heap size.
   d. Change - Xmx to be the Maximum java heap size.
   e. Change –Xmn to be the Young generation java heap size.
   f. Save the file.
5. Modify the ActiveMQ memory limit for the producer flow control:
   a. Access the activemq.xml file in *Data Aggregator installation directory*/broker/apache-activemq-*version*/conf.
   b. Locate the following line and change the value to Memory limit for all messages:
```
<memoryUsage limit=”value”/>
```
   c. Locate the following line, change the value to Memory limit per queue:
```
<policyEntry queue=">" producerFlowControl="true" memoryLimit="value"/>
```
6. Type the following command to start the ActiveMQ broker:
```
service activemq start
```
Your new settings are activated.

(Optional) Change the Opened Port Number on the Data Aggregator Host

After you install Data Aggregator, you can change the port that is opened on the Data Aggregator host.

NOTE
You opened port 61616 before you installed Data Aggregator and Data Collector.

Follow these steps:

1. Log in to the computer where Data Aggregator is installed. Log in as the root user or a sudo user with access to a limited set of commands.
2. Do one of the following steps:
   – Stop the Data Aggregator service:
```
service dadaemon stop
```
   – (Fault tolerant environment) If the local Data Aggregator is running, run one the following commands to shut it down and prevent it from restarting until maintenance is complete:
     • **RHEL 6.x:**
```
service dadaemon maintenance
```
     • **RHEL 7.x, SLES, or OL:**
```
DA_Install_Directory/scripts/dadaemon maintenance
```
3. Type the following commands to remove the data directory and the local-jms-broker.xml file from the deploy directory:
```
rm -rf <caimda> installation directory/apache-karaf-<vers>/data
```

```
rm -rf <caimda> installation directory/apache-karaf-<vers>/deploy/local-jms-
broker.xml
```

4. Edit the `activemq.xml` file in the *DA_installation_directory*`/broker/apache-activemq-<`*vers*`>/conf` directory:
   a. Locate the following lines:

```
<transportConnectors>
     <transportConnector name="openwire" uri="tcp://0.0.0.0:61616"/>
     <transportConnector name="PRQ" uri="tcp://0.0.0.0:61618"/>
     <transportConnector name="IREP" uri="tcp://0.0.0.0:61620"/>
     <transportConnector name="blob" uri="tcp://0.0.0.0:61622"/>
   </transportConnectors>
```

   b. Replace 61616, 61618, 61620, 61622 with the ports that you want to use for incoming connections on Data Aggregator.

NOTE

If you have a fault tolerant environment, ensure that both Data Aggregators use the same ActiveMQ ports.

5. Do one of the following steps:
   – Start the Data Aggregator service:

```
service dadaemon start
```

   – (Fault tolerant environment) Run one the following commands to enable the fault tolerant Data Aggregator so that it can start when necessary:
     - **RHEL 6.x:**

```
service dadaemon activate
```

     - **RHEL 7.x, SLES, or OL:**

DA_Install_Directory`/scripts/dadaemon activate`

6. Wait a few minutes, then type the following command to verify that the port change is successful:

```
netstat -a | grep port
```

   – ***port***
   The port number that you specified previously for incoming connections on Data Aggregator.
7. If the port change is successful, Data Aggregator waits for incoming connections on that port. If Data Aggregator is not waiting for incoming connections, type the following command to review the karaf.log file for errors:

```
grep ERROR karaf.log
```

8. Resolve the errors.
9. Log in to the computer where Data Collector is installed. Log in as the root user or a sudo user with access to a limited set of commands.

NOTE
For more information about the sudo user, see the *Data Aggregator Installation Guide*.

10. Open a command prompt and type the following command:

```
service dcmd stop
```

11. Type the following commands to remove the data directory and the local-jms-broker.xml file from the deploy directory:

```
rm -rf <caimda> installation directory/apache-karaf-<vers>/data
rm -rf <caimda> installation directory/apache-karaf-<vers>/deploy/local-jms-
broker.xml
```

12. Edit the `activemq.xml` file in the *DC_installation_directory*`/broker/apache-activemq-<`*vers*`>/conf` directory:
    a. Locate the following lines:
    ```
    <networkConnector name="da_manager" uri="static:(tcp://scalematda:61616)"
     duplex="true"
         suppressDuplicateTopicSubscriptions="false">
    <networkConnector name="da_manager-PRQ" uri="static:(tcp://scalematda:61618)"
     duplex="true"
         suppressDuplicateTopicSubscriptions="false">
    <networkConnector name="da_manager-IREP" uri="static:(tcp://scalematda:61620)"
     duplex="true"
         suppressDuplicateTopicSubscriptions="false">
    <networkConnector name="da_manager-blob" uri="static:(tcp://scalematda:61622)"
     duplex="true"
         suppressDuplicateTopicSubscriptions="false">
    ```
    b. Replace 61616, 61618, 61620, and 61622 with the ports that you specified previously in the `activemq.xml` file on the Data Aggregator host.
13. Open a command prompt and type the following command to start Data Collector:
    ```
    service dcmd start
    ```
14. Wait a few minutes, then type the following command to verify that each port change is successful:
    ```
    netstat -a | grep port
    ```
    – ***port***
      The port number that you specified in a previous step for incoming connections on Data Aggregator.

    If the port change is successful, the console shows a connection between the Data Aggregator and the Data Collector. If you do not see a connection, type the following command to review the karaf.log file for errors:
    ```
    grep ERROR karaf.log
    ```
15. Resolve the errors.
    The opened port numbers on the Data Aggregator host is changed.

(Optional) Disable the ActiveMQ Admin Console for the Data Aggregator or Data Collector

Generally, the ActiveMQ admin console should not be available on the network. Therefore, you can disable it for the Data Aggregator or Data Collector.

Follow these steps:

1. Go to one of the following files:
   – **Data Aggregator**
     DA_Install_Directory`/broker/apache-activemq-`*version*`/conf/activemq.xml`
   – **Data Collector**
     DC_Install_Directory`/broker/apache-activemq-`*version*`/conf/activemq.xml`
2. Comment out
   ```
   <import resource="jetty.xml"/>
   ```
   .
3. Shut down the ActiveMQ broker on each Data Collector:
   ```
   service activemq stop
   ```
4. Shut down the ActiveMQ broker on the Data Aggregator:
   ```
   service activemq stop
   ```
5. Start the ActiveMQ broker on the Data Aggregator:

```
service activemq start
```

If you do not, the Data Aggregator starts the broker automatically.

6. The Data Collectors automatically restart the ActiveMQ brokers. Use the following command to restart the brokers manually:

```
service activemq start
```

(Optional) Update ActiveMQ Admin Console Access

Generally, the ActiveMQ admin console should not be available on the network. However, if certain users absolutely need the console, you can grant them access.

Follow these steps:

1. Go to one of the following files:
   - **Data Aggregator**
     DA_Install_Directory/broker/apache-activemq-*version*/conf/activemq.xml
   - **Data Collector**
     DC_Install_Directory/broker/apache-activemq-*version*/conf/activemq.xml
2. To update user access, edit the
   `jetty-realm.properties`
   .
3. To encrypt the user passwords, run one of the following commands:
   - **Data Aggregator**

     ```
     java -cp DA_Install_Directory/broker/apache-activemq-version/lib/web/jetty-all-9.2.22.v20170606.jar org.eclipse.jetty.util.security.Password passwordpassword
     ```

   - **Data Collector**

     ```
     java -cp DC_Install_Directory/broker/apache-activemq-version/lib/web/jetty-all-9.2.22.v20170606.jar org.eclipse.jetty.util.security.Password passwordpassword
     ```

4. Shut down the ActiveMQ broker on each Data Collector:

   ```
   service activemq stop
   ```

5. Shut down the ActiveMQ broker on the Data Aggregator:

   ```
   service activemq stop
   ```

6. Start the ActiveMQ broker on the Data Aggregator:

   ```
   service activemq start
   ```

   If you do not, the Data Aggregator starts the broker automatically.
7. The Data Collectors automatically restart the ActiveMQ brokers. Use the following command to restart the brokers manually:

   ```
   service activemq start
   ```

Authenticate and Encrypt ActiveMQ Communication

By default, the communication between the Data Aggregator and Data Collector is unencrypted and unauthenticated. To secure communications, secure the communication between the ActiveMQ brokers on these servers.For more information, see Authenticate and Encrypt ActiveMQ Communication.

Install Mediation Manager

DX NetOps Mediation Manager monitors the performance for non-SNMP based devices, such as mobile wireless, fiber-optic switch, radio access, and 3G or 4G voice data. DX NetOps Mediation Manager supports a wide range of protocols to

access data, such as, SOAP, SSH, XML, SQL, JMS, SFTP, and HTTP. DX NetOps Mediation Manager is portable across all platforms.

You create and migrate device packs using DX NetOps Mediation Manager for Infrastructure Management. The device packs are vendor-specific API plug-ins that collect data from a device or from an Element Management System (EMS). Follow this process to install DX NetOps Mediation Manager for Infrastructure Management.

> **WARNING**
> Install DX NetOps Mediation Manager after you install DX NetOps Performance Management.
>
> **NOTE**
> When the same metric family data is collected through SNMP and through DX NetOps Mediation Manager, some of the data might not be correctly included in rollups. To avoid this problem, collect data from DX NetOps Mediation Manager though a dedicated data collector with a separate IP domain. The data from each IP domain is rolled up independently.

Follow these steps:

1. Install the MultiController on the NetOps Portal host.
2. Install the LocalController on the data collector host.
3. Install the required device packs.

For more information, see the DX NetOps Mediation Manager documentation.

Monitoring System Health with CA Systems Performance for Infrastructure Managers

To monitor the health of the servers in your system that host the Data Repository, Data Aggregator, Data Collector, and NetOps Portal, use the CA Systems Performance for Infrastructure Managers tool. Monitoring system health is optional. You are not required to download the Linux SystemEdge agent.

Install the agent on the distributed servers that you want to monitor. Use NetOps Portal to discover the agent on these servers, and to visualize agent data, such as CPU, memory, file systems, and other critical resources for each server.

You can download the agent image, including a Readme that provides complete installation instructions, from the CA Support Download Center.

Follow these steps to locate the agent image:

1. In a web browser, navigate to support.ca.com.
2. Log in with your email address and password. You must be a registered user on www.ca.com to access the Support pages.
   The Support home page opens.
3. Select **Download Center**, **Download Products**.
4. Type 'CA Systems Performance for Infrastructure Managers' in the **Enter the Product Name here** field.
5. Select **12.8** from the **Select a Release** drop-down list.
6. Select **SP 02** from the **Select a Gen level** drop-down list.
7. Click **Go**.
   The download links appear for the CA Systems Performance for Infrastructure Managers - MULTI-PLATFORM UNIX and Windows.
8. Click **Download**.

Install a Low-Scale System

In a standard large-scale deployment, the data aggregator and data collector components require dedicated nodes. For some smaller deployments of 150,000 polled items or less, you can run these components on a shared single node.

WARNING

Do not deploy the data aggregator and data collector on a single node in high-scale environments.

Limitations

Deploying DX NetOps Performance Management with the data aggregator and data collector on a single node does not support the following options:

- Calculated metrics beyond 150,000 metrics per second
- Fast (1 minute) polling
- Integration with DX NetOps Mediation Manager or DX NetOps Virtual Network Assurance
- Multiple DCs
- Multiple IP domains
- Multiple tenants
- Polling beyond 150,000 items

WARNING

This configuration does not persist through upgrades. After you upgrade the data aggregator, upgrade and configure the data collector according to the procedures described in this article.

How to Install a Low-Scale System

Use the following process to deploy DX NetOps Performance Management with the data aggregator and data collector on a single node:

1. Review Installation Requirements and Considerations.
2. Install .
   For more information, see Install Performance Center.
3. Install the Data Repository.
   NOTE
   Because the Data Repository in a low-scale system is deployed on a single node, skip the steps that refer to cluster installation.
   For more information, see Install the Data Repository.
4. Install the data aggregator
   For more information, see Install the Data Aggregator.
5. Install the data collector on the data aggregator host using the following procedures:
6. Complete the Post-Installation Configuration.

Install the Data Collector

Install and configure the data collector on the same host as the data aggregator.

Follow these steps:

1. Log in to the shared host.
2. Verify that the data aggregator service is running by issuing the following command:
   ```
   service dadaemon status
   ```
   NOTE
   For RHEL 7.x or OL, `service` invokes `systemctl` . You can use `systemctl` instead.
3. If the service is not running, do one of the following steps:
   – Start the data aggregator service by issuing the following command:

```
service dadaemon start
```

– (Fault tolerant environment) Issue one the following commands to enable the fault tolerant data aggregator so that it can start when necessary:
  - **RHEL 6.x:**
    ```
    service dadaemon activate
    ```
  - **RHEL 7.x, SLES, or OL:**
    ```
    DA_Install_Directory/scripts/dadaemon activate
    ```

4. Verify that the ActiveMQ broker is running by issuing the following command:
   ```
   service activemq status
   ```
   If the service is not running, start the ActiveMQ service by issuing the following command:
   ```
   service activemq start
   ```
5. Download the data collector installer:
   ```
   wget -nv http://da_host:8581/dcm/InstData/Linux/VM/install.bin
   ```
6. Make the install file executable by issuing the following command:
   ```
   chmod a+x install.bin
   ```
7. Do one of the following steps:
   – Stop the data aggregator and ActiveMQ services by issuing the following commands:
     ```
     service dadaemon stop
     service activemq stop
     ```
   – (Fault tolerant environment) If the local data aggregator is running, issue one the following commands to shut it down and prevent it from restarting until maintenance is complete:
     - **RHEL 6.x:**
       ```
       service dadaemon maintenance
       ```
     - **RHEL 7.x, SLES, or OL:**
       ```
       DA_Install_Directory/scripts/dadaemon maintenance
       ```
8. Install the data collector by issuing the following command:
   ```
   ./install.bin -i console
   ```
9. Follow the instructions in the console.
   The installer prompts you for the hostname or IP address of the data aggregator host. Supply the hostname of the shared host. The data collector is installed and the data collector and ActiveMQ services start. After a short time, the data collector service fails because the data aggregator service is not running.
10. Uninstall the Data Aggregator ActiveMQ service by issuing the following command:
    ```
    /opt/IMDataAggregator/scripts/activemq uninstall
    ```
11. Install the data collector ActiveMQ service by issuing the following command:
    ```
    /opt/IMDataCollector/scripts/activemq install
    ```

Reconfigure the Data Collector

After you install the data collector on the shared host, reconfigure the data collector to use the shared host. On a shared host, the data aggregator and data collector use a single ActiveMQ broker.

This procedure includes the following variables:

- `<DA_Install_Directory>`
  The installation directory for the data aggregator.
  Default: `/opt/IMDataAggregator`
- `<DC_Install_Directory>`

The installation directory for the data collector.
Default: `/opt/IMDataCollector`

Follow these steps:

1. Stop the data collector and ActiveMQ services by issuing the following commands:
   ```
   service dcmd stop
   service activemq stop
   ```
2. Modify the data aggregator startup script:
   a. Edit the *DA_Install_Directory*`/scripts/dadaemon` file.
   b. Modify the ACTIVEMQ_HOME variable to point to the data collector ActiveMQ broker by issuing the following command:
      ```
      ACTIVEMQ_HOME="<DC_Install_Directory>/broker/apache-activemq-version" export
       ACTIVEMQ_HOME
      ```
      - ***version***
        Specify the ActiveMQ version. To find the correct version, go to the *<DC_Install_Directory>*`/broker` directory.
3. Disable the data aggregator ActiveMQ broker heartbeat:
   a. Create the *<DA_Install_Directory>*`/apache-karaf-2.4.3/etc/com.ca.im.dm.core.amq.cfg` file.

      NOTE
      Create this file as the user running the data aggregator to ensure the correct permissions are set on the new file.
   b. Edit the file and insert the following property configuration:
      ```
      jmsbroker-heartbeat-disabled=true
      ```
   c. To disable log messages that are related to the data aggregator heartbeat manager, edit the *<DA_Install_Directory>*`/apache-karaf-2.4.3/etc/org.ops4j.pax.logging.cfg` file.
   d. Insert the following property configuration:
      ```
      # Disable ActiveMQ Health Monitoring log messages in shared DA/data collector mode
      log4j.logger.com.ca.im.core.jms.heartbeat.JmsHeartbeatManager=OFF
      ```
4. Modify the Apache Karaf JMX management properties:
   a. Edit the *<DC_Install_Directory>*`/apache-karaf-2.4.3/etc/org.apache.karaf.management.cfg` file.
   b. Set the following properties:
      ```
      rmiRegistryPort = 1199
      rmiServerPort = 44445
      ```
5. (Optional) To support the ability to debug the data aggregator and data collector components simultaneously, modify the Karaf debug port:
   a. Edit the *<DC_Install_Direcotry>*`/apache-karaf-2.4.3/bin/karaf` file.
   b. Set the following property:
      ```
      DEFAULT_JAVA_DEBUG_PORT=5006
      ```
6. Do one of the following steps:
   – Start the data aggregator service by issuing the following command:
     ```
     service dadaemon start
     ```
   – (Fault tolerant environment) Issue one the following commands to enable the fault tolerant data aggregator so that it can start when necessary:
     - **RHEL 6.x:**
       ```
       service dadaemon activate
       ```
     - **RHEL 7.x, SLES, or OL:**

```
DA_Install_Directory/scripts/dadaemon activate
```

7. Start the data collector service by issuing the following command:
   ```
   service dcmd start
   ```
8. The data aggregator and data collector services start on the single host. Complete post installation configuration for your deployment.

Validate the Deployment

Validate the deployment by verifying that the required services are functioning as expected.

Follow these steps:

1. Validate that the data aggregator service is running by issuing the following command:
   ```
   service dadaemon status
   ```
2. Validate that the data collector service is running by issuing the following command:
   ```
   service dcmd status
   ```
3. Validate that the ActiveMQ Broker is running by issuing the following command:
   ```
   service activemq status
   ```
4. Validate that the ActiveMQ Broker being run is the data collector broker by issuing the following command:
   ```
   service activemq status
   ```
   The returned message must include the following information:
   `Status for ActiveMQ: INFO: Loading '<DC_Install_Directory>/broker/apache-activemq-version/bin/env'`
   If the message does not show the data collector installation directory, repeat the step in the previous procedure to modify the data aggregator startup script.

Uninstall Performance Management

To uninstall DX NetOps Performance Management from your system, uninstall NetOps Portal and the data aggregator components.

Remove the components in the following order:

1. Uninstall Performance Center.
2. Uninstall the Data Aggregator Component.
3. Uninstall the Data Collectors.
4. Uninstall the Data Repository.

Uninstall Performance Center

To uninstall DX NetOps Performance Management, first remove the NetOps Portal component. To remove NetOps Portal, follow the steps in one of the following sections:

Uninstall NetOps Portal

To uninstall NetOps Portal using command line, run the Uninstallation program from a command prompt.

Follow these steps:

1. Log in to the server as 'root', or use the 'sudo' account that you have configured for the installation.
2. Navigate to the Uninstall_PerformanceCenter program.

NOTE
The Uninstall program is stored in /opt/CA/PerformanceCenter by default.

3. Run the following command:
   ```
   ./Uninstall_PerformanceCenter -i console
   ```
   The uninstallation program starts.
4. Press Enter to continue.
   The uninstaller shows the progress as it removes the NetOps Portal files, folders, registry entries, and shortcuts. The application and all of your unwanted components are removed from the server.

NOTE

MySQL data directory is preserved. In addition, the following directories are preserved:

- CA/PerformanceCenter/InstallLogs: Contains the uninstallation log.
- CA/jre: Preserved for other CA products that use the jre.

Clean Up After a Failed Installation

In some cases, the NetOps Portal installation can fail to complete. For example, if the /tmp directory lacks sufficient space for the installation files, the installation fails. If you experience an installation failure, clean up the directories before you reinstall the software.

Follow these steps:

1. To clean up the /tmp directory, remove unnecessary files.
2. To remove the /opt/CA directory, run the following command:
   ```
   rm  -rf /opt/CA
   ```
3. To remove the installer Registration File, run the following command:
   ```
   rm /var/.com.zerog.registry.xml
   ```
4. To remove all of the NetOps Portal service files, run the following command:
   ```
   rm service caperfcenter_*
   ```
5. Restart the server.
6. Install the software again.

Prepare for an Upgrade with a New Installation Directory

The installation procedure lets you select a new directory for the installation, or use the default directory. In some cases, you want to use a different installation directory for a new version of the software when you upgrade NetOps Portal. This scenario requires extra steps during the uninstallation. The goal is to remove all of the links from an initialization file that are related to NetOps Portal. The new installation can then obtain the correct links for the variables that determine the location of the Device Manager, Event Manager, and other processes.

You can uninstall processes individually to clean up your installation directories for an upgrade that uses a different installation directory.

Follow these steps:

1. Navigate to the installation directory for the old installation.
2. Switch to the DM subdirectory, and then to the uninstall subdirectory.
3. Run the uninstaller for the Device Manager process:
   ```
   ./Uninstall_DM
   ```
4. When the uninstallation of the Device Manager has completed, switch back to the installation directory, and then switch to the uninstall subdirectory of the EM subdirectory:
   ```
   cd ../
   ```

```
cd EM/uninstall
```

5. Run the uninstaller for the Event Manager process:
   ```
   ./Uninstall_EM
   ```
6. (Optional) Do a grep for the Device Manager and Event Manager variables:
   ```
   ls  - al /etc/init.d  | grep cap
   ```
   The variables caperfcenter_devicemanager and caperfcenter_eventmanager should not be present after a successful uninstallation of the Device Manager and Event Manager processes.
7. Uninstall NetOps Portal.

Uninstall the Data Aggregator Component

After you uninstall the NetOps Portal component, uninstall the Data Aggregator component. To remove Data Aggregator, follow the steps in one of the following sections:

(Optional) Configure the Sudo User Account for Data Aggregator

If root user access is not available, create a sudo user with access to a limited set of commands.

Follow these steps:

1. Log in to the computer where you want to install Data Aggregator as the root user.
2. Add the following command alias to the command alias section of the /etc/sudoers file:

   ```
   Cmnd_Alias CA_DATAAGG = /tmp/installDA.bin, service dadaemon, /opt/IMDataAggregator/
   uninstall

   ## Allows the Data Aggregator user to manage the Data Aggregator


   dasudouser_name ALL = CA_DATAAGG
   ```

   This command alias details the commands that the sudo user must be able to run.
   The sudo user account is configured.

Uninstall Data Aggregator from the Command Line

You can uninstall Data Aggregator that is installed on a computer in your networking environment.

Follow these steps:

1. Open a command prompt. Type the following command to log in to the computer where you want to uninstall Data Aggregator as the root user:

   ```
   su - root
   ```

   If you cannot log in as the root user, configure a sudo user account with access to a limited set of commands.
2. Type the following command to access the uninstallation directory:

   ```
   cd <caimda> installation directory/Uninstall
   ```

3. Type the following command to run the uninstaller:

```
./Uninstall
```

You are prompted to uninstall.

4. Select option 1 for a complete uninstallation and press Enter.
   Data Aggregator is uninstalled.

Uninstall Data Aggregator in Silent Mode

You can uninstall Data Aggregator silently. You cannot discover new devices or manage existing polled devices whose IP addresses fall within the IP domain that is associated with the uninstalled Data Aggregator.

Follow these steps:

1. Open a command prompt. Type the following command to log in to the computer where you want to uninstall Data Aggregator as the root user:

   ```
   su - root
   ```

   If you cannot log in as the root user, configure a sudo user account with access to a limited set of commands.
2. Type the following command to access the uninstallation directory:

   ```
   cd <caimda> installation directory/Uninstall
   ```

3. Type the following command to run the uninstaller:

   ```
   ./Uninstall
   ```

   Data Aggregator is uninstalled.

Uninstall Data Aggregator with the Installation Wizard

You can uninstall Data Aggregator using the installation wizard.

Follow these steps:

1. Open a command prompt. Type the following command to log in to the computer where you want to uninstall Data Aggregator as the root user:

   ```
   su - root
   ```

   If you cannot log in as the root user, configure a sudo user account with access to a limited set of commands.
2. Type the following command to access the uninstallation directory:

   ```
   cd <caimda> installation directory/Uninstall
   ```

3. Type the following command to run the uninstaller:

   ```
   ./Uninstall
   ```

   You are prompted to uninstall.
4. Click Next.
   The Uninstall Options dialog opens.

5. Select Complete Uninstall and click Next.
   Data Aggregator is uninstalled.

Uninstall the Data Collectors

After you uninstall the Data Aggregator component, uninstall the Data Collector component.

(Optional) Configure the Sudo User Account for Data Collector

If you cannot log in as the root user, create a sudo user with access to a limited set of commands.

Follow these steps:

1. Log in to the computer where you want to install Data Collector as the root user.
2. Add the following command alias to the command alias section of the /etc/sudoers file:

   ```
   Cmnd_Alias CA_DATACOLL = /tmp/install.bin, service dcmd, /opt/IMDataCollector/
   Uninstall/Uninstall

   ## Allows the <caimdc> user to manage the <caimdc>


   sudouser_name ALL = CA_DATACOLL
   ```

   This command alias details the commands that the sudo user must be able to run.
   The sudo user account is configured.

Uninstall the Data Collector

You can uninstall Data Collector that is installed on a computer in your networking environment.

Follow these steps:

1. Log in to the Data Collector host as the root user or sudo user.
2. Open a command prompt.
3. Access the uninstallation directory:

   ```
   cd DC_installation_directory/Uninstall
   ```

4. Run the uninstaller:

   ```
   ./Uninstall
   ```

   You are prompted to uninstall.
5. Select option 1 for a complete uninstallation, and press Enter.
6. Remove the following file:
   `/var/.com.zerog.registry.xml`
   Data Collector is uninstalled.

Uninstall the Data Repository

After you uninstall the Data Collector and Data Aggregator components, uninstall the Data Repository component.

TIP

Back up your Data Repository for later use.

Follow these steps:

1. Open a console and type the following command to become the Linux user account for the database administrator user:
   `su - Linux user account for the database administrator user`
   For example:
   `su - dradmin`
2. Type the following command:
   `/opt/vertica/bin/adminTools`
   The Administration Tools dialog opens.

   NOTE
   If the database is running in a cluster, you can launch adminTools from any node in the cluster.
3. Select (4) Stop Database and select OK.
   If Data Repository does *not* stop, select (7) Advanced Menu, then (2) Stop Vertica on Host. If Data Repository still does not stop, select (3) Kill Vertica Process on Host in the Advanced menu.

   NOTE
   If Data Repository is running in a cluster, you may need to select (3) Kill Vertica Process on Host on more than one host in the cluster.

   Data Aggregator stops automatically.
4. Note the location of the Data Repository data directory as follows:
   a. Select (6) Configuration Menu.
   b. Select (3) View Database.
   c. The database directory for the database is the parent of the catalog directory that is shown in the output.
5. Drop the database within the Administration Tools dialog as follows:
   a. Select (6) Configuration Menu.
   b. Select (7) Drop Database.
6. Exit as the database admin user and su, or login as the root/sudo user.
7. Type the following command to find the name of the Data Repository package that is installed:
   `rpm -qa | grep vertica`

   NOTE
   If Data Repository is running in a cluster, repeat this step for each host participating in the cluster.
8. Type the following command to remove the Data Repository package:
   `rpm -e package name retrieved in the previous step`

   NOTE
   If Data Repository is running in a cluster, repeat this step for each host participating in the cluster.
9. Delete the following directories:
   a. Type the following command to delete the /opt/vertica/ directory and all subdirectories:
      `rm -rf /opt/vertica/`
   b. Type the following command to display the database directory:
      `ls data_repository_directory`
      - **data_repository_directory**
        Specify the installation directory of your Data Repository.
        Default: `/opt/CA/IMDataRepository_verticaVersion/`

Verify that the database directory specified is correct.
Type the following command to delete the database directory:

```
rm -rf data_repository_directory
```

NOTE
If Data Repository is running in a cluster, repeat this step for each host participating in the cluster.

Data Repository is uninstalled.

Upgrading

To upgrade DX NetOps Performance Management from a previous release, complete the following steps:

NOTE

If you enable FIPS-compliant encryption and you upgrade NetOps Portal before the Data Aggregator as recommended, temporary FIPS compatibility synchronization errors occur. This temporary condition is resolved when the Data Aggregator is upgraded.

WARNING

Do not enable FIPS when a product upgrade is in progress and different versions of NetOps Portal and the Data Aggregator might be in place. Doing so could disable the Data Aggregator data source.

NOTE

If you are setting up fault tolerance for the first time, see Fault Tolerance.

1. Review the Upgrade Requirements and Considerations.
2. Plan the Upgrade or Migration.

   NOTE

   You must upgrade the existing system to the product version you are migrating to before migrating.
3. Upgrade the Data Repository.

   WARNING

   This release includes an update to the Data Repository backup procedure. Configure a new backup. For more information, see Back Up the Data Repository.
4. Upgrade Performance Center.
5. Upgrade the Data Aggregator.

   NOTE

   To upgrade the Data Aggregators in an existing fault tolerant environment, see Upgrade Fault Tolerant Data Aggregators.
6. Upgrade the Data Collectors.

   NOTE

   For a low-scale system where the Data Collector shares a host with the Data Aggregator, complete the configuration to enable the components to run on the same host. For more information, see Install a Low-Scale System.
7. Complete the Upgrade.

   TIP

   During an upgrade that includes a Data Repository upgrade, update the Data Repository first. If the validation script fails, you can restart your system with the previous version to reduce downtime.

Upgrade Requirements and Considerations

Before you upgrade, review the following information to ensure that you have met all the requirements:

For cloud environments, see Rehydrating Data in a Cloud Environment.

Verify System Requirements

To ensure that you successfully upgrade DX NetOps Performance Management, verify that your environment meets the system requirements. To verify the sizing requirements, use the DX NetOps Performance Management Sizing Tool.

The following operating systems were verified:

- SUSE Linux Enterprise Server (SLES) 12 SP2
- Oracle Linux (OL) 7.3 (Red Hat compatible kernel only)

The following RHEL versions were verified:

NOTE

A RHEL installation offers packages and add-ons. DX NetOps Performance Management supports a Minimal Install environment for RHEL.

- RHEL 6.6 - 6.9
- RHEL 6.10
- RHEL 7.3 - 7.5

NOTE

RHEL 6.8, 6.9, 7.3, or 7.4 are recommended.

The following RHEL versions are not supported:

- RHEL 5.x
- RHEL 6.5 and lower
- RHEL 7.0 - 7.2

DX NetOps Performance Management components that are installed on unsupported RHEL versions require an OS upgrade before upgrading the DX NetOps Performance Management components. You might also need to move the DX NetOps Performance Management components, including configurations, customizations, and data, from one system to another system. For more information, see Plan the Upgrade or Migration.

Package Requirements

The installer for each component requires the following packages:

Components	Packages
All (SLES)	• `dialog` • `mcelog` • `zip` • `unzip`
All (RHEL 7.x, OL)	• `dialog` • `mcelog` • `zip` • `unzip` • `chrony`

All (RHEL 6.x)	• dialog • mcelog • zip • unzip • glibc
NetOps Portal (SLES)	• fontconfig • libaio1 • libnuma1 • wget
NetOps Portal (RHEL 6.x)	• fontconfig • libaio • libaio-devel • numactl • wget
NetOps Portal (RHEL 7.x, OL)	• fontconfig • libaio • libaio-devel • numactl-libs • wget
Data Repository (RHEL 6.x, RHEL 7.x, SLES, OL)	• bc • pstack • gstack For RHEL 7.x, the pstack and gstack packages are included in the gdb package.
Data Collectors (RHEL 6.x, RHEL 7.x, SLES, OL)	• at

Direct Upgrade Paths

Because releases 20.2 and higher upgrade the Data Repository to Vertica release 9.1, we support Data Repository upgrades directly from the following releases only to 20.2:

- DX NetOps Performance Management 3.6
- DX NetOps Performance Management 3.7

NOTE

For DX NetOps Performance Management releases before 3.6, you must upgrade to the 3.6 release before upgrading the Data Repository to 20.2. The Vertica release 9.1 is the same for DX NetOps Performance Management 3.7 and 20.2. Therefore, a Data Repository upgrade is unnecessary from 3.7 to 20.2.

For other DX NetOps Performance Management components, we support upgrades directly from the following releases to 20.2:

- DX NetOps Performance Management 3.2
- DX NetOps Performance Management 3.5
- DX NetOps Performance Management 3.6
- DX NetOps Performance Management 3.7

For more information, see Plan the Upgrade or Migration.

Sudo User Installations

For systems without root access, sudo user accounts are used to install and run the software. Add the sudo prefix to commands to upgrade the components. Verify that your sudo accounts contain the correct permissions for each component.

For more information, see the sudo user configuration details for each component:

- Prepare to Install the Data Repository
- Prepare to Install Performance Center
- Prepare to Install the Data Aggregator
- Prepare to Install the Data Collectors

Back Up DX NetOps Performance Management

Before you begin the upgrade process, back up all aspects of the product:

- Data Repository
  For more information, see Back Up the Data Repository.
- NetOps PortalFor more information, see Back Up Performance Center.
- Data Aggregator
  For more information, see Back Up Data Aggregator.

Low-Scale Systems

For a low-scale system where the Data Collector shares a host with the Data Aggregator, complete the configuration to enable the components to run on the same host. For more information, see Install a Low-Scale System.

CA Business Intelligence Integrations

If you have an existing CA Business Intelligence (CABI) integration, this version of DX NetOps Performance Management supports CABI 6.4.2 and higher.

> **WARNING**
>
> After you upgrade to a supported CABI version, you must re-install your CABI content. For more information, see Install CA Business Intelligence Reports and Dashboards.

ActiveMQ Versions

To minimize data loss when you upgrade your Data Collectors, ActiveMQ continues to run unless its version is upgraded with the release.

The following DX NetOps Performance Management releases correspond with the following ActiveMQ versions:

DX NetOps Performance Management	ActiveMQ
2.5 and earlier	5.5.1
2.6, 2.7	5.11.1
2.8, 3.0, 3.1, 3.2	5.13.1
3.5 and later	5.15.2

Apache Karaf Update

Release 2.7 introduced an update to Apache Karaf that changes the name of the Karaf directory. Karaf now uses the following directory: `installation_directory/apache-karaf-2.4.3`

NetOps Upgrade Considerations

The DX NetOps Performance Management user interface now supports supports the co-existence of retired and replacement devices coming from CA Network Flow Analysis. If your system is integrated with CA Network Flow Analysis, you must upgrade CA Virtual Network Assurance along with DX NetOps Performance Management to ensure that Items monitored by both the Data Aggregator and Virtual Network Assurance are properly reconciled when Lifecycle changes occur.

Review the Known Limitations

To ensure that your system runs successfully after the upgrade, review the limitations of DX NetOps Performance Management 3.7. For more information, see Known Limitations.

Plan the Upgrade or Migration

The path that you take for your upgrade or migration depends on various factors. Use the following sections to plan your upgrade:

For cloud environments, see Rehydrating Data in a Cloud Environment.

Verify ETL Health

Before you upgrade, run the `etlHealth.sh` script. For releases 20.2.3 and higher, the script is included with the Data Repository installation. For earlier releases, contact Support. We recommend you run this script well in advance of the upgrade and again directly before the upgrade.

Follow these steps:

1. Log in to one of the Data Repository nodes as the root user.
2. Run the validation script:
   `./etlHealth.sh dauser dapassword`
3. Do one of the following:
   - If the health check passes, proceed with the upgrade.
   - If the health check fails, follow the instructions in the prompt to collect the irep and the data collected by the `etlHealth.sh` script, and submit these details in a Support ticket.

Plan the DX NetOps Performance Management Upgrade

Because releases 20.2 and higher upgrade the Data Repository to Vertica release 9.1, we support Data Repository upgrades directly from the following releases only to 20.2:

- DX NetOps Performance Management 3.6
- DX NetOps Performance Management 3.7

> **NOTE**
>
> For DX NetOps Performance Management releases before 3.6, you must upgrade to the 3.6 release before upgrading the Data Repository to 20.2. The Vertica release 9.1 is the same for DX NetOps Performance Management 3.7 and 20.2. Therefore, a Data Repository upgrade is unnecessary from 3.7 to 20.2.

For other DX NetOps Performance Management components, we support upgrades directly from the following releases to 20.2:

- DX NetOps Performance Management 3.2
- DX NetOps Performance Management 3.5
- DX NetOps Performance Management 3.6
- DX NetOps Performance Management 3.7

The following table describes the best upgrade paths for indirect upgrades to 20.2:

NOTE
For information about previous releases, see the previous documentation. For assistance locating releases 3.1 and earlier, contact your Support representative.

Current Version	Upgrade Path
3.2	3.5 to 3.6 to 20.2 (Data Repository) 20.2 (other components)
3.5	3.6 to 20.2 (Data Repository) 20.2 (other components)

NOTE
For the upgrade paths for releases before 3.2, see the previous documentation.

Plan the Operating System Upgrade or Install

The following operating systems were verified:

- SUSE Linux Enterprise Server (SLES) 12 SP2
- Oracle Linux (OL) 7.3 (Red Hat compatible kernel only)

The following Red Hat Enterprise Linux (RHEL) versions were verified:

NOTE

A RHEL installation offers packages and add-ons. DX NetOps Performance Management supports a Minimal Install environment for RHEL.

- RHEL 6.6 - 6.9
- RHEL 7.4
- RHEL 7.5

NOTE

RHEL 6.8, 6.9, 7.3, or 7.4 are recommended.

To plan your OS upgrade and understand your upgrade scenario, review the following details and answer the following questions:

NOTE

The following numbered questions correlate to the numbered decision steps in the following flowchart.

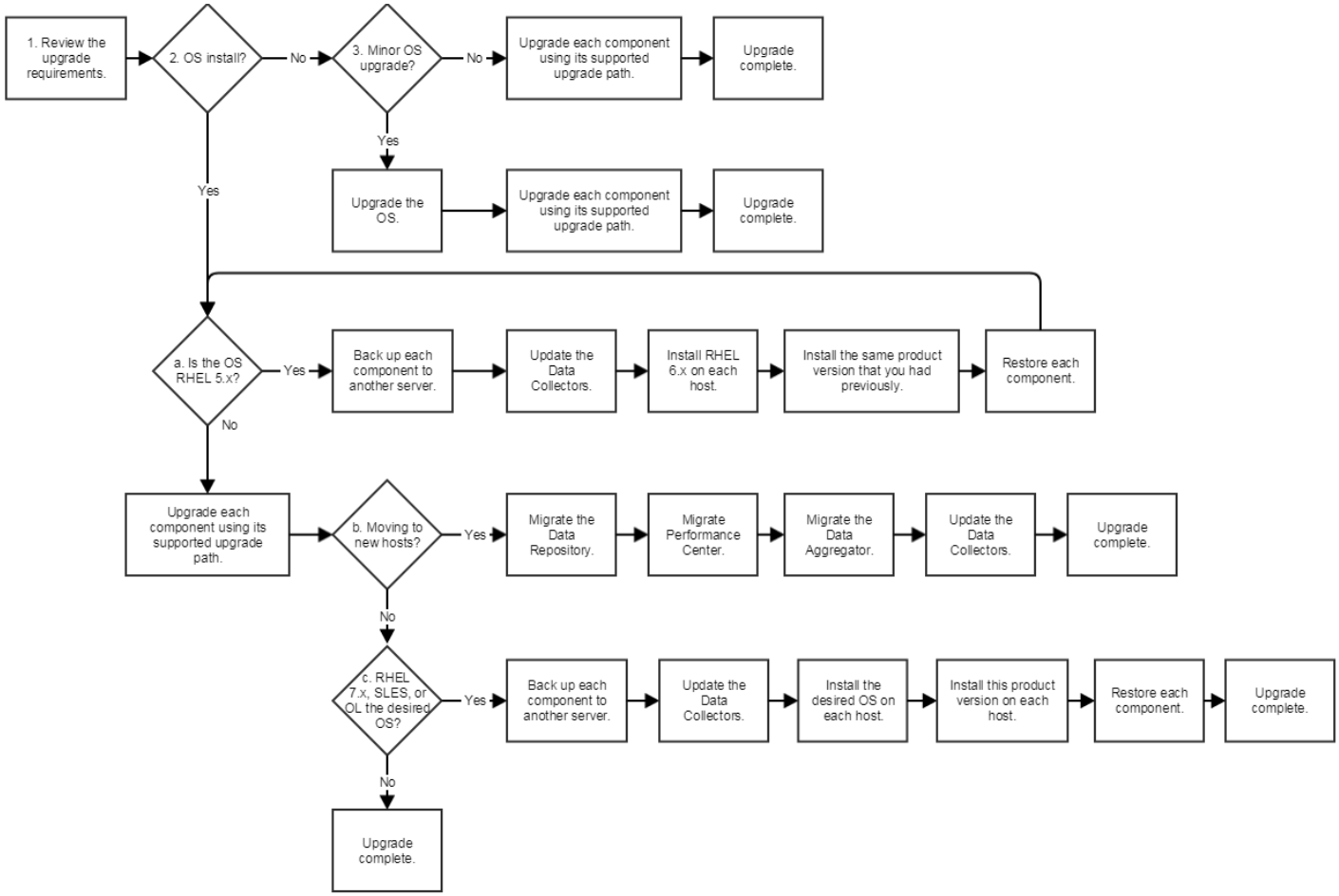
1. If you have not yet, review the Upgrade Requirements and Considerations.
2. Are you going to perform an OS installation (for example, going from RHEL 6.9 to RHEL 7.5 or SLES 12 SP2)?
   a. Is your current OS RHEL 5.x?
   b. Are you going to move to new hosts?
   c. Is RHEL 7.x, SLES, or OL the desired OS?
3. Are you going to perform a minor OS upgrade (for example, RHEL 6.5 to RHEL 6.9)?

The following OS upgrade and install scenarios upgrade or install the OS for all components. However, your unique scenario might upgrade or install the OS for a select set of components.

Upgrade Scenarios

Review the following graphic to help identify your upgrade scenario. Then use the procedures that are outlined in the following sections. The following graphic provides a high-level overview of some common upgrade scenarios:

Figure 6: Plan_Upgrade



Upgrade DX NetOps Performance Management without an OS Upgrade

If this version of DX NetOps Performance Management supports your current OS, you can upgrade each component using its supported upgrade path. For more information, see Upgrading.

Upgrade DX NetOps Performance Management with a Minor OS Upgrade

If you are performing a minor OS upgrade (for example, RHEL 6.5 to RHEL 6.9), do the following tasks:

1. Upgrade your OS.
2. Upgrade each component using its supported upgrade path. For more information, see Upgrading.

Migrate to New Hosts from a Supported OS

Migration involves moving the DX NetOps Performance Management components, including configurations, customizations, and data, from one system to another system.

The following situations require migration:

- You are moving to new hosts for an OS install (for example, RHEL 6.9 to RHEL 7.5 or SLES 12 SP2).
- The current database hardware no longer meets sizing requirements.
- You are moving from virtual machines to physical hardware for the database.

If you are moving DX NetOps Performance Management components to new systems with new IP addresses and hostnames from a supported OS, use the following procedures:

1. Upgrade each component using its supported upgrade path. For more information, see Upgrading.

   NOTE

   You must upgrade the existing system to the product version you are migrating to before migrating.
2. Migrate the Data Repository.

   NOTE

   Depending on the amount of data, migrating the Data Repository can take a significant amount of time.

   TIP

   To minimize downtime, perform incremental backups after your initial backup.
3. Migrate Performance Center.

   NOTE

   This migration procedure copies over NetOps Portal and does not require you to install NetOps Portal on the new host.
4. Migrate the Data Aggregator.
5. Update the Data Collector.
   a. For DX NetOps Mediation Manager, see the CAMM documentation.
   b. For DX NetOps Virtual Network Assurance, see the CA VNA documentation.

Migrate to New Hosts from an Unsupported OS

NOTE

If your current OS is RHEL 5.x, you must first move to RHEL 6.x.

If you are moving DX NetOps Performance Management components to new systems with new IP addresses and hostnames from an unsupported OS, use the following procedures:

1. Upgrade each component using its supported upgrade path until you reach a version from which you can upgrade directly to 20.2. For more information, see the Upgrading documentation for the version from which you are upgrading.
2. Back up each component to another server:
   a. Back Up the Data Repository.

      NOTE

      Depending on the amount of data, backing up and restoring the Data Repository can take a significant amount of time.

      TIP

      To minimize downtime, perform incremental backups after your initial backup.
   b. Back Up Performance Center.
   c. Back Up Data Aggregator.
3. Install the OS on each host.

4. Install the same product version from your recent upgrade on each host. For more information, see the Installing documentation for the recommended version.
5. Restore each component:

   NOTE

   For each component, verify that the backup is copied over.

   a. Restore Data Repository.
   b. Restore Performance Center.
   c. Restore Data Aggregator.
6. Upgrade each component to this product version. For more information, see Upgrading.
7. Migrate the Data Repository.

   NOTE

   Depending on the amount of data, migrating the Data Repository can take a significant amount of time.

   TIP

   To minimize downtime, perform incremental backups after your initial backup.
8. Migrate Performance Center.

   NOTE

   This migration procedure copies over NetOps Portal and does not require you to install NetOps Portal on the new host.
9. Migrate the Data Aggregator.
10. Update the Data Collector.
    a. For DX NetOps Mediation Manager, see the CAMM documentation.
    b. For DX NetOps Virtual Network Assurance, see the CA VNA documentation.

Perform an OS Installation from a Supported OS

If you are performing a major OS upgrade from a supported OS (for example, RHEL 6.9 to RHEL 7.5 or SLES 12 SP2) on the same systems with the same IP addresses and hostnames, use the following procedures:

1. Upgrade each component using its supported upgrade path. For more information, see Upgrading.

   NOTE

   You must upgrade the existing system to the product version you are backing up and restoring to before backing up and restoring.
2. Back up each component to another server:
   a. Back Up the Data Repository.

      NOTE

      Depending on the amount of data, backing up and restoring the Data Repository can take a significant amount of time.

      TIP

      To minimize downtime, perform incremental backups after your initial backup.
   b. Back Up Performance Center.
   c. Back Up Data Aggregator.
3. Update the Data Collector.
   a. For DX NetOps Mediation Manager, see the CAMM documentation.
   b. For DX NetOps Virtual Network Assurance, see the CA VNA documentation.

4. Install the OS on each host.
5. Install this product version on each host. For more information, see Installing.
6. Restore each component:

   NOTE

   For each component, verify that the backup is copied over.

   a. Restore Data Repository.
   b. Restore Performance Center.
   c. Restore Data Aggregator.

Perform an OS Installation from an Unsupported OS

NOTE

If your current OS is RHEL 5.x, you must first move to RHEL 6.x.

If you are performing a major OS upgrade from an unsupported OS (for example, RHEL 5.x to RHEL 6.9) on the same systems with the same IP addresses and hostnames, use the following procedures:

1. Upgrade each component using its supported upgrade path until you reach a version from which you can upgrade directly to 20.2. For more information, see the Upgrading documentation for the version from which you are upgrading.
2. Back up each component to another server:
   a. Back Up the Data Repository.

      NOTE

      Depending on the amount of data, backing up and restoring the Data Repository can take a significant amount of time.

      TIP

      To minimize downtime, perform incremental backups after your initial backup.

   b. Back Up Performance Center.
   c. Back Up Data Aggregator.
3. Update the Data Collector.
   a. For DX NetOps Mediation Manager, see the CAMM documentation.
   b. For DX NetOps Virtual Network Assurance, see the CA VNA documentation.
4. Install the OS on each host.
5. Install the same product version from your recent upgrade on each host. For more information, see the Installing documentation for the recommended version.
6. Restore each component:

   NOTE

   For each component, verify that the backup is copied over.

   a. Restore Data Repository.
   b. Restore Performance Center.
   c. Restore Data Aggregator.
7. Upgrade each component to this product version. For more information, see Upgrading.

Upgrade the Data Repository

Upgrade the Data Repository first. If the dr_validate.sh script detects issues that you must resolve before the upgrade, you can restart your system while you resolve the issue.

NOTE
The Vertica release 9.1 is the same for DX NetOps Performance Management 3.7 and 20.2. Therefore, a Data Repository upgrade is unnecessary from 3.7 to 20.2.

To upgrade the Data Repository, complete the following steps:

Verify the Prerequisites

Verify the following prerequisites before you upgrade Data Repository:

NOTE
For information about previous releases, see the previous documentation. For assistance locating releases 3.1 and earlier, contact your Support representative.

- If your current Data Repository is running Vertica 7.2.3 (the version that is used in DX NetOps Performance Management 3.0 / 3.1), you must do an intermediate upgrade to Vertica 8.1.
  - Check the version running by using admintools as the database admin:
    ```
    /opt/vertica/bin/admintools -t list_allnodes
    ```
  - Use the 3.5_installDR.bin provided with the 3.6 Media to upgrade to Vertica 8.1.
  - Refer to the 3.5 Data Repository upgrade procedures.
  - Once Data Repository is running on Vertica 8.1, proceed with the DX NetOps Performance Management 3.6 Data Repository upgrade.
- If your current Data Repository is running Vertica 7.1.2 (the version that is used in DX NetOps Performance Management 2.7 / 2.8), you must do an intermediate upgrade to Vertica 7.2.3.
  - Check the version running by using admintools as the database admin:
    ```
    /opt/vertica/bin/admintools -t list_allnodes
    ```
  - Use the 3.1_installDR.bin provided with the 3.5 Media to upgrade to Vertica 7.2.3.
  - Refer to the 3.1 Data Repository upgrade procedures.
  - Once Data Repository is running on Vertica 7.2.3, proceed with the DX NetOps Performance Management 3.5 Data Repository upgrade.
- Verify whether the dialog, chrony, zip, and unzip packages are installed on each Data Repository host:

  NOTE

  The chrony package is required only for RHEL 7.x and OL 7.x.

  ```
  rpm -qa | grep ^dialog
  rpm -qa | grep ^chrony
  rpm -qa | grep ^zip
  rpm -qa | grep ^unzip
  ```

  If a required package for your OS is missing, install the package:

  NOTE

  If you are not the root user, use the sudo prefix.

  ```
  yum -y install dialog
  yum -y install chrony
  yum -y install zip
  yum -y install unzip
  ```

SLES:

```
zypper install dialog -y
zypper install zip -y
zypper install unzip -y
```

If this package is not installed, the validation and installation scripts fail.

- Verify that the Data Repository host has at least 2 GB of swap space.
- Verify that the Data Repository hosts use the ext4 file system for data and catalog directories. If not, remake the file system as ext4.

  WARNING

  The default file system for RHEL 7.x and OL 7.x is the XFS file system. The default file system for SLES is btrfs. Vertica does not support XFS or btrfs. The database supports only the ext4 file system.
- Verify that you are not using Logical Volume Manager (LVM) for data and catalog directories.

Verify Hardware and Network Performance

Vertica provides a set of utilities that test the performance of your hardware for the Data Repository. To verify that the environment is ideal for the database, run these tests before you upgrade the Data Repository.

NOTE

If you have already installed the Data Repository, you can perform these tests at any time to verify performance. The utilities are available on each node in `/opt/vertica/bin` .

If the tests do not meet the recommendations, fix the issues before you continue the upgrade.

vcpuperf

This utility measures the CPU processing speed of the host and compares the speed against benchmarks for common server CPUs. The utility measures how long the server requires to complete the test, and determines whether CPU throttling is enabled.

Follow these steps:

1. Execute the following command on *each* Data Repository node:

   ```
   ./vcpuperf > /tmp/vcpuperf.out
   ```

2. Verify that the performance meets the following requirements:
   - The CPU time is consistent with the benchmark values in the output.
   - The low load time and high load time are within 10 microseconds. If the difference is grater than 50 microseconds, CPU throttling might be enabled on your system. Disable CPU throttling.

Example:

The following example shows the return from this utility:

```
$ /opt/vertica/bin/vcpuperf

Compiled with: 4.1.2 20080704 (Red Hat 4.1.2-52)

Expected time on Core 2, 2.53GHz: ~9.5s
```

```
Expected time on Nehalem, 2.67GHz: ~9.0s

Expected time on Xeon 5670, 2.93GHz: ~8.0s




This machine's time:

CPU Time: 7.740000s

Real Time:7.740000s




Some machines automatically throttle the CPU to save power.

This test can be done in <100 microseconds (60-70 on Xeon 5670, 2.93GHz).

Low load times much larger than 100-200us or much larger than the corresponding high
 load time

indicate low-load throttling, which can adversely affect small query / concurrent
 performance.

This machine's high load time: 67 microseconds.


This machine's low load time: 64 microseconds.
```

This test was performed on a system with 2.67-GHz processors, so the real time is acceptable. The difference between the high load time and low load time is within the expected tolerance.

For more information about this utility, see the Vertica documentation.

vioperf

This utility tests the performance of the disk input and output (I/O). The utility performs a series of reads and writes.

NOTE

To measure the read/write speeds when using the same SAN/NAS for the disk or VM disk, you must run vioperf on all nodes of the cluster at the same time for the data or catalog directory.

Follow these steps:

1. Execute the following commands on *each* Data Repository node:

   ```
   ./vioperf /data > /tmp/vioperf.out --duration=60sec
   ```

   /data is the full path of the data directory.

   ```
   ./vioperf /catalog > /tmp/vioperf.out --duration=60sec
   ```

   /catalog is the full path of the catalog directory.
2. Verify the Write and Read counter values at least 40 MB/s per core.
   The recommended I/O is 40 MB/s per physical core on each node. For example, the recommended I/O rate for a node with 2 hyper-threaded six-core CPUs (12 physical cores) is 480 MB/s.

   WARNING

   If the **thread count** column shows a value of 1, the utility cannot determine the number of cores. Add the following arguement to the command to run the utility:

   ```
   --thread-count=CORES
   ```

   Cores defines the number of cores in the system as a fixed integer.

Example:

The following example shows the return from this utility for the data directory:

```
The minimum required I/O is 20 MB/s read and write per physical processor core on each
 node, in full duplex i.e. reading and writing at this rate simultaneously, concurrently
 on all nodes of the cluster. The recommended I/O is 40 MB/s per physical core on each
 node. For example, the I/O rate for a server node with 2 hyper-threaded six-core CPUs
 is 240 MB/s required minimum, 480 MB/s recommended.




Using direct io (buffer size=1048576, alignment=512) for directory "/drdata"
```

```
test      | directory                                        | counter name        |
 counter value       | counter value (10 sec avg)      | counter value/core  | counter
 value/core (10 sec avg) | thread count  | %CPU  | %IO Wait  | elapsed time (s)|
 remaining time (s)

-------------------------------------------------------------------------------------------


Write     | /drdata                                          | MB/s
  | 873                | 873                             | 54.5625             |
 54.5625                         | 16            | 29    | 40        | 10              |
 5


Write     | /drdata                                          | MB/s
  | 868                | 865                             | 54.25               |
 54.0625                         | 16            | 28    | 30        | 15              |
 0

ReWrite   | /drdata                                          | (MB-read+MB-write)/
s| 275+275             | 275+275                         | 17.1875+17.1875     |
 17.1875+17.1875                 | 16            | 13    | 21        | 10              |
 5

ReWrite   | /drdata                                          | (MB-read+MB-write)/
s| 242+242             | 178+178                         | 15.125+15.125       |
 11.125+11.125                   | 16            | 7     | 17        | 15              |
 0


Read      | /drdata                                          | MB/s
  | 735                | 735                             | 45.9375             |
 45.9375                         | 16            | 11    | 23        | 10              |
 5


Read      | /drdata                                          | MB/s
  | 786                | 786                             | 49.125              |
 49.125                          | 16            | 26    | 25        | 15              |
 0

SkipRead  | /drdata                                          | seeks/s
  | 4511               | 4511                            | 281.938             |
 281.938                         | 16            | 14    | 19        | 10              |
 5
```

```
SkipRead  | /drdata                                    | seeks/s
  | 4477                | 4407                        | 279.812            |
 275.438                         | 16            | 3     | 15        | 15              |
 0
```

This server has 16 cores. The Read and Write counter values indicate the I/O is greater than 40 MB/s per core.

For more information about this utility, see the Vertica documentation.

vnetperf

This utility tests the network performance of the Data Repository hosts. The utility measures network latency and the throughput for the TCP and UDP protocols.

> **WARNING**
>
> This utility causes a high network load and affects database performance. Do not run this utility while the database is running.

Follow these steps:

1. Log in as a user that has passwordless ssh between the nodes.
2. Execute the following command on *one* of the Data Repository nodes:

   ```
   ./vnetperf --hosts DAhost,DRhost1,DRhost2,DRhost3 > /tmp/vnetperf.out
   ```

   Specify the hostname or IP address of the Data Aggregator host and each Data Repository host.
3. Verify that the network performance meets the following requirements:
   - Round-trip time (RTT) latency of 200 microseconds or less
   - Clock skew under 1 second
   - Throughput of 800 MB/s or more
     The utility runs a series of throttled tests. Verify the throughput for the highest speed test.

For more information about this utility, see the Vertica documentation.

Verify the Limit on the Number of Open Files

Verify that the user that is installing Data Repository has a value of at least 65536 on the number of open files. Set this value permanently. Complete this procedure for each node in the Data Repository cluster.

Follow these steps:

1. As the root or sudo user, log in to the Data Repository host node.
2. Run the following command:

   ```
   su dradmin
   ```

3. Verify the number of open files:

   ```
   ulimit -n
   ```

   The command returns the ulimit number. This number must be at least 65536.

   > **NOTE**
   >
   > The number must be the same on all nodes in the cluster.

4. If this number is not at least 65536, do the following steps:
   a. Change the ulimit for the open files limit to 65536:
      ```
      ulimit -n 65536
      ```
   b. Open the /etc/security/limits.conf file on each Data Repository node, and add the following lines:
      ```
      # Added by Vertica
      * soft nofile 65536
      # Added by Vertica
      * hard nofile 65536
      # Added by Vertica
      *       soft    fsize   unlimited
      # Added by Vertica
      *       hard    fsize   unlimited
      ```
   c. Restart the sshd service on each Data Repository node:
      ```
      service sshd restart
      ```

      NOTE

      For RHEL 7.x, and OL 7.x `service` invokes `systemctl` . You can use `systemctl` instead.

      NOTE

      If you do not have the restart argument, stop and start sshd:

      ```
      service sshd stop
      service sshd start
      ```

   d. Verify that the number of open files is set properly:
      ```
      ulimit -n
      ```

      The command returns the ulimit number that you specified earlier.

Disable the Automatic Recovery of the Data Aggregator Process

The recovery process restarts the Data Aggregator. Disable the automatic recovery before you upgrade. For a fault tolerant environment, put the inactive Data Aggregator and then the active Data Aggregator into maintenance mode.

Follow these steps:

1. Log in to the Data Aggregator host as the root user.
2. Open a console and type the following command:
   ```
   crontab -e
   ```

   A vi session opens.
3. If the following line exists, add # to the beginning of the line to comment it out:
   ```
   # * * * * * /etc/init.d/dadaemon start > /dev/null
   ```

   The automatic recovery is disabled.

(Fault Tolerant Environment) Turn on Maintenance Mode

WARNING

If you are upgrading the Data Aggregators in an existing fault tolerant environment, you must put the Data Aggregator into maintenance mode before upgrade.

Put the inactive Data Aggregator into maintentance mode first. Then put the active Data Aggregator into maintenance mode.

For more information, see Fault Tolerance.

Follow these steps:

1. Log in to the Data Aggregator host as the root user or a sudo user.
2. Run one of the following commands to shut down the inactive Data Aggregator and prevent it from restarting until the upgrade is complete:

- – **RHEL 6.x:**

```
service dadaemon maintenance
```

– **RHEL 7.x, SLES, or OL:**

```
DA_Install_Directory/scripts/dadaemon maintenance
```

Upgrade Data Repository

You can initiate the Data Repository installation from any of the hosts in the cluster. The required software components are pushed to the other nodes during the installation.

Follow these steps:

1. Log in to one of the Data Repository nodes as the root user.
2. Determine which hosts Vertica is running on:
   a. As the database administrator user, open the Vertica Administration Tools:
   ```
   /opt/vertica/bin/adminTools
   ```
   b. Select option 6 (Configuration Menu).
   c. Select option 3 (View Database).
   d. Select the database.
   e. Note the IP addresses and database name for use later in this procedure.
   f. Exit the adminTools utility, and revert to the root user or sudo user.
3. Copy the installDR.bin file to /tmp.
4. Change directory to the location of the installDR.bin file:
   ```
   cd /tmp
   ```
5. Change the executable permissions:
   ```
   chmod u+x installDR.bin
   ```
6. Extract the installation files:
   ```
   ./installDR.bin
   ```
7. Follow the instructions in the console.
   This script extracts the following installation scripts:

- **dr_validate.sh** verifies that the system meets the prerequisites for the Data Repository upgrade.
- **dr_install.sh** upgrades the Vertica database.

8. If available from the Data Repository with the older version of Vertica, copy the existing drinstall.properties to the following directory:
   `/opt/CA/IMDataRepository_vertica`*`Version`*

   NOTE
   This path is the default location.
9. Change directories to the location where you extracted the installation scripts:
   `cd /opt/CA/IMDataRepository_vertica`*`Version`*

   NOTE

   This path is the default location.
10. Verify that the parameters in the drinstall.properties file are correct. Review the following parameters:
    - DbAdminLinuxUser=*The Linux user who is created to serve as the Vertica database administrato*r
      Default: dradmin
    - DbAdminLinuxUserHome=*The Vertica Linux database administrator user home directory*
      Default: /export/dradmin
    - DbDataDir=*The location of the data directory*
      Default: /data

      NOTE

      To verify the data directory or catalog directory, complete the following steps:

      1. Open the /opt/vertica/config/admintools.conf file.
      2. Scroll down until you see the [Nodes] section.
      3. Locate one of the lines that begins with v_*dbname*_node####.
         This line contains the IP address of the node, the location of the catalog directory, and the location of the data directory.
         Example:
         `v_drdata_node0001 = 10.42.1.1,/catalog,/data`

    - DbCatalogDir=*The location of the catalog directory*
      Default: /catalog
    - DbHostNames=*The comma-delimited list of hostnames for Data Repository*
      Default: yourhostname1,yourhostname2,yourhostname3
    - DbName=*The database name*
      Default: drdata

      NOTE

      This parameter is case-sensitive.
    - DbPwd=*The database password*
      Default: dbpass

      NOTE

      You can use special characters (except for single quotation marks) in passwords. If the `DbPwd` property is not found or blank, the script prompts for this information at runtime.

    NOTE
    The InstallDestination is no longer used and can be safely removed.
11. Verify that Data Repository is running.

12. On the Data Aggregator host, open a command prompt. Do one of the following steps:
    - Stop the Data Aggregator service:
      ```
      service dadaemon stop
      ```
    - (Fault tolerant environment) If the local Data Aggregator is running, run one the following commands to shut it down and prevent it from restarting until maintenance is complete:
      - **RHEL 6.x:**
        ```
        service dadaemon maintenance
        ```
      - **RHEL 7.x, SLES, or OL:**
        ```
        DA_Install_Directory/scripts/dadaemon maintenance
        ```
13. Run the validation script:
    ```
    ./dr_validate.sh -p drinstall.properties
    ```

    TIP

    You can use the `-l` flag to allow `localhost` as the value for the `DbHostNames` property. You can use the `-n` flag to skip database connectivity checks.

    The script validates the system settings. Review and resolve any errors or warnings. You can run this script multiple times to verify that all system configuration options are set properly. The validation script may prompt you to reboot.
14. Use Vertica adminTools to stop the Data Repository:
    a. Switch to the database admin user.
    b. Open adminTools:
       ```
       /opt/vertica/bin/adminTools
       ```
    c. Select (4) Stop Database.
    d. Select the database, confirm the selection, and provide the password.
    e. Exit the adminTools utility, and revert to the root user or sudo user.
15. Run the installation script:
    ```
    ./dr_install.sh -p drinstall.properties
    ```

    The installation script upgrades the data repository and disables unnecessary Vertica processes. You might be prompted for the Vertica Linux database administrator user password.

    NOTE

    If passwordless SSH is not set up, the script prompts for the password to set up.

    If the installation script returns a WARN message for LVM on directories that Vertica does not use, contact CA Support.
16. Verify that you upgraded Data Repository correctly by doing the following steps:
    a. As the database administrator user, open the Vertica Administration Tools:
       ```
       /opt/vertica/bin/adminTools
       ```
    b. Verify that the top of the banner indicates that the database version is 9.1.1-5.
17. Restart the Data Repository selecting option 3 (Start Database) from the main menu of the Administration Tools dialog.

NOTE

If this is the first time you have stopped the database after the 30 June 2015 leap second, Vertica might fail to start. For more information, see Vertica Fails to Start.

The Data Repository restarts.

Update the Backup

This release includes new functionality for the Data Repository backup. Previous backups and backup configuration files are not compatible with this release. Set up a new version of the backup. For more information, see Back Up the Data Repository.

Upgrade Performance Center

After you upgrade the Data Repository component, upgrade the NetOps Portal component.

NOTE

If you enabled FIPS-compliant encryption, and you upgrade NetOps Portal before the Data Aggregator, temporary FIPS compatibility synchronization errors occur. These errors occur until the Data Aggregator is upgraded. To avoid this temporary condition, upgrade the Data Aggregator before upgrading NetOps Portal.

To host the NetOps Portal database on a separate node, first upgrade to this release, then migrate the database to a separate system.

NOTE
If you have the MySQL database on one node and the Performance Center services on another, run the upgrade on the MySQL database first.

The upgrade requires the zip and unzip packages. If these packages are not installed, use one of the following commands to install them:

```
yum -y install zip unzip
```

SLES:

```
zypper install -y zip unzip
```

NetOps Portal requires the wget package. If this package is not installed, use one of the following commands to install it:

```
yum -y install wget
```

SLES:

```
zypper install -y wget
```

To upgrade NetOps Portal, complete the following procedures:

MySQL Password Requirements (3.7.3 and Higher Only)

As a step toward enhanced security, the Performance Center installation prompts you to set a custom MySQL password.

The MySQL password must meet the following requirements:

- Excludes the user names "root" or "netqos"
- Minimum length of 8 characters
- Maximum length of 30 characters
- Contains at least 3 of the following types of characters:

  - Special Characters (!#&?)
  - Uppercase
  - Lowercase
  - Numbers (0-9)
- Excludes the percentage sign (%), apostrophe ('), or quotation mark (")
- **(3.7.3 through 3.7.4 Only)** Excludes the asterisk (*), or the dollar sign ($)

Verify the ulimit Value

Verify that the user account that is installing CA NetOps Portal has a value of at least 65536 on the number of open files. Set this value permanently.

Follow these steps:

1. As the root user or a sudo user, log in to NetOps Portal host.
2. Open a command prompt and type the following command to verify the ulimit value:
   ```
   ulimit -n
   ```
3. If the value is not at lease 65536, change the value:
   a. Type the following command:
      ```
      ulimit -n 65536
      ```
   b. Open the /etc/security/limits.conf file, and add the following lines:
      ```
      # Added by Performance Center
      * soft nofile 65536
      # Added by Performance Center
      * hard nofile 65536
      ```

Upgrade CA NetOps Portal

The CA NetOps Portal installer includes features to support product upgrades. To perform an upgrade of the software, run the installation file for the new version that you received from CA.

To determine which version you have installed, access the .history file in the /opt/CA/PerformanceCenter/InstallLogs directory.

Follow these steps:

1. Save the installation file to /tmp on the CA NetOps Portal server.

   NOTE

   Verify that your /tmp location has at least 2 GB of space available.
2. Run the following command:
   ```
   cd /tmp
   ```
3. Change the directory permissions:
   ```
   chmod +x CAPerfCenterSetup.bin
   ```
4. Run the following command:
   ```
   ./CAPerfCenterSetup.bin -i console
   ```
   The installation begins in Console mode.

   NOTE

   1. To generate a response file for silent installation, add the following argument:
      ```
      -r response_file
      ```

response_file specifies the directory the directory path and file name for the response file.
Example: /tmp/installer.properties

Follow the prompts until you get to the summary, type "quit", and press Enter.

2. (Optional) On upgrade, the disaster recovery script is backed up. If a backup file already exists, you can overwrite it with the current file. To overwrite the disaster recovery script backup, add the following to the response file:

   ```
   -fileOverwrite_/opt/CA/PerformanceCenter/product_version_capc.sh=Yes
   ```

3. To run the installation in silent mode, use the following command:

   ```
   ./installDA.bin -i silent -f response_file
   ```

   response_file is the directory path and file name of the previously generated response file.

5. Follow the instructions in the console.

   NOTE

   If the current install owner is the `root` user, you are prompted to specify an install owner. You can specify a non-root user.

   The installation checks to see whether the partition with the MySQL data directory has enough disk space to handle storage engine upgrades. If there does not appear to be enough disk space to complete the installation successfully, exit the installer, allocate more space for the data partition, and reinstall NetOps Portal.

   (3.7.3 and Higher Only) As a step toward enhanced security, the Performance Center installation prompts you to set a custom MySQL password.

   WARNING

   When you upgrade from a release before DX NetOps Performance Management 3.6 to release 3.6 or higher, the storage engine is migrated. The upgrade might take longer than expected up to an hour. Do not cancel the installation while it runs.

6. (Optional) When prompted, specify the following parameters for fault tolerance. For more information, see Fault Tolerance.
   - **Configure with Fault Tolerant Data Aggregator**
     If fault tolerance was configured for the Data Aggregators, specify 2 for Yes.
     Default: 1
   - **Data Aggregator Proxy Host**
     Specify the host name/IP address of the proxy server.

   The upgrade operation installs the software. The progress indicator may pause.
   When the upgrade is complete, the console displays a confirmation message.

Reapply Custom MySQL Settings

For upgrades from releases before DX NetOps Performance Management 3.0, this upgrade includes updates to MySQL. These updates require replacing my.cnf. The upgrade backs up the existing my.cnf files as my.cnf.bak. If my.cnf.bak already exists, the backup adds a number to the end of the file name. To retain custom MySQL settings, reapply the settings in the my.cnf files.

Follow these steps:

1. Locate my.cnf and my.cnf.bak in the following directories:
   - /etc/my.cnf
   - /opt/CA/MySql/my.cnf
     This path reflects the default.
2. Copy the relevant setting values from my.cnf.bak to my.cnf.
3. Restart MySQL:

```
service mysql restart
```

Clear Browser Cache

Instruct DX NetOps Performance Management users to clear their browser cache before accessing the upgraded version.

(Optional) Restart the CA NFA OData Service

If you have Network Flow Analysis as a data source, you must restart the CA NFA OData service:

1. Click **Administration Services**, and **Services**.
2. Right-click the CA NFA OData Service.
3. Click **Restart**.

(Optional) Review Log Files

Review the log files in the following directory to track any issues with the upgrade:

Installation Errors and Configuration Events

```
/opt/CA/PerformanceCenter/InstallLogs
```

Verify that the directory contains each log with the install date and time:

- `MySql_*`
- `SSO_*`
- `Device_Manager_*`
- `Event_Manager_*`
- `Console_*`
- `Install_*`

If any logs are missing, the installation is incomplete. Review each log for any fatal errors. Verify that the `.history` file is updated to the latest version.

Migrate the Performance Center Database to a Separate Node

You can isolate the resources used for the database and allow for independent performance tuning by installing the NetOps Portal database on a separate node.

To use a separate MySQL database, upgrade to the latest release, *then* migrate data from the existing database to a separate MySQL database.

> **NOTE**
> In the following steps, *original node* refers to the node which has the single node deployment of NetOps Portal. After you complete these steps, this node hosts only the NetOps Portal services.
>
> *Database node* refers to the new node hosts the externalized database.

1. Run the NetOps Portal installation on the original node to upgrade it to the current release.
2. Run the NetOps Portal installation on the database node. Select **Advanced Installation**, and then install the Database feature.
3. Stop the NetOps Portal services on the original node by running the following commands:

```
service caperfcenter_sso stop
 service caperfcenter_console stop
service caperfcenter_devicemanager stop
service caperfcenter_eventmanager stop
```

4. Dump the NetOps Portal **netqosportal** and **em** databases on the original node by running the following commands:
   ```
   /opt/CA/MySql/bin/mysqldump --routines -u root -pnetqos --databases netqosportal
    > <backupdir>/netqosportal.sql
   /opt/CA/MySql/bin/mysqldump --routines -u root -pnetqos --databases em
    > <backupdir>/em.sql
   ```

5. Transfer the dumped database SQL files to a directory (`backupdir`) on the database node. Run the following commands to import the dumped files into the database:
   ```
   mysql -unetqos -pnetqos -e 'source <backupdir>/netqosportal.sql'
    mysql -unetqos -pnetqos -e 'source <backupdir>/em.sql'
   ```

6. Run the `/opt/CA/PerformanceCenter/Tools/bin/uninstall_database_component.sh` script on the original node.
   This script removes the Database feature from the registry and removes the NetOps Portal `/opt/CA/PerformanceCenter/CAPC_Database_SeedFile.xml` database seed file.
7. Rerun the NetOps Portal installation on the original node.
   When prompted, provide the hostname and port (3306) of the externalized database on the database node.
8. Verify that you can now access NetOps Portal through port 8181 on the original node.
9. Uninstall MySQL from the original node by running the following script:
   `/opt/CA/PerformanceCenter/Uninstall_MySql`

   NOTE
   This procedure does not remove the stored data. To remove the stored data, run the following command:
   ```
   rm -rf /opt/CA/MySql
   ```

Upgrade the Data Aggregator

To upgrade the Data Aggregator, complete the following steps:

NOTE

If you enabled FIPS-compliant encryption, and you upgrade NetOps Portal before the Data Aggregator, temporary FIPS compatibility synchronization errors occur. These errors occur until the Data Aggregator is upgraded. To avoid this temporary condition, upgrade the Data Aggregator before upgrading NetOps Portal.

Verify the Prerequisites

Meet the following prerequisites before you upgrade the Data Aggregator:

- If you have authenticated and encrypted activeMQ communication, you must regenerate the keys before this upgrade. For more information, see Authenticate and Encrypt ActiveMQ Communication.

  NOTE
  For the first and last name prompt, you must enter the host name of the system where you are creating the certification.

  Without new keys, the Data Aggregator and Data Collector cannot communicate.
- Verify that ports 8581, 61616, 61618, 61620, and 61622 are open for communication between your Data Collectors and your Data Aggregators.
  For the detailed port list, Review Installation Requirements and Considerations.
- Before you upgrade, verify the zip and unzip packages are installed. If these packages are not installed, use the following command to install them:

```
yum -y install zip unzip
```

- For the Data Aggregator and the Data Collectors, the default maximum memory is 80% of the total system memory. AMQ uses 20% of total memory on both components. Both components reserve 2 GB of memory for the operating system. These maximum values can be modified, however the modifications are not preserved during an upgrade. For more information, see Modify Maximum Memory Usage for Data Aggregator and Data Collector Components.

Verify the Limit on the Number of Open Files on Data Aggregator

Verify that the user that is installing Data Aggregator has a limit of at least 65536 on the number of open files. Set this value permanently.

Follow these steps:

1. As the root user or a sudo user, log in to the Data Aggregator host.
2. Change the ulimit for the open files limit to at least 65536:
   ```
   ulimit -n ulimit_number
   ```
   For example:
   ```
   ulimit -n 65536
   ```
3. Open the following file:
   ```
   /etc/security/limits.conf
   ```
4. Add the following lines:
   ```
   # Added by Data Aggregator
   * soft nofile 65536
   # Added by Data Aggregator
   * hard nofile 65536
   ```

   NOTE
   Restart Data Aggregator for these changes to take effect. If you are upgrading, the upgrade process automatically restarts Data Aggregator.
5. Verify that the number of open files is set properly:
   ```
   ulimit -n
   ```
   The command returns the limit that you specified earlier.

Verify That the Data Repository Is Running

The Data Aggregator Upgrade requires that the Data Repository is running.

Follow these steps:

1. Log in to the Data Repository host.
2. Verify that the Data Repository is running:
   ```
   /opt/vertica/bin/vsql -U dauser -w dapass -c 'select version()'
   ```
   The expected return shows the version of the Data Repository:
   ```
                version
   -----------------------------------
    Vertica Analytic Database vx.x.x-x
   ```

(Optional) Temporarily Connect to Older Versions of Data Collectors

You can reduce data loss during an upgrade by temporarily connecting existing Data Collectors (version 3.0, 3.1, or 3.2) to an upgraded Data Aggregator. Consider this upgrade option if you have many Data Collectors to upgrade.

> **WARNING**
>
> Important! Connecting 3.x versions of Data Collectors to upgraded Data Aggregators is only supported on a temporary basis.
>
> New features in DX NetOps Performance Management 3.5 may not be compatible with older Data Collectors. Before using new features or making configuration changes, upgrade all Data Collectors to the same version as the Data Aggregator.

Follow these steps before upgrading the Data Aggregator:

1. Create a file called com.ca.im.dm.core.collector.controller.cfg in the following location: /opt/IMDataAggregator/apache-karaf-2.4.3/etc/
2. Add the following single line to the file:
   ```
   allowed-mismatch-dc-version=3.0.0.0
   ```
3. Save the file.
4. Proceed with upgrading the Data Aggregator.

Verify ETL Health

Before you upgrade, run the `etlHealth.sh` script. For releases 20.2.3 and higher, the script is included with the Data Repository installation. For earlier releases, contact Support. We recommend you run this script well in advance of the upgrade and again directly before the upgrade.

Follow these steps:

1. Log in to one of the Data Repository nodes as the root user.
2. Run the validation script:
   ```
   ./etlHealth.sh dauser dapassword
   ```
3. Do one of the following:
   - If the health check passes, proceed with the upgrade.
   - If the health check fails, follow the instructions in the prompt to collect the irep and the data collected by the `etlHealth.sh` script, and submit these details in a Support ticket.

Upgrade the Data Aggregator

To upgrade the Data Aggregator, run the installation as the root or sudo user.

Follow these steps:

1. Log in to the Data Aggregator host as the root user or a sudo user.
2. Copy the installDA.bin file to the /tmp directory.
3. Change to the /tmp directory:
   ```
   cd /tmp
   ```
4. Change permissions for the installation file:
   ```
   chmod a+x installDA.bin
   ```

5. To run the upgrade, do one of the following steps:
   - Run the installation as the root user:
     ```
     ./installDA.bin -i console
     ```
   - Run the installation as the sudo user:
     ```
     sudo ./installDA.bin -i console
     ```

     NOTE

     To generate a response file for silent installation, add the following argument:

     `-r response_file`

     response_file specifies the directory the directory path and file name for the response file.
     Example: /temp/installer.properties

     Follow the prompts until you get to the summary, type "quit", and press Enter.

     To run the installation in silent mode, use the following command:

     `./installDA.bin -i silent -f response_file`

     response_file is the directory path and file name of the previously generated response file.
6. Follow the instructions in the console.

   WARNING

   If the Data Repository was migrated, follow the Data Aggregator upgrade instructions in the console. Answer NO when prompted to drop the schema and the upgrade continues. If you answer YES, you lose your migrated data.

   WARNING

   When you are prompted for the data directory, use the default directory. Do not use `DA_Install_Directory/apache-karaf-version/data`.
7. (Optional) When prompted, specify the following parameters to configure fault tolerance. For more information, see Fault Tolerance.

   WARNING

   If you are upgrading an existing fault tolerant environment, see Upgrade Fault Tolerant Data Aggregators.

   - **Configure Data Aggregator For Fault Tolerance**
     Specify 2 to configure fault tolerance.
     Default: 1

     NOTE

     The default is for a non-fault tolerant environment.
   - **Data Aggregator Proxy Host**
   - Specify the host name/IP address of the proxy server.
   - **Consul HTTP port:**
     Specify the port for communication with Consul.
     Default: 8500
   - **Choose host IP address for Consul**

NOTE

This prompt appears only when multiple public IP addresses are configured.

Specify the bind address that the Consul agents use to communicate with each other. The Consul agents include the proxy host and both Data Aggregators in the cluster. If prompted for an address, specify an address that the other two hosts in the Consul cluster can reach.

Data Aggregator is upgraded. The console displays a confirmation message.
The installer restarts Data Aggregator automatically when the upgrade is complete.
Wait for Data Aggregator to synchronize automatically with CA NetOps Portal. This process can take 30 minutes or longer.

8. Verify that Data Aggregator is running:
   ```
   service dadaemon status
   ```

   IMPORTANT
   Before you upgrade the Data Collectors, after you upgrade the Data Aggregator to the 20.2 GA release, the Polling Status column on the System Status page inaccurately says "Not Connected" for each Data Collector. However, a connection remains. No data loss should occur. After you upgrade the Data Collectors, the correct status appears. This issue will be fixed in releases after the 20.2 GA release.
9. Verify access to the Data Aggregator REST endpoints. Open a Web browser on a computer with HTTP access to Data Aggregator. Navigate to the following address:
   ```
   http://data_aggregator:port/rest
   ```

   data_aggregator:port specifies the Data Aggregator host name and the required port number.
   The return is a list of hyperlinks for available web services. When you click a link, the XML content describing the selection displays.

   WARNING

   Table segmentation is required for systems where the original installation was a release earlier than 2.3.3. Segmentation is a one-time task that improves database performance and reduces the required disk space. If you received a message during the upgrade that the database tables require segmentation, complete table segmentation after the upgrade. For more information, see Segment Database Tables.

   NOTE

   The Karaf log in the Data Aggregator includes the following error after upgrading the installation:

   ```
   ERROR | tenderThread-178 | 2013-01-24 13:36:40,431 |
   ndorCertificationPriorityManager | nager.core.cert-mgr.impl |
         | Failed to load the MetricFamilyVendorPriority for bundle:
   BundleURLEntry [bundle=198, resourceURL=file:/opt/IMDataAggregator/
  apache-karaf-<vers>/data/cache/resources/198--xml-vendorpriorities-
  ReachabilityVendorPriorities.xml
   ```

   This error for Reachability is expected and harmless. Other occurrences of this error are not expected.

Upgrade Fault Tolerant Data Aggregators

In this article:

NOTE
If you have enabled FIPS-compliant encryption, and you upgrade NetOps Portal before the data aggregator, temporary FIPS compatibility synchronization errors occur. These errors occur until you upgrade the data aggregator. To avoid this temporary condition, upgrade the data aggregator before upgrading NetOps Portal.

If you are setting up fault tolerance for the first time, see Fault Tolerance.

Verify the Prerequisites

Meet the following prerequisites before you upgrade the data aggregator:

- Verify that ports 8581, 61616, 61618, 61620, and 61622 are open for communication between the data collectors and the data aggregators.
  For the detailed port list, Review Installation Requirements and Considerations.
- Before you upgrade, verify that the zip and unzip packages are installed. If these packages are not installed, use the following command to install them:
  ```
  yum -y install zip unzip
  ```
- For the data aggregator and the data collectors, the default maximum memory is 80% of the total system memory. AMQ uses 20% of total memory on both components. Both components reserve 2 GB of memory for the operating system. You can modify these maximum values, however the modifications are not preserved during an upgrade. For more information, see Modify Maximum Memory Usage for Data Aggregator and Data Collector Components.
- Ensure that you have a shared data directory (example: `/DASharedRepo`) and that the same user ID is shared between data aggregator hosts. Data from whichever data aggregator is active is stored in this directory.
  For information about the sizing requirements, see the DX NetOps Performance Management Sizing Tool.

  NOTE
  If you are using NFS, only NFS 4 and higher is supported because of the ActiveMQ Kaha locking requirements.

  IMPORTANT
  The shared data directory must be accessible at all times. If the shared data directory is down and is inaccessible, no data is loaded and data loss occurs.

Verify the Limit on the Number of Open Files on the Data Aggregator

Verify that the user that is installing the data aggregator has a limit of at least 65536 on the number of open files. Set this value permanently.

Follow these steps:

1. As the root user or a sudo user, log in to the data aggregator host.
2. Change the ulimit for the open files limit to at least 65536:
   ```
   ulimit -n ulimit_number
   ```
   For example:
   ```
   ulimit -n 65536
   ```
3. Open the following file:
   `/etc/security/limits.conf`
4. Add the following lines:
   ```
   # Added by Data Aggregator
   * soft nofile 65536
   # Added by Data Aggregator
   * hard nofile 65536
   ```

NOTE
Restart the data Aggregator for these changes to take effect. If you are upgrading, the upgrade process automatically restarts the data aggregator.

5. Verify that the number of open files is set properly:
   ```
   ulimit -n
   ```
   The command returns the limit that you specified earlier.

Verify That the Data Repository Is Running

The data aggregator upgrade requires that the data Repository is running.

Follow these steps:

1. Log in to the data repository host.
2. Verify that the data repository is running:
   ```
   /opt/vertica/bin/vsql -U dauser -w dapass -c 'select version()'
   ```
   The expected return shows the version of the Data Repository:
   ```
                 version
   ----------------------------------
    Vertica Analytic Database vx.x.x-x
   ```

(Optional) Temporarily Connect to Older Versions of Data Collectors

You can reduce data loss during an upgrade by temporarily connecting existing data collectors (version 3.0, 3.1, or 3.2) to an upgraded data aggregator. Consider this upgrade option if you have many data collectors to upgrade.

IMPORTANT
Connecting 3.x versions of data collectors to upgraded data aggregators is only supported on a temporary basis.

New features in DX NetOps Performance Management 3.5 might not be compatible with older data collectors. Before using new features or making configuration changes, upgrade all data collectors to the same version as the data aggregator.

Follow these steps:

1. Create a file called `com.ca.im.dm.core.collector.controller.cfg` in the following location: /opt/IMDataAggregator/apache-karaf-2.4.3/etc/
2. Add the following single line to the file:
   ```
   allowed-mismatch-dc-version=3.0.0.0
   ```
3. Save the file.
4. Proceed with upgrading the data aggregator.

Upgrade the Proxy Server

Before you upgrade your existing fault tolerant data aggregators, upgrade the proxy server.

Follow these steps:

1. Uninstall the proxy server. For more information, see Install or Uninstall the Proxy Server.
2. (RHEL 7.x, SLES, and OL only) Run the following command on the proxy server host as the root or sudo user:
   ```
   systemctl daemon-reload
   ```
3. Run the following command on the proxy server host as the root or sudo user:
   ```
   service consul stop
   ```
4. Delete the proxy installation directory:

```
rm -rf daproxy/
```

5. Install the proxy server. For more information, see Install or Uninstall the Proxy Server.

Upgrade Fault Tolerant Data Aggregators

If you are upgrading data aggregators in an existing fault tolerant environment, do the following upgrade steps in the following order:

> **NOTE**
> To view the status of the data aggregators in NetOps Portal, hover over **Administration**, and click **Data Sources: System Status**.

1. Put the inactive Data Aggregator(B) into maintenance mode.
2. Upgrade the inactive Data Aggregator(B).
3. Activate the inactive Data Aggregator(B).
   This Data Aggregator(B) becomes available for failover.
4. Put the active Data Aggregator(A) into maintenance mode.

   > **WARNING**
   > This step causes failover. The previously inactive Data Aggregator(B) becomes the active data aggregator.
5. Upgrade the now inactive Data Aggregator(A).
6. Activate the now inactive Data Aggregator(A).
   This Data Aggregator(A) becomes available for failover. The other Data Aggregator(B) is the active data aggregator.

Turn on Maintenance Mode

> **IMPORTANT**
> If you are upgrading the data aggregators in an existing 3.5 GA fault tolerant environment, put the data aggregator into maintenance mode before upgrade.

For more information, see Fault Tolerance.

Follow these steps:

1. Log in to the data aggregator host as the root user or a sudo user.
2. Run one of the following commands to shut down the inactive data aggregator and prevent it from restarting until the upgrade is complete:
   - **RHEL 6.x:**
     ```
     service dadaemon maintenance
     ```
   - **RHEL 7.x, SLES, or OL:**
     ```
     DA_Install_Directory/scripts/dadaemon maintenance
     ```

Upgrade the Data Aggregator

> **WARNING**
>
> If you are upgrading the data aggregators in an existing 3.5 GA fault tolerant environment, you must put the data aggregator into maintenance mode before upgrade.

Upgrade the inactive data aggregator first and then activate it. Then upgrade and activate the active data aggregator.

Follow these steps:

1. Log in to the data aggregator host as the root user or a sudo user.
2. Copy the `installDA.bin` file to the `/tmp` directory.
3. Change to the `/tmp` directory:

```
cd /tmp
```

4. Change permissions for the installation file:
   ```
   chmod a+x installDA.bin
   ```
5. To run the upgrade, do one of the following steps:
   - Run the installation as the root user:
     ```
     ./installDA.bin -i console
     ```
   - Run the installation as the sudo user:
     ```
     sudo ./installDA.bin -i console
     ```

     NOTE
     To generate a response file for silent upgrade, add the following argument:
     `-r response_file`

     response_file_directory specifies the directory the directory path and file name for the response file.
     Example: /temp/installer.properties

     To run the upgrade in silent mode, use the following command:

     ```
     ./installDA.bin -i silent -f response_file
     ```

     response_file is the directory path and file name of the previously generated response file.
6. Follow the instructions in the console.

   WARNING
   If the Data Repository was migrated, follow the data aggregator upgrade instructions in the console. Answer **NO** when prompted to drop the schema and the upgrade continues. If you answer YES, you lose your migrated data.

   The data aggregator is upgraded. The console displays a confirmation message.
7. Verify access to the data aggregator REST endpoints. Open a Web browser on a computer with HTTP access to the data aggregator. Navigate to the following address:
   `http://DA_Proxy:port/rest`

   DA_Proxy:port Specify the data aggregator proxy host name and the required port number.
   The return is a list of hyperlinks for available web services. When you click a link, the XML content describing the selection displays.

   WARNING
   Table segmentation is required for systems where the original installation was a release earlier than 2.3.3. Segmentation is a one-time task that improves database performance and reduces the required disk space. If you received a message during the upgrade that the database tables require segmentation, complete table segmentation after the upgrade. For more information, see Segment Database Tables.

   NOTE
   The Karaf log in the data aggregator includes the following error after upgrading the installation:

   ```
   ERROR | tenderThread-178 | 2013-01-24 13:36:40,431 |
    ndorCertificationPriorityManager | nager.core.cert-mgr.impl |
         | Failed to load the MetricFamilyVendorPriority for bundle:
    BundleURLEntry [bundle=198, resourceURL=file:/opt/IMDataAggregator/
   apache-karaf-<vers>/data/cache/resources/198--xml-vendorpriorities-
   ReachabilityVendorPriorities.xml
   ```

   This error for Reachability is expected and harmless. Other occurrences of this error are not expected.
8. (Optional) In the NetOps Portal user interface, hover over **Administration**, select **Data Sources: System Status**, and verify the status of the data aggregator.

IMPORTANT
Before you upgrade the data collectors, after you upgrade the data aggregator to the 20.2 GA release, the Polling Status column on the System Status page inaccurately says "Not Connected" for each data collector. However, a connection remains. No data loss should occur. After you upgrade the data collectors, the correct status appears. This issue will be fixed in releases after the 20.2 GA release.

Activate the Data Aggregator

If you upgraded the data aggregators in an existing fault tolerant environment, activate each data aggregator after upgrade.

For more information, see Fault Tolerance.

NOTE
The data aggregator might take several minutes to start.

Follow these steps:

1. Log in to the data aggregator host that is in maintenance mode as the root user or sudo user.
2. Issue one the following commands to enable the fault tolerant data aggregator so that it can start when necessary:
   - **RHEL 6.x:**
     ```
     service dadaemon activate
     ```
   - **RHEL 7.x, SLES, or OL:**
     ```
     DA_Install_Directory/scripts/dadaemon activate
     ```

Upgrade the Data Collectors

After you upgrade the Data Aggregator, upgrade the Data Collectors.

IMPORTANT
Before you upgrade the Data Collectors, after you upgrade the Data Aggregator to the 20.2 GA release, the Polling Status column on the System Status page inaccurately says "Not Connected" for each Data Collector. However, a connection remains. No data loss should occur. After you upgrade the Data Collectors, the correct status appears. This issue will be fixed in releases after the 20.2 GA release.

If you are upgrading from DX NetOps Performance Management 3.5 or higher as the `root` user, you can upgrade the Data Collector easily using the Upgrade button in NetOps Portal. This button is an extra and easier avenue for upgrading your Data Collectors. If you are upgrading from DX NetOps Performance Management 3.6 or higher as a non-root user, you can also use the Upgrade button.

WARNING

To use this feature, the Data Collectors must be running as the `root` user or a user with the necessary `sudo` privileges. If you are upgrading from any 3.5 release as a non-root user, you must upgrade the Data Collectors manually. Also, anyone upgrading from a release before 3.5 must upgrade the Data Collectors manually.

NOTE

For a major release, the user interface indicates that an upgrade is required. For a minor incremental upgrade (for example, a monthly update), the user interface does not indicate that an upgrade is required. However, you can still use the Upgrade button to upgrade to the latest release.

The Data Collectors can continue to run during the upgrade and cache polled data to disk. When the Data Aggregator and Data Repository upgrade completes, the Data Collectors load the cached data. All the Data Collectors are upgraded at once through a staged process where up to two Data Collectors upgrade in parallel. This upgrade method allows you to upgrade your Data Collectors with a minimal reporting gap. Also, an upgraded Data Aggregator (at version 3.5 or

higher) can continue to communicate with the previous versions of your Data Collectors (at version 3.5 or higher) as they upgrade.

NOTE

Before you upgrade the Data Collectors, each Data Collector must have a Tenant and IP Domain assigned. The upgrade process skips any Data Collectors with no Tenant or IP Domain assigned.

WARNING

This upgrade method does not support remote upgrade for Data Collectors that are assigned with a Pseudo Tenant Proxy. For more information, see Tenant-Agnostic Data Collectors.

The upgrade for these Data Collectors might hang with an "Upgrade Pending" status. You must manually upgrade these Data Collectors. If this issue occurs, the Data Collectors continue to collect data as normal and other Data Collectors continue to upgrade.

If you have authenticated and encrypted activeMQ communication, you must regenerate the keys before this upgrade. For more information, see Authenticate and Encrypt ActiveMQ Communication.

NOTE
For the first and last name prompt, you must enter the host name of the system where you are creating the certification.

Follow these steps:

1. If you are running the upgrade as a non-root user, complete the following steps:
   a. Locate the following file on the Data Collector host:
      `/etc/sudoers`
   b. Add the following command alias with the following permissions to the file:
      dcuser dchostname = (root) NOPASSWD: *DC_install_directory*/upgrade/install.bin

      NOTE

      If `requiretty` is enabled, add the following text to disable `requiretty` for the Data Collector installer:

      Defaults!*DC_install_directory*/upgrade/install.bin !requiretty
2. Log in to NetOps Portal as the global administrator.
3. Hover over **Administration**, and click **Monitored Items Management: Data Collectors**.
4. Select all the Data Collectors and click **Upgrade**.
   A confirmation message appears.
5. To proceed with the upgrade, click **Yes**.
   The status of the upgrade appears in the Status column.

If you are upgrading from any 3.5 release as a non-root user, you must upgrade the Data Collectors manually. Also, anyone upgrading from a release before 3.5 must upgrade the Data Collectors manually.

Verify the Prerequisites for a Manual Upgrade

Meet the following prerequisites before you upgrade the Data Collector:

- On the Data Aggregator system, verify that ports 61616, 61618, 61620, and 61622 are open. These ports enable communication between the Data Collector and the Data Aggregator.
- Before you upgrade, verify the zip and unzip packages are installed. If these packages are not installed, use the following command to install them:
  ```
  yum -y install zip unzip
  ```
- The Data Collector uses the "at" package to schedule the restart of the application when assigning a Tenant or IP Domain. Verify whether the "at" package is installed on each Data Collector host:

```
rpm -qa | grep ^at
```

If the command does not return a result, install the package:

NOTE

If you are not the root user, use the sudo prefix.

```
yum install at
```

Upgrade the Data Collector Manually

To upgrade the Data Collector manually, run the installation as the root or sudo user.

Follow these steps:

1. Log in to the Data Collector host as the root user or a sudo user.
2. Change to the /tmp directory:
   ```
   cd /tmp
   ```
3. Use `wget` , `curl` , or a browser to download the installation package to the /tmp directory:
   ```
   wget http://data_aggregator:port/dcm/InstData/Linux/VM/install.bin
   curl -o /tmp/install.bin http://data_aggregator:port/dcm/InstData/Linux/VM/
   install.bin
   ```
   Default port: 8581
4. Change permissions for the installation file:
   ```
   chmod a+x install.bin
   ```
5. To run the upgrade, do one of the following steps:
   - Run the installation as the root user:
     ```
     ./install.bin -i console
     ```
   - Run the installation as the sudo user:
     ```
     sudo ./install.bin -i console
     ```

     NOTE

     To generate a response file for silent installation, add the following argument:

     ```
     -r response_file
     ```

     response_file specifies the directory the directory path and file name for the response file.
     Example: /temp/installer.properties

     Follow the prompts until you get to the summary, type "quit", and press Enter.

     To run the installation in silent mode, use the following command:

     ```
     ./installDA.bin -i silent -f response_file
     ```

     response_file is the directory path and file name of the previously generated response file.
6. Follow the instructions in the console.
7. (Optional) When prompted, specify the following parameters for fault tolerance. For more information, see Fault Tolerance.
   - **Is the Data Aggregator configured with fault tolerance?**
     If fault tolerance was configured for the Data Aggregators, specify 2 for Yes.
     Default: 1
   - **Inactive Data Aggregator Host/IP Address**
     Specify the host name/IP address of the inactive Data Aggregator responsible for this Data Collector.

NOTE
To view the status of your Data Aggregators in NetOps Portal, hover over **Administration**, and click **Data Sources: System Status**.

8. When the upgrade is complete, the console displays a confirmation message.
9. Verify that Data Collector is running:
   ```
   service dcmd status
   ```

Verify a Manual Upgrade

After you upgrade the Data Collectors manually, verify the upgrade.

Follow these steps:

1. Log in to NetOps Portal as the global administrator.
2. Navigate to the Data Aggregator data source.
3. Expand **System Status**, and click **Data Collectors**.
4. Verify that Data Collector appears in the list.

NOTE
The list can take several minutes to refresh and show the new Data Collector installation.

Complete the Upgrade

To complete the upgrade, complete the following steps:

Re-Enable the Automatic Recovery of the Data Aggregator Process

Re-enable the automatic recovery of the Data Aggregator process. You disabled the automatic recovery before you upgraded Data Aggregator. When enabled, if the database server runs out of memory, or if Data Repository is unavailable for a time, Data Aggregator shuts down automatically to help ensure that data consistency is maintained.

NOTE

In a fault tolerant environment, this procedure is unnecessary because Consul manages the start and stop state of the Data Aggregator. For more information, see Fault Tolerance.

Follow these steps:

1. Log in to the computer where the Data Aggregator is installed as the root user.
2. Open a console and type the following command:
   ```
   crontab -e
   ```
   A vi session opens.
3. Uncomment out the following line by removing the pound symbol (#) from the beginning of the following line:
   ```
   # * * * * * /sbin/service dadaemon start > /dev/null
   ```
   For example:
   ```
   * * * * * /sbin/service dadaemon start > /dev/null
   ```
   The automatic recovery of the Data Aggregator process is re-enabled.

Perform Post-Upgrade Configuration

Do the following steps after you upgrade DX NetOps Performance Management:

- If you use the Java Cryptography Extension (JCE), upgrade the JCE to version 8.

NOTE
For the Java 8 version of the JCE, visit the Oracle site.

- Upgrading Data Aggregator backs up the /opt/IMDataAggregator/apache-karaf-<*vers*> directory to the /opt/IMDataAggregator/backup/apache-karaf directory. Customizations that are located in the /opt/IMDataAggregator/apache-karaf-<*vers*>/etc/ directory, such as default logging levels or other configurations are backed up, but are not restored to the installation directory automatically. To avoid losing these customizations, restore the customizations manually after you have successfully upgraded.
- Upgrading Data Collector backs up the /opt/IMDataCollector/apache-karaf-<*vers*> directory to the /opt/IMDataCollector/backup/apache-karaf directory. Customizations that are located in the /opt/IMDataCollector/apache-karaf-<*vers*>/etc/ directory, such as default logging levels or other configurations are backed up, but are not restored to the installation directory automatically. To avoid losing these customizations, restore the customizations manually after you successfully upgraded.

Vendor Certification Priorities

New vendor certifications are placed at the bottom of the Vendor Certification Priorities list for the corresponding metric family. To take advantage of the new vendor certifications, manually change the vendor certification priorities.

For example, F5 CPU vendor certifications are modeled as normal CPUs but do not get discovered because F5 also supports Host Resources. After an upgrade, the Host Resources CPU priority entry is higher than the F5 entries appended to the end of the priority list. To discover F5 CPU devices and components, update the vendor certification priority for the CPU metric family.

CA Business Intelligence Integrations

If you have an existing CA Business Intelligence (CABI) integration, this version of DX NetOps Performance Management supports CABI 6.4.2 and higher.

WARNING

After you upgrade to a supported CABI version, you must re-install your CABI content. For more information, see Install CA Business Intelligence Reports and Dashboards.

(Optional) Disable the ActiveMQ Admin Console for the Data Aggregator or Data Collector

Generally, the ActiveMQ admin console should not be available on the network. Therefore, you can disable it for the Data Aggregator or Data Collector.

Follow these steps:

1. Go to one of the following files:
   - **Data Aggregator**
     `DA_Install_Directory/broker/apache-activemq-version/conf/activemq.xml`
   - **Data Collector**
     `DC_Install_Directory/broker/apache-activemq-version/conf/activemq.xml`
2. Comment out
   ```
   <import resource="jetty.xml"/>
   ```
   .
3. Shut down the ActiveMQ broker on each Data Collector:
   ```
   service activemq stop
   ```
4. Shut down the ActiveMQ broker on the Data Aggregator:
   ```
   service activemq stop
   ```
5. Start the ActiveMQ broker on the Data Aggregator:

```
service activemq start
```

If you do not, the Data Aggregator starts the broker automatically.

6. The Data Collectors automatically restart the ActiveMQ brokers. Use the following command to restart the brokers manually:

```
service activemq start
```

(Optional) Update ActiveMQ Admin Console Access

Generally, the ActiveMQ admin console should not be available on the network. However, if certain users absolutely need the console, you can grant them access.

Follow these steps:

1. Go to one of the following files:
   - **Data Aggregator**
     DA_Install_Directory/broker/apache-activemq-*version*/conf/activemq.xml
   - **Data Collector**
     DC_Install_Directory/broker/apache-activemq-*version*/conf/activemq.xml
2. To update user access, edit the
   `jetty-realm.properties`
   .
3. To encrypt the user passwords, run one of the following commands:
   - **Data Aggregator**

     ```
     java -cp DA_Install_Directory/broker/apache-activemq-version/lib/web/jetty-all-9.2.22.v20170606.jar org.eclipse.jetty.util.security.Password passwordpassword
     ```
   - **Data Collector**

     ```
     java -cp DC_Install_Directory/broker/apache-activemq-version/lib/web/jetty-all-9.2.22.v20170606.jar org.eclipse.jetty.util.security.Password passwordpassword
     ```
4. Shut down the ActiveMQ broker on each Data Collector:

   ```
   service activemq stop
   ```
5. Shut down the ActiveMQ broker on the Data Aggregator:

   ```
   service activemq stop
   ```
6. Start the ActiveMQ broker on the Data Aggregator:

   ```
   service activemq start
   ```

   If you do not, the Data Aggregator starts the broker automatically.
7. The Data Collectors automatically restart the ActiveMQ brokers. Use the following command to restart the brokers manually:

   ```
   service activemq start
   ```

Rehydrating Data in a Cloud Environment

If you have DX NetOps Performance Management set up in a cloud environment, you can patch operating systems from a common image instead of patching each operating system individually. The following procedures outline the necessary steps for rehydrating the DX NetOps Performance Management nodes with minimal data loss.

Verify the Prerequisites

Before rehydration, ensure your environment is in a good state.

Follow these steps:

1. Select **Administration**, **Data Source Settings**, and **System Status**.
2. Verify that the Data Aggregator and Data Repository are connected.
3. Verify that the Data Aggregator is up and running.
4. View the System Status page to verify there is no backed up or cached poll data.

   NOTE
   If the PRQ queue is not empty, the Data Collectors need to send the rest of the polled data to the Data Aggregator. The queue fills up when an outage occurs, which causes data to be cached on the Data Collectors. View the System Status page to verify that all the Data Collectors have a green status. The Polling Status column shows whether cached values exist on the Data Collector.
5. If you have DX NetOps Virtual Network Assurance in your environment, do the following steps:
   a. In , hover over **Administration**, and click **Monitored Items Management: VNA Gateways**.
   b. Set the **Administrative Status** to **Down**.

Rehydrate Each Data Collector

Rehydrate each Data Collector one at a time. Make sure each Data Collector recovers and starts polling before you rehydrate each Vertica node and the Data Aggregator. During this process, some polls are cached on the Data Collectors for a time.

Follow these steps:

1. Build the new operating system on the Data Collector container or virtual machine.
2. Copy the `DCM_ID` :
   ```
   grep "manager\-id\=" DC_Install_Directory/apache-karaf-<version>/etc/
   com.ca.im.dm.core.collector.cfg
   ```
3. Bring down the old container or virtual machine.
4. Give the new container or virtual machine a new IP address or name and bring it online.
5. Reinstall the Data Collector with the `DCM_ID` of the original Data Collector.
   ```
   export DCM_ID="Original_DC_Host:DCM_ID"
   cd /tmp;
   rm -rf install.bin;
   wget http://DA_Host:Port/dcm/InstData/Linux/VM/install.bin;
   chmod a+x install.bin;
   ./install -i silent
   ```
   The Data Collector installs, reconnects to the Data Aggregator, and starts polling.

Rehydrate Each Vertica Node

Rehydrate each Vertica node one at a time. Before you start, verify that all nodes are up and running:

```
/opt/vertica/bin/admintools -t list_allnodes
```

Follow these steps:

1. Bring down the Vertica node:
   ```
   /opt/vertica/bin/admintools -t stop_node -s IP_Address
   ```
2. Unmount the `data` directory and the `catalog` directory.
3. Create a new node with the same IP address and name.
4. Mount the `data` directory and the `catalog` directory to the new node.
5. Run the validation script:
   ```
   ./dr_validate.sh -n -p drinstall.properties
   ```

The script validates the system settings. Review and resolve any errors or warnings. You can run this script multiple times to verify that all system configuration options are set properly. The validation script may prompt you to reboot.

6. Install Vertica from an up and running node:
   ```
   /opt/vertica/sbin/install_vertica -u dradmin -l /export/dradmin -d /export/data -L ./
   resources/vlicense.dat -Y -r ./resources/vertica-<version>.rpm
   ```

   NOTE
   Values should match those in the properties files for `dr_install.sh` , and point to the same resources.
7. Start the node and verify that it is up and running:
   ```
   /opt/vertica/bin/admintools -t restart_node -s Host_Name -d DB_Name
   ```

   NOTE
   The state starts as DOWN, then changes to REBUILDING until it changes to UP.
8. Repeat this procedure for each node.
9. After all nodes are rehydrated, verify that all nodes are back up and running as the `dradmin` user:
   ```
   /opt/vertica/bin/admintools -t list_allnode
   ```

Rehydrate the Data Aggregator

During this Vertica refresh, the Data Aggregator should collect data from the Data Collectors. The Data Aggregator pushes data to Vertica the entire time on the up and running nodes. The speed of the ingestion is sometimes cut in half during this time.After you rehydrate the Data Collectors and Vertica, you can rehydrate the Data Aggregator.

Follow these steps:

1. Prepare a new node for the Data Aggregator.
2. Do one of the following steps:
   - Stop the Data Aggregator service:
     ```
     service dadaemon stop
     ```

     NOTE
     For RHEL 7.x, `service` invokes `systemctl` . You can use `systemctl` instead.
   - (Fault tolerant environment) If the local Data Aggregator is running, run one the following commands to shut it down and prevent it from restarting until maintenance is complete:
     - **RHEL 6.x:**
       ```
       service dadaemon maintenance
       ```
     - **RHEL 7.x, SLES, or OL:**
       ```
       DA_Install_Directory/scripts/dadaemon maintenance
       ```

   The Data Aggregator completes processing and the service stops.
3. Move the configuration files to the new node. For more information, see Back Up Data Aggregator.

   NOTE
   In environments with fault tolerant Data Aggregators, use the shared data directory and reattach it. For more information, see Fault Tolerance.
4. Do one of the following steps:
   - Start the Data Aggregator service:
     ```
     service dadaemon start
     ```
   - (Fault tolerant environment) Run one the following commands to enable the fault tolerant Data Aggregator so that it can start when necessary:
     - **RHEL 6.x:**

```
service dadaemon activate
```

- **RHEL 7.x, SLES, or OL:**:

  DA_Install_Directory/scripts/dadaemon activate

The Data Aggregator consumes the queued polls and pushes them to Vertica.

> **NOTE**
> Depending on the total outage time, all cached data is consumed and ready for reporting in approximately two times the outage time. When the ActiveMQ consumption returns to normal, this indicates there is no longer a backlog. View the System Status page to verify that all the Data Collectors have a green status and that the system is receiving approximately the same number of polls as it was before the process started.

Rehydrate NetOps Portal

Finally, you can rehydrate NetOps Portal.

Follow these steps:

1. Prepare a node for NetOps Portal.
2. Bring down NetOps Portal:

   > **NOTE**
   > For RHEL 7.x and higher or OL, `service` invokes `systemctl` . You can use systemctl instead.

   ```
   service caperfcenter_console stop
   service caperfcenter_devicemanager stop
   service caperfcenter_eventmanager stop
   service caperfcenter_sso stop
   ```
3. Move the database to the new node. For more information, see Back Up Performance Center.
4. Start NetOps Portal:
   a. Start the SSO service:
      ```
      service caperfcenter_sso start
      ```
   b. Wait one minute, then start the event manager and device manager:
      ```
      service caperfcenter_eventmanager start
      service caperfcenter_devicemanager start
      ```
   c. Wait one minute, then start the console service:
      ```
      service caperfcenter_console start
      ```

Rehydrate DX NetOps Virtual Network Assurance

If you have DX NetOps Virtual Network Assurance in your environment, rehydrate it now.

Follow these steps:

1. Query the following REST URL to find the engine ID required later:
   http://*VNA_host*:8080/vna/rest/v1/admin/engines
2. Query the following REST URL for your plug-in configuration required later:
   http://*VNA_host*:8080/vna/rest/v1/admin/engines/*Engine_ID*/config
3. Stop WildFly using one of the following commands:
   ```
   service wildfly stop
   systemctl stop wildfly
   ```
4. Back up the existing database:
   VNA_Install_Directory/tools/bin/db_backup.sh *Backup_File_Name*
5. Install DX NetOps Virtual Network Assurance on the new server and restore the database from the backup:

```
VNA_Install_Directory/tools/bin/db_restore.sh Backup_File_Name
```

6. Reconfigure the plug-ins using the information from your original query.

   NOTE
   Use the same Domain ID from the original configuration.

7. In NetOps Portal, hover over **Administration**, and click **Monitored Items Management: VNA Gateways**. Change the new DX NetOps Virtual Network Assurance server ID. Set the **Administrative Status** to **Up**.

Reconnect an Existing DX NetOps Spectrum Data Source

If you rehydrate the DX NetOps Performance Management environment, and want to reconnect it to an existing Spectrum server, complete the following procedure.

Follow these steps:

1. Go to **Administration**, **Data Sources**, **Data Sources**.
2. Select the DX NetOps Spectrum data source, and click **Edit**.
3. Change the Status to Disabled, and click **Save**.
4. Remove the NetOps Portal entries:

```
cd Spectrum_Install_Directory/vnmsh
./connect
./show models | grep CAPC
```

5. Run the following command for every `CAPCIPDomain` and `CAPCTenant` model found.

```
./destroy model mh=0xXXXXX
```

6. Remove the NetOps Portal integration database from DX NetOps Spectrum:

```
bash -login
cd mysql
cd bin
./mysqladmin --defaults-file=../my-spectrum.cnf -unetqos -ppassword drop netqos_integ
```

7. Restart DX NetOps Spectrum tomcat:

```
cd Spectrum_Install_Directory/tomcat/bin
./stopTomcat.sh
./startTomcat.sh
```

8. Go to **Administration**, **Data Sources**, **Data Sources**.
9. Select the DX NetOps Spectrum data source, and click **Edit**.
10. Change the Status to Enabled, and click **Save**.

Rehydrate Network Flow Analysis

If you have Network Flow Analysis in your environment, rehydrate it now.

Follow these steps:

1. Go to **Administration**, **Data Sources**, **Data Sources**.
2. Select the Network Flow Analysis data source, and click **Edit**.
3. Change the Status to Disabled, and click **Save**.
4. Determine the database files to backup.
   - Customized data_retention database (Stand-alone or Harvester server): `data_retention`
   - harvester database (Stand-alone or Harvester server): `harvester`
   - reporter database (Stand-alone or NFA console): `reporter`
5. Copy each of the target directories or files to a remote location.

6. Back up the following databases to a remote location, using `mysqldump` . Back up the `reporter` database last, regardless of the deployment architecture.
   - Customized data_retention database (Stand-alone or Harvester server): `data_retention`
   - harvester database (Stand-alone or Harvester server): `harvester`
   - reporter database (Stand-alone or NFA console): `reporter`

   ```
   mysqldump --routines --events -u root dbname --skip-lock-tables > dbbackupname.sql
   ```
7. (Optional) Verify that the `mysqldump` was successful by checking that the size of the backup is over 1 KB.
8. Restore each of the target directories or files from its remote location to its original location.
9. Restore each of the databases. Restore the reporter database first, regardless of the deployment architecture.
   - reporter database: `reporter` (Stand-alone or NFA console)
   - Customized data_retention database: `data_retention` (Stand-alone or Harvester server)
   - harvester database: `harvester` (Stand-alone or Harvester server)

     IMPORTANT
     For best results, restore to a clean installation.

   ```
   mysql –e “drop database DB_Name;”
   mysql -e "create database DB_Name;"
   mysql -u root DB_Name dbbackupname.sql
   mysql -u root mysql > proc.sql
   ```
10. Go to **Administration**, **Data Sources**, **Data Sources**.
11. Select the Network Flow Analysis data source, and click **Edit**.
12. Change the Status to Enabled, and click **Save**.

Building

To provide data for dashboards and views, DX NetOps Performance Management collects data from devices in your network. Building includes information about device management, discovery, and self-certification, and other topics that relate to how DX NetOps Performance Management collects data.

Self-Certification

Self-certification is the process of updating or creating certifications to support new vendor devices and technology types. DX NetOps Performance Management uses metric families and vendor certifications to support devices. These components determine how DX NetOps Performance Management collects configuration and operational metrics for a device.

Out of the box, DX NetOps Performance Management supports the common vendors, metrics, and components in your network infrastructure. The Technology Certification Portal lists out-of-the-box certifications by data aggregator version, vendor certification, and metric families: http://serviceassurance.ca.com/im/

For unsupported devices, extend monitoring capabilities using self-certification.

TIP
When possible, use an extension instead of a new vendor certification or metric family. When CA Technologies updates the certification or metric family, extended certifications are also updated.

WARNING
Changes to metric families and vendor certifications apply to all tenants.

Types of Self-Certification

Basic Self Certification: Create a custom vendor certification in the UI, but with limitations. You cannot use basic self-certification after using another method. However, you can update vendor certification expressions that you created using the custom certification process in the UI.

Custom Certification: Create a custom vendor certification, metric family, or component.

Extend: Modify an out-of-the-box vendor certification or metric family. Extending a vendor certification creates an XML file, but keeps a backup of the factory certification. Changes to extended metric families or vendor certifications are maintained when the original metric family or vendor certification is updated.

Update: Apply changes to a custom certification. Updating a vendor certification completely replaces the existing XML file.

> **NOTE**
> Some UI features, such as configuring percentiles or projections, automatically extend or update the certification.

When to Self-Certify

Self-certification applies in the following situations:

- To support an existing metric family on a device that DX NetOps Performance Management does not yet support, create a custom vendor certification.
- To support a new vendor or device, create a custom metric family and a custom vendor certification.
- To support a new technology in DX NetOps Performance Management, create a custom component, custom metric family, or a custom vendor certification.
- To add a metric to an out-of-box vendor certification, extend the metric family or extend the vendor certification.
- To change the name of a metric, the calculation of a metric for an out-of-box vendor certification, or to extend the metric family.
- To add custom discovery filtering to an out-of-box vendor certification, extend the vendor certification.
- To change the OID that a device uses, extend the vendor certification.
- To change polled and baseline configuration for metrics in an out-of-box metric family, extend the metric family.

Self-Certification Prerequisites

To ensure a successful self-certification process, verify that you meet the following prerequisites:

- You have access to the MIB and OIDs from the device.
- You have a MIB browser, such as the free version of iReasoning.
- You have a test environment for the certifications.

> **WARNING**
> Metric families and vendor certifications cannot be deleted from your system. Test custom certifications before you implement the changes in your production environment.

The Data Model

Understanding the data collection model helps you understand what is required in the self-certification process.

Data Aggregator supports devices using the following configuration features:

- **Discovery Profile:** Determines which items Data Aggregator discovers in your environment, typically based on a range of IP addresses. The discovery process identifies the "type" for each item that it finds.
- **Device Collections:** Organizes your inventory into groups of related items. Items are automatically added to a device collection based on the item type and IP address.
- **Monitoring Profile:** Controls the polling rate for a device collection, and determines which metric families to poll. Monitoring profiles can poll one or more metric families. To poll the same metric family at different rates, and on different groups, add the same metric family to more than one monitoring profile.

  NOTE
  To ensure that the system is not overloaded with polling traffic, use monitoring profiles to adjust the polling rate for different sets of metrics.
- **Metric Family:** Controls which metrics are gathered for a monitoring profile. Metric families are associated with one or more vendor certifications, which are listed in priority order.
- **Vendor Certification:** Maps attributes from a vendor MIB to the metrics in a metric family. Also determines how metrics that are collected from an item are formatted. Metrics that are provided for an item can vary, depending on the item vendor. Mapping these values ensures that the metric values are reported consistently, regardless of the vendor. Multiple vendor certifications can be associated with a single metric family. In such cases, Data Aggregator maps metric values using a ranked list of vendor certifications. The data aggregator calculates a metric value using the highest-priority vendor certification that matches the polled item.

  NOTE
  MIBs can be imported into the system and compiled as part of building a vendor certification. Import MIBs only using the basic certification process. They are not required for polling.

Example: Support for a Router Device

When running your discovery profile, the data aggregator finds and identifies an item as a router. The router managed item is automatically added to the All Routers device collection. This device collection is associated with the Routers monitoring profile. This profile uses the CPU and Memory metric families to discover the CPU and Memory components on the device. These metric families also determine the vendor certification to use when calculating the metric values for these components. Based on this monitoring profile, the data aggregator polls the router every 5 minutes. The vendor certifications that are associated with a metric family determine how to calculate and format the raw metric data. DX NetOps Performance Management stores the collected metric data for your router, and uses the data in dashboards and reports.

Self-Certification Workflow

To successfully complete a self-certification, do the following procedures in the indicated order:

1. Create Custom Components.
   Skip this step if you have existing components.
2. Create or Extend Metric Families.
   Skip this step if you have existing metric families.
3. Create or Extend Vendor Certifications.
4. Manage Vendor Certification Priorities.

Create Custom Components

To support a new technology, create a custom component. You can create a custom component using REST Web Services. You can create an entry that lists all instances of your component in the Inventory menu and create a Context page by creating a custom component. Use a component to create a REST endpoint, or to create attributes that are shared between metric families.

NOTE
You cannot extend factory components.

WARNING
To avoid possible data loss, back up the certification directory before you create or update a custom component.

Create a Component XML Template

Use an existing component to create an XML template.

TIP
To retrieve a list of existing components, go to the following REST URL:
http://*da_hostname*:8581/typecatalog/components

Follow these steps:

1. Set up a REST client with a connection to the data aggregator server.
2. Select a component that is similar to the component that you require.
3. Enter the following URL to retrieve the template component:
   `http://da_hostname:8581/typecatalog/components/component_name`
   component_name specifies the name of the template component.
4. Select **GET** in the Method tab, and run the method.
   The REST client returns the XML information for the component.
   Use the XML as a template to create the custom component.

Edit the Component XML

To apply the necessary changes to the component, edit the XML file (the component XML). Make changes to the `<ItemSyncDefinition>` section to create an entry that lists all instances of your component in the Inventory menu and to create a Context page for your component.
For more information about the XML structure, see Component XML Structure.

Import the Component XML

To import the component, use a REST client.

Follow these steps:

1. Specify http://*da_hostname*:8581/typecatalog/components as the URL.
2. On the Method tab, Select **POST** for a new custom component, or select **PUT** to update a component.
3. In the Body settings, set **application/xml** as the **Body Content-type**.

   WARNING
   Failing to set the Content-type results in a 415 error.
4. Copy the component XML into the Body tab.
5. Run the method.
   Your custom component is imported. If no errors occur, the Status field in the HTTP Response section displays the following result:
   `HTTP/1.1 200 OK`

Create or Extend Metric Families

If an out-of-the-box metric family does not fit your monitoring needs, extend an existing metric family or create a custom metric family. If an existing metric family is close to what you need, extend that metric family. If you need something new, create a custom metric family. For an example, see Add a New Metric to an Existing Metric Family.

Use the following procedures to create, extend, or update metric families:

WARNING
To avoid possible data loss, always back up the certification directory anytime you extend, create, or update a metric family.

Create a Metric Family XML Template

For a new metric family, use an existing metric family to create an XML template. For an extension, get the XML for the target metric family.

TIP

To retrieve the name of the metric family in the UI, go to the Metric Families page. Click the downward arrow on any column, hover over **Columns**, and click **Internal Name**.

Follow these steps:

1. Set up a REST client with a connection to the Data Aggregator server.
2. Enter the following URL to retrieve the template metric family:
   - **New Metric Family:** `http://da_hostname:8581/typecatalog/metricfamilies/mf_name`
   - **Extension:** `http://da_hostname:8581/typecatalog/metricfamilies/mf_name` **mf_name** specifies the name of the template metric family.
3. Select GET in the Method tab, and run the method.
   The REST client returns the XML information for the metric family.
   Use the XML as a template to create the custom or extended metric family.

Edit the Metric Family XML

To apply the necessary changes to the metric family, edit the XML file. For complete information about the XML structure, see Metric Family XML Structure.

WARNING

If the Units attribute is not defined in the XML, the units label on reports is 'Units'.

NOTE

When you extend a metric family, include only the XML nodes that require changes.

Import a Metric Family

Select one of the following options to import a metric family:

Use a REST Client to Import a Metric Family

To import a metric family, you can use a REST client.

Follow these steps:

1. Specify one of the following URLs:
   - **Custom Metric Family:** `http://da_hostname:8581/typecatalog/metricfamilies/`
   - **Extend Metric Family:** `http://da_hostname:8581/typecatalog/metricfamilies/extension/mf_name`
2. On the **Method** tab, Select **POST** for a new custom metric family, or select **PUT** to update or extend a metric family.
3. In the **Body** settings, select 'application/xml' as the 'Body Content-type'.

WARNING
Failing to set the Content-type results in a 415 error.

4. Copy the metric family XML into the **Body** tab.
   Run the method.
   Your custom metric family is imported. If no errors occur, the Status field in the HTTP Response section displays the following result:
   ```
   HTTP/1.1 200 OK
   ```

Import a Metric Family in the UI

To import a metric family, you can use the UI.

Follow these steps:

1. Click **Metric Families** from the **Monitoring Configuration** menu for a Data Aggregator data source.

   NOTE

   Use the Search feature in any pane to locate specific information that is related to that pane. Alternatively, navigate between pages in a pane using the arrows.
2. Click **Import**.

   NOTE

   The Import button supports ZIP files. For example, you can import the downloaded ZIP file for a certification from the On-Demand Certification support page.
3. Click **Browse**, and select the metric family file.
4. Click **Open**, and then click **Import**.
   Your metric family is imported.

Update Metric Family Properties

To change attributes, such as the display name, update the metric family properties.

Follow these steps:

1. Specify the following URL:
   - **Extend Metric Family:** `http://da_hostname:8581/typecatalog/metricfamilies/extension/mf_name`
2. On the **Method** tab, select **PUT** to update or extend a metric family.
3. In the **Body** settings, select **application/properties** as the **Body Content-type**.
4. Specify updates with one line per property in the **Body** tab as illustrated in the following template and example.
   Template:

   NOTE

   The ***metricfamilyname*** *and* ***attributename*** variables in the following example are case-sensitive and are lowercase. These variables are a different case from their values in the XML. For example, **NormalizedPortInfo** from the XML is **normalizedportinfo** here and **PctDiscardsIn** in the XML is **pctdiscardsin** here.

```
im.ca.com.normalizer.metricfamilyname.displayname=DisplayName
im.ca.com.normalizer.metricfamilyname.documentation=Documentation text
im.ca.com.normalizer.metricfamilyname.attribute.attributename.attributedisplayname=AttributeI
```

```
im.ca.com.normalizer.metricfamilyname.attribute.attributename.documentation=Documentation
 text
```

Example:

```
im.ca.com.normalizer.normalizedportinfo.displayname=Interface
im.ca.com.normalizer.normalizedportinfo.documentation=Defines the identification
 information, configuration information, and polled metrics for interfaces.
im.ca.com.normalizer.normalizedportinfo.attribute.pctdiscardsin.attributedisplayname=Percent
 Discards In
im.ca.com.normalizer.normalizedportinfo.attribute.pctdiscardsin.documentation=The
 percentage of the frames (packets) received by the interface that were discarded.
```

5. Run the method.

Trigger Rediscovery

After you extend a metric family, the changes occur during the nightly automatic rediscovery. If the changes do not apply automatically, trigger the update manually.

WARNING

To avoid a severe impact on performance, do not trigger a rediscovery during normal business hours.

Follow these steps:

1. Navigate to the **Data Aggregator** data source.
2. Click **Monitoring Configuration**, **Metric Families**.
3. Select the metric family.
4. Click **Update Metric Family**.
   The Data Aggregator rediscovers components on all devices that support the selected metric family.

Verify the Metric Family Results

To ensure successful operation, verify the results of the import.

Follow these steps:

1. Log in to the UI, and navigate to the Data Aggregator data source.
2. Go to Monitoring Configuration, Metric Families.
3. Verify that the metric family in the list and that the Last Modified time has been updated.

Create or Extend Vendor Certifications

If you have a new vendor or device that you want to support, create a vendor certification. If you want to change the way that a metric is calculated, or to add custom discovery filtering for an out-of-box vendor certification, extend the vendor certification. To remove a vendor certification extension, import an unchanged vendor certification extension template.

WARNING

To avoid possible data loss, always back up the certification directory anytime you extend, create, or update a vendor certification. Metric families and vendor certifications cannot be deleted from your system. Test custom certifications before you implement the changes in your production environment.

Get a Vendor Certification XML Template

For a new certification, use an existing vendor certification for a similar device to create an XML template. For an extension, get the XML for the target vendor certification.

TIP

To find a suitable vendor certification to use as a template, look at the metric family you want to support with the new certification, and pick one that is similar to your device.

Follow these steps:

1. Set up a REST client with a connection to the Data Aggregator server.
2. Enter the following URL to retrieve the template vendor certification:
   - **New Certification:** *http://da_hostname:8581/typecatalog/certifications/snmp/cert_name*
   - **Extension:** *http:*
     cert_name specifies the name of the vendor certification, which is an attribute of the FacetType tag.
3. Select GET in the Method tab, and run the method.
   The REST client returns the XML information for the vendor certification.
   Use the XML as a template to create the custom or extended vendor certification.

Edit the Vendor Certification XML

To apply the necessary changes, edit the XML file. When you extend a vendor certification, include only the XML nodes that require changes. For complete information about the XML structure, see Vendor Certification XML Structure.

Import the Custom Vendor Certification

Import the vendor certification to the system.

TIP

Verify the vendor certification in a test environment before you import the certification to your production environment. You cannot delete a vendor certification.

Select one of the following options to import the vendor certification:

Use a REST Client to Import a Vendor Certification

To import a single vendor certification, you can use a REST client.

Follow these steps:

1. Specify one of the following URLs:
   - **Import Custom Vendor Certification:** *http://da-hostname:8581/typecatalog/certifications/snmp*
   - **Update Custom Vendor Certification:** `http://da-hostname:8581/typecatalog/certifications/snmp/cert_name`
   - **Extend Vendor Certification:** `http://da-hostname:8581/typecatalog/certifications/snmp/extension/cert_name`
2. On the **Method** tab, select **POST** to import a custom vendor certification, or select **PUT** to update or extend a certification.
3. In the **Body** settings, select 'application/xml' as the 'Body Content-type'.

WARNING
Failing to set the Content-type results in a 415 error.

4. Copy the vendor certification XML into the Body tab.
5. Run the method.
   Your vendor certification is imported. If no errors occur, the Status field in the HTTP Response section displays the following result:
   ```
   HTTP/1.1 200 OK
   ```

NOTE

To import an unchanged vendor certification extension template, use a file that is similar to the following template:

```
<?xml version="1.0" encoding="UTF-8"?>
<DataModel namespace="http://im.ca.com/certifications/snmp" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance" xsi:noNamespaceSchemaLocation="SNMPCertificationFacet.xsd">
  <Author>CA</Author>
  <Version>2.02</Version>
  <FacetType name="IfXTableMib"
 descriptorClass="com.ca.im.core.datamodel.certs.CertificationFacetDescriptorImpl">
    <FacetOf namespace="http://im.ca.com/core" name="Item" />
  </FacetType>
</DataModel>
```

Import Custom Vendor Certifications in the UI

After you create a custom vendor certification, you can share it with other Data Aggregator users who want to gather metrics for the same vendor.

WARNING

The **genericWS** format will be deprecated in a future release. Therefore, we recommend that you use the **typecatalog** format.

Custom vendor certifications can be shared between Data Aggregator users. Custom vendor certifications let users gather metrics for vendor devices when a factory certification is not yet available. To use a shared vendor certification, import it in XML format into your installation of Data Aggregator. You are not required to import the MIB.

Follow these steps:

NOTE

The metric family that is associated with the vendor certification must be available in CA NetOps Portal before importing. The import fails when the associated metric family is unavailable.

1. Click **Vendor Certifications** from the **Monitoring Configuration** menu for a Data Aggregator data source.
2. Click **Import**.

   NOTE

   The Import button supports ZIP files. For example, you can import the downloaded ZIP file for a certification from the On-Demand Certification support page.
3. Click **Browse**, and select the custom vendor certification file.
4. Click **Open**, and then click **Import**.
   Your custom vendor certification is imported. Data Aggregator immediately begins collecting metrics for the metric families that are associated with the newly imported custom vendor certification.

Verify the Vendor Certification Results

To ensure successful operation, verify the results of the import.

Follow these steps:

1. Log in to the UI, and navigate to the **Data Aggregator** data source.
2. Click **Monitoring Configuration**, **Vendor Certifications**.
3. Verify that the certification appears in the list and that the Last Modified time has been updated.
4. Click your metric family in the **Metric Families** view, and verify that the metric family appears.
5. Click the **Vendor Certification Priorities** tab.
   New custom vendor certifications are automatically added to the bottom of the priority list for the specified metric family. If necessary, modify the priority list and move the vendor certification to a higher priority.

Manage Vendor Certification Priorities

If the metric family for a device supports more than one vendor certification, the vendor certification with the highest priority is selected as the backing vendor certification. New custom vendor certifications are automatically added to the bottom of the priority list for the specified metric family. If necessary, modify the priority list and move the vendor certification to a higher priority. To use more than one vendor certification for the same device, group vendor certification priorities.

Prioritize the Vendor Certification Within the Metric Family

When you change the priority of the vendor certifications for a metric family, the metric family is updated on all affected devices. An event is generated on the metric family, indicating that the vendor certification priority on the metric family has changed. If the backing vendor certification changes on a device, an event is generated on the device. The event indicates that a vendor certification has changed. A second event is also generated, indicating the specific changes.

> **NOTE**
>
> To take advantage of any new vendor certifications that are part of an installation upgrade, manually change the vendor certification priorities. For example, F5 CPU vendor certifications are modeled as normal CPUs but are not discovered because F5 also supports Host Resources. After an upgrade, the Host Resources CPU priority entry will be higher than the F5 entries appended to the end of the priority list. To discover F5 CPU devices and components, update the vendor certification priority for the CPU metric family. A fresh installation does not have this issue.

Follow these steps:

1. Log in to the UI, and navigate to the **Data Aggregator** data source.
2. Click **Monitoring Configuration**, **Metric Families**.
3. Select a metric family, and click the **Vendor Certification Priorities** tab.
   The list of prioritized vendor certifications appears.
4. Click **Manage**.
5. Arrange the priority order as necessary, and click **Save**.
6. The metric family uses the new priority to determine which vendor certification to use for monitored devices.

Group Vendor Certification Priorities

Grouping vendor certification priorities lets you use more than one vendor certification for the same device. A single vendor certification priority can belong to as many priority groups as necessary for your application. When a device is discovered, all vendor certifications in the priority group are supported on the device.

WARNING

Do not update PriorityGroup and change the order of the vendor certifications in the same REST call. Changing both simultaneously may disrupt automatic rediscovery. For more information, see Automatic Rediscovery Does Not Run After Updating Vendor Group Priority.

To group vendor certification priorities, use the REST Web Services. You cannot group vendor certification priorities using the UI.

Follow these steps:

1. Run a GET operation on the following REST URL:
   `http://da_host:8581/rest/vendorpriorities/`
   The REST call returns a list of all the vendor certification priorities.
2. Determine the ID of your metric family in the list of vendor certification priorities.
3. Run a GET operation on the following REST URL:
   `http://da_host:8581/rest/vendorpriorities/ID`
   The XML of the vendor certification priority that you want to modify is retrieved.
4. To add a vendor certification priority to a group, add the `<PriorityGroup>` tag to the vendor certification.
   Example:
   In this example, the vendor certification priority is added to the group entitled "F5".

```
<CertificationOrder>
   <CollectionID>167</CollectionID>
   <VendorCertID>{http://im.ca.com/certifications/snmp}F5BigipMultihostCpu</
VendorCertID>
   <PriorityGroup>F5</PriorityGroup>
</CertificationOrder>
```

   WARNING

   Proper XML for the `<VendorCertID>` tag is written on a single line, and excludes spaces and carriage returns. Failure to follow this guideline causes the vendor certification priority to fail.

   NOTE

   To enabled recommended priority groups, replace the `<RecommendedPrioritiyGroup>` tag with the `<PriorityGroup>` tag.
5. (Optional) To add a single vendor certification priority to multiple groups, separate the group names using commas.
   Example:
   `<PriorityGroup>F5, Huawei</PriorityGroup>`
6. Remove the `<ID>` and `<MetricFamilyID>` tags from the vendor certification priority XML.
7. Run a PUT REST call on the following URL to import the new XML file:
   `http://da_host:8581/rest/vendorpriorities/ID`

   ID
   Specifies the ID of the vendor certification priority.
   DX NetOps Performance Management rediscovers all devices that support the metric family.

Verify Vendor Certification Priority Grouping

Verify that the vendor certification priorities are grouped as necessary.

Follow these steps:

1. Log in to the UI and navigate to the Data Aggregator data source.
2. Click **Monitored Inventory**, **Monitored Devices**.

3. Select a device from the Tree View.
4. Click the **Polled Metric Families** tab in the pane on the right.
   The vendor certifications that you grouped under the corresponding metric families appear on multiple rows in the **Vendor Cert** column.

Basic Vendor Self-Certification

For basic vendor certification, use the UI to apply simple changes to a vendor certification. This option is useful for limited changes that you need immediately.

Basic certification has the following limitations:

- Applies only to devices with a single MIB and a single MIB table.
- Cannot modify all aspects of the vendor certification.
- Does not apply to extended vendor certifications or custom certifications created with the advanced method.

Log in as the administrator to perform these tasks:

Create a Custom Vendor Certification in the UI

A vendor certification maps attributes from a vendor MIB to the metrics specified in a metric family. Vendor certifications also determine how metrics collected from an item are formatted for use in the UI and in reports. Metrics that are provided for an item can vary depending on the vendor of the item. Mapping these values ensures that the metric values are reported consistently, regardless of the vendor.

Different vendor certifications can apply to the same metric family. Data Aggregator maps metric values using a ranked list of vendor certifications. Data Aggregator calculates a metric value using the highest priority vendor certification that matches the polled item. If a vendor certification does not exist for a device, you can create one.

To determine if a device supports a vendor certification, the device is queried for each key attribute that is defined in the specified vendor certification. If a device supports an attribute, the device responds to an SNMP GET NEXT request on a given Object ID. A device supports a vendor certification only if all the key attributes are supported. For custom vendor certifications created with the Vendor Certification wizard, any attribute used in an expression is considered “key” and therefore the device supports it.

WARNING

To avoid possible data loss, back up the certifications directory each time that you create or update a vendor certification, metric family, or component.

Follow these steps:

1. Click **Vendor Certifications** from the **Monitoring Configuration** menu for a Data Aggregator data source.
2. Click **New**.
   The New Vendor Certifications wizard opens, guiding you through the following steps to define your vendor certification:
   - Selecting the MIB for your device.
   - (Optional) Importing a new MIB when one does not exist.
   - Selecting the metric family to which you want to map.

     NOTE
     Only scalar metric families or table metric families with a single index appear in the list.
   - Defining the expressions between the metrics in a selected metric family and the vendor certification variables from the selected MIB.

NOTE
The Names and Indexes metric names require an expression. For a custom vendor certification created with the Vendor Certification wizard, Data Aggregator automatically provides the Indexes metric expression for your selected MIB table.

3. Follow the prompts.
   The new vendor certification is created and automatically added to the end of the priority list for the selected metric family. The vendor certification is automatically associated with the All Manageable Devices device collection.

   NOTE
   (Optional) You can change the priority of the vendor certification within the metric family.
4. Run a discovery and verify that Data Aggregator is correctly polling the metric data for your devices.

   NOTE
   Data Aggregator only polls for metrics that have an expression defined in the vendor certification. If you try to report on metrics without a defined expression, your reports display a "no data available" message.

Import Custom Vendor Certifications in the UI

After you create a custom vendor certification, you can share it with other Data Aggregator users who want to gather metrics for the same vendor.

WARNING

The **genericWS** format will be deprecated in a future release. Therefore, we recommend that you use the **typecatalog** format.

Custom vendor certifications can be shared between Data Aggregator users. Custom vendor certifications let users gather metrics for vendor devices when a factory certification is not yet available. To use a shared vendor certification, import it in XML format into your installation of Data Aggregator. You are not required to import the MIB.

Follow these steps:

NOTE
The metric family that is associated with the vendor certification must be available in NetOps Portal before importing. The import fails when the associated metric family is unavailable.

1. Click **Vendor Certifications** from the **Monitoring Configuration** menu for a Data Aggregator data source.

   NOTE

   Use the Search feature in any pane to locate specific information that is related to that pane. Alternatively, navigate between pages in a pane using the arrows.
2. Click **Import**.

   NOTE

   The Import button supports ZIP files. For example, you can import the downloaded ZIP file for a certification from the On-Demand Certification support page.
3. Click **Browse**, and select the custom vendor certification file.
4. Click **Open**, and then click **Import**.
   Your custom vendor certification is imported. Data Aggregator immediately begins collecting metrics for the metric families that are associated with the newly imported custom vendor certification.

Edit a Custom Vendor Certification in the UI

You can edit existing custom vendor certifications to collect additional data for reporting. For example, you can edit the expression that maps to the normalized metric family variables.

This method applies only to custom vendor certifications that you created or edited in the UI.

WARNING

To avoid possible data loss, back up the certifications directory each time that you create or update a vendor certification, metric family, or component.

Follow these steps:

1. Click **Vendor Certifications** from the **Monitoring Configuration** menu for a Data Aggregator data source.
2. Select a custom vendor certification from the list.
3. Select a metric family, and click **Edit**.
4. Manually edit the expression.
   Common edits include the following changes:
   – Change a value that is assigned to the expression, such as an average.
   – Add multiple vendor certification variables to the expression.
   – Remove an expression from a metric family variable by clearing the Expression text box.

   NOTE
   The Names and Indexes metric names require an expression. For a custom vendor certification created with the Vendor Certification wizard, Data Aggregator automatically provides the Indexes metric expression for your selected MIB table.
5. Click **Accept Expression**.
   The expression is mapped, and the top table is populated with the updated values.
6. Click **Save**.
   The vendor certification details grid is updated with the changes to the metric family variables.

Create or Edit Vendor Certification Expressions

To modify the mapping of normalized metric family variables, edit the vendor certification expressions.

Consider the following information when you edit expressions:

- Use delimiters to separate vendor certification variables.
- MVEL functions and custom functions are valid.
- The Names and Indexes metric names require an expression. The remaining metric names are optional.

NOTE

For a vendor certification created with the Vendor Certification wizard, DX NetOps Performance Management automatically provides the Indexes metric expression for your selected MIB table.

Example: Change the Value for Averages

Cisco router CPU statistics are mapped to the normalized variable 'CPU Utilization', as shown in the following expression:

```
cpmCPULoadAvg5min+cpmCPUUseravg5min
```

You can change the 5-minute average to a 1-minute average by editing the expression as follows:

```
cpmCPULoadAvg1min+cpmCPUUseravg1min
```

Example: Use an Advanced Expression

The following expression verifies whether hrStorageSize < 0, and returns the value of hrStorageSize converted to an Unsigned value, and multiplied by 100. Otherwise, the expression returns the following value: hrStorageUsed/ hrStorageSize * 100.

```
(hrStorageSize &lt; 0) ? (hrStorageUsed/
convertSignedIntToUnsignedDecimal(hrStorageSize)) * 100 : hrStorageUsed/
hrStorageSize * 100
```

Functions and Global Variables

MVEL is the language that vendor certifications use to manipulate data from monitored devices. In vendor certification expressions, use MVEL to normalize data and perform calculations.

MVEL is a publicly available Expression Language for Java environments that can be embedded. MVEL supports expressions similar to Java expressions. You can build expressions using operators, use braces to control precedence, and terminate statements using semi-colons. For a detailed reference of the MVEL language, see the MVEL 2.0 Wiki.

Performing Calculations with MVEL

If your expression contains a calculation without at least one attribute variable, you might see unexpected results. Include at least one attribute in your expression that does not affect the calculation result. For example, the following expression might return unexpected results:

```
<Expression destAttr=”metric1”>15 * 15</Expression>
```

However, the following expression does not affect the result and does return the expected result:

```
<Expression destAttr=”metric1”>Index; 15 * 15</Expression>
```

- **Index**
  Specify an attribute variable that is defined in the **Attribute** or **AttributeGroup**.

The following certification shows the proper usage of MVEL functions for vendor certifications: MVEL Test Certification. The **assert** tag is used for internal testing purposes only, and is not used in creating vendor certifications.

> **WARNING**
>
> To ensure that you do not lose all the data from an expression that uses an assert tag, add a space before the semicolon at the end of the expression.

The following custom functions and global variables are available for use in vendor certification expressions:

availabilityWithSysUptime Function

This function calculates availability as a percentage using sysUptime and the poll duration, granting a grace period.

Syntax

This function has the following format:

```
Object availabilityWithSysUptime (Long sysUpTime, Long duration)
```

Parameters

- **sysUpTime**

The time (in centiseconds) since the network management portion of the system was last reinitialized.

- **duration**
  The poll duration time in seconds. Use the global variable _rspDuration. See the Advanced example for more information.

Return Values

Returns the availability as a percentage (0 - 100), or returns "null" when invalid data is passed.

Examples

The following expression produces the following result for a sysUpTime of 30000 and a poll duration of 300:

Expression:

```
availabilityWithSysUptime (sysUpTime, duration)
```

Result:

```
100
```

The same expression produces the following result for a sysUpTime of 6000 and a poll duration of 300:

Result:

```
20
```

The same expression produces the following result for a sysUpTime of 30005 and a poll duration of 300:

Result:

```
100
```

Advanced Example

The following expression is taken from “System Statistics” Vendor Certification:

```
Availability=availabilityWithSysUptime(sysUpTime,_rspDuration)
```

mapModel Function

This function uses the value of an objectID (sysObjectID) and maps the system OID to a model name string. Use this function to certify devices.

Syntax

This function has the following format:

```
String  mapModel ( ObjectID sysObjectID )
```

Parameters

- **sysObjectID**
  The object ID value to parse.

Return Values

Returns the string containing the mapped model name.

Examples

The following expression produces the following result for an OID value of 1.3.6.1.4.1.9.1.223:

Expression:

```
mapModel (oid )
```

Result:

```
Cisco7204VXR
```

The same expression produces the following result for an OID value of 1.3.6.1.4.1:

Result:

```
Unknown 1.3.6.1.4.1
```

Advanced Example

The following expression is taken from “System Statistics” Vendor Certification:

```
Model=mapModel (sysObjectID)
```

mapVendor Function

This function uses the value of an objectID (sysObjectID) and maps the system OID to a vendor name string. Use this function to find the vendor of a device.

Syntax

This function has the following format:

```
String mapVendor( ObjectID sysObjectID )
```

Parameters

- **sysObjectID**
  The object ID value to parse.

Return Values

Returns the string containing the mapped vendor name. If a vendor is not found, returns "".

Examples

The following expression produces the following result for an OID value of 1.3.6.2.1.2.2636.0:

Expression:

```
mapVendor (oid )
```

Result:

```
Juniper
```

The same expression produces the following result for an OID value of 1.3.6.2.1.2.1234567.0:

Result:

```
Unknown
```

Advanced Example

The following expression is taken from “System Statistics” Vendor Certification:

```
Model=mapVendor (sysObjectID)
```

mvelInfo Function

This function populates the INFO level of the karaf.log file with the input parameters. Use this function to log the polled values from a report that returns a result that you believe is incorrect.

The poll log of the Data Collector only shows the values of the polled attributes, while the report only shows the result of the calculation. The mvelInfo function allows you to view the input poll values to determine where the calculation went wrong.

Syntax

This function has the following format:

```
String mvelInfo (Array objects)
```

Parameters

- **objects**
  The object array is logged under the INFO level in the karaf.log file of the Data Collector.

Return Values

Null

Examples

The following expression logs cpmCPUTotal5minRev in the karaf.log file.

Expression:

```
mvelInfo({"cpmCPUTotal5minRev=", cpmCPUTotal5minRev})
```

Result:

Null

Result (karaf.log):

MVEL info: cpmCPUTotal5minRev=15

Advanced Example

```
mvelInfo({"cpmCPUTotal5minRev=", cpmCPUTotal5minRev, " cpmCPUTotal10minRev=",
 cpmCPUTotal10minRev}); cpmCPUTotal10minRev;
```

Result:

12

Result (karaf.log):

MVEL info: cpmCPUTotal5minRev=15 cpmCPUTotal10minRev=12

mvelWarn Function

This function populates the WARN level of the karaf.log file with the input parameters. Use this function to log the polled values from a report that returns a result that you believe is incorrect.

The poll log of the Data Collector only shows the values of the polled attributes, while the report only shows the result of the calculation. The mvelWarn function allows you to view the input poll values to determine where the calculation went wrong.

Syntax

This function has the following format:

```
String mvelWarn (Array objects)
```

Parameters

- **objects**
  The object array is logged under the WARN level in the karaf.log file of the Data Collector.

Return Values

Null

Examples

The following expression logs cpmCPUTotal5minRev in the karaf.log file.

Expression:

```
mvelWarn({"cpmCPUTotal5minRev=", cpmCPUTotal5minRev})
```

Result:

Null

Result (karaf.log):

MVEL warn: cpmCPUTotal5minRev=15

Advanced Example

```
mvelWarn({"cpmCPUTotal5minRev=", cpmCPUTotal5minRev, " cpmCPUTotal10minRev=",
 cpmCPUTotal10minRev}); cpmCPUTotal10minRev;
```

Result:

12

Result (karaf.log):

MVEL warn: cpmCPUTotal5minRev=15 cpmCPUTotal10minRev=12

mvelError Function

This function populates the ERROR level of the karaf.log file with the input parameters. Use this function to log the polled values from a report that returns a result that you believe is incorrect.

The poll log of the Data Collector only shows the values of the polled attributes, while the report only shows the result of the calculation. The mvelError function allows you to view the input poll values to determine where the calculation went wrong.

Syntax

This function has the following format:

```
String mvelError (Array objects)
```

Parameters

- **objects**
  The object array is logged under the ERROR level in the karaf.log file of the Data Collector.

Return Values

Null

Examples

The following expression logs cpmCPUTotal5minRev in the karaf.log file.

Expression:

```
mvelError({"cpmCPUTotal5minRev=", cpmCPUTotal5minRev})
```

Result:

Null

Result (karaf.log):

MVEL error: cpmCPUTotal5minRev=15

Advanced Example

```
mvelError({"cpmCPUTotal5minRev=", cpmCPUTotal5minRev, " cpmCPUTotal10minRev=",
 cpmCPUTotal10minRev}); cpmCPUTotal10minRev;
```

Result:

12

Result (karaf.log):

MVEL error: cpmCPUTotal5minRev=15 cpmCPUTotal10minRev=12

mvelDebug Function

This function populates the DEBUG level of the karaf.log file with the input parameters. Use this function to log the polled values from a report that returns a result that you believe is incorrect.

The poll log of the Data Collector only shows the values of the polled attributes, while the report only shows the result of the calculation. The mvelDebug function allows you to view the input poll values to determine where the calculation went wrong.

Syntax

This function has the following format:

```
String mvelDebug (Array objects)
```

Parameters

- **objects**
  The object array is logged under the DEBUG level in the karaf.log file of the Data Collector.

Return Values

Null

Examples

The following expression logs cpmCPUTotal5minRev in the karaf.log file.

Expression:

```
mvelDebug({"cpmCPUTotal5minRev=", cpmCPUTotal5minRev})
```

Result:

Null

Result (karaf.log):

MVEL debug: cpmCPUTotal5minRev=15

Advanced Example

```
mvelDebug({"cpmCPUTotal5minRev=", cpmCPUTotal5minRev, " cpmCPUTotal10minRev=",
 cpmCPUTotal10minRev}); cpmCPUTotal10minRev;
```

Result:

12

Result (karaf.log):

MVEL debug: cpmCPUTotal5minRev=15 cpmCPUTotal10minRev=12

mvelTrace Function

This function populates the TRACE level of the karaf.log file with the input parameters. Use this function to log the polled values from a report that returns a result that you believe is incorrect.

The poll log of the Data Collector only shows the values of the polled attributes, while the report only shows the result of the calculation. The mvelTrace function allows you to view the input poll values to determine where the calculation went wrong.

Syntax

This function has the following format:

```
String mvelTrace (Array objects)
```

Parameters

- **objects**
  The object array is logged under the TRACE level in the karaf.log file of the Data Collector.

Return Values

Null

Examples

The following expression logs cpmCPUTotal5minRev in the karaf.log file.

Expression:

```
mvelTrace({"cpmCPUTotal5minRev=", cpmCPUTotal5minRev})
```

Result:

Null

Result (karaf.log):

MVEL trace: cpmCPUTotal5minRev=15

Advanced Example

```
mvelTrace({"cpmCPUTotal5minRev=", cpmCPUTotal5minRev, " cpmCPUTotal10minRev=",
 cpmCPUTotal10minRev}); cpmCPUTotal10minRev;
```

Result:

12

Result (karaf.log):

MVEL trace: cpmCPUTotal5minRev=15 cpmCPUTotal10minRev=12

snmpConstArrayMap Function

This function maps a value (index) to a set of constant values (array). If necessary, this function rounds the input value to nearest integer value. Then, it uses the integer value as an index to the set of constant values (array) that are shown as c0, c1, up to cn-1. The c values must be integers. This function checks these values when the expression is parsed and returns c*x*. If the value is not in the domain from 0 to n-1 (inclusive), the result is 0 (without an error message). Use this function to certify devices.

Syntax

This function has the following format:

```
Integer  snmpConstArrayMap(Double index, Integer[] array)
```

Parameters

- **index**
  A Double value, which is used as an index into the array.
- **array**
  Any range of integer values.

Return Values

Returns an integer value from the array. An index value of null returns null.

Examples

The following expression produces the following result for an index of 2 and an array of {5, 6, 7, 8, 9, 4}:

Expression:

```
snmpConstArrayMap (index, array)
```

Result:

```
7
```

The following expression produces the following result for an index of 4.88 and an array of {5, 6, 7, 8, 9, 4}:

Expression:

```
snmpConstArrayMap (value, array)
```

Result:

```
4
```

Advanced Example

The following expression is taken from “Generic Modem” Vendor Certification:

```
SpeedOut=snmpConstArrayMap(mdmCsFinalTxLinkRate,
{0,110,300,600,1200,2400,4800,7200,9600,12000,14000,16000,19000,38000,75,450,0,57000,21000,2400
```

snmpCounter64 Function

This function evaluates two 32-bit numeric values and returns a value containing the 64-bit representation. Use this function to certify devices. The hiVal is shifted 32 bits left and the lowVal is added and the result is placed in a 64-bit return variable.

Syntax

This function has the following format:

```
Object snmpCounter64 (Long hiVal, Long lowVal)
```

Parameters

- **hiVal**
  The 32-bit numeric value representing the high-order bits.
- **lowVal**
  The 32-bit numeric value representing the low-order bits.

Return Values

Returns the 64-bit representation of the two 32-bit numeric values, or returns "null" when either 32-bit value input is null.

Examples

The following expression produces the following result for a hiVal of 88 and a lowVal of 558.

Expression:

```
snmpCounter64 (hiVal, lowVal)
```

Result:

```
377957122606
```

Advanced Example

The following expression is taken from “Cisco CBQos ClassMap” Vendor Certification. This certification contains many snmpMax examples:

```
PrePolicyPackets=snmpMax(0,snmpCounter64(cbQosCMPrePolicyPktOverflow,cbQosCMPrePolicyPkt))
```

snmpGet

This function gets the value of an ObjectID on the polled device.

> **NOTE**
>
> This function issues extra SNMP requests. To avoid a negative performance impact, certifications that include this expression are moved to a potentially slower processing queue.

Syntax

This function has the following format:

```
<GetResponse> snmpGet (OID<string>)
```

Parameters

- **OID** The ObjectID of the polled device.

Functions

- **getError()** Returns the response error code**. Example:** SUCCESS
- **getIp()** Returns the IP address of the device.
- **getOid()** Returns the requested ObjectID.
- **getResult()** Returns the SNMP response object.
- **getValue()** Returns the value from the SNMP response object.
- **getType()** Returns the type of the SNMP response object, such as COUNTER32 or GAUGE.

Return Values

Returns the response object of an SNMP GET request.

Example

The following expression includes sample results for each function:

```
response = snmpGet('1.3.6.1.2.1.3.0');
response.getError() --- SUCCESS
response.getIp() --- 10.42.96.5
response.getOid() --- 1.3.6.1.2.1.3.0
value = response.getResult();
value.getType() --- TIMETICKS
value.getValue() --- 31528546
```

The following expression does a null check (.?) to avoid breaking the logic when the result is null:

```
response = snmpGet('1.3.6.1.2.1.3.0');
response.error --- SUCCESS
response.ip --- 10.42.96.5
response.oid --- 1.3.6.1.2.1.3.0
response.?result.type ---- TIMETICKS
 response.?result.value ---- 31528546
```

snmpGetTable

This function gets the values of all ObjectIDs that share the prefix ObjectID.

> **NOTE**
>
> This function issues extra SNMP requests. To avoid a negative performance impact, certifications that include this expression are moved to a potentially slower processing queue.

Syntax

This function has the following format:

```
<GetBulkResponse> snmpGetTable (OID<string>)
```

Parameters

- **OID**
  The ObjectID of the polled device.

Functions

- **getError()** Returns the response error code**. Example:** SUCCESS
- **getIp()** Returns the IP address of the device.
- **getOid()** Returns the requested ObjectID.
- **getResult()** Returns a list of SNMP response objects.
- **getValue()** Returns the value from the SNMP response object.
- **getType()** Returns the type of the SNMP response object, such as COUNTER32 or GAUGE.
- **getIndex()** Returns the SNMP instance index of the SNMP response object.

Return Values

Returns a table that contains a list of SNMP response objects.

Examples

The following expression includes sample results for each function:

```
response = snmpGetTable('1.3.6.1.2.1.2.2.1.10');
response.getError() --- SUCCESS
response.getIp() --- 10.42.96.5
response.getOid() --- 1.3.6.1.2.1.2.2.1.10
list = response.getResult();
list.size() --- 200
list.get(0).getType() --- COUNTER
list.get(0).getValue() --- 1849
list.get(0).getIndex() --- 1
item = list.get(199);
item.getType() --- COUNTER
item.getResult() --- 1855
item.getIndex() --- 200
for(int i =0; i < list.size(); i ++){
    instance = list.get(i);
    instanceType = instance.getType();
    instanceValue = instance.getValue();
    instanceIndex = instance.getIndex();
}
```

NOTE

The `instanceType = listget(i).getType()` method chain is unsupported. The result of `list.get(i)` must be stored in a variable first. Use the following expression instead:

```
instance = list.get(i);
instanceType = instance.getType();
```

snmpGetUpSinceTime Function

This function returns the time that the system was turned on based on the number of seconds since the current epoch.

Syntax

This function has the following format:

```
snmpGetUpSinceTime(Long upTime)
```

Parameters

- **upTime**
  The number of seconds since the beginning of the current epoch. You can get the system uptime from the following OID: 1.3.6.1.2.1.1.3.0. Convert it into centiseconds before you pass it in.

Return Values

Returns the time that the device was powered on in the form of total seconds since the current epoch.

snmpMax Function

This function returns the larger of two 64-bit values. Use this function to certify devices.

Syntax

This function has the following format:

```
Object snmpMax(BigInteger val1, BigInteger val2)
```

Parameters

- **val1**
  The first 64-bit BigInteger value.
- **val2**
  The second 64-bit BigInteger value.

Return Values

Returns the maximum of the two BigInteger values that are passed in, or returns "null" when either BigInteger value input is null.

Examples

The following expression produces the following result for a val1 of 2^32 and a val2 of 10:

Expression:

```
snmpMax (val1, val2)
```

Result:

```
2^32
```

The same expression produces the following result for a val1 of 5864 and a val2 of 134556890:

Result:

```
134556890
```

Advanced Example

The following expression is taken from “Cisco CBQos ClassMap” Vendor Certification. This certification contains many snmpMax examples:

```
PrePolicyPackets=snmpMax(0,snmpCounter64(cbQosCMPrePolicyPktOverflow,cbQosCMPrePolicyPkt))
```

snmpObjectIDToASCIIString Function

This function converts an SNMP OID value to its string representation. Any leading or trailing spaces are removed.

Syntax

This function has the following format:

```
snmpObjectIDToASCIIString( Object Id oid )
```

Parameters

- **oid**
  The object ID to convert to a string.

snmpOIDParser Function

This function uses the value of an objectID (OID) and parses out a subset of the OID based on the startIndex and endIndex values. The indexes are 1 based. If the endIndex is -1, then we go to the end of the OID. Use this function to certify devices.

Syntax

This function has the following format:

```
ObjectID snmpOIDParser( ObjectID OID, Integer startIndex, Integer endIndex )
```

Parameters

- **OID**
  The object ID (OID) value to parse.
- **startIndex**
  An integer value of the index at which to begin parsing.
- **endIndex**
  An integer value of the index at which to stop parsing.

Return Values

Returns the parsed subset ObjectID (OID).

Examples

The following expression produces the following result for an OID value of 1.2.3.4.5.6.7.8.9.10, a startIndex value of 1, and an endIndex value of 5:

Expression:

```
snmpOIDParser(oid, startIndex, endIndex )
```

Result:

```
1.2.3.4.5
```

The same expression produces the following result for an OID value of 1.2.3.4.5.6.7.8.9.10, a startIndex value of 6, and an endIndex value of -1:

Result:

```
6.7.8.9.10
```

Advanced Example

The following expression is taken from “Cisco CBQos ClassMap” Vendor Certification:

```
ItemUniqueIDs=snmpOIDParser(cbQosConfigIndex, 2, 2)
```

snmpOctetStringFloat Function

This function converts an SNMP octet string to a floating-point value. Use this function to certify devices. An SNMP octet string is a seven-bit ASCII string.

Syntax

This function has the following format:

```
Object snmpOctetStringFloat(byte[] octetString)
```

Parameters

- **octetString**
  The SNMP octet string.

Return Values

Returns the converted string value, or returns "null" when the function cannot convert the string.

Examples

The following expression produces the following result for an octetString of {0x33, 0x33, 0x2E, 0x33, 0x33}:

Expression:

```
snmpOctetStringFloat (octetString)
```

Result:

```
33.33
```

The same expression produces the following result for an octetString of {0x36, 0x36, 0x36}:

Result:

```
666.0
```

snmpProtectedDiv Function

This function divides two Double values and returns the result of the division as a Double. If the dividend or divisor is null or 0.0 the return value is 0.0. Use this function to protect the expression from dividing with null or 0. Data Repository can contain null or zero values, such as when a poll fails. In this case, you avoid a divide-by-zero exception by using this function.

Syntax

This function has the following format:

```
Double snmpProtectedDiv(Double val1, Double val2)
```

Parameters

- **val1**
  The dividend, which is a Double value (floating number) to be divided by val2. (Double is a Java data type.)
- **val2**
  The divisor, which is a Double value (floating number). (Double is a Java data type.)

Return Values

Returns the result of the division as a Double or 0.0 if the dividend or divisor is null or 0.0 (*Double* is a Java data type).

Examples

The following expression produces the following result for a val1 of 7.2 and val2 of 2:

Expression:

```
snmpProtectedDiv(val1, val2)
```

Result:

```
3.6
```

The following expression produces the following result for a val1 of 7.2 and val2 of null or 0.0:

Result:

```
0.0
```

Advanced Example

The following expression is taken from Vendor Certification:

```
Utilization=snmpProtectedDiv((cpuStatsUser + cpuStatsSys),(cpuStatsUser + cpuStatsSys +
 (isdef(cpuStatsWait)?cpuStatsWait:0) + cpuStatsIdle))*100
```

snmpRound Function

This function rounds a numeric value to the nearest integer value.

Syntax

This function has the following format:

```
Long snmpRound(Double dNumber)
```

Parameters

- **dNumber**
  A Double value (floating number) to be rounded (*Double* is a Java data type).

Return Values

Returns a Long value, which is the nearest integer value to the value provided in dNumber (*Long* is a Java data type).

Examples

The following expression produces the following result for a dNumber of 3.5:

Expression:

```
snmpRound(dNumber)
```

Result:

```
4
```

The same expression produces the following result for a dNumber of 3.4:

Result:

```
3
```

Advanced Example

The following expression is taken from “Cisco IPSLA Jitter Precision Statistics” Vendor Certification:

```
PathAvailability=snmpRound(rttMonJitterStatsNumOfRTT / (rttMonJitterStatsNumOfRTT
 + rttMonJitterStatsPacketLossSD + rttMonJitterStatsPacketLossDS +
 rttMonJitterStatsPacketOutOfSequence + rttMonJitterStatsPacketMIA +
 rttMonJitterStatsPacketLateArrival + rttMonJitterStatsError + rttMonJitterStatsBusies +
 1/100) * 100)
```

snmpStringParser Function

This function was written for internal use only. The function parses IP addresses that are received from a CA Application Insight Module (AIM). CA has only tested this function with an internal class. Another type of class may not be supported.

Syntax

This function has the following format:

```
snmpStringParser(Delimiter, Type to convert to, String to parse 1, String to parse 2)
```

Parameters

- **Type to convert to**
  The type of class to which to parse the supplied strings.
- **Strings to parse 1 and 2**
  The IP address to parse. Two strings enable you to provide addresses in IPv4 and IPv6 format.

Return Values

Returns the converted value of the string, or returns "null" when the function cannot parse the string.

snmpSvcs Function

This function takes the values from sysObjectOID, sysService, and ipForwarding MIB variables of an agent and determines what services the SNMP agent supports. For example, Router/Switch/Repeater/Host could be a supported service, as defined in the SNMP MIB RFC 1213.

The return from the function is evaluated as follows because custom device types have high precedence over system ones:

- If sysObjectID OID is mapped in the DeviceTypes file, then the return services is from the file.
- If sysObjectID OID is not mapped in the DeviceTypes file, then the sysServices and ipForwarding is used to return the supported services.

Syntax

This function has the following format:

```
DeviceService[] snmpSvcs(ObjectID sysObjectID, Integer sysServices,
 Integer ipForwarding)
```

Parameters

- **sysObjectID**
  The object ID value to parse.
- **sysServices**
  An integer where each bit represents a different service, such as switch/repeater/host.
- **ipForwarding**
  An integer indicating whether this entity is acting as an IP gateway or IP host regarding the forwarding of datagrams. This entity receives the forwarded datagrams, but the forwarded datagrams are not addressed to this entity. Possible values are 1 (Forwarding) and 2 (notForwarding).

Return Values

Returns a list of one or more of the following device services:

- ROUTER
- REPEATER
- SWITCH
- HOST
- UNKNOWN_TYPE

Example

The following expression produces the following result for a sysServices value of 8, an ipForwarding value of 0, and sysObjectID not found in the DeviceTypes file:

Expression:

```
snmpSvcs(sysObjectOID,sysServices,ipForwarding)
```

Result:

```
DeviceService[HOST]
```

Advanced Example

The following expression is taken from the "System Statistics" Vendor Certification:

```
Services=snmpSvcs(sysObjectID,isdef(sysServices)?sysServices:0,isdef(ipForwarding)?
ipForwarding:0)
```

storePortReconfig Function

This function returns a string containing XML representing the values of ifNumber, ifTableLastChange, ifStackLastChange. You can use XML to track device changes and to rediscover interfaces on a device, when needed.

Syntax

This function has the following format:

```
String storePortReconfig ( Integer ifNumber, Long ifTableLastChange,
 Long ifStackLastChange )
```

Parameters

- **ifNumber**
  The number of ports on the device.
- **ifTableLastChange**
  The date and time of the latest port table change in milliseconds, calculated from a start date and time of January 1, 1970 GMT.
- **ifStackLastChange**
  The date and time of the latest port stack change in milliseconds, calculated from a start date and time of January 1, 1970 GMT.

Return Values

Returns a string containing XML in the form that is shown in the following example.

Example

The following expression produces the following result for an ifNumber of 5, ifTableLastChange of 123456, and ifStackLastChange of 234567:

Expression:

```
storePortReconfig ( ifNumber, ifTableLastChange, ifStackLastChange )
```

Result:

```
<ReconfigData>
     <ReconfigValue name="ifNumber" value="5"/>
     <ReconfigValue name="ifTableLastChange" value="123456"/>
     <ReconfigValue name="ifStackLastChange" value="234567"/>
</ReconfigData>
```

Global Variables

Data Aggregator supports the following global variables:

- **_ rspDuration**
  A Long (Java data type) containing the duration of the current poll cycle in seconds.

NOTE
The “System Statistics” Vendor Certification contains an example of the use of _rspDuration.

- **_rspTimestamp**
  A Long (Java data type) containing the timestamp at which the current poll cycle started in milliseconds since January 1, 1970 GMT.
- **_context**
  Java class that contains the details of the polled item, such as the Device Item ID.

WARNING
_context is a reserved keyword and global variable that is intended only for system use. Do not use **_context** under any circumstance.

Vendor Certification Expression Operators

Use MVEL syntax in vendor certification expressions. You can build expressions using operators, use braces to control precedence, and terminate statements by semi-colons. The MVEL language has vendor certification utility functions and vendor certification global variables that you can also use in vendor certification expressions.

MVEL is a publicly available embeddable Expression Language for Java environments that has a syntax close to Java. MVEL supports expressions similar to Java expressions. For a detailed reference of the MVEL language, see https://en.wikibooks.org/wiki/Transwiki:MVEL_Language_Guide.

To study the use of functions, operators, and global variables, you can use the Vendor Certification tab in NetOps Portal.

The following table summarizes the available operators:

NOTE
In XML documents, use the XML Named Entities (XNE) presentation.

Operator	XML Named Entities	Description	Example
=	N/A	Assign	a = 1
==	N/A	Equals	"fred" == "fred"
!=	N/A	Not Equals	"fred" != "tom"
>	&gt;	Greater Than	1 > 0 is true
<	&lt;	Less Than	0 < 1 is true
>=	N/A	Greater Than or Equal	1 >= 0 is true
<=	N/A	Less Than or Equal	1 <= 1 is true
contains	N/A	Verify if the value on the left side contains the value on the right	"tomcat" contains "cat"
isdef	N/A	Tests whether a variable is defined	isdef a
+	N/A	Add	1 + 1
+	N/A	Concatenate	"one " + "two"
-	N/A	Minus	2 - 1
*	N/A	Multiply	2 * 2
/	N/A	Divide	4 / 2
%	N/A	Modulus	5 % 2
&&	&&	Logical AND	(x>-1) && (x<1)
||	N/A	Logical OR	(x<-1) || (x>1)

&	&amp;	AND bit operation	17 & 0xF
|	N/A	OR bit operation	4 | 1
^	N/A	Exclusive OR bit operation	5 ^ 1
!	N/A	Negate	! True
?	N/A	Ternary operator	age > 17 ? "allow" : "deny"

Self Certification XML

Vendor certifications, metric families, and components are defined in XML. To create, extend, or update certifications, update the associated XML files.

The following pages provide details about the XML attributes and structure:

Certification Schema Files and Examples

The schema files that are provided with Data Aggregator have detailed information about element types, occurrence, and allowed lengths. The files also contain annotations that provide more information, such as allowed characters and naming conventions.

Before you create custom metric family and vendor certification XML files, download and review the related schema XSD files. You need the schema to validate your XML files.

Download the schema and example XML files by entering the following URLs in your web browser Address field. *Hostname* is the Data Aggregator host, and the default *port* is 8581.

- http://*hostname*:*port*/resource/xsd/IMDBCertificationFacet.xsd
- http://*hostname*:*port*/resource/xsd/ComponentFacet.xsd
- http://*hostname*:*port*/resource/xsd/ItemSyncDefinition.xsd
- http://*hostname*:*port*/resource/xsd/SNMPCertificationFacet.xsd
- http://*hostname*:*port*/resource/xsd/CertificationFacet.xsd
- http://*hostname*:*port*/resource/xsd/webservices.xsd
- http://*hostname*:*port*/resource/xsd/basewebservices.xsd
- http://*hostname*:*port*/resource/xsd/datamodel.xsd

Example XML files are located in /opt/IMDataAggregator/examples.

Restricted XML Tags

You cannot use the following XML tags in custom and extended vendor certifications:

- AggregateToDevice
- SupportsDeviceAggregation

When you extend a vendor certification or metric family, you cannot modify the following XML tags:

- AlarmRules
- Author
- ComponentFacets
- descriptorClass
- DeviceType
- IsDynamicDiscoveryAttribute
- ItemFacets
- ItemRelationshipMappings
- Normalized
- ParentNodeList
- Protocol
- RollupExpression
- TableName
- type
- UsesDynamicIndex
- Variance
- VCSupportExpression
- WriteOnPoll

Vendor Certification XML Structure

A vendor certification uses XML to map vendor- and device-specific data to performance metrics and configuration data that are defined in a metric family. Mapping this data from various sources to the "normalized" metric family values helps Data Aggregator uniformly report on this data, regardless of the device vendor.

WARNING

When you extend a vendor certification, do not edit restricted tags or attributes. For more information, see Restricted XML Tags.

NOTE

You must list some properties in the XML in a particular order. The properties included in the XML example and listed in the following descriptions are presented in the recommended order.

Example:

This example of a custom vendor certification XML supports Frame-Relay PVC. The example custom metric family, frPVCInfo, is included in the ExpressionGroup section:

NOTE

If you view the vendor certification XML in a browser, certain tags are hidden. For this reason, copy and paste the vendor certification XML only from a REST client

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- Auto-generated by the type catalog local manager.-->
<DataModel namespace="http://im.ca.com/certifications/snmp" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance"
 xsi:noNamespaceSchemaLocation="SNMPCertificationFacet.xsd">
  <Author>Custom</Author>
  <Version>1.0</Version>
```

```
 <FacetType name="frPVCInfoCustom"
descriptorClass="com.ca.im.core.datamodel.certs.CertificationFacetDescriptorImpl">
   <Documentation>Frame Relay PVC Vendor Certification</Documentation>
   <FacetOf namespace="http://im.ca.com/core" name="Item" />
   <DisplayName>Frame Relay PVC Certification</DisplayName>
   <MIB>RFC1315-MIB</MIB>
   <Protocol>SNMP</Protocol>
   <AttributeGroup name="AttributeGroup" external="true" list="true">
     <Documentation />
     <Attribute name="INDEX" type="ObjectID">
     <Documentation />
     <Source>1.3.6.1.2.1.10.32.2.1.4</Source>
     <IsIndex>true</IsIndex>
     <IsKey>false</IsKey>
     <NeedsDelta>false</NeedsDelta>
   </Attribute>
   <Attribute name="frCircuitReceivedBECNs" type="Long">
     <Documentation />
     <Source>1.3.6.1.2.1.10.32.2.1.5</Source>
     <IsIndex>false</IsIndex>
     <IsKey>true</IsKey>
     <NeedsDelta>true</NeedsDelta>
   </Attribute>
   <Attribute name="frCircuitSentFrames" type="Long">
     <Documentation />
     <Source>1.3.6.1.2.1.10.32.2.1.6</Source>
     <IsIndex>false</IsIndex>
     <IsKey>true</IsKey>
     <NeedsDelta>true</NeedsDelta>
   </Attribute>
   <Attribute name="frCircuitSentOctets" type="Long">
     <Documentation />
     <Source>1.3.6.1.2.1.10.32.2.1.6</Source>
     <IsIndex>false</IsIndex>
     <IsKey>true</IsKey>
     <NeedsDelta>true</NeedsDelta>
   </Attribute>
   <Attribute name="frCircuitReceivedFrames" type="Long">
     <Documentation />
     <Source>1.3.6.1.2.1.10.32.2.1.8</Source>
     <IsIndex>false</IsIndex>
     <IsKey>true</IsKey>
     <NeedsDelta>true</NeedsDelta>
   </Attribute>
   <Attribute name="frCircuitReceivedOctets" type="Long">
     <Documentation />
```

```
      <Source>1.3.6.1.2.1.10.32.2.1.9</Source>
      <IsIndex>false</IsIndex>
      <IsKey>true</IsKey>
      <NeedsDelta>true</NeedsDelta>
    </Attribute>
  </AttributeGroup>
  <Expressions>
    <ExpressionGroup destCert="{http://im.ca.com/normalizer}frPVCInfo"
 name="frPVCInfoDS">
      <Expression destAttr="Indexes">INDEX</Expression>
      <Expression destAttr="Names">"Frame Relay " + INDEX</Expression>
      <Expression destAttr="FECNIn">frCircuitReceivedFECNs</Expression>
      <Expression destAttr="BECNIn">frCircuitReceivedBECNs</Expression>
      <Expression destAttr="FramesIn">frCircuitReceivedFrames</Expression>
      <Expression destAttr="FramesOut">frCircuitSentFrames</Expression>
      <Expression destAttr="BytesIn">frCircuitReceivedOctets</Expression>
      <Expression destAttr="BytesOut">frCircuitSentOctets</Expression>
    </ExpressionGroup>
  </Expressions>
 </FacetType>
</DataModel>
```

Basic Properties

The basic properties of your custom vendor certification help to distinguish it from other custom vendor certifications you create. Also, these properties indicate from which vendor MIB you are collecting metric data.

Consider the following restrictions when you determine basic properties:

- The FacetType/name and FacetType/DisplayName properties must be unique for each vendor certification.
- The Protocol tag is either SNMP or EMS.
  - `/typecatalog/certifications/snmp` support only SNMP certifications. In this case, the only value that is supported is SNMP.
  - `/typecatalog/certifications/camm` support only CAMM certifications. In this case, the only value that is supported is EMS.
- Set the FacetType/descriptorClass property and all DataModel and FacetOf properties as shown in the example XML in the previous illustration.

The following list details basic vendor certification properties:

- **FacetType/name**
  Uniquely identifies a vendor certification.
  Recommendation: Conform to "<MibName><TableName>Mib."
  Can be updated: No
  Possible values: Alphanumeric and underscore. Dot and dash are not permitted.
  The FacetType section manifests a particular vendor certification. The same XML document can contain multiple FacetType sections when those vendor certifications expose various aspects of the vendor-specific device such as TCP and UDP statistics from a MIB-2 implementation.
  The FacetType section contains some basic properties. For example, this section contains the name of the vendor MIB, followed by one or more AttributeGroup sections. These AttributeGroup sections define which attributes this certification uses from the MIB. One or more ExpressionGroup sections map attributes from the AttributeGroup sections to the metrics specified in a metric family.

NOTE

All of the following items can be updated and support plain text.

- **FacetType/Documentation**
  Describes what is certified with the vendor certification.
  Recommendation: Include the details about the vendor, MIB name, and table name.
  Effect of updating: None

  NOTE

  This property should be listed first under the `FacetType/name` .
- **FacetType/DisplayName**
  Specifies the name of the vendor certification as it displays in NetOps Portal.
  Recommendation: Start with the vendor name and include the MIB and functionality information.
  Effect of updating: A change to the name in the Administrator user interface.
  When does the update take effect: Immediately
  Required actions for updates to take effect: Refresh the user interface.

  WARNING

  Ensure that the DisplayName property is unique to the vendor certification.
- **FacetType/MIB**
  Specifies the name of the MIB, which the DEFINITIONS clause defines in the ASN.1 file.
  Recommendation: Conform to "<MibName>”
  Effect of updating: Change to the “SNMP MIB Name” column in the Vendor Certification tab of the Administrator user interface.
  When does the update take effect: Immediately
  Required actions for updates to take effect: Refresh the user interface.
- **FacetType/Protocol**The Protocol tag is either SNMP or EMS.
  - `/typecatalog/certifications/snmp` support only SNMP certifications. In this case, the only value that is supported is SNMP.
  - `/typecatalog/certifications/camm` support only CAMM certifications. In this case, the only value that is supported is EMS.

AttributeGroup

The following example illustrates the AttributeGroup section of your custom vendor certification. This section identifies the attributes (variable OIDs) of a particular table in the vendor MIB that are used to map raw device data. This data is mapped to the performance metrics and the configuration data that is defined in a metric family.

You set the AttributeGroup/list and AttributeGroup/external properties to true, as shown in the example XML in the previous illustration. These properties specify that each attribute represents a list of values that is obtained from an external source (a MIB table). The following information summarizes the XML elements to customize.

NOTE

All of the following items can be updated and support plain text. The update does not affect performance.

- **AttributeGroup/name**
  Specifies the attribute group name.
  Recommendation: Conform to "<FacetType/name>Group."
- **Documentation**
  (Optional) Specifies the description for the attribute group.

NOTE

This property should be listed first under the `AttributeGroup/name` .

- **UseIndex**
  Specifies the name of the attribute to be used as the index for this attribute group for joining multiple MIB tables.

  NOTE

  When using MultiMIB Table Support, the AttributeGroup order must match the IndexTagList order.

  Recommendation: Set to the value of the AttributeGroup/name property.

  NOTE

  In the attributes list, the attributes used for calculating indexes and names should be listed first.

General Attributes

The general attributes for all vendor certifications are as follows:

NOTE

Unless specified otherwise, the update takes effect immediately and no actions are required to trigger the update. All entries in this list can be updated.

- **Attribute/name**
  Specifies the attribute name.
  Recommendation: Set to the MIB variable name, which the OBJECT-TYPE clause defines in the ASN.1 file.
  Possible values: Alphanumeric and underscore. Dot and dash are not permitted.
  Effect of updating: Update any expressions that reference this attribute.
- **Attribute/type**
  Specifies the data type of the attribute.
  Recommendation: Use the attribute type that best matches the variable type that the SYNTAX clause defines in the ASN.1 file.
  Possible values: Boolean, Int, Long, Double, BigInteger, String, DateTime, IPAddress, MACaddress, IPSubnet, OctetString, ObjectID
  Effect of updating: Polled SNMP data is converted to this type.
  When does the update take effect: Next poll
- **Documentation**
  (Optional) Specifies the description for the attribute, which documents the semantics (such as the unit) of the MIB variable.
  Recommendation: Use the descriptions that are taken from the MIB ASN.1 file.
  Possible values: Plain text
  Effect of updating: None

  NOTE

  This property should be listed first under the `Attribute/name` and `Attribute/type` .
- **IsKey**
  (Optional) Uses a flag to indicate whether the MIB variable is key for determining support for a table. When multiple fields are specified as keys, all of the fields are considered together as a compound key.

  NOTE

  You cannot use `IsKey` and `VCSupportExpression` in the same vendor certification. Vendor certifications that contain both `IsKey` and `VCSupportExpression` do not discover anything.

  Default: false

Recommendation: Set to true if it is a key MIB object for component discovery. If the contents of the MIB attributes are necessary to determine support, use `VCSupportExpression` instead.
Possible values: true, false
Effect of updating: Components could change to a new vendor certification.
When does the update take effect: After component rediscovery.
Required actions for updates to take effect: Update the metric family or change the vendor certification priority.

- **IsIndex**
  (Optional) Uses a flag to indicate whether this variable is an index to the MIB table.
  Default: false
  Recommendation: Set to true for an index attribute.
  Possible values: true, false
  Effect of updating: Component indexing could change.
  When does the update take effect: After component rediscovery.
  Required actions for updates to take effect: Update the metric family or change the vendor certification priority.
- **NeedsDelta**
  (Optional) Uses a flag to indicate whether to delta (that is, store the difference between current and last poll for Counters) the MIB variable.
  Default: false
  Recommendation: Set to true if the variable is defined as a Counter, Counter32, Counter64, or TimeTicks quantity in the MIB.
  Possible values: true, false
  Effect of updating: The polled data changes.
  When does the update take effect: Next poll

  NOTE

  This attribute can only be used during polling. This attribute cannot be used during the discovery phases. For example, this attribute cannot be used as part of the Name or Description.
- **Source**
  Specifies the ObjectID of the attribute.
  Recommendation: Set to the fully qualified MIB variable OID that the OBJECT-TYPE defines.
  Possible values: Dot-separated numbers (for example, 1.3.6.1.4.1…)
  Effect of updating: Data is polled from the specified OID.
  When does the update take effect: Next poll

  NOTE

  By default, the `Source` attribute specifies an OID to poll from the device. This default behavior is defined with `src='polled'`. You can set `src='mvel'` to process an MVEL expression using any polled attribute instead of an OID. For example, you could use the `src='mvel'` parameter to combine two 32-bit OIDs into a single 64-bit counter. You could also use the `src='mvel'` parameter to poll a counter that is not stored as a numeric type.

Example: The following example uses the `src='mvel'` parameter to combine two 32-bit OIDs into a single 64-bit counter:

```
<Attribute name="memberbitsout" type="Long">
     <NeedsDelta>true</NeedsDelta>
     <Source src='mvel'>snmpCounter64(memberbitsoutHi32,memberbitsoutLo32)</Source>
</Attribute>
```

NOTE

The `src='mvel'` parameter can only be used during polling. This parameter cannot be used during the discovery phases. For example, this parameter cannot be used as part of the Name or Description.

- **Version**
  Specifies the version of the vendor certification. Update this attribute when you update the certification. You cannot decrease the Version value.
  Possible values: Floating point or decimal number Example: 1.0 or 1.01
- **Author**
  Specifies the creator of the vendor certification.
  Default: "Custom"
  Possible values: Any alphanumeric string.
  Effect of updating: The author attribute is updated.
- **UsesDynamicIndex**(Optional) Enables dynamic ObjectID values for IPSLA polling.
  Default: False
  Possible values: true, false
- **IsDynamicDiscoveryAttribute**(Optional) Specifies whether the attribute should be used to discover the dynamic index.

  WARNING

  If this is not specified, the Data Collector chooses the first dynamic OID it encounters in the attribute group. The device might not support the OID that the Data Collector chooses, which can result in poll failure.

The list of attributes specifies the set of data that a metric family collects when supported by this vendor certification. Typically, this data falls into two categories:

- Configuration data of the device component (such as name or indexes) that is collected only at discovery time.
- Performance data that is collected every poll cycle.

Configuration Data Attributes

An attribute with the name INDEX and type ObjectID is mapped to the Indexes attribute of the target metric family. You can set the value for the Source tag to any variable OID. However, you typically use one of the variables that are listed in the INDEX clause of the table. For example, consider ifIndex in the interfaces table of MIB-2. This variable serves as the index for the other variables in the same MIB table. In addition, the IsIndex tag (and typically also the IsKey tag) for this attribute is set to true.

In this example, attributes such as ifDesc or ifType provide more configuration information about an interface. Therefore, these attributes are useful for the Names and Description attributes of the target metric family.

Performance Data Attributes

These attributes provide the raw data for performance metrics in the target metric family. Consider the following points:

- You can directly map one of these attributes to a metric family performance metric, or
- You can use the attribute in an expression with other attributes to compute a value for the metric.

IndexTagList

To poll attributes from multiple MIB tables, we need an attribute group per MIB table containing these attributes. The index tag list provides a mechanism to relate two attribute groups (or MIB tables) with different indexes. The groups are related such that one item (row) of one table is linked to a corresponding row in a second table.

NOTE

For these items, the following criteria apply to all:

    - All entries in this list can be updated.
    - The update changes indexing.
    - The update takes effect after component rediscovery
    - For the updates to take effect, update the metric family or change the vendor certification priority.

- **PrimaryTag**
  References the primary Attribute group (that is, the group that defines an index attribute with the ObjectID type). The value of this element must equal the 'UseIndex' tag of the attribute group for the primary group.
  Possible values: The ‘UseIndex’ tag of the attribute group corresponding to the primary attribute group.
- **IndexTag**
  Defines how to relate rows of the primary group (or MIB table) to rows in the secondary group. This element relates the rows by specifying attributes of both groups that must match.
- **IndexTag/Name**
  References the secondary group (or MIB table). The value of this element must equal the 'UseIndex' tag of the secondary attribute group that you are trying to relate with the primary one.
  Possible values: The ‘UseIndex’ tag of the secondary attribute group.
- **IndexTag/PrimaryKeyExpression**
  Specifies an MVEL expression containing attributes of the primary attribute group or an attribute group corresponding to any of the previously defined IndexTag. The calculated value is matched up with the 'ThisTagKeyExpression'. If there is a match, the rows of both attribute groups (or MIB tables) are linked. Then, these attributes can be used together in an 'Expression' backing a destAttr (or Metric).
  Possible values: A valid MVEL expression
- **IndexTag/ThisTagKeyExpression**
  Specifies an MVEL expression containing attributes of the secondary attribute group. The calculated value is matched up with the 'PrimaryKeyExpression'. If there is a match, the rows of both groups (or MIB tables) are linked. Then, these attributes can be used together in an 'Expression' backing a destAttr (or Metric).
  Possible values: A valid MVEL expression

ExpressionGroup

The ExpressionGroup maps attributes as follows:

- From the AttributeGroup (that defines how to get a metric from an SNMP MIB)
- To the metrics specified in a metric family (that defines how an attribute is stored in the database)

    NOTE

    The `Filter, VariableGroup, VCSupportExpression, Expression` , and `SetExpression` properties must be listed in this order in the XML.

You can store a MIB value in the database as it is received from the device or after some normalization operations are performed. For example, normalization operations include dividing or multiplying with 1024 to transform to/from kilobytes.

NOTE

Unless specified otherwise, the update takes effect immediately and no actions are required to trigger the update. All entries in this list can be updated.

- **ExpressionGroup/destCert**
  Specifies the metric family that contains the destAttrs to populate.
  Possible values: Any valid metric family
  Effect of updating: Changes the permissible expression destAttr.
- **ExpressionGroup/name**
  (Optional) Specifies the expression group name.
  Possible values: Plain text

Effect of updating: None

- **ExpressionGroup/Filter**
  (Optional) Specifies which components are discovered. Using the Filter reduces the number of components that are managed.

  WARNING

  The expression group filter *does not* exclude the specified components. The filter selects the specified components and excludes components that do not match the criteria.

  Possible values: Boolean MVEL expression using available Attributes
  Effect of updating: Changes which components are discovered.
  When does the update take effect: After component rediscovery.
  Required actions for updates to take effect: Update the metric family or change the vendor certification priority.
- **VariableGroup**
  Defines variables that are used in the ExpressionGroup.

  NOTE

  Within a VariableGroup, variables are processed in the order listed.

  Possible Values: Calculated vendor certification values.
  Effect of updating: Changes which components are discovered.
  When does the update take effect: After component rediscovery.
  Required actions for updates to take effect: Update the metric family or change the vendor certification priority.

  NOTE

  The Juniper and Cisco/Standard High Speed Interface vendor certifications include the UtilizationMaxPercent variable. This variable defines the percentage at which to drop the data for the utilization metric. Dropped data preserves the integrity of rollup data for the interface and results in a gap in views and reports.
- **VCSupportExpression**
  (Optional) Extracts and calculates the MIB attribute values to determine whether the VC is supported.

  NOTE

  You cannot use `IsKey` and `VCSupportExpression` in the same vendor certification. Vendor certifications that contain both `IsKey` and `VCSupportExpression` do not discover anything.

  Possible values: Boolean MVEL expression using available Attributes
  Effect of updating: Changes which components are discovered.
  When does the update take effect: After component rediscovery.
  Required actions for updates to take effect: Update the metric family or change the vendor certification priority.
- **Expression**
  Converts vendor certification attribute values to normalized attribute values.
  Possible values: Normalized attribute value
- **SetExpression**
  Converts normalized attribute values to vendor certification attribute values.
  Possible values: Vendor certification attribute value
  Effect of updating: Changes which components are discovered.
  When does the update take effect: After component rediscovery.
  Required actions for updates to take effect: Update the metric family or change the vendor certification priority.

ExpressionGroup Filter Examples

Filters limit the device components that a vendor certification discovers and polls. Discovery occurs only if all the criteria in the filter are true. If any of the specified attributes in the filter criteria cannot be evaluated for a given component,

the component is discovered because the complete filter criteria could not be evaluated. The most common reason an attribute cannot be evaluated is that the component has no value for that attribute.

Example 1:

```
<Filter>(ifType!=24) &amp;&amp; (ifType!=1)</Filter>
```

DX NetOps Performance Management does not poll the device component if the ifType value is 1 or 24. Interfaces with no value for ifType are discovered and polled.

Example 2:

> **NOTE**
>
> You must use .toString() when comparing OctetString attributes, as OctetStrings are not Strings.

```
<Filter> hrStorageType.toString() == "1.3.6.1.2.1.25.2.1.4" &amp;&amp;
         hrStorageSize != 0
</Filter>
```

DX NetOps Performance Management discovers and polls the device component if the the StorageType is hrStorageFixedDisk (1.3.6.1.2.1.25.2.1.4) and the size is not 0. However, if a component has no value for the hrStorageSize, that component is discovered. If this behavior is not intended, extend the filter to use the **isdef** function to verify that the attribute has a valid value.

```
<Filter> hrStorageType.toString() == "1.3.6.1.2.1.25.2.1.4" &amp;&amp;
         isdef (hrStorageSize) &amp;&amp; hrStorageSize != 0
</Filter>
```

DX NetOps Performance Management discovers and polls the device component if the StorageType is hrStorageFixedDisk and the size is not 0. Only components with a specified value for hrStorageSize are discovered and polled.

Example 3:

> **NOTE**
>
> You must use .toString() when comparing OctetString attributes, as OctetStrings are not Strings.

```
<Filter> (rttMonCtrlAdminRttType==9) &amp;&amp;
         ( !(rttMonCtrlAdminOwner.toString() contains "Network Health") )
</Filter>
```

DX NetOps Performance Management discovers and polls the device component if the rttMonCtrlAdminRttType is 9 and the rttMonCtrlAdminOwner.toString does not contain Network Health.

Expression/destAttr Metrics

The following information describes the Expression/destAttr metrics. You can update all of these metrics:

> **NOTE**
>
> Unless specified otherwise, the update takes effect immediately and no actions are required to trigger the update.

- **Indexes**
  Specifies to use the vendor certification attributes of the ObjectID to define the MVEL expression to provide the value for the Indexes metric family attribute.
  Recommendation: Set to INDEX.
  Possible values: Any attribute that has <IsIndex>true</IsIndex>.

Effect of updating: Component indexing could change.
When does the update take effect: After component rediscovery
Required actions for updates to take effect: Update the metric family or change the vendor certification priority.

- **Names**
  Specifies to use the vendor certification attributes to collect configuration data. This configuration data helps define the MVEL expression to provide the value for the Names metric family attribute.
  Recommendation: Include as much information as necessary to identify an instance uniquely.
  Possible values: String MVEL expression using available Attributes
  Effect of updating: Component name change
  When does the update take effect: After component rediscovery
  Required actions for updates to take effect: Update the metric family or change the vendor certification priority.
- **Descriptions**
  (Optional) Specifies to use the vendor certification attributes to collect configuration data. This configuration data helps define the MVEL expression to provide the value for the Descriptions metric family. Not all metric families support a Descriptions attribute.
  Recommendation: Include as much information that is available to describe an instance.
  Possible values: String MVEL expression using available Attributes
  Effect of updating: Component description change
  When does the update take effect: Component rediscovery
  Required actions for updates to take effect: Update the metric family or change the vendor certification priority.
- **Other Metrics**
  Specifies to use the vendor certification attributes to collect configuration or performance data. This data is used to define the MVEL expression to provide the value for the metric family attribute.
  Can be added: Yes, if the destAttr exists in the metric family.
  Possible values: MVEL expression using available Attributes, producing a value that matches the type of the destination attribute.
  Effect of updating: Polled value changes
  When does the update take effect: Next poll

The metric family exposes URIs (such as, {http://im.ca.com/normalizer}*FamilyName.AttributeName*), which are separately referred to in the ExpressionGroup. The ExpressionGroup/destCert property is set to the URI (for example, {http://im.ca.com/normalizer}*FamilyName*), and the Expression/destAttr is set to *AttributeName*.

Speed Override

Vendor certifications for interfaces include variables in the ExpressionGroup that can override the SpeedIn and SpeedOut values. The SpeedInOverride and SpeedOutOverride variables let you override the SpeedIn and SpeedOut values in the UI. The following example shows how to use the override variables:

```
<VariableGroup>
   <Variable name="SpeedInOverride" providedBy="override"/>
   <Variable name="SpeedOutOverride" providedBy="override"/>
   <Variable name="RawIfSpeed">ifSpeed</Variable>
   <Variable name="CalculatedSpeedIn">
      isdef(SpeedInOverride) ? SpeedInOverride : RawIfSpeed
   </Variable><Variable name="CalculatedSpeedOut">
      isdef(SpeedOutOverride) ?  SpeedOutOverride : RawIfSpeed
   </Variable>
   <Variable name="CalculatedIfInOctets">ifInOctets <= 786432000 ? ifInOctets : null</
Variable>
   <Variable name="CalculatedIfOutOctets">ifOutOctets <= 786432000 ? ifOutOctets :
 null</Variable>
```

```
</VariableGroup>
```

HierarchyList

The following list defines the hierarchy behavior:

> **NOTE**
>
> For these items, the following criteria apply to all:
>
> - All entries in this list can be updated.
> - The update changes the hierarchy construction.
> - The update takes effect after component rediscovery
> - For the updates to take effect, update the metric family or change the vendor certification priority.

- **Hierarchy/ParentFacet**
  Specifies the QName of the facet that is used to find the candidate parent items.
  Possible values: Any valid facet
- **Hierarchy/ParentAttribute**
  Specifies the QName of the attribute that is used to identify the specific parent item.
  Possible values: Any valid attribute QName
- **Hierarchy/ChildAttribute**
  Specifies the QName of the attribute on the child item that is used to match the ParentAttribute on the parent item.
  Possible values: Any valid attribute QName

Multi-MIB Table Support

Some situations exist where you must collect the raw data for a particular metric family from two or more MIB tables. Custom certification includes support for multiple MIB tables. The XML structure is similar to a standard vendor certification, and uses a common key (index) to join data that is collected from multiple tables.

Example:

In this example, the Frame Relay PVCs can be named using a combination of the ifName MIB object in the ifXTable and the frCircuitDlci object that provides the data link connection identifier (DLCI) for this PVC. This kind of naming convention is useful for determining which Frame Relay interface the PVC is layered onto.

To support both MIB tables, add the following information to the XML:

- Add an attribute to the existing AttributeGroup to represent the frCircuitDlci MIB object.
- The ifName MIB object comes from a MIB that is not included in your custom vendor certification. Add an AttributeGroup (in this case, ifXTable), and then add the new attribute (ifName).

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- Auto-generated by the type catalog local manager.-->
<DataModel namespace="http://im.ca.com/certifications/snmp" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance"
 xsi:noNamespaceSchemaLocation="SNMPCertificationFacet.xsd">
   <FacetType name="frPVCInfoCustom"
 descriptorClass="com.ca.im.core.datamodel.certs.CertificationFacetDescriptorImpl">
     <Documentation>Frame Relay PVC Vendor Certification</Documentation>
     <FacetOf namespace="http://im.ca.com/core" name="Item" />
     <AttributeGroup name="ifXTableGroup" external="true" list="true">
```

```
<Documentation>This pulls data from the ifXTable so that the ifName corresponding to the
 PVC can be referenced</Documentation>
<UseIndex>ifXIndexTag</UseIndex>
     <Attribute name="ifXTableIndex" type="ObjectID">
<Documentation />
<IsKey>false</IsKey>
<IsIndex>true</IsIndex>
<Source>1.3.6.1.2.1.31.1.1.1.1</Source>
<Polled>false</Polled>
</Attribute>
<Attribute name="ifName" type="OctetString">
<Documentation />
<IsKey>false</IsKey>
<IsIndex>false</IsIndex>
<Source>1.3.6.1.2.1.31.1.1.1.1</Source>
<Polled>false</Polled>
</Attribute>
</AttributeGroup>
     <IndexTagList>
<PrimaryTag>PVCIndexTag</PrimaryTag>
<IndexTag>
<Name>ifXIndexTag</Name>
<PrimaryKeyExpression>snmpOIDParser(INDEX,1,1)</PrimaryKeyExpression>
<ThisTagKeyExpression>ifXTableIndex</ThisTagKeyExpression>
</IndexTag>
</IndexTagList>
     <AttributeGroup name="AttributeGroup" external="true" list="true">
       <Documentation />
       <UseIndex>PVCIndexTag</UseIndex>
       <Attribute name="INDEX" type="ObjectID">
         <Documentation />
         <Source>1.3.6.1.2.1.10.32.2.1.4</Source>
         <IsIndex>true</IsIndex>
         <IsKey>false</IsKey>
         <NeedsDelta>false</NeedsDelta>
       </Attribute>
       <Attribute name="frCircuitReceivedFECNs" type="Long">
<Documentation />
<Source>1.3.6.1.2.1.10.32.2.1.4</Source>
<IsIndex>false</IsIndex>
<IsKey>true</IsKey>
<NeedsDelta>true</NeedsDelta>
</Attribute>
       <Attribute name="frCircuitReceivedBECNs" type="Long">
         <Documentation />
```

```
          <Source>1.3.6.1.2.1.10.32.2.1.5</Source>
          <IsIndex>false</IsIndex>
          <IsKey>true</IsKey>
          <NeedsDelta>true</NeedsDelta>
        </Attribute>
        <Attribute name="frCircuitSentFrames" type="Long">
          <Documentation />
          <Source>1.3.6.1.2.1.10.32.2.1.6</Source>
          <IsIndex>false</IsIndex>
          <IsKey>true</IsKey>
          <NeedsDelta>true</NeedsDelta>
        </Attribute>
        <Attribute name="frCircuitSentOctets" type="Long">
          <Documentation />
          <Source>1.3.6.1.2.1.10.32.2.1.6</Source>
          <IsIndex>false</IsIndex>
          <IsKey>true</IsKey>
          <NeedsDelta>true</NeedsDelta>
        </Attribute>
        <Attribute name="frCircuitReceivedFrames" type="Long">
          <Documentation />
          <Source>1.3.6.1.2.1.10.32.2.1.8</Source>
          <IsIndex>false</IsIndex>
          <IsKey>true</IsKey>
          <NeedsDelta>true</NeedsDelta>
        </Attribute>
        <Attribute name="frCircuitReceivedOctets" type="Long">
          <Documentation />
          <Source>1.3.6.1.2.1.10.32.2.1.9</Source>
          <IsIndex>false</IsIndex>
          <IsKey>true</IsKey>
          <NeedsDelta>true</NeedsDelta>
         </Attribute>
         <Attribute name="frCircuitState" type="int">
<Documentation />
<Source>1.3.6.1.2.1.10.32.2.1.3</Source>
<IsIndex>false</IsIndex>
<IsKey>false</IsKey>
<NeedsDelta>false</NeedsDelta>
</Attribute>
      <Attribute name="frCircuitDlci" type="int">
<Documentation />
<Source>1.3.6.1.2.1.10.32.2.1.2</Source>
<IsIndex>false</IsIndex>
<IsKey>false</IsKey>
<NeedsDelta>false</NeedsDelta>
```

```
</Attribute>
      </AttributeGroup>
      <Protocol>SNMP</Protocol>
      <DisplayName>Frame Relay PVC Certification</DisplayName>
      <Expressions>
       <ExpressionGroup destCert="{http://im.ca.com/normalizer}frPVCInfo"
 name="frPVCInfoDS">
       <Filter>(frCircuitState==2)</Filter>
       <Expression destAttr="Indexes">INDEX</Expression>
       <Expression destAttr="Names">isdef(ifName)? (isdef(frCircuitDlci) ? ifName + "
 DCLI:" + frCircuitDlci : "Frame Relay " + INDEX) : "Frame Relay " + INDEX</Expression>
       <Expression destAttr="FECNIn">frCircuitReceivedFECNs</Expression>
       <Expression destAttr="BECNIn">frCircuitReceivedBECNs</Expression>
       <Expression destAttr="FramesIn">frCircuitReceivedFrames</Expression>
       <Expression destAttr="FramesOut">frCircuitSentFrames</Expression>
       <Expression destAttr="BytesIn">frCircuitReceivedOctets</Expression>
       <Expression destAttr="BytesOut">frCircuitSentOctets</Expression>
       <Expression destAttr="BitsIn">frCircuitReceivedOctets*8</Expression>
       <Expression destAttr="BitsOut">frCircuitSentOctets*8</Expression>
      </ExpressionGroup>
     </Expressions>
    <MIB>RFC1315-MIB</MIB>
   </FacetType>
</DataModel>
```

AttributeGroup

Each table must go into its own AttributeGroup section. Each Attribute on that table is added as a child of that AttributeGroup.

Refer to these sections for the following information:

- The AttributeGroup information
  Details about the XML elements that are used to define primary and secondary table attributes.
- The UseIndex and IndexTagList information
  Details about the XML elements that are used to join the primary and secondary attribute groups.

In the example, the primary attribute group represents the table that you want to “extend” with more information. The secondary attribute group contains the “extension” information for the primary one.

The primary AttributeGroup contains an Attribute identifying the MIB table variable serving as the common “key” into the secondary AttributeGroup.

The secondary AttributeGroup includes the Attribute definitions for all MIB table variables carrying the “extension” information for the primary table. In addition, there is an Attribute identifying the variable matching the common “key” from the primary AttributeGroup.

UseIndex

Each AttributeGroup is given a UseIndex tag. The UseIndex tag lets you group OIDs under a common name. This common name is associated with a given variable serving as the common key (index) into the respective MIB table.

The following information summarizes the XML elements to customize:

- **AttributeGroup/UseIndex**
  Uniquely identifies the primary and secondary tag name (respectively) that is used in the IndexTagList section.
  Recommendation: Set to the value of the AttributeGroup/name property.

IndexTagList

The IndexTagList section is a mechanism to relate two attribute groups (or MIB tables) with different indexes. When the groups are related, one item has multiple index IDs from multiple tables.

The IndexTagList section contains all the join information, including an IndexTag section for every secondary attribute group.

- **IndexTagList/PrimaryTag**
  Defines the primary attribute group (or MIB table). Set to the value of the UseIndex property of the primary AttributeGroup.
- **IndexTag/Name**
  Defines the secondary attribute group. Set to the value of the UseIndex property of the secondary AttributeGroup.
- **IndexTag/PrimaryKeyExpression**
  Specifies the expression to generate the common key in the primary table. Consider using the MVEL functions to derive a common key from the designated primary table index Attribute.
- **IndexTag/ThisTagKeyExpression**
  Specifies the expression to generate the common key in the secondary table. Consider using the MVEL functions to derive a common key from the designated secondary table index Attribute.

The multitable approach supports the chaining of more than two tables. Two types of relationships exist in multiple table joins:

- **Primary - > Secondary #1, Primary - > Secondary #2**
  Ordering of the secondary tables does not matter in an index tag list.
- **Primary - > Secondary #1 - > Secondary #2**
  List secondary table #1 before secondary table #2 because of the way tables are merged.

One or more rows in the primary table can legally map to the same row in the secondary table. Keys on the secondary table are searched in order, and the first match wins.

Metric Family XML Structure

A metric family uses XML to define the set of metrics to collect and report on for a given technology. These metrics are normalized so that reporting is uniform regardless of the vendor (data source). Metrics are "null" when the vendor does not provide a value. Any report views based on the null metrics are empty.

A metric family also defines attributes that are captured during discovery, like the item name and index. There can also be discovery rules defined that reconcile component matching. You include a metric family in a monitoring profile. The set of metric families in a monitoring profile determines which metrics to collect for the devices in each device collection that is associated with the profile.

WARNING

When you extend a metric family, do not edit restricted tabs or attributes. For more information, see Restricted XML Tags.

NOTE

You must list some properties in the XML in a particular order. The properties included in the XML example and listed in the following descriptions are presented in the recommended order.

Example:

This example metric family supports the vendor certification for Frame-Relay PVC:

> **NOTE**
>
> If you view the metric family XML in a browser, certain tags are hidden. For this reason, copy and paste the metric family XML only from a REST client

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- Auto-generated by the type catalog local manager.  -->
<DataModel xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance namespace="http://
im.ca.com/normalizer" xsi:noNamespaceSchemaLocation="IMDBCertificationFacet.xsd">
  <Author>Custom</Author>
  <Version>1.0</Version>
  <FacetType name="frPVCInfo"
 descriptorClass="com.ca.im.core.datamodel.certs.NormalizedFacetDescriptorImpl">
    <Documentation>Frame Relay Permanent Virtual Circuit</Documentation>
    <FacetOf namespace="http://im.ca.com/core" name="Item" />
    <DisplayName>Frame Relay PVC</DisplayName>
    <TableName>FR_PVC_INFO</TableName>
    <Protocol>IMDB</Protocol>
    <Normalized>true</Normalized>
    <ComponentFacets>
         <Facet>{http://im.ca.com/inventory}frPVC</Facet>
    </ComponentFacets>
    <AttributeGroup name="AttributeGroup" external="true" list="true">
      <Documentation />
      <Attribute name="Indexes" type="ObjectID[]">
         <Documentation />
         <AttributeDisplayName />
         <Polled>false</Polled>
         <Baseline>false</Baseline>
         <IsDbColumn>false</IsDbColumn>
         <Variance>false</Variance>
         <StandardDeviation>false</StandardDeviation>
         <Minimum>false</Minimum>
         <Maximum>false</Maximum>
         <WriteOnPoll>false</WriteOnPoll>
         <RollupStrategy />
         <Percentile>0</Percentile>
       </Attribute>
       <Attribute name="Names" type="String">
         <Documentation>The name of the frame relay circuit</Documentation>
         <AttributeDisplayName />
         <Polled>false</Polled>
         <Baseline>false</Baseline>
         <IsDbColumn>false</IsDbColumn>
         <Variance>false</Variance>
```

```
          <StandardDeviation>false</StandardDeviation>
          <Minimum>false</Minimum>
          <Maximum>false</Maximum>
          <WriteOnPoll>false</WriteOnPoll>
          <RollupStrategy />
          <Percentile>0</Percentile>
        </Attribute>
        <Attribute name="Description" type="String">
          <Documentation>A description for the frame relay circuit</Documentation>
          <AttributeDisplayName />
          <Polled>false</Polled>
          <Baseline>false</Baseline>
          <IsDbColumn>false</IsDbColumn>
          <Variance>false</Variance>
          <StandardDeviation>false</StandardDeviation>
          <Minimum>false</Minimum>
          <Maximum>false</Maximum>
          <WriteOnPoll>false</WriteOnPoll>
          <RollupStrategy />
          <Percentile>0</Percentile>
        </Attribute>
        <Attribute name="BECNIn" type="Double">
          <Documentation>Backward congestion since the virtual circuit was created</
Documentation>
          <AttributeDisplayName />
          <Polled>true</Polled>
          <Baseline>false</Baseline>
          <IsDbColumn>true</IsDbColumn>
          <Variance>false</Variance>
          <StandardDeviation>false</StandardDeviation>
          <Minimum>false</Minimum>
          <Maximum>false</Maximum>
          <WriteOnPoll>false</WriteOnPoll>
          <RollupStrategy>Sum</RollupStrategy>
          <Percentile>0</Percentile>
        </Attribute>
        <Attribute name="FECNIn" type="Double">
          <Documentation>Forward congestion since the virtual circuit was created</
Documentation>
          <AttributeDisplayName />
          <Polled>true</Polled>
          <Baseline>false</Baseline>
          <IsDbColumn>true</IsDbColumn>
          <Variance>false</Variance>
          <StandardDeviation>false</StandardDeviation>
          <Minimum>false</Minimum>
```

```
          <Maximum>false</Maximum>
          <WriteOnPoll>false</WriteOnPoll>
          <RollupStrategy>Sum</RollupStrategy>
          <Percentile>0</Percentile>
        </Attribute>
        <Attribute name="FramesIn" type="Double">
          <Documentation>Frames received since the virtual circuit was created</
Documentation>
          <AttributeDisplayName />
          <Polled>true</Polled>
          <Baseline>false</Baseline>
          <IsDbColumn>true</IsDbColumn>
          <Variance>false</Variance>
          <StandardDeviation>false</StandardDeviation>
          <Minimum>false</Minimum>
          <Maximum>false</Maximum>
          <WriteOnPoll>false</WriteOnPoll>
          <RollupStrategy>Sum</RollupStrategy>
          <Percentile>0</Percentile>
        </Attribute>
        <Attribute name="FramesOut" type="Double">
          <Documentation>Frames sent since the virtual circuit was created</
Documentation>
          <AttributeDisplayName />
          <Polled>true</Polled>
          <Baseline>false</Baseline>
          <IsDbColumn>true</IsDbColumn>
          <Variance>false</Variance>
          <StandardDeviation>false</StandardDeviation>
          <Minimum>false</Minimum>
          <Maximum>false</Maximum>
          <WriteOnPoll>false</WriteOnPoll>
          <RollupStrategy>Sum</RollupStrategy>
          <Percentile>0</Percentile>
        </Attribute>
        <Attribute name="BytesIn" type="Double">
          <Documentation>Bytes received since the virtual circuit was created</
Documentation>
          <AttributeDisplayName />
          <Polled>true</Polled>
          <Baseline>false</Baseline>
          <IsDbColumn>true</IsDbColumn>
          <Variance>false</Variance>
          <StandardDeviation>false</StandardDeviation>
          <Minimum>false</Minimum>
          <Maximum>false</Maximum>
```

```
          <WriteOnPoll>false</WriteOnPoll>
          <RollupStrategy>Sum</RollupStrategy>
          <Percentile>0</Percentile>
        </Attribute>
        <Attribute name="BytesOut" type="Double">
          <Documentation>Bytes sent since the virtual circuit was created</
Documentation>
          <AttributeDisplayName />
          <Polled>true</Polled>
          <Baseline>false</Baseline>
          <IsDbColumn>true</IsDbColumn>
          <Variance>false</Variance>
          <StandardDeviation>false</StandardDeviation>
          <Minimum>false</Minimum>
          <Maximum>false</Maximum>
          <WriteOnPoll>false</WriteOnPoll>
          <RollupStrategy>Sum</RollupStrategy>
          <Percentile>0</Percentile>
        </Attribute>
      </AttributeGroup>
      <Attribute name="SourceFacetTypes" cached="true" list="true" persistent="true"
 type="QName">
        <Documentation />
      </Attribute>
      <Expressions>
        <ExpressionGroup destCert="{http://im.ca.com/core}Item">
          <Expression destAttr="Name">Names</Expression>
        </ExpressionGroup>
        <ExpressionGroup destCert="{http://im.ca.com/inventory}DeviceComponent">
          <Expression destAttr="IndexList">Indexes</Expression>
        </ExpressionGroup>
      </Expressions>
  </FacetType>
</DataModel>
```

Basic Properties

The basic properties of your custom metric family help to distinguish it from other custom metric families you create.

Consider the following restrictions when you determine basic properties:

- The FacetType/name, DisplayName, and TableName properties must be unique for each metric family.
- The Protocol tag is always IMDB.
- The Normalized tag is always true.
- Set the FacetType/descriptorClass property and all DataModel and FacetOf properties.

The following list details basic metric properties:

NOTE

: Unless stated otherwise, all entries in this list can be updated.

- **FacetType/name**
  Specifies the metric family name. For each metric family, the name must be a unique name that identifies it internally within the system. Carefully select a name with a minimal possibility of naming conflicts with future similar metric families. For example, define a naming scheme that ensures that these metric family names are unique.

  NOTE
  This name is never exposed externally. To display a metric family name in the user interface, use the DisplayName element.

  Can be updated: No
  Possible values: Alphanumeric and underscore. Dot and dash are not permitted. The value must be unique across all metric families.
- **Documentation**
  Specifies the external description for the metric family. To make these comments useful, describe why and when you added or changed the metric family.
  Possible values: Plain text
  Effect of updating: None

  NOTE

  This property should be listed first under the `FacetType/name` .
- **FacetOf** Asserts that this metric family is an item.
  Possible values: namespace="http://im.ca.com/core" name="Item"
- **DisplayName**
  Specifies the metric family name that displays in the user interface.
  Possible values: Plain text

  WARNING

  Ensure that the DisplayName property is unique to the metric family.

  Effect of updating: Change to the name in the administrator user interface.
  When does the update take effect: Immediately
  Required actions for updates to take effect: Refresh the user interface.

  NOTE

  This property should be listed before the `AttributeGroup` property.
- **TableName**
  Specifies the database table name that is used to store the metrics that the metric family collects.
  Possible values: Uppercase alphanumeric and underscore. The value must begin with a letter. The value must be unique across all metric families.
  Example: PROCESS_STATS
  Effect of updating: Poll data is stored in a new set of database tables.

  WARNING
  When you update the TableName, the old poll data is lost. Old report views are broken.

  When does the update take effect: Immediately. Before new views can be created, there is a delay of up to 5 minutes while DX NetOps Performance Management loads the new MIB files.
  Required actions for updates to take effect: Views must be recreated.
- **Protocol**
  The Protocol tag is always IMDB.
- **Normalized**
  The Normalized tag is always true.
- **SupportsDeviceAggregation**
  Supports thresholding at the device level for some metrics.

WARNING

This attribute is not supported for custom or extended certification.

ComponentFacets

The ComponentsFacets section lists the facets that are created during discovery. Discovery identifies items as device components or creates a hierarchy relationship between items.

- **Facet**
  Specifies a facet that is attached to the component item during component discovery.
  Can be updated: Yes
  Possible values: QName of the facet
  Effect of updating: If the component facet is synchronized to CA NetOps Portal, the component is visible in CA NetOps Portal.
  When does the update take effect: Rediscover
  Required actions for updates to take effect: Delete the device and rediscover.

ItemFacets

WARNING
ItemFacets is a new section that will likely change to support future, complex metric family structures. Its use is discouraged.

ItemFacets lists the facets that are created during discovery that identify items as devices.

- **Facet**
  Specifies a facet that is attached to the item during discovery.
  Can be updated: Yes
  Possible values: QName of the facet
  Effect of updating: Component is visible on the REST service for the specified facet. If the component facet is synchronized to CA NetOps Portal, the component is visible in CA NetOps Portal.
  When does the update take effect: Rediscover
  Required actions for updates to take effect: Delete the device and rediscover.

Example:

```
<ItemFacets>
   <Facet>{http://im.ca.com/inventory}Host</Facet>
   <Facet>{http://im.ca.com/inventory}Device</Facet>
   <Facet>{http://im.ca.com/inventory}ConsolidatedAndDiscoveredMetricFamilyHistory</
Facet>
   <Facet>{http://im.ca.com/core}Syncable</Facet>
   <!-- The IPDomainID attribute will be filled in by discovery  -->
   <Facet>{http://im.ca.com/core}IPDomainMember</Facet>
</ItemFacets>
```

AttributeGroup (Metric Family)

An AttributeGroup is a collection of item discovery attributes and metric attributes. The item discovery attributes are set during discovery, like item descriptions. The metric attributes are collected during polling. The following information describes the elements that you use in the AttributeGroup section.

Set the AttributeGroup/list and AttributeGroup/external properties to true. These properties specify that each attribute represents a list of values that is obtained from an external source. Customize the following XML elements:

NOTE

All entries in this list can be updated using plain text. The update has no behaviorial impact.

- **AttributeGroup/name**
  Specifies the attribute group name. Conform to the "<FacetType/name>Group" naming scheme.
- **Documentation**
  (Optional) Specifies the description for the attribute group.

General Attributes (Metric Family)

The general attributes for all metric families are as follows:

NOTE

All entries in this list can be updated. Unless specified otherwise, the update takes effect immediately and no actions are required to trigger the update.

- **Attribute/name**
  Specifies the unique, internal name. For metrics, this name is also used for naming the database column.

  NOTE
  This name is never exposed externally. To display an attribute name in the user interface, use the AttributeDisplayName element. To change the AttributeDisplayName, see Create or Extend Metric Families and update the metric family properties.

  Possible values: Alphanumeric and underscore.
  Effect of updating: For metrics, the values for this attribute are stored in a new database column corresponding to the updated name. The user loses the historical data that is collected for this metric (with the older name). The custom reports reporting on this metric fails.
  When does the update take effect: Next poll
  Required actions for updates to take effect: None
- **Attribute/type**
  Indicates the data type of this attribute. The most frequently used data types are Int, Long, Double, String, or ObjectID. The database stores metric attributes as a float. Therefore, these attributes must use a numeric type (we recommend a Double). Other types are used for item attributes.
  Possible values: Boolean, Int, Long, Double (floating-point), BigInteger, String, DateTime, IPAddress, MACaddress, IPSubnet, OctetString (hex representation), ObjectID, ItemID, QName (Qualified Name)

  NOTE
  The type names are case insensitive, for example, "boolean" is the same as Boolean.

  Effect of updating: For metrics, none. All metrics are stored in the database as a float. For item attributes, the device must be deleted and rediscovered.
  When does the update take effect: For metrics, next poll. For item attributes, on rediscover.
  Required actions for updates to take effect: For metrics, none. For item attributes, delete the device and rediscover.
- **Documentation**
  Displays the attribute description in the user interface. The documentation is also displayed in tool tips when you hover the cursor over the attribute name.
  Possible values: Plain text
  Effect of updating: Hovering the cursor over the attribute name shows the updated documentation.

  NOTE

  This property should be listed first under the `Attribute/name` and `Attribute/type`. This property and the `AttributeDisplayName` property must be listed before the other attribute properties in the XML.
- **AttributeDisplayName**

Specifies the name of the attribute in the UI. To change the AttributeDisplayName, see Create or Extend Metric Families and update the metric family properties.
Possible values: Alphanumeric, space, and underscore.
Effect of updating: The metric reflects the updated AttributeDisplayName in the Metric Families UI and custom reports.

NOTE

This property and the `Documentation` property must be listed before the other attribute properties in the XML.

- **AttributeAbbreviation**This parameter is not supported.
- **Polled**
  Indicates whether the attribute is polled. If it is set to false, it is only accessed during discovery.
  Possible values: true, false
  Effect of updating: If set to false, the OIDs corresponding to this attribute/metric are not polled when no other polled attribute/metric is using that OID in its expression. If set to true, the OIDs corresponding to this attribute/metric are polled.
  When does the update take effect: Next poll
  Required actions for updates to take effect: None
- **IsDbColumn**
  Stores its value in the database table. IsDbColumn is used for metric attributes. Set the IsDbColumn value to true when Polled is set to true.
  Possible values: true, false
  Effect of updating: If set to false, the data for this attribute/metric is not stored in the database. If set to true, the data for this attribute/metric is stored in the database.

Discovery Attributes

For many attributes only the value that is retrieved during discovery is stored in the database. No further polling or processing, such as an evaluation of a baseline, is performed.

The Indexes and Names attributes must exist for all metric families. The Descriptions attribute is optional.

```
<Attribute name="Indexes" type="ObjectID[]"  />
<Attribute name="Names" type="String"  />
<Attribute name="Descriptions" type="String"  />
```

The metric families supporting Hierarchy must include these attributes:

```
<Attribute name="ItemUniqueIDs" type="String" />
<Attribute name="ParentUniqueIDs" type="String" />
```

Polled and Baseline Attributes

The following information describes the polled and baseline attribute elements:

NOTE

All entries in this list can be updated.

- **Baseline**
  Indicates whether to calculate a mean value for this attribute. If it is set to true, a corresponding BaselineList definition must be defined.

  NOTE
  The Baseline attribute requires that the StandardDeviation attribute is set to true.

Possible values: true, false
Effect of updating: Baseline values are calculated when true.
When does the update take effect: Next poll
Required actions for updates to take effect: None

- **Maximum**
  Indicates whether to calculate the maximum of this attribute during the rollup. Creates a 'max_' column in the database table. If RollupStrategy is defined, this attribute must also be defined.
  Possible values: true, false
  Effect of updating: True provides a calculation of, and a reporting field for, “Maximum.”
  When does the update take effect: Next poll
  Required actions for updates to take effect: None
- **Minimum**
  Indicates whether to calculate the minimum of this attribute during the rollup. Creates a 'min_' column in the database table. If RollupStrategy is defined, this attribute must also be defined.
  Possible values: true, false
  Effect of updating: True provides a calculation of, and a reporting field for, “Minimum.”
  When does the update take effect: Next poll
  Required actions for updates to take effect: None
- **StandardDeviation**
  Indicates whether to calculate the standard deviation of this attribute during the rollup. Creates a 'std_' column in the database table. If RollupStrategy is defined, this attribute must also be defined.
  Possible values: true, false
  Effect of updating: True provides a calculation of, and a reporting field for, “Standard Deviation.”
  When does the update take effect: Next poll
  Required actions for updates to take effect: None
- **DeviationFromBaseline**
  Requires that the Baseline attribute is set to true. Provides two extra reporting fields, “Average Baseline” and “Percent Deviation,” calculated using baseline data. These fields are not available for building custom views. No changes are made to the database table.
  Possible values: true, false
  Effect of updating: True provides the “Average Baseline” and “Percent Deviation” fields for the internal report development.
  When does the update take effect: Immediately
  Required actions for updates to take effect: None
- **Percentile**
  Indicates whether to calculate the 95th percentile of this attribute during the rollup. Creates a 'pct_' column in the database table. If RollupStrategy is defined, this attribute must also be defined.
  Possible values: 0, 95
  Effect of updating: A value of 95 provides a calculation of, and a reporting field for, “95th Percentile.” Zero specifies that no calculation is performed, and the reporting field is not available.
  When does the update take effect: Next poll
  Required actions for updates to take effect: None
- **Percentile2**(Optional) Specifies the value of a user-configurable percentile.
  Possible values: 0-99
  Effect of updating: A non-zero value specifies the percentile to calculate. Zero specifies that no calculation is performed, and the reporting field is not available.
  When does the update take effect: Next poll
  Required actions for updates to take effect: None
- **Percentile3**(Optional) Specifies the value of a user-configurable percentile.
  Possible values: 0-99

Effect of updating: A non-zero value specifies the percentile to calculate. Zero specifies that no calculation is performed, and the reporting field is not available.
When does the update take effect: Next poll
Required actions for updates to take effect: None

NOTE

You cannot set two percentiles to the same non-zero value. You cannot set Percentile2 or Percentile3 to 95.

WARNING

Changes to **Percentile2** and **Percentile3** may cause discontinuity in the trend view depending on the time range.

- **ProjectionPercentile**
(Optional) Specifies the percentile to calculate for metric projection.
Possible values: 0-99
Effect of updating: Changes the percentile to use to calculate projections. Zero specifies that no calculation is performed.

WARNING

Changes to **ProjectionPercentile** cause inaccurate projections for up to 90 days. When you change the value of **ProjectionPercentile**, the percentile values for days before the change are not recalculated.

When does the update take effect: Next poll
Required actions for updates to take effect: None

- **AggregateToDevice**Supports thresholding at the device level for some metrics.

WARNING

This attribute is not supported for custom or extended certification.

- **RollupStrategy**
Specifies the operation that is performed every cycle during the rollup of the individually polled values. When Polled and IsDbColumn are set to true, this element is required.
Possible values: Sum (a summation for counters), Avg (an average for gauges)
Effect of updating: The specified strategy is used to perform rollup calculations.
When does the update take effect: Next poll
Required actions for updates to take effect: None

- **Rate**
Provides an extra reporting field, “Average Rate,” calculated as AVG (metric value / time). No changes are made to the database table.

NOTE
The Rate is available for reporting but not for use when monitoring the profile event rules.

Possible values: true, false
Effect of updating: Provides the “Average Rate” field for reporting.
When does the update take effect: Immediately
Required actions for updates to take effect: None

- **Units**
Specifies the name of the units label used in reports. The actual label that is displayed is translated according to the language setting of the report.

WARNING

If this attribute is not defined, the units labels on reports is 'Units'.

Possible values: Percent, Packets, PacketsPerSecond, DiscardedPackets, ErroredPackets, Bits, BitsPerSecond, Bytes, BytesPerSecond, Seconds, Microseconds, Milliseconds, UnixTime, Observations, FramesPerSecond, Frames, RequestsPerSecond, Requests
Effect of updating: The specified units label is displayed in reports. **When does the update take effect:** Immediately **Required actions for updates to take effect:** None

Example: Using polled and baseline attribute elements

```
<Attribute name="Utilization" type="double">
   <AttributeDisplayName>Utilization</AttributeDisplayName>
   <IsDbColumn>true</IsDbColumn>
   <Baseline>true</Baseline>
   <Minimum>true</Minimum>
   <Maximum>true</Maximum>
   <RollupStrategy>Avg</RollupStrategy>
   <StandardDeviation>true</StandardDeviation>
   <DeviationFromBaseline>true</DeviationFromBaseline>
   <Percentile>95</Percentile>
   <Polled>true</Polled>
   <Units>Percent</Units>
</Attribute>
```

SourceFacetTypes Attribute

A **SourceFacetTypes** attribute is required for discovery and must be defined.

Use these required values:

- Name: SourceFacetTypes
- Type: QName
- Cached: true
- Persistent: true
- List: true

Example: <Attribute name="SourceFacetTypes" type="QName" cached="true" persistent="true" list="true" />

Can be updated: No

Expressions

The Expressions section is composed of ExpressionGroup tags that are used for component discovery. During the component discovery, the values for the component item properties (such as the IndexList, Name, and Description) are calculated. The vendor certification expressions supporting the metric family expressions are used for this calculation.

NOTE
Do not confuse the metric family and vendor certification ExpressionGroup tags.

The ExpressionGroup tags for the following DestCert URIs must exist:

DestCert	DestAttr
{http://im.ca.com/core}Item	Name
{http://im.ca.com/core}Item	Description
{http://im.ca.com/inventory} DeviceComponent	IndexList

NOTE

All entries in this list can be updated.

- **ExpressionGroup/name**
  (Optional) Specifies the expression group name.
  Possible values: Plain text
  Effect of updating: None
- **ExpressionGroup/destCert**
  Specifies the component facet that contains the destAttrs to populate. The facet name typically comes from the ComponentFacets section, except the Item and DeviceComponent facets.
  Possible values: Facets that are defined in ComponentFacets, or the Item, DeviceComponent facet.
  Effect of updating: Changes permissible expression destAttr
  When does the update take effect: Component Rediscovery
  Required actions for updates to take effect: None
- **ExpressionGroup/Expression**
  Specifies the expression for the component facet attribute.
  Possible values: Any valid metric
  Effect of updating: Changes permissible expression destAttr
  When does the update take effect: Component Rediscovery
  Required actions for updates to take effect: None
- **ExpressionGroup/Expression/destAttr**
  Specifies the component facet attribute name.
  Possible values: Any valid attribute from that component facet.
  Effect of updating: Changes the attribute name
  When does the update take effect: Component Rediscovery
  Required actions for updates to take effect: None

Hierarchy

A hierarchy, or parent-child relationship, can be defined between items of different metric families, for example, Interface and CBQoS Classmap. In the metric family definitions, the Hierarchy must be specified in the child metric family with:

- The Hierarchy QName in the ComponentFacets
- The ItemUniqueID and ParentUniqueID destAttr values in the Hierarchy ExpressionGroup
- The ItemUniqueIDs and ParentUniqueIDs attributes in the AttributeGroup

The supporting expressions are defined in the vendor certifications.

The Hierarchy ExpressionGroup tags for the following DestCert URIs must exist:

DestCert	DestAttr
{http://im.ca.com/inventory}Hierarchy	ItemUniqueID
{http://im.ca.com/inventory}Hierarchy	ParentUniqueID

Example:

```
<ComponentFacets>
   <Facet>{http://im.ca.com/inventory}QoSClassMap</Facet>
   <Facet>{http://im.ca.com/inventory}Hierarchy</Facet>
</ComponentFacets>
<ExpressionGroup name="Hierarchy" destCert="{http://im.ca.com/inventory}Hierarchy">
   <Expression destAttr="ItemUniqueID">ItemUniqueIDs</Expression>
```

```
    <Expression destAttr="ParentUniqueID">ParentUniqueIDs</Expression>
 </ExpressionGroup>
 <AttributeGroup name="QosCosGroup" list="true" external="true">
      <Attribute name="ItemUniqueIDs" type="String" />
      <Attribute name="ParentUniqueIDs" type="String" />
      ...
 </AttributeGroup>
```

BaselineDefinitions

The BaselineDefinitions section contains the baseline definitions to calculate for this metric family. A baseline definition must be specified for each metric in the AttributeGroup section whose Baseline property is set to true.

You can define two types of baselines: Hourly (required) and Daily (optional). Hourly baselines are used both for event processing and for displaying baselines in reports. Daily baselines are used for displaying baselines in reports with a time frame of one month or greater.

The following information describes the baseline elements used:

- **Name**
  Specifies the type of baseline definition for a metric. The type is either hourly or daily.
  Can be updated: No
  Possible values: HourlyBaseline, DailyBaseline
- **ID**
  Specifies a value that is no longer used. However, the field must be specified as a positive integer and must be unique across all hourly and daily baseline definitions within this metric family.
  Can be updated: Yes
  Possible values: Any unique and positive integer.
  Effect of updating: None
- **PerformanceMetric**
  Specifies the name (case sensitive) of the metric for which the baseline is calculated. Set the Polled and Baseline properties for the metric attribute to true.
  Can be updated: Yes
  Possible values: A valid metric name (case sensitive).
  Effect of updating: Baseline calculations are performed for the metric.
  When does the update take effect: Next baseline calculation, either hourly or daily.
  Required actions for updates to take effect: None
- **Period**
  Specifies the type of baseline calculation, hourly or daily. Specify the value "1 Hour" for an HourlyBaseline Name or "1 Day” for a DailyBaseline Name.
  Can be updated: Yes
  Possible values: 1 Hour, 1 Day
  Effect of updating: Baseline calculations are performed either hourly or daily.
  When does the update take effect: Next baseline calculation, either hourly or daily
  Required actions for updates to take effect: None
- **ProjectionInterval**
  (Optional) Specifies the projection period in days for metric projection. Specify this expression in DailyBaseline.
  Can be updated: Yes
  Possible values: 0 or any positive integer
  Effect of updating: Changes the period that the system calculates projection values for. Zero specifies that no calculation is performed.
  When does the update take effect: Next baseline calculation.

Required actions for updates to take effect: PredicitionInterval requires that ProjectionPercentile is set to a non-zero value.

- **ProjectionInterval2**
  (Optional) Defines a second projection interval. For details, see ProjectionInterval.
- **ProjectionInterval3**(Optional) Defines a third projection interval. For details, see ProjectionInterval.
- **Window**
  Specifies a value that is no longer used. However, the field must be specified as “30 Days” for hourly baselines and “90 Days” for daily baselines.
  Can be updated: No
  Possible values: 30 Days, 90 Days
- **StartDate, EndDate, DaysOfWeek**
  Specifies more values that are not used, but they must be specified as 0 (zero).
  Can be updated: No
  Possible values: 0

ComponentReconciliation

The following information defines the component reconciliation logic that is used in component discovery. First, this information determines whether the system has already discovered this component or not. Then, the reconciliation logic determines whether to update an existing component or create a new one.

There can be multiple ordered algorithms per metric family. If a metric family does not define a reconciliation algorithm, a default one with match attribute Item.Name is applied.

ItemReconciliation

> **WARNING**
> ItemReconciliation is a new section that will likely change to support future, complex metric family structures. Its use is discouraged.

The following information defines the item reconciliation logic that is used in item discovery. The logic determines whether the system has already discovered an item or not. Based on this determination, an existing item is updated or a new item is created. Item reconciliation is similar to component reconciliation. However, item reconciliation is used for items that are not components, such as virtual hosts. The ItemFacets are added to any new items or to any matching items (if the facets do not exist).

Example:

```
<ItemReconciliation>
   <SourceAgentScopedReconciliation>
    <MatchAlgorithms>
       <ExactMatch>
          <MatchAttribute name="{http://im.ca.com/
inventory}SourceAgentInfo.SourceAgentIndexes" />
       </ExactMatch>
    </MatchAlgorithms>
   </SourceAgentScopedReconciliation>
   <GlobalScopedReconciliation matchDevices="true" />
</ItemReconciliation>
```

- **SourceAgentScopedReconciliation**
  Defines the match algorithms that are used to reconcile items.
  Can be updated: Yes
  Effect of updating: Changes the item reconciliation logic.

When does the update take effect: Rediscovery
Required actions for updates to take effect: Update the metric family or change the vendor certification priority.

- **GlobalScopedReconciliation**
  Defines the match algorithms that are used when items could not be reconciled for the source agent. The GlobalScopedReconciliation algorithms are used to locate items that have been created for other agents but match the potential new items. If the matchDevices property is set to true, the system default (built-in, not visible in XML) ComponentReconciliationInfo match algorithm is used. The match algorithm is based on the device primary IP address and host name.
  Can be updated: Yes
  Effect of updating: Changes the item reconciliation logic.
  When does the update take effect: Rediscovery
  Required actions for updates to take effect: Update the metric family or change the vendor certification priority.

MatchAlgorithms

Component reconciliation and item reconciliation use match algorithms. Two match algorithms are supported:

- **ExactMatch**All specified attributes must be matched to say the item matches with the new data.
- **BestOfMatch**
  Users must specify the least number of attributes to be matched by using the "leastMatchCount" value. Also, each attribute has a "required" key property. If the required property is set to true, that attribute must be matched to be considered a match.

The algorithm has a match precedence when multiple algorithms are provided for a metric family. The order of the algorithms determines the precedence. The algorithm at the top has the highest precedence. The bottom one has the lowest precedence.

Each algorithm must have at least one matching attribute. When data matches to multiple items with the same algorithm, the item with the most matched attributes wins. When multiple matched items have the same number of matched attributes, the winner is picked at random from these items.

Examples: How the reconciliation works

Two match algorithms are provided for a metric family: alg1 and alg2. Alg1 has higher precedence than alg2. The metric family has three existing component items: 1, 2 and 3. Rediscovering the metric family finds three entries: A, B, and C. Now, we apply the two algorithms to determine which entry is new, changed, and unchanged.

```
    Reconciliation Meta Data                    New Data          Existing Components
 <ComponentReconciliation>                         A                   1
   <MatchAlgorithms>                               B                   2
     <MatchAlgorithm1>                             C                   3
       <MatchAttribute name="attr1"/>
       <MatchAttribute name="attr2"/>
     </MatchAlgorithm1>


     <MatchAlgorithm2>
       <MatchAttribute name="attr1"/>
       <MatchAttribute name="attr3"/>
       <MatchAttribute name="attr4"/>
     </MatchAlgorithm2>


   </MatchAlgorithms>
```

```
</ComponentReconciliation>
```

<MatchAlgorithm1> and <MatchAlgorithm2> can be either <ExactMatch> or <BestOfMatch>. The order of the two match algorithms tells us that MatchAlgorithm1 has a higher precedence than MatchAlgorithm2.

- **Case 1: Unique 1-to-1 Match**
  Entry A matches to item 1, and item 1 does not have any other match.

  ```
  A -----> 1
  ```

  This example is the simplest case. This match is unique, so it does not matter if it also matches alg1 or alg2. Entry A matches item 1.
  A good match algorithm produces more unique matches.
- **Case 2: One entry has multiple matches**
  Entry A matches to item 1 by alg1 and also matches to 2 by alg2.

  ```
             ---> 1 (alg1)  (1 wins)
       /
     A                    Since alg1 has higher precedence, item 1 wins the match.
       \
             ---> 2 (alg2)
  ```
- **Case 3: Multiple entries match to the same item with different algorithms**
  Entry A matches to 1 by alg1 and entry B also matches to item 1 by alg2.

  ```
  A ------> 1 (alg1)  (A wins)
  B ------> 1 (alg2)
  ```

  Since alg1 has higher precedence, entry A wins.
- **Case 4: Multiple entries match to the same item with same algorithm but different numbers of matched attributes**
  Both A and B match to 1 by alg1.

  ```
  A ------> 1 (alg1, # of matched attrs: 2)  (A wins)
  B ------> 1 (alg1, # of matched attrs: 1)
  ```

  Because A has more matched attributes, A wins.
  If the number of match attributes is the same, the winner is randomly picked and a warning is generated.
- **Case 5: Mixed match 1**

  ```
   alg1
    A ----------------------------------> 1
      /  alg2(match attr count: 3)
    B
      \  alg2(match attr count: 2)
             --------------------------------> 2
  ```

  A matches to 1 because it matches with a higher precedence algorithm.
  B matches to 2 because 1 has matched to A.
- **Case 6: Mixed match 2**

  ```
         ---------> 3
      /  alg1
    A                ==> A wins 3 because alg1 has a higher matching precedence
      \  alg2
         ---------> 1
      /  alg2
    B                ==> B wins 2 because alg1 has a higher matching precedence
      \  alg1
  ```

```
         ---------> 2
      /  alg2
   C                    ==> C has no match because 2 is matched to B and 3 is matched
  to A
      \  alg2
         ---------> 3
```

Entry C is treated as a new component. 1 is considered as an unmatched item.
The more match case 1 (unique match), the better the match algorithm is.
Can be updated: Yes
Effect of updating: Changes the component reconciliation logic.
When does the update take effect: Rediscovery
Required actions for updates to take effect: Update metric family or change vendor certification priority.

ReconfigDetectionAttr

This element defines a metric family attribute that is used for change detection. You can enable the change detection on a monitoring profile. Data Aggregator polls that attribute only to verify whether the target device has changed, instead of doing a complete rediscovery. This feature helps performance and helps reduce the network traffic.

- **Can be updated:** Yes
- **Possible values:** The full name of the metric family attribute. The specified metric family attribute flag must have the cached, persistent, and external flags set to true.
- **Effect of updating:** Change to component reconfiguration detection.
- **When does the update take effect:** After rediscovery
- **Required actions for updates to take effect:** Update metric family or change vendor certification priority.

Tags in Custom and Extended Metric Families

The following tags have restricted use in custom and extended metric families:

- **Variance**The value of this tag must be 'false'.
- **RollupExpression**
  Remove this tag or the value must be blank.
- The Normalized tag is always true.

Component XML Structure

A device component uses XML to define a class of component items that are associated with a device. Many factory-defined components are provided, but a custom component is typically defined for a custom metric family. Components can define an optional ItemSyncDefinition, which synchronizes component items to NetOps Portal. You can then view the components in Inventory lists, groups, and context pages.

NOTE

You must list some properties in the XML in a particular order. The properties included in the XML example and listed in the following descriptions are presented in the recommended order.

Example:

This example shows a custom component named frPVC:

NOTE

If you view the component XML in a browser, certain tags are hidden. For this reason, copy and paste the component XML only from a REST client

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Auto-generated by the type catalog local manager.-->
<DataModel namespace="http://im.ca.com/inventory" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance" xsi:noNamespaceSchemaLocation="ComponentFacet.xsd">
  <Author>Custom</Author>
  <Version>1.0</Version>
  <FacetType name="frPVC">
    <Documentation>A Frame Relay PVC</Documentation>
    <FacetOf namespace="http://im.ca.com/core" name="Item" />
    <Component>true</Component>
    <ItemSyncDefinition itemTypeName="component" itemSubtypeName="frpvc"
 itemTypeLabel="FrameRelayPVC" itemTypeLabelPlural="FrameRelayPVCs" categorize="false"
 groupBy="false" context="true">
    </ItemSyncDefinition>
  </FacetType>
</DataModel>
```

Set the context in the ItemSyncDefinition to “true” to enable a link to a custom context page. You can navigate to this page from the frPVC device component where it appears in the Inventory, Device Components list. Setting it to “true” also enables you to select the metric family as a “context,” which makes your custom metric family available in the Dynamic Trend chart type.

Basic Properties

The basic properties of your custom component help to distinguish it from other custom components that you create.

NOTE

In the following list, only the documentation property can be updated.

- **FacetType/name**
  Specifies the component name. Each component must have a unique name that identifies it internally within the system. Carefully choose a name with a minimal possibility of naming conflicts with future similar components. For example, define a naming scheme that ensures that these component names are unique.

  NOTE

  This name is never exposed externally. To display a component name in the user interface, use the `ItemSyncDefinition` , `itemTypeLabel` , and `itemTypeLabelPlural` elements.

  Possible values: Alphanumeric and underscore. Dot and dash are not permitted.
  - **Documentation**
    Specifies internal comments for the component. To make these comments useful, describe why and when you added or changed the component.
    Possible values: Plain text
    Effect of updating: None
    When does the update take effect: Immediately
    Required Actions for updates to take effect: None

NOTE

This property should be listed first under the `FacetType/name` . This property and the `FacetOf` property must be listed before the `Component` and `ItemSyncDefinition` properties in the XML.

- **FacetOf**Asserts that this component is an item.
  Possible values: namespace="http://im.ca.com/core" name="Item"

NOTE

This property and the `Documentation` property must be listed before the `Component` and `ItemSyncDefinition` properties in the XML.

- **Component**
  Asserts that this item is a component.
  Possible values: true

ItemSyncDefinition

The ItemSyncDefinition attribute is optional. This attribute specifies how the component items are synchronized and displayed in NetOps Portal. If the component items are not specified, they do not display in the NetOps Portal Inventory lists (for example, Device Components). But, they can still be reported on in custom views.

NOTE

Unless otherwise, there are no required actions for the updates to take effect.

- **ItemSyncDefinition/itemTypeName**
  Specifies the item type. For custom components, this value must be “component.”
  Can be updated: No
  Possible values: Component
- **ItemSyncDefinition/itemSubtypeName**
  Specifies the internal name of the component in NetOps Portal. This value must be unique for all components. Use a naming convention that avoids conflicts with future factory and custom components, such as using a prefix representing your organization, 'acmeFan'.
  Can be updated: No
  Possible values: Alphanumeric, unique for all components
- **ItemSyncDefinition/itemTypeLabel**
  Specifies the user interface label that is used when displaying a single component of this type. For example, this value is used in the Inventory, Device Components UI “Type” column.
  Can be updated: Yes
  Possible values: Plain text, unique for all components
  Effect of updating: Label is displayed in NetOps Portal Inventory user interfaces.
  When does the update take effect: Allow for resync to occur and up to 15 minutes to complete updates.
- **ItemSyncDefinition/itemTypeLabelPlural**
  Specifies the user interface label that is used when displaying multiple components of this type. Used by the Inventory menu (see groupBy) and the Group name (see categorize).
  Can be updated: Yes
  Possible values: Plain text, unique for all components
  Effect of updating: Label is displayed in NetOps Portal Inventory UIs.
  When does the update take effect: Allow for resync to occur and up to 15 minutes to complete updates.
- **ItemSyncDefinition/categorize**
  Instructs NetOps Portal to create an inventory group under Inventory, All Items. This group contains all items of this component type. The group is named '*{itemTypeLabelPlural}'.*

NOTE
The inventory group that is created cannot be used for reporting dashboards. Instead, this group is for inventory purposes only. If the group is selected for reporting, then no data displays. Other, device-based inventory groups (under Inventory, All Items) can be used for reporting, such as Routers and Servers. However, component-based inventory groups cannot, such as Device Components.

Can be updated: Yes
Possible values: true, false
Effect of updating: Creates or removes an Inventory group in NetOps Portal. For NetOps Portal group administrators, on the Manage Groups page, the group is created under Inventory, All Items, *{itemTypeLabelPlural}*.
When does the update take effect: Allow for resync to occur and up to 30 minutes to complete updates.
Required actions for updates to take effect: Items typically appear in the group within 30 minutes. If they do not, manually resync the Data Aggregator datasource. Select 'Perform a Full Resynchronization' before clicking the Resync confirmation button.

- **ItemSyncDefinition/groupBy**
  Instructs NetOps Portal to create an inventory menu item (under Inventory) for viewing all items of this component item type. The menu is named "*{itemTypeLabelPlural}*." This attribute also causes the component type show up in the Context Type drop-down list when setting a view context. When false, the components are listed in the Inventory, Device Components table with the type of "*{itemTypeLabel}*." The groupBy property does not create a group (see categorize).
  Can be updated: Yes
  Possible values: true, false
  Effect of updating: The menu item is created when true, or removed when false.
  When does the update take effect: Allow for resynchronization to occur and allow up to 15 minutes to complete updates.
- **ItemSyncDefinition/context**
  Makes each component item name a context hyperlink in the Inventory component views that, when clicked, navigate to the individual component context page.
  Can be updated: Yes/No
  Possible values: Plain text
  Effect of updating: Makes each component item name a context hyperlink in the Inventory component views.
  When does the update take effect: Allow for resynchronization to occur and allow up to 15 minutes to complete updates.

Remove an ItemSyncDefinition

Completely removing an ItemSyncDefinition requires a specific procedure.

Follow these steps:

1. Remove the ItemSyncDefinition section, including the <ItemSyncDefinition> start and end tags.
2. After you apply the component change to Data Aggregator, log in to NetOps Portal as the administrator.
3. Select the DataAggregator datasource on the Data Sources page.
4. Click the Resync button.
5. Check the “Perform a Full Resynchronization” option, then click the Resync confirmation button.
   The resync process begins. Allow between 15-30 minutes for the changes to synchronize.
   When this process completes, all NetOps Portal behaviors that the ItemSyncDefinition defined for the component are removed.

Unsupported Properties for Custom Components

The following properties are *not* supported for custom components and should not be present in your XML:

- Attribute
- WebService
- ItemSyncDefinition/isDeviceComponent
- ItemSyncDefinition/mapped
- ItemSyncDefinition/ItemProperty

Self-Certification Workflows

Self-certification workflows are examples of real use cases that you may encounter. Use the following workflows as a model to help you with self-certification:

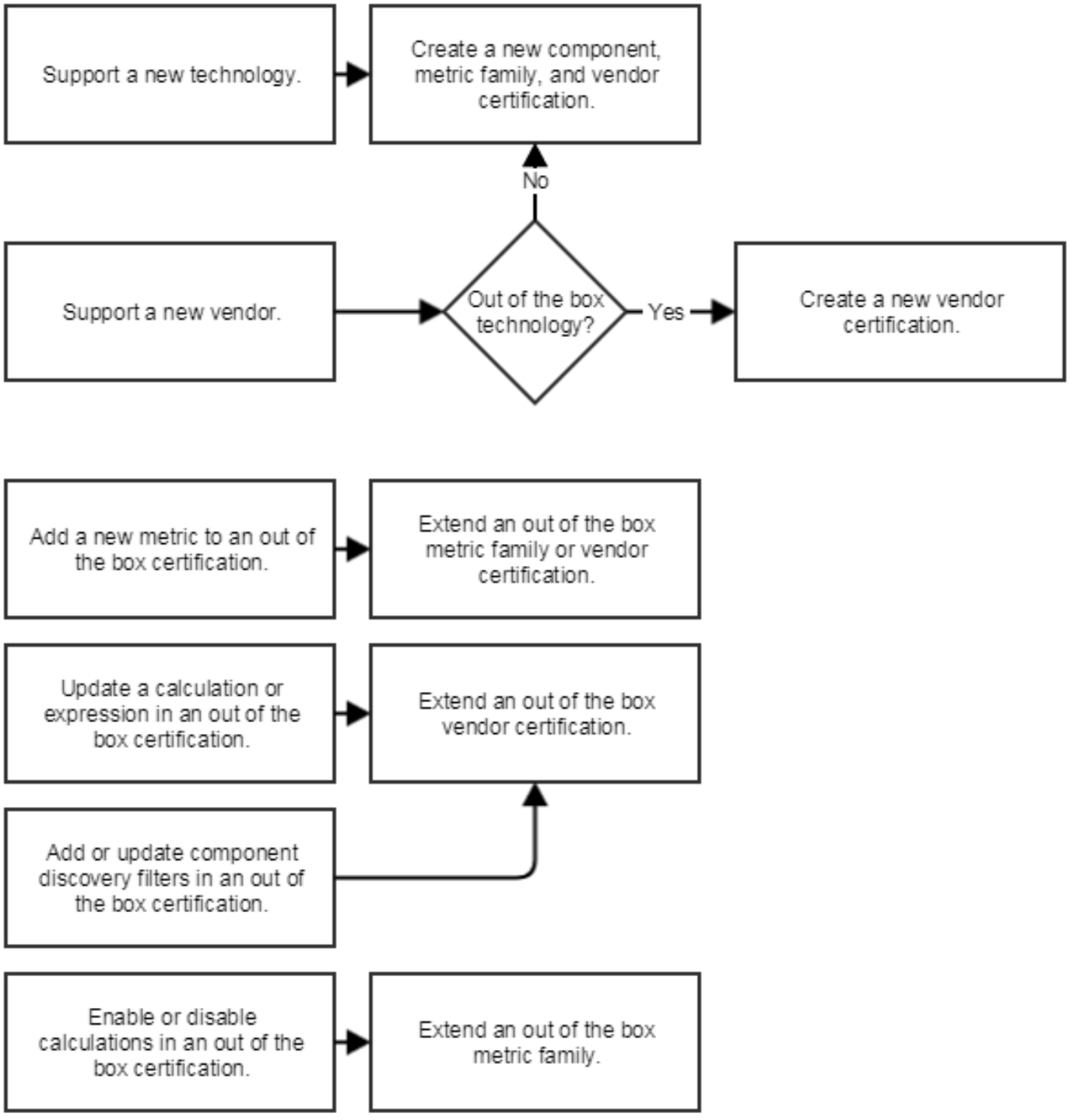
- Basic Self-Certification Workflow

Basic Self-Certification Workflow

You can create a vendor certification using the following basic self-certification workflow. This workflow covers the basic concepts and requirements to create a component, metric family, and vendor certification using the available REST services:

The following graphic provides a high-level overview of some common self-certification scenarios:

Figure 7: Self-Cert



NOTE
If you design a new vendor and technology certification containing only device-level metrics, a component definition is unnecessary (for example, see the Availability metric family).

For more information, see the following articles:

- Create Custom Components
- Create or Extend Metric Families
- Create or Extend Vendor Certifications

Create a New Component

You can introduce a new managed item type using the components XML. The components XML defines how a discovered component is synced over to NetOps Portal. The XML creates a component in your inventory and is required only when a component does not exist already.

Follow these steps:

1. Customize the following XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Auto-generated by the type catalog local manager.-->
<DataModel namespace="http://im.ca.com/inventory"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="ComponentFacet.xsd">
  <Author>Custom</Author>
  <Version>Version</Version>
  <FacetType name="Name">
    <Documentation>Clear_Text_Description</Documentation>
    <FacetOf namespace="http://im.ca.com/core" name="Item" />
    <Component>true</Component>
    <ItemSyncDefinition itemTypeName="component"
itemSubtypeName="Internal_Name"
itemTypeLabel="External_Name"
itemTypeLabelPlural="External_Names_Plural"
categorize="false"
groupBy="false"
context="true">
    </ItemSyncDefinition>
  </FacetType>
</DataModel>
```

- **Version**
  Specifies the sequential number of the version definition. If you configure the version correctly, it should not change frequently.
  Example: `1.0`
- **Name** Specifies the component name. Each component must have a unique name that identifies it internally within the system. Carefully choose a name with a minimal possibility of naming conflicts with future similar components. For example, define a naming scheme that ensures that these component names are unique. This string is used in the Component Facet Definition in the Metric Family and cannot be updated.
  Possible values: Alphanumeric and underscore. Dot and dash are not permitted.

  NOTE
  This name is never exposed externally. To display a component name in the user interface, use the `ItemSyncDefinition` , `itemTypeLabel` , and `itemTypeLabelPlural` elements.
- **Clear_Text_Description**
  Specifies internal comments for the component. To make these comments useful, describe why and when you added or changed the component.
  Possible values: Plain text

Effect of updating: None
When does the update take effect: Immediately
Required Actions for updates to take effect: None

- **Internal_Name** Specifies the internal name of the component in NetOps Portal. This value must be unique for all components. Use a naming convention that avoids conflicts with future factory and custom components, such as using a prefix representing your organization, '`acmeFan` '.
  Can be updated: No
  Possible values: Alphanumeric, unique for all components
- **External_Name** Specifies the user interface label that is used when displaying a single component of this type. For example, this value is used in the Inventory, Device Components UI “Type” column.
  Can be updated: Yes
  Possible values: Plain text, unique for all components
  Effect of updating: Label is displayed in NetOps Portal Inventory user interfaces.
  When does the update take effect: Allow for resynchronization to occur and up to 15 minutes to complete updates.
- **External_Name_Plural**
  Specifies the user interface label that is used when displaying multiple components of this type. Used by the Inventory menu (see `groupBy`) and the Group name (see categorize).
  Can be updated: Yes
  Possible values: Plain text, unique for all components
  Effect of updating: Label is displayed in NetOps Portal Inventory UIs.
  When does the update take effect: Allow for resynchronization to occur and up to 15 minutes to complete updates.
- **categorize**
  Instructs NetOps Portal to create an inventory group under Inventory, All Items. This group contains all items of this component type. The group is named '*{*`itemTypeLabelPlural`*}'.*

  > **NOTE**
  > The inventory group that is created cannot be used for reporting dashboards. Instead, this group is for inventory purposes only. If the group is selected for reporting, then no data displays. Other, device-based inventory groups (under Inventory, All Items) can be used for reporting, such as Routers and Servers. However, component-based inventory groups cannot, such as Device Components.

  Can be updated: Yes
  Possible values: true, false
  Effect of updating: Creates or removes an Inventory group in NetOps Portal. For NetOps Portal group administrators, on the Manage Groups page, the group is created under Inventory, All Items, *{* `itemTypeLabelPlural}` .
  When does the update take effect: Allow for resynchronization to occur and up to 30 minutes to complete updates.
  Required Actions for updates to take effect: Items typically appear in the group within 30 minutes. If they do not, manually resync the Data Aggregator data source. Select **Perform a Full Resynchronization**, and then click **Resync confirmation**.
- **groupBy**
  Instructs NetOps Portal to create an inventory menu item (under Inventory) for viewing all items of this component item type. The menu is named "*{* `itemTypeLabelPlural` *}*." This attribute also causes the component type show up in the Context Type drop-down list when setting a view context. When false, the components are listed in the Inventory, Device Components table with the type of "*{* `itemTypeLabel` *}*." The `groupBy` property does not create a group (see categorize).
  Can be updated: Yes
  Possible values: true, false
  Effect of updating: The menu item is created when true, or removed when false.

When does the update take effect: Allow for resynchronization to occur and allow up to 15 minutes to complete updates.

- **context**
  Makes each component item name a context hyperlink in the Inventory component views that, when clicked, navigate to the individual component context page.
  Can be updated: Yes/No
  Possible values: Plain text
  Effect of updating: Makes each component item name a context hyperlink in the Inventory component views.
  When does the update take effect: Allow for resynchronization to occur and allow up to 15 minutes to complete updates.

  NOTE
  You can create a component without the `ItemSyncDefinition` tag. In this case, a component discovered in Performance Management for this managed item type is not synced over to NetOps Portal (like CPU, or memory). Although the component is not synced over, you can still report on these managed items at the device level.

2. Set up a REST client with a connection to the data aggregator server.
3. Use the following format for the URL in the REST client:
   http://DA_host:8581/typecatalog/components

   NOTE
   To update or extend a vendor component, use the following format for the URL in the REST client:

   `http://`*`DA_host`*`:8581/typecatalog/components/`*`component_name`*

   component_name Specify the name of the existing component.
4. Select `POST` for the **HTTP Method**.

   NOTE
   To update an existing component, select `PUT` .
5. Select **application/xml** as the **Body Content-type**.
6. Add the customized XML within the "Body" text section.
7. Run the method.

Create a New Metric Family

A metric family describes the metrics available for collection for a component. Different vendors can provide information for all or part of the metrics in a metric family. For each metric, a metric family defines the behavior of the system. For example, a metric family can define whether to calculate baselines, percentiles, minimum, maximum, or projections. They also define how to calculate rollups. Only some fields in the metric family XML require changes. The `Indexes` , `Names` , and `Description` attributes are required. You should add one or more metric attributes to reflect the number and type of metrics in the metric family.

Follow these steps:

1. Customize the following XML:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- Auto-generated by the type catalog local manager.  -->
<DataModel xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance namespace="http://
im.ca.com/normalizer" xsi:noNamespaceSchemaLocation="IMDBCertificationFacet.xsd">
  <Author>Custom</Author>
  <Version>1.0</Version>
 <FacetType name="Internal_Metric_Family_Name"
descriptorClass="com.ca.im.core.datamodel.certs.NormalizedFacetDescriptorImpl">
```

```
<Documentation>Metric_Family_Description</Documentation>
    <FacetOf namespace="http://im.ca.com/core" name="Item" />
    <AttributeGroup name="AttributeGroup" external="true" list="true">
      <Documentation />
      <Attribute name="Indexes" type="ObjectID[]">
         <Documentation />
         <Polled>false</Polled>
         <Baseline>false</Baseline>
         <IsDbColumn>false</IsDbColumn>
         <Variance>false</Variance>
         <StandardDeviation>false</StandardDeviation>
         <Minimum>false</Minimum>
         <Maximum>false</Maximum>
         <WriteOnPoll>false</WriteOnPoll>
         <RollupStrategy />
         <AttributeDisplayName />
         <Percentile>0</Percentile>
       </Attribute>
       <Attribute name="Names" type="String">
 <Documentation>Element/Component_Name</Documentation>
         <Polled>false</Polled>
         <Baseline>false</Baseline>
         <IsDbColumn>false</IsDbColumn>
         <Variance>false</Variance>
         <StandardDeviation>false</StandardDeviation>
         <Minimum>false</Minimum>
         <Maximum>false</Maximum>
         <WriteOnPoll>false</WriteOnPoll>
         <RollupStrategy />
         <AttributeDisplayName />
         <Percentile>0</Percentile>
        </Attribute>
        <Attribute name="Description" type="String">
<Documentation>Element/Component_Description</Documentation>
          <Polled>false</Polled>
          <Baseline>false</Baseline>
          <IsDbColumn>false</IsDbColumn>
          <Variance>false</Variance>
          <StandardDeviation>false</StandardDeviation>
          <Minimum>false</Minimum>
          <Maximum>false</Maximum>
          <WriteOnPoll>false</WriteOnPoll>
          <RollupStrategy />
          <AttributeDisplayName />
          <Percentile>0</Percentile>
        </Attribute>
```

```
<Attribute name="Metric_Name" type="MIB_Metric_Type">
<Documentation>Metric_Description</Documentation>
          <Polled>true</Polled>
<Baseline>false</Baseline>
          <IsDbColumn>true</IsDbColumn>
 <Variance>false</Variance>
  <StandardDeviation>false</StandardDeviation>
<Minimum>false</Minimum>
          <Maximum>false</Maximum>
          <WriteOnPoll>false</WriteOnPoll>
  <RollupStrategy>Sum</RollupStrategy>
          <AttributeDisplayName />
  <Percentile>0</Percentile>
        </Attribute>
</AttributeGroup>
      <Attribute name="SourceFacetTypes" cached="true" list="true" persistent="true"
 type="QName">
        <Documentation />
      </Attribute>
<DisplayName>External_Metric_Family_Name</DisplayName>
      <Expressions>
        <ExpressionGroup destCert="{http://im.ca.com/core}Item">
          <Expression destAttr="Name">Names</Expression>
        </ExpressionGroup>
        <ExpressionGroup destCert="{http://im.ca.com/inventory}DeviceComponent">
          <Expression destAttr="IndexList">Indexes</Expression>
        </ExpressionGroup>
        </Expressions>
<TableName>DB_Table_Metric_Family_Name</TableName>
        <ComponentFacets>
<Facet>{http://im.ca.com/inventory}Component_Name</Facet>
        </ComponentFacets>
        <Protocol>IMDB</Protocol>
        <Normalized>true</Normalized>
       </FacetType>
     </DataModel>
```

– **Internal_Metric_Family_Name**
Specifies the metric family name. For each metric family, the name must be a unique name that identifies it internally within the system. Carefully select a name with a minimal possibility of naming conflicts with future similar metric families. For example, define a naming scheme that ensures that these metric family names are unique.

> **NOTE**
> This name is never exposed externally. To display a metric family name in the user interface, use the DisplayName element.

Can be updated: No

Possible values: Alphanumeric and underscore. Dot and dash are not permitted. The value must be unique across all metric families.

- **Metric_Family_Description** Specifies the external description for the metric family. To make these comments useful, describe why and when you added or changed the metric family.
  Possible values: Plain text
  Effect of updating: None
- **Element/Component_Name**
  The documentation is also displayed in tool tips when you hover the cursor over the attribute name.
- **Element/Component_Description** Displays the attribute description in the user interface. The documentation is also displayed in tool tips when you hover the cursor over the attribute name.
  Possible values: Plain text
  Effect of updating: Hovering the cursor over the attribute name shows the updated documentation.
- **Metric_Name**
  Specifies the unique, internal name. For metrics, this name is also used for naming the database column.

  NOTE
  This name is never exposed externally. To display an attribute name in the user interface, use the AttributeDisplayName element. To change the AttributeDisplayName, see Create or Extend Metric Families and update the metric family properties.

  Possible values: Alphanumeric and underscore.
  Effect of updating: For metrics, the values for this attribute are stored in a new database column corresponding to the updated name. The user loses the historical data that is collected for this metric (with the older name). The custom reports that report on this metric fails.
  When does the update take effect: Next poll
  Required Actions for updates to take effect: None
- **MIB_Metric_Type** Indicates the data type of this attribute. The most frequently used data types are `Int` , `Long` , `Double` , `String` , or `ObjectID` . The database stores metric attributes as a float. Therefore, these attributes must use a numeric type (we recommend a Double). Other types are used for item attributes.
  Possible values: `Boolean, Int, Long, Double` (floating-point), `BigInteger, String, DateTime, IPAddress, MACaddress, IPSubnet, OctetString` (hex representation), `ObjectID, ItemID, QName` (Qualified Name)

  NOTE
  The type names are case insensitive, for example, "boolean" is the same as Boolean.

  Effect of updating: For metrics, none. All metrics are stored in the database as a float. For item attributes, the device must be deleted and rediscovered.
  When does the update take effect: For metrics, next poll. For item attributes, on rediscover.
  Required Actions for updates to take effect: For metrics, none. For item attributes, delete the device and rediscover.
- **Metric_Description**
  Displays the attribute description in the user interface. The documentation is also displayed in tool tips when you hover the cursor over the attribute name.
  Possible values: Plain text
  Effect of updating: Hovering the cursor over the attribute name shows the updated documentation.
- **Baseline**
  Indicates whether to calculate a mean value for this attribute. If it is set to true, a corresponding `BaselineList` definition must be defined.

  NOTE
  The Baseline attribute requires that the StandardDeviation attribute is set to true.

  Possible values: true, false
  Effect of updating: Baseline values are calculated when true.
  When does the update take effect: Next poll

Required Actions for updates to take effect: None

- **Variance** This tag has restricted use in custom and extended metric families. The value of this tag must be 'false'.
- **Standard Deviation**
  Indicates whether to calculate the standard deviation of this attribute during the rollup. Creates a '`std` _' column in the database table. If `RollupStrategy` is defined, this attribute must also be defined.
  Possible values: true, false
  Effect of updating: True provides a calculation of, and a reporting field for, “Standard Deviation.”
  When does the update take effect: Next poll
  Required Actions for updates to take effect: None
- **Minimum**
  Indicates whether to calculate the minimum of this attribute during the rollup. Creates a '`min` _' column in the database table. If `RollupStrategy` is defined, this attribute must also be defined.
  Possible values: true, false
  Effect of updating: True provides a calculation of, and a reporting field for, “Minimum.”
  When does the update take effect: Next poll
  Required Actions for updates to take effect: None
- **Maximum**
  Indicates whether to calculate the maximum of this attribute during the rollup. Creates a 'max_' column in the database table. If `RollupStrategy` is defined, this attribute must also be defined.
  Possible values: true, false
  Effect of updating: True provides a calculation of, and a reporting field for, “Maximum.”
  When does the update take effect: Next poll
  Required Actions for updates to take effect: None
- **RollupStrategy**
  Specifies the operation that is performed every cycle during the rollup of the individually polled values. When Polled and `IsDbColumn` are set to true, this element is required.
  Possible values: Sum (a summation for counters), Avg (an average for gauges)
  Effect of updating: The specified strategy is used to perform rollup calculations.
  When does the update take effect: Next poll
  Required Actions for updates to take effect: None
- **Percentile**
  Indicates whether to calculate the 95th percentile of this attribute during the rollup. Creates a 'pct_' column in the database table. If `RollupStrategy` is defined, this attribute must also be defined.
  Possible values: 0, 95
  Effect of updating: A value of 95 provides a calculation of, and a reporting field for, “95th Percentile.” Zero specifies that no calculation is performed, and the reporting field is not available.
  When does the update take effect: Next poll
  Required Actions for updates to take effect: None
- **External_Metric_Family_Name** Specifies the metric family name that displays in the user interface.
  Possible values: Plain text

  > **IMPORTANT**
  > Ensure that the `DisplayName` property is unique to the metric family.

  Effect of updating: Change to the name in the administrator user interface.
  When does the update take effect: Immediately
  Required Actions for updates to take effect: Refresh the user interface.
- **DB_Table_Metric_Family_Name**
  Specifies the database table name that is used to store the metrics that the metric family collects.
  Possible values: Uppercase alphanumeric and underscore. The value must begin with a letter. The value must be unique across all metric families.
  Example: PROCESS_STATS
  Effect of updating: Poll data is stored in a new set of database tables.

WARNING
When you update the `TableName` property, the old poll data is lost. Old report views are broken.

When does the update take effect: Immediately. Before new views can be created, there is a delay of up to 5 minutes while DX NetOps Performance Management loads the new MIB files.
Required Actions for updates to take effect: Views must be recreated.

– **Component_Name**
Specifies a facet that is attached to the component item during component discovery.
Can be updated: Yes
Possible values: `QName` of the facet
Effect of updating: If the component facet is synchronized to NetOps Portal, the component is visible in NetOps Portal.
When does the update take effect: Rediscover
Required Actions for updates to take effect: Delete the device and rediscover.

2. Set up a REST client with a connection to the Data Aggregator server.
3. Use the following format for the URL in the REST client:
`http://DA_host:8581/typecatalog/metricfamilies`

NOTE
To update or extend a metric family, use the following format for the URL in the REST client:

`http://`*`DA_host`*`:8581/typecatalog/metricfamilies/`*`metric_family_name`*

metric_family_name Specify the name of the existing metric family.

4. Select `POST` as the **HTTP Method**.

NOTE
To update an existing component, select `PUT` .

5. Select **application/xml** as the **Body Content-type**.
6. Add the customized XML within the "Body" text section.
7. Run the method.

Create a New Vendor Certification

A vendor certification provides a link between the vendor MIB and the metrics in the metric family. A vendor certification specifies which MIB variables are used for the relevant metrics. A vendor certification also specifies the formulas or expressions that are used to calculate the relevant values. Some variables directly match a metric (for example, rate, errors, discards, and volume). Others variables require calculations (for example, utilization or free disk space).

NOTE
If scalar OIDs need to be polled in the vendor certification, the definition of attributes changes slightly. If you are grouping scalar OIDs for better organization within an `AttributeGroup` , set the list variable to false (list=”false”). Or, omit the variable in the `AttributeGroup` tag definition. `list=″true″` in a vendor certification appends the index to the OID when polling. When you define the attribute source, append a ‘.0’ to the end of the OID. You can also have scalar OIDs defined outside of an `AttributeGroup` .

Follow these steps:

1. Customize the following XML:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- Auto-generated by the type catalog local manager.-->
<DataModel namespace="http://im.ca.com/certifications/snmp"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:noNamespaceSchemaLocation="SNMPCertificationFacet.xsd">
  <Author>Custom</Author>
```

```
<Version>Version</Version>
  <FacetType name="Internal_Vendor_Cert_Name"
descriptorClass="com.ca.im.core.datamodel.certs.CertificationFacetDescriptorImpl">
    <Documentation>Vendor_Cert_Description</Documentation>
    <FacetOf namespace="http://im.ca.com/core" name="Item" />
    <AttributeGroup name="AttributeGroup" external="true" list="true">
      <Documentation />
      <Attribute name="INDEX" type="ObjectID">
      <Documentation />
    <Source>Index_OID</Source>
      <IsIndex>true</IsIndex>
      <IsKey>false</IsKey>
      <NeedsDelta>false</NeedsDelta>
    </Attribute>
    <Attribute name="Internal_Metric_Name" type="Data_Type_from_MIB">
      <Documentation />
<Source>MIB_Variable_OID</Source>
      <IsIndex>false</IsIndex>
      <IsKey>true</IsKey>
      <NeedsDelta>true</NeedsDelta>
    </Attribute>
</AttributeGroup>
  <Protocol>SNMP</Protocol>
<DisplayName>Vendor_Cert_Name</DisplayName>
  <Expressions>
 <ExpressionGroup destCert="{http://im.ca.com/normalizer}Metric_Family_Name"
 name="Metric_Family_Name">
      <Expression destAttr="Indexes">INDEX</Expression>
<Expression destAttr="Names">"Element_Name" + INDEX</Expression>
<Expression destAttr="Metric_Name">Internal_Metric_Name_or_MVEL_Expression</
Expression>
    </ExpressionGroup>
  </Expressions>
 <MIB>MIB_Name</MIB>
 </FacetType>
</DataModel>
```

- **Version**
  Specifies the vendor certificate version.
  Example: 1.0
- **Internal_Vendor_Cert_Name**
  Uniquely identifies a vendor certification.

  IMPORTANT
  Conform to "`<MibName><TableName>Mib` ."

  Can be updated: No
  Possible values: Alphanumeric and underscore. Dot and dash are not permitted.
- **Vendor_Cert_Description**
  Describes what is certified with the vendor certification.

TIP
Include the details about the vendor, MIB name, and table name.

Effect of updating: None

– **Index_OID** Specifies the `ObjectID` of the attribute.

TIP
Set to the fully qualified MIB variable OID that the OBJECT-TYPE defines.

Possible values: Dot-separated numbers (for example, 1.3.6.1.4.1…)
Effect of updating: Data is polled from the specified OID.
When does the update take effect: Next poll

NOTE
By default, the `Source` attribute specifies an OID to poll from the device. This default behavior is defined with `src='polled'` . You can set `src='mvel'` to process an MVEL expression using any polled attribute instead of an OID. For example, you could use the `src='mvel'` parameter to combine two 32-bit OIDs into a single 64-bit counter. You could also use the `src='mvel'` parameter to poll a counter that is not stored as a numeric type.

Example: The following example uses the `src='mvel'` parameter to combine two 32-bit OIDs into a single 64-bit counter:

```
<Attribute name="memberbitsout" type="Long">
     <NeedsDelta>true</NeedsDelta>
     <Source src='mvel'>snmpCounter64(memberbitsoutHi32,memberbitsoutLo32)</Source>
</Attribute>
```

NOTE
The `src='mvel'` parameter can only be used during polling. This parameter cannot be used during the discovery phases. For example, this parameter cannot be used as part of the Name or Description.

– **Internal_Metric_Name**
Specifies the attribute name.

TIP
Set to the MIB variable name, which the OBJECT-TYPE clause defines in the ASN.1 file.

Possible values: Alphanumeric and underscore. Dot and dash are not permitted.
Effect of updating: Update any expressions that reference this attribute.

– **Data_Type_from_MIB** Specifies the data type of the attribute.

TIP
Use the attribute type that best matches the variable type that the SYNTAX clause defines in the ASN.1 file.

Possible values: `Boolean, Int, Long, Double, BigInteger, String, DateTime, IPAddress, MACaddress, IPSubnet, OctetString, ObjectID`
Effect of updating: Polled SNMP data is converted to this type.
When does the update take effect: Next poll

– **MIB_Variable_OID** Specifies the `ObjectID` of the attribute.

TIP
Set to the fully qualified MIB variable OID that the OBJECT-TYPE defines.

Possible values: Dot-separated numbers (for example, 1.3.6.1.4.1…)
Effect of updating: Data is polled from the specified OID.
When does the update take effect: Next poll

NOTE
By default, the `Source` attribute specifies an OID to poll from the device. This default behavior is defined with `src='polled'` . You can set `src='mvel'` to process an MVEL expression using any polled attribute instead of an OID. For example, you could use the `src='mvel'` parameter to combine two 32-

bit OIDs into a single 64-bit counter. You could also use the `src='mvel'` parameter to poll a counter that is not stored as a numeric type.

Example: The following example uses the `src='mvel'` parameter to combine two 32-bit OIDs into a single 64-bit counter:

```
<Attribute name="memberbitsout" type="Long">
     <NeedsDelta>true</NeedsDelta>
     <Source src='mvel'>snmpCounter64(memberbitsoutHi32,memberbitsoutLo32)</Source>
</Attribute>
```

NOTE
The `src='mvel'` parameter can only be used during polling. This parameter cannot be used during the discovery phases. For example, this parameter cannot be used as part of the Name or Description.

- **Vendor_Cert_Name** Specifies the name of the vendor certification as it displays in NetOps Portal.

  TIP
  Start with the vendor name and include the MIB and functionality information.

  Effect of updating: A change to the name in the Administrator user interface.
  When does the update take effect: Immediately
  Required Actions for updates to take effect: Refresh the user interface.

  WARNING
  Ensure that the `DisplayName` property is unique to the vendor certification.
- **Metric_Family_Name**
  Specifies the metric family that contains the `destAttrs` to populate.
  Possible values: Any valid metric family
  Effect of updating: Changes the permissible expression `destAttr` .
- **Element_Name** Specifies to use the vendor certification attributes to collect configuration data. This configuration data helps define the MVEL expression to provide the value for the Names metric family attribute.

  TIP
  Include as much information as necessary to identify an instance uniquely.

  Possible values: String MVEL expression using available Attributes
  Effect of updating: Component name change
  When does the update take effect: After component rediscovery
  Required Actions for updates to take effect: Update the metric family or change the vendor certification priority.
- **Metric_Name**
  Specifies the metric name as defined in the metric family.
- **Internal_Metric_Name_or_MVEL_Expression**
  Specifies the internal metric name as defined in the attribute section. Or, specifies the relevant formula that is used to calculate the metric from the MIB variable represented by the internal metric name.
- **>MIB_Name**
  Specifies the name of the MIB, which the DEFINITIONS clause defines in the ASN.1 file.

  IMPORTANT
  Conform to "<MibName>”

  Effect of updating: Change to the “SNMP MIB Name” column in the Vendor Certification tab of the Administrator user interface.
  When does the update take effect: Immediately
  Required Actions for updates to take effect: Refresh the user interface.

2. Set up a REST client with a connection to the Data Aggregator server.
3. Use the following format for the URL in the REST client:
   `http://DA_host:8581/typecatalog/certifications/snmp`

   NOTE
   To update or extend a vendor certification, use the following format for the URL in the REST client:

`http://DA_host:8581/typecatalog/certifications/snmp/cert_name`

cert_name Specify the name of the existing certification.

4. Select `POST` for the **HTTP Method**.

   NOTE
   To update or extend a vendor certification, select `PUT` .
5. Select **application/xml** as the **Body Content-type**.
6. Add the customized XML within the "Body" text section.
7. Run the method.

Add a New Metric to an Existing Metric Family

Add a new BitsRateCustom metric that is based on collected MIB objects to the Interface metric family.

Update the Metric Family with a New Metric

Follow these steps:

1. Make the following GET REST call to export existing extensions for the Interface metric family:

   ```
   http://da_hostname:8581/typecatalog/metricfamilies/extension/NormalizedPortInfo
   ```

   The current extensions are returned. If you have not defined any extensions, the following XML is returned:

   ```
   <DataModel xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" namespace="http://im.ca.com/normalizer"
    xsi:noNamespaceSchemaLocation="IMDBCertificationFacet.xsd">
       <Author>CA</Author>
       <Version>2.03</Version>
       <FacetType name="NormalizedPortInfo"
    descriptorClass="com.ca.im.core.datamodel.certs.NormalizedFacetDescriptorImpl">
       <FacetOf namespace="http://im.ca.com/core" name="Item"/>
       </FacetType>
   </DataModel>
   ```

2. Define the BitsRateCustom attribute in the extension XML file.

   ```
   <DataModel xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" namespace="http://
   im.ca.com/normalizer" xsi:noNamespaceSchemaLocation="IMDBCertificationFacet.xsd">
       <Author>CA</Author>
       <Version>2.03</Version>
       <FacetType name="NormalizedPortInfo"
    descriptorClass="com.ca.im.core.datamodel.certs.NormalizedFacetDescriptorImpl">
       <FacetOf namespace="http://im.ca.com/core" name="Item"/>
       <AttributeGroup name="PortInfoPollable" list="true" external="true">
           <Attribute name="BitsRateCustom" type="Double">
               <Documentation/>
               <IsDbColumn>true</IsDbColumn>
               <Baseline>false</Baseline>
               <Minimum>true</Minimum>
               <Maximum>true</Maximum>
               <RollupStrategy>Sum</RollupStrategy>
               <StandardDeviation>false</StandardDeviation>
   ```

```
            <Variance>false</Variance>
            <Percentile>95</Percentile>
            <Percentile2>0</Percentile2>
            <Percentile3>0</Percentile3>
            <ProjectionPercentile>0</ProjectionPercentile>
            <Polled>true</Polled>
            <Units>BitsPerSecond</Units>
         </Attribute>
      </AttributeGroup>
      </FacetType>
   </DataModel>
```

3. Make the following PUT REST call to import the extensions for the Interface metric family:
   `http://da_hostname:8581/typecatalog/metricfamilies/extension/NormalizedPortInfo`

Update the Vendor Certification with the New Metric

Follow these steps:

1. Make the following GET REST call to export existing extensions for the vendor certification:

   TIP
   You can retrieve the name of the vendor certification in the user interface. Go to the Vendor Certifications page. Click the downward arrow on any column, hover over **Columns**, and click **Internal Name**.

   http://da_hostname:8581/typecatalog/certifications/snmp/extension/cert_name

2. Add an expression for calculating the new BitsRateCustom attributes.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Auto-generated by the type catalog local manager.-->
<DataModel namespace="http://im.ca.com/certifications/snmp"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:noNamespaceSchemaLocation="SNMPCertificationFacet.xsd">
   <Author>CA</Author>
   <Version>2.02</Version>
   <FacetType name="IfTableMib"
 descriptorClass="com.ca.im.core.datamodel.certs.CertificationFacetDescriptorImpl">
   <FacetOf namespace="http://im.ca.com/core" name="Item" />
        <Expressions>
            <ExpressionGroup destCert="{http://im.ca.com/
normalizer}NormalizedPortInfo" name="PortNRMDS">
            <Expression destAttr="BitsRateCustom">((ifInOctets+ifOutOctets)*8)/
_rspDuration</Expression>
            </ExpressionGroup>
        </Expressions>
   </FacetType>
</DataModel>
```

3. Make the following PUT REST call to import the extensions for the Interface metric family:
   `http://da-hostname:8581/typecatalog/certifications/snmp/extension/cert_name`

After you import the vendor certification, the DX NetOps Performance Management resumes polling and uses the new extended vendor certification.

NOTE
The BitsRateCustom metric becomes available 10 minutes after you import the extension.

Validate in NetOps Portal

After you update the vendor certification with the new metric, verify that the BitsRateCustom metric appears in the Metric Families pane. Data that is polled for the BitsRateCustom metric appears in views after 3 poll cycles (15 minutes).

Follow these steps:

1. Click **Admin**, **Data Aggregator**.
   The Monitored Devices screen appears.
2. Click **Monitoring Configuration**, **Metric Families**.
3. Verify that the BitsRateCustom metric is added as needed.

TIP
You can report data on the metric that you added by creating a view for the metric.

Add a New Filter to a Vendor Certification

This self-certification workflow is an example of a real use case that you may encounter. Use this workflow as a model to help you with self-certification.

In this example, you add a filter to the IfTableMib vendor certification to prevent items with ifType 53 (propVirtual) from being discovered.

After the filter is applied, you can expect the following results:

- In NetOps Portal, if you go to **Administration**, **Monitored Devices**, **Polled Metric Families**, the interface appears as **Not Available** or **Not Present** and **Not Polled**.
- In NetOps Portal, if you go to **Administration**, **Monitored Devices**, **Filter Report**, the filter report is unaffected. The report shows only the filters from the monitoring profile. The report excludes the filter from the vendor certification.
- The interface is still available through REST on the Data Aggregator.
- The interface is unavailable through OpenAPI.

After the next sync, you can expect the following results in NetOps Portal:

- If you go to **Inventory**, **Interfaces**, the interface is removed.
- The interface is removed from all dashboards and context pages.

Follow these steps:

1. Make a GET REST call to export existing extensions for the vendor certification:
   `http://da-hostname:8581/typecatalog/certifications/snmp/extension/Vendor_Certification_Name`

   For example:
   `http://da-hostname:8581/typecatalog/certifications/snmp/extension/IfTableMib`

   TIP
   To retrieve the name of the vendor certification in the UI, go to the Vendor Certification page. Click the downward arrow on any column, hover over **Columns**, and click **Internal Name**.
2. Find the metric definition in the XML file.

```
<ExpressionGroup name="PortNRMDS" destCert="{http://im.ca.com/
normalizer}NormalizedPortInfo">
```

3. Enter the metric filtering expression inside a Filter tag:
   ```
   <Filter>(ifType!=24) &amp;&amp; (ifType!=1) &amp;&amp; (ifType!=53)</Filter>
   ```
4. Make a PUT REST call to import the new XML file:
   ```
   http://da-hostname:8581/typecatalog/certifications/snmp/extension/IfTableMib
   ```
   After you import the vendor certification, the Data Aggregator resumes polling using the new extended vendor certification.
5. Click **Update Metric Family** on the Metric Families page.
   The vendor certification no longer monitors interfaces of ifType 53.
6. To verify that the filter is added, click **Admin**, **Monitored Devices**.
7. >Click **Polled Metric Families**.
   In the list of Components, the Status of the filtered interfaces is changed to **Not Available** or **Not Present**, and the **SNMP Poll Rate** is changed to **Not Polled**.

Example

The following example shows how to define the metric filtering expression in the IfTableMib vendor certification:

```
<?xml version="1.0" encoding="UTF-8"?>

<!--Auto-generated by the type catalog local manager.-->

<DataModel namespace="http://im.ca.com/certifications/snmp" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance"
 xsi:noNamespaceSchemaLocation="SNMPCertificationFacet.xsd">

  <Author>CA</Author>

  <Version>2.02</Version>

  <FacetType name="IfTableMib"
 descriptorClass="com.ca.im.core.datamodel.certs.CertificationFacetDescriptorImpl">

    <FacetOf namespace="http://im.ca.com/core" name="Item" />

    <Expressions>

      <ExpressionGroup destCert="{http://im.ca.com/normalizer}NormalizedPortInfo"
 name="PortNRMDS">

        <Filter>(ifType!=24) &amp;&amp; (ifType!=1) &amp;&amp; (ifType!=53)</Filter>

      </ExpressionGroup>

     </Expressions>

  </FacetType>
```

```
</DataModel>
```

Change the Calculation Method for an Existing Metric

You want to change how the Discards metric is calculated for the IfTableMIB vendor certification. Discards is an existing out-of-the-box metric.

1. Make a GET REST call to export the full IfTableMIB vendor certification.

   ```
   http://da_hostname:8581/typecatalog/certifications/snmp/IfTableMIB
   ```

2. Make a GET REST call to export existing extensions for the IfTableMIB vendor certification.
3. Find the metric expression that you want to change.

   ```
   <Expression destAttr="Discards"></Expression>
   ```

4. Enter the new calculation in the vendor certification extension file.

   ```
   <Expression destAttr="Discards">(ifInDiscards+ifOutDiscards+ifInErrors+ifOutErrors)</
   Expression>
   ```

5. Import the new extended vendor certification.

   ```
   http://da-hostname:8581/typecatalog/certifications/snmp/extension/IfTableMIB
   ```

   The DA resumes polling using the new extended IfTableMIB vendor certification. The new calculation for the Discards metric is applied.
6. After 3 poll cycles (15 minutes), verify the new calculation in a CAPC view.
   a. Create a view using the new extended IfTableMIB vendor certification.
   b. Verify in CA NetOps Portal that the new calculation for Discards is used.

Example

The following example shows how to define the new calculation in the vendor certification extension XML file:

```
<?xml version="1.0" encoding="UTF-8"?>

<DataModel xsi:noNamespaceSchemaLocation="SNMPCertificationFacet.xsd" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance" namespace="http://im.ca.com/certifications/snmp">

  <Author>CA</Author>

  <Version>100.0</Version>
```

```
  <FacetType
 descriptorClass="com.ca.im.core.datamodel.certs.CertificationFacetDescriptorImpl"
 name="IfTableMib">

    <Documentation>Supports interfaces new metrics</Documentation>

    <FacetOf namespace="http://im.ca.com/core" name="Item"/>

    <Expressions>

      <ExpressionGroup name="PortNRMDS" destCert="{http://im.ca.com/
normalizer}NormalizedPortInfo">

        <Expression destAttr="Discards">(ifInDiscards+ifOutDiscards+ifInErrors
+ifOutErrors)</Expression>


      </ExpressionGroup>

    </Expressions>

  </FacetType>

</DataModel>
```

Deploy a New Metric Family or Vendor Certification on the Production System

You have created a new metric family or vendor certification on your test environment. You want to deploy the new metric family or vendor certification to your production system. Do not experiment on the production system. Always develop metric families and vendor certifications on the test environment, and then deploy them to the production system.

1. Export the metric family or vendor certification from the test environment type catalog.
   Export a Metric Family

   ```
   GET http://da_testhostname:8581/typecatalog/metricfamilies/mf_name
   ```

   Export a Vendor Certification

   ```
   GET http://da_testhostname:8581/typecatalog/certifications/snmp/vc_name
   ```

2. Import the XML file that contains the metric family or vendor certification to the production system.
   Import a Metric Family

   ```
   POST http://da_hostname:8581/typecatalog/metricfamilies
   ```

   Import a Vendor Certification

   ```
   POST http://da_hostname:8581/typecatalog/certifications/snmp
   ```

Manage Missing Vendor Certifications

If you see views with no data, or views that have data for some fields and none for others, you might have missing vendor certifications. After discovery completes, you can view a list of metric families with missing vendor certifications and the affected devices.Follow these steps:

1. Hover over **Administration**, and click **Monitored Items Management: Monitored Devices**.
2. In the **Monitoring Configuration** section, click **Missing Vendor Certifications**. A list of metric families with missing vendor certifications appears.
3. Click one of the metric families. A list of the affected devices appears.
4. Go to the Technology Certification Portal to find the missing certification or to request a new one.

SNMP Profiles

Simple Network Management Protocol (SNMP) profiles contain the information necessary to enable secure queries of device management information bases (MIBs). SNMP profiles provide SNMP parameters to data sources and ensure data security. DX NetOps Performance Management supports SNMPv1, SNMPv2c, and SNMPv3. DX NetOps Performance Management encrypts community strings and credentials.

DX NetOps Performance Management uses SNMP profiles during inventory discovery to determine what credentials to use when accessing a device. Each profile is ranked for device access. During discovery, each profile is tried for device access. The profile with the highest rank that can access a device is used. If a device that uses an SNMPv1/SNMPv2c profile responds to SNMPv1 and SNMPv2c, DX NetOps Performance Management uses SNMPv2c.

Polling devices with SNMPv3 adds an extra load of about 30 percent to the CPU of Data Collectors.

> **NOTE**
> To limit the SNMP profiles that are used during discovery, use a specific list of assigned SNMP profiles. For more information, see Discovery Profiles.

CA Application Delivery Analysis, Network Flow Analysis, and CA Unified Communications Monitor use SNMP profiles to query the MIBs of managed items for performance information. When you register one of these data sources, any profiles that were created in that data source are added to DX NetOps Performance Management. Naming conflicts are resolved automatically. Any changes to SNMP profile in NetOps Portal are propagated to these data sources during synchronization.

Users with the Administer SNMP Profiles role right can create, edit, and delete SNMP profiles. SNMP profiles are specific to tenants. The Default Tenant Administrator sees a list of SNMP profiles that are associated with the Default Tenant. In multi-tenant environments, each tenant administrator sees only the SNMP profiles for that tenant.

To view the list of SNMP profiles, hover over **Administration**, **Configuration Settings**, and then click **SNMP Profiles**. The list includes high-level information about the contents of each profile.

Create an SNMP Profile

To enable the system to query devices through SNMP, define SNMP profiles with communication credentials.

This procedure requires the Administer SNMP Profiles role rights.

Follow these steps:

1. Hover over **Administration**, **Configuration Settings**, and then click **SNMP Profiles**.
2. Click **New**.

3. Complete the fields, and change any default settings. Some fields apply only to SNMPv3.
   - **Profile Name**
     SNMP profile names must be unique, cannot be duplicated across SNMP versions, and are not case-sensitive.
   - **SNMP Version**
     Specifies whether the profile uses SNMPv1/v2c or SNMPv3.
   - **Port**
     The port that is used to make SNMP connections to devices associated with this profile.
     Default: 161

     NOTE
     This port can also be used to send SNMP traps to trap receivers associated with this profile through notifications. In this scenario, use 162 by default. For more information, see Configure Notifications.
   - **User Name**
     (SNMPv3 Only) Identifies the user for the profile, whose secret keys were used to authenticate and encrypt the SNMPv3 packets.
   - **Context Name**
     (SNMPv3 Only) Identifies the collection of management information that is accessible by an SNMP entity. An octet string that is necessary for providing end-to-end identification and for retrieving data from an SNMPv3 agent.

     NOTE
     The Data Aggregator does not use Context Name on the SNMPv3 profiles to communicate with the device.
   - **Community Name**
     (SNMPv1/v2c Only) Defines a secure string that lets the data source query the MIB of the associated device. The community that you supply must provide read access to the device MIB.

     NOTE
     In the default SNMP profile, the community is 'public'.
   - **Authentication Protocol**
     (SNMPv3 Only) Specifies the authentication protocol to use when contacting devices associated with this profile. The following algorithms for authenticating SNMPv3 packets are supported:
     - None
     - MD5 (Message Digest 5)
     - SHA (Secure Hash Algorithm)
   - **Authentication Password**
     (SNMPv3 Only) Specifies the password for authentication using SNMPv3 and the selected authentication protocol.
   - **Privacy Protocol**
     (SNMPv3 Only) Specifies the encryption protocol to use for data flows sent to any devices or servers:

     NOTE
     The privacy protocol option is enabled when authentication is enabled for the profile.

     - None
     - DES
     - AES 128
     - Triple DES
     - AES 256 with 3DES key
   - **Privacy Password**(SNMPv3 Only)
     Defines the password that is used when exchanging encryption keys.
   - **Use by default for new devices**

Specifies whether CA Application Delivery Analysis, Network Flow Analysis, and CA Unified Communications Monitor use this profile to contact any new items. To stop these data sources from using this SNMP profile for discovery, disable this parameter.

– **Use for SNMP SET**
The profile provides write credentials on the discovered devices. Profiles with this property *do not* participate in device discovery. Associate this profile with devices which require SET authorization, for example, to configure RTT tests.
For more information, see Configure Round Trip Time (RTT) Tests.

4. Click **Save**.
The SNMP profile is added to the system and used for discovery and polling.
NetOps Portal automatically performs a global synchronization to send the profile information to registered data sources.

Change SNMP Profile Order

To determine the selection order of SNMP profiles for discovery and polling, change the priority order of the SNMP profiles. The **Order** parameter determines which profile DX NetOps Performance Management uses for polling when the device responds to multiple profiles.

Changes to the order do not affect existing polled devices. DX NetOps Performance Management continues to use the associated SNMP profile to poll those devices.

The new order takes effect in the following situations:

- A new device is discovered.
- An existing device becomes unreachable through SNMP for at least two poll cycles.
- The SNMP profile for a device is deleted.

Administrator users can modify the priority order of SNMP profiles.

Follow these steps:

1. Hover over **Administration**, **Configuration Settings**, and then click **SNMP Profiles**.
2. Click and drag or click **Move Up** and **Move Down** to change the order.
DX NetOps Performance Management uses the new order for unreachable devices.

SNMP Profile Changes

When the SNMP credentials on a polled device change, add the new SNMP profile information to DX NetOps Performance Management. When the device becomes unreachable with the deprecated SNMP profile for two poll cycles, DX NetOps Performance Management attempts to contact the device with other profiles. When DX NetOps Performance Management successfully contacts the device with an SNMP profile, that profile is assigned to the device for future polling.

NOTE
To see the SNMP profile that DX NetOps Performance Management uses to poll the device, go to the administration page for the device.
For more information, see Manage Devices.

Modify the Timeout and Retries Parameters

You can modify the timeout and retries parameters for each SNMP profile on your system using a REST client. If SNMP requests go across a WAN or across a slow network connection, they might time out. The timeouts can cause missing polled data or device discovery failure.

- **Timeout**

The amount of time a device is given to respond to an SNMP request per tryDefault: 3000 milliseconds

- **Retries**
  The number of times an SNMP query is reissued before it times out
  Default: 2 retries

For example, by default, an SNMP request is given the following amount of time:

- 3 seconds x (first attempt + 2 retries)= 3 seconds x 3 tries = 9 seconds to respond before it times outs

> **WARNING**
> Modifying these parameters without careful consideration can result in unintended consequences. For example, modifications could result in resource starvation (CPU/Memory) and unnecessary traffic on the Data Collector. Modify these parameters only if you have a basic understanding of SNMP communication.

Follow these steps:

1. Set up a REST client with a connection to the Data Aggregator server.
2. Specify the following URL:
   `http://da_hostname:8581/rest/profiles/profile_item_id`
3. PUT the XML for modifying the parameters:

```
<?xml version="1.0" encoding="UTF-8"?>
<CommunicationProfile version="1.0.0">
    <CommunicationFailurePolicy version="1.0.0">
        <Timeout>3000</Timeout>
        <Retries>2</Retries>
    </CommunicationFailurePolicy>
</CommunicationProfile>
```

Show Secure SNMP Data in Clear Text

By default, secure data is encrypted in the Add and Edit SNMP Profiles pages. To enable an administrator to troubleshoot issues with SNMP polling, allow that administrator to view secure data in clear text. By default, this role right is not assigned to any roles. Only the predefined Administrator role can have this role right. Only Administrator users can modify role rights.

By default, the predefined Administrator role is assigned only to the global administrator. To allow another user to view secure SNMP data, assign the Administrator role to another user account.

Follow these steps:

1. Hover over **Administration**, **User Settings**, and then click **Roles**.
2. Select the **Administrator** role, and click **Edit**.
3. Select **NetOps Portal**, and then click **Edit**.
4. Add the **SNMP Clear Text** role right to the **Selected Rights** list, and then click **OK**.
5. Click **Save**.
   Users with the Administrator role can now view secure SNMP data in clear text.

Discover Logical Devices Through SNMP Context

You can discover logical components on devices where you need to use alternate Simple Network Management Protocol (SNMP) contexts by adding Check Point Virtual Firewalls to managed devices.

Add Check Point Virtual Firewalls to Managed Devices

A new vendor certification and metric family are defined for Check Point Virtual Firewalls.

Follow these steps:

1. Use SNMPv3 profiles to discover the physical parent devices.
2. Attach the new Context System metric family to each device.
3. Wait for discovery to complete.
   The Check Point Virtual Firewalls are automatically discovered.

   NOTE
   All the Check Point Virtual Firewall items have the same IP address as the parent device. However, each Check Point Virtual Firewall item has its own context name. All the polling data for the Check Point Virtual Firewall item (CPU, Memory, Interfaces, and so on) appear under the context name.

IP Domains

IP domains resolve IP address conflicts by separating monitored devices from different networks. For example, you monitor two networks and each network includes a separate device with the same IP address.

As a service provider, use IP domains to monitor multiple discrete networks that belong to different customers. Each customer, or tenant, contains one or more IP domains. The data collectors associate managed items and data with the assigned IP domain. Customer user accounts see only the relevant IP domains. Service provider administrators see the data from all IP domains.

Administrators and Designers can create custom dashboards to monitor activity on a specific domain or group of domains.

To view IP domains, go to **Administration**, **IP Domains**.

In this article:

IP Domains and Other Data Sources

Network Flow Analysis, CA Application Delivery Analysis, and CA Unified Communications Monitor data sources can use IP domains. To apply domains to the data sources, register the data source with NetOps Portal. When you create IP domains in NetOps Portal, the domain identifiers become visible in the registered data sources after synchronization.

You can associate the following managed item types with an IP domain:

- Devices
- Interfaces and interface addresses
- Networks
- VoIP Locations

Data sources report a domain identifier that associates monitored items with a specific IP domain. Items that are not assigned to a custom domain in the data source are associated with the default domain.

Add or Edit an IP Domain

IP domains are required for monitoring multiple tenants or environments with overlapping IP addresses. Create custom IP domains so the data sources can associate monitored items with specific domains and tenants.

The Default Domain is automatically created. This domain includes any items that are not assigned to a custom domain in the data source.

The procedure requires the Administer IP Domains role right.

Follow these steps:

1. Select **Administration**, **IP Domains**.
2. Click **New**.
3. Specify a name and description.
4. (Optional) Configure a device name alias or interface description override. Click **Browse**, select the file, and then click **Open**.
   Use a CSV or TXT file with the following format:
   ```
   IP_Address,name,description,alias/alternate_description
   ```
   Use the primary IP address of the associated devices. To find the primary IP address, look at the Address column of the Inventory Devices list.
   You can use the same alternate interface descriptions for more than one interface.
   Example:
   ```
   172.24.36.107,ethernet_7,vmxnet3 Ethernet Adapter,Connection to Dallas
   ```

   NOTE
   To change or remove an alternate description or device name alias, import a new file. To remove the alias or alternate description, leave the final column blank. When you remove the alias or alternate description, the original name or description reappears. If you change or remove alternate descriptions or device name aliases from a file using a spreadsheet program, include a column heading for that column.

   Updates to device name aliases and interface descriptions can take up to 5 minutes to appear.

   TIP
   The preferred method to apply device name aliases and alternate interface descriptions is through REST. For more information, see Set the Alias Name for a Device and Set Interface Name Aliases.

   To update multiple device aliases, use the script that is included with DX NetOps Performance Management. For more information about this script, see Set Alias Names For Multiple Monitored Devices.
5. (Optional) To assign a primary and secondary DNS address for the domain in Network Flow Analysis, select **DNS Settings**, and then specify the required values.

   NOTE
   Only Network Flow Analysis uses the DNS settings. DX NetOps Performance Management does not directly use the DNS settings.
6. Click **Save**.
   The IP domain appears in the list immediately, and the changes apply to the data sources in the next synchronization. If you edit an IP domain, historical data for monitored items remains unchanged.

Assign Items to an IP Domain

To add items to an IP domain, associate a data collector with the IP domain and create a discovery profile that uses the IP domain. Each data collector associates managed items with a single IP domain. To enable multi-tenant deployments, assign an IP domain to each data collector. Before you install a data collector, create the tenants and IP domains that you require. You can associate multiple data collectors with a single IP domain.

NOTE
You cannot move items from one IP domain to another.

To assign items from other data sources to an IP Domain, see the documentation for that data source or see the following pages:

- Assign CA Network Flow Analysis Items to an IP Domain
- Assign CA Application Delivery Analysis Items to an IP Domain
- Assign CA Unified Communications Monitor Items to an IP Domain

TIP
DX NetOps Spectrum determines which models are synchronized with DX NetOps Performance Management using IP domains.
For more information, see the DX NetOps Spectrum documentation.

This procedure requires the Administer Data Sources role right, and the Administrator product privilege for the data aggregator data source.

Follow these steps:

1. Select **Administration**, and then click the data aggregator data source.
2. Expand **System Status**, and then click **Data Collectors**.
3. Select an IP domain and a tenant for each data collector in the list, and then click **Assign**.
   You cannot assign a data collector that is monitoring devices other than itself to a different IP domain.
4. Create a Discovery profile that is associated with the IP domain and run discovery.
   For more information, see Discovery.

Delete an IP Domain

When you delete an IP domain, all monitored items in that IP domain are deleted, including historical data. The items are also deleted from registered data sources that use IP domains, such as Network Flow Analysis, CA Application Delivery Analysis, and CA Unified Communications Monitor. DX NetOps Spectrum devices in the IP domain are removed from DX NetOps Performance Management, but DX NetOps Spectrum continues to monitor the devices.

When you delete an IP domain, the associated data collectors and discovery profiles become inactive. Assign another IP domain to continue polling or discovery.

IMPORTANT
To retain associated data, do not delete an IP domain. You cannot recover deleted data.

Assign Network Flow Analysis Items to an IP Domain

Interfaces and CVIs inherit their initial tenant-domain setting from the parent router and Harvester. The setting is inherited when the parent Harvester is added and the router and interfaces first become active. If the Harvester is not associated with a custom domain, the routers and interfaces are assigned to the Default Domain as they become active.

You can edit the settings for interfaces and CVIs to associate them with any tenant and domain at any time. The setting does not have to match the parent router or Harvester.

Changing this setting can affect which operators have access to interface data. The setting does not affect which Simple Network Management Protocol (SNMP) profiles you are using for polling. The router tenant determines the set of SNMP profiles for polling.

NOTE
You can also change the tenant-domain setting for Harvesters and routers.

Follow these steps:

1. Open the Active Interfaces page:
   a. Select **Administration** from the Network Flow Analysis console menu.
   b. Select Interfaces: Physical & Virtual from the Administration menu.
2. Select one or more interfaces that you want to associate with a tenant and domain.
3. • To search for parent routers, interfaces, or CVIs, enter all or part of a router IP address, a router or interface name, or an interface description in the **Search** field, and then click **Search**. Expand the router details.
   • To navigate to an interface or CVI manually, go to the page that contains the parent router, and click the arrow next to the router name. The router details expand to show the interfaces and CVIs.
4. Click **Edit**.

The Domain selection list is included in the dialog only if multiple domains exist.
5. Select a tenant/domain option from the Domain list.
6. Click **Save**.
The changes are shown on the Active Interfaces page.

Assign Application Delivery Analysis Items to an IP Domain

CA Application Delivery Analysis can observe duplicate Internet Protocol (IP) traffic, which occurs in a managed service provider environment. The provider can host an application on a single server for multiple customers whose environments contain overlapping client IP addresses.

You enable CA Application Delivery Analysis to identify separate IP traffic during data collection setup. As you verify and modify data collection parameters, assign the same IP domain to the appropriate:

- Monitor feeds
- Client networks
- Servers or server subnets

With the same IP domain assignments for these feeds, CA Application Delivery Analysis reports on the application traffic between a client and a server by domain.

Applications are domain-independent. Therefore, you are not required to define the same application twice, such as Exchange Company A and Exchange Company B, to enable CA Application Delivery Analysis to report on application performance across domains. However, to set different thresholds for application performance, performance operational level agreements (OLAs), and availability OLAs, create an application for each IP domain.

If you do not need to separate duplicate IP traffic, you can use the DNS settings in the Default Domain to query Domain Name System (DNS) and resolve the hostname of a CA Application Delivery Analysis server. Otherwise, CA Application Delivery Analysis uses the monitor feed that is assigned to the server to resolve the hostname.

View a List of Domains in CA Application Delivery Analysis

You can view a list of domain definitions and current domain associations in the Administration section of the CA Application Delivery Analysis management console.

Follow these steps:

1. Click the Administration tab in the management console.
2. Click **Data Monitoring**, **Domains** in the Show Me menu.
The Domains page opens.
3. (Optional) View the DNS settings for a domain by clicking the magnifying glass in the View column.
The Domain Properties page opens.
4. Verify the properties.
5. Click **OK** when you have finished.
The Domains page appears.

Assign a Domain to a Monitor Feed

You can instruct each Standard Monitor to associate the items that it monitors with a custom domain as part of CA Application Delivery Analysis collection device setup.

Follow these steps:

1. Click the **Administration** tab in the management console.
2. Click **Data Monitoring**, **Monitoring Devices** in the Show Me menu.
3. Click **Edit** to edit a multifunction monitoring device, such as a Standard or Multi-Port Monitor.

4. Scroll down to the Monitor Feeds list.
5. Click to edit a monitor feed.
6. Select a custom IP domain.
7. Click **Update**.
   All items detected by this monitor feed are automatically associated with the selected IP domain.

Assign a Domain to a Client Network

After you add a client network, you cannot change its IP domain association. If you need to change the assigned IP domain, first delete the network, and then add it to the correct domain.

Follow these steps:

1. Click the **Administration** tab in the management console.
2. Click **Data Monitoring**, **Networks** in the Show Me menu.
3. Select the IP domain from the list.
4. Click **Add Network**.
5. Enter the required information to add the network.
6. Click **OK**.
   The network is added to the domain.

Assign a Domain to a Server or Server Subnet

After you add a server or server subnet, you cannot change its IP domain. If you add a server or server subnet to the wrong IP domain, first delete it, and then add it to the correct domain.

Follow these steps:

1. Click the **Administration** tab in the management console.
2. Click **Data Monitoring**, **Servers** in the Show Me menu.
3. Select the IP domain from the list.
4. Enter the required information to add the Server or Server Subnet.
5. Click **OK**.
   The Server or Server Subnet is added.

Assign CA Unified Communications Monitor Items to an IP Domain

In the CA Unified Communications Monitor management console, you can instruct collectors to associate the items that they discover with custom domains in NetOps Portal. The act of creating a single custom domain in NetOps Portal enables domain associations for locations, voice gateways, and call servers in any registered data sources.

Items appear with domain designations when they are discovered from call traffic. Previously discovered items do not receive retroactive associations.

Locations are automatically associated with IP domains by the subnets that they contain. To preserve the flow of data collection and the appropriate association of data with IP domains, take care when moving Locations to new IP domains. Follow the procedure in the CA Unified Communications Monitor online Help to change IP domain assignments.

Instruct collectors to associate items with custom IP domains.

Follow these steps:

1. Click Administration, Data Collection, Collectors.
2. Edit each collector to select its domain for the IP Domain parameter.
3. Reload the collectors to send them the domain information.
   Domains are populated with managed items after the next product synchronization.

Discovery

Discovery is the process that DX NetOps Performance Management uses to identify the devices on your network. Devices are identified using the IP domain, IP addresses, IP ranges, and hostnames that you specify in discovery profiles.

Attempts to discover devices are made through SNMP and ICMP protocols. If a device does not respond to SNMP but *does* respond to ICMP, a pingable device is created.

Inventory Discovery

Inventory discovery, or device discovery identifies the following information about devices:

- Which protocols the device responds to, SNMP or ICMP
- The classification of the device, such as router or switch
- The device vendor, such as Cisco or Juniper
- The device type, such as 7700 or 8200

Discovered devices and monitored components take up to 5 minutes to begin synchronizing with NetOps Portal.

Discovered devices are automatically added to device collections depending on the rules that control each device collection membership. During the first synchronization between Data Aggregator and NetOps Portal *after* devices are discovered, the devices are added to collections.

You can use two methods to discover devices:

- Run discovery through DX NetOps Performance Management.
- Discover devices from other data sources.
  For more information, see Discovery From Other Data Sources.

The following workflow offers a best practice to use as a quick reference when performing a discovery of your inventory.

Perform this process as either a user with the Administrator role or as the tenant administrator.

1. To enable the Data Collector to perform queries of device MIB tables that use SNMP, create SNMP profiles.
   For more information, see SNMP Profiles.
2. Create discovery profiles. For more information, see Discovery Profiles.
3. Run discovery and view the results. For more information, see Run Discovery.

Metric Family Discovery

Metric family discovery, also known as component discovery, is a separate process that determines whether a specific metric family is supported for a device.

A metric family defines the set of metrics to collect and report on for a given technology. The values of the metrics are normalized so that reporting is uniform regardless of the data source.

Polling begins automatically after metric family discovery completes. Operational metrics are collected and retained at regular polling intervals for reporting. Examples of operational metrics include error rate, utilization, and bytes in. Configuration data represents or identifies a component or the component configuration, such as port type or component index.

For more information, see Configure Monitoring Profiles.

Quickly Discover SNMP Devices

Discovery is the process that DX NetOps Performance Management uses to build an inventory of devices in your network. You can quickly discover SNMP devices without having to configure SNMP profiles and discovery profiles manually.

For more information, see Discovery.

Follow these steps:

1. Hover over **Administration**, and click **Monitored Items Management: Quick Device Discovery**.
2. Select **SNMP devices**.
3. Select the SNMP protocol to use.
4. Complete the following fields:
   - **IP Domain**
     If you have multiple IP domains, select the IP domain for discovery.
   - **Community Name** (SNMPv1/v2c Only)
     Specify a secure string that lets the data source query the MIB of the associated device. The community that you supply must provide read access to the device MIB.
   - **Verify Community Name**
     Specify the Community Name again for verification.
   - **Port**
     Specify the port that is used to make SNMP connections to your devices.
     Default: 161

     NOTE

     This port can also be used to send SNMP traps to trap receivers associated with this profile through notifications. In this scenario, use 162 by default. For more information, see Configure Notifications.
   - **IPs/Host**
     Specify the IP address ranges that you want to discover for IPv4. Range discovery is not supported for IPv6 addresses.
5. Click **Discover**.
   DX NetOps Performance Management discovers the devices within the specified IP addresses and hostnames, and adds the devices to the system inventory.

Quickly Discover Virtual Network Devices

Discovery is the process that DX NetOps Performance Management uses to build an inventory of devices in your network. You can quickly discover Virtual Network devices without having to configure discovery profiles manually.

For more information, see Discovery.

Follow these steps:

1. Hover over **Administration**, and click **Monitored Items Management: Quick Device Discovery**.
2. Select **Virtual Network Devices**.
3. Select a technology.
4. Complete the following fields for the selected technology:
   - **Viptela SD-WAN**
     - **IP Address**
     - **Protocol**
       The HTTP security scheme for vManage
     - **Port**
     - **User Name**
     - **Administrator Password**
     - **Re-enter Administrator Password**
   - **Cisco ACI**
     - **APIC Host IP Address**

The IP address of the APIC controller host
- **Protocol**
  The communication protocol with the APIC controller
- **Port**
- **User Name**
- **Administrator Password**
- **Re-enter Administrator Password**

– **128T**
- **Host Conductor IP Address/Name**
  The IP address of the 128T Conductor
- **Protocol**
  The HTTP security scheme for 128T SD-WAN
- **Port**
- **User Name**
- **Administrator Password**
- **Re-enter Administrator Password**

– **Nuage**
- **VSD Host IP Address**
  The Virtualized Services Directory (VSD) host
- **Protocol**
  The communication protocol with the VSD
- **VSD Port**
  The port that the VSD UI/API server listens on
- **Stats Host IP Address**
  The IP address of the stats server
- **Stats Protocol**
  The communication protocol with the stats server
- **Stats Port**
  The port the stats server is listening on for REST requests
- **Nuage Organization**
  The name of the Nuage enterprise to manage
- **Time Zone**
  The time zone of the system, which must match the VSD time zone
- **User Name**
- **Administrator Password**
- **Re-enter Administrator Password**

– **SilverPeak**
- **Host Orchestrator IP Address/Name**
  The Unity Orchestrator Management host
- **Protocol**
  The communication protocol with the Orchestrator
- **Port**
- **User Name**
- **Administrator Password**
- **Re-enter Administrator Password**

5. Click **Discover**.
   DX NetOps Performance Management discovers the devices within the specified IP addresses and hostnames, and adds the devices to the system inventory.

Discover Devices

If you do not use quick discovery, you can configure SNMP profiles and discovery profiles manually.

The following video shows the detailed discovery process starting with the required setup:

Follow these steps:

1. Configure SNMP Profiles.
2. Configure IP Domains.
3. Configure Discovery Profiles.
4. Run Discovery.

Discovery Profiles

Discovery profiles specify how discovery operates. Discovery profiles determine the IP domain, IP addresses, IP address ranges, and host names for discovery. You can only specify one IP domain for each discovery profile. Newly discovered devices are created in the selected IP domain. When multiple Data Collectors are deployed in one IP domain, each Data Collector issues a discovery request to each device as specified by the discovery profile. When more than one Data Collector can contact a device, a specific Data Collector is randomly selected to monitor the device. Review the capacity of the Data Collectors and rebalance as required. For more information, see Rebalance the Load on Data Collector. Discovery profiles are only accessible to users in the tenant space where the discovery profile was created. Users that are assigned to the Default Tenant can access discovery profiles in the Default Tenant space. Log in to the correct tenant *before* you create a discovery profile.

Administrators can manage discovery profiles through the UI or through the Data Aggregator REST web services. Log in as a tenant administrator to perform these tasks.

The following video walks through a single device discovery:

Create Discovery Profiles

To specify how inventory discovery operates in your environment, create discovery profiles. To optimize discovery and reduce SNMP traffic, set up granular discovery profiles:

- Use separate discovery profiles for each group of devices that share an SNMP profile.
- Use separate discovery profiles for groups of devices that require different rediscovery schedules.

Follow these steps:

1. Select **Administration**, and click the Data Aggregator data source.
2. Click **Discovery Profiles** from the Monitored Inventory menu.
3. Click **New**.
4. Specify a name and select an IP Domain.
   The following characters are not permitted:
   – Single quotes
   – Double quotes
   – Backward slashes
   – Forward slashes
   – Ampersands
5. Select the IPs/Hosts tab and do one or more of the following actions:
   – (Optional) Navigate to and import a CSV file of IP addresses. The CSV file can contain a comma-separated list of IPv4 addresses, IPv6 addresses, IPv4 address ranges, and hostnames. Browse to select the file and click **Open**.

NOTE
To apply Chinese characters to the alias name, save the CSV file in UTF-8 format.

– Specify IP address ranges, individual IP addresses, and host names. Comma-delimited values are accepted.

NOTE
If an IP range includes multiple IP addresses, and one of the IP addresses maps to the hostname, discovery always uses the hostname IP as the primary IP address.

6. (Optional) To regularly update information for discovered devices and discover new devices, configure a schedule for the discovery profile. Select the **Schedule** tab, and specify a schedule.
7. (Optional) To use specific SNMP profiles, select the **SNMP** tab, and complete the following steps:
   a. Select **Use specific list of assigned SNMP profiles**.
   b. Move one or more SNMP profiles from the list of available profiles to the assigned list.

TIP

Using a subset of SNMP profiles reduces SNMP traffic.

8. Select the Advanced tab, and configure advanced options:
   – **Naming Order**
   Change the attribute priority, which the system uses to name the discovered devices. Any device item that the discovery profile creates is named with the highest available naming convention. If a higher priority attribute is unavailable for the device, DX NetOps Performance Management uses the next highest priority attribute. For virtual machines, DX NetOps Performance Management ignores the naming order and uses the names from vCenter.

NOTE

If you use host name to name devices, the device name is updated automatically when the hostname changes. If you use another attribute, such as system name, the change to the device name occurs when the discovery profile runs again. Run discovery manually or configure a schedule for the discovery profile.

   In some unusual configurations, the network might not have unique DNS host names. To reconcile devices by the IP address and system name only, select **Exclude Host Name**.
   – **ICMP Discovery**
   To determine if a device can respond to ICMP, select **Use ICMP**.
   To create pingables for devices that respond to ICMP but not SNMP, select **Create Pingables**.
9. Click **Save**.
   The discovery profile is created and is displayed in the Discovery Profiles list. If the profile has a schedule, discovery runs at the scheduled time.

Discovery Profile IP Ranges

In a discovery profile, you can specify the IP address ranges that you want to discover for IPv4. Range discovery is not supported for IPv6 addresses.

When you specify IP ranges in the discovery profile, the following rules apply:

- An IPv4 range can contain wildcards (*). A wildcard represents a full range for an IP octet: 0-255
- An IPv4 range can contain hyphens (-). A hyphen can exist between the lower IP address and upper IP address. A hyphen can also be in the IP octets in the lower IP address.

Examples: Valid IP Ranges

- Both of the following examples attempt to discover devices at every IP address from 10.25.1.0 to 10.25.1.190:

```
10.25.1.0-10.25.1.190
```

OR

```
10.25.1.0-190
```

- Both of the following examples attempt to discover devices at every IP address from 10.25.0.0 to 10.25.255.255:

  ```
  10.25.*.*
  ```

  OR

  ```
  10.25.0.0 - 10.25.255.255
  ```

- Both of the following examples attempt to discover devices at every IP address from 10.25.0.3 to 10.25.0.40 and from 10.25.1.3 to 10.25.1.40:

  ```
  10.25.0-1.3-40
  ```

  OR

  ```
  10.25.0.3 - 10.25.0.40, 10.25.1.3 - 10.25.1.40
  ```

- Both of the following examples attempt to discover devices at every IP address from 10.25.0.0 to 10.25.0.5, from 10.25.1.0 to 10.25.1.5, and so on, up to 10.25.255.0 to 10.25.255.5:

  ```
  10.25.*.0-5
  ```

  OR

  ```
  10.25.0.0 - 10.25.0.5, 10.25.1.0 - 10.25.1.5 ... 10.25.255.0 - 10.25.255.5
  ```

Examples: Invalid IP Ranges

- The following example is invalid because the upper IP address is incomplete:

  ```
  10.25.1.0 - 10.23
  ```

- The following example is invalid because when a hyphen (-) is used in an octet in the lower IP address, the upper IP address cannot be present:

  ```
  10.25.1.0-190 - 10.25.1.255
  ```

- The following example is invalid because when a wildcard (*) is used in an octet in the lower IP address, the upper IP address cannot be present:

  ```
  10.25.*.0 - 10.25.255.255
  ```

- The following example is invalid because it is unclear whether the wildcard octet (1*) implies 10.25.10-19.0 or 10.25.10-199.0:

```
10.25.1*.0
```

Run Discovery

Inventory discovery finds devices as specified by a discovery profile. Run discovery manually, or schedule the discovery.

If discovery hangs for more than 10 minutes, the discovery is aborted. A discovery is considered to be hanging when no new devices are discovered within 10 minutes and the state for the discovery profiles have not changed within 10 minutes. An audit event is generated on the Data Aggregator device. If no devices were discovered, the state of the discovery profiles indicates FAILURE. If at least one device was discovered, the state indicates PARTIAL_FAILURE.

The discovered devices and monitored components can take up to 5 minutes to synchronize with NetOps Portal. When the synchronization is complete, the devices and components appear in the **Inventory** tab.

Manual Discovery

An administrator or a tenant administrator can run manual discovery. To run a discovery as the administrator, first configure the Data Collector for the Default Tenant domain.

Follow these steps:

1. Go to **Administration**, and click the Data Aggregator data source.
2. Click **Discovery Profiles** from the Monitored Inventory menu.
3. Select one or more discovery profiles, and then click **Run**.
   The state of the selected profiles must be READY or SCHEDULED.
4. Click **Yes** in the confirmation dialog.
   Discovered devices are added to device collections, which initiates component monitoring and polling.

Scheduled Discovery

You can configure a discovery to run on a daily or weekly schedule. When the discovery is scheduled, the State for the discovery profile is SCHEDULED and the next scheduled run time appears.

To run schedule discoveries, configure the options in the **Schedule** tab when you create or edit a discovery profile.

View Discovery Results

The discovery results show the number of devices that were discovered during a specific discovery instance. You can view specific details about these discovered devices, including the IP address, model, type, vendor name, location, and protocols.

Follow these steps:

1. From the **Discovery Profiles** list, select a discovery profile, and click **History**.
2. Select a Discovery instance.
3. (Optional) Filter the **Discovered Devices** table by device type or state.
   The discovery results appear in the Discovered Devices table.
   The SNMP Profile column shows the highest ranked SNMP profile that the device responded to.
   The State column indicates one of the following states:
   - **New**
     Indicates a device that was discovered for the first time when this discovery profile was run.
   - **Changed**
     Indicates that a device type has changed from a previous discovery. For example, a previously discovered pingable device is now discovered as a manageable device. Or a previously manageable device with the

device type of Switch has now changed to the device type of Router. Devices with only attribute changes, such as hostname, or system description are not classified as Changed.

- **Unchanged**
  Indicates that existing devices have not changed. Existing devices with only attribute changes are also classified as Unchanged. Existing devices that show different IP addresses were discovered and are being monitored with a different IP address. Many devices can respond to multiple IP addresses. DX NetOps Performance Management maintains the full set of IP addresses for each device.
- **Deleted**
  Indicates that the device was deleted from Data Aggregator since the discovery was run.

  NOTE
  If a single IP address or hostname that was specified in the discovery profile is not found, the device type indicates Inaccessible. DX NetOps Performance Management does not report inaccessible devices from IP ranges.

Get a List of Unreachable Devices

If discovery fails for any of your devices, you can access a list of the unreachable devices using the `discoveryinstances` rest service.

NOTE
This method works only when you specify each IP address or host in your discovery profile. This method does not provide a list when you specify an IP range in your discovery profile.

Follow these steps:

1. View a list of the discovery instances available at the following location:
   `http://DA_HOST:8581/rest/discoveryinstances/`

2. Find the discovery instance that corresponds to when you ran your discovery.
   Example:
   ```
   <DiscoveryInstance version="1.0.0">
        <ID>6508</ID>
        <IPSweepStartTime>Mon Sep 18 15:44:31 2017 -0400</IPSweepStartTime>
   ```

3. Note the discovery instance ID.
4. View the results for your discovery instance at the following location:
   `http://DA_HOST:8581/rest/discoveryinstances/DISCOVERY_INSTANCE_ID`

5. View the `<UnreachableDevicesList>` section.
   Example:
   ```
   <DiscoveryInstance version="1.0.0">
        <ID>6508</ID>
        <IPSweepStartTime>Mon Sep 18 15:44:31 2017 -0400</IPSweepStartTime>
        <TestedCommProfilesList>
             <TestedCommProfiles>4657</TestedCommProfiles>
             <TestedCommProfiles>4603</TestedCommProfiles>
             <TestedCommProfiles>4656</TestedCommProfiles>
        </TestedCommProfilesList>
        <ProgressPercentage>100</ProgressPercentage>
        <PingResponseDeviceCount>0</PingResponseDeviceCount>
        <StartTime>Mon Sep 18 15:44:31 2017 -0400</StartTime>
   ```

```
        <CompletionTime>Mon Sep 18 15:44:39 2017 -0400</CompletionTime>
        <IPSweepCompletionTime>Mon Sep 18 15:44:39 2017 -0400</IPSweepCompletionTime>
        <CompletionStatus>SUCCESS</CompletionStatus>
        <ProfileID>6507</ProfileID>
        <IPSweepTotalSuccess>0</IPSweepTotalSuccess>
        <UnreachableDevicesList>
              <UnreachableDevices>10.255.255.255</UnreachableDevices>
              <UnreachableDevices>192.168.0.0</UnreachableDevices>
              <UnreachableDevices>172.31.255.255</UnreachableDevices>
              <UnreachableDevices>192.168.255.255</UnreachableDevices>
              <UnreachableDevices>10.0.0.0</UnreachableDevices>
              <UnreachableDevices>169.254.0.0</UnreachableDevices>
              <UnreachableDevices>169.254.255.255</UnreachableDevices>
              <UnreachableDevices>172.16.0.0</UnreachableDevices>
         </UnreachableDevicesList>
        <NewAccessibleOnlyButDeletedDevicesCount>0</
NewAccessibleOnlyButDeletedDevicesCount>
        <Item version="1.0.0">
             <CreateTime>Mon Sep 18 15:44:31 2017 -0400</CreateTime>
        </Item>
 </DiscoveryInstance>
```

Rediscovery

Rediscovery updates information about existing devices and discovers new devices. When you run a discovery profile that includes existing devices, DX NetOps Performance Management detects the following changes to the device:

- System name
- System contact
- Device type
- Location
- Vendor
- Device description
- Device model

Changes to device attributes can result in changes to the groups and device collections that a device is in. Changes to groups and device collections can potentially add or remove monitoring profiles. Running the discovery profile does not evaluate or update the metric families for the device.For information about how to update metric families for the device, see Rediscover Metric Families.

Rediscover a Device

To update a single monitored device, run rediscovery for that device.

Follow these steps:

1. Select **Administration**, and click the Data Aggregator data source.
2. Click **Monitored Devices** from the Monitored Inventory menu.
3. Select the device.

4. Click **Rediscover**.
   DX NetOps Performance Management runs device discovery and metric family discovery for the device. Changes in the device attributes can take up to 5 minutes to be seen in inventory or dashboards views.

Discovery From Other Data Sources

You can configure whether all new inventory from a data source contributes to the Data Aggregator.

To enable this option, edit a data source and select **Contribute inventory to the Data Aggregator**. If unselected, the existing inventory remains the same, and no new data source inventory contributes to the Data Aggregator.

Automatic synchronization includes only devices that are discovered after you select this option. To discover previously discovered devices, perform a full synchronization of the Data Aggregator. DX NetOps Performance Management creates a discovery profile that includes the IP addresses of the devices. This discovery profile attempts discovery 1 minute after new IP addresses are pushed into the IP domain discovery profile for the Data Aggregator. Otherwise, the discovery profile attempts discovery once per day. For more information, see Configure a Data Source and Synchronize Data Sources.

Discovery and Polling in VMware Environments

DX NetOps Performance Management can discover and monitor your VMware virtual machines and ESX hosts. The discovery and monitoring process for these devices and components differs to accommodate the collection of data from vCenter. You can discover the VMs and ESX Hosts directly with SNMP, and you can collect vCenter data through the vCenter Server Application Insight Module (VCAIM).

DX NetOps Performance Management discovers ESX hosts and virtual machines in the following ways:

- Through ICMP
- Through SNMP, if the servers have an SNMP agent deployed
- Through discovery of a server running systemEdge with the VCAIM

Only one device item is created for each ESX host or virtual machine.

By default, the change detection rate for virtual machines is 15 minutes. The change detection rate for ESX hosts is 24 hours.

> **NOTE**
>
> Any virtual device (Virtual Machine or ESX), discovered with systemEdge is not polled for its vCenter statistics when the device lifecycle state is Retired.

systemEdge Discovery

When discovery runs for a server running systemEdge with the VCAIM, DX NetOps Performance Management uses the following logic during discovery of the vCenter Server:

The VMware ESX Host monitoring profile discovers all ESX hosts, and DX NetOps Performance Management uses the following rules to create devices:

- DX NetOps Performance Management creates SNMP devices for each device with an IP address that is used by another discovered device. The device type is ESX Host and Server.
- DX NetOps Performance Management creates pingable devices with the ESX Host device type in the following situations:
  - A device does not have an IP address.
  - The IP address of a device is not used by another device.

The VMware Virtual Machine monitoring profile discovers all virtual machines, and DX NetOps Performance Management uses the following rules to create devices:

- DX NetOps Performance Management creates SNMP devices for each device with an IP address that is used by another discovered device. The device type is Virtual Machine and Server.
- DX NetOps Performance Management creates pingable devices with the Virtual Machine device type in the following situations:
  - A device does not have an IP address.
  - The IP address of a device is not used by another device.

Groups

Groups determine the data that you see in dashboards when you log in. The group that is applied as a filter to the current dashboard is the group context for that dashboard. When you log in to CA NetOps Portal, the pages reflect the context of your default permission group. You can change the default group for your user account to view data from another group in the dashboards.

Groups are organized into a hierarchical tree structure. The Groups tree helps you define relationships, policies, and dependencies among services, devices, applications, locations, and users within your organization. Organize your group structure according to business and reporting needs. To create a regional structure that represents regions, countries, and locations, use site groups. Use custom groups for other types of organizations, such as customers, services, or technologies.

Threshold profiles apply threshold rules to all items in a group. The group hierarchy requirements for thresholding are probably different from the requirements for reporting. Create separate groups that address both sets of requirements. Consider the different layers of the network and how to create thresholds for components in those layers. For example, you might threshold on CPU, memory, and interface metrics on the core network differently to the distribution layer. Create multiple groups to apply threshold rules appropriately.

Tenants include special types of system groups to maintain separation among customer deployments. Tenants can also contain entire custom grouping structures.

NOTE

The lock icon that appears on a group icon, such as System Groups, indicates that a group cannot be edited.

System Groups

System Groups are read-only groups that are automatically created based on information from data sources. System groups can be viewed, applied as permission groups to user accounts, or copied to custom or site groups.

Inventory Group

The Inventory group includes all managed items that are discovered by all registered data sources. This group also organizes data sources, IP domains, and managed items in subgroups. The Inventory group contains its own system subgroups to organize managed items by their type.

WARNING

DataSource@Hostname groups, such as DataAggregator@mydahost.ca.com, are not synced to data sources. These groups cannot be used for reporting, or as default user groups. Using these groups causes synchronization errors.

The following system groups appear under the Inventory node:

- **All Items**

Includes subgroups of managed items, which are categorized by type.
  - **All Pingable Devices**
    Includes all discovered devices that cannot be contacted using SNMP.
  - **ESX Hosts**
    Includes all VMware servers that host virtual machines.
  - **Interfaces**
    Includes router and switch interfaces from all data sources.
  - **Routers**
    Includes all routers from all data sources.
  - **Servers**
    Includes all servers from all data sources.
  - **CA Application Delivery Analysis**
    Includes all networks that CA Application Delivery Analysis (ADA) has observed. A ADA network consists of an IP address and mask.
  - **Switches**
    Includes switches from all data sources.
  - **Virtual Machines**
    Includes all virtual machines running on all ESX servers.
- **Data Sources**
  Includes all registered data sources. Each data source has a dedicated group under this node. Some data sources have their own system subgroups, which appear when you expand the data source group.

  WARNING

  Some data sources, such as Data Aggregator and Event Manager, do not have groups. The groups are not designed as reporting groups and might not provide data when selected as the context.

- **IP Domains**
  Includes all custom IP domains that are created by the administrator. Also includes the Default Domain, which contains all items that are not explicitly assigned to a custom domain. For more information, see IP Domains.
- **VNA Domains**
  These groups contain hierarchically organized items from DX NetOps Virtual Network Assurance. The structure groups the items into logical reporting groups. Use these groups for reporting. The highest level groups under this node are user configurable in DX NetOps Virtual Network Assurance.

Custom Collections

The Custom Collections group represents the collections of devices. Collections are groupings of devices that are monitored using the rules that are specified in monitoring profiles. The factory collections are not visible in the Groups tree. Collections include only devices.

This group lets you create custom collections. Any subgroup that you add to the Collections group is synchronized to the Data Aggregator as a collection.

Custom Groups

Custom groups create hierarchical levels and organize items into logical relationships within the Groups tree. Custom groups at the top level of the Groups tree represent geographical, topological, or functional divisions within your organization. Lower-level custom groups, or subgroups, typically represent one of the following options:

- Applications
- Devices
- Job functions of IT staff
- Services

Only users that have the "Administer Groups Owned by You" role right can create and edit custom groups, which filter the data in dashboards and views. The group context for a dashboard or view determines the data that is presented.

Custom groups are used to monitor and manage your system. Creating custom groups lets you organize data and assign operator permissions to access data.

You can use group rules to add items to groups automatically as they are discovered. Setting up rules makes it easier to populate and maintain groups. You can also populate custom groups by manually adding specific items, such as routers or interfaces that are logically or geographically related.

Site Groups

Site groups are special custom groups that are based on sites, such as branch offices, or on physical locations, such as regions or cities. Site groups let you create navigation functions within dashboards to present views across all sites. They include a Time Zone and a Business Hours parameter to let you see prioritized data from business-critical times of day.

Site groups also provide a granular context to apply to dashboards. For example, after you create a site group for each of your sites, a single dashboard can individually report on each site. Create a site group for each data center within your enterprise, and for other major infrastructure locations.

Groups for Multi-Tenant Deployments

When the administrator for the Default Tenant creates at least one tenant, features to support multi-tenancy are enabled. Multi-tenant deployments consist of multiple discrete enterprises with IP addresses that might overlap. More groups appear in the Groups tree to let the administrator organize tenant inventories and allocate permissions:

- **Tenants**
  The Tenants group includes all tenants. Tenants are used with IP domains to monitor separate customer environments with a single CA NetOps Portal instance. Each tenant can contain multiple subgroups of items that are not shared among tenants.
  Tenant administrators can create custom groups within their tenant. For the global administrator, tenant groups appear under the Tenant node in the Groups tree.
- **Global Tenant Groups**
  Global Tenant Groups contain groups of items that help the global administrator manage tenant environments. These groups let the administrator visualize and organize shared items, which are not explicitly associated with a tenant IP domain. The groups that allocate access to data from shared items appear under each tenant.
  When you expand the top-level Inventory group, the following group appears in a multi-tenant deployment:
- **IP Domains**
  Includes all custom IP domains that are used to associate managed items with tenants. Also includes the Default Domain, which contains all items that are not explicitly assigned to a custom domain. For more information, see IP Domains.

In a multi-tenant deployment, each tenant has its own groups. Tenant users cannot see items outside of the tenant group unless the global administrator grants access with Service Provider groups.

- **Groups (Tenant)**
  Lets the global administrator or tenant administrator create custom groups. Select this node to enable the Add Group button.
- **Inventory (Tenant)**
  Includes all managed items that are associated with the tenant IP domains. Items from all registered data sources might appear in this group.

Each tenant also has the following system subgroups in its Inventory group:

- **IP Domains**

The IP Domains subgroup represents the IP domains that are associated with this tenant. Any managed items that have been discovered are associated with this tenant through its IP domains. To see the managed items of the tenant, click a tenant IP domain in the Groups tree.

- **Global Tenant Groups**
  The Global Tenant Groups include groups that the global administrator has populated with shared items that this tenant can access. Use these groups to grant access to data from shared devices to selected tenant user accounts.
  For example, a router that the service provider owns handles traffic from multiple tenant domains. Using Service Provider Defined groups, the global administrator can allocate tenant access to data from that router. The tenant can then independently monitor and verify system performance.
- **Global Tenant Items**
  Items that are not explicitly associated with a tenant IP domain are automatically placed in the Service Provider Items group. The global administrator can place these items into Service Provider Defined Groups to allocate tenant access to data from shared items.

Device Collections

Device collections are logical groupings of monitored devices. Custom device collections provide granular control over polling. Most deployment require custom device collections. Collections include *only* devices.

Custom Device Collections

The factory device collections are mostly for use in a lab or in a demo setting. In a real production deployment, the best practice is to design and configure custom device collections to have granular control over what is being polled. For example, disable polling of a device by disassociating the device from any other device collection that has monitoring profiles that are associated to it. If you are associating monitoring profiles to the factory device collections (such as All Routers), then you cannot stop a single device from being polled. Devices cannot be removed from factory device collections, so you disassociate the monitoring profiles instead to disable polling. You then create custom device collections that contain devices that you want to apply the same polling policy to. Associate monitoring profiles (or custom monitoring profiles) to those custom device collections to begin polling.

Create custom device collections in NetOps Portal, then either synchronize them immediately with Data Aggregator, or wait for the automatic synchronization. Upon synchronization, Data Aggregator creates the corresponding device collections for use in monitoring devices.

Access the Monitoring Configuration menu to see a list of device collections and to see the monitoring profiles that are applied to each. Administrators can view the device collections for the tenant they are administering. A tenant administrator can view its own list of device collections.

> **NOTE**
>
> Only users with the Administrator role can create or edit collections because collections control polling. Collections can significantly impact system load and performance.

Consider the following best practices for organizing devices into collections for monitoring:

- Create custom collections that match the monitoring requirements in the environment.
  - Consider the different layers of the network, access, distribution, and core. Devices in different layers might require different levels of monitoring.
  - Consider which technologies and metric families are required. Metric families that would be applied to all devices, such as CPU and memory, apply to broad collections. Targeted monitoring, such as QoS and IPSLA, apply to limited collections.
- Create collections that enable the flexibility to break out monitoring.

- Some devices are included in multiple collections so that specific metric families are polled at different rates.
- Devices in different collections have different filtering criteria.
- Different monitoring requirements depending on importance of device

Factory Device Collections

Several factory (out-of-the-box) device collections are provided to get data into your system quickly and test the product. Devices that are detected during discovery are added to these device collections depending on their type. For example, routers are added to the factory All Routers device collection. Upon synchronization, these monitored devices are added to the corresponding device collections in NetOps Portal.

Factory monitoring profiles are then automatically applied to factory device collections, allowing data to be collected immediately without any intervention on your part. Once this data has been collected, you can run reports on the data to gain a better understanding of your network.

The following factory device collections are provided:

> **NOTE**
> The factory device collections are mostly for use in a lab or in a demo setting. In a real production deployment, the best practice is to design and configure custom device collections to have granular control and optimal data collection.

Access the Monitoring Configuration menu to see a list of device collections and to see the monitoring profiles that are applied to each. Administrators can view the device collections for the tenant they are administering. A tenant administrator can view its own list of device collections.

All Devices

The All Devices device collection is a factory device collection. Manageable and pingable devices that are detected during a discovery are automatically placed into the All Devices device collection. Inaccessible devices are not included in the All Devices device collection.

> **WARNING**
> Do not associate monitoring profiles with the All Devices device collection. Doing so can cause extra SNMP requests being made to pingable-only devices, and can result in sporadic metric family support.

All Routers

The All Routers device collection is a factory device collection. Routers that are detected during a discovery are automatically placed into the All Routers device collection.

> **NOTE**
> Routers can appear in both the All Routers device collection and the All Switches device collection.

All Servers

The All Servers device collection is a factory device collection. Physical and virtual servers (hosts) that are detected during a discovery are automatically placed into the All Servers device collection. Network devices such as routers and switches are not included in the All Servers device collection.

All Switches

The All Switches device collection is a factory device collection. Switches that are detected during a discovery are automatically placed into the All Switches device collection.

> **NOTE**
> Switches can appear in both the All Routers device collection and the All Switches device collection.

All Manageable Devices

The All Manageable Devices device collection is a factory collection. Manageable devices collect advanced performance statistics and are monitored with a protocol such as SNMP. Manageable devices that are detected during a discovery are automatically placed into the All Manageable Devices device collection.

Pingable devices can only be monitored for availability and do not provide any additional performance metrics. Therefore, pingable devices are not included in the All Manageable Devices device collection.

Note: Manageable devices can appear in both the All Devices device collection and the All Manageable Devices device collection.

All ESX Hosts

The All ESX Hosts device collection is a factory device collection. ESX hosts that are detected during discovery are placed into the All ESX Hosts device collection automatically.

All Virtual Machines

The All Virtual Machines device collection is a factory device collection. VMware virtual machines that are detected during discovery are placed into the All Virtual Machines device collection automatically.

All VMware vCenters

The All VMware vCenters device collection is a factory device collection. All servers that are running systemEdge with the VCAIM that are detected during discovery are placed into the All VMware vCenters device collection automatically.

Manage Groups

Before you create a group, consider the types of access permissions that users require to perform their monitoring duties. If you are deploying business hours, create a site group. Site groups are custom groups that are based on physical locations, such as a city, region, office, or campus. They include Time Zone and Business Hours parameters to let you precisely filter data from business-critical times of day.

> **NOTE**
> You can apply business hours filters only in whole hour increments. Only site groups that have time zones with only whole hour offsets are available for selection.

Create groups under the **All Groups** node in the Groups tree, or within an existing custom or site group. You cannot add groups to system groups, which appear locked in the Groups tree. However, while the Collections group appears locked, it can contain child groups, but cannot be modified. To keep reporting times within reasonable limits, a maximum of 2000 child groups is recommended for any parent group.

By default, only users with the Administrator or Tenant Administrator role can administer groups that other users create. To administer groups that other users created, you require the Administer Groups Owned by You and Others role right. With this role right, you can modify child groups that other users created if you have the rights to administer the parent group.

> **WARNING**
> If you create a group for a data aggregator data source, limit the group membership to 10,000 items, including the children of managed items. Limiting the number of items keeps reporting times within reasonable limits.

> **NOTE**
> For optimal performance, collapse the **Items** section until you need to manage items. Likewise, collapse each managed item type until you need to manage a specific item type.

Create or Edit a Group

Follow these steps:

1. Log in as an Administrator or as a user with the My Custom Groups functionality enabled.
2. Do one of the following tasks:
   - Hover over **Administration**, and click **Group Settings: Groups**.
   - Click the name of your user account in the upper-right corner, and then click **My Custom Groups**.

     NOTE

     The groups that you can see are groups that the administrator selected for you, based on your responsibilities. The custom groups that the administrator created cannot be edited within the My Custom Groups interface. These read-only groups appear as group references; their properties tab shows a path to the original group.
3. Do one of the following tasks:
   - To add a group, select a location for the new group in the **Groups** tree, and click **Add Group**.
   - To edit a group, select the group from the **Groups** tree.
4. Specify or edit the values for the following parameters:
   - **Group Name**
     Use a strategic naming convention. You can use all special characters in the group name except the vertical bar (|), the forward slash (/), and the backslash (\).

     NOTE
     Groups that are synchronized from DX NetOps Spectrum retain restricted characters except the backslash (\). This character is removed from synchronized group names.
   - To create a custom group, select **Custom** from the **Group Type** list. To create a site group, select **Site**.
     To create a site group, optionally specify the following parameters:
     - Latitude
     - Longitude
     - Location
     - Elevation
     - Time zone
     - Business hours
   - **Description**
5. Confirm the setting for the following parameter:
   - **Include the children of managed items**
     Adds the children of managed items automatically when the items are added to this group. If you disable this option and you add a router to the group, the interfaces on that router are not included. Therefore, their data is not visible in drilldown views.
     Default: Selected
6. Click **Save**.
   The new group appears in the Groups tree. To configure views or on-demand reports with the group, wait for the group to be synchronized to the data sources.
   The group is empty until you add items. You have two options for adding items to a custom group:
   - Manually populate the group by adding items in the **Manage Groups** interface.
   - Create rules to manage the group membership.

Add Managed Items to a Group Manually

To populate custom groups manually, add managed items to groups.

NOTE
System groups with a lock symbol in the **Groups** tree are read-only. You cannot add items to or remove items from system groups. Custom groups that are created by the administrator are also locked.

Follow these steps:

1. Select the group to which you want to add managed items from the **Groups** tree.
   Managed items that have already been added to this group appear in the right pane.

   NOTE
   Managed items that are manually added directly to a group appear as **Direct Items** in the **Group Properties** pane. Managed items that are added to a group as children of a managed item are **Inherited Items** in the Group Properties.
2. Scroll to the managed item type in the **Items** section.
   The available items depend on the item type, the data sources that are registered, and the items that are discovered. To see more pages of items, click the links below the list. You can also search for an item in the list using the Quick Filter.
3. Click to add a managed item.
4. Select items by selecting their checkboxes. To select all items on a page, select the checkbox in the table header row.
5. Click **Add**, and then **Save**.
   The new items are added to the group.

View Group Membership

View a sortable list of all items that have been added to a system group or custom group on the **Manage Groups** page. You can verify group rules, or whether custom scripts have created and populated groups appropriately. You can also view a list of items in a selected group.

Use filters to select the types of items that you want to see, such as all items that were added to the group manually. By default, the list on the Items tab only shows items that are added directly to the group. The items are added either manually or by the application of a rule (Direct Items membership). If you report on a dashboard in the context of a group, all items in the subgroups of the group appear in the dashboard. The items in the subgroups do not appear on the **Group Administration** page.

Follow these steps:

1. Select the group for which you want to view membership details from the **Groups** tree.
2. Scroll to the **Items** section.
3. To specify the items to display, select a filter from the **Show Items** drop-down list.
   The following membership types are applicable, depending on the type of group that you selected:
   - **Direct Items**
     Includes the items that were added directly to the group, either manually or by the application of a rule. You can add or remove items only when **Direct Items** is selected. The **Added By** column indicates whether the item was added manually or by a group rule.
   - **Direct and Inherited Items**
     Includes all items in the group, whether they were added directly or inherited as the children of items that were added directly.
     The checkbox next to **Options** on the Properties tab determines the ability to inherit items. Excluded items are not inherited.
   - **Inherited Items**
     Includes only the children of managed items in the group. When you enable inheritance for this group and add a router, all interfaces that are associated with the router are added to the group.
     You can remove inherited items only by removing the parent item.
   - **Excluded**

Items in this list are not added to the group by group rules. To prevent groups rules from adding an item to the group, add the item to the **Excluded** list.
This list also includes manually-deleted items if a group rule originally added that item. To add an excluded item to the group, remove the item from the **Excluded** list, and run group rules.

4. Select an item type from the list.
   A list of all items of the selected type that are included in the group appear.

Manage Group References

A group reference is a copy of another group. Any group that is referenced in the Groups tree has a **References** section at the bottom of the page.

To view and remove references to that group, scroll to the **References** section.

To delete group references, scroll to the **References** section for the original group. While deleting a group that has been referenced deletes all of its references, deleting a group reference does not affect the original group. Deleting a subgroup that is a reference does not affect the original group or its parent group. To prevent problems when deleting a group that contains subgroup references, delete all the references before deleting the group.

Delete Groups

The global administrator can delete custom groups, including groups that belong to any tenants. A tenant administrator can also delete custom groups that belong to that tenant definition. The subgroups of the deleted group are also deleted. You cannot delete system groups. Likewise, you cannot delete the Default Domain group.

Manage Subgroups

After you have created custom groups, you can populate them by adding subgroups that contain managed items.

Add Subgroups to a Group

To create a hierarchical structure, you can create new groups within custom groups that you previously created. You can also add an existing group to another group to create a subgroup.

Follow these steps:

1. Navigate to the **Manage Groups** page.
   The current groups appear in a tree structure.
2. Right-click the parent group, and select **Add Group**.
   The Add Group dialog appears.
3. Specify the parameters.
   The Groups tree appears.
4. Click **Save**.
   The existing group and all of its subgroups are added to the selected parent group as reference groups.

Copy a Subgroup into a Group

You can copy system groups or other custom groups into high-level groups to create subgroups. When you copy a group, you create a group reference, which cannot be modified but can be removed. When you copy a group, the original group displays the References section at the bottom of the page. Scroll to the **References** section to view the locations where copies of the group have been placed. Any changes that you make to the original group are reflected in all the references of the group. Removing a group also deletes all of its references.

Follow these steps:

1. Navigate to the **Manage Groups** page.
   The current groups appear in a tree structure.
2. Select the group that you want to copy. All its subgroups are automatically included in the selection.
3. Right-click, and select **Copy Group**.
4. Select the parent group where you want to add the subgroup.
5. Right-click, and select **Paste Group**.
   The existing group and all its subgroups are copied to the selected parent group.
   The group icons indicate that they are read-only group references.

Manage Group Rules

Use rules to keep custom groups up-to-date when systems and networks change. Newly discovered items that meet rule specifications are added to groups. Similarly, if the items do not meet rule requirements, or if the items are no longer monitored, items are removed. After you create a rule, you can modify it by deleting filters or adding subrules.

> **NOTE**
> If you remove an item from a group manually, the group rule cannot add the item to the group again.

In this article:

The following default restrictions apply to group rules:

- Maximum 50 rules per group
- Maximum 50 conditions per rule
- Maximum 50 'OR' matches in a condition
- Maximum 20 'AND' subrules

Create Group Rules

Follow these steps:

1. Log in as an Administrator or as a user with the My Custom Groups functionality enabled.
2. Do one of the following tasks:
   – Hover over **Administration**, and click **Group Settings: Groups**.
   – Click the name of your user account in the upper-right corner and click **My Custom Groups**.
   The My Custom Groups page shows a tree view of group structure and a tabbed view of group properties.
3. Select the group to which you want to add managed items.
   Items that have already been added to this group appear in the right pane.
4. Scroll to the **Rules** section, and then click **New Rule**.
5. Type a name for the rule.
6. Select the type of managed item that you would like to add to the group from the **Add** drop-down list.
   The list might include item types that have not been synchronized to your system. These item types let you create rules that are applied later when you discover these items. To view the existing item types on your system, see the Inventory menu.
7. Click **Add Condition**.
   A row of drop-down lists and fields appears.

   > **NOTE**
   > By default, a condition that filters added items to your top-level permission group exists in the list of conditions. You can adjust the filter to a group of your choice. Filter to the group with the fewest items to improve rule processing.
8. Select a method for identifying managed items. For example, select **Device Type**.

The remaining lists are updated to match the type of selected item.

NOTE

If you select a string field like Device Context Type, the 'is equal to' operator returns only *exact* string matches (for example, '`server,WirelessController` '). Use the 'is like' operator to return matches on a portion of the string (for example, '`server,WirelessController` ', '`device,WirelessController` ', and '`router,WirelessController` '). Do not use spaces.

9. Select an operator from the second list. For example, select 'is equal to'.

   WARNING
   Use CIDR notation for the IP addresses that you specify for the 'is in subnet' and 'is not in subnet' options. Use dot-decimal notation for the IP addresses that you specify for the 'is between' and 'is not between' options.

10. (Optional) Enter a text string to match in the remaining condition field. For example, to add all routers and servers in the Southwest region, enter a string with the appropriate naming convention, such as "sw*".

    NOTE
    The * wildcard character is accepted in this field for a multicharacter match.

    WARNING
    Some methods have a limited list of acceptable condition values. For example, when the method for identifying managed items is device alarm state, the text value must match one of the actual device alarm states. Acceptable values are discussed further on this page.

11. (Optional) To add 'OR' matches, click + at the end of the condition.
12. (Optional) To add 'AND' matches, click **Add Condition**.
13. Click **Save**.
14. Click **Preview Results** to confirm that the new rule includes the correct items.
    The results are shown in the Preview section. You can scroll to each item type to view the specific items that were added.
15. (Optional) Click **New Rule** to add other item types to the group.
    Each item type requires its own rule.
16. When you have finished creating rules, click the following options:
    - **Save**
      Saves the rules without running them. The group is populated during the next global synchronization, which occurs approximately every 5 minutes.
    - **Run Rules**
      Populates the group immediately.

Edit Group Rules

Follow these steps:

1. Log in as a user with the required administrative role rights.
2. Navigate to the **Manage Groups** page.
   The current groups appear in a tree structure.
3. Expand the **All Groups** node in the Groups tree.
4. Select the group that contains the rule that you want to modify.
5. Scroll to the **Rules** section.
6. Click a rule in the list.
7. Click **Edit Rule**.
8. Edit existing filters, add filters or subrules, or remove filters or subrules as needed.
9. Click **Save**.
10. Click **Preview Results** to confirm that the modified rule adds the appropriate items the group.

11. When you have finished editing the rules, click the following options:
    - **Save**
      Saves the rules without running them. The group is populated during the next global synchronization. Global synchronization occurs approximately every 5 minutes.
    - **Run Rules**
      Populates the group immediately.

Delete Group Rules

You can delete the rules that you have created. Deleting a group rule removes any items in the group to which the rule is applied immediately. The items are not deleted from the inventory. They are no longer available on the Items tab for the affected group.

Follow these steps:

1. Log in as a user with the required administrative role rights.
2. Navigate to the **Manage Groups** page.
   The current groups appear in a tree structure.
3. Select the group with the rule that you want to delete in the Groups tree.
4. Scroll to the **Rules** section.
5. Click a rule in the list.
6. Click **Remove Rule**.
7. Click **Yes**.
   The rule is no longer applied to the group.

View Groups Change Log

Changes that are made to groups are logged in the Groups Change Log. You can use this log to troubleshoot unwanted changes. The log shows the username of the user who made the change, the time, and the nature of the change. The log provides a high-level summary of changes without specific details, such as the old name of a group when the name is changed. To view the Groups Change Log, a user needs the View Groups Change Log role right. By default, the Administrator and Tenant Administrator have this role right.

To view the Groups Change Log, click **Administration**, **Groups Change Log**. The Owner column specifies the username of the group creator. The Tenant column specifies the tenant under which the group was created. Customize the groups change log as follows:

- To change the Group context, click **[change]** next to 'All Groups'.
- To change the time selection, click **[change]** next to 'Last Hour'.
- To enable or disable tracking changes for groups, or to set the tracking duration, reach out to CA Support.

Organize Group Items Geographically

As an IT Manager, you have administrator-level access to NetOps Portal. You have registered a data source to monitor infrastructure usage, status, and performance enterprise-wide. You need to set up NetOps Portal groups to organize infrastructure monitoring and reporting. For example, place routers and switches in groups according to their data centers. Group managed items according to their country, region, state, or city.

Geographical Grouping Strategy

Consider a global enterprise, Mod-Lex Corporation, with the following major branch offices:

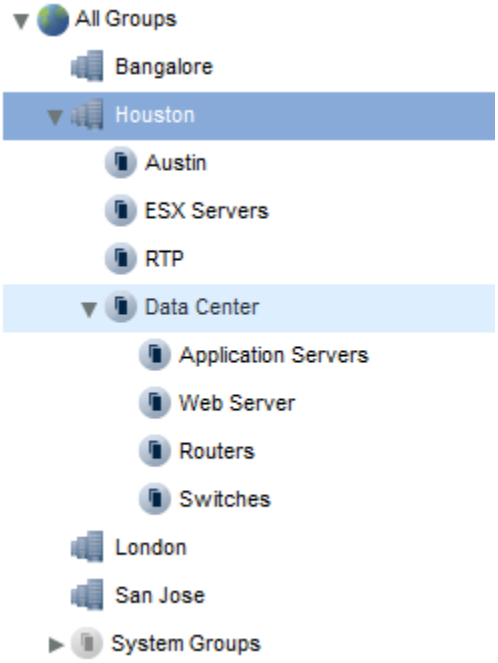
- Bangalore
- Houston, TX
- London
- San Jose, CA

Mod-Lex has a significant data center at each of these offices. Data centers support branch office operations and smaller Sales offices nearby.

Mod-Lex also has small branch offices in Austin, TX and Research Triangle Park, NC. The Austin office does not have a dedicated data center. Its three application servers are connected by a T3 line to the Houston data center. The RTP office is also connected to the Houston data center. All of its application servers are virtual machines, running on an ESX server in Houston.

Most IT staff are located at Mod-Lex corporate headquarters in London. The staff have user accounts for NetOps Portal, most with user-level access. The NetOps Portal server is running in the data center in London. Dedicated IT staff are assigned to each of the major data centers. Staff charged with monitoring Houston are also responsible for Austin and RTP.

Here is an example of a grouping strategy that takes geography and logical network structure into account:



To monitor the Research Triangle Park (RTP) site effectively, place a copy of the 'Houston\ESX Servers' group into the RTP group. Metrics from the critical application server are then reflected at the RTP group level. You can ensure that a user is monitoring that group.

Bangalore, Houston, London, and San Jose are site groups, which can contain subgroups. The Houston Data Center group contains subgroups to break out different types of infrastructure items to be monitored separately. Individual users can be granted access to data based on entire sites or on subgroups.

For example, the Network Managers for Houston, Austin, and RTP can access the entire Houston site group, or they can access the Data Center subgroup. The Austin Network Manager also needs access to the Austin group. Ensure that any Austin-specific items are added to that group. In this example, the RTP site does not have a dedicated Network Manager.

Create a Custom Group

NetOps Portal site groups are useful in a geographical grouping strategy. Site groups contain items and subgroups of items that are grouped by location. Adding site groups to other custom groups in your tree structure allows you to build geographically and logically organized reports.

Before you create groups, consider the types of access permissions that operators require to perform their monitoring duties.
Create the site groups:

- London
- San Jose
- Houston
- Bangalore

Create the custom groups:

- Austin
- RTP

To create the grouping structure, copy the Austin and RTP groups into the Houston site group as subgroups. Problems that affect users in Austin or Research Triangle Park are visible in dashboard views for the Houston site group. You can drill down into event or performance information for those specific subgroups. You can quickly pinpoint the source of the issue.

After you have created custom groups, you can populate groups by adding subgroups that contain managed items. When you copy a group, you create a group reference. You cannot modify a group reference, but you can remove it. Groups that have been copied display an extra tab in the right pane. Any changes that you make to the original group are reflected in all of its references of the group. Removing a group also deletes all of its references.

Add Managed Items to a Group

Add managed items to the subgroups in the Austin and RTP custom groups. For the Austin subgroups, add the three application servers. For the RTP custom groups, add the virtual machines that run on the ESX server in Houston.

Use Group Rules to Manage Items

Use group rules to keep the Austin and RTP groups up-to-date when systems and networks change. Newly discovered items that meet rule specifications are added to groups. Similarly, if the items do not meet rule requirements, or the items are no longer monitored, items are removed.

Mod-Lex consistently uses a geographical prefix to identify the location of devices and servers as they are purchased and brought into service. For example, the main router that is dedicated to ESX servers in Houston is named 'hou-esx-rtr'. This naming convention lets us use group rules to group items automatically.

Create a rule to populate the following parent group: 'Houston\Data Center\Application Servers'. Select the parent group in the group hierarchy to add it as a condition automatically. Add a second condition that prevents routers and switches with the 'hou' prefix in their names from being placed in the Application Servers group.

Use Groups and Group Rules to Organize Devices

You want to organize your devices into logical groups that Data Aggregator can monitor. Create a group that lets you discover and monitor your end-user switches separately. You want to monitor those switches separately because the uplink interfaces on those end-user switches have a significant impact on your network. Separate monitoring lets you report on the inventory that is related to that device collection for better troubleshooting. This scenario shows how REST web services can be used to configure device groups in NetOps Portal and Data Aggregator. This configuration

also provides the framework to write proprietary automation scripts that integrate with your third-party or proprietary management system.

> **NOTE**
> This workflow does not explain the configuration of monitoring profiles in Data Aggregator, which is done after group configuration is complete. Monitoring profiles are associated with device collections and are used to keep discovered inventory up-to-date.

To execute web service calls manually, use a REST client editor or an HTTP tool that sends requests and gets responses. In this scenario, we refer to the REST client editor to perform these procedures.

> **NOTE**
> To automate, write your own application or script that leverages the web services that are described in this documentation.

To create custom groups and group rules for organizing devices and component items using REST web services, follow these steps:

> **NOTE**
> Perform steps 1-2 using NetOps Portal REST web services. Perform step 3 using the Data Aggregator REST web services.

Create a Custom Group and Group Rules

This scenario involves creating a custom group for end-user switches only. Factory (out-of-the-box) device collections do not exist for specific switches in Data Aggregator. Instead, there is a factory device collection for *all* switches, named "All Switches". Since your uplink interfaces have significant performance impact in your network, you want to monitor these specific switches for uplink interfaces separately for better troubleshooting.

You first create a custom monitored group in NetOps Portal. Upon synchronization, Data Aggregator creates a corresponding device collection for use in device monitoring.

Add rules to your custom group so that the corresponding the device collection is kept up-to-date with discovered end-user switches. Newly discovered devices that meet rule specifications are added to device collections. Similarly, if they do not meet rule requirements or they are no longer monitored, devices are not included or removed. For this scenario, we assume that your end-user switches are easily identifiable with the word "EndUser" at the end of their descriptions.

Follow these steps:

1. Open a REST client editor that has a connection to the NetOps Portal web server.
2. Set the Content-type to application/xml.
3. Provide a valid Username and Password in the request header for a user account that has Administrator access to NetOps Portal.
4. Enter the Body text in a REST client and modify the attributes as needed:
   - **Group Name**
     Specifies a name for the group. For this scenario, type the name **My End-User Switches**.

     > **NOTE**
     > Do not use the following special characters in group names: /&\,%.
   - **Group Tree Path**
     Specifies the path where the monitored group is created. Type the path **All Groups/Collections,** which is required for the device collection to show up in the Data Aggregator administration UI.
   - **Rules**
     Specifies the rules that automatically group the devices. In this case, create a rule that groups the end-user switches that have the value "EndUser" at the end of their descriptions.
   - **type**

Indicates the type of group.

– **inherit**
Indicates whether the group includes child items of group members. In this case, set the "inherit" attribute to **true** so that the device interfaces become group members when the devices are added to the group.

Example:

```
<GroupTree path="/All Groups/Collections">
 <Group inherit="true" name="My End-User Switches" desc="" type="user group">
  <Rules allowDeletes="true" saveRules="true">
   <Rule add="Device" name="Add Devices">
    <Match>
     <Compare readOnly="true" using="MEMBER_OF">
      <Property name="ItemID" type="Device"/>
      <Value reference="/All Groups">1</Value>
     </Compare>
     <Compare readOnly="false" using="LIKE">
      <Property name="DisplayName" type="Device"/>
      <ValueList>
       <Value>EndUser</Value>
      </ValueList>
     </Compare>
    </Match>
   </Rule>
  </Rules>
 </Group>
</GroupTree>
```

5. POST the following URL:
   `POST http://pc_host:8181/pc/center/webservice/groups/false/true`
   The 'false' and 'true' values refer to the following parameters:
   – **useIds**
   Indicates that the groupId parameter is used to identify the group. In this example, the XML does not contain a group ID, so the value is 'false'.
   – **allowDeletes**
   Enables deletion of the group that you are creating.
   The My End-User Switches device collection is created. After synchronization, Data Aggregator adds all discovered end-user switches automatically to the My End-User Switches device collection.

Verify the Group Was Added to the Group Tree

Verify that the group was added to the group tree.

Follow these steps:

1. Enter the following URL:
   http://PC_host:8181/pc/center/webservice/groups/groupPath/All%20Groups%2FCollections
2. Look for the group "My End-User Switches".
   This group change in the web service corresponds to the group change in the NetOps Portal user interface.

Verify the Device Collection Appears in Data Aggregator

After automatic synchronization occurs between NetOps Portal and Data Aggregator, verify that the new device collection appears in Data Aggregator.

Follow these steps:

1. Enter the following URL: `http://da_host:8581/rest/groups`
2. Look for the device collection (group) "My End-user Switches".
   This change in the web service also appears on the Data Aggregator user interface administration pages.

You have successfully configured a custom group and group rules. You can proceed to create your own automation script with your third-party or proprietary software to populate information in Data Aggregator.

Configure Monitoring Profiles

The monitoring profile determines which metric families are collected for all devices in associated device collections and how often polling occurs. The metric families in the monitoring profiles determine which components are added to the system for devices in assigned device collections and which metrics are collected for those components.

Monitoring profiles are global across all tenants.

The relationship of monitoring profiles to device collections governs monitoring. Monitoring can be triggered in the following ways:

- A monitoring profile is associated with a device collection.
- A device is added to a device collection that has an associated monitoring profile.
- A metric family is added to a monitoring profile with an associated device collection.
- A new vendor certification is added for an existing metric family that is polled in a monitoring profile.
- The vendor certification priority changes for a metric family.
- A change occurs on a device. If **Automatically Update Metric Families** is selected, the change to monitoring occurs automatically according to the change detection rate. If **Automatically Update Metric Families** is cleared, the change to monitoring occurs when you click **Update Metric Family** on the **Polled Metric Families** tab of the **Monitored Devices** view.

Monitoring Profile Best Practices

Consider the following best practices for creating monitoring profiles:

- Before you configure monitoring profiles, see Determine Monitoring Requirements.
- Copy default monitoring profiles to create customer-specific monitoring profiles. Remove the collections that are applied to the default monitoring profiles. This model makes it easier to customize and manage device polling.

  NOTE

  Do not modify the DA Health Fast, Normal, and Slow monitoring profiles. These profiles are used for self-monitoring of the DX NetOps Performance Management components.
- For monitoring flexibility, use multiple monitoring profiles. Do not add all metric families that you monitor in your network to a single monitoring profile.
- Some metric families, such as CPU and Memory, apply broadly to all devices. You can apply a monitoring profile with these metric families to all-encompassing collections, such as All Routers or All Managed Devices.

- Monitoring profiles control the poll rate. To poll the same metric family on different devices at different rates, create monitoring profiles with different poll rates. Apply the faster poll rate to a collection that includes only the devices that require fast polling.
- Monitoring profiles control filtering. If different filtering criteria is required for different sets of devices, create a monitoring profile for each set of requirements.
- Apply only metric families that are applicable to the items in the associated collections. For example, do not apply a monitoring profile that contains the VMWare metric families to a collection of routers. Unnecessary metrics add processing to the Data Aggregator at every change detection period.

Create Monitoring Profiles

To specify which metrics to gather from which devices and components, create monitoring profiles. For example, create a custom Systems monitoring profile and configure the profile to poll resource utilization counters only from SNMP.

Monitoring profiles are available in all tenant work spaces. For example, create a "Service Router Monitoring" monitoring profile and use the profile for all tenants. You do not have to create a *separate* "Service Router Monitoring" monitoring profile for each tenant. Device collections that are associated with the monitoring profiles are tenant-scoped.

TIP

To create basic monitoring profiles, copy the out of the box monitoring profiles. Custom profiles provide flexibility.

Only Administrator users in the default tenant can modify monitoring profiles. In an MSP environment, all changes to monitoring profiles apply to all tenants.

Follow these steps:

1. Go to **Administration**, and click the Data Aggregator data source.
2. Expand **Monitoring Configuration**, and then click **Monitoring Profiles**.
3. Click **New**.
   To use an existing profile as a template, select the profile, and then click **Copy**. The copy function does not copy event rules.
4. Specify a unique profile name and description.
5. Select a poll rate.
   – Polling occurs at the fastest rate for all the monitoring profiles that are associated with a device collection.
   – Changes to the poll rate take up to two cycles to apply.
   – When you use the 60-minute rate for an existing device, a **No Data To Display** message appears in views with a time range of Last Hour. If you change the dashboard setting to a prior hour, you can see earlier data. However, the view does not display the latest data until the new poll cycle completes.

   WARNING
   One-minute polling on large device collections or devices with many components can affect the performance of the system.

   NOTE
   To poll critical interfaces at a faster rate, use a separate monitoring profile with a filter.
   For more information, see Poll Critical Interfaces Faster than Non-critical Interfaces.
6. (Optional) To detect when the components on a device change, select **Enable Change Detection**. This option is enabled by default. If you select this option, configure the behavior:
   – Specify the change detection rate. The change detection rate determines how often the system identifies changes to component items on monitored devices. If a device is associated with multiple monitoring profiles, change detection occurs as the fastest specified rate. To determine the configuration changes, DX NetOps Performance Management uses reconciliation algorithm that is defined in the metric family.

NOTE
Consider how frequently the monitored components on the devices are likely to change. Avoid setting change detection rates that are more frequent than necessary. Some metrics, such as availability, never or rarely change. Because change detection uses some system resources, do not enable change detection for such metrics.

– (Optional) To automatically start monitoring newly detected components, select **Automatically Update Metric Families**. To manually control when changes to monitoring occur, clear this option.

NOTE
Clearing this option does not automatically apply the changes to monitoring. The Events Display dashboard displays configuration events for the detected changes. To apply the changes to the devices, go to the Data Aggregator administration menu, **Monitored Devices** view, and open the **Polled Metric Families** tab. Select the appropriate metric family, and click **Update Metric Family**.

When monitoring changes are applied to devices, components that are no longer detected on the devices are marked "Not Present".

7. Assign metric families to the **Selected Metric Families** list.

TIP
By default, all metrics in each selected metric family are collected. To select a limited set of metrics, see Configure Metric Filtering.

8. Click **Save**.
The monitoring profile is applied to any assigned collections during the next poll cycle. To start monitoring with a new monitoring profile, assign the profile to device collections.

Add Event Rules to Monitoring Profiles

Event rules set conditions to trigger Threshold Violation events. Each rule is based on a single metric family and contains criteria to raise or clear the violation. The event rules apply to each component on all devices in collections that are assigned to the monitoring profile.

You cannot add event rules to factory monitoring profiles. Only Administrator users in the default tenant can modify event rules on monitoring profiles.

TIP
For best performance and granular control, configure event rules on threshold profiles instead of monitoring profiles. For more information, see Configure Threshold Profiles.

Follow these steps:

1. Select a custom monitoring profile.
2. Click the **Event Rules** tab.
3. Click **New** to create a rule, or click **Edit** to modify an existing rule.
For more information about how to create event rules, see Configure Threshold Profiles.
Events are raised and cleared when the devices in associated collections satisfy the event rule conditions.

Configure Monitoring Profile Filters

Monitoring profile filters specify criteria that governs which components are monitored. Only the component items that match the filter criteria are polled for the associated metric family. Filtering limits SNMP traffic and ensures that the system monitors only relevant components. The filters of each monitoring profile are assessed independently.

WARNING
DX NetOps Performance Management monitors device components if any of the monitored components pass the filter criteria of any associated monitoring profile.

NOTE
Monitoring profile filters apply after discovery. The system creates the component items but does not send SNMP requests to monitor the items. Monitored components are reassessed according to the change detection interval. *Vendor certification filters* prevent the creation of component items that do not match the filter criteria. Vendor certification filters might not recognize changes at the component level. Both filter types reduce SNMP traffic to the same level. Use vendor certification filters only when the monitoring profile filter cannot filter component item.

Example: Multi-Rate Polling

You want to monitor all interfaces with 5-minute polling, and one type of interface with 1-minute polling. One monitoring profile has a poll rate of 5 minutes. This monitoring profile is associated with a device collection. A second monitoring profile has a poll rate of 1 minute. The interface metric family on that profile has a filter that specifies the type of interface to poll at a 1-minute interval. This monitoring profile is also associated with the same device collection. DX NetOps Performance Management polls the specified interface type at a 1-minute rate and all other interfaces at a 5-minute rate.

Example: How and When Filters Are Combined

You want to apply multiple filter profiles to multiple devices. You want to exclude an attribute from one of the filters. However, the product Poll filters are not *exclusion* filters. They are *inclusion* filters. The product behavior is to poll as much as you can, as fast as you can. So, if two filter criteria conflict, the product *includes* interfaces in the polling rather than *exclude* them. The following example explains how filtering works and how you can achieve your goal.
For example, you have two filters with the following configuration:

- **Filter 1**:
  (Description contains "Virtual-Access" or Description contains "Tunnel")
- **Filter 2**:
  Name does not contain "Internal"

This situation means that Filter 1 and Filter 2 are OR'ed. Only components that meet the previously mentioned criteria are polled.
However, if you want to *exclude* any interfaces that contain "Internal" in the Name attribute, add that filter criteria to each monitoring profile as an AND. Filter 1 should appear as follows:

- **Filter 1**:
  (Description contains "Virtual-Access" or Description contains "Tunnel") AND (Name does not contain "Internal)

When you have the same devices that are associated to two monitoring profiles, the two profiles must have filter criteria. In this scenario, effectively, you combine the two filters by stating exclusion criteria in both. Otherwise, the monitoring profile without the filter causes all components on the device to be polled. Therefore, Filter 2 should appear as follows:

- **Filter 2:**
  (Description contains "Virtual-Access" or Description contains "Tunnel") AND (Name does not contain "Internal)

Configure Profile Filters

Only Administrator users in the default tenant can add or edit monitoring profile filters.

Follow these steps:

1. Select a custom monitoring profile.
2. In the Metric Families tab, select a metric family.
3. Click **Edit Filter**.
4. Click the existing AND condition, then click a logic button, such as New AND or New OR.
5. Select an attribute and an operation, and specify a value for the filter condition.
6. Click **Add Condition**.
7. Create all required filter conditions.

8. Click **Save**.
   The filter applies to the selected metric family on the next poll cycle.
   When you view metric families the assigned filters, an asterisk (*) appears next to each metric family that does not have any assigned filters.

   NOTE
   To clear a filter, select the metric family, and click **Clear Filter**. The filter is removed on the next poll cycle.

Associate Monitoring Profiles and Collections

To apply the polling behavior of a monitoring profile to devices, associate the profile with a device collection. When multiple monitoring profiles are associated with a device collection, Data Aggregator uses the fastest specified poll rate.

WARNING

Do not associate monitoring profiles with the All Devices device collection. Doing so can cause extra SNMP requests being made to pingable-only devices, and can result in sporadic metric family support.

Metric families such as QoS, MPLS, and various Response Path Test metric families can contribute to significant SNMP requests. For more information about the implications and restrictions of SNMP requests to your network devices, see the vendor documentation or contact the vendor.

Administrators and Tenant Administrators can modify associations between monitoring profiles and device collections. Administrators see only the associated device collections for the current tenant.

Assign Collections to Monitoring Profiles

Follow these steps:

1. From the **Monitoring Profiles** page, select the monitoring profile, and click the **Collections** tab.
2. Click **Manage**.
3. Select collections for the **Assigned Collections** list, and click Save.
   The monitoring behavior that is defined in the monitoring profile is applied to the assigned device collections.

Assign Monitoring Profiles to Collections

Follow these steps:

1. From the **Collections** page, select a collection, and click the **Monitoring Profiles** tab.
2. Click **Manage**.
3. Select monitoring profiles for the **Assigned Monitoring Profiles** list, and click **Save**.
   The monitoring behavior that is defined in assigned monitoring profiles is applied to the device collection.

Factory Monitoring Profiles

Factory monitoring profiles let you start monitoring devices in the environment quickly. Factory monitoring profiles are automatically associated with factory device collections. For example, the Router monitoring profile is associated with the All Routers device collection.

NOTE
You cannot edit or delete factory monitoring profiles.

The following factory monitoring profiles can have a significant performance impact:

- CBQoS
- MPLS
- Network Interface
- Response Path

Configure Threshold Profiles

Threshold profiles let users monitor specific devices or components, such as interfaces and IPSLA tests. Different components often require different threshold configuration. Creating threshold profiles lets users determine a threshold for a given metric or set of metrics. Failing to meet the threshold produces a violation event, and returning to acceptable operation produces an event which indicates that the violation has been cleared.

Example

You want to monitor the utilization of an interface, and trigger a violation when the utilization is above 75 percent. When the utilization drops below 75 percent, you want the violation to clear.

- **Event Rules**
  Event rules let users define logic, which uses metric data to determine when a violation of a threshold occurs. Each event rule uses metrics from a single metric family, and various operators that compare the threshold to the actual value. Up to five conditions can be added to an event rule, and a violation is produced when the threshold is exceeded. In the example on this page, at least two rules are needed. One rule determines when the violation occurs, while the other determines when it is cleared.
- **Event Condition**
  Event conditions automatically appear in a view when all the conditions in a certain event rule are met. Conditions include "Violation" and "Cleared".

The following video shows the threshold profile configuration process:

Threshold Best Practices

Consider the following best practices when you configure thresholding:

- Apply threshold profiles to groups with specific components instead of devices. This group structure increases the granularity and flexibility of thresholding.
- Expand threshold monitoring slowly. Start with a small group of components and verify that the monitoring engine does not become degraded. For more information, see Threshold Event Processing Self-Monitoring Metrics.
- Thresholds on components with 1-minute polling have a high resource cost to the system.
- Threshold evaluations might slow down after a Data Aggregator restart while cached poll data from the the Data Collectors is processed.

Create Threshold Profile

Log in as a user or administrator to create a threshold profile. Users with the Create DA Threshold Profile or the Administer DA Threshold Profile role right can create threshold profiles.

You can edit or delete exiting threshold profiles.

Follow these steps:

1. Hover over **Administration**, **Data Sources**, and click the data source.
2. Expand **Monitoring Configuration**, and click **Threshold Profiles**.
3. Create a folder, or select an existing folder.
4. Click **New Profile**.
5. Specify the required information.
6. (Administrator Only) Select an owner. Only the owner or a user with the Administer DA Threshold Profile role right can edit the profile.

7. Add event rules to the profile.
8. Click **Save**.
   The threshold profile is added to the system. To generate events, assign the profile to a group.

Add Event Rules

Each event rule is based on a single metric family, and determines the conditions that cause or clear a violation. Each threshold profile requires at least one event rule.

You can edit or delete existing event rules.

> **TIP**
>
> To use an existing event rule as a template, select an event rule, and click Copy.

Follow these steps:

1. Create or edit a threshold profile.
2. In the Event Rules pane, click **New**.
3. Specify the required information for the event rule. The following fields require explanation:
   - **Duration**
     Specifies the total amount of time a given condition must be true within the specified Window to generate an event. The poll cycles that trigger the condition do not need to be consecutive.
   - **Window**
     Specifies the overall range of time to evaluate the rule condition.
   - **Aggregation**
     Specifies whether the threshold applies to an aggregate value of all components for the device. This field appears only when you select a supported metric family.

     > **NOTE**
     >
     > Currently, only the Utilization (%) Metric for the CPU and Memory metric families are supported for aggregation. When you select this option, the event rule must use Fixed Value for the Condition Type.
4. Save the event rule.
5. Save the threshold profile.

Duration and Window Example:

A monitored device has a poll rate of 5 minutes. An associated threshold profile has an event rule with a duration of 600 (10 minutes) and a window of 3600 (1 hour). An event does not occur when the conditions are triggered for a single poll result because the 5-minute poll does not reach the 10-minute duration. The event occurs only if the conditions of the rule are triggered for a second poll result within one hour of the first triggering poll.

> **NOTE**
>
> When a threshold is breached, an alarm is created. When the event clears, the threshold is rechecked with the next poll cycle. If the threshold is breached again, a new alarm is created.

Standard Deviation Event Conditions

Event rules that use standard deviation compare the poll results to the baseline for the device or component. The baseline and the standard deviation value are calculated for the specific hour of the day of the week. For more information about these calculations, see Baseline Calculations.

Standard deviation rules are triggered when the value of the metric differs from the baseline by the specified number of standard deviations. For rules with the Above operator, the rule is triggered when the value of the metric exceeds the baseline value plus the number of standard deviations. For rules with the Below operator, the rule is triggered when the value of the metric is lower than the baseline value minus the number of standard deviations.

Example

The baseline is 65% and the standard deviation is 10%. The rules states that an event triggers when CPU utilization is above 2 standard deviations. This condition triggers when the CPU utilization is greater than 85%.

Percent of Baseline Event Conditions

Event rules that use Percent of Baseline compare the poll results to the calculated baseline plus or minus a percentage of the calculated baseline for the device or component. An event is triggered when the qualifying poll data meets the criteria that are specified for a Percent of Baseline event rule condition. Percent of Baseline event conditions are useful when there is a lot of or very little variation in the metric values. Consider using Percent of Baseline conditions when the standard deviation is above 3 or extremely low like 0.1 or 0.0.

Examples

The calculated baseline is 60 degrees and the specified Percent of Baseline is 50%. The rule states that an event triggers when the temperature rises above 50% of the baseline. This condition triggers when the temperature is higher than 90 degrees.

Math: 60 + (+50%*60) = 60 + 30 degrees = 90 degrees

The calculated baseline is 60 degrees and the specified Percent of Baseline is -50%. The rule states that an event triggers when the temperature falls below -50% of the baseline. This condition triggers when the temperature is lower than 30 degrees.

Math: 60 + (-50%*60) = 60 - 30 degrees = 30 degrees

Assign Groups to Threshold Profile

To generate violation events, associate groups to the threshold profile. Assigning groups to a profile identifies the devices or components that the profile monitors. When a device is in the group, the threshold profile applies to each component of the device that supports the selected metric family.

> **NOTE**
>
> Threshold profiles that are assigned to collections apply the event rule only to devices in the collection. Components and interfaces in the collections are not analyzed. Use custom groups to generate events for components and interfaces.

Follow these steps:

1. Select a threshold profile from the Folder View or Table View.
2. Click the **Groups** tab in the right-hand pane.
3. Click **Manage** at the bottom of the screen.
4. Select the groups from the **Available Groups** tree, and click the right arrow to add it to the Selected list.
5. Click OK.
   The groups in the Selected list are assigned to the threshold profile.

View Threshold Profile Events

Event rules that you create trigger events when a threshold is violated and when the violation is cleared. The Threshold Profiles screen shows events that have occurred as a result of specific event rules. In contrast, the Events Display dashboard shows threshold events that relate to all event rules in all threshold profiles.

Follow these steps:

1. (Administrator) Select **Administration**, **Threshold Profiles**.
   (User) Click the name of your user account in the upper-right corner and click **Manage Threshold Profiles**.
2. Select a threshold profile.

3. Click the **Events** tab.
4. (Optional) Click the **Details** button.
5. (Optional) Click **Change** next to the time range, and select a default time range.
   You can also select **Custom Time Range** to set a different time range.

Create Custom Attributes

You can add custom attributes to items such as devices or interfaces. Populate these items using REST Web Services. Custom devices and interfaces are then available in reports, dashboards, and group rules. Custom devices, device components, and interfaces are also available for OpenAPI queries with some delay due to ETL job scheduling.

Custom attributes for devices and interfaces appear in the Information section of the Details tabs in a context page for an item. For more information, see Context Pages.

> **WARNING**
>
> After custom attributes are created, you cannot delete them.

Define Custom Attributes

To define custom attributes for items such as devices or interfaces, POST the XML using a REST client.

By default, the maximum number of custom attributes is 5 for string and 5 for integer for each supported item type. If you attempt to exceed the maximum number of attributes, the Data Aggregator returns an error.

Follow these steps:

1. Set up a REST client with a connection to the Data Aggregator server.
2. Specify the following URL:
   `http://da_hostname:8581/rest/customattributedefinition`
3. POST the XML for defining your custom attributes.
   The following example defines a **CustomerID** attribute for all devices.

```
<CustomAttributeDefinition version='1.0.0'>
   <Label>Customer ID #</Label>

   <Description>Customer account number</Description>

   <Hidden>true</Hidden>

   <Storage>

      <AttributeName>CustomerID</AttributeName>
      <Type>String</Type>
      <ItemType>Device</ItemType>

   </Storage>

</CustomAttributeDefinition>
```

- **<Label>**Specify a label that can be displayed in the NetOps Portal user interface for the custom attribute.**Limit:** 64 characters
- **<Hidden>**Specify whether to hide the property by default from the NetOps Portal user interface. If this tag is set to "true," you can manually display the custom attribute in the user interface. This tag impacts whether custom attributes appear by default in the NetOps Portal user interface only. They are available by default in other areas, such as the OpenAPI QueryBuilder.
  Default: "true"

  NOTE

  If this tag is set to "true", custom attributes are hidden from context pages and cannot be manually shown in the user interface. If you want custom attributes to appear on context pages, you must set this tag to "false."
- **<Description>**Specify a description for the attribute.
- **<AttributeName>**Specify an internal reference name for the custom attribute. Use this name to set a value for the custom attribute in the following procedure.
  Limit: 26 characters
- **<Type>**Specify whether the custom attribute is a string or an integer.**Valid entries:** "String" and "Integer"
- **<Item Type>**Specify whether the custom attribute is for a device, component, or interface ("Port")**.Valid entries:** "Device", "DeviceComponent", and "Port"

  NOTE

  Custom device component attributes are available only for OpenAPI queries with some delay due to ETL job scheduling. They are *not* available in reports, dashboards, and group rules.

The POST returns an ID for the definition of the custom attribute.

Set Values for Custom Attributes

To set values for your custom attributes, PUT the XML using a REST client.

Follow these steps:

1. Specify one of the following URLs:
   - `http://da_hostname:8581/rest/devices/customattributes/itemID`
   - `http://da_hostname:8581/rest/devices/components/customattributes/itemID`
   - `http://da_hostname:8581/rest/ports/customattributes/itemID`
     - **devices**
       If the custom attributes are for devices, specify this item type in your URL.
     - **devices/components**
       If the custom attributes are for device components, specify this item type in your URL.
     - **ports**
       If the custom attributes are for interfaces, specify this item type in your URL.
     - ***itemID***
       Specify the device or interface ID.
2. PUT the XML for setting your custom attributes.
   String attributes support up to 255 characters. Integer attributes support up to 64-bit integers.
   The following example sets the **CustomerID** attribute from the previous example.

```
<DeviceCustomAttributes version='1.0.0'>
     <CustomerID>CustomAttributeValue</CustomerID>
</DeviceCustomAttributes>
```

The following example sets the custom attributes for a device component:

```
<DeviceComponentCustomAttributes version='1.0.0'>
     <AttributeName>CustomAttributeValue</AttributeName>
</DeviceComponentCustomAttributes>
```

The following example sets the custom attributes for a port:

```
<PortCustomAttributes version='1.0.0'>
     <AttributeName>CustomAttributeValue</AttributeName>
</PortCustomAttributes>
```

Change the Maximum Custom Attributes

By default, the maximum number of custom attributes is 5 for string and 5 for integer for each supported item type. However, you can increase or decrease these values.

TIP

Consider the performance impact of increasing the maximum number of custom attributes for each supported item type. For example, the NetOps Portal user interface allows only a defined number of custom attributes to display. We recommend that you avoid exceeding a maximum of 30 attributes total for string and integer combined.

Follow these steps:

1. Create the following file:
   DA_install_directory/apache-karaf-*version*/etc/
   com.ca.im.item.custattr.CustAttrColumnCache.cfg
2. Add new limits.
   Example:

```
MaxStringColumnsPerItemType=7

MaxIntegerColumnsPerItemType=2
```

   A message appears logged in the `karaf.log` file.
   Example:

```
INFO | AttrColumnCache) | 2016-07-29 09:46:34,464 | CustAttrColumnCache |
 tem.custattr.CustAttrColumnCache 80 | ository.webservices.impl | | Max Integer
 custom attributes per item type:2

INFO | AttrColumnCache) | 2016-07-29 09:46:34,465 | CustAttrColumnCache |
 tem.custattr.CustAttrColumnCache 93 | ository.webservices.impl | | Max String custom
 attributes per item type:7
```

Configure Business Hours Filtering

Business hours represent the times of the day and week during which business is normally conducted. These hours are relevant for capacity planning. You can filter data to these hours for investigation purposes by applying a business hours filter to a dashboard or view.

Business hours definitions apply to views that report data from the data aggregator. They apply to all the devices and components in a site group. Site groups are custom groups that are based on physical locations, such as a city, region, office, or campus. They include Time Zone and Business Hours parameters so that you can precisely filter data from business-critical times of day. When you select the associated group in the context, business hours apply to all applicable views as you navigate between dashboards. The subtitle of each view indicates whether business hours apply to the view.

Alternatively, you can apply a business hours filter to a custom view.

For more information, see Customize Views.

You can also apply a business hours filter to a view on a context page.

For more information, see Context Pages.

Subgroups do not directly inherit business hours and time zone filters from site groups. Associate the business hours definition with each relevant subgroup. However, when rendering views, these filters apply to all items based on the selected site group. The filters of the selected site group apply to all items in that group and in any subgroups. When you change the selected site group to a subgroup, the filters of the parent group are not applicable.

Reference groups inherit associated business hours and time zone filters from the original site group.

In this article:

Business Hours Support and Limitations

Business hours filtering can apply to the data in the following view types:

- Gauge chart
- Pie chart
- Table grid

> **NOTE**
> You can apply a business hours filter to the data in a group scorecard trend view only when the resolution is As Polled or Hourly. The subtitle on the view indicates whether the filtering is applied.

Trend views and heat charts apply shading to show business hours. Shading appears only in the following circumstances:

- A site group with a business hours filter is specified at the page level.
- A site group with a business hours filter is specified at the view level.
- A business hours time filter is specified at the view level.

> **NOTE**
> The data in these views is not limited to the business hours.

You cannot apply business hours filters to the following metrics, data, views, dashboards, and charts:

- Baseline and projection metrics
- Daily rollup data and any view that shows daily resolution
- Table views in On-Demand reports
- Dashboards with views created with a device or interface context
- Trend charts where the option to enable events is selected

Configure a Business Hours Definition

Before you can apply business hours filters to views, create a business hours definition and associate the business hours with a site group. Each definition includes the times of day and days of the week. Select the hours and days that reflect periods of increased activity. Create definitions for every distinct location in your enterprise.

Applying a business hours filter that is associated with a site group to a view applies the filter to that group. Otherwise, apply the business hours filter to the view by associating the business hours definition with a site group.

Only users with the Administer Business Hours role right can add, edit, or delete business hours.

Follow these steps:

1. Hover over **Administration**, and then click **Business Hours**.
2. Click **New**.
3. Specify a name and description.
   The name appears when you associate business hours with site groups. The description appears only on the Manage Business Hours Definitions page.
4. Select the days of the week to include.
5. Select the start time and end time for the business hours from the drop-down lists.
   Half-hour increments are not supported. The same hours are applied to all selected days.
6. Associate the business hours definition with a site group:
   a. Click **Select Site Groups to Associate**.
   b. Add one or more site groups to the **Selected Sites** list.

      NOTE
      You can select only site groups that already have a time zone. Because you can apply business hours filters only in whole hour increments, only site groups that have time zones with only whole hour offsets are available for selection.
      You can assign time zones and business hours when you create or edit site groups.

   c. Click **OK**.
7. Click **Save**.

Schedule Maintenance Indicators

Maintenance indicators represent times when maintenance is occurring. After you schedule maintenance indicators, views indicate maintenance with shading.

Maintenance indicators apply to all the devices and components in a site group. When the associated site group is selected in the context, maintenance indicators appear in applicable views as you navigate between dashboards. The subtitle of each view indicates whether maintenance indicators apply to the view.

Subgroups do not directly inherit maintenance indicators from the site groups. Associate the maintenance indicators with each relevant subgroup. However, when rendering views, these filters apply to all items based on the selected site group. The filters of the selected site group apply to all items in that group and in any subgroups. When you change the selected site group to a subgroup, the filters of the parent group are no longer applicable.

Reference groups inherit associated maintenance indicators from the original site group.

Maintenance Indicators Support and Limitations

Maintenance indicators can apply to the data in the following view types:

- Trend charts
- Calendar heat charts

Trend views and calendar heat charts apply shading to show maintenance indicators. Shading appears only in the following circumstances:

- A site group with maintenance indicators is specified at the page level.
- A site group with maintenance indicators is specified at the view level.

NOTE

The data in these views is not limited to the maintenance indicators.

Maintenance indicators are not supported for the following cases:

- Maintenance hours longer than 24 hours
- Daily rollup data and any view that shows daily resolution
- Table views in On-Demand reports
- Views on context pages

Schedule Maintenance Indicators

To apply maintenance indicators to views, create maintenance indicators and associate the maintenance indicators with a site group. Each definition includes the times of day and day of the week. Select the hours and day that reflect maintenance periods. Create definitions for every distinct location in your enterprise.

Only users with the Administer Maintenance Indicators role right can add, edit, copy, or delete maintenance indicators.

1. Select **Administration**, and click **Maintenance Indicators**.
2. Click **New**.
3. Specify a name and description.
   The name appears on views displaying maintenance indicators. The description appears only on the Manage Maintenance indicators page.
4. Select the days of the **Maintenance Date**.
5. Select the start time and end time for the maintenance indicators from the drop-down lists.
   Half-hour increments are not supported. The same hours are applied to all selected days.
6. Add one or more site groups to the **Selected Sites** list.

   NOTE

   You can only select site groups that already have a time zone. The Maintenance Indicators feature currently supports filtering only in whole hour increments. Therefore, the feature offers only whole hour offsets. Time zones with offsets of 30 or 45 minutes are unsupported.
7. Click **Save**.
   If you associated the maintenance indicators with a site group, the maintenance indicators apply to that group.

Manage Devices

You can view details for monitored devices and can view their associations with device collections, components, monitoring profiles, and metrics. You can also view a Filter Report. This information helps you see information in context, such as which monitoring profiles are being used to poll device components.

NOTE
Some features require administrator privileges.

Monitored devices are manageable or pingable (accessible). Inaccessible devices are not monitored devices.

The following device types are supported:

- **Routers**
- **Switches**
- **Servers**
- **Pingable Devices**

In this article:

Manage the Device Details

To do any of the following tasks, go to the Details tab of the Monitored Devices page:

- View and edit device details.
- Rediscover a device.
- Reconfigure components for any configuration updates.

Follow these steps:

1. Hover over **Administration**, and then click **Monitored Items Management: Monitored Devices**.
2. Do one of the following tasks and select a specific device:
   - On the Tree View tab, select **Device by Collection** or **Device by Monitoring Profile** from the drop-down. Select a specific device from the corresponding tree view.
   - On the Search tab, search by host name, device name, or IP address. You can enter a partial name or IP address to return a list of devices that contain that partial match. Wildcards and regular expressions are not supported.
3. Select the **Details** tab. View details for the selected monitored device.
4. Edit any of the following fields:
   - **IP address**
     For more information, see Change the Primary IP Address for a Device.
   - **Data Collector host**
   - **SNMP profile**
   - **SNMP version for the device**
5. To rediscover the device, click **Rediscover**. The following set of attributes can be updated as a result of this discovery:
   - System name
   - Hostname
   - Device type (as it appears in CA NetOps Portal)
   - Location
   - Vendor
   - Device description
   - Device model

   NOTE
   You cannot rediscover CAMM devices.
6. To verify that the device was rediscovered, look for the event that the rediscovery triggered.
7. To pause polling of the device, click **Stop Polling**. Automatic change detection is disabled. You can manually update metric families or run the rediscovery.

   NOTE
   You cannot stop the polling of CAMM devices.
8. To resume polling, click **Start Polling**.
9. To reconfigure components for any configuration updates, see Device Reconfiguration.

View the Polled Metric Families for a Device

To view the components of monitored devices, go to the **Polled Metric Families** tab of the **Monitored Devices** page. View the total set of metric families that are polled on a device and their poll rates. This total set is based on the consolidation of all the monitoring profiles on the device.

Follow these steps:

1. View the Polled Metric Families table. This table includes the following columns:
   - **Metric Family**
     View the metric families polled on the device.
   - **Vendor Cert**
     View the vendor certifications that are used for each metric family. When Vendor Cert Priority Grouping is occurring on a device, more than one row appears on this tab. One row appears for each vendor certification that is used for a metric family.
   - **Status**
     View whether the device supports the metric family.
   - **Last Discovered**
2. View the Components table. This table shows the polling status on the components for a metric family component that was previously discovered. One of the following values displays in the Status column:
   - **Active**
     The component is being polled.

     NOTE
     CAMM devices show Not Polled in the SNMP Poll Rate column. DX NetOps Performance Management does not poll CAMM devices.
   - **Inactive**
     Polling has stopped on the component because the metric family is no longer monitored for the device.
   - **Not Present**
     The component no longer exists on the physical device. Polling is stopped on the component. You can view historical data for reporting purposes. By default, these components are not synchronized with CA NetOps Portal. To enable this option, select **Synchronize items that are no longer present** in the **Edit Data Source** dialog on the **Manage Data Sources** page in NetOps Portal.
3. To reconfigure components for any configuration updates, see Device Reconfiguration.

View More Device Details

To view more device details, go to one or more of the following tabs:

- **Monitoring Profiles**
  Select a device collection to view the associated profiles names. Hover over a profile to see the description.
- **Threshold Profiles**
  View the threshold profiles that are applied to the selected device due to the groups to which the device belongs.
- **Metrics**
  View a list of metrics that this device supports. Select a metric family to view the following details:
  - The backing vendor certification.
  - The vendor source (the MIB table source is displayed if it is an SNMP vendor certification).
  - Whether the metric is collected.
  - The expression that is used to calculate each metric.
- **Filter Report**
  View which interface filter criteria were used during the component monitoring. The tab also shows a report of all of the interfaces that are identified on the device. The tab shows whether the interfaces matched the specified filter criteria. If you change the rules on a custom monitoring profile, the Interface Filter Criteria pane does not reflect those changes.

If you disassociate the monitoring profile from a group, the Interface Filter Criteria pane does not reflect those changes. Rediscover the device to filter the interfaces using the changes made to the filter criteria and monitoring profile.

- **Events**
  View the events that have occurred on the selected device.

Change the Primary IP Address for a Device

The primary IP address is the IP address that DX NetOps Performance Management uses to monitor a device. When a device is first discovered with the IP ranges discovery profile, DX NetOps Performance Management tries to use the IP address that maps to the hostname as the primary IP address.

If the primary IP address on a devices item changes, an event is generated on that device item.

Automatic Change Detection

By default, when the IP address that maps to a hostname on the physical device changes, the primary IP on the device item does not change. To update the primary IP automatically, enable IP change detection. DX NetOps Performance Management tries to detect the IP address change only after two consecutive polling failures. The data collector uses a reverse hostname lookup to find the new IP address.

NOTE

If your DNS is not up to date, automatic change detection may cause errors.

Use the following REST to enable or disable IP change detection:

URL: http://{da-host}:8581/rest/discoverydefaultconfig/{itemID}**Method:** PUT
Body:

- **Enable:**

```
<DiscoveryDefaultConfig  version="1.0.0">

   <DetectIPChange>true</DetectIPChange>

</DiscoveryDefaultConfig>
```

- **Disable:**

```
<DiscoveryDefaultConfig  version="1.0.0">

   <DetectIPChange>false</DetectIPChange>

</DiscoveryDefaultConfig>
```

Change the Primary IP Address

Situations may arise when you want to change the primary IP address of a device. For example, you want to change the hostname IP address to the loopback IP address, or the IP address is no longer reachable on the device.

Follow these steps:

1. Click Monitored Devices from the Monitored Inventory menu for a Data Aggregator data source.
   The Tree View tab displays.

2. Select **Device by Collection** or **Device by Monitoring Profile** from the drop-down list, and select a specific device.

   NOTE
   Alternatively, select the Search tab to search by host name, device name, or IP address. You can enter a partial name or IP address to return a list of devices that contain that partial match. Wildcards and regular expressions are not supported.
3. Change the primary IP address by taking one of the following steps:
   – Edit the IP Address field and then click Save.
   – Right-click an IP address in the IP Addresses table, select **Set this IP as the device's primary IP**, and click Save.

   The primary IP address is changed.

Change the Primary IP Address Through REST

If you change the primary IP address of a device to an IP address that another devices uses, the REST call returns and error.

To change the primary IP address of a device with a REST client or HTTP tool, enter the following criteria:

- **URL:** http://*da-hostname*:8581/rest/devices/*deviceitemID*
  deviceitemID specifies the target device.
- **Method:** PUT
- Enter the changed primary IP address in the Body tab of the HTTP Request pane.
  For example:

```
<Device version="1.0.0">

      <PrimaryIPAddress>IP</PrimaryIPAddress>

</Device>
```

- **IP** specifies the new primary IP address of the device.

Example: In this example, you change the primary IP address on the device to 1.2.3.4:

```
<Device version="1.0.0">

      <PrimaryIPAddress>1.2.3.4</PrimaryIPAddress>

</Device>
```

Delete Components That Are Not Present

Data Aggregator includes a script to delete components that are not present. These components are components that no longer exist on physical devices. The presence of excessive numbers these components can impact user interface performance.

Follow these steps:

1. Open a command prompt and access the /opt/IMDataAggregator/scripts directory.
2. Type the following command:
   `./remove_not_present_items.sh`
   The script usage is described.

NOTE

- If you filter the components by IP domain name or IP domain ID, also specify a specific IP address to return correct results.
- If your filter criteria return too many components, the REST interface does not return a response. Use other filtering options to narrow the results. More filter criteria are available at http://*hostname*:*port*/rest/retired/xsd/filterselect.xsd.

Automate the Removal of Components That Are Not Present

As an administrator, you can automate the removal of components that are not present from your network. For example, you can set up a weekly cron job to delete the components that are a month old.

The remove_not_present_items script that is included with Data Aggregator is comprised of two parts. The first part of the script identifies and returns data about the components, which is based on the filter that you set. The second part of the script issues the delete of the component list. To automate the process, understand how this script was built.

Example: Filter the List of Components By a Device IP Address

In this example, you want to find all of the components for a device that has a primary IP address of 10.252.1.1. Filtering by IP address is a two-step process because no direct component filter by IP address is available. To filter the components, first make note of the IP address for the device that the components are associated with. With the IP address information, you will determine the device item ID for the device. Then, using the device item ID, you will determine what the components are. Finally, you delete the components.

NOTE
This example uses the curl command, but you can use any command that you are familiar with.

1. Create the filterDeviceIP.xml file. You will use this file to return the device item ID for the device that has a primary IP address of 10.252.1.1. The file must look like the following example:

```
<FilterSelect xsi:noNamespaceSchemaLocation="filter.xsd" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance">
  <Filter>
    <And>
     <Device.PrimaryIPAddress type="EQUAL">10.252.1.1</Device.PrimaryIPAddress>
    </And>
  </Filter>
</FilterSelect>
```

2. Run the following command:
   curl -X http://*hostname*:*port*/rest/devices/filtered -H "Content-Type: application/xml" -T "filterDeviceIP.xml" > returnedDeviceID.xml
   - **-X**
     Creates the filter that you indicate.
   - ***hostname*:*port***
     Specifies the Data Aggregator host name and the port number.
     Default port: 8581
   - **-H**
     Indicates the content type of the file that you are posting.
   - **-T**
     Indicates the file that you are posting.

   The following result is returned as an HTTP response:

```
<?xml version="1.0"?>
<DeviceList>
```

```
 <Device version="1.0.0">
   <ID>107881</ID>
   <PrimaryIPAddress>10.252.1.1</PrimaryIPAddress>
   <supportsOnDemandMFDiscovery>true</supportsOnDemandMFDiscovery>
   <SupportedProtocolsList>
     <SupportedProtocols>ICMP</SupportedProtocols>
   </SupportedProtocolsList>
   <DiscProfileID>107503</DiscProfileID>
   <HostName>rtp003723rts.ca.com</HostName>
   <RelatesTo>
     <MonitoredGroupIDList relatesURL="relatesto/monitoredgroups"
rootURL="monitoredgroups">
       <ID>509</ID>
     </MonitoredGroupIDList>
     <GroupIDList relatesURL="relatesto/groups" rootURL="groups">
       <ID>547</ID>
       <ID>530</ID>
       <ID>509</ID>
     </GroupIDList>
   </RelatesTo>
   <IsAlso>
     <IsA name="MetricFamilyDiscoveryHistory" rootURL="devices/mfdiscoveryhistory"/>
     <IsA name="AccessibleDevice" rootURL="devices/accessible"/>
     <IsA name="Syncable" rootURL="syncable"/>
     <IsA name="IPDomainMember" rootURL="ipdomainmember"/>
   </IsAlso>
   <DataCoectionMgrId version="1.0.0">
     <DcmID>dcname.ca.com:8f53bc55-f442-42fc-9bd5-a907d0261421</DcmID>
   </DataCollectionMgrId>
   <Syncable version="1.0.0">
     <SyncID>-1</SyncID>
   </Syncable>
   <Item version="1.0.0">
     <DisplayName>router.ca.com</DisplayName>
     <CreateTime>Wed Feb 05 10:20:26 EST 2014</CreateTime>
     <Name>router.ca.com</Name>
   </Item>
   <IPDomainMember version="1.0.0">
     <IPDomainID>2</IPDomainID>
   </IPDomainMember>
   <DeviceMonitoringProfile version="1.0.0">
     <ConsolidatedMonitoringProfile>2509</ConsolidatedMonitoringProfile>
   </DeviceMonitoringProfile>
 </Device>
</DeviceList>
```

Device item ID 107881 is returned. The results also display detailed information about the device.

3. Create the filterNotPresent.xml file. You will use this file to return the components that are not present and are associated with the device whose device item ID is 107881. This file must look like the following example:

```
<FilterSelect xsi:noNamespaceSchemaLocation="filter.xsd" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance">
  <Filter>
    <And>
        <DeviceComponent.DeviceItemID type="EQUAL">107881</
DeviceComponent.DeviceItemID>
    </And>
  </Filter>
  <Select use="exclude">
    <Item use="exclude">
      <DisplayName use="include"/>
    </Item>
  </Select>
</FilterSelect>
```

4. Run the following command:
   curl -X post http://*hostname*:*port*/rest/retired/filtered -H "Content-Type: application/xml" -T "filterNotPresent.xml" > returnedNotPresentItems.xml
   The following result is returned as an HTTP response:

```
<?xml version="1.0"?>
<RetiredList>
  <Retired version="1.0.0">
    <ID>128452</ID>
    <Item version="1.0.0">
      <DisplayName>GigabitEthernet0/239 - GigabitEthernet0/239</DisplayName>
    </Item>
  </Retired>
  <Retired version="1.0.0">
    <ID>128451</ID>
    <Item version="1.0.0">
      <DisplayName>GigabitEthernet0/238 - GigabitEthernet0/238</DisplayName>
    </Item>
  </Retired>
</RetiredList>
```

   Two components that fit the filter criteria are returned. The item IDs for the components are 128452 and 128451.
5. Create the deleteNotPresentList.xml file. You will use this file to delete the returned list of components. The file must look like the following example:

```
<DeleteList>
  <ID>128452</ID>
  <ID>128451</ID>
</DeleteList>
```

6. Run the following command:
   curl -X post http://*hostname*:*port*/rest/retired/deletelist -H "Content-Type: application/xml" -T "deleteRetiredList.xml" > deletelistreponse.xml
   The following result is returned as an HTTP response:

```
<?xml version="1.0"?>
```

```
<DeleteListResult>
  <DeleteResult>
    <ID>128452</ID>
    <Error>SUCCESS</Error>
  </DeleteResult>
  <DeleteResult>
    <ID>128451</ID>
    <Error>SUCCESS</Error>
  </DeleteResult>
</DeleteListResult>
```

The components are successfully removed.

Example: Delete a Large Number of Retired Components

You can easily delete all of the retired components where the total exceeds 100,000:

- To review all of the retired components where the total exceeds 100,000, type the following command:
  ```
  ./remove_not_present_items.sh -h host_name  -o ouputfile
  ```
  – **-o *outputfile***
  Is the output of all of the retired components. The output is a .csv file.

  For example, the following command outputs a list of all of the retired components. The .csv file format includes the device item ID, the device display name, the retired component ID, and the retired component display name:
  ```
  ./remove_not_present_items.sh -h my_host_name  -o myretired.csv
  ```
- To delete all of the retired components where the total exceeds 100,000, and to log the information to a .csv file, type the following command:
  ```
  ./remove_not_present_items.sh -h host_name  -o ouputfile  -c Yes
  ```
  – **-o *outputfile***
  Is the output of all of the retired components. The output is a .csv file.
  – **-c Yes**
  Confirms the deletion of all retired components.

  For example, the following command deletes all of the retired components:
  ```
  ./remove_not_present_items.sh -h my_host_name  -o myretired.csv  -c Yes
  ```

Delete Devices

To remove a device item, all associate components, and historic data from the system, delete the device.

When you delete a device, the following results occur:

- All the associated device components are deleted.
- Historical data on the deleted devices and device components is no longer accessible.

If discovery finds a deleted device, DX NetOps Performance Management creates a *new* device item. The new device item has no association with the old device item.

TIP

To stop polling the device and retain historical data, edit the device life cycle state. For more information, see Manage Device Life Cycles.

Follow these steps:

1. Go to Administration, and click the Data Aggregator data source.
   The Monitored Devices page opens.

2. (Optional) Open the Search tab, and search for the device to delete.

   NOTE
   Do not use the global Search box at the top of the page.
3. Select the devices to delete, and click **Delete**.
4. In the confirmation dialog, click **Yes**.
   The devices are deleted and are removed from the Monitored Devices inventory. Historical data for the devices is deleted from the database. After the next synchronization cycle, the devices are removed from inventory and groups in NetOps Portal.

   NOTE

   If another data source is managing these devices, the devices continue to appear in the Inventory view and in groups.

Device Reconfiguration

Device reconfiguration includes changes to physical device components and the software configuration, such as monitoring response path tests for protocols. Data Aggregator uses the same method to monitor both types of reconfiguration. To keep device components up-to-date, you can monitor and update device reconfiguration changes in Data Aggregator.

Additional examples of reconfiguration changes include:

- Adding a board to a device, which adds more ports.
- Adding memory, CPUs, physical interfaces, or any metric family to a discovered device.
- Reconfiguring a virtual switch.
- Changing the configuration of a device to include a discovered device in routing protocols.

When a change is detected, Data Aggregator generates reconfiguration events and can update its representation of the metric family to reflect the changes to device components. View reconfiguration events by selecting Dashboards, Operations, Events Display.

Understanding how change detection works in Data Aggregator helps you to select the options that are best suited to monitoring device reconfiguration in your environment. For example, you can set the frequency for change detection monitoring.

How to Manage Change Detection

Change detection management planning helps ensure that Data Aggregator detects and monitors device reconfigurations in your environment according to your needs. You can plan ahead for any device reconfiguration when you first set up Data Aggregator to discover new devices. You can also edit these options at any time after devices are discovered.

Refer to the following guidelines when planning change detection:

- Likelihood of change.
- Frequency of change.
- Tolerance for outdated data.

Monitor metric families, such as CPUs, for reconfiguration infrequently. For dynamic metric families, such as virtual systems, choose a more frequent rate.

Follow this process:

1. Create or edit a custom monitoring profile. (You can also copy a factory monitoring profile and edit the copy.)
2. Select **Enable Change Detection** and set the **Change Detection Settings**, **Rate** in the monitoring profile.

The Change Detection Settings, Rate option is used to set the frequency at which Data Aggregator checks for changes. The rate of detection can be set in minutes or hours. By default, the rate is set to 24 hours.

3. Update the Data Aggregator representation of the metric families.
   After you set the change detection rate, you have two options for correcting the Data Aggregator configuration: automatic or manual update of the metric families. Manual Update updates the representation of the metric families by ensuring the correct set of components is being monitored.
   – To avoid having to make changes when a reconfiguration is detected, select **Automatically Update Metric Families** (selected by default). Data Aggregator automatically starts monitoring any new components, and marks any components that are no longer detected as Not Present.
     View the Events Display dashboard to see reconfiguration events:
     • If the components have changed for a device, an event is generated on the associated device. This event describes that a component change has been detected and will be applied after a short period.
     • After component reconciliation is applied, another event is generated. This event describes how many components were added, how many are not present, and how many remained unchanged.
   – To stop the Data Aggregator from automatically monitoring new components or Not Present components, deselect **Automatically Update Metric Families**.
     View the Events Display dashboard to see reconfiguration events:
     • If the components have changed for a device, an event is generated on the associated device. This event describes a component change that has occurred without reconciliation.

     To have the reconfiguration changes applied, manually click **Update Metric Family** on the Polled Metric Families page for a device.
4. To enable the monitoring profile, assign the custom monitoring profile to a device collection.

TIP

- If your environment is undergoing major maintenance, turn off the automatic update until the maintenance is complete. For small, regular changes, enable the automatic update feature to ensure that your Data Aggregator stays up-to-date.
- Monitoring profiles are assigned to device collections. If you want to monitor special devices differently, create a custom device collection and assign a custom monitoring profile with the desired change detection settings. For example, you can monitor critical core routers more frequently than other routers by creating a device collection and assigning a custom monitoring profile that performs change detection hourly. Other routers remain in the 'All Routers' device collection using the factory monitoring profile (with no change detection), or a custom monitoring profile that you set to less frequent change detection.

Update Device Reconfiguration Automatically

Reconfiguration changes to a discovered device can affect the metric families that are associated with the device. The device reconfiguration can update automatically in the monitoring profile to which the metric families are assigned, which applies to any metric families that the monitoring profile includes. This option is set by default when you create a custom monitoring profile, but it can also be edited at any time.

When the metric family is updated, Data Aggregator has an accurate representation of the device configuration. Reports that you generate reflect accurate information.

Follow these steps:

1. Select the monitoring profile that you want to update automatically and click **Edit**.
2. Select **Enable Change Detection**.
3. Set the **Change Detection Settings, Rate** to a value that is greater than zero.

NOTE
Consider how frequently the metric family is likely to change, and how many devices the monitoring profile is applied to. Avoid setting change detection rates that are more frequent than necessary.

4. Select **Automatically Update Metric Families**.
5. Click **Save**.
   When you make a configuration change to a device that is associated with this monitoring profile, the device configuration is updated automatically.
   When a device configuration is updated, Data Aggregator does the following steps:
   – Generates an event on the monitored device.
   – Identifies new components and creates them.
   – Identifies components that are no longer present and marks them as Not Present.
   By default, components that are not present are not synchronized with NetOps Portal. To enable this option, select **Synchronize items that are no longer present on the device** on the Manage Data Sources page.
   – Identifies existing components that have changed from a previous discovery. The Name column changes, if applicable.
   Historical data is accessible and can be reported on.

Update Device Reconfiguration Manually

Reconfiguration changes to a discovered device can affect the metric families that are associated with the device. The device reconfiguration can be updated manually when **Automatically Update Metric Families** is not selected in the associated monitoring profile. View the event logs to identify reconfiguration events for which you want to update the metric families.

When the metric family is updated, Data Aggregator has an accurate representation of the device configuration. Reports that you generate reflect accurate information.

Follow these steps:

1. View the event logs to identify reconfiguration events for which you want to update the metric families.
2. Click **Monitored Devices** from the Monitored Inventory menu for a Data Aggregator data source.
   The Tree View tab appears.
3. Select **Device by Collection** from the drop-down list, and select the monitored device that was updated from the corresponding tree view.
   The Polled Metric Families tab shows the consolidated monitoring profiles that are associated with a device. Devices only have one consolidated monitoring profile. Each consolidated monitoring profile lists every metric family that can be polled on the device, and whether the device supports the metric family.
4. Select the metric family for which you want to update the configuration and click **Update Metric Family**.
   Your device configuration is updated, and Data Aggregator does the following steps:
   – Generates an event on the monitored device.
   – Identifies new components and creates them.
   – Identifies components that are no longer present and marks them as Not Present.
   – Identifies existing components that have changed from a previous Discovery. The Name column changes, if applicable.
   Historical data is accessible and can be reported on.

Reconfiguration Detection of Ports

Reconfiguration changes to a discovered device may cause ports to be reconfigured. The new port and the old port are identical if either of the following is true for the Interface metric family after the reconfiguration:

- The PortType and Description attributes in the Interface metric family remain the same.
- The PortType and at least two of the following remain the same:

- Alias
- Description
- MacAddress
- Index

For all other metric families, the new component and the old component are identical if both Name and Index remain the same after the reconfiguration.

Manage Device Life Cycles

Life cycle states define the monitoring behavior for devices. Life cycle management enables you to define the usage state of SNMP and ICMP devices. Components inherit the life cycle state of the associated device.

NOTE
Life cycle management does not apply to devices from DX NetOps Mediation Manager.

You can manage the device life cycle state using the Manage Device Life Cycle user interface in NetOps Portal or the NetOps Portal API. To use the NetOps Portal API, see Update the Device Life Cycle State.

You can also manage the device life cycle state from the device inventory. You can change the life cycle state of a device to Active or Maintenance from an alarms view.

Life Cycle States

- **Active**
  The normal operating status of a device.
- **Retired**
  Specifies that the device is no longer in use and that no monitoring occurs. This state disables polling, threshold monitoring, notifications, and change detection. The system does not update the SNMP Profile or change the hostname.

  NOTE
  Virtual devices (Virtual Machine or ESX) discovered with systemEdge are not polled for its vCenter statistics when the device lifecycle state is "Retired".
- **Maintenance**
  Specifies that the device is temporarily under maintenance. The behavior of this state is configurable.

Change Life Cycle Status

To define when a device is active, retired, or in maintenance, change the life cycle state.

Prerequisite: You have the Administer Life Cycle role right.

Follow these steps:

1. Do one of the following steps:
   - Go to **Administration**, **Device Life Cycle**.
   - Go to **Inventory**, **Devices**.
   - Go to an alarms view.

     NOTE
     You can change the life cycle state of a device only to "Active" or "Maintenance" from an alarms view. You cannot change the life cycle state to "Retired" from an alarms view.
2. Select the devices, and then click **Manage Life Cycle**.
3. Select the **Life Cycle State**, and then click **OK**.
   The system applies the selected state to the device and logs a life cycle event that marks the change.

NOTE
When a change to the life cycle state stops polling a device, if the data is within the rollup time frame, the polled data is included in rollups. For example, the system stops polling at 9:20 am. The data is included in rollups for the 9:00-10:00 hour, but not for the 10:00-11:00 hour. Baselines use data up to 90 days in the past, even if the device is no longer polled.

Configure Maintenance Behavior

To define the behavior for devices in the "Maintenance" state, specify the life cycle behavior.

WARNING
Changes to maintenance behavior do not apply to devices already in the "Maintenance" state. The changes apply only to devices set to "Maintenance" after you change the maintenance behavior.

Follow these steps:

1. Go to **Administration, Life Cycle Behavior**.
2. Under **Maintenance State**, specify the behavior:
   - **Polling**
     Specifies whether DX NetOps Performance Management polls the device.
   - **Threshold Evaluation**
     If polling is enabled, specifies whether DX NetOps Performance Management analyzes event rules for the device.
   - **Event Notification**
     Specifies whether DX NetOps Performance Management sends notifications for the device.
3. Click **Save**.
   DX NetOps Performance Management applies the defined behaviors to devices that you put in the "Maintenance" state.

DX NetOps Spectrum Integration

For systems with an integrated instance of DX NetOps Spectrum, you can configure DX NetOps Spectrum to control the life cycle state of devices in DX NetOps Performance Management. Changes in DX NetOps Spectrum trigger changes in DX NetOps Performance Management. If you change the state of a device in DX NetOps Performance Management, the state does not change again unless the state changes in DX NetOps Spectrum. With this option enabled, DX NetOps Performance Management uses the following behavior:

- Active devices in DX NetOps Spectrum are Active in DX NetOps Performance Management.
- Devices in Maintenance in DX NetOps Spectrum, have the Maintenance state in DX NetOps Performance Management.
- The state of the device in DX NetOps Spectrum overwrites the state in DX NetOps Performance Management.

To manage the life cycle state of a device from DX NetOps Spectrum, edit the DX NetOps Spectrum data source and select the **Synchronize device life cycle state from Spectrum** checkbox.

The following video explains the DX NetOps Spectrum integration behavior:

For more information, see Configure a Data Source.

Retirement Consolidation

The DX NetOps Performance Management user interface now supports supports the co-existence of "Retired" and replacement devices coming from Network Flow Analysis.

NOTE
You must upgrade to Network Flow Analysis 10.0 and DX NetOps Performance Management 3.7 to leverage this functionality.

In the earlier versions of DX NetOps Performance Management, it was not possible to have multiple devices with the same IP address. With this enhancement to the device life cycle, you can now have multiple devices with the same IP address displayed in DX NetOps Performance Management. When a device is "Retired", Network Flow Analysis retires the old device and creates a new device with an interface. The new and the retired device show in DX NetOps Performance Management after synchronization. A new device can be discovered using the same IP. You can now report on the new and retired device in the same or separate group. If you want to report on the new and old device together, we recommend that you place them in the group.

When Network Flow Analysis 10.0 detects a flow from a different device with the same IP , the existing device is marked as retired. Network Flow Analysis then creates a new entry for the new device with new set of interfaces.

NOTE
If you upgrade DX NetOps Performance Management to 3.7 before upgrading Network Flow Analysis to 10.0, a full Synchronization of Network Flow Analysis data source will be required when you upgrade Network Flow Analysis.

NOTE
When a device has been set to "Retired" state and a new device has been created, as with any new device, you can choose to configure and manage the new device accordingly.

Manage Hostname Changes

When the hostname of a monitored device changes, DX NetOps Performance Management detects the change and updates the hostname for the device item. If the device item name uses the hostname, DX NetOps Performance Management updates the device name. The default detection rate for hostname changes is 24 hours. However, the change might not appear in the system for up to twice the detection period. For example, with the default reevaluation rate of 24 hours, changes to the hostname are updated within 48 hours.

NOTE

DX NetOps Performance Management only looks for hostname changes on devices with defined hostnames. If the device did not have a DNS hostname when discovery created the device item, DX NetOps Performance Management does not detect changes. To add the hostname to the device item, run the discovery profile again. For more information, see Run Discovery.

Modify the Hostname Change Detection Rate

The default hostname change detection rate is 24 hours. To reduce DNS traffic or to detect changes faster, modify the reevaluation interval. Each Data Collector has a separate evaluation interval. To change the interval for a specific device or group of devices, apply the changes to each Data Collector individually.

Follow these steps:

1. Log in to the Data Collector host that monitors the device.
2. Locate the Data Collector configuration file:
   `APACHE_KARAF/etc/com.ca.im.dm.core.collector.cfg`
3. Add or modify the following line:
   `hostname-reevaluation-interval-in-hours=`*`number`*
   number specifies the length in hours of the reevaluation interval.
4. Save the file.
   The Data Collector uses the new interval for hostname reevaluation. Changes to the hostname take up to twice the interval to appear in the system.

Override Device Types

Based on the device service information, the Data Aggregator automatically classifies manageable devices. Classifications include Router, Switch, Server, Firewall, Load Balancer, Wireless Controller, and Wireless Access Point. During synchronization, the Data Aggregator inspects each device and its classification. The Data Aggregator designates a primary device type, and forwards any additional types as context types. The associated context types, including the primary device types, appear listed in a column within the device inventory tables in NetOps Portal.

The following list describes more device type classifications:

- **Pingable**
  The device is not manageable.
  Example: The device does not respond to SNMP requests and its device type cannot be determined.
- **Manageable**
  The device type is classified as 'Manageable' when the following criteria are true:
  – The manageable device cannot be classified as Router, Switch, or Server.
  – The manageable device *can* be classified as Firewall, Load Balancer, Wireless Controller, or Wireless Access Point.
- **Other**
  The device type is classified as 'Other' when the following criteria are true:
  – The manageable device cannot be classified as Router, Switch, or Server.
  – The manageable device cannot be classified as Firewall, Load Balancer, Wireless Controller, or Wireless Access Point.

Override Device Types

If the device types of some SNMP manageable devices were not classified as expected, you can fully override the device types.

Scenarios:

- An existing device is discovered as 'Other'. However, you want the device to be classified as 'Router' instead. To override the existing 'Other' classification, you can add 'Router' to the DeviceTypes.xml.
- An existing device is discovered as 'Switch'. The SNMP agent advertised the wrong type. The device should have been advertised as 'Router' and it is also a 'Firewall'. To override the incorrectly discovered types, you can add 'Router' and 'Firewall' to the DeviceTypes.xml.

Map the device `sysObjectID` MIB value explicitly to the correct device type in the following file on the Data Aggregator host:

```
DA_install_directory/data/custom/devicetypes/DeviceTypes.xml
```

In a fault tolerant environment, a shared directory (example: `/DASharedRepo`) is defined to help limit data loss. Therefore, in a fault tolerant environment the file would be located in the following directory:

```
DASharedRepo/custom/devicetypes/DeviceTypes.xml
```

For more information, see Fault Tolerance.

> **NOTE**
> You cannot add new device types to the DeviceTypes.xml file.

The DeviceTypes.xml file contains a template to map the `sysObjectID` to appropriate device types. By default, the file does not contain any `sysObjectID` -to-type mapping entry. To classify a device type with a particular `sysObjectID` , modify the template to add the `sysObjectID` -to-type entries into the file. Before you add a `sysObjectID` , uncomment the section where you are adding the `sysObjectID` .

NOTE
Updates to the DeviceTypes.xml file can take up to one minute to apply.

A device can be classified into multiple device types. However, the type, Device, is mutually exclusive to other device types. For example, if you add a `sysObjectID` to one or more of the Router, Switch, or Server device types and you also add that `sysObjectID` to the 'Device' device type, the 'Device' device type is dropped and is not recognized.

Fully override the device types when the following criteria are true:

- No automatically discovered device types were found or the discovered device types are wrong.
- You want to override the types discovered during device discovery.
- You want to assign the correct types manually.

Example: Map a Device sysObjectID to Another Device Type

WARNING

The `sysObjectID` -to-type mappings in the following example override any already discovered device types.

Follow these steps:

1. Open the following file:
   DA_install_directory/data/custom/devicetypes/DeviceTypes.xml
   In a fault tolerant environment, the file would be located in the following directory:
   DASharedRepo/custom/devicetypes/DeviceTypes.xml
2. Enter the following information:

```
<DeviceType>
     <Routers>
          <sysObjectID>1.3.6.5.1.34</sysObjectID>
     </Routers>

     <Switches>
          <sysObjectID>1.3.6.5.5.3</sysObjectID>
          <sysObjectID>1.3.6.5.1.34</sysObjectID>
     </Switches>

     <Servers>
          <sysObjectID>1.3.6.5.567.1</sysObjectID>
     </Servers>

     <Devices>
          <sysObjectID>1.3.6.5.49.1</sysObjectID>
     </Devices>
</DeviceType>
```

3. Run discovery on the discovery profile that contains the devices.

   NOTE
   The changes that you make to the DeviceTypes.xml file do not take effect on existing devices until you rerun discovery.

   When discovery is run, the following results occur:

- All devices that have a `sysObjectID` of 1.3.6.5.1.34 are classified as a device type of Router and Switch.
- All devices that have a `sysObjectID` of 1.3.6.5.5.3 are classified as a device type of Switch.
- All devices that have a `sysObjectID` of 1.3.6.5.567.1 are classified as a device type of Server.
- All devices that have a `sysObjectID` of 1.3.6.5.49.1 are classified as a device type of Device.

4. To verify your changes, go to the following locations:
   - Hover over **Administration**, and click **Monitored Items Management: Device Life Cycle**.
   - Hover over **Inventory**, and click **Devices**.

Preserve Device Types

Sometimes, a device might have an incomplete classification. For example, a device that is discovered as a Router may be a Firewall too. In this case, you want to preserve the discovered Router classification. You also want to augment the classification with one or more manually assigned device types (in this example, Firewall).

Scenarios:

- An existing device is discovered automatically as a 'Server'. You want the device to be classified as a 'Firewall' too. To preserve the discovered 'Server' classification, add 'Firewall' to the DeviceTypes.xml with the `sysServiceOverride` tag attribute set to 'false'.
- An existing device is discovered automatically as a 'Router'. You want the device to be classified as a 'Firewall' and 'Switch' too. To preserve the existing 'Router' classification, add 'Firewall' and 'Switch' to the DeviceTypes.xml with `ysServiceOverride` tag attribute set to 'false'.

The `sysServiceOverride` tag attribute provides a way to disable the default override behavior. You can assign a `sysObjectID` to one or more device types in the DeviceTypes.xml (for example, Firewall) in addition to the types that the system automatically discovers (for example, Router). To preserve the device types, use the `sysServiceOverride` tag attribute and set it to 'false' for the `sysObjectIDs` or device types.

Use the `sysServicesOverride` tag attribute when the following criteria are true:

- You want to preserve the type that is discovered during device discovery.
- You accept the automatically discovered device type as correct and do not want to lose it.
- You want to add more types.

Example: Preserve the device types already mapped to the sysObjectIDs

Follow these steps:

1. Open the following file:
   DA_install_directory/data/custom/devicetypes/DeviceTypes.xml
   In a fault tolerant environment, the file would be located in the following directory:
   DASharedRepo/custom/devicetypes/DeviceTypes.xml
2. Enter the following information:

```
<DeviceType>
     <Routers>
          <sysObjectID sysServicesOverride="false">1.3.6.5.1.34</sysObjectID>
     </Routers>

     <Firewalls>
          <sysObjectID sysServicesOverride="false">1.3.6.1.4.1.8072.3.2.10</
sysObjectID>
     </Firewalls>

     <Firewalls sysServicesOverride="false">
```

```
            <sysObjectID>1.3.6.1.4.1.9.1.522</sysObjectID>                <!--
    cat6500FirewallSm -->
            <sysObjectID>1.3.6.1.4.1.2620.1.6.123.1.56</sysObjectID>      <!-- Check
    Point 21800 -->
            <sysObjectID>1.3.6.1.4.1.2620.1.6.123.1.16</sysObjectID>      <!-- Smart-1
    150 -->
        </Firewalls>

        <WirelessAccessPoints>
            <sysObjectID>1.3.6.1.4.1.2620.1.6.123.1.48</sysObjectID>
        </WirelessAccessPoints>

        <WirelessControllers>
            <sysObjectID>1.3.6.1.4.1.9.1.770</sysObjectID>
        </WirelessControllers>

        <LoadBalancers>
            <sysObjectID>1.3.6.1.4.1.6527.1.6.1</sysObjectID>
        </LoadBalancers>

    </DeviceType>
```

3. Run discovery on the discovery profile that contains the devices.

   NOTE
   The changes that you make to the DeviceTypes.xml file do not take effect on existing devices until you rerun discovery.

   When discovery is run, the following results occur:
   - All devices that have a `sysObjectID` of 1.3.6.5.1.34 are classified as a device type of Router. The device types that are already mapped to the `sysObjectID` are preserved.
   - All devices that have a `sysObjectID` of 1.3.6.1.4.1.8072.3.2.10 are classified as a context type of Firewall. The device types that are already mapped to the `sysObjectID` are preserved.
   - All devices that have a `sysObjectID` of 1.3.6.1.4.1.9.1.522 are classified as a context type of Firewall. The device types that are already mapped to the `sysObjectID` are preserved.
   - All devices that have a `sysObjectID` of 1.3.6.1.4.1.2620.1.6.123.1.56 are classified as a context type of Firewall. The device types that are already mapped to the `sysObjectID` are preserved.
   - All devices that have a `sysObjectID` of 1.3.6.1.4.1.2620.1.6.123.1.16 are classified as a context type of Firewall. The device types that are already mapped to the `sysObjectID` are preserved.
   - All devices that have a `sysObjectID` of 1.3.6.1.4.1.2620.1.6.123.1.48 are classified as a context type of Wireless Access Point. The device types that are already mapped to the `sysObjectID` are preserved.
   - All devices that have a `sysObjectID` of 1.3.6.1.4.1.9.1.770 are classified as a context type of Wireless Controller. The device types that are already mapped to the `sysObjectID` are preserved.
   - All devices that have a `sysObjectID` of 1.3.6.1.4.1.6527.1.6.1 are classified as a context type of Load Balancer. The device types that are already mapped to the `sysObjectID` are preserved.
4. To verify your changes, go to the following locations:
   - Hover over **Administration**, and click **Monitored Items Management: Device Life Cycle**.
   - Hover over **Inventory**, and click **Devices**.

Set Alias Names For Multiple Monitored Devices

To set aliases for multiple monitored devices simultaneously, use a script that is included with DX NetOps Performance Management. The alias appears in the inventory lists for devices and interfaces. An alias that you set using this script takes precedence over the alias that is set by importing a CSV file when you add an IP domain.

The script returns a list of device item IDs and device names in CSV format. Add the alias names that you want to set on each monitored device to the CSV file. The script reads the updated CSV file and sets the alias names for the monitored devices.

Follow these steps:

1. Open a command prompt and access the following directory:
   `Performance_Center_installation_directory/PerformanceCenter/Tools/bin`
2. To set alias names for monitored devices, type the following command:
   `./update_alias_name.sh`
3. To return a complete list of monitored devices, type the following command:
   `./update_alias_name.sh -h host_name -u username -p password [-T item_type] [-o output_filename]`
   - **-h host_name**
     Specifies the NetOps Portal host name.
   - **-u username**
     Specifies the username of the administrator who sets the alias names.
   - **-p password**
     Specifies the password for the CA NetOps Portal administrator who sets the alias names.
   - **-T item_type**
     Specifies the type of item for which you want to set alias names.
     Acceptable values: device, interface, or component.
     Default: device
     Keep the default value.
   - **(Optional) -o output_filename**
     Creates a CSV file with the total number of monitored devices by itemID and Device Name. Use this command to specify to override the default file name. If you do not enter a value for this parameter, DeviceList.csv is used for the .csv file.
     The CSV file has the following format: Device ItemID, Device Name.
     Examples:
     - `560, MyRouter1`
     - `561, MyRouter2`
   - Modify the CSV file as needed. Take note of the alias name that you want to set for each monitored device. This file has the following format: Device ItemID, Device Alias Name.
     If the Item IDs in your CSV file are invalid, the entries are ignored.
     Examples:
     - `560, MyRouter1AliasDisplayName`
     - `561, MyRouter2AliasDisplayName`

     NOTE

     For devices, the Alias Name value must be URL-encoded. Commas are allowed in the Alias Name field of the .csv file. Spaces must be URL-encoded as “%20”.
     For interfaces and components, the Alias Name value must be XML-encoded. Commas and spaces are allowed in the Alias Name value of the .csv file. The ampersand character must be encoded “&”. The less-than character must be encoded “<”. The greater-than character must be encoded as ">".
   - Type the following command:
     `./update_alias_name.sh -h host_name -u username -p password [-T device] -i input_file`

     -i input_file

Specifies the name of the CSV file.The alias names are set for monitored devices. If this parameter is not specified, the script finds all the item IDs that are required for the specified type, and creates a CSV file with item IDs and item names.

– (Optional) To control the workload when you set alias names for many monitored devices, type the following command, which adjusts the batch size and creates pauses between batches:

```
./update_alias_name.sh -h host_name -u username -p password -T device -i input_file -b batch_size -
t time_in_seconds
```

- **–b batch_size**
  Indicates the number of items to process in each batch.
  Default: 10000
  Default with the -i parameter unspecified: 150
- **–t time_in_seconds**
  Indicates the time, in seconds, to pause between batches.
  Default: 1
  Default with the -i parameter unspecified: 1
  Example:

```
./update_alias_name.sh -h host_name -u username -p password -T device -
i input_file -b 20 -t 2
```

Device Deduplication

Without deduplication, a single physical device is modeled as multiple device items when it is discovered from multiple sources (Simple Network Management Protocol (SNMP), DX NetOps Mediation Manager, DX NetOps Virtual Network Assurance, or another source). Deduplication models a single device in DX NetOps Performance Management.

The following general rules apply to deduplication. Exceptions to these rules are covered in the following sections.

- DX NetOps Performance Management deduplicates devices only when they are in the same IP domain.
- DX NetOps Performance Management successfully deduplicates devices regardless of discovery order.
- When the deduplication criteria of a device (IP address, hostname, and so on) from one source matches an existing device item from another source, DX NetOps Performance Management does not create a new device item. Instead, the same existing device item is used for both sources.
- If attributes conflict and SNMP is a source, the attributes from SNMP take precedence.

SNMP Deduplication

New SNMP devices are not created in the following scenarios:

> **NOTE**
>
> The primary IP address is the IP address that DX NetOps Performance Management uses to monitor a device. When a device is first discovered with the IP ranges discovery profile, DX NetOps Performance Management tries to use the IP address that maps to the hostname as the primary IP address.

- The primary IP address matches an existing device. The existing device could be a DX NetOps Mediation Manager (CAMM) or DX NetOps Virtual Network Assurance (VNA) device with the same primary IP Address.
- The hostname matches an existing device.
- The primary IP address for a new device is in the IP address list of an existing device, and the primary IP address of the existing device is in the IP address list of the new device. If the primary IP address of the new device is listed as a single IP address in the discovery profile, `sysName` is verified. If the `sysName` of the new and existing device match, the new device is not created.
- For devices that support the Device Unique Identifier metric family, the `UniqueID` matches an existing device.

SNMP and DX NetOps Mediation Manager Deduplication

DX NetOps Mediation Manager (CAMM) monitors devices that do not support SNMP or provides metrics that are not accessible through SNMP polling. CAMM injects data into the Data Aggregator through one of the Data Collectors. When the CAMM device pack provides an IP address or hostname that matches another device in the system, DX NetOps Performance Management deduplicates the devices to a single item.

The following table illustrates that DX NetOps Performance Management deduplicates SNMP and CAMM devices when the IP address or hostname match:

IP Address Match	Hostname Match	Deduplication
Yes	Yes	Yes
Yes	No	Yes
No	Yes	Yes
No	No	No

To deduplicate devices by the IP address only, edit the discovery profile to exclude hostname. For more information, see Discovery Profiles.

The following criteria must be met for deduplication to occur:

- Both the Data Collector and the CAMM Local Controller are in the same IP domain and poll the same device.
- Either a matching IP address or a matching hostname is present in both SNMP and CAMM.
- The CAMM device pack supports deduplication.

  WARNING

  Deduplication with CAMM devices occurs only for device packs that support deduplication. To determine which device packs support deduplication, see the information file of each device pack.
- For multiple devices packs, the device names on each device pack match.
- CAMM and SNMP discovery is successful.

DX NetOps Performance Management deduplicates devices only when the Data Collector and CAMM Local Controller and are in the same IP domain. However, the Data Collector and CAMM Local Controller can reside on different servers. The following server configurations are supported:

- Data Collector server for SNMP polling
- Local Controller server for CAMM polling
- Data Collector and Local Controller on the same server for SNMP and CAMM polling

If the combined workload is manageable, the CAMM Local Controller can be installed on the same server as the Data Collector for SNMP polling. However, ensure that your servers have enough capacity to continue operating with the normal SNMP polling load and the CAMM requirements.

For new installations, install the CAMM Local Controller on a Data Collector in the same IP domain with the devices that you want to monitor. Devices that are discovered through CAMM and through SNMP are deduplicated.

For existing installations, rediscover the devices with SNMP or CAMM. CAMM deduplication does not delete any existing duplicated device items. The historical data is still available on the existing device items. Historical data that was captured with the device model from an old IP domain still exists, but is unconnected to the newly reconciled device. We recommend that you delete or retire the device from the old IP domain to avoid double polling from CAMM. If the device from the old IP domain is not deleted, or the historical data is not aged out, DX NetOps Performance Management continues to have two devices.

CAMM components and SNMP components from the same Metric Family are not reconciled with each other. SNMP component discovery is delayed until the next change detection. An error occurs when all the following conditions are true:

- SNMP and CAMM polled devices in the same IP domain.
- The SNMP and CAMM devices were polled by the same Data Collector.
- The SNMP and CAMM device packs contribute to the same Metric Family.
- One of the two contributors (CAMM or SNMP) have late arriving data (greater than 30 minutes after the rollup period ends). Late arriving data is more likely to happen with CAMM data, especially with a 15-minute polling interval.

When the load on the Data Collectors is rebalanced, CAMM and VNA devices are not rebalanced. Therefore, deduplicated SNMP, CAMM, and VNA devices are not rebalanced. For more information, see Rebalance the Load on Data Collector.

Manage Metric Families

Viewing a metric family shows its associations with device collections, vendor certifications, and monitoring profiles. Understanding the relationships between metric families, device collections, and device types helps you control how to monitor your devices. Also, you can determine whether you need more metric families to monitor your environment sufficiently.

Follow these steps:

1. Hover over **Administration**, and click the Data Aggregator data source.
2. Expand **Monitoring Configuration**, and click **Metric Families**.
   Click the heading columns to sort the Metric Family columns, as needed.
3. Select a metric family from the list.
4. Click a tab to get more information:
   - **Metrics tab**
     View the metrics that are included in the selected metric family and various properties for each metric.
     - **Name**
     - **Polled**
     - **Min**
     - **Max**
     - **Percentiles**
     - **Projections (Days)**
     - **Projections Percentile**
     - **Baseline**
     - **Rollup Strategy**
     - **Standard Deviation**
   - **Vendor Certification Priorities tab**
     View a list of device collections that are associated with the selected metric family. Typically, a metric family is associated with a single device collection. When you select a device collection, a prioritized list of MIB sources (vendor certifications) appears. This information shows the order in which the vendor certifications are applied to that device collection for the metric family.
   - **Monitoring Profiles tab**
     View a list of the associated monitoring profiles and their poll rates.

Configure Metric Filtering

Metric filtering reduces the storage footprint of collected performance data by limiting the metrics collected from a metric family for a particular monitoring profile. For devices in associated collections, only the selected metrics are loaded.

The following example helps to describe the behavior for multiple monitoring profiles.

Example:

- One monitoring profile (A) has no metric filtering.
- Another monitoring profile (B) has some metric filtering to collect only BitsIn, BitsOut.
- Each monitoring profile is associated with a different collection.

Result:

- If a device ends up in both collections, the configuration from (B) takes priority. The system only collects BitsIn, BitsOut.

  WARNING

  If a device is associated with multiple monitoring profiles, all selected metrics from all monitoring profiles are collected and saved for the device at the fastest rate among all monitoring profiles.

Storage savings are not linear. To estimate the storage savings, use the Configure Data Retention Rates.

Metric filtering can only be applied to custom monitoring profiles. You cannot apply metric filtering to a monitoring profile with a lock icon.

NOTE

Metric filtering does not reduce SNMP traffic. Filtered metrics are still polled. Metric filtering applies only to metrics collected through SNMP.

Filtered metrics are still available for selection when building dashboards or configuring event rules for thresholds. Consider dashboard and threshold requirements before you filter a metric.

Follow these steps:

1. Hover over **Administration**, and click **Monitored Items Management: Monitoring Profiles**.
2. Select the monitoring profile, and click the **Metric Families** tab.
3. Select the metric family, and click **Edit Collected Metrics**.
4. Move the metrics to collect to the **Selected** pane.

   TIP

   By default, all metrics are selected. To pick a limited set of metrics, select all metrics, and move the metrics to the **Available** pane. Then move the metrics to collect to the **Selected** pane.
5. Click **Save**.
   Only the selected metrics are loaded to the Data Repository for devices in collections associated with the monitoring profile.

   WARNING

   If the collection is associated with another monitoring profile that includes the same metric family, or if the devices are in another collection associated with a monitoring profile with the same metric family, the metrics are still collected.

To view the metrics that are collected for a particular device, go to the device administration page, click the **Metrics** tab, and select a metric family. A column in the right pane indicates where the metric is collected.

NOTE

This tab represents consolidated information across all monitoring profiles associated with all collections the device belongs to. If you see a metric that you do not want to collect, view the metric families associated with each monitoring profile for each collection.

Edit a Metric

Use the UI to edit metric properties for any metric in a specific metric family. When you edit a metric, the metric family is extended automatically with the new values.

For complete information about how to extend metrics and metric families through REST services, see Create or Extend Metric Families.

> **NOTE**
>
> You can only edit a metric that is being polled. If the value in the Polled column on the Metrics tab is False, then you cannot edit the corresponding metric.

Follow these steps:

1. Click Metric Families from the Monitoring Configuration menu for a Data Aggregator data source.
   A list of metric families appears, including factory and custom metric families.
2. Select a metric family from the list.
3. Click a metric in the Metrics tab, and then click the Edit button.
   The Edit Metric dialog opens.
4. Enter the desired values in the Percentile, Projection, and Projections Percentile fields. Acceptable values for the fields in the Edit Metric dialog are as follows:
   - **95th Percentile:** Enabled or Disabled
   - **Percentiles 2 and 3**: 0-99, except 95
   - **Projections 1-3**: 0-730
   - **Projection Percentile:** 0-99

     > **WARNING**
     >
     > Changes to **Projection Percentile** cause inaccurate projections for up to 90 days. Changes to **Percentile 2** and **Percentile 3** cause a gap in the trend view. When you change the values in these fields, the percentile values for days before the change are not recalculated.

   For more information, see Percentiles and Metric Projection.
5. Click the Save button.
   The selected metric is updated in the parent metric family. The metric family is marked as extended and the Last Modified and Last Modified By fields are updated.
   The new data is available for reporting within several poll cycles.

Populate Components List for Response Path Metric Family

The Monitored Devices page can display a Supported status for a response path metric family on the Polled Metric Families tab, but the Components list for that metric family can be empty. To populate the Components list, update the Metric Family.

Follow these steps:

1. Verify that the device is configured to run the test that is associated with the metric family.
   For example, if the "Response Path Test DHCP" metric family shows no components, verify that the device is configured to run the DHCP test.
2. Select the row for the metric family, and click Update Metric Family.

Rediscover Metric Families

To ensure that you have the latest metric family and component support information for your devices, rediscover metric families.

NOTE
Log in as a tenant administrator to perform this task.

Follow these steps:

1. Select **Administration**, **Data Sources**, and click a Data Aggregator data source.
2. Click **Monitored Devices** from the Monitored Inventory menu.
3. Select a device for which you want to rediscover metric families from the Tree View tab.
   The device information displays in the Polled Metric Families view.
4. Click the **Update Metric Families** button.
   The Update Metric Families dialog opens asking you to confirm whether to update all metric families.
5. Click **Yes**.
   The status of each polled metric family is updated, such as whether it is newly or no longer supported. This status includes updating the Last Discovered time.
   The Components view is also updated for each polled metric family. Components that are no longer found during the rediscovery get a status of Not Present.

Manage Interfaces

Interfaces represent monitored communications ports, such as Ethernet or serial ports. DX NetOps Performance Management includes the following options for managing interfaces:

Poll Critical Interfaces Faster than Non-critical Interfaces

As the Administrator, you need frequent data about your most critical systems while maximizing the overall performance of your performance management systems. One way to accomplish your goal is by polling only critical interfaces at a high rate, while polling noncritical interfaces at a normal or slow rate. You can poll at differing rates by using a filter on the Interfaces metric family that is associated with your monitoring profile. By fast-polling interfaces sparingly, you can reduce unnecessary network traffic and performance management system load while still sufficiently monitoring the health of your network system.

For example, your data center access switch connects many application servers to only two aggregation switches. You decide to poll the interfaces supporting these aggregation switches at a higher rate. These links are critical, because they support network traffic to all other connected switches. However, polling *all* interfaces at a higher rate would cause unnecessary network traffic, wasting system resources and possibly causing network performance issues. After consulting with your network operations and engineering teams, you decide that a normal polling rate is sufficient for the interfaces connecting each attached server. To apply different polling rates, you implement two monitoring profiles for interfaces.

NOTE
Filters that you set on metric families are ignored when event rules that are applied to monitoring profiles trigger events.

View Your Monitoring Profiles

As the CA NetOps Portal Administrator, you decide to poll critical interfaces as often as possible. However, you want to minimize unnecessary network traffic that polling *all* interfaces at this fast rate can produce. You decide to create two monitoring profiles for interfaces -- one with normal polling, and one with fast polling.

Before you create a monitoring profile, you review the existing monitoring profiles to find one that closely matches your needs.

Follow these steps:

1. Click Monitoring Profiles from the Monitoring Configuration menu for your Data Aggregator data source.
   A list of monitoring profiles is populated.

2. Select a monitoring profile.
   Details for the selected monitoring profile populate the tabs:
   – Metric Families tab -- Displays a list of metric families that are associated with that specific monitoring profile. Metric families contain the metrics that are used for polling devices and components.
   – Collections tab -- Displays a list of device collections that are associated with that specific monitoring profile.

Copy a Factory Monitoring Profile

As the CA NetOps Portal Administrator, you find that the factory Network Interfaces monitoring profile closely matches your needs and requires only minor changes. Therefore, you create a copy and use it to poll only critical interfaces at a faster polling rate.

NOTE
Log in as the administrator to perform this task.

Follow these steps:

1. Navigate to the list of all monitoring profiles in CA NetOps Portal.
2. Select the Network Interfaces monitoring profile and click Copy.

   NOTE
   Factory monitoring profiles cannot be edited or deleted. All monitoring profiles, including custom, are global.

   The Create/Edit Monitoring Profile dialog opens.
3. Enter the following information for your monitoring profile:
   – **Name:** Uplink Interfaces
   – **Description** (optional)**:** Monitors performance of interfaces in all critical Uplink devices.
   – **SNMP Poll Rate:** 1 minute

   NOTE
   We recommend that you rename the profile. Unique naming is enforced across all tenants.

   Consider the following information about poll rates:
   – When the poll rate is changed, it takes up to two cycles for the new poll rate to take effect. When the 60-minute rate is used to poll an existing device, a 'No Data To Display' message appears in the dashboard view given the default time range of Last Hour. If you change the dashboard setting to a prior hour, it is possible to see earlier data. However, the view does not display the latest data until the new poll cycle completes.
   – Interfaces that are assigned to multiple monitoring profiles with different poll rates are polled at the fastest assigned rate.
4. Leave the Change Detection Settings, Rate value at 24 Hours.
   Consider the following information about change detection rates:
   – The *change detection rate* is how often Data Aggregator checks whether any components on a device have been reconfigured. Changes can include new components that have been created or existing components that have been retired.

     NOTE
     The reconciliation algorithm specified in the metric family defines the configuration changes to watch for.
   – The Change Detection Settings, Rate option is used to set the frequency at which Data Aggregator checks for changes. The rate of detection can be set in minutes or hours. By default, the rate is set to 24 hours.
   – Changes are detected at the fastest rate you specified for all of the monitoring profiles that are associated with a collection of devices.
5. Leave the 'Automatically Update Metric Families' check box selected.

This option controls the Data Aggregator response once a change or reconfiguration is detected. Selecting this option automatically causes Data Aggregator to start monitoring new components or to stop monitoring retired components. When this option is not selected, you can manually control monitoring of components, as follows:

a. Manually check the Events Display dashboard to watch for configuration events.
b. Navigate to the Data Aggregator administration menu, Monitored Devices, Polled Metric Families view.
c. Select the appropriate metric family, and click Update Metric Family to help ensure that Data Aggregator picks up the latest device reconfiguration.

NOTE
If an interface filter is applied, Data Aggregator monitors only the interfaces that pass the filter conditions after reconfiguration.

6. Leave the Interfaces metric family as the only metric family in the Selected Metric Families list.
7. Click Save.
   Your copied monitoring profile is added to the Monitoring Profiles list. However, this monitoring profile is not active until you assign it to a device collection.

Set an Interface Filter

By default, the factory network interface monitoring profile includes a filter to prevent modeling interfaces that are administratively down. In addition, interfaces with a type (ifType) of 1 (Other) or 24 (Loopback) are not modeled, regardless if those interfaces are administratively up or down. IPSLAs with the rttMonCtrlAdminOwner MIB object that contains the string “Network Health" are not modeled either.

Filtering reduces the number of interfaces that are monitored, which reduces unwanted data collection and network traffic.

In addition to polling only administratively up interfaces, you also want to poll the most critical interfaces more frequently. To isolate and poll only these interfaces faster, you add a second filter condition to the interface filter associated with your custom monitoring profile. This second filter condition isolates the critical interfaces by finding only interfaces that contain "uplink" in their description.

NOTE
Log in as the administrator to perform this task.

Follow these steps:

1. Select your interfaces monitoring profile (called "Uplink Interfaces") from the Monitoring Profiles page.
2. Click Interface metric family row on the Metric Families tab and click Edit Filter.

   NOTE
   Do not click directly on the metric family name, because it is linked to take you to the metric family definition. Instead, click the row the metric family name is in to activate the Edit Filter option.
3. Click the Add Condition button.

   NOTE
   Multiple conditions are connected with an "and" operation. That is, all conditions must be met to satisfy the filter.
4. Configure the filter conditions with the following options and click Save:
   – Attribute: Description
   – Operation: Contains
   – Filter Value: uplink

   NOTE
   The Filter Value field is case-sensitive.

   Consider the following details about additional attributes you can use for filtering:

- For Speed In and Speed Out, you can use a decimal in the text field (such as 1.544) and can specify bps, Kbps, Mbps, or Gbps.
- For more information about configuring Type (that is, ifType), see the iana web site: .
- For Description and Alias, you can use a regular expression for filtering only when you select the Matches Regex or the Does Not Match Regex operation.
- When you save your changes, the filter criteria display on the Metric Families tab. You can now apply this monitoring profile to the appropriate device collection to begin polling your selected interfaces.

NOTE
Data Aggregator applies filtering after discovery. Interface components that do not match the filter criteria are not polled. If you add or edit an Interface filter *after* you run a discovery, polling on these components stops. These interface components are *not* displayed in CA NetOps Portal dashboards and data views.

Considerations for Interface Filters and Multiple Monitoring Profiles

When multiple monitoring profiles are assigned to a device collection, the filter matching criteria follows the "or" rule. So, Data Aggregator monitors all interfaces that satisfy the criteria for any of the monitoring profiles in the group.

Some of the monitoring profiles may have filters and some may not. Plus, these profiles can specify differing poll rates. In this case, Data Aggregator monitors the interfaces that match any monitoring profile, but the polling rates can differ. If more than one monitoring profile applies to an interface, Data Aggregator polls the interface once, and polls it at the fastest polling rate:

- Monitoring Profile 1 -- Filter: Description contains "X," Poll Rate: 1 minute
- Monitoring Profile 2 -- Filter: None, Poll Rate: 5 minutes
- Monitoring Profile 3 -- Filter: Description contains "Y," Poll Rate: 10 minutes

In this example, interfaces that match Monitoring Profile 1 are polled every minute. All other interfaces are polled every 5 minutes. Interfaces that match Monitoring Profile 3 also match Monitoring Profile 2, which does not include a filter. The fastest poll rate applies, so no interfaces are polled at 10-minute intervals.

In this case, if one monitoring profile has no filter, the result is that many interfaces may be polled more frequently than necessary. Therefore, after you set a filter, remove associations from other monitoring profiles to make sure that only components matching the specified filter are monitored.

Assign Your Monitoring Profile to a Device Collection

As the administrator or a tenant administrator, you associate the new Uplink Interfaces monitoring profile with a device collection to begin polling. In this case, you associate the profile with the Switches device collection, which is the same device collection that is associated with the factory Network Interfaces monitoring profile. Polling rates are applied to the interfaces in this device collection, as follows:

- Fast polling: Interfaces that satisfy the filter criteria of the Uplink Interfaces monitoring profile.
- Normal polling: All other interfaces that the Network Interfaces monitoring profile discovers.

WARNING
All custom monitoring profiles are global and visible to tenant administrators. However, the association of a monitoring profile with a specific device collection can be scoped to a tenant.

Follow these steps:

1. Click Collections from the Monitoring Configuration menu for your Data Aggregator data source.
   A list of device collections displays. Administrators can view the device collections for the tenant they are administering. A tenant administrator can view its own (tenant) list of device collections.
2. Select the All Switches device collection and click the Monitoring Profiles tab.

A list displays the monitoring profiles that are associated with the selected device collection. The Network Interface device collection exists in this list.

3. Click Manage.
   The Assign Collection Monitoring Profiles dialog opens.
4. Select the Uplink Interfaces monitoring profile and click Add.
   The selected monitoring profile moves to the Assigned Monitoring Profiles list.
5. Click Save.
   Your changes are saved.

View Monitored Devices to Verify Results

After you set up your monitoring profiles, review the monitored devices and the Filter report to verify that only your critical devices are polled at the higher rate. This information helps you to see information in context, such as which monitoring profiles are being used to poll device components. Verifying the results can help you identify any necessary adjustments to help you achieve the polling results that you want.

Note: Monitored devices are manageable devices and pingable (accessible but not manageable). Inaccessible devices are not monitored devices. Components of monitored devices can be viewed from the Polled Metric Families tab.

Follow these steps:

1. Run an on-demand discovery.

   NOTE
   If your discovery profile runs automatically, you can wait for the next scheduled discovery.
2. Click Monitored Devices from the Monitored Inventory menu for a Data Aggregator data source.
3. Select one of these options from the drop-down list to locate one of your aggregation switch devices in the corresponding tree view:
   - Device by Collection -- Your devices appear under the All Switches device collection.
   - Device by Monitoring Profile -- Your critical interfaces appear under Devices under the Uplink Interfaces monitoring profile.

     NOTE
     Alternatively, select the Search tab to search by host name, device name, or IP address. You can enter a partial name or IP address to return a list of devices that contain that partial match. Wildcards and regular expressions are not supported.

   The Polled Metric Families tab shows the consolidated monitoring profiles that are associated with the switch device. Devices have only one consolidated monitoring profile. Each consolidated monitoring profile lists every metric family to poll on the device and whether the device supports the metric family.
4. Select the Interface metric family.
   The Components table for the Interfaces metric family shows one of the following polling statuses for the discovered Interface components:
   - **Active**
     Indicates that the component is being polled.
   - **Inactive**
     Indicates that polling has stopped on the component because the metric family is no longer monitored for the device.
   - **Not Present**
     Indicates that the component no longer exists on the physical device. Polling is stopped on the component. You can view historical data for reporting purposes. By default, retired components are not synchronized with CA NetOps Portal.
   - **Filtered (interface components only)**
     Indicates that the component does not pass the filter criteria and polling on the component is stopped.

NOTE

Filtered interfaces are not displayed in CA NetOps Portal dashboards and data views.

5. (Optional) Select the Interface metric family and click Update Metric Family.
   Data Aggregator reconfigures components for any configuration updates. For example, if you add a disk drive on a server, you can use the Update Metric Family button to rediscover the configuration update. The configuration update creates a disk component.
6. Click the Filter Report tab and follow these steps:
7. Look at the filters on each of the other Interface monitoring profiles to see if they are monitoring the same device collection that you want to filter.
8. Remove any relationships between other Interface monitoring profiles and device collections that will block your filter criteria. For example, if your new Interface monitoring profile is associated with the All Routers device collection, remove the relationship between *other* Interface monitoring profiles and the All Routers device collection.
9. Run another discovery and review the updated Filter report to verify that the new filter criteria is active. If the Filter report shows that an unwanted monitoring profile was included, repeat the previous steps until you are monitoring only the interfaces that you want.
   The Filter Report tab shows which interface filter criteria have been used during component monitoring. The tab also shows a report of all of the interfaces that are identified on the device and whether they matched the specified filter criteria.

   NOTE
   If you change the rules on a custom monitoring profile, the Interface Filter Criteria pane does not reflect those changes. If you disassociate the monitoring profile from a group, the Interface Filter Criteria pane does not reflect those changes. Rediscover the device to filter the interfaces that are based on the changes you made to the filter criteria and monitoring profile.

Interface Components Naming Convention

The naming convention for interface components that the Interface vendor certification or the High Speed Interface vendor certification backs is based on the following logic:

- If the ifName attribute exists and has a value, the interface uses this value for its name.
- If the ifName attribute does *not* exist or does *not* have a value, the interface uses the value of ifDescr for its name.

  NOTE
  New certifications for the Interface metric family can provide a different expression for the interface name.

Override Speed In and Speed Out Values on Interfaces

Data Aggregator provides a means for overriding the Speed In and Speed Out values for any interface to ensure utilization calculations use the appropriate values. For example, you could use the bandwidth command to configure ifSpeedIn and ifSpeedOut on your router interfaces to affect routing decisions. In this case, provide an override speed with Data Aggregator to ensure that utilization is calculated correctly.

The settings that you make on the device can change the value to one that is higher or lower than the actual available data rate. So, the utilization calculations that are made for the interface can appear inaccurate, due to this manipulation of the bandwidth. To ensure that interface utilization is calculated correctly, you want to provide an override speed on the interface within Data Aggregator.

By default, utilization is calculated using the Speed In and Speed Out values that the device, which the interface is a component of, reports. However, you can override these speed values. Reporting on interface utilization can then be more accurate.

Follow these steps:

1. Click Monitored Devices from the Monitored Inventory menu for a Data Aggregator data source.
   The Tree View tab displays.
2. Select Device by Collection or Device by Monitoring Profile from the drop-down list. Select the device that you want to override the Speed In and Speed Out values for an interface on and select the appropriate interface metric family on the Polled Metric Families tab.
   The interface components that are monitored on the device appear in the Interface Components table.
3. Select the interface component that you want to override the Speed In and Speed Out values for and click Edit.
   The Edit Interface dialog appears. The dialog displays the default discovered Speed In and Speed Out values.
4. Enter Speed In and Speed Out values in bits per second and click Save.

   NOTE
   You can remove overrides by clicking Clear and clicking Save. Going forward, bandwidth utilization charts in NetOps Portal for the interface display utilization using the speed values that the device reports. An event is generated on the interface, indicating that the speed overrides have been removed. The event can be seen in the Events Display dashboard in NetOps Portal.

   The dialog closes. The overridden Speed In and Speed Out values on the interface appear in the Interface Components table with asterisks.
   An event is generated on the interface, indicating that the Speed In and Speed Out values have been overridden on an interface. The event can be seen in the Events Display dashboard in NetOps Portal.
   Going forward, bandwidth utilization charts in NetOps Portal for the interface display utilization using the speed values that you specified.

Configure Counter Behavior

SNMP uses counters to record data points. The Data Collector polls the counters, and reports the differences in the counters over time. When the counter hits a preset limit, the counter resets to zero. The Data Collector assumes that the rollover happens no more than once per poll. If a rollover happens more than once, the resulting data is inaccurate. To resolve the issue of inaccurate data, take one of the following actions:

- Poll more often to ensure that the data is accurate
- Use a 64-bit counter instead of a 32-bit counter
- Show gaps in the data and leave out the inaccurate data

A counter can also go backwards, which indicates that the data is incorrect. The Data Collector interprets the backward behavior as a large spike in data, and discards the data point.

Bad Counter Deltas

The delta value for counters is calculated by comparing one poll cycle to the previous poll cycle. When Data Collector receives bad delta values, the values are discarded for calculation purposes. Occasionally, the counter value receives a bad poll result, and the value decreases. When the counter value shows a decrease, Data Collector skips the bad value.

When showGapsOnCounterRollover=true, the Data Collector discards the previous delta baseline. As a result, the Data Collector cannot calculate the delta for the next poll cycle.

The following table shows an example of the delta behavior when showGapsOnCounterRollover=true:

Poll Cycle	Counter Value	Delta Baseline	Delta Value	Utilization
Poll1	96	95	1	normal
Poll2	97	96	1	normal
Poll3	**30 (bad value)**	97	**null (drop)**	**gap**

Poll4	99	**reset**	**null (drop)**	**gap**
Poll5	100	99	1	normal

Show Gaps in Data

To show gaps in the data, change the default behavior. Repeat this procedure on each Data Collector.

WARNING

The following global settings apply to all devices that a particular Data Collector polls.

Follow these steps:

1. Create the following file on the Data Collector host:

   ```
   DC_install_directory/apache-karaf-<vers>/etc/
   com.ca.im.dm.snmp.collector.SnmpCollector.cfg
   ```

2. Add the following line to the file:

   ```
   showGapsOnCounterRollover=true

   showGapsOnCounterRollover32=true
   ```

   Default:
   - showGapsOnCounterRollover=false
   - showGapsOnCounterRollover32=false
3. Save the file.
   The new behavior takes effect immediately.

When you hide gaps, the following logic continues to protect against some spikes in the data:

- If a 64-bit counter appears to wrap at the 32-bit mark, the SNMP agent on the device is likely using the 32-bit counter. If the counter goes from an initial value below the maximum value of a 32-bit counter to a lower value, DX NetOps Performance Management computes the delta at the counter wrap as a 32-bit counter.
- If the delta exceeds a certain limit for either a 32-bit or a 64-bit counter, the data point is dropped.

Limit Delta Values

For 32-bit or 64-bit counters, the delta value is likely an error when both of the following criteria apply:

- A counter rollover occurs
- The delta is greater than the maximum allowable delta value
  Default: 2^32 for 32-bit; 2^63 for 64-bit

If an error occurs, the value is discarded for calculation purposes, and the report data shows a gap of one poll cycle. You can configure the behavior to increase or decrease the maximum allowable delta value. Repeat this procedure on each Data Collector.

Follow these steps:

1. Open the following file:

   ```
   DC_install_directory/apache-karaf-<vers>/etc/
   com.ca.im.dm.snmp.collector.SnmpCollector.cfg
   ```

2. To set a non-default threshold, add the following line to the file:

```
largeDeltaValueThreshold=integer

largeDeltaValueThreshold32=integer
```

Integer
Specifies the threshold for delta values.

3. **Default:**
   - largeDeltaValueThreshold32=2147483648 (0x80000000, or 2^31)
   - largeDeltaValueThreshold=9223372036854775807 (0x7fffffffffffffff, or 2^63-1)

   When a counter rollover occurs, DX NetOps Performance Management ignores delta values at or above the threshold.

Counter Rollover Log

The counter rollover log tracks detailed information about each counter rollover for all devices on a Data Collector. Use this log to verify counter behavior and troubleshoot counter issues.

Locate the log on the Data Collector that polls the relevant device:

```
/opt/IMDataCollector/apachxxxx/data/log/CounterRollover.log
```

The log contains currently applied configuration values, rollover events, IP addresses, and OIDs.

Example:

```
2016-03-07 12:54:13,480 | INFO  | ecutor-thread-58 | CounterRollover
      | .dm.snmp.rdp.impl.SnmpDeltaCache  327 | 186 - com.ca.im.data-collection-
manager.core.interfaces - 2.8.0.SNAPSHOT |  | Delta calculated is greater than or
 equal to 2147483648; dropping response: previous=1 / current=0 for ip 10.42.96.32, OID
 1.3.6.1.4.1.9.9.42.1.3.5.1.39.2222, itemID 906, in poll group 211.
```

Manage Interface Polling Behavior

If you have the Modify Component Polling role right, you can disable or enable polling capability on specific interfaces from the UI. This feature allows for more granular polling control than monitoring profile filters alone. Monitoring profile filters are limited because they can apply only to common attributes.

> **NOTE**
>
> When polling is disabled, polling is turned off and polling cannot occur. When polling is enabled, polling can occur, but this feature does not turn on polling. For polling to occur, the appropriate monitoring profiles and life cycle states must be set. For more information, see Configure Monitoring Profiles and Manage Device Life Cycles.

Disabling or enabling polling generates administrative events. For more information, see Event Types.

You can also use a Data Aggregator REST web service to disable or enable polling on other components. For more information, see Manage Polling Behavior for Components.

By default, polling is enabled for all new components. You can use a Data Aggregator REST web service to disable polling for all new components that are associated with specific metric families. For more information, see Manage Default Polling Behavior.

Follow these steps:

1. Hover over **Inventory** and click **Interfaces**.
2. Select the desired interfaces.
3. Click the **Select Polling State** drop-down and select **Disable** or **Enable**.
4. Click **OK**.
   The polling state changes.

Manage Network Flow Processing

You can enable and disable network flow processing for your Network Flow Analysis interfaces. You can also delete interfaces.

> **NOTE**
> From Network Flow Analysis v10.0.3, you can enable/ disable the interface/ device. From Network Flow Analysis v10.0.5, all the features are supported.

Perform the desired operation on the target routers or interfaces.

Navigate to the Network Flow Processing UI

Follow these steps:

1. Log in as an Administrator.
2. Hover over **Administration**, click **Data Sources**, and select the Network Flow Analysis data source.

Enable or Disable Interface

Follow these steps:

1. In the right pane, click the **All Interfaces** tab, and select the desired interfaces.
   Select multiple interfaces to perform bulk action.
2. Click **Enable** or **Disable**.
3. Click **Yes** at the prompt.

Edit an Interface

Follow these steps:

1. In the right pane, click the **Active Interfaces** tab, and select the desired interfaces.
   Select multiple interfaces to perform bulk action. However, only a few fields are editable in the bulk edit option. To edit all allowed fields, select only one interface.
2. Click **Edit Interface**.
3. Set the desired interface parameter values.
4. Click **Save**.

Delete an Interface

Follow these steps:

1. In the right pane, click the **Active Interfaces** tab, and select the desired interfaces.
   Select multiple interfaces to perform bulk action.
2. Click **Delete**.
3. Click **Yes** at the prompt.

SNMP Refresh

Follow these steps:

1. In the left pane, select the desired router.
   NOTE
   You can refresh only one router at a time.
2. Click **SNMP Refresh**.
3. Click **Yes** at the prompt.

Enable or Disable Devices (Router)

Follow these steps:

1. In the left pane, select the desired router.
   Select multiple devices to perform bulk action.
2. Click **Enable** or **Disable**.
3. Click **Yes** at the prompt.

Edit a Device (Router)

Follow these steps:

1. In the left pane, select the desired router.
   Select multiple routers to perform bulk action. However, only a few fields are editable in the bulk edit option. To edit all allowed fields, select only one router.
2. Click **Edit**.
3. Set the desired router parameter values.
4. Click **Save**.

Delete a Device (Router)

Follow these steps:

1. In the left pane, select the desired router.
   Select multiple devices to perform bulk action.
2. Click **Delete**.
3. Click **Yes** at the prompt.

Configure Round Trip Time (RTT) Tests

To enable round-trip time tests, configure the tests using the Data Aggregator REST API. DX NetOps Performance Management supports the following test types:

- **DNS**
  Measures the DNS lookup time.
- **ICMP Echo (Ping)**
  Measures the round-trip delay for the full path.
- **ICMP Path Echo**
  Measures the round-trip delay and hop-by-hop round-trip delay.
- **ICMP Jitter**
  Measures the hop-by-hop jitter, packet loss, and delay measurement statistics in an IP network.
- **HTTP**

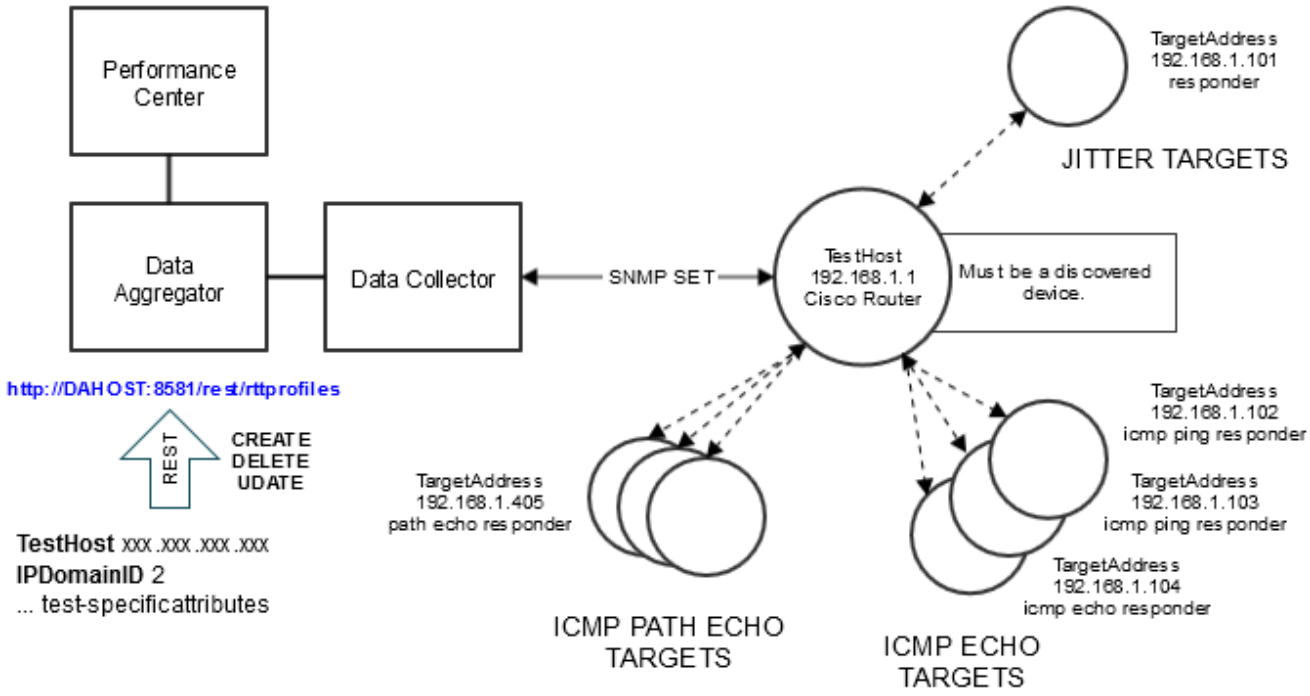
Measures the round-trip time to retrieve a web page.

- **TCP Connect**
  Measures the time to connect to a target device with TCP.
- **UDP Jitter**
  Measures the round-trip delay, one-way delay, one-way jitter, and one-way packet loss. Codec simulation is provided for MOS and ICPIF voice quality scoring capability.

RTT Test Architecture

The following diagram shows how the various components participate in the RTT test feature:

Figure 8: RTT Test Architecture



Configure the Tests

To configure RTT tests, complete the following steps:

Gather Required Information

The following information is required to configure a test:

- Required for all RTT tests in your environment:
  - DA host/address
  - IpDomainId

    NOTE

    To determine the IpDomainId, use the Data Aggregator REST API.
- Required for the specific test:
  - TestHost

The IP address of the device where DX NetOps Performance Management creates the test.

– Depending on the test type, one of the following targets is required:
  - TargetAddress
    The device that receives the test probes.
  - (DNS) TargetAddressString
    IP address or hostname of the target that receives the probes. Use this string instead of TargetAddress.
  - (HTTP) Url
    The URL that the HTTP probe communicates with.

Create an SNMP "SET" Profile

Create an SNMP profile and specify **Yes** for **Use in SNMP SET**.

For more information, see SNMP Profiles.

Discover Devices Capable of Running RTT Tests

Apply a monitoring profile with the IPSLA-related metric families that you want to monitor to the target devices. Then, run discovery for those devices.

For more information, see IPSLA Polling.

Associate SET Profiles with the Devices

For each device, associate the SET profile to the device item.

Through the UI:

1. Go to **Administration**, and click the Data Source.
2. Select the monitored device.
3. Select the SNMP SET Profile and the SNMP SET Version.
4. Click **Save**.
   You can now use DX NetOps Performance Management to configure RTT tests on the device.

Through the Data Aggregator API:

Documentation: `http://DA_HOST:8581/rest/devices/manageable/documentation`
PUT URL: `http://DA_HOST:8581/rest/devices/manageable/<device item id>`
Headers: [Content-Type=application/xml]
Body Syntax:

```
<ManageableDevice version="1.0.0">
  <RWSNMPProfileID>DA_RWSNMP_PROFILE_ID</RWSNMPProfileID>
  <RWSNMPProfileVersion>SNMP_VERSION</RWSNMPProfileVersion>
</ManageableDevice>
```

TIP

To find the Item ID of an SNMP profile, go to the following REST URL: `http://DA_HOST:8581/rest/profiles/snmpv1`

Example: This example associates the SNMP Profile identified by Item ID 736 with device 1144:

PUT URL

```
http://DA_host:8581/rest/devices/manageable/1144
```

Body

```
<ManageableDevice version="1.0.0">

  <RWSNMPProfileID>736</RWSNMPProfileID>

  <RWSNMPProfileVersion>SNMPV1</RWSNMPProfileVersion>

</ManageableDevice>
```

Configure the Tests

Use a REST client to create RTT profiles. DX NetOps Performance Management uses the RTT profiles to configure the tests on the target devices. Each profile initiates the specified action type to create, update, or delete a test on the device. The profile is not used again after DX NetOps Performance Management performs the specified action. Old RTT profiles are deleted.

For information about how to configure the RTT profiles, see RTT Configuration Examples.

RTT Configuration Details

Use a REST client to issue REST POST requests to create, delete, or update tests.

Endpoints

```
http://DA_host:8581/rest/rttProfiles/TestType
```

The following example shows the basic structure of the POST REST request:

```
<RTT_Profile version="1.0.0">
   <Item version="1.0.0">
    <Name>...</Name>
   </Item>


   <AttrGroupList>
     <AttrGroup>
       <ActionType>...</ActionType>
       <TestHost>...</TestHost>
       <IPDomainID>...</IPDomainID>
       <OptionalAttribute0>...</OptionalAttribute0>
       <OptionalAttribute1>...</OptionalAttribute1>
     </AttrGroup>
  </AttrGroupList>


</RTT_Profile>
```

RTT_Profile specifies the test type and determines which attributes are required and optional. Use one of the following values:

- DNSRoundTripTestProfile
- IcmpEchoRoundTripTestProfile
- IcmpPathEchoRoundTripTestProfile
- ICMPJitterRoundTripTestProfile
- HTTPRoundTripTestProfile
- TCPRoundTripTestProfile
- JitterRoundTripTestProfile

Action Types

The following table provides details about the operations that you use to configure the tests:

Logical Operation	REST Operation	Action Type	Endpoint	Description
Create	POST	CREATE	/rest/rttprofiles /<*TestType*>	Creates a test profile.
Force Create	POST	FORCE_CREATE	/rest/rttprofiles /<*TestType*>	Creates a test profile, but does not verify uniqueness.
Delete	POST	DELETE	/rest/rttprofiles /<*TestType*>	Deletes a test profile, including all the associated test instances.
Update	POST	UPDATE	/rest/rttprofiles /<*TestType*>	Deletes a test profile, and creates a new test profile that uses the same underlying MIB object index.
Get all test profiles	GET	GET	/rest/rttprofiles /<*TestType*>	Fetches all profiles of the specified test type.
Get specific test profile	GET	GET	/rest/rttprofiles /<*TestType*>/<*ID*>	Fetches a profile with the specified type and ID.

TestType specifies the RTT test type. Use one of the following values:

- dns
- icmpecho
- icmppathecho
- icmpjitter
- http
- tcp
- jitter

CREATE

This operation creates an instance of a test profile of the specified type with ActionType equal to CREATE. The created profile conceptually represents a background job to perform an SNMP set to create an RTT test on a device. The test is created permanently.

The create operation succeeds only if:

- The TestHost is a discovered device.
- The attributes of the requested test do not match an existing test in DX NetOps Performance Management.
- The attributes are semantically correct. The TargetHost does not reject the attributes because values are incorrect or the combination of attributes is invalid.

The result of a CREATE request is reflected in the read-only <Result> attribute of the created test. Perform a GET to view the result.

FORCE_CREATE

This operation is similar to CREATE, but creates the test even if an existing test has the same attribute values.

DELETE

This operation deletes a test that is identified by ItemIDs or by supplied attributes.

Deletes all tests that match the supplied attributes or the specific tests that matches the ItemIDs.

ItemIDs supersede attributes.

UPDATE

This operation re-applies attributes to an existing test. The operation deletes the test, and creates a test with new attributes. UPDATE changes only the supplied attributes. UPDATE runs DELETE, then CREATE. To preserve the relationship between the test configuration and the discovered Response Path Test components, the new test reuses the underlying MIB object index.

Attributes

NOTE

Depending on the Cisco IOS, some attributes may not apply to your device.

The following attributes apply to all test types:

- **ActionType** The action type of RTT test.
- **TestHost**
  The address of the device on which the test runs.
- **ItemID**
  Specifies the ID of the target test for DELETE and UPDATE actions. To get the ItemID, see Get the ItemID for a Test
- **IPDomainID** Specifies the IP Domain of the test. IP Domain is required for all actions that do not include ItemID.
  - CREATE always requires IPDomainID.
  - DELETE requires IPDomainID to delete a test without specifying the ItemID.
  - UPDATE never requires IPDominID.
- (Optional) **Owner**
  Specifies the test owner.
- (Optional) **Tag**
  A short string that identifies the test in logging and notification.
- (Optional) **Threshold**
  Specifies that the test generates a threshold event if test takes longer than specified milliseconds.
- (Optional) **Frequency**
  Duration in seconds between initiating each RTT test.
- (Optional) **Timeout**
  Duration in milliseconds to wait for an RTT operation completion.
- (Optional) **VrfName**

Specifies the VPN name where the RTT operation is used . The agent uses this field to identify the VPN routing table for the operation.

- (Optional) **Persist**
  Indicates whether this test configuration should be saved when persisting agent configuration to non-volatile storage. If left unspecified, the default is true.

  NOTE

  Cisco recommends persisting the tests. If the configuration is lost on the device, DX NetOps Performance Management does not re-provision the test.

This table summarizes the attributes that are associated with each IPSLA test. R indicates Required, and O indicates Optional:

Parameter	Description	ICMP Echo (Ping)	ICMP Path Echo	ICMP Jitter	UDP Jitter	UDP Jitter (VoIP)	DNS	TCP	HTTP
SourceAddress	The address of the device on which the test runs.	O	O	O	O	O	O	O	O
SourcePort	Specifies the source address port number. If the port is unspecified, the system selects a port.	N/A	N/A	N/A	O	O	O	O	O
TargetAddress	The destination IP addresses of the RTT test.	R	R	R	R	R	N/A	R	N/A
TargetPort	The destination port to which test probes are sent.	N/A	N/A	N/A	R	R	N/A	R	N/A
RequestSize	The request probe payload size.	(28)	(28)	N/A	(32)	(32)	N/A	N/A	N/A
ResponseSize	The response probe payload size.	N/A	O	N/A	O	O	N/A	N/A	N/A

TypeOfService	The type of service octet in an IP header.	O	O	O	O	O	N/A	O	O
Interval	The inter-packet delay in milliseconds between packets.	N/A	N/A	N/A	O	O	N/A	N/A	N/A
NumPackets	Number of packets to transmit.	N/A	N/A	N/A	O	O	N/A	N/A	N/A
CodecType	The codec type to use with jitter probe.	N/A	N/A	N/A	N/A	O	N/A	N/A	N/A
CodecInterval	The inter-packet delay in milliseconds between packets. Valid only for jitter probe which uses CodecType.	N/A	N/A	N/A	N/A	O	N/A	N/A	N/A
CodecPayload	Number of octets to place into the Data portion of the message. Valid only for jitter probe which uses CodecType.	N/A	N/A	N/A	N/A	O	N/A	N/A	N/A
CodecNumPackets	Number of packets to transmit . Valid only for jitter probe which uses CodecType.	N/A	N/A	N/A	N/A	O	N/A	N/A	N/A
ICPIFAdvFactor	Used while calculating jitter ICPIF values.	N/A	N/A	N/A	N/A	O	N/A	N/A	N/A

TargetAddressString	Specifies the address of the target. This string can be an IP address or a hostname.	N/A	N/A	N/A	N/A	N/A	R	N/A	N/A
NameServer	Specifies the IP address of the name-server.	N/A	N/A	N/A	N/A	N/A	R	N/A	N/A
Url	Specifies the URL that the HTTP probe targets.	N/A	N/A	N/A	N/A	N/A	N/A	N/A	R
Operation	Specifies the HTTP operation that represents the specific type of RTT operation.	N/A	N/A	N/A	N/A	N/A	N/A	N/A	R
HTTPVersion	Specifies the version number of the HTTP server.	N/A	N/A	N/A	N/A	N/A	N/A	N/A	O
String1	Stores the content of a raw HTTP request. If the request cannot fit into String1, use more String attributes. *Required only if **Operation** is set to **httpRaw (2)**.	N/A	N/A	N/A	N/A	N/A	N/A	N/A	R*
String2	Continues the raw HTTP request from **String1**.	N/A	N/A	N/A	N/A	N/A	N/A	N/A	O

String3	Continues the raw HTTP request from **String2**.	N/A	N/A	N/A	N/A	N/A	N/A	N/A	O
String4	Continues the raw HTTP request from **String3**.	N/A	N/A	N/A	N/A	N/A	N/A	N/A	O
String5	Continues the raw HTTP request from **String4**.	N/A	N/A	N/A	N/A	N/A	N/A	N/A	O

NOTE

When you specify an attribute value that has an associated enumerated name as defined by the MIB, only the numeric value is recognized.

Result

The result attribute is a read-only attribute that indicates the outcome of the request:

Meaning	<Result> value
The operation is in progress.	PENDING
The operation succeeded.	SUCCESS
The TestHost is not discovered.	FAILURE: Unable to identify device at 192.168.96.1
The requested test is already configured.	ALREADY_EXISTS: Existing Count = 1
TestHost is unreachable.	FAILURE: {http://im.ca.com/normalizer}NormalizedJitterInfo.RttType: NO_RESPONSE - {http://im.ca.com/certifications/snmp}CiscoIPSLAJitterMib.rttMonCtrlAdminRttType: SNMP timeout {http://im.ca.com/normalizer}NormalizedJitterInfo.TargetAddress: NO_RESPONSE -{http://im.ca.com/certifications/snmp}CiscoIPSLAJitterMib.rttMonEchoAdminTargetAddress: SNMP timeout {http://im.ca.com/normalizer}NormalizedJitterInfo.Interval: NO_RESPONSE -{http://im.ca.com/certifications/snmp}CiscoIPSLAJitterMib.rttMonEchoAdminInterval: SNMP timeout {http://im.ca.com/normalizer}NormalizedJitterInfo.CodecInterval: NO_RESPONSE - {http://im.ca.com/certifications/snmp}CiscoIPSLAJitterMib.rttMonEchoAdminCodecInterval: SNMP timeout

Invalid credentials, such as SNMP SET profile.	{http://im.ca.com/normalizer}NormalizedICMPInfo.RttType: OPERATION_NOT_ALLOWED - {http://im.ca.com/certifications/snmp}CiscoRttMonStatsMib.rttMonCtrlAdminRttType: SNMP error 6: No access{http://im.ca.com/normalizer}NormalizedICMPInfo.TargetAddress: ATTRIBUTE_NOT_SET{http://im.ca.com/normalizer}NormalizedICMPInfo.Owner: ATTRIBUTE_NOT_SET{http://im.ca.com/normalizer}NormalizedICMPInfo.Tag: ATTRIBUTE_NOT_SET
The Data Collector associated with the TestHost is down.	• SOURCE_NOT_AVAILABLE • SHUTDOWN
The test was not present on the TestHost because another SNMP client deleted the test.	FAILURE: 1.3.6.1.4.1.9.9.42.1.2.1.1.9.546811228: Commit failed. Error Index: 1

Get the ItemID for a Test

The UPDATE and DELETE operations use ItemID to identify tests. To get the ItemID, use a filtered REST endpoint:

- *http://DA_HOST:8581/rest/responsepathtest/filtered*
  Searches all test types.
- *http://DA_HOST:8581/rest/responsepathicmp/filtered*
  Searches ICMP Echo tests.
- *http://DA_HOST:8581/rest/responsepathecho/filtered*
  Searches ICMP Path Echo tests.
- *http://DA_HOST:8581/rest/responsepathjitter/filtered*Searches UDP Jitter tests.
- http:// *DA_HOST:8581*/rest/responsepathicmpjitter/filtered
  Searches ICMP Jitter tests.
- http://*DA_HOST*:8581/rest/responsepathhttp/filtered
  Searches HTTP tests.
- http://*DA_HOST*:8581/rest/responsepathtcp/filtered
  Searches TCP Connect tests.
- *http://DA_HOST:8581/rest/responsepathdns/filtered*
  Searches DNS tests.

Example Request

This example shows a filtered request for tests with the specified target:

```
<FilterSelect xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
 xsi:noNamespaceSchemaLocation='filter.xsd'>
  <Filter>
    <ResponsePathTest.Target type='EQUAL'>10.42.94.250</ResponsePathTest.Target>
  </Filter>
  <Select use="exclude"/>
</FilterSelect>
```

TIP

To exclude all attributes except ItemID, the following statement:

```
Select use="exclude"
```

Example Response

This example shows the response to a filtered request:

```
<ResponsePathIcmpList>
    <ResponsePathIcmp version="1.0.0">
        <ID>955</ID>
    </ResponsePathIcmp>
    <ResponsePathIcmp version="1.0.0">
        <ID>956</ID>
    </ResponsePathIcmp>
</ResponsePathIcmpList>
```

RTT Configuration Examples

Use these examples to help configure your tests:

RTT REST Configuration Example Script

The following script creates an example POST. Use the example as a template to create, delete, and modify RTT tests.

rtt_rest_configuration_examples.py

```
$ ./rtt_rest_configuration_examples.py
Usage: rtt_rest_configuration_examples.py [--quiet] [--dry-run] <DA host> create|delete
 ping|jitter [<attribute_name> <attribute_value> ...]


Options
      --quiet      Suppress output
      --dry-run    Construct REST request, but do not send


    Test     Attributes
    --------
 ------------------------------------------------------------------------------------------
    icmpecho DeviceIp IpDomainId Owner Threshold Tag Frequency Timeout TargetAddress
 RttTestName SourceAddress
    jitter   DeviceIp IpDomainId Owner Threshold Tag Frequency Timeout TargetAddress
 RttTestName SourceAddress


Examples
    % ./rtt_rest_configuration_examples.py testda1.ca.com create icmpecho DeviceIp
 10.165.170.222 IpDomainId 2 TargetAddress 10.132.159.140 SourceAddress 10.117.139.209
 Frequency 180
    % ./rtt_rest_configuration_examples.py testda1.ca.com delete jitter DeviceIp
 10.103.224.132 IpDomainId 2 TargetAddress 10.132.159.140


(See script comments for more examples)
```

Example RTT Configurations

ICMPECHO CREATE

This example creates an ICMP Echo test:

```
<IcmpEchoRoundTripTestProfile version="1.0.0">
   <Item version="1.0.0">
    <Name>Example ICMP Echo Test</Name>
   </Item>

   <AttrGroupList>
     <AttrGroup>
       <ActionType>CREATE</ActionType>
       <TestHost>10.250.134.47</TestHost>
       <IPDomainID>2</IPDomainID>
       <TargetAddress>10.0.63.106</TargetAddress>
       <Frequency>62</Frequency>
       <Timeout>5000</Timeout>
       <Threshold>5000</Threshold>
       <Owner>tedison</Owner>
       <Tag>ECHO02</Tag>
       <RequestSize>40</RequestSize>
     </AttrGroup>
  </AttrGroupList>

</IcmpEchoRoundTripTestProfile>
```

ICMPECHO DELETE

This example deletes three ICMP Echo test with the specified ItemIDs:

```
<IcmpEchoRoundTripTestProfile version="1.0.0">
   <Item version="1.0.0">
    <Name>Delete ICMP Echo Test</Name>
   </Item>

   <AttrGroupList>
     <AttrGroup>
       <ActionType>DELETE</ActionType>
       <ItemId>800<ItemId>
     </AttrGroup>
     <AttrGroup>
       <ActionType>DELETE</ActionType>
       <ItemId>2311<ItemId>
     </AttrGroup>
     <AttrGroup>
       <ActionType>DELETE</ActionType>
       <ItemId>1021<ItemId>
     </AttrGroup>
  </AttrGroupList>

</IcmpEchoRoundTripTestProfile>
```

ICMPECHO UPDATE

This example updates the TargetAddress and Owner for the ICMP Echo test with the specified ItemID:

```
<IcmpEchoRoundTripTestProfile version="1.0.0">
   <Item version="1.0.0">
    <Name>Update ICMP Echo Test</Name>
   </Item>

   <AttrGroupList>
     <AttrGroup>
       <ActionType>UPDATE</ActionType>
       <ItemID>921</ItemID>
       <TargetAddress>10.52.217.32</TargetAddress>
       <Owner>admin_andy</Owner>
     </AttrGroup>
  </AttrGroupList>

</IcmpEchoRoundTripTestProfile>
```

ICMPPATHECHO CREATE

This example creates an ICMP Path Echo test:

```
<IcmpPathEchoRoundTripTestProfile version="1.0.0">
   <Item version="1.0.0">
    <Name>Example ICMP Path Echo Test</Name>
   </Item>

   <AttrGroupList>
```

```
      <AttrGroup>
        <ActionType>CREATE</ActionType>
        <TestHost>10.250.134.47</TestHost>
        <IPDomainID>2</IPDomainID>
        <TargetAddress>10.40.29.130</TargetAddress>
        <Owner>att</Owner>
        <Tag>seattle1</Tag>
        <Frequency>180</Frequency>
        <Timeout>7777</Timeout>
        <Threshold>6666</Threshold>
      </AttrGroup>
   </AttrGroupList>


 </IcmpPathEchoRoundTripTestProfile>
```

ICMPPATHECHO DELETE

This example deletes an ICMP Path Echo test with the specified attributes:

```
 <IcmpPathEchoRoundTripTestProfile version="1.0.0">
    <Item version="1.0.0">
     <Name>Example ICMP Path Echo Test</Name>
    </Item>


    <AttrGroupList>
      <AttrGroup>
        <ActionType>DELETE</ActionType>
```

```
        <TestHost>10.250.134.47</TestHost>
        <IPDomainID>2</IPDomainID>
        <TargetAddress>10.0.63.106</TargetAddress>
      </AttrGroup>
   </AttrGroupList>

</IcmpPathEchoRoundTripTestProfile>
```

JITTER CREATE

This example creates two Jitter tests:

```
<JitterRoundTripTestProfile version="1.0.0">
    <Item version="1.0.0">
     <Name>Create Jitter Test</Name>
    </Item>

    <AttrGroupList>
      <AttrGroup>
        <ActionType>CREATE</ActionType>
        <TestHost>10.188.72.48</TestHost>
        <IPDomainID>2</IPDomainID>
        <TargetAddress>10.16.75.221</TargetAddress>
        <TargetPort>33333</TargetPort>
        <Owner>admin_dan</Owner>
        <Tag>JITTER022</Tag>
      </AttrGroup>
```

```
      <AttrGroup>
        <ActionType>CREATE</ActionType>
        <TestHost>10.84.206.153</TestHost>
        <IPDomainID>2</IPDomainID>
        <TargetAddress>10.42.96.10</TargetAddress>
        <Owner>admin_steve</Owner>
        <Tag>JITTER023</Tag>
        <Frequency>120</Frequency>
        <Timeout>10000</Timeout>
        <CodecType>1</CodecType>
        <CodecInterval>20000</CodecInterval>
        <CodecPayload>172</CodecPayload>
        <CodecNumPackets>100</CodecNumPackets>
      </AttrGroup>

   </AttrGroupList>

 </JitterRoundTripTestProfile>
```

JITTER DELETE

This example deletes a Jitter test with the specified attributes:

```
 <JitterRoundTripTestProfile version="1.0.0">
    <Item version="1.0.0">
     <Name>Example Jitter Test</Name>
```

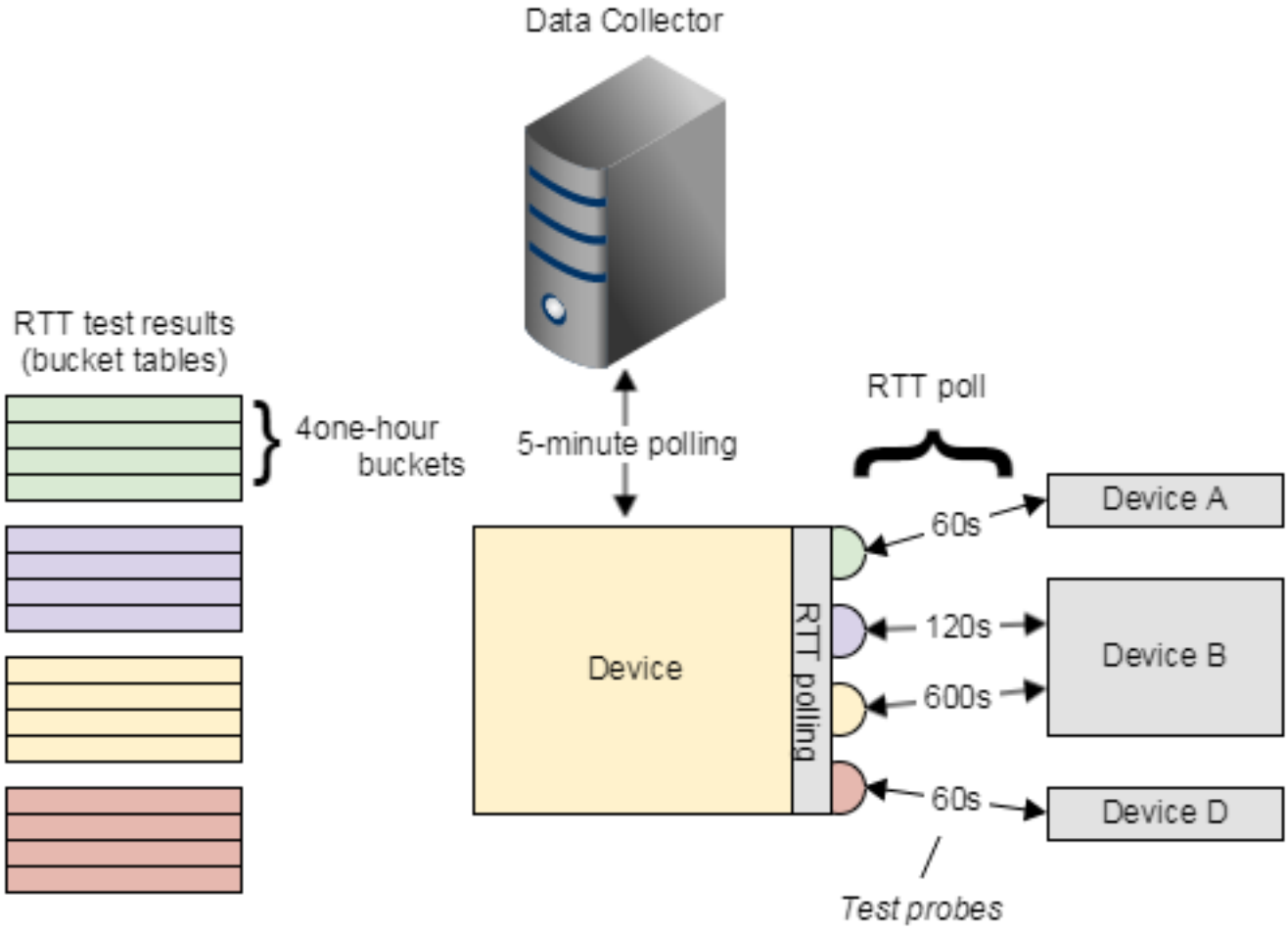
```
    </Item>


    <AttrGroupList>
      <AttrGroup>
        <ActionType>DELETE</ActionType>
        <TestHost>10.250.134.47</TestHost>
       <IPDomainID>2</IPDomainID>

        <TargetAddress>10.237.30.166</TargetAddress>
      </AttrGroup>
   </AttrGroupList>


 </JitterRoundTripTestProfile>
```

IPSLA Polling

The Data Aggregator and Data Collectors poll IPSLA-enabled devices, which can run RTT tests. An RTT test frequency controls how often an active test runs (for example, every 60 seconds). While a test is active its results are stored in a statistics table. Each row represents one hour of results for a specific test. Test result metrics (for example, `rttMonStatsCollectTimeouts`) are stored as counters and increment after each execution of the test. At the end of an hour a new bucket is created. When the maximum number of buckets is reached, the oldest bucket is dropped. Metrics for new buckets are reset to zero and begin to increment. Disabled tests stop counting. The DX NetOps Performance Management poll rate controls how often the Data Collector requests these statistics. Each time the Data Collector polls the counter, it subtracts the last value from the current to produce a delta value for a given metric. For more information about setting the poll rate, see Configure Monitoring Profiles.

The following diagram illustrates IPSLA polling:

Figure 9: IPSLA Polling



Using

DX NetOps Performance Management monitors the health of your environment, including networks, applications, and devices. To display the relevant data, DX NetOps Performance Management uses dashboards and context pages. Both types of pages include views that show different information about your infrastructure.

Launch NetOps Portal

NetOps Portal is the web user interface for DX NetOps Performance Management. Use NetOps Portal to view infrastructure data, configure, collect, and perform administrative tasks.

NetOps Portal is supported in modern web browsers.

The following video shows the login process and provides more information about the authentication process:

Follow these steps:

1. Open a web browser.

2. In the address field, enter the following address: `http://PC_host:8181`
   - **PC_host** The IP address or hostname of the NetOps Portal host
   - **8181**
     The default port number for NetOps Portal
3. Specify your NetOps Portal username and password.
4. (Optional) Select **Remember me on this computer** to save your login session for 15 days. This option prevents your session from being deleted when you log out, time out, or you close the browser.
5. Click **Log In**.
   The NetOps Portal console opens to your home dashboard.

Search and Filter in Performance Center

Global search locates items in DX NetOps. For high scale environments, you can limit the scope and improve performance using the search controls that DX NetOps includes.

In this article:

Global Search

Global searches return lists of the items in the inventory that match the search string. The lists are sorted by item type (such as devices, interfaces, device components).

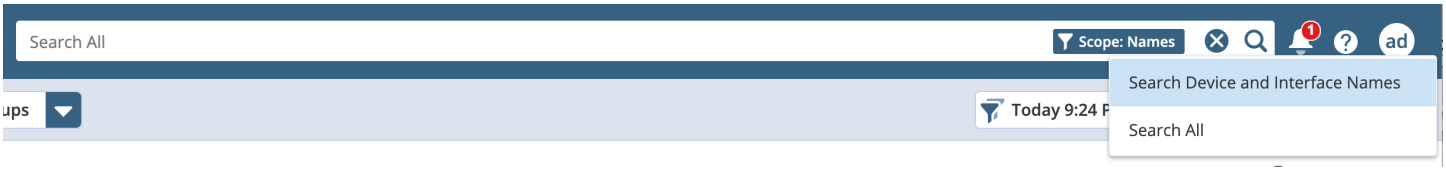
To perform a global search, click **Search** (the magnifying glass) at the top of any page. The following image shows an example of the global search bar:

Figure 10: Global_Search_bar

By default, the global search is set to search only device and interface names that contain the search string (Scope: Names).

To widen the search scope and search all managed items (for example, to search for tunnels) or search across other identifying information about an entity (for example, to search for ip-addresses) by selecting **Scope** in the search box, and then selecting **Search All** (Scope: All) from the drop-down. The following image shows the drop-down:

Figure 11: Global_Search_dropdown



Limit Global Search

You can limit the search scope in the **Search All** context using filter expressions. In the most basic form, the filter expression uses the following syntax:

```
ViewName:ColumnName=value
```

To search multiple views or columns, use the following syntax:

```
ViewName1:Column1;ColumnN&ViewName2:Column1;ColumnN=value
```

Filter Expression Examples

The following are examples of filter expressions you can use to limit the search scope:

Example: Search Devices Containing the String ‘foo’ Anywhere in the ‘Name’ Column

Use the following search expression:

```
Devices:Name=foo
```

Example: Search Interface Names or Descriptions and Devices that Include ‘foo’

Use the following search expression:

```
Interfaces:Interface Name;Description&Devices:Name=foo
```

Example: Search All Views

Use the following search expression:

```
Name;Description=foo
```

If the view does not include these columns, the expression searches all columns of this view.

Example: Search All Columns for a View

Use the following search expression:

```
Interfaces:*=foo
```

Use Wildcards in Filter Expressions

By default, global search searches for text that is contained in an item string. For example, searching for 'foo' or 'oo' returns ‘Foo’ as a result. To narrow or broaden your search, for example, to locate column names, add the asterisk (*) wildcard character to the filter expression.

You can search within interfaces view columns that contain ‘Name’--such as “Name Alias”, “Interface Name”, “Device Name Alias”--using the following filter expression:

```
Interfaces:*Name*=value
```

Using asterisks at the beginning and end of a keyword (‘contains’ searches), for example ‘*Name*’, is equivalent to entering ‘Name’ as the search string.

> **NOTE**
> ‘Starts with’ searches can return quicker results than ‘contains’ searches. For example, ‘foo*’ (starts with ‘foo’) executes faster than the ‘*foo*’ (contains ‘foo’) search.

To narrow the search, add multiple search words. For example, searching for devices using the ‘server 192.168*’ search string returns all servers on the 192.168.0.0/16 network.

Other Wildcard Examples

- Return all rows with entries that start with 'serv':
  `serv*`
- Return all rows with entries that end in ‘erver’:
  `*erver`
- Return all rows with entries that start with ‘fo’ and end with ‘200’. For example, the following filter expression returns ‘Foo_5976_10.92.200.200’, but it does not return ‘Foo_5976_10.92.200.201’:
  `fo*200`
- Return all rows with entries that:

- Start with ‘fo’
- Contain at least one character following 'fo'
- Contain '200'
- End with any character

```
fo*200*
```

Quick Filter

You can limit the contents of many table views to only those items that contain the search criteria that you enter using the quick filter. Quick filtering a table (for example, an inventory table that shows devices) is similar to performing a global search and looking at the results for that kind of managed item.

The following image shows an example of a quick filter:

Figure 12: Global_Search_bar

To quick filter a table view, type the search string into the quick filter box, and then either click filter (the funnel icon) or press the Return key on the keyboard.

To clear a quick filter, delete the search term, and then click filter (the funnel icon) or press the Return key on the keyboard.

Customize Your User Settings

You can customize your user account's default dashboard and personal settings.

In this article:

Set a Dashboard as Your Home Page

To log in to your preferred dashboard, set that dashboard as your home page. By default, the first dashboard in your first menu is your home page.

> **TIP**
> To return to your home page from any other page, click the logo in the upper left corner.

The following video shows how to set your default dashboard:

Follow these steps:

1. Navigate to the dashboard that you want to set as your home page.
2. (Optional) To set a specific context as your default, click the **[change]** link, and then select the group context for the dashboard. The home page saves the context.

   > **NOTE**
   > If the selected group is removed from your permission set, your default permission group is used for the dashboard context.
3. Click **More**, and then click **Set as Home Page**.
4. In the confirmation dialog, click **Yes**.

The selected dashboard is now your home page.

Customize your User Account Settings

User accounts include settings that you can customize, such as the preferred language and the time zone. Your role rights determine the settings that you can customize. If your user account has the required role right, you can change the settings.

Follow these steps:

1. Click the name of your user account in the upper-right corner, and then click **User Settings**.
2. Modify your user account settings:
   - **Preferred Language**
     Specify a language for the NetOps Portal user interface (UI). NetOps Portal displays the selected language regardless of the language selected for the operating system or for the browser language.

     NOTE
     For a language to display appropriately, the relevant fonts must be installed.

     For more information about the languages that DX NetOps Performance Management supports, see Language Support.
   - **Email Address**
   - **Time Zone**
     The default time zone is UTC (Coordinated Universal Time).

     NOTE
     Changing the time zone after email schedules are set up in NetOps Portal can cause incorrect times to appear in the Scheduled Emails UI.
   - **Time Display Format**
     Select the default time format, either 12 hours or 24 hours.
   - **Default Group**
     This group is the default context when you log in. The list only includes groups from your permission groups.
   - Select one of the following options from the **View Suppression** drop-down:
     View suppression hides views when the required data source is not registered or when a required technology is not configured. Similar behavior applies to context tabs and custom menus. Menus and tabs that include only suppressed views are hidden. View suppression applies to the default views on out-of-the-box dashboards. View suppression does not hide views from administrators when they use the view categories to edit a dashboard.
     When the data source that populates a view is registered, that view is no longer hidden.
     - **Suppress Views**
       View suppression is enabled and views are hidden by default. You can disable it for troubleshooting purposes, or to help you decide whether to deploy another data source.
     - **Display All Views**
       View suppression is disabled and all views appear.
   - Select one of the following options from the **Item Name Display Setting** drop-down:
     This option determines whether device and interface names appear as the display name or as the alias in dashboards and views. By default, all users see the display name.

     NOTE
     This option does *not* control how devices and interfaces appear in the device or interface inventory lists.

     To modify this option, your user account requires the View Item Display Name or Name Alias role right.
     - **Use Display Name**
     - **Use Item Name Alias**
3. Click **Save**.

Change User Password

You can change the password for your user account.

By default, user passwords must meet the following requirements:

- Be different than the username
- Minimum length of 8 characters
- Maximum length of 30 characters
- Contain at least 3 of the following types of characters:
  - Special characters
  - Uppercase letters (A-Z)
  - Lowercase letters (a-z)
  - Numbers (0-9)

If desired, you can disable these requirements.

Follow these steps:

1. Click the name of your user account in the upper-right corner and click **Change Password**.
2. Enter the old password.
3. Enter the new password.
4. Confirm the new password.
5. Click **Save**.

Auto Refresh

You can turn on (enable) and off (disable) auto refresh using the option in the upper-right corner of the NetOps Portal UI. Enabling auto refresh displays the most recent data. Auto refresh defaults to refresh every 60 seconds, however, you can adjust this setting. For example, if you set the poll cycle to 5 minutes, you can work with CA Support to set auto refresh to 5 minutes.

> **NOTE**
> You can set auto refresh only to a rate that is higher than the poll cycle.

Share Data with Other Users

You can create or access reusable On-Demand report templates, and download a report for sharing. You can also access a dashboard and download it for sharing.

> **NOTE**
> When you generate CSV output, the following leading characters are removed from cells: (+, -, @, =).

Multiple options let you share dashboards and views. To share dashboards and views:

- Export a dashboard to a static report in PDF format. The PDF includes only the information currently visible in your dashboard.
- Set up schedules to send reports automatically.
- Export individual views. Publish views on a web page, such as an intranet site.
- Export data from a view to a file in CSV format. The CSV includes only the information up to the row limit (1,000 for custom views and 5,000 for common views).

For all data-export options, certain user account role rights are required.

When you send or schedule a report by email, you can choose the following format options:

- **PDF**
  - Select **Portrait** or **Landscape** to specify the page layout of the PDF document.
  - Select whether to run the report on only the **First Page** or **All Pages**.

For **First Page**, only the first page is attached and sent in an email message. For **All Pages**, an email message is sent with a time-sensitive link to the report containing all pages.

- **CSV**
  Select **Scaled** or **Unscaled** to specify whether the values in the exported dashboards are scaled.
  For **CSV**, a file is attached that contains all the data for each view based on the limit for the view.

NOTE

By default, each metric in a trend view appears as its own table in the CSV output. CSV reports can support having a column for each metric included in a trend view. To include a column for each metric in a trend view of a CSV report, reach out to CA Support.

WARNING

Running reports on All Pages can significantly impact performance. To minimize the performance impact, adhere to our best practices.

The following best practices apply to running reports on All Pages:

- Before you schedule reports on All Pages, ensure all your servers, especially the Data Repository, meet the minimum requirements and sizing guidelines.For information about the sizing requirements, see the DX NetOps Performance Management Sizing Tool.
- Schedule reports on All Pages to run during non-business hours.
- Run reports on All Pages only as a user *without* the High Resolution role right.
- Running reports on All Pages for As Polled data can be problematic especially for trend views and group scorecard trend views.
- Do not run reports on All Pages for the context page of a site group.
- Before you schedule a report to run on All Pages, try running it in isolation to understand its performance impact.

If a report on All Pages times out, when you click the link for the report, the PDF contains a "Query failed" error message. If a report times out, see Report on All Pages Times Out.

The System Status page includes details about the Report Generation Services. For more information, see View the Health of the System

Print a Report to PDF or CSV

If your user account has the required role right, you can export the current dashboard contents as a printed report. The Print feature lets you preview the current dashboard page in PDF or CSV format.

NOTE

When exporting Chart/Table views, the PDF does not show the gauge or chart that is associated with the currently selected row in the table.

Follow these steps:

1. Do one of the following tasks:
   - Hover over **Reports**, and click **On-Demand Report Templates**, and run an existing report template.
   - Hover over **Dashboards**, and select a dashboard.
2. Click the **Print** icon on the toolbar.
3. Click **Print PDF** or **Print CSV**:
   - **Print PDF**
     Select **Portrait** or **Landscape** to specify the page layout of the PDF document.
   - **Print CSV**
     Select **Scaled** or **Unscaled** to specify whether the values in the exported dashboards are scaled.

Scaled values appear with larger units, for example, 1 KB. Unscaled values appear in the raw form for the metric, for example, 1000 bytes.

4. Save the file to the local computer using the options that you selected.

Send or Schedule a Report by Email

If your user account has the required role right, you can export dashboard contents as a report attached to an email message. The Email feature lets you specify the email address of the recipient and the subject of the message. For CVS and single-paged PDF, the file is attached and sent in an email message. For full-paged PDF, an email message is sent with a time-sensitive link to the report. Download and save the report as soon as possible. Reports sent or scheduled by email include only the information up to the row limit. For example, the limit for the IM Table (Top) view is 2,500 by default.

When an IM Table (Top) view contains a mix of core and baseline metrics, the results might be capped at 1,000 rows. This reduced row limit depends on the number of queried items. The following warning message appears, “Selected metric has a sum roll-up strategy. Detailed projections support only averaged metrics.”

You can send a report by email immediately, or you can create a schedule for recurring report emails. For example, you can email interface utilization reports to the IT department each week for capacity planning.

NOTE

By default, up to 5 email tasks can run concurrently.

To send reports by email, your user account must have the 'Send Reports by Email' role right.

To set up a recurring email schedule, your user account needs the 'Send Reports by Email' and 'Send Reports on a Schedule' role rights.

NOTE
The administrator must specify an email server to enable this feature.

Follow these steps:

1. Do one of the following tasks:
   - Hover over **Reports**, and click **On-Demand Report Templates**, and run an existing report template.
   - Hover over **Dashboards**, and select a dashboard.
2. Click the **Email/Schedule Report** icon on the toolbar.
3. Complete the following fields:
   - **Dashboard**
     A read-only field that identifies the name of the dashboard. The dashboard name appears in the filename of the report that is attached to the email. The dashboard name also appears in the Dashboard column in the list of email schedules on the Manage Scheduled Emails page.
   - **Send To**
     Specifies the email addresses of the recipients. Use the standard format:
     `<name>@<domain>`
   - **Subject**
     Describes the emailed report.
     Example: The dashboard title and any components for which data is included in the report.
   - **Message**
     (Optional) Message that accompanies the emailed report.
   - **Security** Specifies whether to require a login to view the report.
4. Select one of the following formats from the **Format** drop-down list:
   - **PDF**
     - Select **Portrait** or **Landscape** to specify the page layout of the PDF document.
     - Select whether to run the report on only the **First Page** or **All Pages**.

For **First Page**, only the first page is attached and sent in an email message. For **All Pages**, an email message is sent with a time-sensitive link to the report containing all pages.

- **CSV**
  Select **Scaled** or **Unscaled** to specify whether the values in the exported dashboards are scaled.

5. Specify a frequency for running the report with a start time, time zone, and time range for the report:
   - **Send Now**
     Runs the report and sends the email message immediately.
   - **Run Daily**
     Runs the report and sends the email message once per day. If selected, reveals checkboxes where you can select the day of the week when the report is run.
     Default: Run the report every weekday (Monday - Friday) at 00:30 hours in the time zone of the logged-in user. The data in the report reflects the previous 24 hours.
   - **Run Weekly**
     Runs the report and sends the email message once per week. If selected, lets you select the day that the report is run.
     Default: The report is run every Sunday at 01:00 in your time zone. The data in the report reflects the previous seven days (Saturday - Sunday).
   - **Run Monthly**
     Runs the report and sends the email message once per month. If selected, lets you select the day of the month to run the report.
   - **Run Quarterly**
     Runs the report and sends the email message once per quarter. If selected, lets you specify a starting month and day of the month to run the report.
   - **Run Yearly**
     Sends the email message once per calendar year. If selected, lets you specify a starting month. Reports are run on the first day of the specified month.
6. Click **OK**.
   The report is sent immediately, or according to the schedule that you selected. For CVS and single-paged PDF, the file is attached and sent in an email message. For full-paged PDF, an email message is sent with a time-sensitive link to the report. Download and save the report as soon as possible.

Manage Scheduled Reports

Users with the required role rights can set up schedules to email reports on a recurring basis. Selected dashboard data is exported in report format and sent to designated users according to a regular schedule.

If you have the role right to schedule emails, you can manage your own report schedules. Only users with the Administrator role can manage report schedules that belong to other users.

Follow these steps:

1. Log in as a user with the 'Send Reports by Email' role right.
2. Do one of the following tasks:
   - Hover over **Administration**, and click **Configuration Settings: All Scheduled Reports**.

     NOTE
     Tenant administrators only see the items that are associated with their tenant.
   - Hover over **Reports**, and click **Scheduled Reports**.

   The Manage Scheduled Reports page opens, and a list of scheduled reports appears.
3. Select the report schedule that you want to change, and click **Edit**.
4. View or change the settings for report schedules.
5. Click **Save**.
   The new report schedule settings are saved.

Generate a URL for a View

To share a view with users who do not have access to a dashboard, generate a URL for the view. The URL recreates the selected view on demand. The URL lets you add the view to a web page or intranet site to share performance data.

A security token is included with each URL. This token is based on the user who is logged in at the time of URL generation. Any user who accesses the exported view sees the same data as the user who exported the URL.

This procedure requires the Generate URLs from Views role right.

Follow these steps:

1. Open the dashboard that contains the view for which you want to generate a URL.
2. Click the **Edit** (gear) icon on the view, and select **Generate URL**.
   The Generate URL dialog opens. The URL is displayed in the URL field.
3. Enable or disable the following required parameters for the exported view:
   - **View Container**
     Displays the chart or graph with a surrounding container. The container includes the title of the view and a black outline around the chart or graph.
   - **Copyright**
     Shows the copyright information for the web page in the view.
   - **Drill Down**
     Lets users drill down from the view into the underlying data source for more detailed data. To use this feature, users must have access to the data source and the Drill into Data Sources role right.
   - **Detailed View Logging**
     If you encountered an issue with this view, enable logging. You can use this method to provide the necessary details to CA Support.
4. Select from the following time frame options:
   - **Time Options**
     Specify the time range for the data in the view.
   - **Token Expiration Options**
     Control view expiration. The default, 'Never' expires, lets the exported view display indefinitely.
     If you want the view to expire, select a timeout period from the **Token Expiration** list. The URL includes an encrypted token that causes the view to expire after the specified timeout period.
     The token does not enable the user who interacts with the generated view to drill down for more data.
5. (Optional) Click **Preview** to see how the view looks with the options that you have selected.
6. Copy the URL displayed at the top of the page to the clipboard.
7. Click **OK**.
8. Paste the URL to the destination where you want to display the view.
   The URL provides access to the selected view.

Export a View to a CSV File

You can export the contents of a view to a file in comma-separated values (CSV) format. The CSV file format is compatible with spreadsheet applications, such as Microsoft Excel. When you export a view, all view contents are exported as raw data.

> **NOTE**
> Limits of managed items are enforced for CSV export (1,000 for custom views and 5,000 for common views). Items that exceed the limit are not exported from the database.

Follow these steps:

1. Log in as a user with the 'Export to CSV' role right.
2. Navigate to the report that contains the view that you want to export.

3. Click the gear icon on the view, and select **Export to CSV (scaled)** or **Export to CSV (unscaled)**.
   The browser prompts you with options to open or save the exported file.
4. Select **Save** if you want to supply a filename.
5. Browse to the location where you want to save the file, and click **Save**.
   The view is saved as a file in CSV format.

Inventory Pages and Views

The Inventory pages show lists of managed items from all data sources. The **Inventory** menu shows only categories of items currently available to the product from the registered data sources. This menu is also limited to items types that are members of the groups in your user account permission set.

Some dashboards and context pages show relevant inventory list views. These views show the same information as inventory pages.

Inventory lists provide minimal information to identify each item, such as device hostnames or IP addresses.

Based on the device service information, Data Aggregator automatically classifies manageable devices as Router, Switch, and Server types. Also, where applicable, managed devices are associated with context types (Firewall, Load Balancer, Wireless Controller, and Wireless Access Point). The associated context types, including device types, appear listed in a column within device inventory tables.

The following list describes more device type classifications:

- **Pingable**
  The device is not manageable.
  Example: The device does not respond to SNMP requests and its device type cannot be determined.
- **Manageable**
  The device type is classified as 'Manageable' when the following criteria are true:
  – The manageable device cannot be identified as Router, Switch, or Server.
  – The manageable device *can* be identified as Firewall, Load Balancer, Wireless Controller, or Wireless Access Point.
- **Other**
  The device type is classified as 'Other' when the following criteria are true:
  – The manageable device cannot be identified as Router, Switch, or Server.
  – The manageable device cannot be identified as Firewall, Load Balancer, Wireless Controller, or Wireless Access Point.

If the type of some SNMP manageable devices were not identified as expected, override the device or context types. For more information, see Override Device Types.
The following procedures are common to the inventory views:

- To drill down to the context page for an item, click the item in the list.
- To sort by a column, click the column heading.
- To add or remove columns, hover over a column heading, and click the gear icon. Expand **Columns**, and select or clear columns.
- To run and On-Demand report, select an item, and click **On Demand**.
  For more information, see On-Demand Reports.
- To retire a device, or mark it for maintenance, select the device, and click **Manage Life Cycle**.
  For more information, see Manage Device Life Cycles.
- To enable or disable polling on specific interfaces, select the interfaces, and click **Select Polling State**.
  For more information, see Manage Interface Polling Behavior.

Device Inventory with Alarm States

If you have DX NetOps Spectrum configured as a data source, you can view alarm states from the device inventory.For more information, see Integrate CA Spectrum, Integrate CA Spectrum with CA Performance Management, and Customize Event Integration.The following columns in the device inventory contain alarm state information:

- **Current Alarm State**
  View the current alarm state of the device. This column is displayed by default.
- **Last Alarm State Change**
  View the date the alarm state last changed. This column is hidden by default.

The following alarm states are displayed:

- Normal
- Minor
- Major
- Critical
- Maintenance
- Suppressed
- Initial

If the device is not synchronized over from DX NetOps Spectrum, the alarm state information is blank.

Inventory Hierarchy View

The inventory hierarchy view shows the group tree, devices, and components or interfaces. The group tree is pinned to the current group context for the dashboard or context page.

- Select a group to filter the devices, components, and interfaces to the members of that group.
- Select a device in the top pane to filter the interfaces or components in the bottom pane.

  NOTE

  The filter only shows results for a single device. If you select multiple devices, the bottom pane is not filtered.

Configure an Inventory Hierarchy View

For standard view configuration details, see Customize Views.

The inventory hierarchy view can show devices, components, and interfaces. To configure which item types appear, configure the following properties:

- Select whether to show devices, components, or both.
- If the view shows components, select the Component Context type. This property determines the type of components that appear in the view.

Dashboards

Dashboards contain sets of views that let you see the polled data as meaningful information. You can also generate reports from dashboards. In addition to custom dashboards that the administrator has created, several out-of-the-box dashboards are available. When you log in, all dashboards that are assigned to your user account are accessible.

When you register a new data source, you can use the out-of-the-box dashboards and views that are associated with it. With the required role rights, you can edit dashboards, and can save the changes to your own user account.

DX NetOps Performance Management includes two categories of pages:

- **Dashboard Pages**

Provide high-level information, such as average performance of monitored items in a group. Dashboards often provide a drill-down path to more detailed, related pages from a selected context.

- **Context Pages**
  Provide focused performance and status data that is scoped to a specific item, such as a single router or server. These pages are available as drill-down links or tabs from dashboard pages.

To drill down to a context page from a dashboard, take one of the following steps:

- Right-click a hyperlink on an item and select a context page tab
- Click a hyperlink on an item to open the default context page tab.

NOTE
The drill-down feature requires the Drill into Views role right.

Custom Dashboards

All dashboards can be customized with different set of views, context, and layouts. Custom dashboards can be used to troubleshoot issues or long term to monitor categories of items.

Examples:

- A regional manager uses a dashboard that pins views to each site group in that region.
- A systems administrator uses a dashboard to monitor all servers.
- A network engineer uses a dashboard with views pinned to problematic routers or critical interfaces.

Change the Context

The dashboard context filters the data appears in views on the dashboard. When you select a group for a standard dashboard, you apply a filter to all views on the page. When you select a group context, items from all subgroups that are available to you appear in views on the dashboard. However, the same subgroups might not appear on the Items tab in Group administration. For more information on administering groups, see Groups.

You can also view dashboards in multiple windows, and apply a different data context to each dashboard.

Follow these steps:

1. Click the [change] link under the title of the dashboard.
2. Select a group from the group hierarchy.
3. Click OK.
   All views on the page with dynamic context are refreshed to reflect the new data context. The change applies until you log out.

Change the Time Range

To filter data based on specific time periods, specify time ranges for dashboards. Changing the time range is useful for troubleshooting performance issues. For example, you can change the time range to show data from the last seven days. In this case, the time range helps you to determine whether an issue is occurring regularly.

You can select a precise time range for the performance data shown in the current dashboard. Use the time period selectors to select the day, the start time, and the end time. When you change the time range for a dashboard, the change applies to all dashboards in that window. To compare content in different time ranges, open the dashboard in multiple browser windows or modify specific views to show a fixed custom time range.

Follow these steps:

1. Click **Change** in the upper-right corner of the dashboard page.
2. Select a default time period from the list or specify a custom time range.
   The selected time range is applied to the dashboard.

Apply Business Hours to a Dashboard

Business hours filtering is supported for some views on a dashboard. Business hours filtering is not supported for trend charts where the option to enable events is selected. For more information, see Configure Business Hours Filtering.

Business hours override other business hour settings in the following priority order:

1. Business hours configured and associated with a site group. For more information, see Configure Business Hours Filtering.
2. Business hour filters assigned to a view. For more information, see Customize Views.
3. Business hours applied to a dashboard.

Follow these steps:

1. Click **Apply Business Hours**.
2. Apply the business hours to a site listed in the **Site to apply** drop-down.
3. Click **Apply**.
   The business hours for the specified site are applied to the dashboard. The business hours remain applied when you switch between tabs, change time ranges, and so on. If you go to a top-level dashboard or you change the context, the business hours filtering is removed.

Manage Dashboards

To add a page with a custom set of views, create a custom dashboard. To modify the views or layout, edit an existing dashboard. To use an existing dashboard as a template, copy the dashboard.

Create a Dashboard

To add a new dashboard to the dashboard menu, create a dashboard. This procedure requires the Create a Dashboard role right.

Follow these steps:

1. Click the **Dashboards** tab.
   A list of available dashboards appears. Each pane on the page corresponds to a menu.
2. Click **Add Dashboard** next to the menu where you want the new dashboard to appear.
3. Complete the following fields:
   - **Dashboard Menu**
     The menu where the dashboard appears
   - **Menu Item**
     The name of the dashboard in the menu
   - **Dashboard Title**
   - The name that appears at the top of the dashboard
4. Select a layout template for the dashboard. Each layout treats the page as a table with rows and columns for views. The Layout buttons indicate the number of views in each column and row on the page. Select a layout before you add views.

   TIP

   Some views, such as scorecards, MultiView, and MultiTrend views, include a lot of detail and require more screen space. These views do not render well in layouts with more than one column.
5. Click and drag views to the page layout. The maximum number of views per dashboard is 25.
   To customize the view settings, click the Edit (gear) icon for the view.
   For more information about configuring views, see Customize Views.

TIP

To limit the list of views, click **Select Context**, and select a group, item, or device. Views that you add to the layout are pinned to the selected context.

6. Click **Save**.
   The dashboard is saved, and is added to the selected menu.
   The dashboard page refreshes to reflect your changes. The changes persist across login sessions.

Edit a Dashboard

To add or remove views, or rearrange views, edit a dashboard.

This procedure requires one of the following role rights:

- Administer Shared Dashboards
- Create a Dashboard

Follow these steps:

1. Use the **Dashboards** tab to access the dashboard that you want to edit.
2. Click the **More** menu in the upper right corner, and select **Edit Dashboard**.
3. Edit the dashboard as required.
4. Click **Save**.
   The dashboard reloads and shows the new settings.

Copy a Dashboard

To use an existing dashboard as a template for a new dashboard, copy the dashboard.

This procedure requires the Create a Dashboard role right.

NOTE

Dashboards in your My Dashboards menu can be copied only within your My Dashboards menu. You cannot copy a dashboard from your My Dashboards menu to another menu.

Follow these steps:

1. Open the dashboard that you want to copy.
2. Click **More** in the upper right corner, and click **Copy Dashboard**.
3. Select the dashboard menu where you want the copied dashboard to appear.
4. Specify the name for the dashboard in the dashboard menu.
5. Specify the title that appears at the top of the dashboard page.
6. Click **Save**.
   A copy of the dashboard is created. The new dashboard opens.

Organize Dashboards in Menus

Dashboards are organized in customizable menus. You see a list of available dashboards and menus when you hover on the Dashboards tab. Before you add custom menus, only predefined menus are included in the list. The user account role determines the menus that each user can access. Custom menus are defined for each tenant. Only the factory menus are shared among tenants.

Users with the required administrative role rights have the following options to manage menus:

- Create custom menus.
- Add or remove dashboards from menus.
- Associate menus with role.
- Reorganize menus for a role.

To access the **Manage Menus** page, hover over **Administration**, and click **User Settings: Menus**.

My Dashboards

With the required role right, you can add, move, or copy dashboards to a menu. Dashboards in the My Dashboards menu are not visible to other users. Users can copy and customize a dashboard from another menu to the My Dashboards menu.

> **TIP**
>
> To edit the My Dashboards menu for another user, proxy that user account. The action requires the Proxy User role right.

Add a Menu

Custom menus let you organize dashboards and make them available to users with specific roles. Administrators and designers can create custom menus, and can select dashboards for each menu. Custom menus are not available to any users until the administrator adds the menu to a role. All users with that role see the menu.

Follow these steps:

1. Hover over **Administration**, and click **User Settings: Menus**.
2. Click **New**.
3. Specify a name and description for the menu.
4. Add dashboard to the **Available** list for the menu.
5. Click **Save**.

Delete a Menu

You can delete a menu that is no longer in use, which removes it from the Dashboards tab. Any dashboards that are assigned to the menu are not affected, and remain available to other menus. Deleting a user account that is associated with a custom menu does not delete that menu.

Out-of-the-Box Dashboards

Out-of-the-box dashboards provide data immediately after you run discovery and begin collecting metrics from your environment. These dashboards show you examples of dashboards that you can build to monitor your environment. The dashboards provide views for different users across your organization. Use these dashboards as a starting point to create your own dashboard.

> **NOTE**
>
> Many out-of-the-box dashboards include views with data from CA Application Delivery Analysis or Network Flow Analysis. These views appear only if these data sources are installed in your monitoring system and register in NetOps Portal. For dashboards without a listed data source, the data comes from the Data Aggregator in DX NetOps Performance Management.

- **Infrastructure Overview Dashboard**View key health metrics across multiple technologies in the environment.
  User: Operations, Network Engineer

Data Source: Data Aggregator, CA Application Delivery Analysis, Network Flow Analysis

- **Server Device Health Dashboard**View consumption and status-related metrics specific to servers. **User:** Operations, System Administrator
- **Server Performance Dashboard**
  View server performance metrics.**User:** Operations, System Administrator**Data Source:** CA Application Delivery Analysis
- **Network Device Health Dashboard**View performance and consumption metrics for routers, switches, and interfaces.**User:** Operations, Network Planner**Data Source:** Data Aggregator, Network Flow Analysis
- **Network Performance Dashboard**View network, site, and component performance and health metrics for network devices.**User:** Operations, Capacity Planner**Data Source:** Data Aggregator, CA Application Delivery Analysis, Network Flow Analysis
- **Management Overview Dashboard**View availability, reachability, and flow.**User:** Operations
  Data Source: Data Aggregator, Network Flow Analysis
- **Network Overview Dashboard** View high-level key health metrics across multiple technologies in the environment.**User:** Operations, Network Engineer **Data Source:** Data Aggregator, Network Flow Analysis
- **Interfaces Display Dashboard**View interface utilization and traffic health.**User:** Operations, Network Engineer **Data Source:** Data Aggregator, Network Flow Analysis
- **Server Overview Dashboard**View a multitechnology focus on server consumption metrics.**User:** Operations
- **Data Source:** Data Aggregator, CA Application Delivery Analysis
- **CPU/Memory Display Dashboard**View top utilized CPU and memory for all devices in the monitoring environment.**User:** Operations, Capacity Planner
- **Interface Trends Dashboard**View interface utilization, errors, discards trend graphs.**User:** Capacity Planner, Network Engineer
- **Anomaly Detector Dashboard**View anomalies in network flow data. **User:** Operations, Capacity Planner, Network Engineer**Data Source:** Network Flow Analysis
- **Interface Capacity Watch Lists**View interfaces with the top utilization and baseline deviation.**User:** Capacity Planner
- **Router/Switch Capacity Watch Lists**View device level flow and consumption metrics.**User:** Capacity Planner, Network Engineer **Data Source:** Data Aggregator, Network Flow Analysis
- **Server Capacity Watch Lists**View server consumption metrics and baseline deviation.**User:** Capacity Planner, System Administrator

Technology-Specific Dashboards

The following dashboards are associated with specific technologies. The views on each dashboard are customized to fit the needs of each technology.

CBQoS Device Component Views

View the overall status of a device component and determine if the component is causing an issue.

CBQoS Overview Dashboard and Device Views

Use the CBQoS Overview dashboard and device views to analyze the performance of your class-based quality of service policies. CBQoS policies help you manage bandwidth. For example, policies can determine the handling of VoIP and video conferencing data to minimize packet loss.

Some views let you gauge the effects of policies on packet discards. You can discover disparities in the discard rates of prepolicy and postpolicy data for a device. A view such as the Top CBQoS Policing Packets Transmitted view shows the number of packets that violated or exceeded the rate limits set by Class-Based Policing on the device. This information helps you to pinpoint the problem.

DS1 Interface Statistics Device Component Views

View the overall status of a device component and determine if the component is causing an issue.

Firewall Statistics Device Component Views

View the overall status of a device component and determine if the component is causing an issue.

IP Statistics Device Component Views

View the overall status of a device component and determine if the component is causing an issue.

Mobile Wireless Device Component Views

View the overall status of a device component and determine if the component is causing an issue.

Modem Statistics Device Component Views

View the overall status of a device component and determine if the component is causing an issue.

MPLS Device Component Views

View the overall status of a device component and determine if the component is causing an issue.

MPLS Overview Dashboard and Device Views

Use the MPLS Overview dashboard and device views to see the status of the Multi-Protocol Label Switching (MPLS) environment. This information lets you gain a broad sense of overall MPLS performance for the enterprise or a specific device.

QoS Statistics Device Components

View the overall status of a device component and determine if the component is causing an issue.

Response Path Tests Overview Dashboard and Device Views

This dashboard and device views help you see the overall status of response tests for devices that support service level agreements (SLA) using various protocols. This information lets you gain a broad sense of overall SLA performance for the enterprise or a specific device.

For example, as a network infrastructure manager, you can identify which protocols are experiencing the highest latency and slowest response times from the dashboard. You can then look at the device-level views to determine which devices are experiencing the problems. This information can help determine if the cause is increased activity, a device that is down, or some other cause.

Response Path Device Component Views

Use these views, for any supported protocol, to see the overall status of a device component and determine if the response path tests are causing an issue.

Sonet/SDH Device Component Views

View the overall status of a device component and determine whether the component is causing an issue.

Vendor-Specific Dashboards

The following dashboards are associated with specific vendors. The views on each dashboard are customized to fit the needs of each vendor.

Cisco UCS Overview Dashboard and Device Views

The Cisco UCS platform integrates networking, virtualization, storage access, and management of the UCS components chassis, blade server, and Fabric Interconnect (a FCoE switch).

The Cisco UCS Overview dashboard provides performance and status information about Cisco UCS blade servers, chassis, and fabric interconnects. From the Overview page, you can drill down to the Blade, Chassis, and Fabric Interconnect Overview pages. From these Overview pages, you can drill down to component-level views that allow you to view trends.

All these views exploit the capabilities of the Cisco UCS manager that is integrated within Cisco UCS. A malfunctioning component such as a memory module or a fan can be easily identified and replaced before it becomes a severe problem.

Engineers can use this information to troubleshoot problems with the blade chassis physical components, including power supplies, fans, CPUs, and memory modules.

Cisco UCS Blade Overview Dashboard and Device Views

The Cisco Unified Computing System (UCS) Blade Overview dashboard provides performance and status information about Cisco UCS blade servers. From the Overview page, you can drill down to component-level views that allow you to view trends.

For example, a single blade can have up to 24 memory modules. Using this view, an overheated memory module can easily be identified. You can also a spot a malfunctioning fan at a single glance.

Cisco UCS Chassis Overview Dashboard and Device Views

The Cisco Unified Computing System (UCS) Chassis Overview dashboard provides performance and status information about Cisco UCS chassis. From the Overview page, you can drill down to component-level views that allow you to view trends.

For example, a single chassis can have up to eight blades. Using this view, an overheated power supply can easily be identified. You can also spot a malfunctioning fan at a single glance.

Cisco UCS Fabric Interconnect Overview Dashboard and Device Views

The Cisco Unified Computing System (UCS) Fabric Interconnect Overview dashboard provides performance and status information about Cisco UCS fabric interconnects. From the Overview page, you can drill down to component-level views that allow you to view trends.

Use this view to identify CPU consumption of a heavily loaded fabric interconnect. You can also a spot a malfunctioning fan or power supply at a single glance.

Citrix VDI Overview Dashboard and Device Views

The Citrix VDI Overview dashboard provides performance and status information on the Citrix VDI Management Servers and Desktops. Citrix Virtual Desktop Infrastructure (VDI) is the market leading technology for providing virtualized desktops using Hypervisor technologies. Currently, DX NetOps Performance Management supports VMware vSphere Hypervisors. From the Overview page, you can drill down to device-level views to isolate a problem.

Engineers can use this information to troubleshoot problems. Capacity planners can use this information to see which services are overloaded and plan for more VDI servers.

The CPU and Memory reports available in both the Citrix VDI Overview dashboard and the Citrix VDI Server Overview dashboard can display metrics that are not explicit to the Citrix VDI metric family. To avoid this problem, we recommend that you edit the view and save its context to a particular group to ensure the dashboard only reports on Citrix VDI controllers.

Desktop information provided includes:

- Number of desktops
- Desktops that are showing exceptional utilization
- Content of desktop groups
- Available VMs in the VDI catalog

Engineers can use this information to get a quick overview of the infrastructure and its utilization.

Infrastructure servers (called desktop controllers) information provided includes:

- State
- CPU
- Memory
- Storage
- Network utilization

Engineers can use this information to identify bottlenecks on the controllers before they become problems.

Citrix VDI Desktop Overview Dashboard and Device Views

The Citrix VDI Desktop Overview dashboard provides performance and status information on the Citrix VDI Desktops. Citrix Virtual Desktop Infrastructure (VDI) is the market leading technology for providing virtualized desktops using Hypervisor technologies. Currently, DX NetOps Performance Management supports VMware vSphere Hypervisors. From the Overview page, you can drill down to device-level views to isolate a problem.

Desktop information provided includes:

- The number of desktops
- The desktops that are showing exceptional utilization
- The content of desktop groups
- Available VMs in the VDI catalog

Engineers can use this information to get a quick overview of the infrastructure and its utilization.

Citrix VDI Server Page

The Citrix Server page contains the following views:

- Citrix VDI Controller - Registered Desktops (Table)
- Citrix VDI Controller - Services Health (Table)
- Citrix VDI Controller - State (Time Chart)
- Citrix VDI Controller - Configuration (Table)
- Top Citrix VDI Desktop - CPU/Latency/Logon Time (Table)
- Citrix VDI Desktop Group - Used Desktops/Group Enabled (Table)
- Citrix VDI Desktop Group - Desktops Available/Connected (Table)
- Citrix VDI Catalog - Used VMs (Table)
- Citrix VDI Catalog - Assigned VMs (Table)

On the Citrix Server page all metrics are in the scope of its controller, except for desktop group and catalog metrics.

IBM LPAR Overview Dashboard and Device Views

Use the IBM LPAR Overview dashboard and device views to see the status of overall LPAR performance for the enterprise or a specific device.

IBM LPAR Device Component Views

The IBM LPAR Device dashboard lets you view the overall status of a device component and determine if the component is causing an issue.

Microsoft Hyper-V Overview Dashboard and Device Views

Use the Hyper-V Overview dashboard and device views to see the status of overall Microsoft Hyper-V performance for the enterprise or a specific device.

Microsoft Hyper-V Device Component Views

View the overall status of a device component and determine if the component is causing an issue.

Microsoft Exchange Server and Active Directory Overview Dashboard and Device Views

This overview dashboard helps you see the overall status of the monitored devices in your Exchange 2007, Exchange 2010, and Active Directory 2008 environments. This information includes directory domain services as well as mailbox and hub-transport mail services.

For those services you can identify the key performance metrics, which includes mail traffic and directory traffic metrics. You can then access device-level views to isolate a problem.

Engineers can use this information to troubleshoot problems. Capacity planners can use this information to see which services are overloaded and plan for more domain and mail servers.

Microsoft Exchange Server and Active Directory Device Component Views

View the overall status of a specific device component and determine whether the component is causing an issue.

Microsoft Cluster Service Overview Dashboard and Device Views

To see the overall status of the monitored devices that are part of your Microsoft Cluster Service (MSCS), use the Microsoft Cluster Service Overview Dashboard.

You can identify which devices are the top offenders for:

- Resource failures
- Crypto and registry checkpoint restoration and saves
- CPU and memory usage
- Node traffic
- Time spent in various states.

You can then access device-level views to isolate a problem.

Engineers can use this information to troubleshoot problems. Capacity planners can use this information to see which MSCS devices are nearing their utilization capacity and plan for more resources.

Microsoft Cluster Service Device Component Views

View the overall status of a device component and determine if the component is causing an issue.

Solaris Zones Overview Dashboard and Device Views

Use the Solaris Zones Overview dashboard and device views to see the status of overall Solaris Zones performance for the enterprise or a specific device.

Solaris Zones Device Component Views

View the overall status of a device component and determine if the component is causing an issue.

VMware Overview Dashboard and Device Views

The VMware Overview dashboard shows the overall status of monitored devices in your VMware environment, such as virtual centers, virtual machines, ESX hosts, and other supported devices. Use this information to identify which devices are the top offenders for attributes such as CPU and memory usage, bytes in/out, capacity, and the percentage of time that was spent in various states. You can then access device-level views to isolate a problem.

Engineers can use this information to troubleshoot problems. Capacity planners can use this information to see which virtual devices are maximizing their CPU shares and plan for more resources.

To access this dashboard, select Dashboards, VMware Overview. Drill down from the dashboard view to a specific device, and then select a VMware related tab for more detail about that device. To access the device pages directly, select Inventory, Devices, and select a supported device. Then select either the VMware, VMware ESX Host, or VMware Virtual Machine tab.

VMware Device Component Views

View the overall status of a device component and determine whether the component is causing an issue.

IWF Statistics Device Component Views

View the overall status of a device component and determine if the component is causing an issue.

Context Pages

You can often access more information about individual managed items from dashboards. Most dashboards are composed of views of summary data, such as hourly rollups or averages from a group of items. If additional data is available from the data source, with the Drill into Views role right, you can click a link on the dashboard page to drill down into the context pages.

The views on context pages show filtered data from a narrow context, such as a view of data from a single managed item. Drill down into specific data to determine the source of a performance problem using the links.

You can right-click the name of an item in some table views to access a menu. For example, right-click the link that corresponds to an item name in the Inventory section. In the menu, you can select a related context page, containing more granular data.

In this article:

Extended Flow Views

You can drill down Network Flow Analysis data for an interface within NetOps Portal without having to navigate to Network Flow Analysis. The interface context page supports the following views:

- **IP Performance**
  Provides a broad summary of information for the selected interface extracted from the flow data such as Top N Protocols, Top N Hosts, Top N Conversations and Top N ToS values. The interface also provides the interface utilization and discards details for the same interface provided through the data aggregator.
  To navigate to the context page of the protocol or host and so on, you can select the required protocol or host and so on from any of the views.

  NOTE
  If the data source is Network Flow Analysis 10.x and higher, the links in the charts and tables in the IP Performance page open the corresponding network flow pages within DX NetOps Performance Management. If the data source is Network Flow Analysis 9.x and earlier, the links open the corresponding pages in the Network Flow Analysis console.
- The following interface pages are available only if the data source is Network Flow Analysis 10.x and higher:
  - **Network Flow Protocol**
    Provides trends and next level drill downs (hosts and conversations) for the selected Protocol in the table. Host and conversation trend tables provide utilization information in the context of the selected Protocol.
  - **Network Flow Host**
    Provides host trends and protocol stacked trends for the selected Host in the table.
  - **Network Flow Conversation**
    Provides host trends and protocol stacked trends for the selected Conversation in the table.
  - **Network Flow ToS**
    Provides trends and next level drill downs (hosts and conversations) for the selected Type of Service(ToS) in the table. Host and conversation trend tables provide utilization information in the context of the selected ToS.

For all context pages, selecting a row in the top table view sets the corresponding context for the other views in the page. You can change the measurement (Rate/Bytes/Utilization) and direction (In/Out) in the settings for each of the views.

NOTE
You can send or schedule these views in a report by email. By default, the first row is selected in the Top table and the data for the following views are of the first-row selection.
For more information, see Share Data with Other Users.

Pick a Context

You can pick a context using the context picker that is available next to the name of the managed item. Context pickers are also available for the children of the managed item.

To pick a different context, click **Change**, select the context, and then click **OK**.

Reachability Status

Some context pages, such as routers and servers, include the reachability status of the device. The reachability status indicates whether DX NetOps Performance Management is able to reach the device during the selected time range of the context page.

For more information, see Reachability Status and Contact Status.

Edit a Context Page

Prerequisite: You have the Edit Context Pages and Drill into Views role rights.

> **NOTE**
> The predefined Administrator and Designer roles have these role rights by default.

Unlike standard dashboard pages, item context pages are clustered in sets of tabbed pages. You can edit the predefined tabs and can change the views that are displayed on those tabs. You can add tabbed pages. You can also rearrange the tabs in an item context to change their order.

Based on the device service information, the data aggregator automatically classifies manageable devices as Router, Switch, and Server primary device types. Also, where applicable, managed devices are associated with context types (Firewall, Load Balancer, Wireless Controller, and Wireless Access Point). The associated context types, including primary device types, appear listed in a column within device inventory tables.

When you add or edit tabs for a device context page, you can associate them with a primary device type or context type. Tabs that are associated with a primary device type or context type appear on the device context pages that are associated with the specified types.

Examples:

- If you add or edit a tab and associate it with the server type, it appears for all server types.
- If you add or edit a tab for a server type and associate it with firewall, it appears only for all server, firewall types.
- If you add or edit a tab for a router type and associate it with wireless access point, it appears only for all router, wireless access point types.
- If you add or edit a tab for a server type and associate it with wireless controller, it appears only for all server, wireless controller types.

> **TIP**
> Associating a tab with a primary device type or context type impacts all manageable devices that are associated with the primary device type and context type. For change control, minimize edits to the tab after you associate it with a primary device or context type.

Revert changes to all tabs in the current context by selecting **Restore Tabs to Defaults** from the **Edit Tab** menu.

Modifications apply to the current tenant.

Follow these steps:

1. Log in as a user with the Edit Context Pages and Drill into Views role rights.
2. Navigate to the item context whose page you want to edit. For example, click the link for a router on any dashboard to open the Router context pages.
   The tab that is selected includes an **Edit** icon so that you can access the Edit menu.
3. Select the tab that you want to modify.
   The Edit icon appears.
4. Click the **Edit** icon, and then select the **Add** or **Edit** tab.
5. (Optional) Select one of the default tab templates from the menu. Each template populates the page with the default views for that type of page.
6. Change the tab title. A title is required.
   The tab title determines the name that appears at the top of the tabbed context page.
7. (Optional) To associate the tab with a primary device type or context type, select a context page type.
8. Select a layout template for the page from the layout buttons.
9. Remove unwanted views from the page if desired. In the **Layout** pane, click one of the following:

- **Clear Layout:** Changes the positioning of all views on the page.
- **[X]:** Removes an individual view from the page.

10. The views that you can add to the page are shown in categorized lists. The lists are filtered by the selected group or item context.
    All registered data sources are represented. However, the available views are limited to those views that are applicable to the context.

    NOTE
    The item context for the page is preselected for the present context.
11. Click to expand the categories of views.
12. Select a view, drag it to the Layout pane, and drop it where you want it to appear.
13. Click **Save**.
    The context page refreshes to reflect your changes. The changes persist across login sessions, but they are only applied to the current tenant.

Edit Item Properties

You can edit the fields (item properties) that are listed for an item in the Information section of the **Details** tabs in a context page.

Prerequisite: You have the following role rights:

- Modify Device Alias
- Modify Interface Alias
- Modify Device IP Address
- Modify Interface Speed Overrides

Follow these steps:

1. Select the **Details** tab.
2. Click the **Edit** icon in the **Information** section.
3. Edit the available details as desired.
   If **Alias**, **Speed In (bps)**, or **Speed Out (bps)** are not set, they appear blank.
   If you edit an interface that does not exist on the data aggregator, the following fields are unavailable:
   - **Speed In (bps)**
   - **Speed Out (bps)**
4. Click **Save**.
   If you cleared **Speed In (bps)** or **Speed Out (bps)**, the change takes a few minutes to appear.

Apply a Business Hours Filter to a Context Page

You can apply a business hours filter to most views on context pages. Business hours filtering is not supported for trend charts where the option to enable events is selected. For more information, see Configure Business Hours Filtering.

Follow these steps:

1. Click **Apply Business Hours**.
2. Apply the business hours to a site listed in the **Site to apply** drop-down.
3. Click **Apply**.
   The business hours for the specified site are applied to the context page. The business hours remain applied when you switch between tabs, change time ranges, and so on. If you go to a top-level dashboard or you change the context, the business hours filtering is removed.

Rearrange Context Tabs

Each item context consists of sets of tabbed pages. In addition to modifying individual tabbed pages, you can rearrange the tabs in an item context. Modifications are only saved to the current tenant.

Prerequisites: You have the Edit Context Pages and Drill into Views role rights.

> **NOTE**
> The predefined Administrator and Designer roles have these role rights by default.

Follow these steps:

1. Click the **Edit** icon on the selected tab, and select **Reorder** tabs.
   A list of Current Context Page Tabs appears. The list reflects the current ordering of tabs, from left to right.
2. Select a tab to move, and then drag it to another location in the list.
3. Click **Save**.
   The context page refreshes to reflect your changes. The tabs are displayed in a new order from left to right.
   If too many tabs are available for the context to display without horizontal scrolling, an arrow appears on the right. Click the arrow to see additional tabs.

Views

Dashboards and context pages render views. Views report collected data in a chart and/or a table format. Depending on the view, the data comes from the various registered data sources.

Views that show data for a group contain collated and aggregated data from data sources. Views that show data for individual items provide a drilldown path to the context page for the item.

View Contexts

View contexts act as filters that determine the nature of the data that views display. The context of a view is either in "Fixed" or "Dynamic" mode:

- A Dynamic context indicates that the context of the view changes with the context of the dashboard page.
- A Fixed context indicates that the view uses a specified group, device, or component as a context for the data.

If the context of a view is Fixed, changing the dashboard context does not affect the Fixed view.

Views on a context page default to the context of an item. However, from a context page, you can dynamically change the filter to view the context of another item.

The context level of a view determines which data appears in the view:

- **Group**
  Group context level views show data for devices and component items that belong to the selected group. Views that show a single value aggregate the results to the group level.
- **Device**
  Device context level views show data for a specific device. Views that show a single value aggregate the results to the device level.
- **Component / Interface**
  Component or interface context level views show data only for the individual selected device component or interface. Because this context shows only a single item, no aggregation occurs.

Out-of-the-box and Custom Views

Out-of-the-box views show data for a preselected metric or a limited set of metrics. These views include only limited customization options. Each registered data source includes an array of associated out-of-the-box views.

Custom views use the information from the Data Aggregator to provide flexible configuration options. These views let you select from many collected metric in multiple graphical and tabular formats. To add a custom view to a dashboard or page, expand the **Custom View - Infrastructure Management** in the **Views** pane, and drag the view to the **Layout** pane.

Customize Views

Customize view options determine the method that is used to filter the data reported in views. The icons in the upper right of a view let you collapse views, modify view settings, and access context-sensitive documentation.

By default, all views on a dashboard page show data from the same group and for the same time frame. You can change the data context for a particular view to show a specific group. You can select custom time range for views to compare data that was collected at different times.

For some types of views, you can customize the table or chart format. You can also select the data that is reported in custom views.

For more information:

- About how to add a view to a dashboard, see Manage Dashboards.
- About how to add a view to a context page, see Context Pages.

In this article:

Set the Scope for View Settings

To determine which users see view changes, set the scope when changing the view settings. Select the scope from the **Apply Changes** drop-down list whenever you edit a view. The options are scoped according to permissions:

- **My Current Session**
  Save the changes to the current session. Changes do not persist after the logged in session ends. Values set at this level supersede settings at the **My User Account** and **For All Tenant Users** levels during the logged in session. This option is available only after a view has been configured and added to a dashboard page.
- **My User Account** (Default)
  Save the changes to the current user account rendering the view. Changes persist after the logged in session ends and the user logs back in. Values set at this level supersede settings at the **For All Tenant Users** level for the user account.
- **For All Tenant Users**
  Save the changes to all user accounts associated with the tenant rendering the view. Values set at this level supersede the out of the box default settings for all user accounts associated with tenant.

Restore Defaults

To return views setting to default values, click **Use Defaults**. The view reverts to the default options. These changes apply only to the selected scope when you save the view.

Expect the following behavior for each scope:

- **My Current Session**
  Restoring defaults clears the settings set at the **My Current Session** level. Views use values set at the **My User Account** level, values set at the **For All Tenant Users** level, or the out of the box default settings.
- **My User Account**
  Restoring defaults clears the settings set at the **My User Account** level. Views use values set at the **For All Tenant Users** level or the out of the box default settings.
- **For All Tenant Users**
  Restoring defaults clears the settings set at the **For All Tenant Users** level. Views use the out of the box default settings.

NOTE
New configuration options added to a view during and upgrade do not appear in views. To use new configuration options, reset the view to the default options.

Set the Resolution for a View

Trend charts and related views show data as a time series. Time series data shows values for multiple time points across the time range of the report. The resolution setting of a trend view determines the amount of time that each data point in the chart represents.

Table views also include data resolution settings. For table views, the resolution represents the data rollup level to show: as polled, hourly rollup, or daily rollup.

The **Use default resolution** option uses the time range to determine the appropriate resolution automatically.

When you set a custom resolution, the view shows that resolution only if the following conditions are true:

- The user can view report data at that resolution for that time range.
- The data resolution is within the data retention policy of the data source.

If the data is unavailable or the time range is too great, the system overrides the specified resolution automatically. The system returns the closest permissible resolution for the time range.

For trend views, the *chart* subtitles indicate the data resolution.

For tables views, the *view* subtitle indicates the data resolution. The following examples describe some table view subtitles:

- **Resolution: Hourly Roll-Up (Overridden)** Hourly resolution is requested for the selected time range for the current user.
- **Resolution: Hourly Roll-Up (Overridden) Resolution Returned: Daily Roll-Up** Hourly resolution is requested for the selected time range for the current user. The data source does not have hourly data for the entire time range, so the view provides daily resolution.

For more information, see Data Resolution.

Metric Filtering and Baseline Metrics

To include a metric in a view, select a metric from the **Metric Value** list. Metrics fall into two profile types: counter or gauge and are reported in several categories, such as bytes, and percentages, such as utilization.

Metrics with a counter profile type are summed up overtime and rolled up as such. Metrics with a gauge profile type are averaged overtime.

The following options impact the available metrics in the **Metric Value** list:

- **Metric Family**
  The selected **Metric Family** populates the **Metric Value** list with the metrics associated with the **Metric Family**. A metric family is a normalized categorization of metrics associated with device collections, vendor certifications, and monitoring profiles. The metric family contains information about the supported units and rollup strategy for when an associated metric is populated.
- **Metric Filtering**
  Select to limit the **Metric Value** list to only the metrics that are appropriate to the metric profile type (counter or gauge). Clear this option to make available the metrics for the selected **Metric Family**. If you select a metric that is inconsistent with the metric profile type, the metric is flagged with a warning.
- **Baseline Metrics**
  Select to add baselines to the **Metric Value** list. Baseline data helps to characterize past performance for selected monitored parameters, assess present performance, and estimate future performance.
  For more information, see Baseline Calculations.

Set the Custom Time Range

To filter data based on specific time periods, select a custom time range for the views. A custom time range on a view overrides the time range that is set for the dashboard.

You can specify a custom time range for all views except the Calendar Heat Chart view an On-Demand Report Templates.

Follow these steps:

1. Click the View Settings (gear) icon on the view, and then select **Edit** from the menu.
2. Select **Enabled** for **Custom Time Range**.
3. Select the custom time range from the **Time Range** drop-down list.
4. Click **Save**.
   The custom time range is indicated in the subtitle of the view.

Assign a Time Zone and Apply a Business Hours Filter to a View

To show data for particular business hours in a view, assign a time zone and apply a business hours filter to the view. Only items in site groups with the selected business hours filter appear in the view with this filter. If the group context does not include groups with the assigned time zone and business hours, the view shows no data. The subtitle of the view indicates which time zone and business hours filters apply to the view.

> **NOTE**
> Trend views and heat charts apply shading to show the selected business hours, but the data is not limited to the business hours.

Time filtering provides the following options:

- Select a time zone filter without a business hours filter. The view shows data from devices in site groups associated with that time zone.
- Select a time zone and a business hours filter. The view shows data for devices in sites groups associated with that time zone for the selected business hours.
- Select **All Time Zones** and a business hours filter. The view shows data for devices in site groups with the selected business hours.

  > **WARNING**
  > This option loads slowly.

> **NOTE**
> For views with an applied business hours filter, if the time frame is entirely outside the business hours, the view displays the following message:
> `No data is within business hours for this time frame`

For more information, see Configure Business Hours Filtering.

Follow these steps:

1. Click the View Settings (gear) icon in the view, and then click **Edit**.
2. Select **Enable** for **Time Filtering**.
3. Select a **Time Zone**.
   The list contains only time zones that are associated with site groups for the current tenant.
4. Select a **Business Hours** filter.
   The list contains only business hours that are associated to site groups for the current tenant.
5. (Optional) To show a particular site group in the view, select a fixed context. If you select a group context that does not have the selected time zone and business hours, the view shows no data.
6. Click **Save**.
   The view filters data to the selected business hours according to the view type.

Change the Data Context for a View

View contexts act as filters that determine the nature of the data that views display. The context of a view is either in Dynamic or Fixed mode:

- A Dynamic context indicates that the context of the view changes with the context of the dashboard page. For more information, see Dashboards.
- A Fixed context indicates that the view uses a specified group, device, or component as a context for the data.

To show data from a specific item or group in an individual view, set a fixed context for the view.

Follow these steps:

1. Click the **View Settings** (gear) icon in the view, and then click **Edit**.
2. For **Context**, select **Fixed**.
3. Select the context according to the view type.
4. Click **Save**.
   The view shows data only from the selected item or group. The view subtitle indicated the selected context and includes a lock icon.

Data Resolution

Some view types, such as table views and trend views, provide the option to set the resolution for the reported data.

For trend views, the resolution specifies the amount of time that each data point represents. The granularity ranges from minutes to daily resolution depending on the time range.

For table views, the resolution specifies the granularity of the reporting data: as polled, hourly rollup, or daily rollup.

The time range for a view determines the default data resolution:

- **As Polled Data**
  Less than, or equal to, 24 hours
- **Hourly Resolution**
  Greater than 24 hours to less than 14 days
- **Daily Resolution**
  14 days to less than one year
- **Weekly Resolution**
  One year or longer

When you edit a view, the current selected time range determines the possible resolution options. The following values show the maximum time range for each data resolution:

- **As Polled Data**
  Less than, or equal to, 24 hours
- **Hourly Resolution**
  Less than 30 days, and more than 24 hours
- **Daily Resolution**
  More than, or equal to, 30 days

Users with the Run Dashboards at Higher Resolution role right use the following maximum time ranges:

- **As Polled Data**
  Less than, or equal to, 31 days
- **Hourly Resolution**

More than 31 days
- **Daily Resolution**
  More than three months

To view the report resolution settings, select **Administration**, **Report Resolution**. The resolution settings cannot be modified on the Report Resolution page.

When you set a custom resolution, the view shows that that resolution only if the following conditions are true:

- The user can view report data at that resolution for that time range
- The data resolution is within the data retention policy of the data source

If the data is unavailable or the time range is too great, the system overrides the specified resolution automatically. The system returns the closest permissible resolution for the time range. For trend views, the *chart* subtitles indicates the data resolution. For tables views, the *view* subtitle indicates the data resolution.

Alarms View

If you have DX NetOps Spectrum configured as a data source, an alarms view lets you view and manage DX NetOps Spectrum alarms. For more information, see Integrate CA Spectrum, Integrate CA Spectrum with CA Performance Management, and Customize Event Integration.

NOTE

The DX NetOps Spectrum data source now requires a user name and password.

An alarms view provides a prioritized list of DX NetOps Spectrum alarms, which helps you quickly focus on resolving the most impactful problems. The alarms view also provides visibility into other, potentially related, issues on the same device, or connected to a device.

Each device context page includes an Alarms tab. For more information, see Context Pages.

NOTE

The Alarms tab was added to all device-level context pages in the DX NetOps Performance Management 3.6 release.

The following role rights are associated with an alarms view:

- **Allow Alarm Modification Actions**
  Lets users acknowledge, clear, or assign a troubleshooter to alarms.
- **Allow Alarm Triage Actions**
  Lets users perform alarm triage actions.
- **Allow Alarm Filter Creation**
  Lets users create and manage alarm filters.
- **Allow Alarm Filter Management**
  Lets users manage and assign alarm filters to other users.

For more information, see Data Source Role Rights.

NOTE

When you sort, search, or filter an alarms view, the returned results show only the alarms up to the maximum alarms retrieved, which is specified in the view settings.

Standard View Concepts

An alarms view includes standard view configuration options, such as time filters and context settings. For standard view configuration details, see Customize Views.

By default, all views on a dashboard page show data from the same group and for the same time frame. You can change the data context for a particular view to show a specific group. You can select a time range for views to compare data that was collected at different times.

Context Settings

In DX NetOps Spectrum, you can select a particular Global Collection to view all the alarms asserted on it.

Group filters in the upper-left corner of NetOps Portal behave similarly. You can select a particular group, which can correspond to your Global Collections, to view all the alarms asserted on it.

If you select a custom group for reporting, the alarms view is scoped to the elements within that custom group. NetOps Portal groups can include DX NetOps Spectrum Landscapes and Global Collections. Landscapes and Global Collections are available under All Groups, Inventory, Data Sources, and *your DX NetOps Spectrum data source*. If the group context includes Landscapes or Global Collections, an alarms view can show alarms for devices that are not synced to NetOps Portal. For more information, see Manage Groups.

Time Filters

In the WebClient Alarms tab in DX NetOps Spectrum, time filters display the alarms that occurred in the last few hours. The options available under this filter are: 1, 2, 4, 6, 12, 24 hours.

Example: Suppose, four alarms were generated an hour and a half ago, and two alarms were generated 30 minutes ago in your DX NetOps Spectrum environment. In this case, selecting the **2-hours** option displays six alarms. If you select the **1-hour** option, two alarms are displayed.

Time filters in the upper-right corner of NetOps Portal behave similarly. However, you can also specify custom time ranges. We recommend that you specify a time range that includes the latest list of alarms.

> **WARNING**
>
> Only alarms that are raised within the specified time range appear in an alarms view. After an alarm is cleared, it does not appear regardless of the selected time filter.
>
> **TIP**
>
> Ensure the specified time range is large enough to include all active alarms.

Alarms View Panes

An alarms view shows alarms data from DX NetOps Spectrum in the following available panes:

> **TIP**
>
> We recommend that you turn auto-refresh off when you view alarms. If you have auto-refresh turned on while looking at an alarms view, any selected alarms are cleared when the page refreshes.

- **Alarms**
  View a list of the latest DX NetOps Spectrum alarms and their severity.
- **Alarm Details**
  View the details of the selected alarm. You can add or remove the attributes to display in the Alarm Details pane. By default, the Alarm Details pane displays the following attributes.

  - Severity
  - Date/Time
  - Item Name
  - IP Address
  - Model Type
  - Acknowledged
  - Contact Person
  - Troubleshooter
  - Trouble Ticket ID
  - Number of Occurrences
  - Impact
- **Impact: Management Lost**
  View the devices that this alarm impacts.
- **Impact: Symptoms**
  View the alarms leading up to this alarm.
- **Neighbor Topology**
  View all the neighboring models that are connected to the device of the selected alarm.
- **Interfaces**
  View the interfaces associated with the selected alarm.
- **Events**
  View the events associated with the selected alarm. If you have the Export to CSV role right, you can click the gear icon to export the events list to CSV. You can also filter and search the events list.

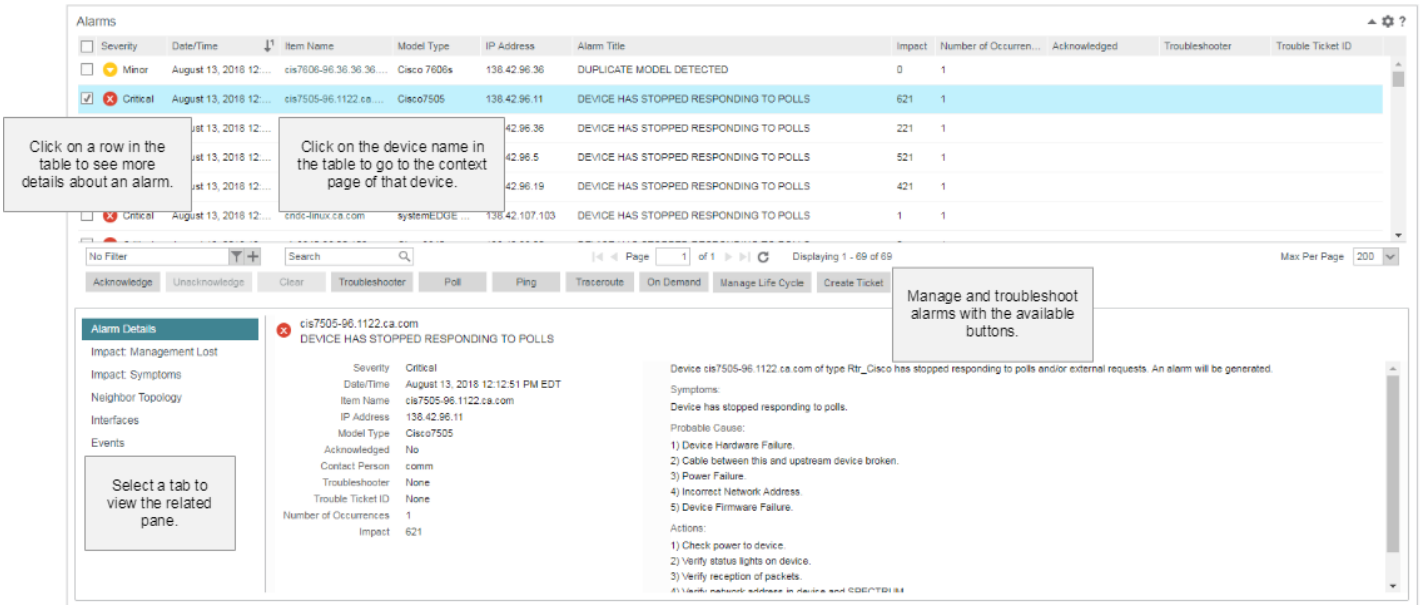
NOTE

The Events tab is available only with DX NetOps Spectrum 10.3.0 or higher.

Alarms View Elements

The following example shows the important elements of an alarms view:

Figure 13: Alarms_Views



Sort Multiple Columns

You can sort by up to three columns in an alarms view.

Follow these steps:

1. In the column header to sort first, click the Gear icon.
2. Select **Sort First**, and click one of the following options:
   - **Sort Acsending**
     Sort the column in ascending order.
   - **Sort Descending**
     Sort the column in descending order.
   - **Remove Sort**
     Remove the sort order from the column.

     NOTE

     You can CTRL+Click on a column heading to remove the sort order.
3. In the column header to sort second, click the Gear icon.
4. Select **Sort Second**, and click a sort option.

   NOTE

   You can SHIFT+Click on a column heading to add it as a secondary sort.
5. In the column header to sort third, click the Gear icon.
6. Select **Sort Third**, and click a sort option.

Create an Alarm Filter

If you have the Allow Alarm Filter Creation role right, you can create alarm filters.

You can filter by the the following attributes:

- **Acknowledged**
  Indicates whether the alarm is acknowledged
  Attribute ID: 0x11f4d
  Type: BOOLEAN
- **Cause Code**
  Identifies the cause of the alarm
  Attribute ID: 0x11f50
  Type: HEX
- **Clearable**
  Indicates whether the alarm is clearable
  Attribute ID: 0x11f9b
  Type: BOOLEAN
- **Contact Person**
  The person to contact when device problems occur
  Attribute ID: 0x23000c
  Type: STRING
- **IP Address**
  The IP address for the item
  Attribute ID: 0x12d7f
  Type: ADDRESS_RANGE
- **Location**

The location of the device
Attribute ID: 0x23000d
Type: STRING

- **Manufacturer**
  The manufacturer of the device
  Attribute ID: 0x10032
  Type: STRING
- **Model Class**
  The model class of the item
  Attribute ID: 0x11ee8
  Type: ENUM
- **Name**
  The device or model name
  Attribute ID: 0x1006e
  Type: STRING
- **Number of Occurrences**
  The number of times that the alarm occurred
  Attribute ID: 0x11fc5
  Type: INTEGER
- **Severity**
  The severity of the alarm
  Attribute ID: 0x11f56
  Type: ENUM
- **Show Symptoms**
  Indicates whether the alarm is the result of symptoms
  Attribute ID: 0x12a07
  Type: BOOLEAN
- **Symptom Count**
  The number of symptoms for the alarm
  Attribute ID: 0x12a06
  Type: INTEGER
- **System Name**
  The system name of the device
  Attribute ID: 0x10b5b
  Type: STRING
- **Troubleshooter**
  The troubleshooter assigned to the alarm
  Attribute ID: 0x11fc6
  Type: ENUM
- **Trouble Ticket ID**
  The Trouble Ticket ID assigned to the alarm
  Attribute ID: 0x12022
  Type: STRING

An administrator can also use web services to manage alarm filter attributes. For more information, see Use Web Services to Manage Alarm Attributes.

Follow these steps:

1. To the right of the filter drop-down, click the plus icon.
   The Create Filter dialog opens.
2. Complete the fields and specify the conditions for the filter.
3. Click **Save**.
   The new filter appears in the filter drop-down.

Manage Alarm Filters

If you have the Allow Alarm Filter Management role right, you can manage and assign alarm filters to other users.

When you create or edit a filter from the Manage Filters dialog, you can set the scope to one of the following options:

- **For My Use**
  The filter is available only for your user account.
- **For All Users With My Role**
  The filter is available only for users with your role.
- **For All Tenant Users**
  The filter is available for all the tenant users.

  NOTE

  If a new role is added, users with that new role do not have access to the shared filter. You can assign the filter to the new role.
- **For Specific Users or Roles**
  The filter is available for selected users or roles.

Follow these steps:

1. Click the filter drop-down.
2. Click **Manage Saved Filters**.
3. **Add**, **Edit**, **Copy**, or **Delete** filters as necessary.

Assign Filters to Specific Users or Roles

If you have the Allow Alarm Filter Management role right, you can assign shared filters to specific users or roles.

Follow these steps:

1. Click the filter drop-down.
2. Click **Manage Saved Filters**.
3. Select a filter to assign to users or roles.
4. Click **Users** or **Roles**.
5. Select the users or roles to assign the filter.
6. Click **Save**.

Manage Alarms

An alarms view contains the following alarm management buttons:

- **Acknowledge**
  Acknowledge the selected alarms. "Yes" appears in the Acknowledge column in the top Alarms pane.
- **Unacknowledge**
  Unacknowledge the selected alarms. This button removes "Yes" from the Acknowledged column in the top Alarms pane.
- **Clear**

Remove the selected alarms from the top Alarms pane. This option is available only for clearable alarms. To determine whether an alarm is clearable, show the Clearable column.

- **Troubleshooter**
  Manage the troubleshooter assignment for the selected alarms.
- **Poll**
  Poll the devices for the selected alarms. The poll is initiated from the SpectroSERVER.
- **Ping**
  Send an ICMP ping to the devices for the selected alarms. The ICMP ping is initiated from the SpectroSERVER.
- **Traceroute**
  Determine the route (path) to the device of the selected alarm over a maximum of 30 hops. The round-trip time of packets that are received from each hop is also measured. Traceroute is initiated from the SpectroSERVER. This option is unavailable when you select multiple alarms.
- **On Demand**
  Launch On-Demand reports for the selected alarms. Alarms and items that are not synchonized to NetOps Portal are filtered out. For more information, see On-Demand Reports. This option is available when you have the Create On-Demand Report Templates role right and the Run On-Demand Report Templates role right.
- **Create Ticket**
  Create a service desk ticket for a single alarm that is selected in the table. To launch the trouble ticket system from an alarms view, a trouble ticket management system at your organization must be set up with DX NetOps Spectrum. For more information, see the Alarms Tab Preferences in the CA Spectrum documentation. This option is available when you have either the Allow Alarm Modification Actions role right or the Allow Alarm Triage Actions role right.
- **Manage Life Cycle**
  Change the life cycle state of a device to Active or Maintenance. This option is available when you have the Administer Life Cycle role right. The **Synchronize device life cycle state from Spectrum** option should be selected for the DX NetOps Spectrum data source. For more information, see Manage Device Life Cycles.

Configure an Alarms View

To display an alarms view, go to the **Alarms Console** dashboard or add the **Alarms** view available from the **Alarms and Events** section. An Alarms view is also available from device, router, server, switch, and interface context pages.

Configure the following alarms view settings:

- **Max Alarms to Retrieve**
  Specify the maximum number of alarms to retrieve from DX NetOps Spectrum starting with the most recent.
  Default: 20,000
  Maximum: 20,000
- **Ping Count**
  Specify how often to send an ICMP ping to each device. You can specify a value of 1 through 5.
  Default: 3
- **Grid Height**
  Specify the height of the Alarms grid.
  Default: 300
- **Sort First**
  Select a column on which to sort first.
  Default: Date/Time
- **Sort Second**
  Select a column on which to sort second.
  Default: None
- **Sort Third**
  Select a column on which to sort third.

Default: None
- **Sort Direction**
  Select whether to sort the selected column in **Ascending** or **Descending** order.
  Default: Descending
- **Customize Panels**
  Select the panes to enable or disable.
- **Attributes to Show**
  Add or remove the attributes to display in the Alarm Details pane.
- **Filter Selection**
  Select whether to pin an alarm filter to the view. If enabled, select the filter to pin to the view.

Browser Views

The browser view lets you show the content of a web page directly in NetOps Portal. The browser view also lets you update internal and external data dynamically.

Configure a Browser View

To show a web page or application, configure a browser view.

Follow these steps:

1. Specify a URL for the view.

   WARNING
   Use a URL for a web page that supports being an embedded iFrame.
2. (Optional) Customize the URL parameters.

   TIP

   Use URL parameters to add NetOps Portal data to the URL. These parameters support adding context-sensitive information, such as time range or group, to the URL. Use these parameters to provide specific information to OpenAPI applications.
3. Set a height for the view, in pixels.
4. Click **Save**.
   The web page or application that the URL references appears in the browser view window.

Customize URL Parameters

You can customize a browser view to your preference. When you configure a browser view, select parameters to add to the URL of the view. Each parameter is defined when the URL is accessed.

TIP

Add an identifier to the URL before each property. The identifier helps you to determine the meaning of each property in the URL.

The parameters that are available from the user interface appear in a drop-down with two columns. The first column shows the property. The second column shows the current values for the logged in user. The second column values are based on the current context and time on the system. The properties resolve in the following manner:

- **{Culture}**
  Culture of the logged in user
- **{ItemDesc}**
  Description of the item or group of the current context

The context is specified at the top of the page or within the dashboard builder.

- **{ItemId}**
  NetOps Portal item ID of the current item or group
- **{ItemIdDA}**
  Data Aggregator item ID of the current item or group
  This parameter appears only when you have a Data Aggregator data source.
- **{ItemName}**
  Name of the current item or group
- **{ItemSubType}**
  Subtype of the current item or group
- **{ItemSubTypeName}**
  Subtype name of the current item or group
  The subtype name is usually the same as the subtype, but not always.
- **{ItemType}**
  Type of the current item or group
- **{ItemTypeLabel}**
  Singular form of the item type label
- **{ItemTypeLabels}**
  Plural form of the item type label
- **{ItemTypeName}**
  Item type name
- **{Locale}**
  Locale of the logged in user
  The locale is similar to culture, but has a slightly different format.
- **{ModelHandleSpectrum}**
  Model handle from DX NetOps Spectrum for the device or interface
  This parameter is unavailable for groups. This parameter appears only when you have a DX NetOps Spectrum data source.
- **{PageSize}**
  Number of items that are specified to display in the view
- **{Resolution}**
  Data resolution in seconds
- **{ResolutionLabel}**
  Data resolution in a descriptive text
- **{ServerName}**
  NetOps Portal server name or IP address
- **{ServerNameDA}**
  Data Aggregator server name or IP address
  This parameter appears only when you have a Data Aggregator data source.
- **{ServerPort}**
  NetOps Portal server port
- **{ServerPortDA}**
  Data Aggregator server port
  This parameter appears only when you have a Data Aggregator data source.
- **{TimeEndUTC}**
  Number of seconds of the end of the time range (based on Unix time) at the top of the page
- **{TimeEndUTCExpanded}**
  Expanded form of the end of the time range
- **{TimeSpan}**

Description of the time range that is specified at the top of the page
- **{TimeStartUTC}**
  Number of seconds of the start of the time range (based on Unix time) at the top of the page
- **{TimeStartUTCExpanded}**
  Expanded form of the start of the time range
- **{UserEmailAddress}**
  Email address of the logged in user (if specified)
- **{UserId}**
  NetOps Portal ID of the current user
- **{UserName}**
  Current user name
- **{UserRoleId}**
  NetOps Portal role ID of the current user
- **{UserRoleName}**
  Role name of the current user
- **{UserSsoToken}**
  SSO token for the current user
  This token can be used to log in to other data sources that honor SSO (for example, Network Flow Analysis).
- **{UserTimeZone}**
  Time zone of the current user

Example:

If you want aspects of your browser view to adjust dynamically, use custom URL parameters.

For this example, you want to display the content at the following URL:

`http://PC_host:port/pc/apps/user/appsubdirectory/MyPageFile.html`

You can append the URL with the following URL parameters:

`?id={ItemIdDA}&startTime={TimeStartUTC}&endTime={TimeEndUTC}`

- **{ItemIdDA}**
  This parameter resolves to the ID of the interface for the selected context page.
- **{TimeStartUTC}**
  This parameter resolves to the start time selected in the time picker for the context page.
- **{TimeEndUTC}**
  This parameter resolves to the end time selected in the time picker for the context page.

Follow these steps:

1. Select a URL parameter from the drop-down list, and click the Append to URL button.
   The URL parameter is added to the end of the current URL in the URL field.
2. Click the URL field, and enter an identifier before each parameter.
   Enter a '?' before the first parameter. Enter a '&' before each subsequent parameter.

Bar Chart Views

Bar charts, also known as horizontal bar charts, show a comparative visualization of data. These charts are useful to compare items in a small group when the expected value of the metric is similar for all items. The visualization enables you to identify which item differs from the others. Bar charts can be useful for capacity planning.

Business hours filtering can apply to the data in bar chart views. The applied business hours filter appears in the subtitle. For more information, see Configure Business Hours Filtering.

For standard view configuration details, see Customize Views.

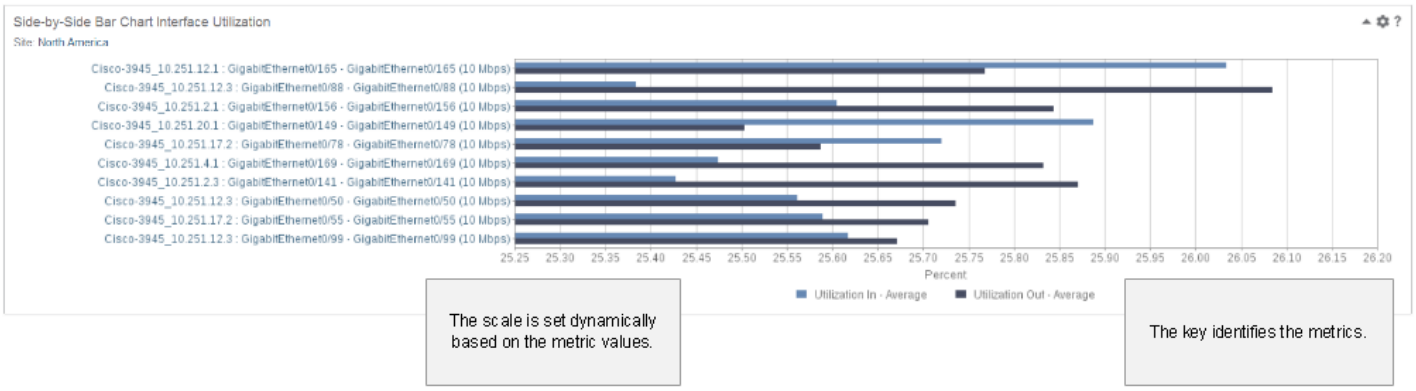
Side-by-Side Bar Charts

Side-by-side bar charts compare pairs of related metrics for interfaces.

For the out-of-the-box views, select the Metric Value to display on the view: Percentage, Rate, or Volume.

The following example shows the important elements of a side-by-side bar chart:

Figure 14: Side-by-Side Bar Chart Elements

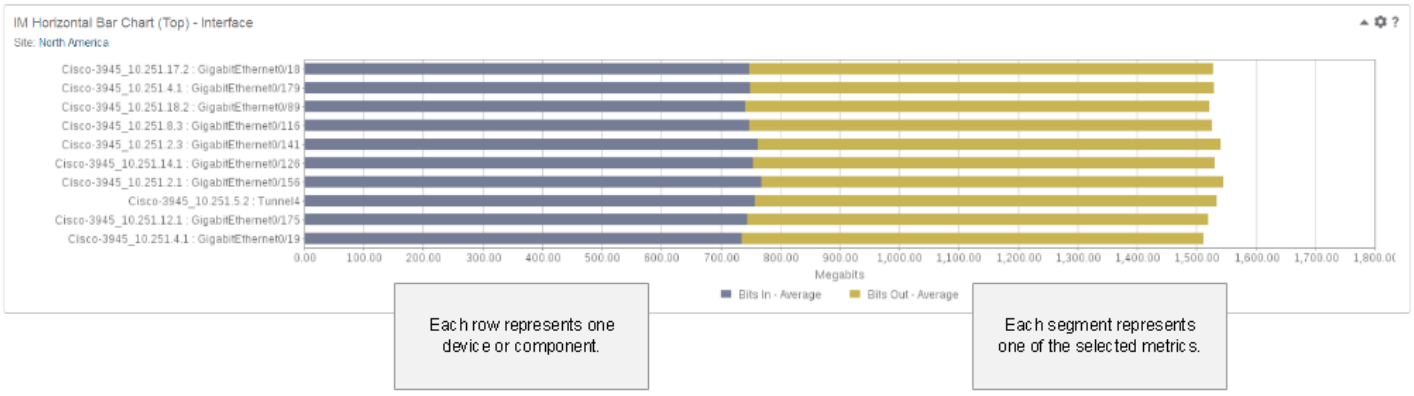


Stacked Bar Charts

Stacked bar charts show a horizontal bar with segments representing each selected metric.

The following example shows the important elements of a horizontal bar chart with the stacked chart type:

Figure 15: Stacked Bar Chart Elements



Configure a Horizontal Bar Chart View

The IM Horizontal Bar Chart is a custom view that lets you select the metrics to display. The chart can display multiple metrics for each device or component in the view context.

Follow these steps:

1. Select the metrics to show in the chart.
   Each selected metric appears as different colored bar in the chart.

TIP

The custom bar chart is best used to compare a few similar metrics for a small group of items. If you select too many items or too many metrics, the view is difficult to read.

Select metrics with consistent unit types (bits, percentage, and so on) for more valuable comparisons.

Also, select metrics with a consistent rollup or aggregation strategy. For example, compare two sets of averages for two different metrics. This comparison is more valuable than comparing the total for one metric and the average for another metric.

2. Select the sort order for the view.
   The metric that you select from the Metric Sort list represents the first bar on the bar chart.
3. Select the Metric Calculate Level.
   This option determines what level of aggregation each row in the view represents: a device, or a component.
4. Select the chart type:
   - **Stacked Chart**
     For each device or item, this chart type shows a horizontal bar with segments representing each selected metric.
   - **Side-by-Side Chart**
     For each device or item, this chart type shows a horizontal bar representing each selected metric.

Bar Chart Tables

Some out-of-the box table views show a column with a bar graph. These tables display a bar chart column for metrics with percentage values. For more information about these views, see Table Views.

Calendar Heat Chart Views

Calendar heat chart views provide a month-long overview with hourly intervals that are color coded based on custom thresholds. The calendar shows one month with a colored block to show the status for each hour.

NOTE
If the hourly data retention is less than 1 month, the calendar heat chart still shows hourly data.

This view helps network engineers identify patterns in percentage-based metrics such as utilization, latency, and loss metrics. Identifying patterns can be a critical step in locating the source of performance issues that might appear to be intermittent. Usage patterns also aid in capacity planning.

You can limit the data that appears in the view by applying one of the following pattern-matching filters:

- **Busy Hour**
  View the hour each day with the highest value for the selected metric.
- **Business Week Pattern**
  View patterns where the same hours have similar values on three or more days in the business week.
- **Calendar Week Pattern**
  View patterns where the same hours have similar values on four or more days in the calendar week.
- **Repeating Hours by Business Day**
  View patterns where the same hour on the same day of the business week has a similar value for three or more times in the month.
- **Repeating Hours by Calendar Day**
  View patterns where the same hour on the same day of the calendar week has a similar value for three or more times in the month.

Maintenance indicators apply to all the devices and components in a site group. When you have selected the associated site group in the context, maintenance indicators appear as shaded cells in the calendar heat chart.

For more information, see Schedule Maintenance Indicators.

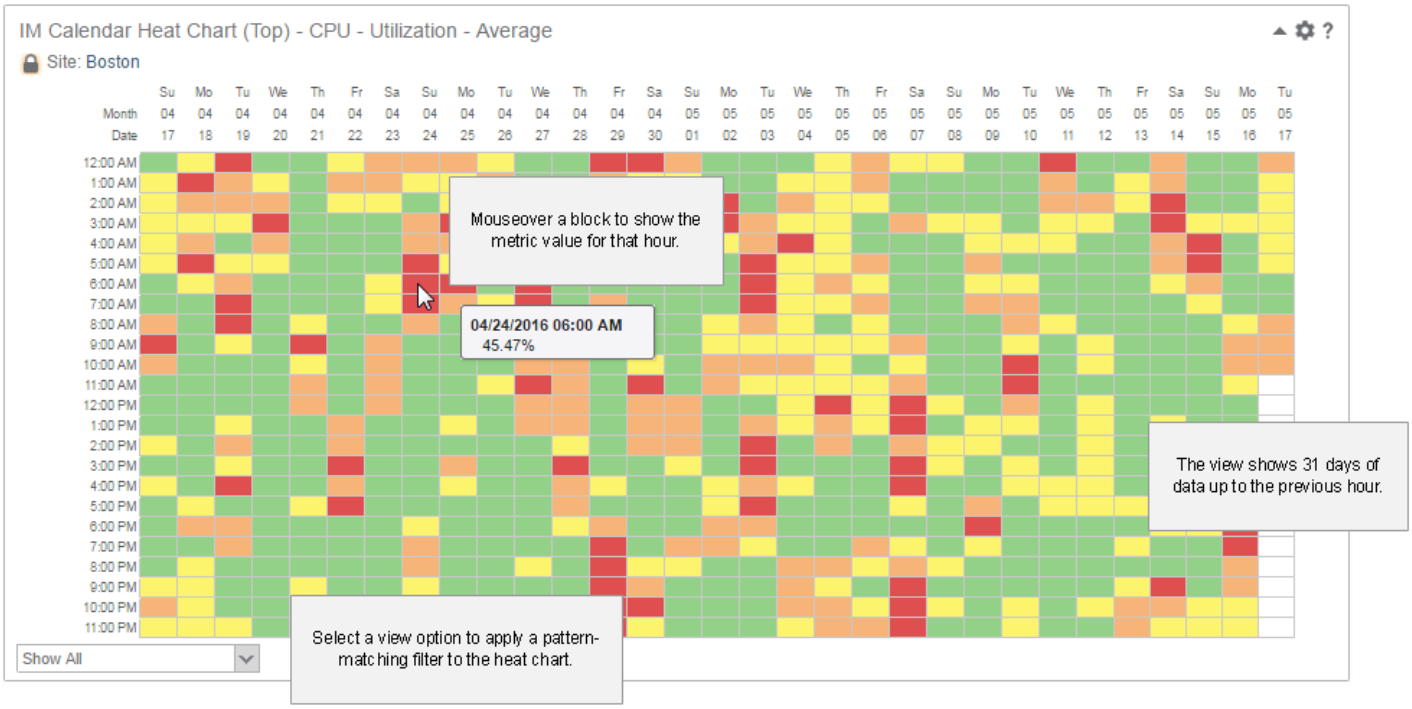
Applying a business hours filter to a calendar heat chart view displays the same data in the view, but the periods outside of the business hours are shaded.

For more information:

- About business hours filtering, including how to define business hours definitions, see Configure Business Hours Filtering.
- About how to apply a business hours filter to views on a dashboard, see Dashboards.
- About how to apply a business hours filter to views on a context page, see Context Pages.

The following example shows the important elements of a calendar heat chart:

Figure 16: Calendar Heat Chart Elements



Configure the Behavior of Calendar Heat Chart Views

Calendar heat charts include configuration options to set the threshold values for the colors to show in the chart. The view also lets you change the starting day and time display formats.

> **TIP**
> Because the color status thresholds are set in integers, this view works best with metrics that are gradable by integer, such as utilization. Metrics that have narrow ranges in value, such as percent errors, do not provide meaningful visualizations.

The IM Calendar Heat Chart lets you select the metric to show in the view. The built-in calendar heat charts on context pages have fixed metric selections.

For standard view configuration details, see Customize Views.

Specify the following properties:

- **Business Week Start**
  Define the start of the five-day business week for the pattern-matching filters.
- **Time Display Format**

Specify whether the hours are identified with 12-hour or 24-hour time.

- **Green Zone Start**
  Specify the threshold for the metric to show a normal status indicated with green.
- **Yellow Zone Start**
  Specify the threshold for the metric to show a minor status indicated with yellow.
- **Orange Zone Start**
  Specify the threshold for the metric to show a major status indicated with orange.
- **Red Zone Start**
  Specify the threshold for the metric to show a critical status indicated with red.

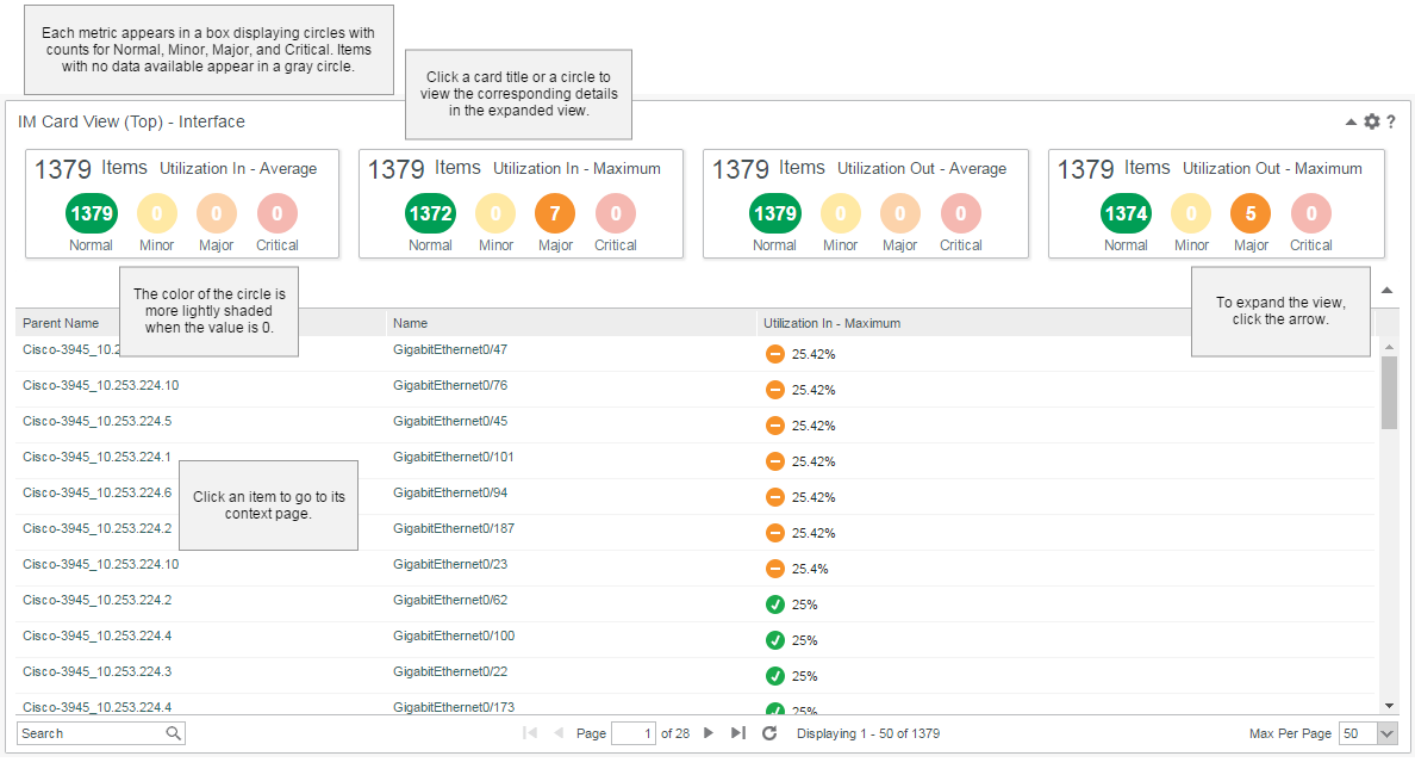
TIP
You can configure calendar heat chart views to reflect reverse value severity. For metrics where low values are bad and high values are good, set Green highest and Red lowest. To omit a severity level, set the threshold value to zero. For example, setting Orange to zero removes the major severity when profiling metric thresholds.

Card Views

Card views display cumulative counts of groups, devices, or components for the threshold ranges of each specified metric. The counts appear within green, yellow, orange, and red circles. The color of each circle is based on a set of user-defined thresholds. You can specify separate thresholds for each metric.

The following image shows the important elements of a card view:

Figure 17: Card_Views



Configure the IM Card View

To display a card view, add the IM Card view. Card views include standard view configuration options, such as time filters and context settings. For standard view configuration details, see Customize Views.

Follow these steps:

1. Select the metrics to show in the card view.
   Each selected metric appears as a box in the view.
2. Select the **Metric Calculate Level**.
   This option determines what level of aggregation for the cumulative counts: by groups, devices, or components.
3. For each metric, specify the thresholds for the status indicators.
   If the value for the metric is at or above the specified value, the icon indicates the status by color.
   Green indicates normal, yellow indicates minor, orange indicates major, and red indicates critical.
   To remove the status indicators, set the threshold values for all status levels to zero.
   You can specify thresholds with up to two decimals places with the lowest value being 0.01. Values with more than two decimal places are rounded.

   TIP

   For metrics where low values are bad and high values are good, set Minor highest and Critical lowest. To omit a severity level, set the threshold value to zero.

Dynamic Trend Views

Dynamic trend views combine data from multiple managed items, or groups of items, in a single view. You can add this view to a dashboard or context page. You can compare the data from hundreds of managed items, or groups of managed items, in one or more charts. You can select display units and determine how data is plotted using the view settings.

NOTE
Dynamic trend views support only the data aggregator data source.

You can create a trend chart for a selected metric using the dynamic trend view. The view is termed "dynamic" because it has the following options:

- Display an individual chart for the selected metric for each item that you select.
- Display a single chart for all the selected items, for a single metric.
- Display a single trend line for a selected metric that reflects aggregated data from all the selected items.

NOTE
Percentile metrics appear as dashed lines.

Use dynamic views during advanced troubleshooting. These views let you build a complex comparison of data from multiple groups, or a temporary grouping of items. You can quickly select items from across a wide range of possibilities, without creating a group in advance.

For example, you can use dynamic views to analyze data from any of the following items:

- Multiple interfaces in a group
- The interfaces in your inventory with the worst performance metrics
- The interfaces on a single device

A dynamic view is empty until you edit the view settings that you want to apply.

Trend data in dynamic views represents averages of metric values across all items that are included.

TIP
Dynamic trend views require more page space than other types of views. To add a dynamic trend view to a dashboard, add it to a single-column area in the layout. Do not add a dynamic trend view to a small section on a dashboard layout.

Dynamic trend views also require more space for the PDF output.

Maintenance indicators apply to all the devices and components in a site group. When the associated site group is selected in the context, maintenance indicators appear as shading in dynamic trend views.

For more information, see Schedule Maintenance Indicators.

Applying a business hours filter to a dynamic trend view displays the same data in the view, but the periods outside of the business hours are shaded.

For more information:

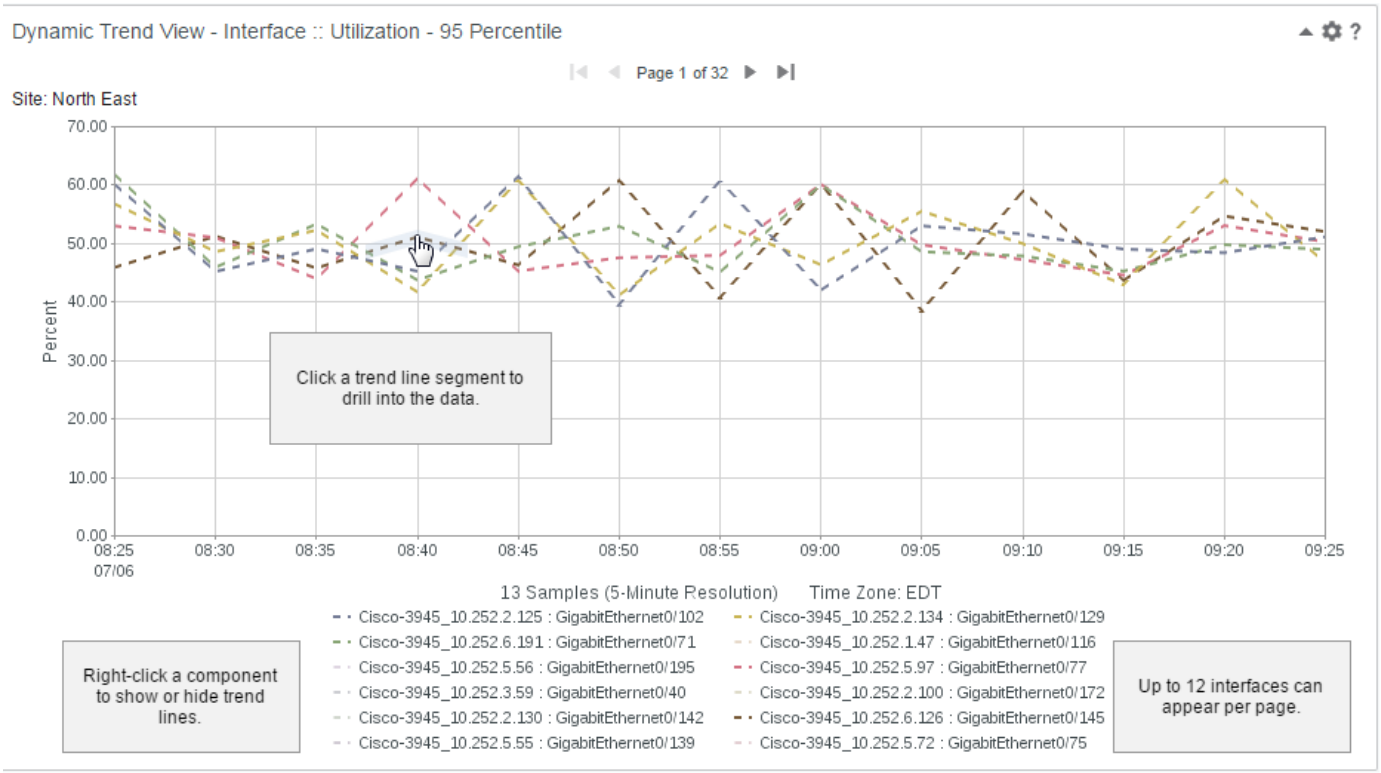
- About business hours filtering, including how to define business hours definitions, see Configure Business Hours Filtering.
- About how to apply a business hours filter to views on a dashboard, see Dashboards.
- About how to apply a business hours filter to views on a context page, see Context Pages.

In this article:

Elements of a Dynamic Trend View and Dynamic Trend MultiView

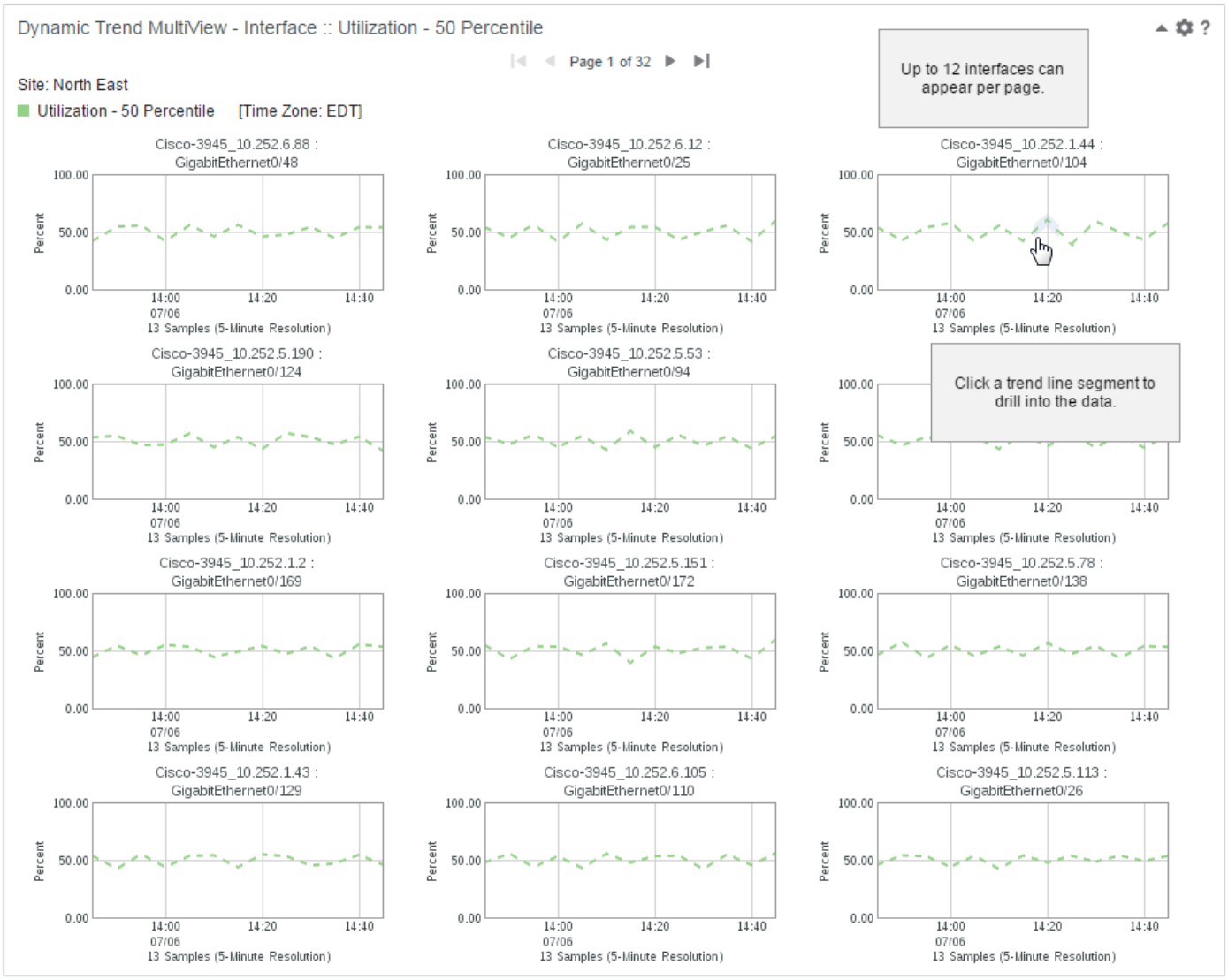
The following image shows the important elements of a dynamic trend view:

Figure 18: DynamicTrendView



The following image shows the important elements of a dynamic trend MultiView:

Figure 19: DynamicTrendMultiView



Configure a Dynamic Trend View

To configure a dynamic trend chart, customize or edit a dynamic trend view. Dynamic trend views contain no data until you edit them to select settings.

For standard view configuration details, see Customize Views.

Follow these steps:

1. Select one of the following Chart Type options:
   - **Trend Chart**
     View a traditional trend line for each device, component, or metric.
   - **Stacked Chart**
     View a stacked line on top of each other for each device, component, or metric.
2. Select one of the following **Standardized Axis** options for the Y-axis of each trend chart:
   - **Fixed at 0 to 100**

Maintain a static range, 0 through 100, for the Y-axis.
  - **Calculated**
    Let the Y-axis adjust dynamically, based on the range of metric values that are included. The calculated setting is applied to all charts in the view.
  - **Scale per Chart**
    Let the Y-axis adjust dynamically, based on the range of metric values for each chart in the view. The Y-axis scaling of one chart in a view does not affect the scaling of any other chart.
3. Optionally, enable **Baseline Metrics**.
   You can determine whether utilization trends are changing using the baseline trend line. The baseline data that is plotted in many views shows statistical deviations from "normal" performance for a given statistic.
   For more information, see Baseline Calculations.
4. If you are adding the dynamic trend view to a dashboard, set the **Context** to one of the following:
   - **Dynamic:** A Dynamic context indicates that the context of the view changes with the context of the dashboard page.
   - **Fixed:** A Fixed context indicates that the view uses a specified group, device, or component as a context for the data.

Gauge Views

Gauge views highlight where the value of a metric differs from the expected value. The out of the box gauge views display percentage metrics. However, custom gauge views do allow for configuring metric values that are not of type percentage.

Business hours filtering can apply to the data in gauge views. The applied business hours filter appears in the subtitle. For more information, see Configure Business Hours Filtering.

Gauge View Elements

The gauge shows the minimum, maximum, and average values in the time range. The gauge needle shows the average value. The gray inner arc shows the minimum and maximum range of the metric value.

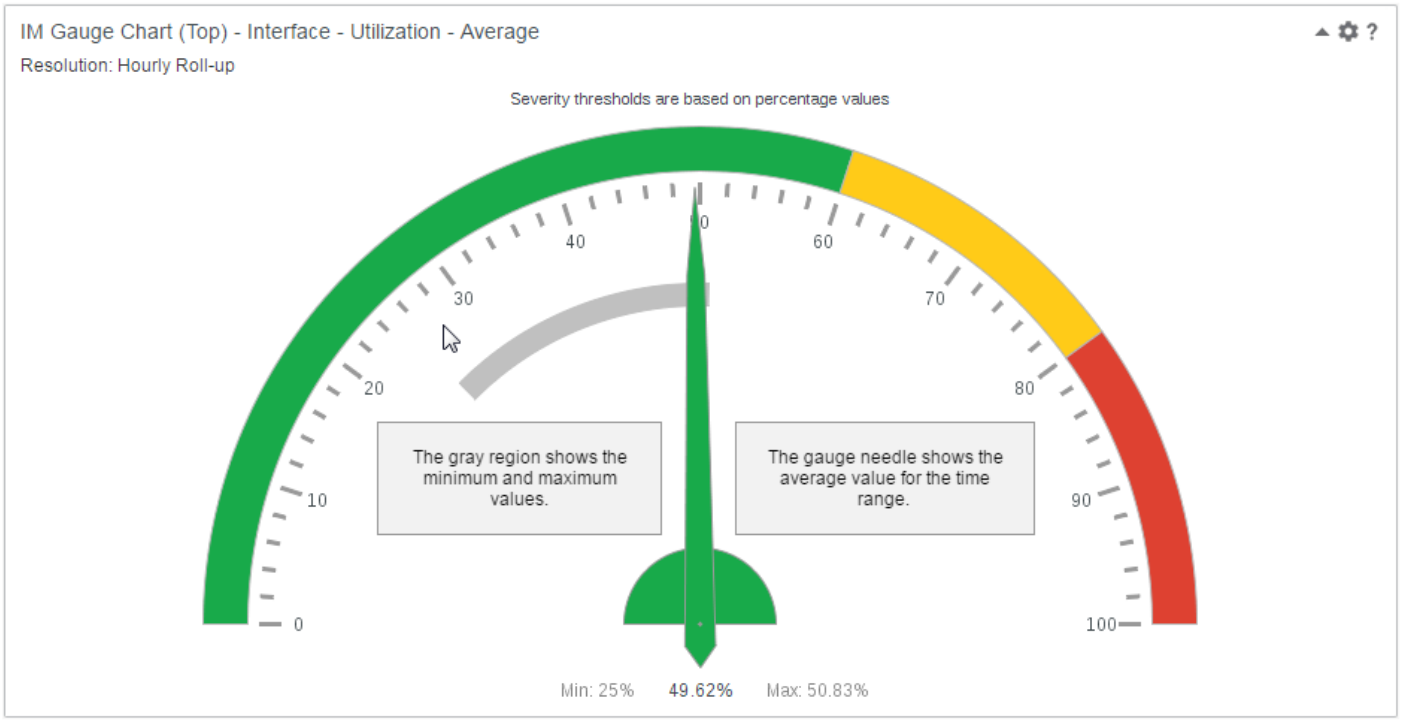
The outer arc uses the following colors to show customizable threshold ranges:

- **Green** The normal status range
- **Yellow** The moderate status range
- **Red** The critical status range

The gauge chart legend shows the resolution, which specifies the granularity of the reporting data: as polled, hourly rollup, or daily rollup. For custom views, the minimum and maximize thresholds are based on this resolution. For out of the box views, the minimum and maximum thresholds are based on the as polled resolution.

The following example shows the important elements of a gauge view:

Figure 20: Gauge View Elements



Configure Gauge Views

All gauge views let you customize the threshold values for metric status. The IM Gauge Chart custom view provides more configuration options.

For standard view configuration details, see Customize Views.

Follow these steps:

1. Select the **Metric Needle** from the list of metrics for the selected metric family. This option determines the selected metric for the view.
2. Do one of the following tasks:
   - If you are configuring a custom view, select whether **Gauge End Points** are calculated or fixed. This option determines the range of the gauge.
     - If fixed, define the minimum and maximum values.
     - If calculated, the view determines the gauge end points based on the minimum and maximum values of the data from the appropriate item level.
   - If you are configuring an out of the box view, the gauge end points are fixed. Define the minimum and maximum values.
3. If you are configuring a custom view, select whether **Determine Threshold** is by percentage value or by numeric value.
   - **By Percentage Value**
     The status start values are a percentage of the range between the gauge endpoints.
   - **By Numeric Value**
     The status start values are fixed numeric values.
4. Define the thresholds for the gauge:

- **Moderate Status Start** The gauge shows yellow for this zone.
- **Critical Status Start**
  The gauge shows red for this zone.

  TIP

  For metrics where low values are bad and high values are good, such as availability, set Moderate Status Start higher than Critical Status Start.

5. If you are configuring a custom view, select **Scaled** or **Unscaled** to specify whether the values in the view are scaled. Scaled values appear with larger units, for example, 1 KB. Unscaled values appear in the raw form for the metric, for example, 1000 bytes.

Gauge/Table Views

Gauge/table views highlight where the value of one metric differs from the expected value when compared to other metrics. The gauge shows the minimum, maximum, and average values for one item in the time range. The table lets you see the values for all items and select an item to show on the gauge. For more information about these views, see Table Views.

Group Scorecard Trend Views

Group scorecard trend views display performance metrics by subgroup, device, or component for the selected group. Group scorecard trend views provide line-of-business owners a group-level summary of how key metrics perform over time. Performance is based on a set of user-defined thresholds. These views incorporate red, orange, yellow, and green icons as visual indicators of performance levels.

Group scorecard trend views show the average, minimum, and maximum utilization, or the selected metric at different time intervals for a group. Each interval length is equal to the time range selected for the dashboard. Out-of-the-box views show either the utilization average or the 95 percentile. The IM Group Scorecard Trend view shows any selected metric.

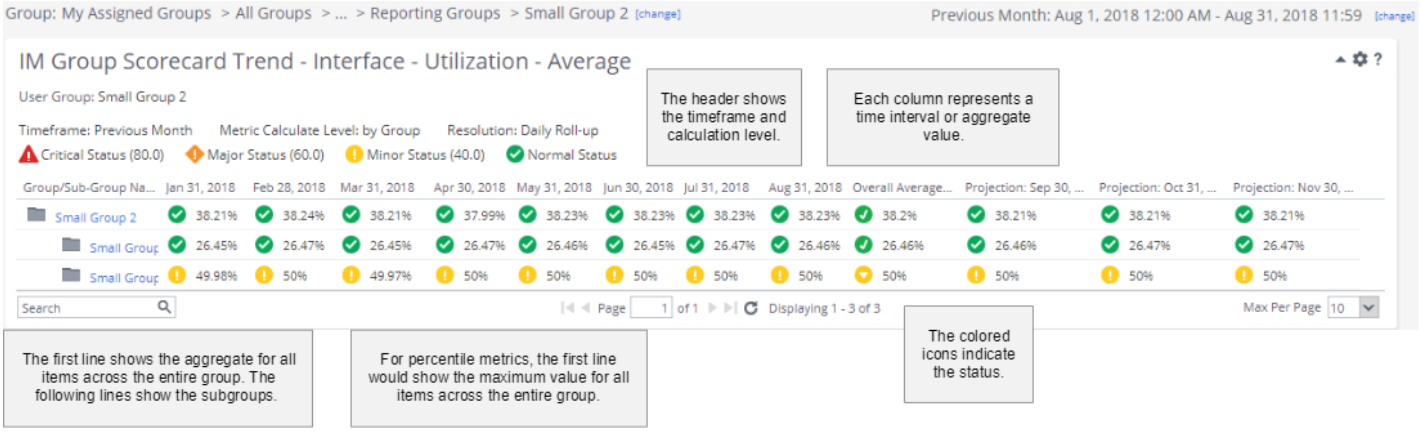
You can select counter metrics (for example, Bits Out - Total). However, group scorecard trend view thresholds are intended for gauge metrics (for example, Bits Out - Average Rate).

Group scorecard table views are also available, which let you view multiple metrics in the same scorecard. For more information, see Group Scorecard Table Views.

The following images show the important elements of a scorecard view.

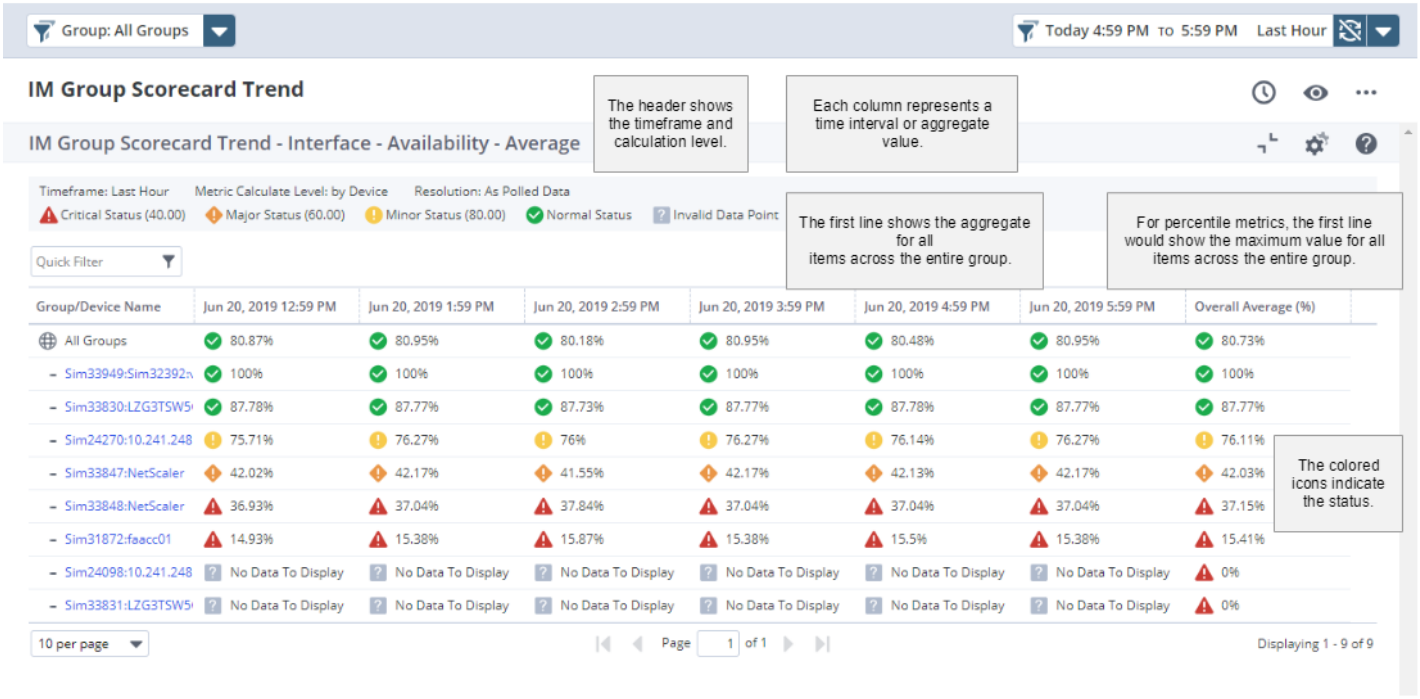
The Metric Calculate Level in the following image is set to Group:

Figure 21: Group Scorecard Trend



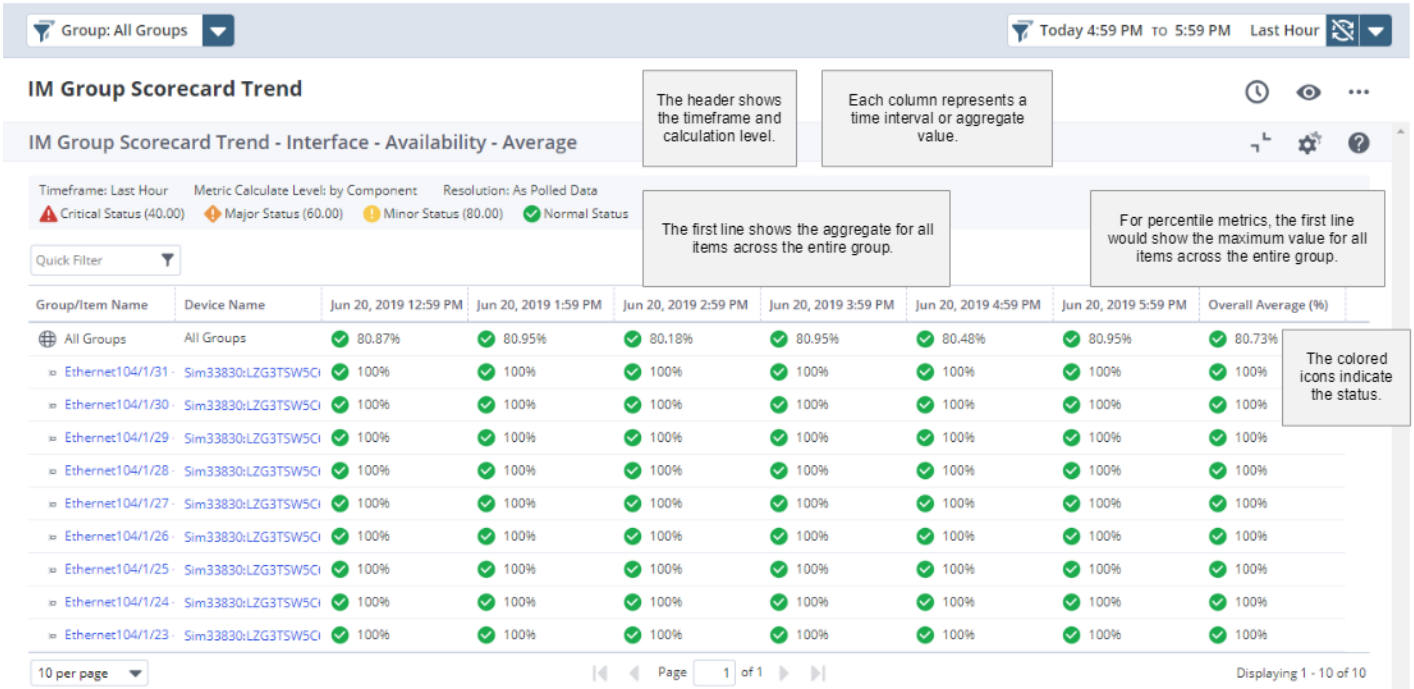
The Metric Calculate Level in the following image is set to Device:

Figure 22: Group Scorecard Trend Device



The Metric Calculate Level in the following image is set to Component:

Figure 23: Group Scorecard Trend Component



NOTE

If a time period renders nonnumeric value for the reporting metric, the scorecard view and header contains a gray icon. This icon indicates that the data for that timeframe includes an invalid data point.

Group scorecard trend views calculate and display projected values as follows:

- For monthly intervals, the first projection is the last day of the following month.
- For weekly intervals, the first projection is midnight on the following Sunday.
- For daily intervals, the first projection is the following midnight.
- For hourly intervals, the first projection is the top of the following hour.

IM Group Scorecard Trend View

The IM Group Scorecard Trend view is a fully customizable scorecard. This scorecard displays historical time intervals for any metric, and up to three projected values. The projected values are calculated from the trend of historical data in the time range of the view. The items in a group determine the metrics that are available in this view.

Interface 95th Maximum Percentile Trend Scorecard View

Most group scorecard trend views always attempt to use daily data unless your data retention policy prevents it. However, the Interface 95th Maximum Percentile Trend Scorecard view uses polled data for accuracy.

When you select larger time ranges (for example, Previous Month), you can easily surpass the configured data retention rate, which limits the number of columns available for comparison.

Configure Group Scorecard Trend Views

Group scorecard trend views include standard view configuration options, such as time filters and context settings. For standard view configuration details, see Customize Views.

TIP

On dashboards, scorecards render best in single-column layouts. Scorecards with fewer columns also render well in two-column layouts. Do not use scorecards in three-column layouts.

The following configuration options apply only to scorecard views.

Follow these steps:

1. (Trend and Custom Scorecard Views Only) Select the number of time intervals.
   Each time interval is one column on the view.

   NOTE

   This setting directly relates to the Time Range setting for the view. The time range for the view determines the length of each time interval. For example, if you select Last 30 days for the Time Range, each interval equals 30 days.
2. (Custom Scorecard View Only) Configure scorecard projection.
   - Select the number of projections to display.
     Each projection is a column on the view.
   - Select the calculation method for the projection.

   For more information, see Scorecard Projection.
3. Select the Metric Calculate Level.
   This option determines what level of aggregation each row in the scorecard view represents: a subgroup, a device, or a component.
4. Specify the thresholds for the status indicators.

If the value for the metric is at or above the specified value, the status icon in the view indicates the status by color. Red indicates critical, orange indicates major, yellow indicates minor, and green indicates normal.
To remove the status indicators, set the threshold values for all status levels to zero.

TIP

For metrics where low values are bad and high values are good, set Minor highest and Critical lowest.

5. Set the results limit.
   This limit determines the number of items included in the data results shown on each page of the scorecard.
   Default: 10

Scorecard Projection

Scorecard projections use a customizable set of data points to predict future values for metrics. Projected values are calculated when the view is rendered, and are based on the historical time frame of the view. You can add the projected values to IM Custom View Group Scorecards.

TIP

Do not use scorecard projections for error metrics. Each error is a discrete event that is not affected by historical errors.

The scorecard view includes two methods to calculate projections:

- **Approximation**
  The Approximation method uses the average from each time frame in the view to calculate the projection values. DX NetOps Performance Management calculates a least squares regression on the averages, then uses the line equation to project future values. This calculation method is faster than the Detailed Data method.
- **Detailed Data**
  The Detailed Data method uses the polled data for the entire time frame of the view. DX NetOps Performance Management calculates a least squares regression for the entire set of data points. This calculation is more statistically accurate than the Approximation method, and provides extra columns in the view.

NOTE

Detailed data scorecard projections are supported only for gauge metrics (for example, Bits Out - Average Rate). Detailed data scorecard projections are *not* supported for counter metrics (for example, Bits Out - Total). Projection values are calculated on the As Polled (rate) data to ensure precision.

These columns are hidden by default:

- **Slope**
  Indicates the slope of the line equation.
- **Intercept**
  Indicates the intercept of the line equation.
- **Degrees**
  Degrees of freedom, which indicates the sample size.
- **Linear Fit**
  Indicates the confidence level of the projected values as related to the sample data.
- **Days to Threshold**
  Indicates the projected number of days before the specified critical threshold is reached.

Group Scorecard Table Views

Group scorecard table views let you view multiple metrics in the same scorecard. Group scorecard table views display the metrics by subgroup, device, or component for the selected group. These views incorporate red, orange, yellow, and green icons as visual indicators of performance levels. The color of the status icons is based on a set of user-defined thresholds. You can specify separate thresholds for each metric.

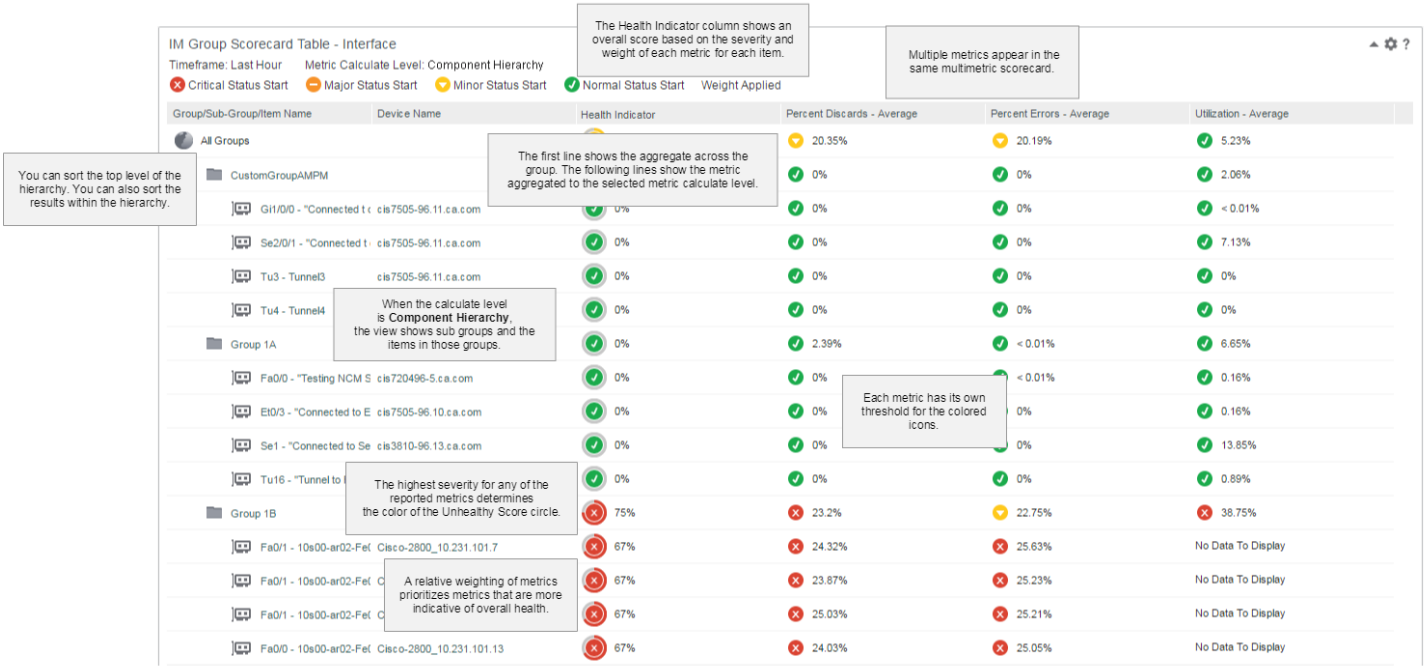
You can select counter metrics (for example, Bits Out - Total). However, group scorecard table view thresholds are intended for gauge metrics (for example, Bits Out - Average Rate).

Group scorecard trend views are also available, which provide line-of-business owners a group-level summary of how key metrics perform over time. For more information, see Group Scorecard Trend Views.

The following images show the important elements of a group scorecard table view. The Health Indicator column is shown.

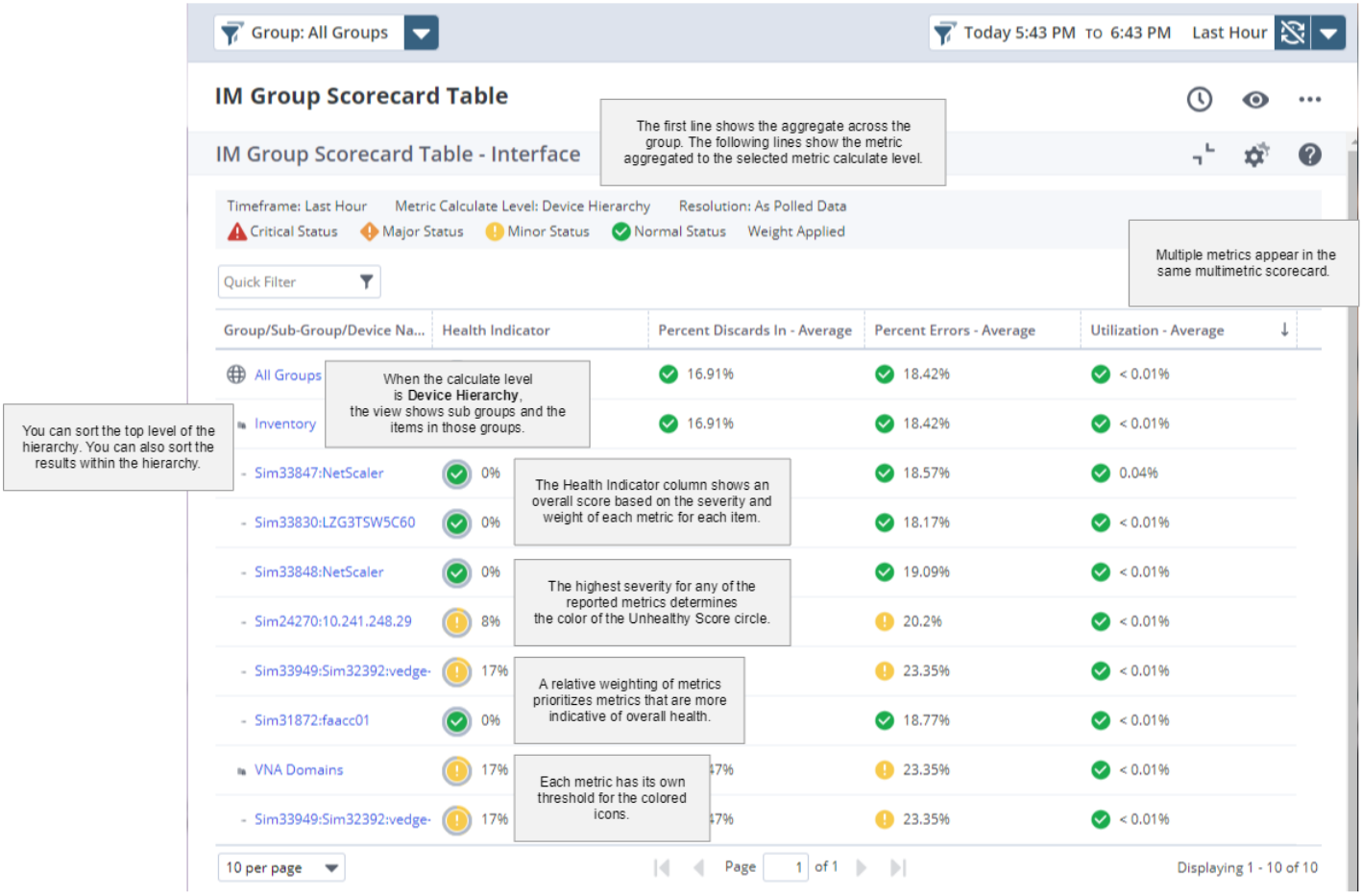
The Metric Calculate Level in the following image is set to Component Hierarchy.

Figure 24: Group Scorecard Table Component Hierarchy



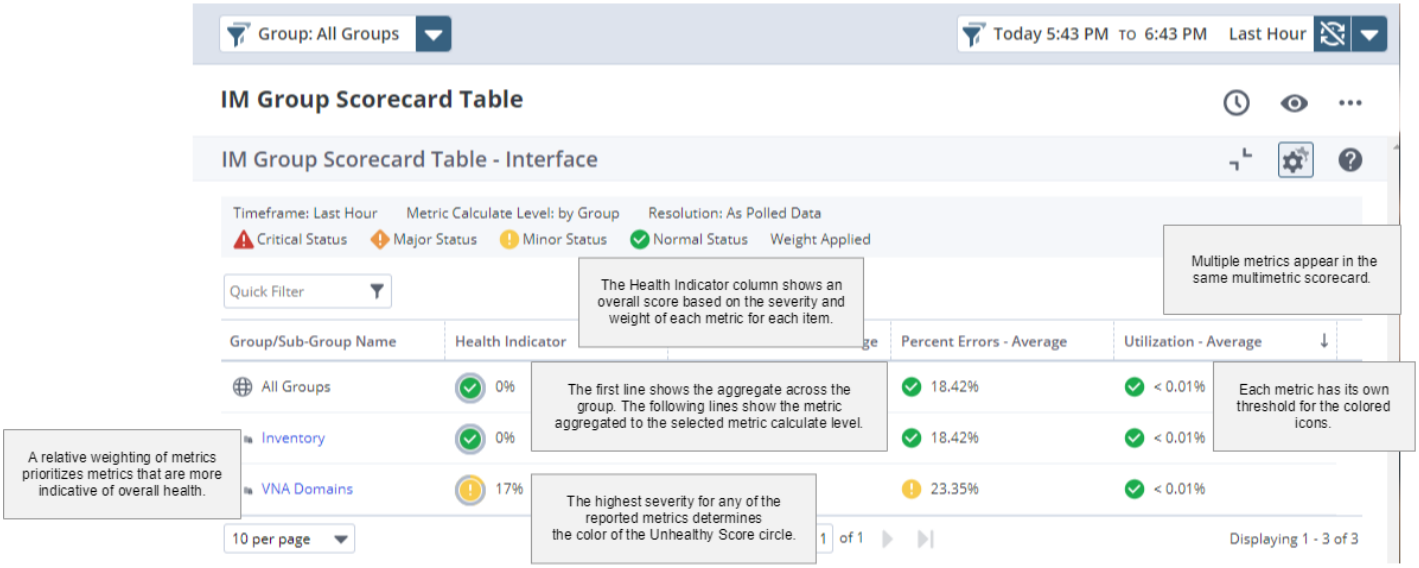
The Metric Calculate Level in the following image is set to Device Hierarchy:

Figure 25: Group Scorecard Table Device Hierarchy



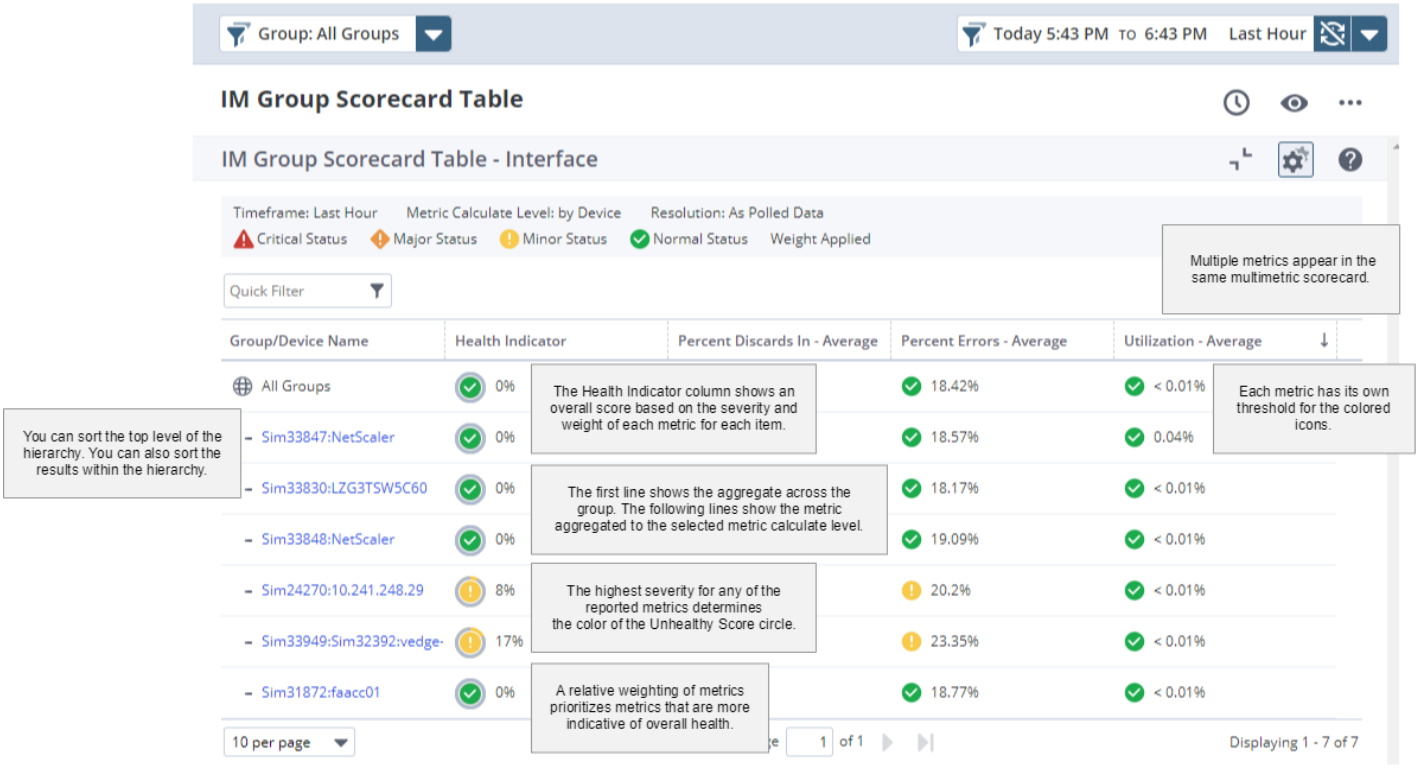
The Metric Calculate Level in the following image is set to Group:

Figure 26: Group Scorecard Table Group



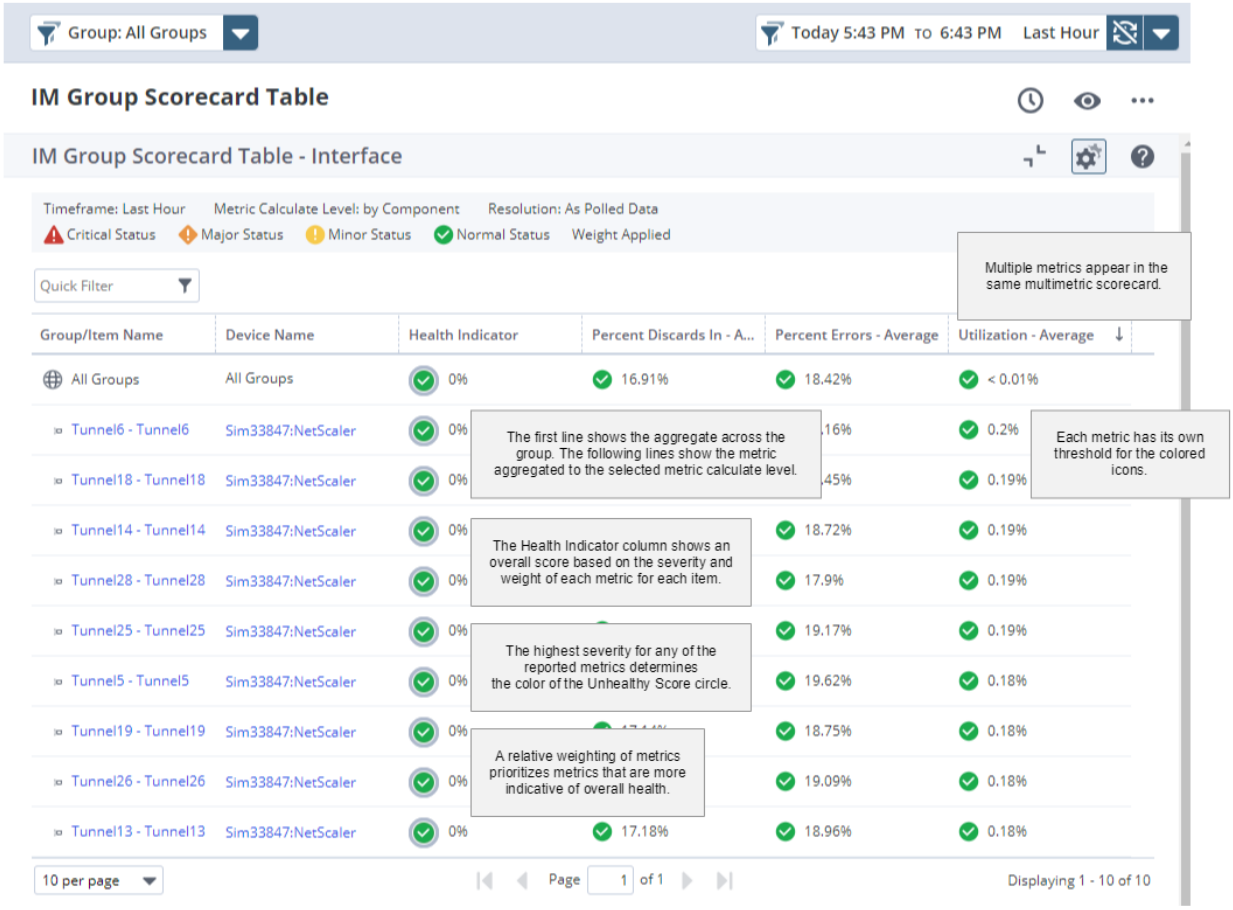
The Metric Calculate Level in the following image is set to Device:

Figure 27: Group Scorecard Table Device



The Metric Calculate Level in the following image is set to Component:

Figure 28: Group Scorecard Table Component



Configure a Group Scorecard Table View

To display a group scorecard table view, add the IM Group Scorecard Table view. Group scorecard table views include standard view configuration options, such as time filters and context settings. For standard view configuration details, see Customize Views.

TIP

On dashboards, group scorecard tables render best in single-column layouts. Group scorecard tables with fewer columns also render well in two-column layouts. Do not use group scorecard tables in three-column layouts.

Follow these steps:

1. **Show** or **Hide** the Health Indicator column.
2. Select the **Metric Calculate Level**.
   This option determines what level of aggregation each row in the scorecard view represents: a group, a device, a component, a device hierarchy, or a component hierarchy.
   Component Hierarchy shows aggregation to the current group and to each group that is a child of that group.
3. Set the **Results Limit**. This limit determines the number of items that are included in the data results shown on each page of the scorecard.

NOTE

If the **Results Limit** is less than the child item count, only the data results within the limit are aggregated. For accuracy, we recommend the **Results Limit** exceed the child item count. However, your child item count might exceed the supported maximum limit to meet performance expectations. In this scenario, we recommend that each parent group contain a smaller set of child items.

4. For each metric, specify the thresholds for the status indicators.
   If the value for the metric is at or above the specified value, the icon indicates the status by color.
   Red indicates critical, orange indicates major, yellow indicates minor, and green indicates normal.
   To remove the status indicators, set the threshold values for all status levels to zero.
   You can specify thresholds with up to two decimals places with the lowest value being 0.01. Values with more than two decimal places are rounded.

   TIP

   For metrics where low values are bad and high values are good, set Minor highest and Critical lowest. To omit a severity level, set the threshold value to zero.
5. If the Health Indicator column is shown, for each metric, specify a weight:
   - **0**
     This weight shows the metric in the table, but does not factor the metric into the Health Indicator.
   - **1**
     This weight applies a standard weight to the metric when calculating the Health Indicator.
   - **2**
     This weight doubles the metric value when calculating the Health Indicator.
   - **3**
     This weight triples the metric value when calculating the Health Indicator.

Map Views

Map views are available for SD-WAN tunnels and application/SLA paths only. Map views show the location of each site for the selected group. You must select a DX NetOps Virtual Network Assurance-based site. Sites with no data appear as gray icons. In previous versions, sites with no data were excluded from the Map view. Map views show only a single level below the selected group

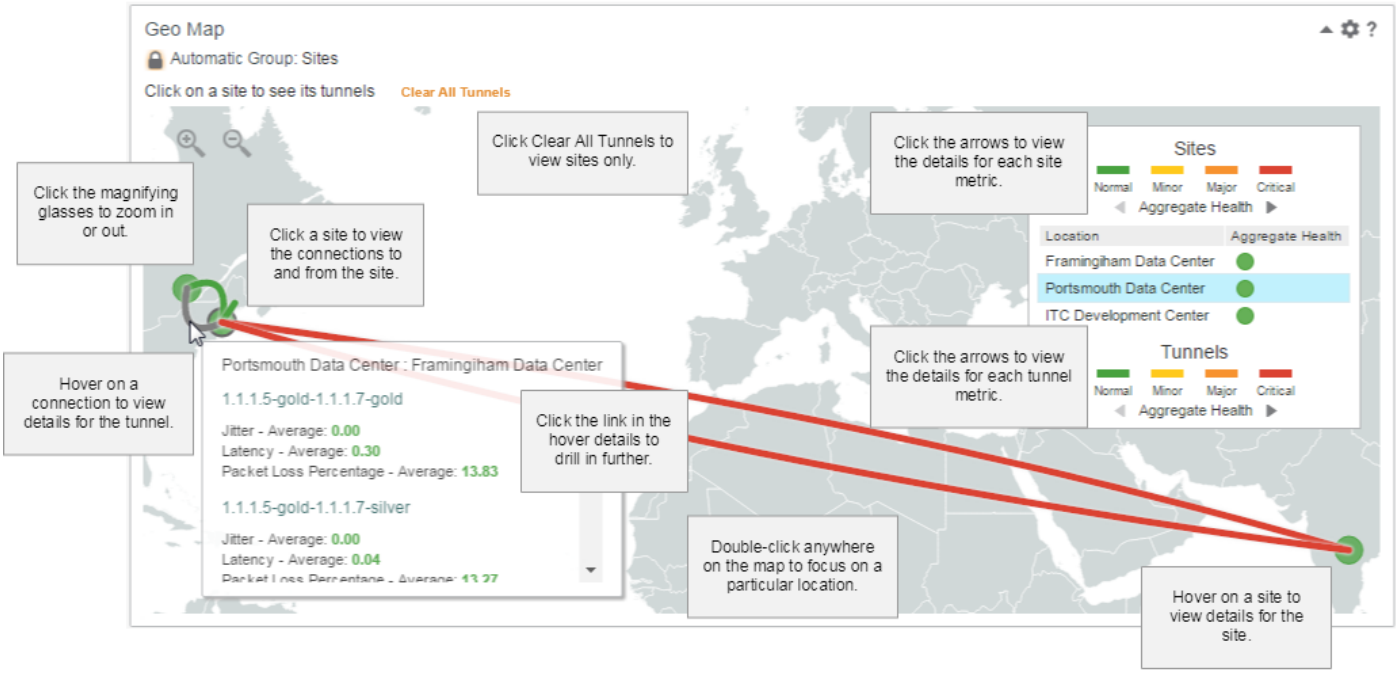
Example: A USA site group contains site groups for states. The site groups for states contain site groups for cities. If you select USA, the map view shows the site groups for states, but not for cities. If you select a state, the map view shows the site groups for cities within the state.

When a site is selected, the connections to and from other sites appear. The connection lines are color-coded based on health metrics. Site router details appear when you hover over a site. Tunnel or application/SLA path details appear when you hover over a connection.

For more information about monitoring SD-WAN, see Monitor SD-WAN.

The following example shows the important elements of a map view for tunnels:

Figure 29: Map_View



Configure a Map View

Map views include standard view configuration options, such as time filters and context settings. For standard view configuration details, see Customize Views.

> **NOTE**
>
> Map views are available only for SD-WAN tunnels and application/SLA paths. Other managed item types are unsupported.

Follow these steps:

1. Specify the thresholds for each site metric (CPU utilization, memory utilization, and virtual interface utilization).
   If the value for the site is at or above the specified value, the icon indicates the status by color.
   Green indicates normal, yellow indicates minor, orange indicates major, and red indicates critical.
   You can specify thresholds with up to two decimals places with the lowest value being 0.01. Values with more than two decimal places are rounded.

   > **TIP**
   >
   > For metrics where low values are bad and high values are good, set Minor highest and Critical lowest. To omit a severity level, set the threshold value to zero.
2. Specify the thresholds for each tunnel or application/SLA path metric.

Pie Chart Views

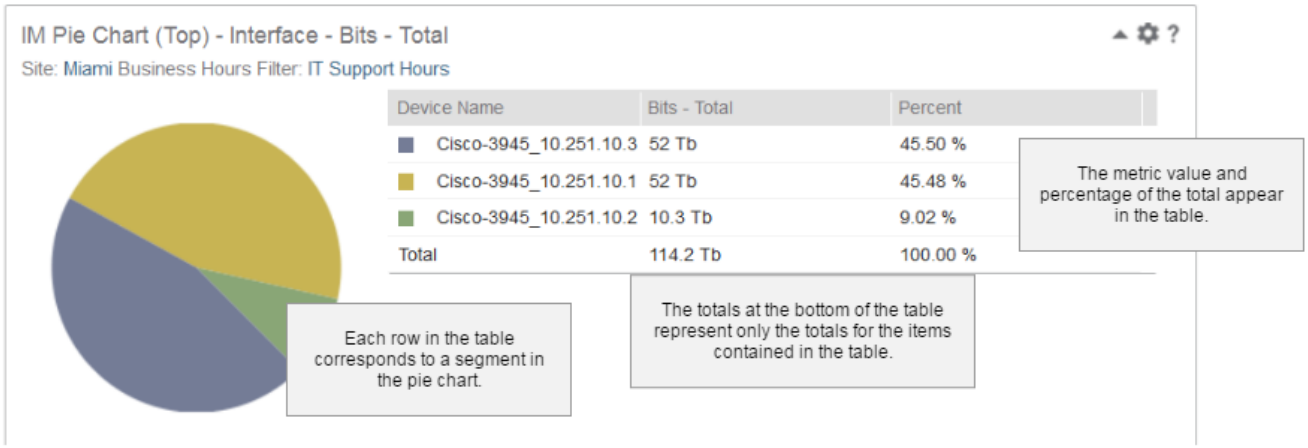
Pie charts show the relative values of a metric. Use pie charts to view metric values that represent parts of a whole. Pie charts are best used for small groups of items. This view includes a table that shows the metric value and the percentage of the total value for each item.

You can apply a business hours filter to the data in pie chart views. The applied business hours filter appears in the subtitle.

For more information about how to configure business hours definitions, see Configure Business Hours Filtering.

The following example shows the important elements of a pie chart view:

Figure 30: Pie Chart Elements



To set up a pie chart, configure the following properties:

- **Metric Label**
  The selected attribute (Device Names, Item name, or Description) appears as the first column of the table. When the selected attribute is Device Name, the metrics are aggregated at the device level.
- **Sort Direction**
  The pie chart includes the items for the selected metric in descending or ascending order.
- **Max Rows**The maximum number of table rows determines the maximum segments that appear in the pie chart.

For more information about how to configure a standard view, see Customize Views.

Pie/Table Views

For more information about pie/table views, see Table Views.

Table Views

Table views are the most useful way to show vertically and horizontally dense data, include multiple metrics, many items, or many properties. Tables let you sort the data according to any metrics, so you can view the best and worst results for each metric.

Business hours filtering can apply to the data in table views. The applied business hours filter appears in the subtitle. For more information, see Configure Business Hours Filtering.

Table View Options

Table views provide the following functionality:

- To drill down to detailed data for individual items, click the item name.
- To sort by a particular column or reverse the sort order, click the column heading. An arrow icon indicates the sorted column.

When you sort by a different column, DX NetOps Performance Management queries the data source to get new results for the table that match the top values for the new sorted column. The sort order applies only to the currently rendered view. The sort order does not persist and cannot be saved on the view.

- For column options, mouseover column heading and click the gear icon. Use the following options to manage columns:
  - **Sort Ascending/Sort Descending**
    Sort the entire data set from the data source.
  - **Sort Results Ascending/Sort Results Descending**
    Sort the already retrieved and rendered result set.
  - **Add/Hide Columns**
    Expand the Columns menu, and select or clear columns for the table. This list includes the selected metrics from the view configuration and more details about the items, such as Alias and Life Cycle State.
  - **Move**
    Click and drag the column header to the desired location.
  - **Resize**
    Hover over the separator line between two columns in the grid header, and click and drag the column to be wider or narrower.
  - **Save**
    Save the column settings for the view at the tenant, user, or session level. The save operation is applied only to the column selection and the order of columns.

    NOTE

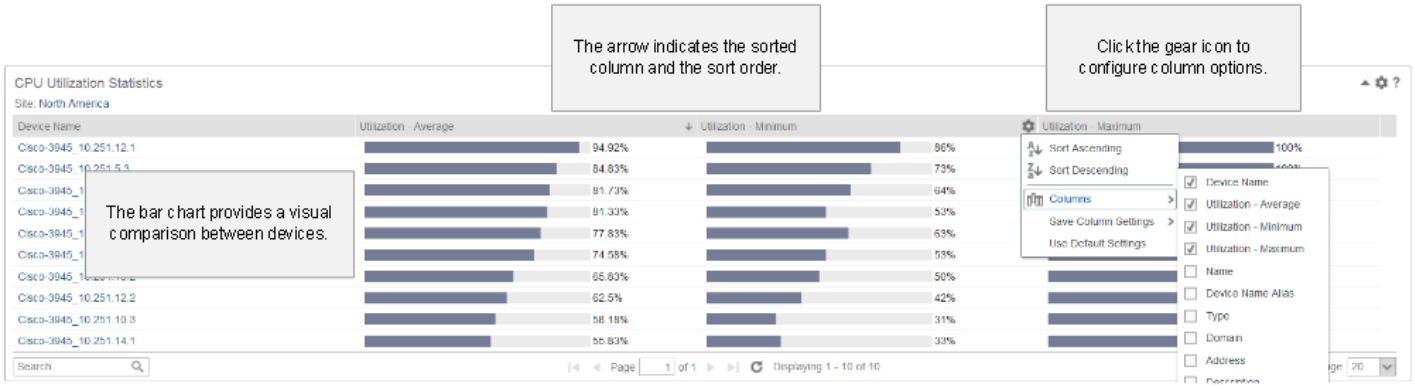
    Save does not preserve column sort order.
  - **Use Default Settings**
    Revert any custom changes applied to the view column settings.
- By default, **Enable Scrolling** is selected and the Search bar is locked in place when you display more than 20 rows. To change this setting, click the **Max Per Page** drop-down in the lower-right corner, and change **Enable Scrolling**.

Bar Chart Table Views

Some out-of-the-box views, show tables with a bar chart column. These views show metrics with percentage values. Use these views to identify problematic devices quickly.

The following example shows the important elements of a bar chart table:

Figure 31: Table Bar Chart Elements



Deviations from Normal Table Views

The Top Deviations from Normal views compare actual values from the selected time frame to a calculated baseline value. Use these views to identify places where performance has changed. These views display the items that deviated the most from that normal value. The way the baseline is calculated varies by data source.

The table shows the following information:

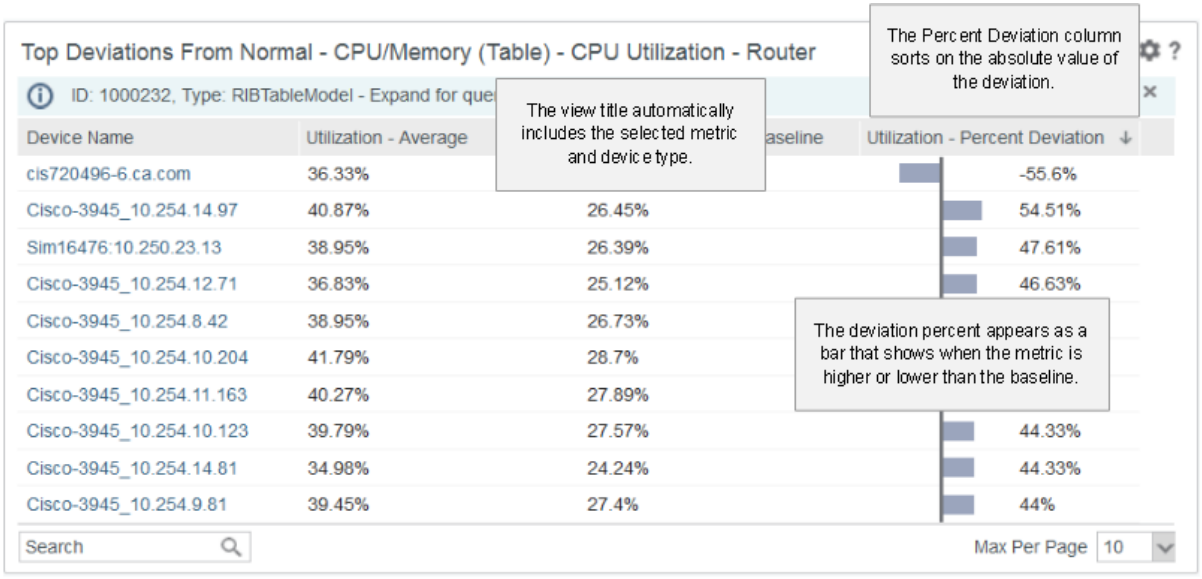
- The average for the selected metric
- The average baseline for the selected metric
- The percent divination with a relative visualization sorted by absolute value

Edit the view to specify the following information:

- The type of device: router, server, or switch
- The selected metric

The following example shows the important elements of a deviations from normal table:

Figure 32: Deviations from Normal Table Elements



Chart/Table Views

Chart/table views combine aspects of other visualizations with table views. These views support only percentage type metrics.

> **NOTE**
>
> When you print a combi-view, only the table portion of the view appears in the PDF.

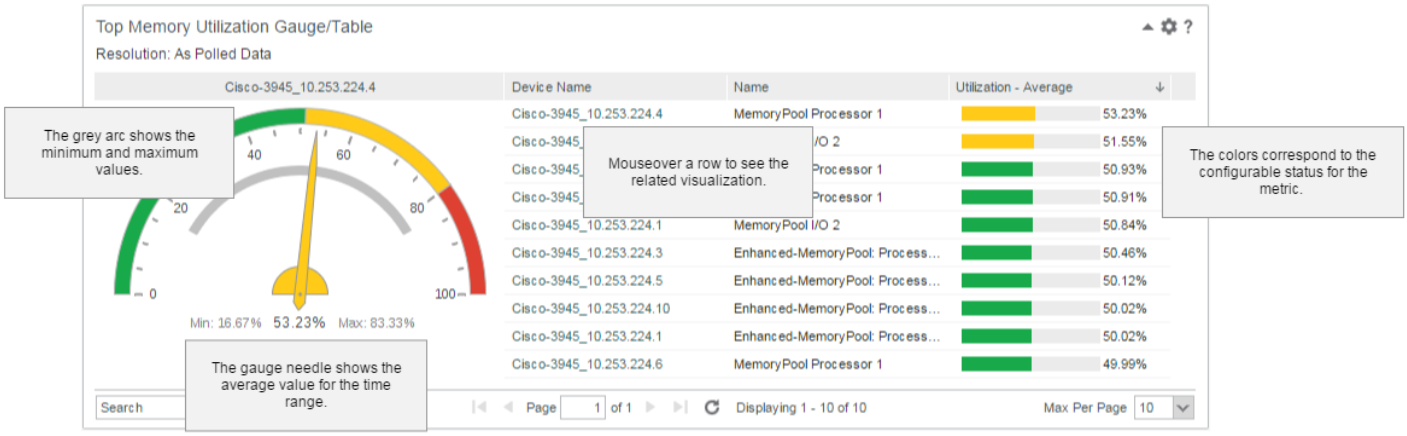
Gauge/Table Views

Gauge/table views highlight where the value of one metric differs from the expected value when compared to other metrics. These views are good for metrics where the value is a percentage. The gauge shows the minimum, maximum, and average values for one item in the time range. The table lets you see the values for all items and select the item to show on the gauge.

The gauge chart legend shows the resolution, which specifies the granularity of the reporting data: as polled, hourly rollup, or daily rollup. For customized views, the minimum and maximize thresholds are based on this resolution. For out of the box views, the minimum and maximum thresholds are based on the as polled resolution.

The following example shows the important elements of a gauge/table view:

Figure 33: Gauge Table Elements



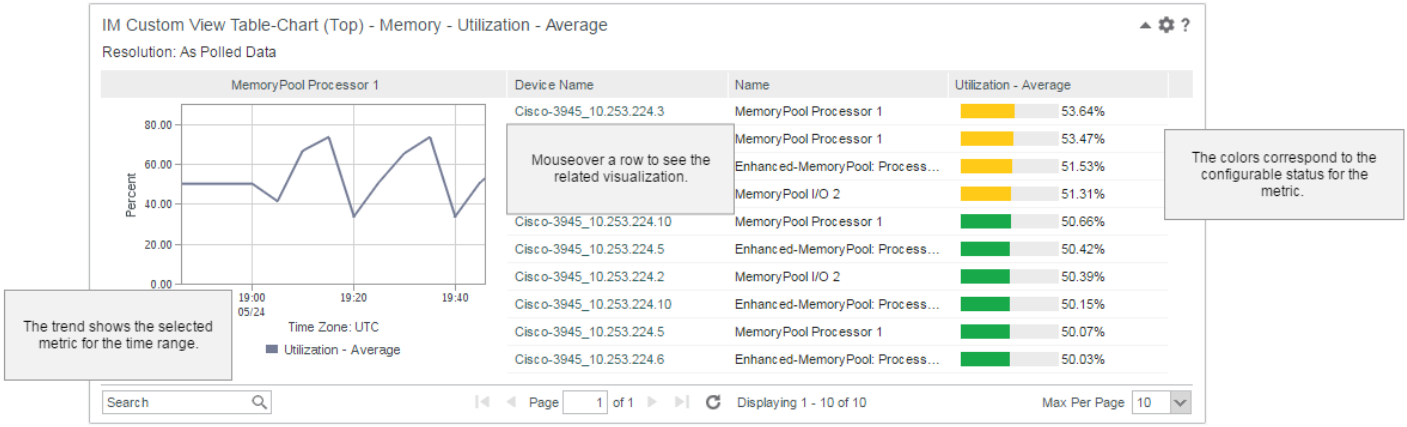
Trend/Table Views

Trend/table views provide the flexibility of table data and use the trend view to show the change in a metric over time. The trend shows the value over the time range for one item. The table lets you see the values for all items and select the item to show on the trend. Percentage-based metrics are represented with a bar.

Most trend/table views have a trend chart that shows the value of a single metric. Some trend/table views have a trend chart that shows the value of multiple metrics. Each metric is represented with its own trend line.

The following example shows the important elements of a trend/table view:

Figure 34: Trend Table Elements

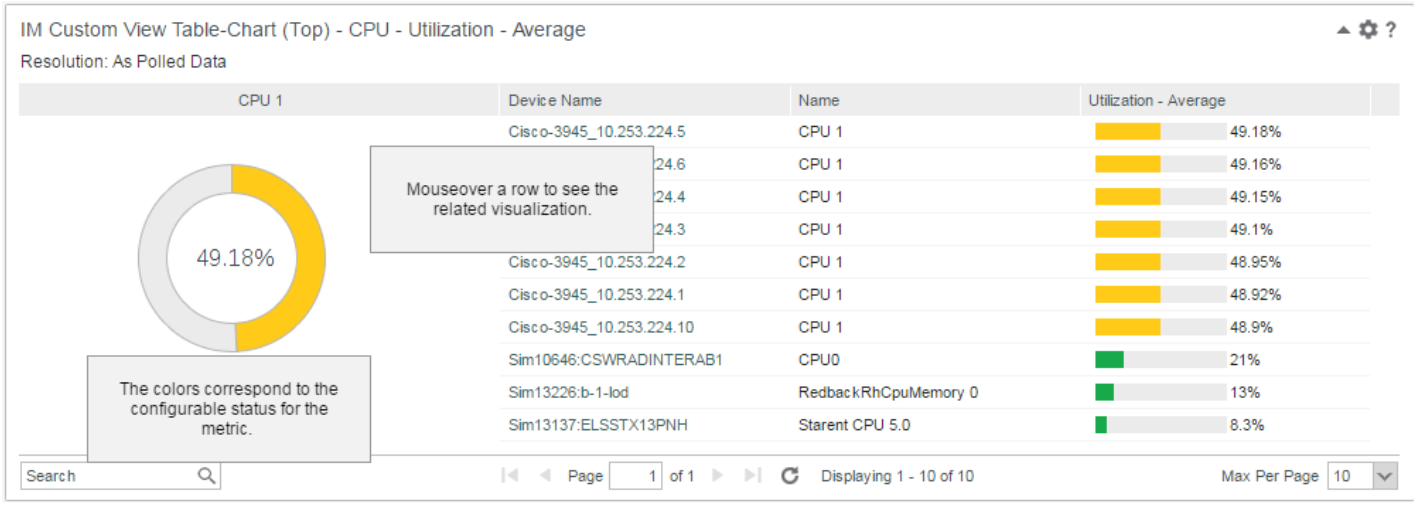


Radial Bar/Table Views

Radial bar/table views show percentage values within a circular bar.

The following example shows the important elements of a radial bar/table view:

Figure 35: Radial_Bar

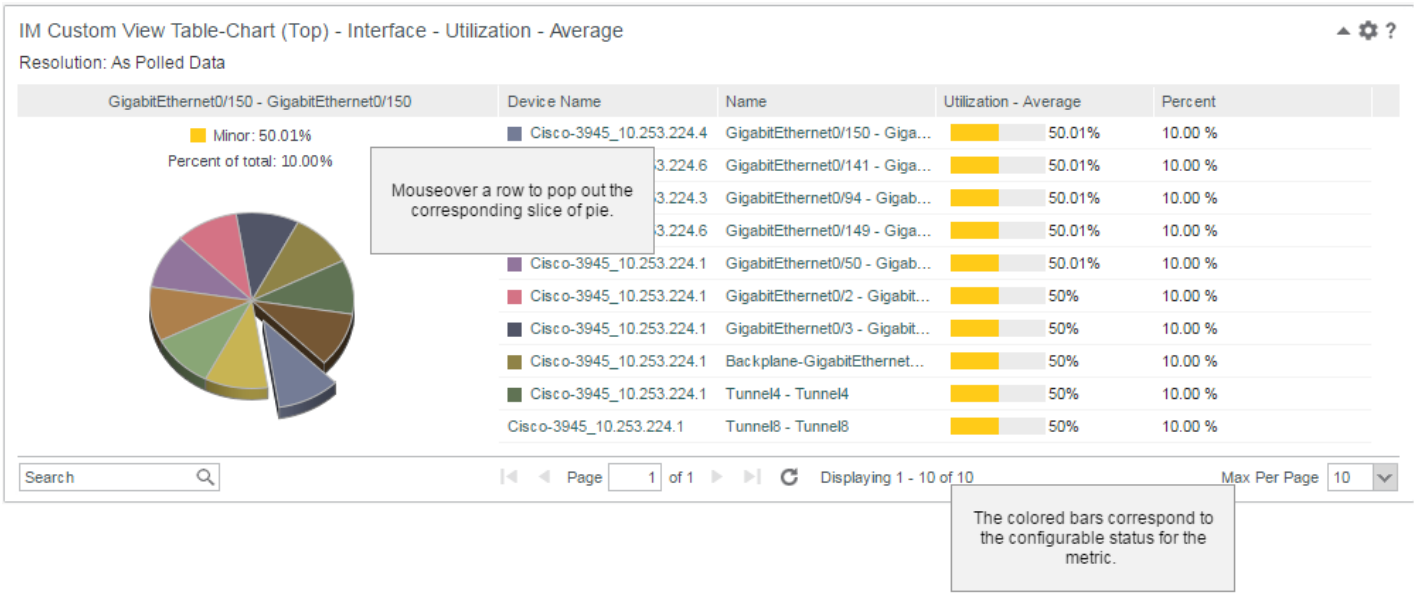


Pie/Table Views

Pie/table views show the relative values of a metric. When you hover over the corresponding item in the table, the corresponding slice of pie pops out. Use pie charts to view metric values that represent parts of a whole. Pie charts are best used for small groups of items.

The following example shows the important elements of a pie/table view:

Figure 36: Pie_Table_View



Configure Chart/Table Views

The IM Custom Chart-Table view is a custom view that reports data from the data aggregator data source.

Not all settings are available for all table view types. For standard view configuration details, see Customize Views.

Follow these steps:

1. Select a **Chart Type**.
2. Select the **Metric Name** from the list of metrics for the selected metric family. This option determines the selected metric for the view.
3. If you are configuring a custom view, select the **Metric Calculate Level**.
   This option determines what level of aggregation each row in the view represents: a device, or a component.
4. Change definitions for the status levels:
   - **Minor Status Start**
     Metrics before this value have a normal status indicated with green.
     Metrics at or after this value have a minor status indicated with yellow.
   - **Major Status Start** Metrics at or after this value have a major status indicated with orange.
   - **Critical Status Start**
     Metrics at or after this value have a critical status indicated with red.

     NOTE

     Some out-of-the-box views show **Moderate Status Start**. For these views, moderate status is yellow and orange is unavailable.

     TIP

     You can configure chart/table views to reflect reverse value severity. For metrics where low values are bad and high values are good, set Green highest and Red lowest. To omit a severity level, set the threshold value to zero. For example, setting Orange to zero removes the major severity when profiling metric thresholds.
5. Select the **Max Rows**. This limit defines the total number of rows that the table can display. If the number of results exceeds the max rows, the table shows the top metrics according to the sorted column.

Configure Table Views

The IM Table (Top) view is a custom view for data from the Data Aggregator data source.

Not all settings are available for all table view types. For standard view configuration details, see Customize Views.

Follow these steps:

1. Select the **Metric Fields** from the selected metric family to display in the table. Each selected field is a column in the table.

   NOTE

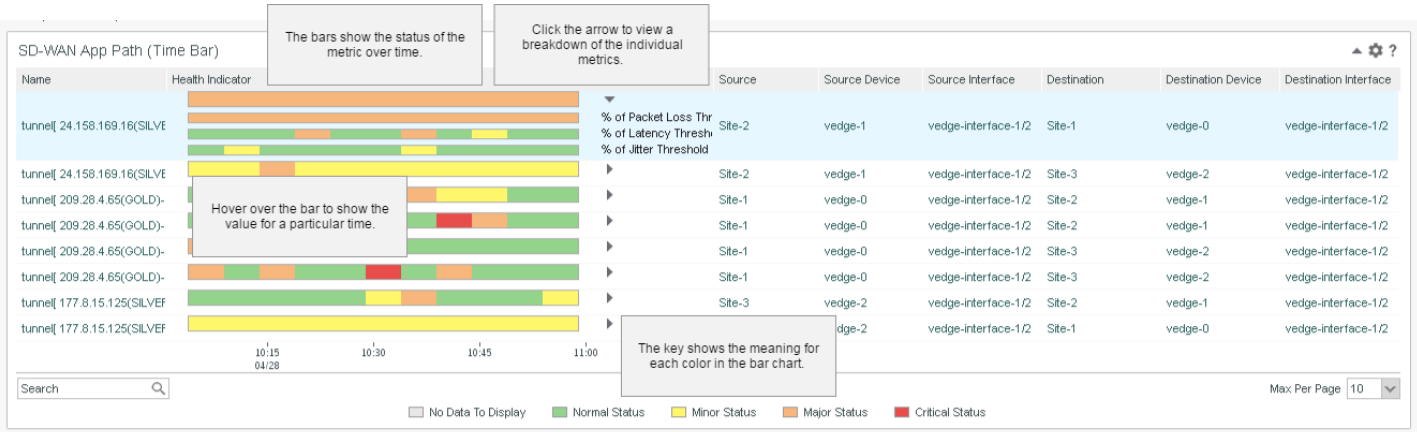
   If you select Device Name without Name, the metrics are aggregated to the device level. If you select Name or Description with or without Device Name, the metrics are aggregated to interface or component level.
2. Select the **Metric Sort**, which defines the column to use to sort the table by default.
3. Select the default **Sort Direction** for the **Metric Sort** column.
4. Select a requested resolution. For more information, see Data Resolution.
5. Select the **Max Rows**. This limit defines the total number of rows that the table can display. If the number of results exceeds the max rows, the table shows the top metrics according to the sorted column.

Time Bar Chart Views

These views use a bar to show the status of percentage metrics by color over the selected time range. Time bar views show threshold values within a time series. The time slices represent the statuses as bars of colors. Hover over a section of the time range to view a tooltip with the corresponding value.

The following example shows the important elements of a time chart:

Figure 37: Time Chart Elements



Sort a Time Bar Chart

Time bar views report only percentage metrics. Percentage values range inclusively from 0% to 100%. Values that are outside that range are marked as invalid, shown as a gray bar, and excluded from the sort order.

Use the Sort Direction in the view settings to sort the highest and lowest items based on the maximum rows across the entire time range. This also sorts the colored bars within the time bar itself.

You can also click the gear icon in the table heading to sort as follows:

- **Sort Ascending/Sort Descending**
  Sort the already retrieved and rendered result set.

Configure a Time Bar Chart

The IM Time Bar Chart is available on dashboards, device context pages, and group context pages. This view is not available on interface or component context pages. For standard view configuration details, see Customize Views.

> **TIP**
>
> On dashboards and in context pages, time bar charts render best in single-column or two-column layouts.

To configure the behavior of a custom time bar chart, specify the following properties:

- **Metric Value**
  Select a specific metric to include in the view. To view a time bar chart aggregating the associated metrics, select the following option:
  - **Health Indicator**
    You can click to expand and view a time bar chart for each metric.
- **Metric Calculate Level**
  This option determines what level of aggregation each row in the view represents: a device, or a component.
- Define the thresholds for the bar:
  - **Critical Status Start**
    The bar shows red for metrics with this value for this time.
  - **Major Status Start**
    The bar shows orange for metrics with this value for this time.
  - **Minor Status Start**
    The bar shows yellow for metrics with this value for this time.

Time Bar/Table Views

For more information about these views, see Table Views.

Trend Views

Trend views show the value of a metric across time. Trends show spikes or dips in activity and help identify whether a metric is increasing or decreasing. Trend views provide one or more of the following chart types:

- **Trend Chart**
  View a traditional trend line for each device, component, or metric.
- **Stacked Chart**
  View a stacked line on top of each other for each device, component, or metric.

  NOTE
  Trend views displaying data up to the present moment might contain dips or spikes for the present moment as data is processed.

Maintenance indicators apply to all the devices and components in a site group. When the associated site group is selected in the context, maintenance indicators appear as shading in trend views.
For more information, see Schedule Maintenance Indicators.

Applying a business hours filter to a trend view displays the same data in the view, but the periods outside of the business hours are shaded.
For more information:

- About business hours filtering, including how to define business hours definitions, see Configure Business Hours Filtering.
- About how to apply a business hours filter to views on a dashboard, see Dashboards.
- About how to apply a business hours filter to views on a context page, see Context Pages.

In this article:

Trend View Options

Trend views let you quickly change the trend lines that are displayed on the graph. The following options also apply to MultiTrend views:

- To remove a metric from the chart temporarily, right-click the metric in the legend, and click **Hide**.
- To add a metric back into the chart, right-click the metric in the legend, and click **Show**.
- To hide all other metrics, right-click a metric in the legend, and click **Focus**.
- To narrow the view on a precise time frame, click and drag across the time frame in the chart.

  NOTE
  Select a time frame of at least 30 minutes. For MultiTrend views, you must drill into an individual trend view before you can select a time frame. If non business hours compression is enabled, zooming is not supported.
- To apply the precise time frame to the entire dashboard, click **Apply to Dashboard** at the bottom of the chart.
- To revert the view to the default settings, click **Undo** in the lower-left corner.

Trend View Element Examples

Trend views can include the following elements:

NOTE
A specific trend view might or might not include the following elements. A specific trend view might include elements in addition to following elements.

- **Baselines**
  If available, the baseline trend line lets you determine whether utilization trends are changing. The baseline data that is plotted in many views shows statistical deviations from "normal" performance for a given statistic.
  For more information, see Baseline Calculations.

  NOTE
  Baseline metrics are processed up to hourly and daily increments. When the resolution is less than an hour, data points are interpolated between known hourly data points.
- **Events**
  If available, events appear as flags indicating the times that the events occurred.

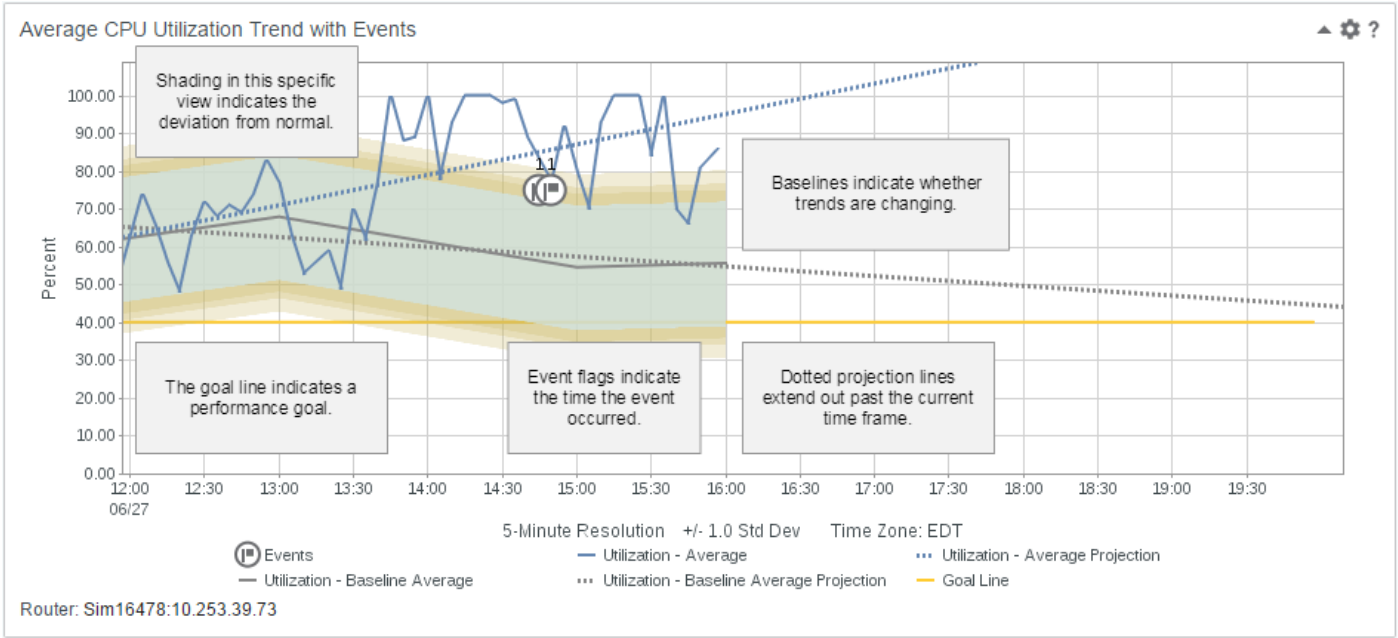
  NOTE
  Events are based on as-polled data. If a rollup is applied to the trend line, a threshold event might appear outside the trend line.

  Trend charts with events are disabled when the time range is greater than 3 months.
- **Trend Projections**
  If available, projections on charts appear as dashed trend lines that predict future performance of metrics. Each projection is based on trends in recent values in the time frame of the chart. Projections help determine whether current performance is within normal bounds.
  By default, each projection includes a future time frame equivalent to one time beyond the current time frame. For example, if the time frame is the Last Hour, the projection shows values for the next hour.
  For views that support projections, you can show or hide the projection data by editing the view settings.

The following image shows the important elements of a trend view:

NOTE
Percentile metrics appear as dashed lines.

Figure 38: TrendView

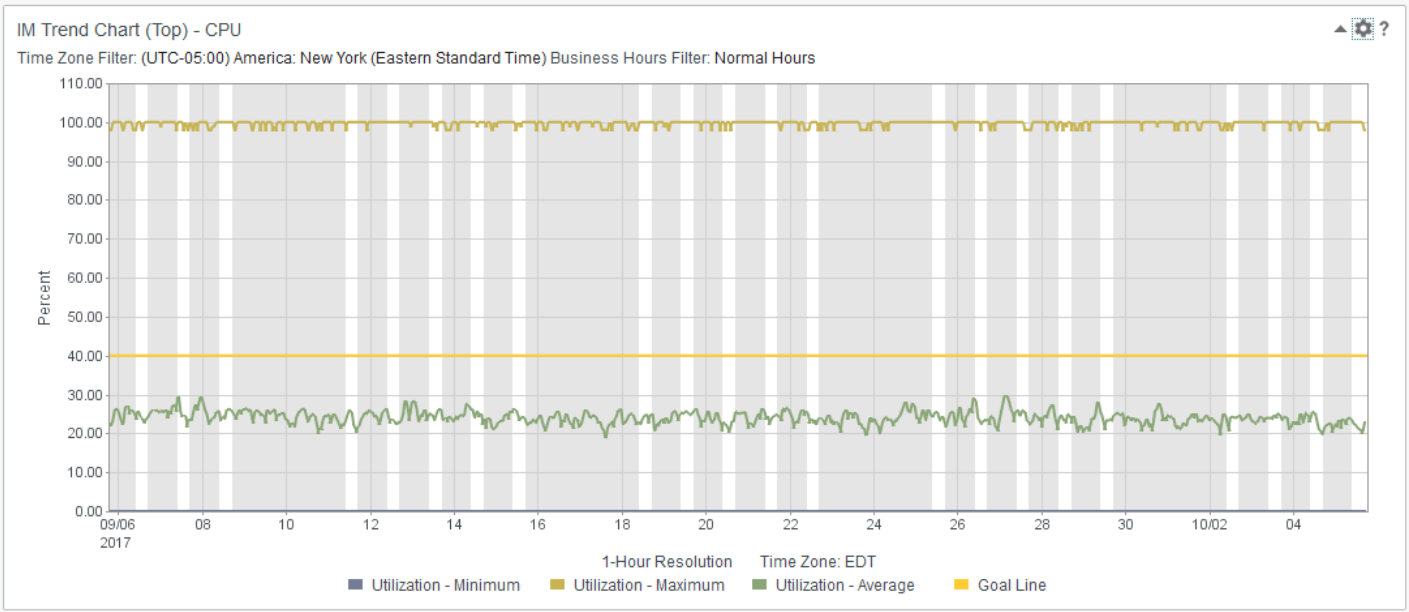


Compress Non Business Hours in a Trend View

If the resolution is less than one day and business hours filtering is configured, trend views apply shading. The shading differentiates between the selected business hours and non business hours.
For more information about how to configure business hours definitions, see Configure Business Hours Filtering.

The following image shows a trend view with shaded non business hours:

Figure 39: Non_Business_Hours

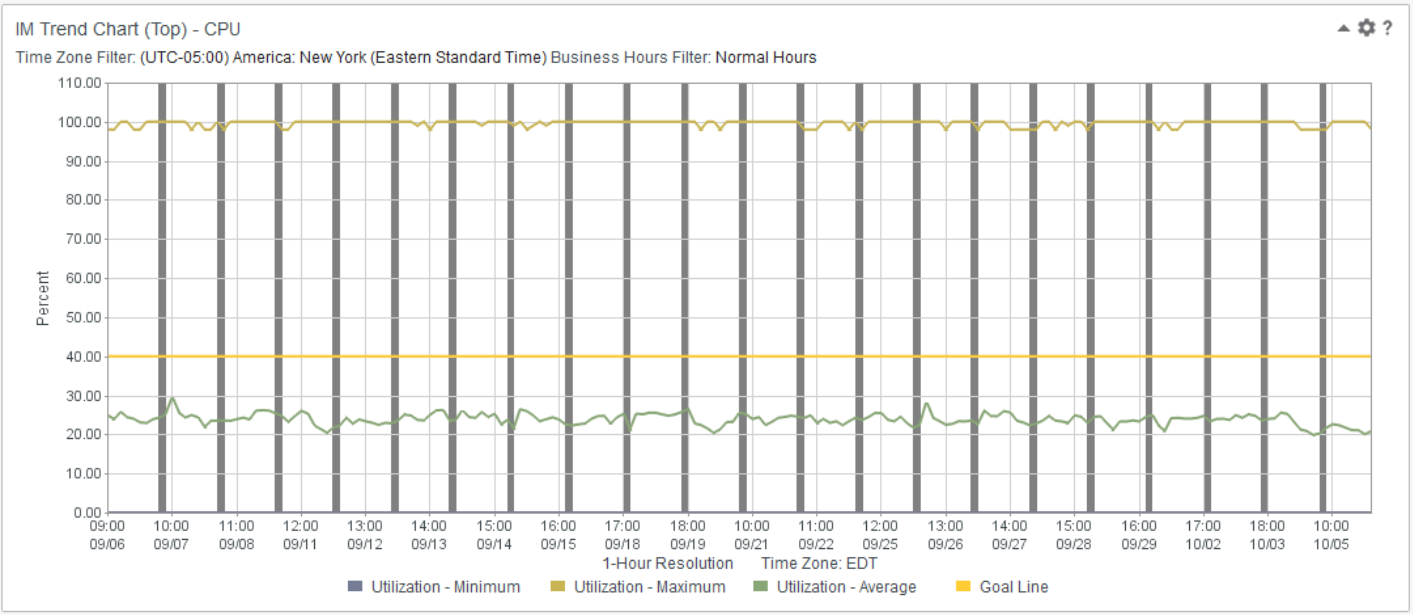


You can also configure a trend view to compress non business hours. If you compress non business hours, the shaded regions are compressed into a single region equal to the resolution period. The resolution is listed at the bottom of the chart.

> **NOTE**
> If non-business hours compression is enabled, you cannot zoom.

The following image shows a trend view with compressed non business hours:

Figure 40: Compressed_Non_Business_Hours



Configure a Trend View

To configure a trend chart, customize or edit a trend view. For standard view configuration details, see Customize Views.

Follow these steps:

1. Select **Show** or **Hide** from the drop-down list to control any of the following trend lines or data points:
   - **Projection**
     The line that plots projected average values for twice the current time period.
   - **95th Percentile**
     The line that plots the values in the top 5 percent of measurements that were taken during the reporting interval.
   - **Events**
     The data points that show when events were reported during the reporting interval.
   - **Goal Line**
     The line that plots the value that you selected as the goal for the metric in this view. This option is available only for some of the trend views.

     NOTE
     Not all trend line options are available for all trend chart types.
2. Set a value for the **Goal Line** if you selected to show it.
   Goal lines serve as visual indicators of fixed metric values. For example, select a value within a critical threshold range to determine whether performance is approaching a degraded level quickly.
3. Supply a label for the goal line to identify the value that you selected.
4. Select whether to **Compress Non Business Hours**.

MultiTrend Views

MultiTrend views let you combine trend data from multiple components in a single-line trend chart. You can compare the data from up to 12 components in a single MultiTrend view. You can use MultiTrend views to view data from the following interfaces:

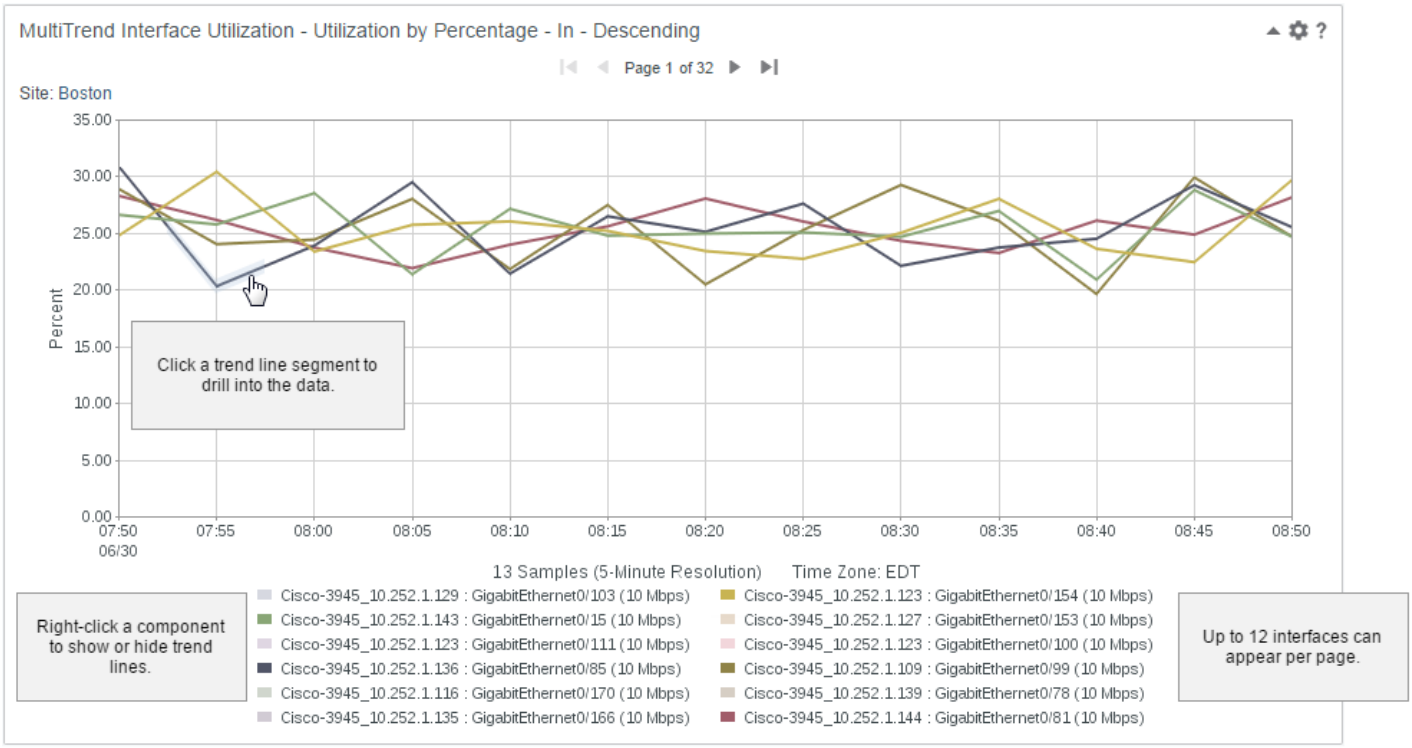
- Multiple interfaces in a group
- Interfaces in your inventory with the worst performance metrics
- Interfaces on a single device

TIP
MultiTrend views require more page space than other types of views. To add a MultiTrend view to a dashboard, add it to a single-column area in the layout. Do not add a MultiTrend view to a small section on a dashboard layout.

MultiTrend views also require more space for the PDF output.

The following image shows the important elements of a MultiTrend view:

Figure 41: MultiTrendView



MultiViews

MultiViews let you combine statistics from multiple interface charts in a single view. Use MultiViews to diagnose performance issues on a device. For example, you can compare the recent performance of all interfaces on a card in a single view. More MultiViews that query for different types of metrics are available in the Device category when you edit a dashboard.

TIP
MultiViews require more page space than other types of views. To add a MultiView to a dashboard, add it to a single-column area in the layout. Do not add a MultiView to a small section on a dashboard layout.

MultiViews also require more space for the PDF output.

The following image shows the important elements of a MultiView:

Figure 42: MultiView

Configure a MultiTrend View

To configure a MultiTrend chart, customize or edit a MultiTrend view. For standard view configuration details, see Customize Views.

Follow these steps:

1. Specify the **Sort Direction**, which is Descending order (highest values first) by default. The sort direction determines which interfaces are reflected in the first pages of trend charts.
2. Select one of the following **Standardized Axis** options for the Y-axis of each trend chart:
   - **Fixed at 0 to 100**
     Maintain a static range, 0 through 100, for the Y-axis.
   - **Calculated**

Let the Y-axis adjust dynamically, based on the range of metric values that are included. The calculated setting is applied to all charts in the view.
- **Scale per Chart**
  Let the Y-axis adjust dynamically, based on the range of metric values for each chart in the view. The Y-axis scaling of one chart in a view does not affect the scaling of any other chart.

3. Select one of the following options for the **Chart Display Order**:
   - **Display by Metric**
     Sort the interface charts by metric value, with the highest (most severe) values shown first, top to bottom.
   - **Display by Name**
     Sort the interface charts in alphanumeric order, by interface name.

Configure a MultiView

To configure a MultiView, customize or edit a MultiView. For standard view configuration details, see Customize Views.

Follow these steps:

1. Select a **Metric Family**. From the **Metrics Fields Available** list, select up to 10 metrics to display in the report. You can select multiple metrics from only one metric family. You can use the arrows to specify an order for the selected metrics and the order is saved.
2. Select the **Metric Sort**, which defines the metric for the primary Y-axis.
3. Specify the **Sort Direction**, which is Descending order (highest values first) by default. The sort direction determines which interfaces are reflected in the first pages of trend charts.
4. Select one of the following options for the **Chart Display Order**:
   - **Display by Metric**
     Sort the charts by metric value, with the highest (most severe) values shown first, top to bottom.
   - **Display by Name**
     Sort the charts in alphanumeric order, by device name and then item name. If the **Metric Calculate Level** is **by Device**, the charts are sorted only by device name.
5. To determine the level of aggregation for metrics, select the **Metric Calculate Level**:
   - **by Component**
   - **by Device**
6. Select one of the following **Standardized Axis** options for the Y-axis of each trend chart:
   - **Fixed at 0 to 100**
     Maintain a static range, 0 through 100, for the Y-axis.
   - **Calculated**
     Let the first Y-axis adjust dynamically, based on the range of metric values that are included. The calculated setting is applied to all charts in the view.
   - **Scale per Chart**
     Let the Y-axis adjust dynamically, based on the range of metric values for each chart in the view. The Y-axis scaling of one chart in a view does not affect the scaling of any other chart.
7. Specify a 1, 2, or 3-column layout.
8. Specify up to 48 charts per pages.
9. Specify a **Maximum Number of Charts** up to 1200.

Trend/Table Views

Trend/table views provide tables with data. As you hover over each metric, a related trend chart displays. The trend chart shows the metric value over the selected time range for each item. The table lets you see the values across the entire time range for each item being charted. Some out-of-the box trend/table views show a column with a line graph.
For more information about these views, see Table Views.

On-Demand Reports

On-demand reports retrieve data sets from specific sets of items or groups without building a dashboard. To view the same data in different ways, change the view type of the report.

You can add on-demand reports to group and device context pages. For group context pages, the reports use the selected group as the context for the view. For device context pages, the reports use the device as the context.

The following role rights apply to on-demand reports:

- **Create On-Demand Report Templates**
  Allows users create on-demand report templates.
- **Run On-Demand Report Templates**
  Allows users run on-demand report templates.

  NOTE
  On-demand reports do not show data for the SD-WAN Tunnel metric family.

Applying a business hours filter to trend views on on-demand reports displays the same data in the view, but the periods outside of the business hours are shaded.

For more information:

- About business hours filtering, including how to define business hours definitions, see Configure Business Hours Filtering.
- About how to apply a business hours filter to views on a dashboard, see Dashboards.
- About how to apply a business hours filter to views on a context page, see Context Pages.

In this article:

On-Demand Report View Types

On-demand report view types determine the chart format. For options that require aggregation, the system determines the aggregation method. For example, Bits has the counter metric profile type and uses sum. The Rate metric is profiled as type gauge and uses average.
Many view types provide more configuration options, such as Number of Charts on Page and Maximum Number of Charts.

TIP
To change the visualization of the data, edit the report and change the view type. The summary table view shows only aggregate metrics. If you change to this view type, the view editor does not remove selected nonaggregate metrics. However, the view shows only the aggregate values. For example, you can you configure a Chart with Multiple Metrics to show the CPU Utilization maximum. When you change the view type to the summary table, the view shows the CPU Utilization average.

Chart with Multiple Metrics

This view consists of one chart that displays a trend line for each selected metric. The trend line is an aggregate for all the selected groups or items.

Figure 43: ChartWithMultMet

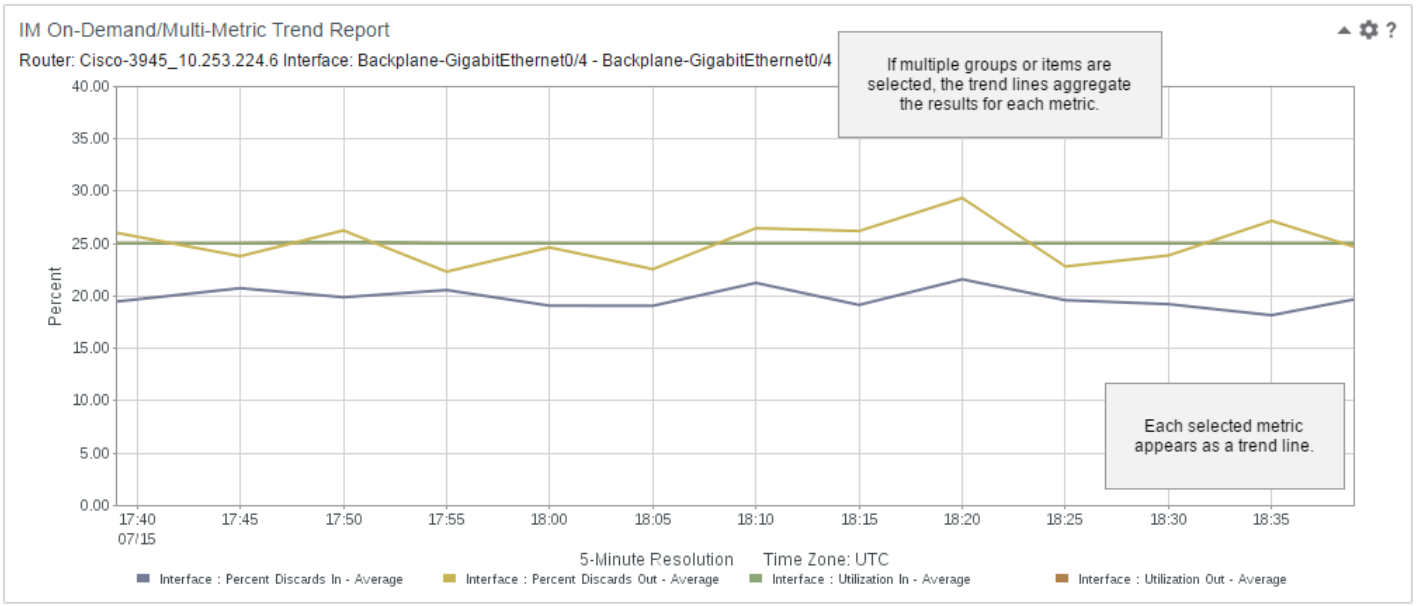


Chart Per Metric

This view consists of one chart for each selected metric. Each chart displays a trend line for the metric. The trend line is an aggregate for all the selected groups or items.

Figure 44: ChartPerMet

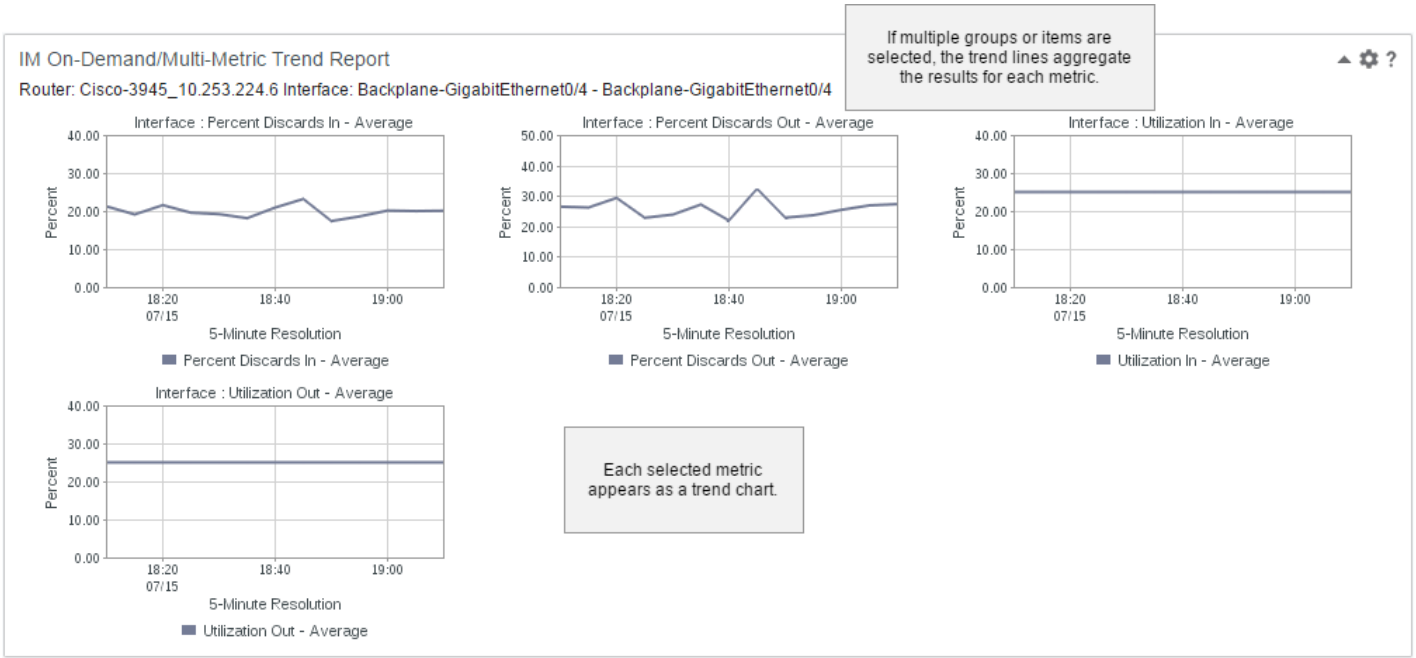


Chart Per Item with Multiple Metrics

This view consists of one chart for each selected item or group. Each chart displays trend lines for each selected metric.

Figure 45: ChartPerItemWithMultMet

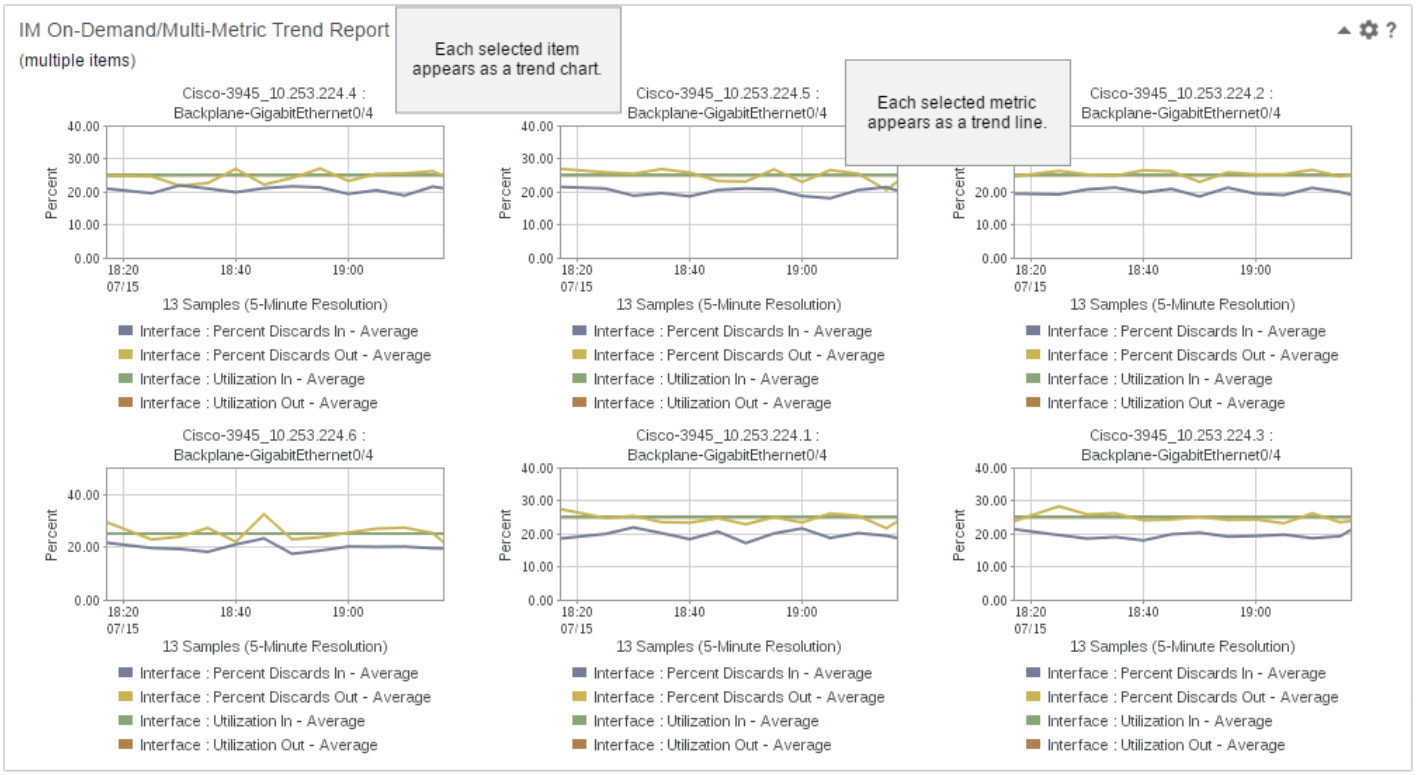


Chart Per Metric with Multiple Items

This view consists of one chart for each selected metric. Each chart displays trend lines for every selected item or group.

Figure 46: ChartPerMetWithMultItems

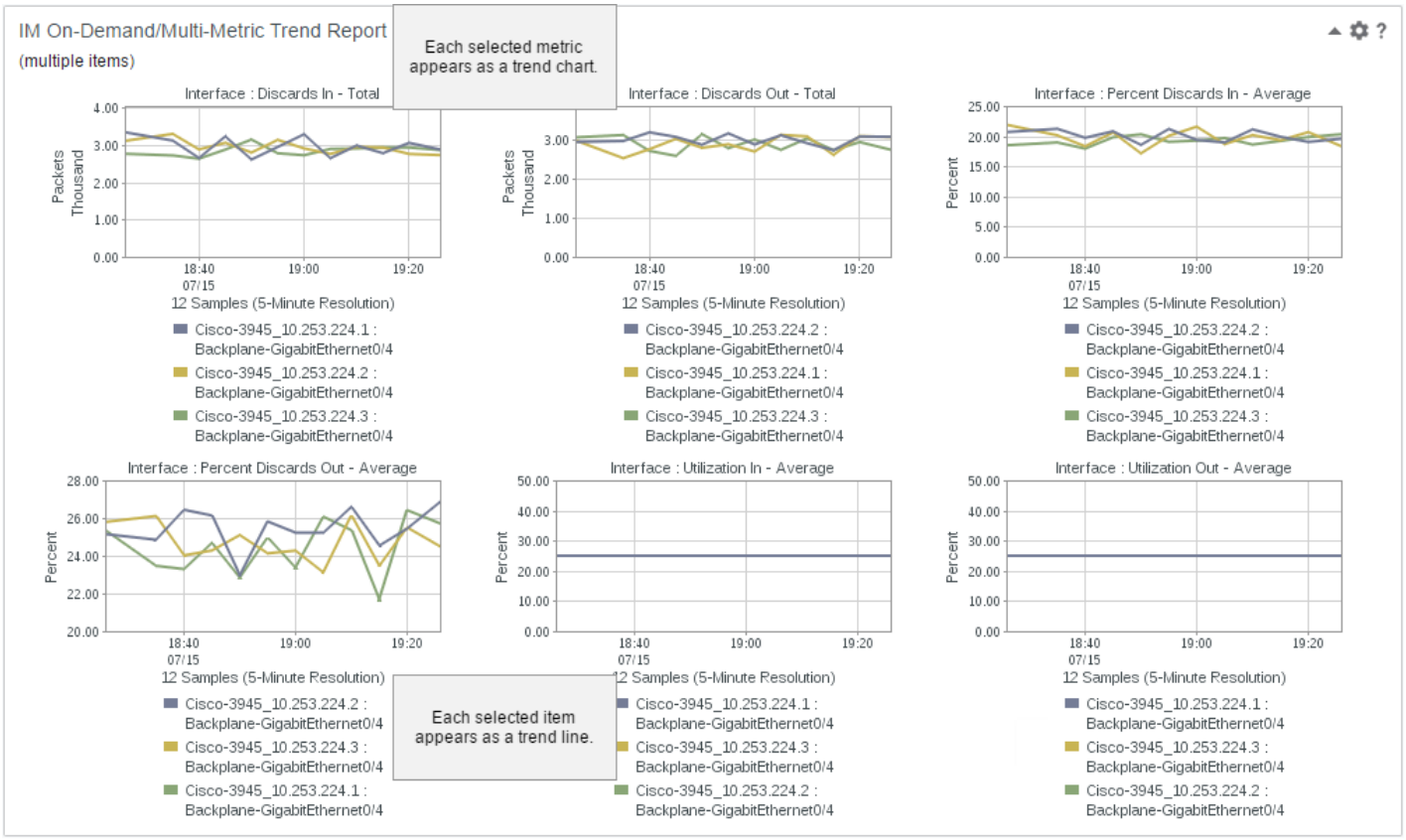


Chart per Metric by Single Item

This view consists of chart groupings for each selected item or group. Each chart grouping consists of one chart for each selected metric.

Figure 47: ChartperMetricbySingleItemReport

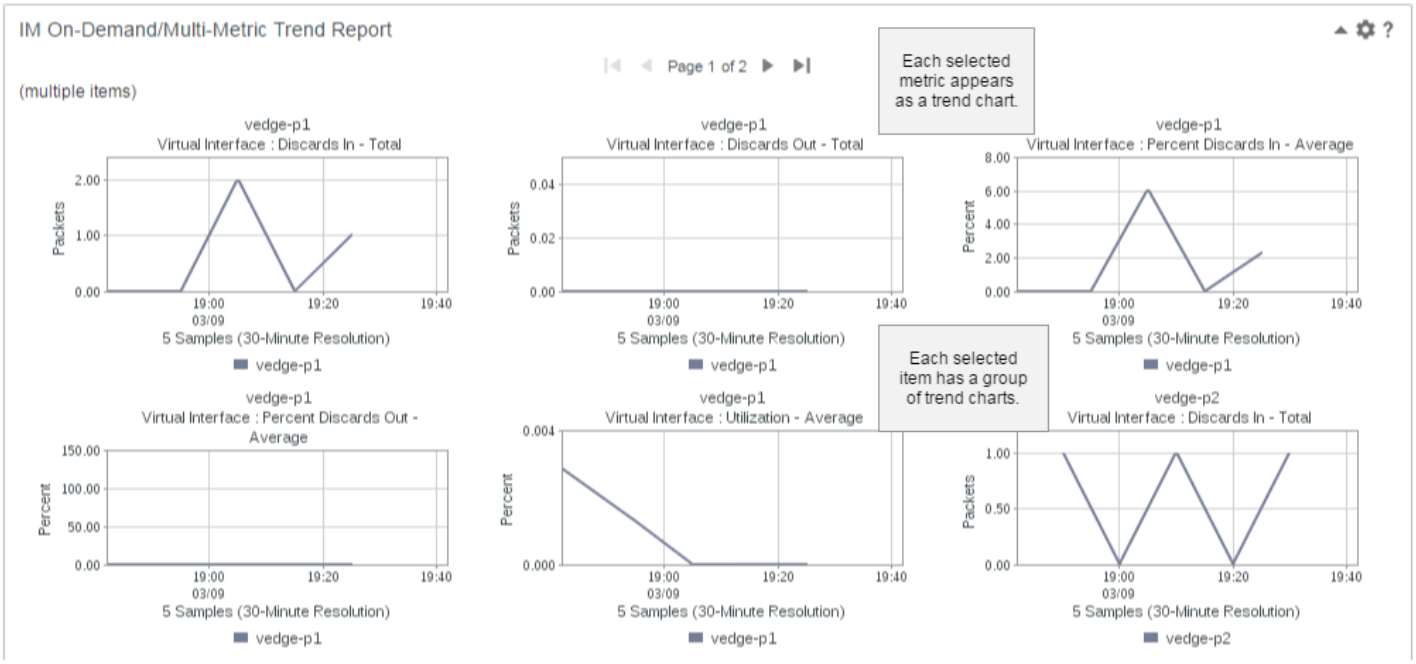


Table with Multiple Metrics

This view consists of one table that displays a list of the selected metrics. The table shows aggregate values for all the selected groups or items.

Figure 48: TableWithMultMet

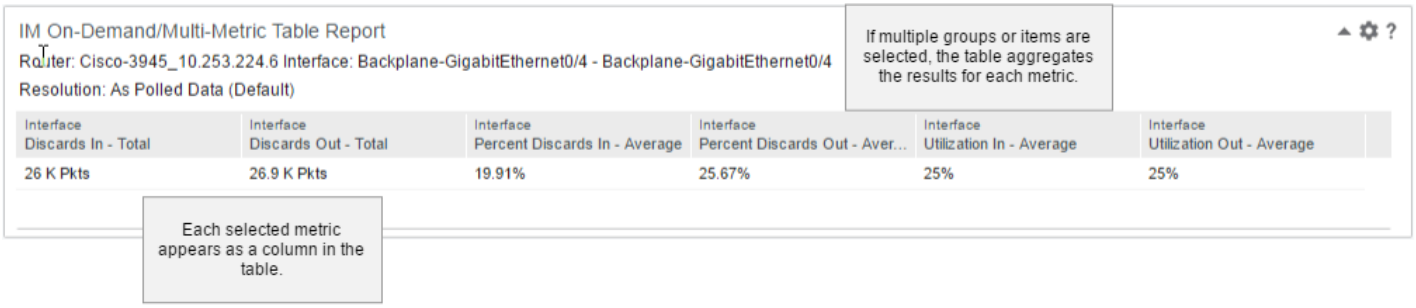
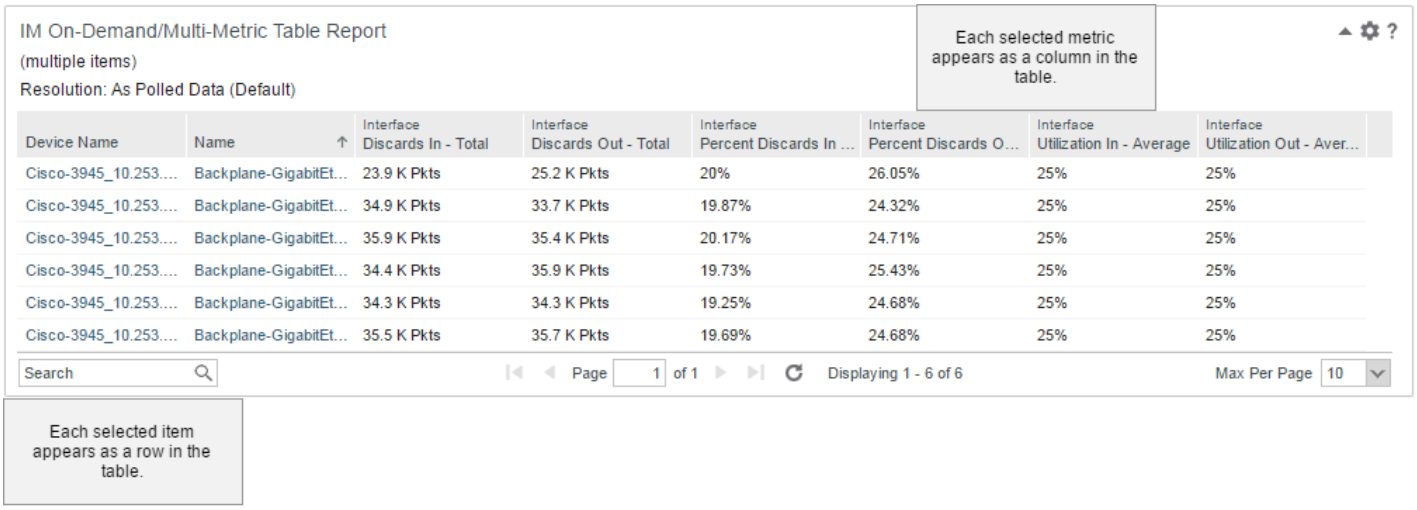


Table Per Item with Multiple Metrics

This view consists of one table that displays a list of the selected metrics. Each row in this table represents a single item.

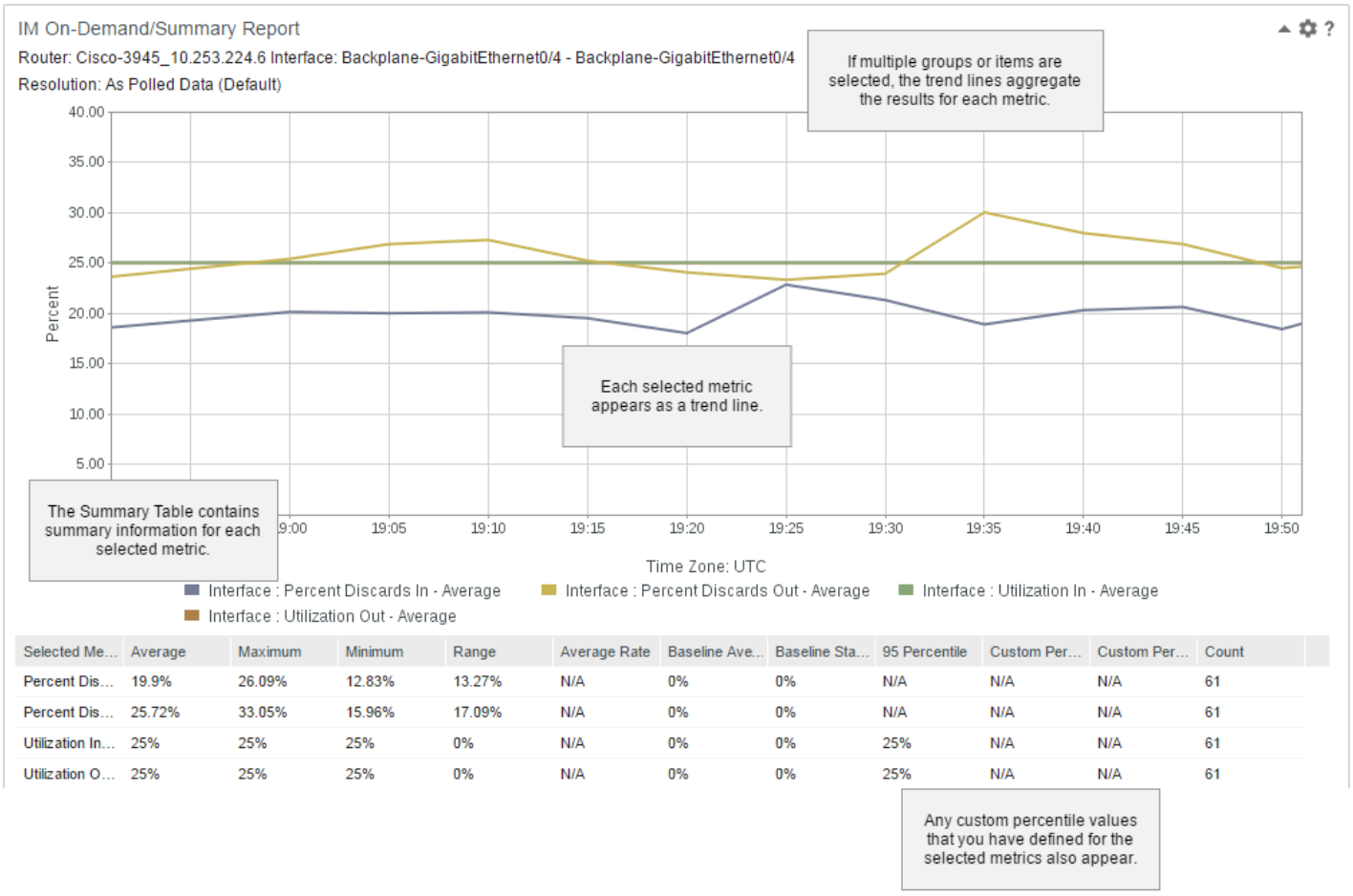
Figure 49: TablePerItemWithMultMet



Summary Table/Chart by Metrics

This view consists of one trend graph that represents one or more items and multiple metrics. The trend graph aggregates all items for each selected metric. The summary table below the graph is sorted by metric. Any custom percentile values that you have defined for the selected metrics also appear in the summary table.

Figure 50: SummaryTableChartByMet

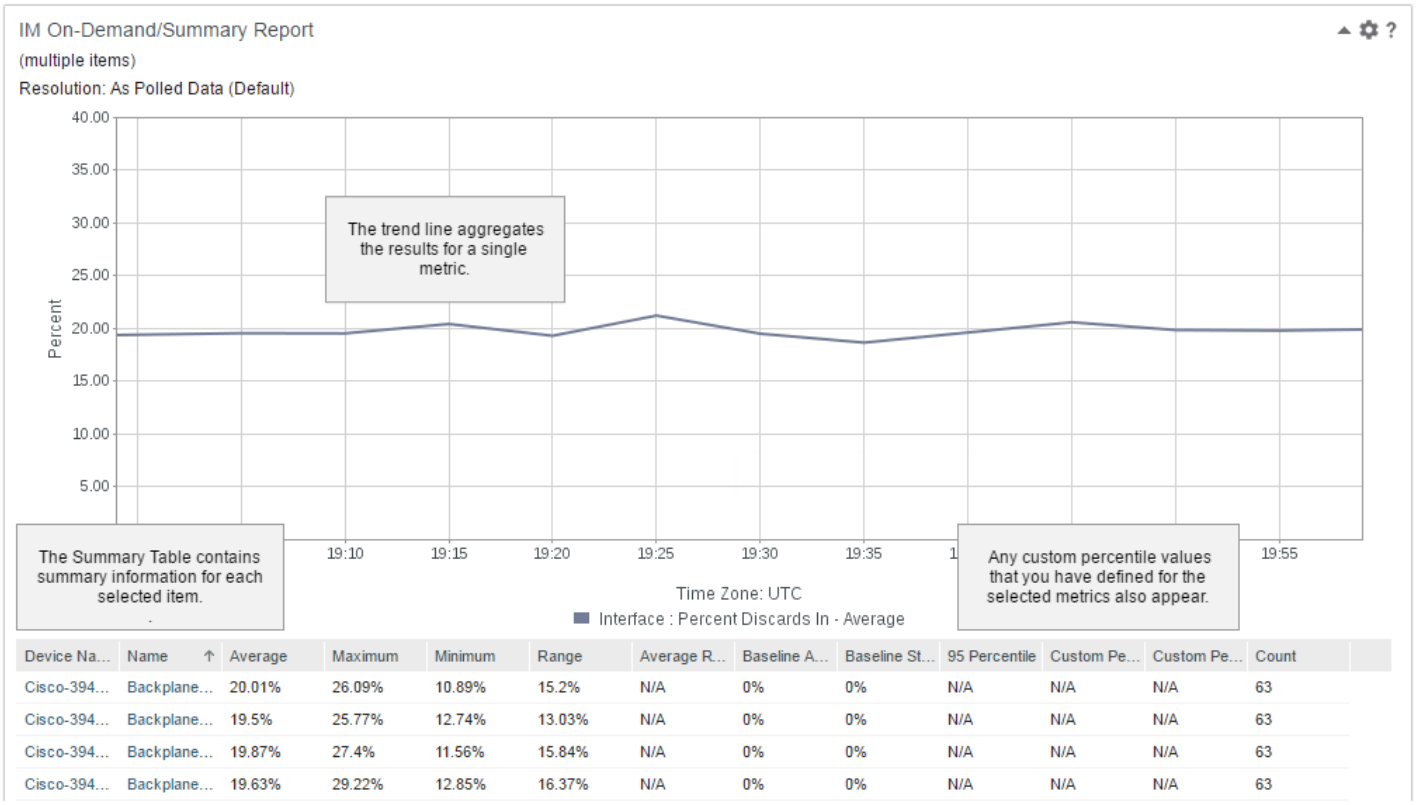


Selected Me...	Average	Maximum	Minimum	Range	Average Rate	Baseline Ave...	Baseline Sta...	95 Percentile	Custom Per...	Custom Per...	Count
Percent Dis...	19.9%	26.09%	12.83%	13.27%	N/A	0%	0%	N/A	N/A	N/A	61
Percent Dis...	25.72%	33.05%	15.96%	17.09%	N/A	0%	0%	N/A	N/A	N/A	61
Utilization In...	25%	25%	25%	0%	N/A	0%	0%	25%	N/A	N/A	61
Utilization O...	25%	25%	25%	0%	N/A	0%	0%	25%	N/A	N/A	61

Summary Table/Chart by Items

This view consists of one trend graph that represents multiple items and a single metric. The trend graph aggregates all items for the metric. The summary table below the graph is sorted by item. Any custom percentile values that you have defined for the selected metrics also appear in the summary table.

Figure 51: SummaryTableChartByItems



View the Report Templates

You can view a list of reusable report definitions on the **Manage On-Demand Report Templates** page. Users see only the report templates that are associated with the current tenant. The global administrator sees only reports that are associated with the Default Tenant.

Prerequisite: You have the Administer Reports or Run Reports role right.

Follow these steps:

1. Log in as a user with either the Administer Reports or Run Reports role right.
2. Select **Reports**, and then click **On-Demand Report** Templates.
   The **Manage On-Demand Report Templates** page opens, and the current list of reports appears.
3. (Optional) To filter the list to show only your reports, click **My Reports**.
   Use this page to view, add, edit, delete, or generate the reports in your tenant.

Generate an On-Demand Report

Generate an on-demand report to view a static data set from a narrow context. Use on-demand reports to investigate and troubleshoot issues.

Select managed items and metric families that complement each other. Changing the item context can clear the original selections. For example, if you select three routers for reporting, and then add interfaces, the routers are cleared. The report reflects interface data, but no rollups to the router level.

Different device types and components are compatible for reporting in a single on-demand report. For example, routers and servers can be included in the same report.

Use one of the following methods to generate an on-demand report:

- Select an item on an inventory page, and then click **On-Demand**.
- Go to **Reports**, **On-Demand Report Templates**, and then click **New**.
- Go to **Reports**, **On-Demand Report Templates**, select an existing report template, and then click **Edit**.
- Add the **IM On-Demand/Multi-Metric Trend Report** to a dashboard or context page.

> **TIP**
> On dashboards and in context pages, this report renders best in single-column layouts.

Follow these steps:

1. Specify the **Title**, **Name**, and **Description**.
   The title appears on the view and in the report.
   The name identifies the report in the **On-Demand Report Templates** list and appears as a title for the report.
2. Select the **View Type**.
3. Select a **Resolution** option. For more information, see Customize Views.

   > **NOTE**
   > On-Demand reports that were configured before release 2.6 cannot use the 'As Polled Data' resolution option. To enable this option, delete the template, and recreate the report.
4. To add baseline metrics to the metric value drop-down list or item selector, select the **Baseline Metrics** checkbox. Baseline data helps characterize past performance for the selected monitored parameters, assess present performance, and estimate future performance.
5. If you are adding the IM On-Demand/Multi-Metric Trend Report to a dashboard, set the **Context** to one of the following:
   - **Dynamic**
     A Dynamic context indicates that the context of the view changes with the context of the dashboard page.
   - **Fixed**
     A Fixed context indicates that the view uses a specified group, device, or component as a context for the data.
6. If you select **Dynamic** as the context, skip this step. Otherwise, select whether to **Add Items** or **Add Groups**:
   - If you selected **Add Items**, click **Add / Remove Items**, select a **Context Type**, and add up to 15 items.
   - If you selected **Add Groups**, click **Add / Remove Groups**, and add up to 15 groups.

     > **NOTE**
     > Only the groups and items that are included in your permission groups are displayed.

     > **WARNING**
     > You cannot include any groups that are in the Collections category in reports.

   The **Metrics to Include** list is populated with the available metric families and metrics for the selected items or groups.
7. To determine the level of aggregation for metrics, select the **Metric Calculate Level** to one of the following:
   - **by Group**
   - **by Device**
   - **by Component**

   This option is available only when you select one of the following view types:
   - Chart per Item with Multiple Metrics
   - Chart per Metric with Multiple Items
   - Table per Items with Multiple Metrics
   - Summary Table/Chart by Items
8. In the **Metrics to Include** list, expand from the list of available metric families, and select up to 15 metrics to display in the report. You can select metrics from multiple metric families. You can use the arrows to specify an order for the selected metrics and the order is saved.

NOTE
For SD-WAN metric families (Tunnels and Application Paths), percentile and projection metrics are available, but unsupported and the views render incorrectly.

9. If the view supports business hours, select whether to **Compress Non Business Hours**.
10. (Optional) To preview the report, click **Run**.
    A preview dashboard shows the view format that you have selected.

    NOTE
    If the metric families that you selected do not apply to the selected components, N/A appears in the report.
11. From the preview, click **Save Template**.
12. Click **Save**.
    The report template is added to the **Manage On-Demand Report Templates** page. Run, email, or print the report as required.

Share On-Demand Report Results

After you generate the on-demand report, select an output format to share the results. Role rights to print reports and send them by email are required.

Performance Metrics

Data sources collect, process, and aggregate performance data that is shown in NetOps Portal dashboards. However, it is helpful to understand how NetOps Portal handles data and rolls up values for display in views. In some cases, views in NetOps Portal are rollups of more granular data that is available in the data source interfaces, or by drilling down into details from the main dashboard pages.

We also offer some guidance for interpreting the metrics you see in NetOps Portal dashboards and context pages.

Baseline Calculations

Some views include baseline data as a basis for comparison. The baseline calculation method varies by the registered data source. The baseline data that is plotted in many views shows statistical deviations from "normal" performance for a given statistic. Metrics are considered to be "normal" based on the calculated baseline average. The Standard Deviation is used to gauge the statistical validity of the baseline values. Baseline values are included in charts to help you see places where performance values are changing rapidly.

Baseline data helps to characterize past performance for selected monitored parameters, assess present performance, and estimate future performance. For example, comparing current CPU utilization to a known baseline average level helps to determine whether current utilization is within a typical range. A monitored parameter that exceeds a baseline can indicate additional load on the server from a new application process, an increase in the number of users or sessions, or an increase in the amount of data being processed.

Baseline Averages

Depending on the amount of polled data that is collected, *baseline averages* are calculated in two ways:

- Initially, averages are calculated for the same hour regardless of the day.
- After enough data is collected, averages are calculated for the same day of the week and the same hour.

Baseline averages help to characterize past performance for selected monitored metrics, and helps to assess present performance. Baseline averages and related standard deviations are continually calculated as each hour passes. The standard deviation provides a statistical indicator of how much variability exists in the population data that factored into the baseline average calculations.

In Data Aggregator, "normal" for a specified duration within a window of time is based on the calculated baseline average.

Baseline Average Calculations

When a limited amount of data is first collected, the baseline average is calculated for the same hour for every preceding day of the week. For example, after two days worth of history, a baseline average value for the 9:00 AM to 10:00 AM time period is calculated by averaging the hourly rollups for the same time periods for two consecutive days.

Eventually, when more data is available, a switchover in the calculation method occurs automatically and Data Aggregator establishes "normal" by averaging hourly samples across available preceding same days of the week. This method, then, considers the day of the week patterns in utilization. This method produces a better approximation of what is "normal", which can lead to a reduction in the number of missed violations and false positive events that are generated. In the same example as above, after three weeks of history, a baseline average is calculated by averaging the 9:00 AM to 10:00 AM hourly rollups for the three Mondays within the three-week period.

NOTE

By default, this automatic switchover occurs when at least three same day of the week, same hour data samples are available for the past 12 weeks. Data Aggregator switches back to the every day, same hour calculation method automatically when the required number of data points is no longer available. These default settings are configurable. For more information, see Configure Data Retention Rates.

Baseline averages are calculated for event and report generation purposes.

Standard Deviation Calculations

The standard deviation is calculated from the baseline average for rollups, threshold events, and report generation purposes.

Rollups:

- For hourly rollups, the standard deviation is calculated for the polled values.
- For daily rollups, the standard deviation is calculated for hourly averages.
- For weekly rollups and beyond, the standard deviation is calculated for the daily averages.

Threshold Events:

- The standard deviation provides a statistical indicator of how much variability exists in the population data that factored into the baseline average calculations.

Reporting:

- For hourly reporting, the standard deviation is calculated for the polled values.
- For daily reporting, the standard deviation is calculated for hourly averages.
- For weekly reporting and beyond, the standard deviation is calculated for the daily averages.

The formula for calculating this standard deviation is:

```
population deviation = Square root of (Sum ( X - population mean)/number of data
points)
```

- **X**
  The data point value in the population
- **Population**
  The set of potential values that includes observed cases and potentially observable cases

Example: Calculate the Same Hour Average and Population Standard Deviation for CPU Utilization

The following example shows how the "same hour" average (mean) and population standard deviation are calculated for CPU utilization on a specific device, when there are three points of data for 2:00 AM on Monday, Tuesday, and Wednesday.

1. Collect three points of data.

```
Day:                                Monday      Tuesday     Wednesday
Mean (Average) CPU utilization:     76          65          10
```

2. Calculate the population mean.
   The formula for calculating the population mean is as follows:

```
The population mean = sum of data point values in population/number of data points.
```

   The equation for this example is as follows:

```
(76+65+10)/3
The population mean= 50.33
```

3. Calculate the difference of each data point from the mean.
   The differences for this example are:

```
25.67    14.67    -40.33
```

4. Calculate the square of the difference for each data point.
   The squares for this example are:

```
658.78     215.11    1,626.778
```

5. Calculate the sum of the squares:
   The sum of the squares for this example is 2,500.67.
6. Calculate the sum of the squares, divided by the number of data points in the population.
   The result for this example is 833.56.
7. Calculate the square root of the sum of squares of data point value from the population mean.
   The square root for this example is 28.87.
   The standard deviation for this example is 28.87.

The following table depicts the hourly averages (mean) of rate data by day, the average (mean) of hourly averages, and the population standard deviation of the hourly averages for the same hour:

Time	Monday	Tuesday	Wednesday	...	Mean	Standard Deviation
2:00 AM	76	65	10	...	50.33	28.87
3:00 AM	87	18	32	...	45.67	29.78
4:00 AM	10	56	40	...	35.33	19.07
5:00 AM	60	45	19	...	41.33	16.94
Hour...	...	...	...	...	...	...

Example: Calculate the Same Day of the Week Same Hour Average and Population Standard Deviation for CPU Utilization

The following example shows how the average (mean) and population standard deviation are calculated for CPU utilization on a specific device, when there are three points of data for three Mondays at 2:00 AM.

1. Collect three points of data.
   ```
   Monday of Week:                      1       2       3
   Mean (Averages) CPU utilization:    76       4       6
   ```

2. Calculate the population mean.
   The formula for calculating the population mean is as follows:
   ```
   The population mean = sum of data point values in population/number of data points.
   ```

   The equation for this example is as follows:
   ```
   (76+4+6)/3
   The population mean = 28.67.
   ```

3. Calculate the difference of each data point from the mean.
   The differences for this example are:
   ```
   47.33    -24.67    -22.67
   ```

4. Calculate the square of the difference for each data point.
   The squares for this example are:
   ```
   2,240.44    608.44    513.78
   ```

5. Calculate the sum of the squares.
   The sum of the squares for this example is 3,362.67.
6. Calculate the sum of the squares, divided by the number of data points in the population.
   The result for this example is 1,120.89.
7. Calculate the square root of the sum of squares of the data point value from the population mean.
   The square root for this example is 33.48.
   The standard deviation for this example is 33.48.

The following table depicts the hourly averages (mean) of rate data by day, the average (mean) of hourly averages and the population standard deviation of the hourly averages for the same day of the week, same hour:

Time	Week 1		Week 2		Week 3	Monday			
	Monday	...	Monday	...	Monday	...	Mean	Standard Deviation	
2:00 AM	76	...	4	...	6	...	28.67	33.48	
3:00 AM	87	...	71	...	56	...	71.33	12.66	
4:00 AM	10	...	27	...	58	...	31.67	19.87	
5:00 AM	60	...	3	...	32	...	31.67	23.27	
Hour	...	...	...	...	...	...	...	...	

Example: Deviation from Normal using the Same Day of the Week Same Hour Average and Population Standard Deviation for CPU Utilization

Assume that Data Aggregator is polling CPU utilization data at a 5-minute interval. You define an event rule to generate an event when CPU utilization is greater than one standard deviation above the mean for a single 5-minute poll interval.

In this example, event rule duration and window are both set to 5 minutes.

The formula for calculating when an event is raised is as follows:

```
CPU utilization = mean value + 1(standard deviation value)
```

Therefore, substituting mean and standard deviation values from the preceding same day of the week, same hour for Monday at 2:00 AM is as follows:

```
CPU utilization = 28.67 + 1 (33.48)
CPU utilization = 62.15
```

As a result, if CPU utilization were to exceed 62.15 for a single 5-minute poll interval between 1:05 AM and 2:00 AM on Monday, an event would be raised. This event indicates that the CPU utilization deviated from normal for that timeframe.

Example: Examine CPU Utilization Events in a Trend Chart View

Assume that Data Aggregator is polling CPU utilization data at a 5-minute interval. In this example, you want to be alerted whenever CPU utilization on one of your business critical servers drops below the expected level. You define an event rule to generate an event when CPU utilization is one standard deviation below the mean for a single 5-minute poll interval.

For illustrative purposes only, assume that CPU utilization is 50 percent from Monday, 12:00 AM to Sunday, 12:00 AM. From Sunday, 12:00 AM to Monday, 12:00 AM, CPU utilization drops to 10 percent. You expect this drop in utilization. However, when Data Aggregator begins to calculate the baseline average, an event is raised when the CPU utilization drops to 10 percent. The event clears when the CPU utilization goes back up to 50 percent. The erroneous event is raised because, initially, when a limited amount of data is collected, the baseline average is calculated for the same hour for every day, not taking into account the difference in utilization across days of the week. Data Aggregator is expecting the CPU utilization to be 50 percent *always*.

After three weeks pass, three same days of the week, same hour data samples are available, and the baseline average calculation method changes. Data Aggregator establishes "normal" by averaging hourly samples across same days of the week. Data Aggregator is now expecting the CPU utilization to be 10 percent every Sunday at 12:00 AM to Monday at 12:00 AM. The erroneous event that was raised previously every Sunday at 12:00 AM is no longer raised.

Rate Metrics

The Dynamic and Custom view types let you select metric families and metrics to display for the selected managed item or group. For some metrics, two additional options are available: Average Rate and Total Rate. These Rate variants represent a “per second” value for the metric, achieved by dividing the raw sample value by the number of seconds in the sample’s poll period.

The “Average Rate” and “Total Rate” variants are aggregated differently across managed items when a group is selected for reporting. They are also aggregated differently when a selected device has child items that are monitored, or when the selected time period requires aggregation. For example, in a table, a single value can be shown that represents the aggregated result of one hour of samples: 12 samples from five-minute polling are aggregated into a single value. In this case, using “Average Rate” would average out the “rate” values for the 12 samples, where “Total Rate” would simply sum them.

The following calculations illustrate the difference:

Average Rate (AvgRate) = ((sample1/duration1) + (sample2/duration2) + (sample3/duration3)) / numberOfSamples

Total Rate (TotalRate) = ((sample1/duration1) + (sample2/duration2) + (sample3/duration3))

For most situations, Average Rate is the preferable metric to use.

Examples

Particularly for the Composite Trend view type, selecting the more useful rate metric option can yield more revealing results. Average Rate is adequate for most situations. But if you select Total Rate for a view that reports on groups, you can see the total for all devices of a similar type, or for all devices in the same region, for example.

The Total Rate metric lets you see metrics for a device across all of its interfaces. For a device with child interfaces, you can see an average across all of its interfaces, or you can see totals per interface. Select trend chart settings that display data from each interface as a separate line.

In cases where you are reporting on subinterfaces, consider that each one has a rate of N bytes/sec. The rate per interface is an aggregation of rate data from all subinterfaces. Typically, you would select Total Rate to see a metric such as Bytes/second per interface. To see data from all subinterfaces, use Total Rate to see a Bytes/second total for all interfaces.

If you do an aggregation of the 12 samples from an hour time period, those samples would be averaged to get the Average Rate. They would be added together to get the Total Rate.

Interface Reporting

Digital network interfaces use serial transmission to send data from the transmission port to the receive port on the other end of the communications channel or circuit. Some transmission channels, such as copper Gigabit Ethernet, aggregate serial data across multiple channels to establish their overall circuit capacity. For copper Gigabit Ethernet, 4 channels of 250 Mbps are used to establish the 1-Gbps Ethernet circuit.

Most digital interfaces are *full-duplex.* The term means that they can transmit outbound data at the same moment that they are receiving inbound data. Because data transmission and data reception are independent interface tasks, they are reported separately.

Interface Utilization represents the average amount of data that is transmitted by the interface in a single direction (In or Out), divided by the interface bandwidth, or capacity. Interface utilization can be expressed as a percentage or as a transmission rate in bits per-second (bps).

Interface utilization rates can contribute to network performance issues. For a given interface, monitor whether it is transmitting frames at or below the rate at which it is receiving them. Acceptable interface utilization rates also depend on various SLAs and failover scenarios within your network. For example, two interfaces use a load-sharing algorithm to balance outbound traffic to the next hop. The average interface utilization must remain low enough that a failure of one link does not saturate the remaining available link, which now transmits all data.

Interface Utilization

Interface utilization refers to the transmission and reception of data and associated framing of device interfaces. Interface utilization is commonly referred to as "network utilization," "circuit utilization," or "uplink utilization."

The interface utilization percentage metric is calculated from average data because an instantaneous reading of individual interface utilization is either 100% (actively transmitting or receiving a frame) or 0% (not actively transmitting/receiving a frame). The average utilization percentage value includes the amount of time that the interface was in use over the given interval.

The interface utilization rate metric takes into account the interface speed, or its available bandwidth. For a physical interface, the available bandwidth of an interface is defined as the actual clockspeed rate at which the interface is capable

of transmitting data. For example, 1536 Kbps, 44.728 Mbps, 100 Mbps, 1 Gbps, and 10 Gbps describe clockspeed rates. For logical interfaces, such as subinterfaces, the available bandwidth is defined as the bandwidth value assigned to the interface by a network administrator. However, the total amount of real bandwidth available to the logical interfaces cannot exceed the physical interface capacity in terms of actual transmission rates.

Full-duplex interfaces have the capability to transmit data independently at the same time that they can receive data. This capability requires independent hardware dedicated in the transmit direction and the receive direction of the interface. Accordingly, the average utilization of an interface is reported separately in either the inbound or outbound direction. For example, separate views show "Average Utilization Out" and "Average Utilization In".

The network utilization can be derived from interface utilization values averaged over time from interfaces that are in use or not in use.

Interface Errors and Discards

Elevated interface error rates usually indicate a problem with the transmission medium. For example, the cable, fiber, or interface hardware can cause errors. Each error indicates that the associated packet was dropped during the attempt to transmit or receive it.

When detected in the inbound direction, errors typically indicate problems with the transmission medium (for example, cable or fiber). Outbound errors indicate problems with the interface hardware. The acceptable rate of errors for any given interface is typically zero (0) errors.

Interface discards typically occur when interface buffers no longer have the capacity to store packets (because, for example, buffer memory is exhausted). Buffer congestion often indicates that the rate at which packets are arriving at the interface exceeds its transmission rate.

Each reported discard is a packet that the reporting interface threw out. The sending host must retransmit such packets if a reliable protocol such as TCP is used to send the data end-to-end. Interface discards are typically the result of congested queues serving the interface. Discards frequently occur in bursts. Elevated discard rates may be the result of either microcongestion or chronic congestion issues.

Acceptable discard rates depend on the applications being served, the transmission protocols, and the SLAs established within your organization. Views of interface error and discard totals use a "k" to indicate thousands. Units labeled 'kErrors' or 'kDiscards' therefore refer to thousands of packet errors or thousands of packet discards.

Interface Bandwidth

The term "bandwidth" generally refers to the available capacity of an interface to transmit data at a given bit rate. The bandwidth associated with an interface depends on the type of interface.

For a physical interface, the bandwidth is the physical clock rate that the interface uses to transmit data. The clock rate is also typically a function of the type of interface. The following list shows the clock rates for common physical interfaces:

- DS-0: 64 Kbps
- DS-1: 1.536 Mbps
- E-1: 2.048 Mbps
- E-3: 34.368 Mbps
- DS-3: 44.278 Mbps
- OC-3: 155 Mbps
- Fast Ethernet: 100 Mbps
- Gigabit Ethernet: 1000 Mbps
- Ten Gigabit Ethernet: 10 Gbps

CPU Utilization

The CPU utilization metric is based on average CPU usage over the time period selected for the view.

Utilization percent is a term applied to the portion of a time period that a component is doing work, divided by the total amount of time in the time period. The result is multiplied by 100 to obtain a percentage.

For a CPU, the busy time is spent processing program instructions. Here is an example of how to interpret a utilization rate of 70% for a 5-minute time period: "For 70% of the 5 minutes, the CPU was fully utilized."

High rates of CPU utilization can indicate poor application performance. With high CPU utilization, processes must wait in the processor queue for a previous process to complete execution.

Memory Utilization

In addition to utilization, the data sources send information about memory capacity and usage to NetOps Portal. The Memory Utilization metric is an average utilization statistic derived from the percentage of available memory in use at a given moment, averaged over the reporting interval.

High rates of memory utilization may indicate that processes are paging instruction sets into and out of virtual memory. Such paging leads to slower memory read times, and may require the CPU to be interrupted to manage the paging process. The result is decreased performance for the associated application processes.

In addition, a steady increase in memory utilization over time may indicate a 'memory leak' condition. Such a condition exists where memory is allocated by processes as they start, but is not being released as the processes end. Memory leaks degrade device performance over time. Typically, the device becomes unresponsive when memory is no longer available.

Device Availability and Reachability

Availability measures system uptime and reachability measures device connectivity to DX NetOps Performance Management.

Availability is a percentage of time that the device is powered on and capable of processing data. DX NetOps Performance Management uses system uptime to calculate the percentage of time within a poll cycle that the device was available. The metric value is recorded as up (100), or down (0), or a percentage of the poll cycle.

If DX NetOps Performance Management is unable to reach a device for a particular timeframe, availability is backfilled when the device sends a poll response. DX NetOps Performance Management uses the system uptime to calculate which poll cycles to mark as available. If the device was restarted, or if a counter rollover for uptime occurred, DX NetOps Performance Management backfills only from the 0 to the current system uptime value. Poll cycles before the rollover or restart, where no poll responses were received, remain blank. Hourly and daily rollups reflect the backfill data only if the data is received before the rollup occurs. Availability is the only metric that DX NetOps Performance Management backfills.

A device that is available might be unreachable because of a network or communications failure by another device.

Reachability refers to whether a device is reachable from the data source. Typically, data sources use ICMP (ping testing) to communicate regularly with the target device. Any communication failures, including the loss of the network path or routing, affect the reachability statistics. If ICMP is blocked, DX NetOps Performance Management uses SNMP to determine reachability.

Reachability data comes from regular ping testing of all devices that support ICMP. The reachability value is the percentage ping responses that are received from the device during each reporting interval.

Reachability Status and Contact Status

Reachability status and contact status indicate whether DX NetOps Performance Management can communicate with a device. These states appear on various pages, such as the device details page and context pages.

Reachability Status

Some context pages, such as routers and servers, include the reachability status of the device. The reachability status indicates whether DX NetOps Performance Management can reach the device during the selected time range of the context page. The value is based on the polled reachability metric for the device at the last poll cycle of the time range.

- **Reachable** indicates that the device reachable during the last poll cycle of the time range.
- **Unreachable** indicates that the device was unreachable during the last poll cycle of the time range.
- **Unknown** indicates that the device was not polled during the last poll cycle of the time range. This status may indicate that the device is retired or offline for maintenance.

Contact Status

The Details tab on the Monitored Device page shows the device Status. Status indicates the live status of a pingable or manageable device and is based on the outcome of active polling.

- **Up** indicates that the device is being successfully polled or pinged.
- **Down** indicates that at least two polls failed to receive a response. The managed item cannot be reached by SNMP or ping.
- **Lost Connection to Data Collector** indicates that the Data Collection that is associated with the device is no longer network reachable or was shut down.
- **Not Monitored** indicates that polling has been administratively disabled, for example by using the Stop Polling button.
- **Management Lost** indicates that the device is still pingable, but SNMP is no longer responding to poll requests. This status often indicates incorrect details in a recently changed SNMP profile.
- **Unknown** indicates an unexpected condition. The details are written to the Data Aggregator or Data Collector logs.

Scorecard Projections

Scorecard projections use a customizable set of data points to predict future values for metrics. Projected values are calculated when the view is rendered, and are based on the historical time frame of the view. You can add the projected values to IM Custom View Group Scorecards.

> **TIP**
>
> Do not use scorecard projections for error metrics. Each error is a discrete event that is not affected by historical errors.

The scorecard view includes two methods to calculate projections:

- **Approximation**
  The Approximation method uses the average from each time frame in the view to calculate the projection values. DX NetOps Performance Management calculates a least squares regression on the averages, then uses the line equation to project future values. This calculation method is faster than the Detailed Data method.
- **Detailed Data**
  The Detailed Data method uses the polled data for the entire time frame of the view. DX NetOps Performance Management calculates a least squares regression for the entire set of data points. This calculation is more statistically accurate than the Approximation method, and provides extra columns in the view.

NOTE

Detailed data scorecard projections are supported only for gauge metrics (for example, Bits Out - Average Rate). Detailed data scorecard projections are *not* supported for counter metrics (for example, Bits Out - Total). Projection values are calculated on the As Polled (rate) data to ensure precision.

These columns are hidden by default:

- **Slope**
  Indicates the slope of the line equation.
- **Intercept**
  Indicates the intercept of the line equation.
- **Degrees**
  Degrees of freedom, which indicates the sample size.
- **Linear Fit**
  Indicates the confidence level of the projected values as related to the sample data.
- **Days to Threshold**
  Indicates the projected number of days before the specified critical threshold is reached.

Percentiles

A percentile is the value of a variable below which a certain percent of observations fall. For example, the 95th percentile is the value (or score) below which 95 percent of the observations are found. Percentiles are calculated for rollups, dashboards, and report generation purposes.

Percentile monitoring is useful in measuring throughput data. This statistic more accurately reflects the required capacity of the monitored link for applications that are bandwidth sensitive. The 95th percentile says that 95 percent of the time, the bandwidth usage is below this amount. The remaining 5 percent of the time, the bandwidth usage is above that amount.

Metric values are aggregated during rollups.

By default, DX NetOps Performance Management calculates the 95th percentile for some metrics. Metric families can include up to two more configurable percentile calculations.

Metric Property	Default	Values
Percentile	Disabled or 95	0 or 95
Percentile2	Disabled	0-99
Percentile3	Disabled	0-99

A value of 0 disables the percentile calculation.

Changes to these properties generate Administration events. After you update the metric family, the percentile data is available for reporting within several poll cycles. For trend views, changes cause a gap in the trend line. For table views, changes affect the value of the percentile for time ranges when the change occurred. For example, in a daily time range, the value is inaccurate for one day.

TIP

When you disable a percentile, modify views that include the percentile to avoid an error in the view.

WARNING

Calculate only percentiles that you need. Each percentile that you calculate might significantly affect system performance.

Three percentile values appear in the Metric Families table under the Monitoring Configuration menu for a Data Aggregator. A dash in the Percentiles column indicates that all three percentile values are equal to zero.

Percentile Calculations

Percentile calculations use the Microsoft Excel method.

Hourly and daily rollup uses the polled rate data. Weekly rollup performs the percentile calculation on the results of the daily rollups.

Percentiles are calculated in the database and exposed to the reports and dashboards.

Metric Projection

To calculate future values based on historical metric data, use metric projection. Metric projection is useful for capacity planning. For example, to verify that the interface bandwidth is sufficient for a specific time in the future, calculate the projected interface utilization.

To see future trends, metric projection supports up to three configurable intervals. For example, you can project to 20, 60, and 180 days in the future for the metric. Projection shows an overall trend. Typically, the longer the projection interval, the less accurate the exact value.

Scorecard Projections

- Scorecard views provide line-of-business owners a group-level summary of how key metrics perform over time. Performance is based on a set of user-defined thresholds.
- The scorecard view displays historical time intervals, and up to three projected values.
- Projected values are calculated when the view is rendered, and are based on the historical time frame of the view.

Metric Projections

- Metric projection is designed for network and capacity planners that want the system to calculate and store projections.
- Projections are configured for individual metrics. Up to three projection intervals can be specified per metric.
- Projected values are based on up to 90 days of historical data. Once configured, projected metric values can be included in custom table views.

For more information, see Scorecard Projections.

You can add the projection values to custom table views.

> **TIP**
>
> Table views are useful for capacity planning.

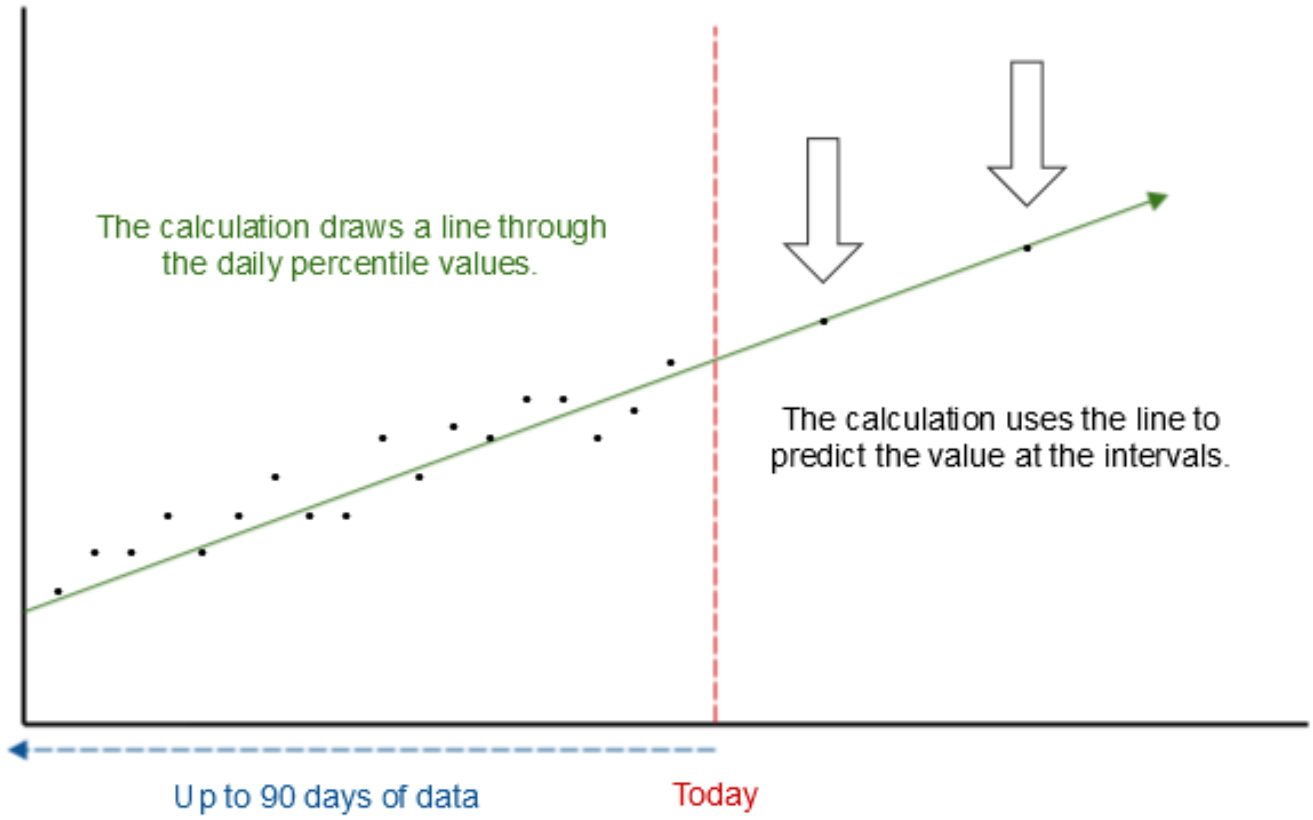
Metric Projection Calculations

Projection values are calculated for a specified interval and are based on a configurable daily percentile calculation on the metric. For capacity planning, 95 percent is a typical percentile value. The system uses the following process to calculate the projection:

1. Calculates and stores the daily percentile value using the Microsoft Excel method from the as polled data.
2. Calculates a linear regression line from the daily percentile values. The calculation uses a simple linear regression (least squares regression).
   The calculation uses all the available daily percentile values from the last 90 days as input data. Projection requires at least two days of daily percentile values. The accuracy of the projection typically increases with the available data points.
3. Calculates the future value for the interval from the linear equation.

The following diagram illustrates the calculation method:

Figure 52: Metric Projection Calculation



Configure Metric Projection

To configure metric projection, edit the metric through the UI. For more information, see Edit a Metric.

TIP

When you disable a projection, modify views that include the projection to avoid an error in the view.

WARNING

Calculate only projections that you need. Each projection that you calculate may significantly affect system performance. For more information, see the DX NetOps Performance Management Sizing Tool.

To configure metric projection using REST web services, execute a PUT operation on the target metric. Configure **ProjectionPercentile** and **ProjectionInterval** for the metric. For more information, see Metric Family XML Structure.

WARNING

Changes to ProjectionPercentile cause inaccurate projections for up to 90 days. When you change the value of ProjectionPercentile, the percentile values for days before the change are not recalculated.

Changes to these properties generate Administrative events.

Example:

This example shows the XML for interface utilization. This projection is calculated based on the 95th percentile. The projection is calculated for 20, 30, and 90 days.

Endpoint:

http://DA_host:8581/typecatalog/metricfamilies/extension/NormalizedPortInfo

```
<DataModel xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" namespace="http://
im.ca.com/normalizer" xsi:noNamespaceSchemaLocation="IMDBCertificationFacet.xsd">
  <Author>CA</Author>
  <Version>1.2</Version>
  <FacetType name="NormalizedPortInfo"
 descriptorClass="com.ca.im.core.datamodel.certs.NormalizedFacetDescriptorImpl">
    <Documentation/>
    <FacetOf namespace="http://im.ca.com/core" name="Item"/>
    <AttributeGroup external="true" list="true" name="PortInfoPollable">
      <Documentation/>
      <Attribute name="Utilization"
 type="double">
        <ProjectionPercentile>95</ProjectionPercentile>
      </Attribute>
    </AttributeGroup>
    <BaselineDefinitions>
      <Baseline name="DailyBaseline">
        <ID>26</ID>
        <PerformanceMetric>Utilization</PerformanceMetric>
        <Period>1 Day</
Period>
        <ProjectionInterval>20</
ProjectionInterval>
        <ProjectionInterval2>30</
ProjectionInterval2>
        <ProjectionInterval3>90</ProjectionInterval3>
        <Window>90 Days</Window>
        <StartDate>0</StartDate>
        <EndDate>0</EndDate>
        <DaysOfWeek>0</DaysOfWeek>
      </Baseline>
    </BaselineDefinitions>
  </FacetType>
</DataModel>
```

Total, Average, Minimum, and Maximum Values

The total, average, minimum, and maximum values are calculated for rollups and for reporting purposes. These values let you observe the upper and lower bounds of performance across a given time interval.

A *rollup* is the process during which metric values are aggregated. In an hourly rollup, the 1 minute, 5 minute, 15 minute, 30 minute, and 60 minute polled values for metrics are aggregated every hour. In a daily rollup, as-polled values

for metrics are aggregated once a day. In a weekly rollup, daily values for metrics are aggregated once a week. A given metric is aggregated by either averaging or summing data points during the rollup (not both) based on the "RollupStrategy" configured for that metric.

Hourly rollups:

- Total: Sum of all the as-polled data points for the hour.
- Average: Sum of all the as-polled data points divided by the number of data points for the hour.
- Minimum: The lowest single poll value of the hour.
- Maximum: The highest single poll value of the hour.

Daily rollups:

- Total: Sum of all the as-polled data points for the day.
- Average: Sum of all the as-polled data points for the day divided by the number of data points.
- Minimum: The lowest single value of the hourly minimums.
- Maximum: The highest single value of the hourly maximums.

Weekly rollups and beyond:

- Total: Sum of all the daily data points for the week.
- Average: Sum of all the daily data points divided by the number of data points for the week.
- Minimum: The lowest single value of the daily minimums for the week.
- Maximum: The highest single value of the daily maximums for the week.

Five-minute resolution reporting:

- Total: Sum of all the as-polled data points for the 5 minute increment.
- Average: Sum of all the as-polled data points divided by the number of data points for the 5-minute increment.
- Minimum: The lowest single poll value of the five-minute increment.
- Maximum: The highest single poll value of the five-minute increment.

One hour resolution reporting:

- Total: Sum of all the as-polled data points or the hourly rollup value for the hour.
- Average: Sum of all the as-polled data points divided by the number of data points or the hourly rollup value for the hour.
- Minimum: The lowest single value of the as-polled data or hourly rollup minimum for the hour.
- Maximum: The highest single value of the as-polled data or hourly rollup maximum for the hour.

Day resolution reporting:

- Total: Sum of all the hourly data points or the daily rollup value for the day.
- Average: Sum of all the hourly data points divided by the number of data points or the daily rollup value for the day.
- Minimum: The lowest single value of the hourly data or daily rollup minimum for the day.
- Maximum: The highest single value of the hourly data or daily rollup maximum for the day.

Events

An event is a message that provides information about what is happening in DX NetOps Performance Management. Events provide information for monitoring the health and status of your system and your environment. All events include basic information, such as related devices and the time of the occurrence that triggered the event.

TIP

You can integrate the Event Manager with CA Spectrum. In CA Spectrum, you can configure the events to generate alarms. For more information, see the CA Spectrum documentation.

View Events

Events View

This view displays all the events that occurred in the selected time range for the dashboard. This view can be filtered for a specific group. This view is the default view in the Events Display dashboard.

Filtered Event Views

This view includes filters for data source, severity, event type, event subtype, and threshold profile.

> **TIP**
>
> To see the complete event properties, select an event in any view, and click **Details**.

Using Events

Some typical reasons to view events include the following scenarios.

Troubleshoot Performance Issues

To troubleshoot performance issues with a specific device, you can filter the events to that device. The Events view filters the complete list of events to display only events for the selected device.

Monitor Thresholds

Thresholds are configured to generate events for specified devices, components, or groups. Use these events to track when the monitored devices do not meet operating requirements.

Track Configuration Changes

When Automatically Update Metric Families is not selected for a custom monitoring profile, view the events log to configuration changes. Reconfiguration events with the Detected subtype indicated detected changes on the monitored devices. Reconfiguration events with the Changed subtype indicate changes to device items.

Audit Changes to the System

Administration events provide useful information about changes to the system configuration, such as the time of a change.

Event Types

The event type determines what information the event contains. Each event has an event type and many events have an event subtype. Events can be categorized as the following types:

> **NOTE**
>
> More event types might appear when DX NetOps Performance Management is integrated with other products.

Administration Events

Administration Events include system status notifications and audit events. These events include the following event subtypes:

- **ActiveMQ Status**
  Occurs when the ActiveMQ status changes.
- **Administration Message**Several administrative tasks that are related to discovery and tenants can trigger this event type.
- **Certification Change**
  Occurs when a vendor certification or metric family is extended or updated.
- **Data Collector Status**

Occurs when the status of a data collector changes.
- **Monitoring Profile**
  Occurs when the monitored metrics selected for a monitoring profile are modified.
- **Percentile Change**
  Occurs when percentile calculations are changed for a metric family. When the percentile is changed in the UI, the event includes the user.
- **Projection Change**
  Occurs when projection calculations are changed for a metric family. When the projection is changed in the UI, the event includes the user.
- **Threshold Monitoring Status**
  Provides information about the threshold monitoring engine. For more information, see Threshold Monitoring and Threshold Limiter Behavior.
- **Vendor Certification**
  Occurs when the vendor certification priority order is modified for a metric family.

Data Collection

Data collection events provide information about contact with monitored devices. These events include the following subtypes:

- **Contact Established**
  Occurs when the data collector established contact with a device.
- **Contact Lost**
  Occurs when the data collector loses contact with a device.
- **Management Established**
  Occurs when the data collector established contact with the management protocol of the device, such as SNMP.
- **Management Lost**
  Occurs when the data collector loses contact with the management protocol of the device, such as SNMP.
- **Polling Disabled**
  Occurs when polling for a device stops.
- **Polling Enabled**
  Occurs when polling for a device starts.

Data Repository State

Data Repository State events provide information about the status of the data repository. All Data Repository State events include information about the data repository host. These events include the following subtypes:

- **Connected**
  Occurs when the data aggregator connects to a data repository node after another node goes down. The event includes information about the connection.
- **Data Repository Failover**
  Occurs when the data repository cluster changes the primary host. The event includes the new host name and the previous host name.
- **Data Repository Failure**
  Occurs when a data repository cluster fails. The event includes the name of the host that failed.
- **Degraded**
  Occurs when the data repository node response time to the data aggregator heartbeat is over 20 seconds.
- **Down**
  Occurs when a data repository node is down.
- **Start Up**

Occurs when a data repository node starts up. The event includes availability information about all the data repository nodes.

Life Cycle

Life Cycle events track changes to the device life cycle. These events include the following subtype:

- **State Change**
  Occurs when the life cycle state of a device changes. The event includes the current state, the previous state, and the user who changed the state.

Override

Override events provide information about manual changes to metric values. These events include the following subtypes:

- **Override Cleared**
  Occurs when a user clears the override for the Speed In and Speed Out values for an Interface.
- **Override Set**
  Occurs when a user manually updates the Speed In and Speed Out values for an Interface.

Polling State

Polling State events track whether polling is disabled or enabled on an interface. The events include the following subtypes:

- **State Change**
  Occurs when the polling state of an interface changes. The event includes the current state, the previous state, and the user who changed the state.

Reconfiguration

Reconfiguration events provide information about change detection. These events include the following subtypes:

- **Changed**
  Occurs when the system applies a change to a component item.
- **Detected**
  Occurs when the system detects a change to a component.
- **Rebooted**
  Occurs when the system detects that a devices has rebooted.

Threshold Violation

Threshold Violation events occur when monitored groups or devices violate the configured criteria in assigned threshold profiles. These events include the following subtypes:

- **Cleared**
  Occurs when a device or group that is in violation of a threshold clears the violation. The event includes detailed information about the violation and the reason the violation was cleared.
- **Raised**
  Occurs when the monitored device or group violates the event rule in an assigned threshold profile. The event includes detailed information about the violation.

For more information about thresholds, see Configure Threshold Profiles and Use Events to Monitor Device Performance.

Change Event Properties

To change the properties of the Event Manager, update the `em.properties` file.

For example, you can modify the `Event.Retention` property to change how long the event manager stores events. By default, the Event Manager database retains events for 30 days.

You can also modify the `em.notification.script_notification_execution_time` property to change the maximum time (in seconds) that a script can take to complete execution before it is killed. For more information about notification scripts, see Configure Notifications.

Follow these steps:

1. Open the Event Manager properties file:
   `/opt/CA/PerformanceCenter/EM/webapps/EventManager/WEB-INF/em.properties`
2. Edit the properties as desired:

```
db.driverClassName=com.mysql.jdbc.Driver
db.timeout=120
db.maxActive=40
rib.queryTimeout=100
em.ws.maxqueue=50

Event.Retention=45
db.url=jdbc\:mysql\://localhost\:3306/em?useUnicode\=true&characterEncoding\=UTF-8
em.notification.trap.eventmanager_trap_use_new_format=false
em.event.analysisThreshold=10000
em.notification.notification_pool_size=10
db.username=netqos
nqevents.dbHost=
em.ws.queuetimeout=180
em.notification.notification_max_pool_size=20

em.notification.script_notification_handler_max_pool_size=1

em.notification.script_notification_handler_pool_size = 1

em.notification.script_notification_handler_queue_size=5000


em.notification.script_notification_execution_time=300


em.web.port=8281
db.password=netqos
```

3. Save the changes.
4. Stop the Event Manager service using the command line:
   `service caperfcenter_eventmanager stop`
5. Restart the Event Manager service using the command line:
   `service caperfcenter_eventmanager start`

The Event Manager starts and uses the new value to determine the retention period.

Use Events to Monitor Device Performance

DX NetOps Performance Management can be configured to generate events when devices deviate from normal performance expectations. These events help you monitor the health of your network and react to correct performance issues.

DX NetOps Performance Management generates events using monitoring profiles. Monitoring profiles contain a set of event rules. Using metrics, these rules define the conditions that trigger events. To implement your event rules, associate the monitoring profile with a device collection.

Example:

An organization recently virtualized several business critical applications. The IT Architect and the Operations Center Manager want to monitor their virtual servers to ensure that the servers can handle the load from these applications. They create a monitoring profile, and they add event rules to find over-utilized CPUs and virtual memory issues for the collection of virtual devices. They associate the collection that includes the virtual servers with the monitoring profile. DX NetOps Performance Management automatically evaluates all the devices in the collection after every poll for each device. If needed, DX NetOps Performance Management raises or clears events when the devices trigger event rule conditions.

NOTE
You can generate user-visible alarms in CA Spectrum from events that are processed and logged in DX NetOps Performance Management. For more information, see the CA Spectrum documentation.

To monitor device performance against specific thresholds, complete the following process:

Create a Custom Device Collection

The device collection defines which devices DX NetOps Performance Management monitors with a monitoring profile. To create a device collection that includes the target devices, create a custom group and add the devices to the group. Use group rules, or manually add the devices to the group

Create a Monitoring Profile and Add Event Rules

To specify the conditions that generate an event, create a monitoring profile and add event rules to the profile.

Example:

The IT Architect and the Operations Center Manager define the threshold criteria. You add the following event rules:

- Add a VMware memory utilization rule, as follows:
  - Violation occurs when memory utilization is above 80 percent for 300 seconds (5 minutes) within a 900-second (15-minute) window.
  - Clear the violation when the memory utilization is equal to or below 75 percent for 300 seconds within a 900-second window.
- Add a VMware CPU utilization rule, as follows:
  - Violation occurs when both of the following conditions are met:
    - Condition 1: CPU utilization is above 70 percent.
    - Condition 2: CPU utilization is above one standard deviation.
  - These conditions occur for 300 seconds within a 900-second window.

Assign the Monitoring Profile to the Custom Device Collection

To begin monitoring the devices, and to activate event rules, assign the monitoring profile to the custom device collection.

Follow these steps:

1. Navigate to the Data Aggregator data source.
2. Click **Monitoring Configuration**, **Collections**.
3. Select the device collection, and click the **Monitoring Profiles** tab.
4. Click **Manage**.
5. Click and drag the monitoring profile to the **Assigned Monitoring Profiles** list.
6. Click **Save**.
   DX NetOps Performance Management begins monitoring this collection of devices using the monitoring profile and event rules. Events that are generated appear in the Events Display dashboard.

TIP

You can configure notifications for the threshold events.

Threshold Monitoring and Threshold Limiter Behavior

The threshold limiter monitors how long the evaluation engine takes to process rules in the Data Aggregator. If the threshold monitoring exceeds the specified percentage of the poll cycle, the evaluation engine enters a DEGRADED state. In the DEGRADED state, the evaluation engine waits for the monitoring to drop below the specified percentage. After a specified time, the threshold monitoring engine reassesses whether to suspend threshold evaluations. If threshold violations continue to exceed the percentage during the time period, the evaluation engine is suspended. Threshold evaluations will not resume, even if you restart the Data Aggregator. If the threshold violation does not exceed the specified percentage, the evaluation engine returns to normal operation.

WARNING
Do not modify the threshold limiter settings. The default settings provide protection against potential polled data loss. For any changes to the limiter settings, contact CA Support.

View the Threshold Monitoring Dashboard

The Threshold Monitoring dashboard provides information about the state of the threshold monitoring engine. Use this dashboard to see changing trends over time. This dashboard includes two views:

- **The Number of Event Rules Evaluated - Total**
  This view displays the number of actual rule evaluations that have occurred for an associated set of polled items.
- **Percentage of Poll Cycle to Complete Event Processing**
  This view displays the percentage of the poll cycle that the threshold monitoring engine takes to complete event processing.

Follow these steps:

1. Navigate to **Administration**, and click a Data Aggregator data source.
2. In the Tree View tab, expand the **All Data Aggregators** collection, and select the same Data Aggregator data source.
3. Click the name of the Data Aggregator on the Details tab.
4. Click the **Threshold Monitoring** tab in the Data Aggregator Pages view.

Threshold Monitoring Engine Status Events

Threshold monitoring engine status events describe the status of threshold evaluations. You can see these events in the Data Aggregator Events tab. The following table shows the possible Threshold monitoring engine status events:

Event Type	Event Subtype	Description
Administration event	Threshold monitoring engine status	Threshold evaluations have been enabled.
Administration event	Threshold monitoring engine status	The Threshold Monitoring Engine has transitioned to a degraded state. This means that threshold evaluations are taking longer than the configured threshold {X} and will be suspended in {X} minutes if this condition persists.
Administration event	Threshold monitoring engine status	Threshold evaluations have been disabled by the System Administrator.
Administration event	Threshold monitoring engine status	Threshold evaluation operations are still taking longer than the configured threshold {X}. The system is suspending threshold evaluations. Please contact the System Administrator to evaluate the monitoring configuration.
Administration event	Threshold monitoring engine status	The Threshold Monitoring Engine is no longer degraded and is functioning normally.
Administration event	Threshold monitoring engine status	The Threshold Monitoring limiter has been disabled.
Administration event	Threshold monitoring engine status	The Threshold Monitoring limiter has been enabled.
Administration event	Threshold monitoring engine status	Threshold evaluation operations are longer than the configured maximum threshold {X}. The system is suspending threshold evaluations. Please contact the System Administrator to evaluate the monitoring configuration.

Take Action If Threshold Evaluations Are Suspended

If threshold evaluations are suspended, consider the following options before you resume evaluations:

- Try to correlate the change in performance to configuration changes in the system.
- Reduce the overall number of active event rules. Turn off event rules one at a time. Check the performance after you turn off each rule before turning off another rule.
- Reduce the overall number of active event rules that have windows greater than 300 seconds.
- Reduce the number of Violation event conditions within event rules.
- Reduce the number of event rules that use a condition type of Standard Deviation.
- Verify that only *required* collections are applied to the monitoring profile or threshold profiles that contains event rules.
- Verify that only *required* devices are contained within collections that are associated with these monitoring profiles or threshold profiles.

Threshold Limiter Behavior

To determine whether to suspend threshold evaluations, the limiter looks at how long the engine takes to evaluate thresholds as a percentage of the poll cycle time.

Percentage of Poll Cycle Threshold specifies the percentage of the poll cycle that can be used to monitor thresholds. By default, the Percentage of Poll Cycle Threshold attribute value is 80 percent.

For example, 4 minutes for items that are polled at a 5-minute rate. The threshold monitoring engine becomes DEGRADED when the engine takes more than 240 seconds to complete threshold evaluations. An event is generated on the Data Aggregator item when the threshold monitoring engine becomes DEGRADED.

Recovery Interval specifies how long the threshold monitoring engine remains in the DEGRADED state. By default, the Recovery Interval attribute is 15 minutes. If the processing time does not drop below the specified percentage with the recover interval, threshold evaluations are suspended. An event is generated on the Data Aggregator item when threshold evaluations are suspended.

Resume Threshold Evaluations

Threshold evaluations are not resumed automatically after they are suspended. Resume them manually.

WARNING

If threshold evaluations are suspended frequently, contact CA Support.

Follow these steps:

1. Enter the following information in a web browser:
   ```
   URL: http://DA_host:port/rest/thresholdmonitoring/config
   ```

2. Take note of the ID value of the ThresholdMonitoringConfiguration item.
   Example:
   ```
   <ThresholdMonitoringConfigurationList>
      <ThresholdMonitoringConfiguration version="1.0.0">
         <ID>16</ID>
        <ThresholdMonitoringEnabled>true</ThresholdMonitoringEnabled>
        <PercentOfPollCycleThreshold>80</PercentOfPollCycleThreshold>
        <ThresholdMonitoringLimiterEnabled>true</ThresholdMonitoringLimiterEnabled>
        <RecoveryIntervalInMinutes>15</RecoveryIntervalInMinutes>
      </ThresholdMonitoringConfiguration>
   </ThresholdMonitoringConfigurationList>
   ```

3. Open a REST client editor or HTTP tool that sends requests and gets responses.
4. Set the Content-type to application/xml.
5. Enter the following filter criteria:
   - URL: http://*DA_host*:*port*/rest/thresholdmonitoring/config/*ID*
     - ***ID***
       The identification number that is assigned to the ThresholdMonitoringConfiguration item.
   - HTTP method = PUT
   - Resume threshold evaluations on the Body tab of the HTTP Request pane:
     ```
     <ThresholdMonitoringConfiguration version="1.0.0">
          <ThresholdMonitoringEnabled>true</ThresholdMonitoringEnabled>
     </ThresholdMonitoringConfiguration>
     ```

   Threshold evaluations resume.

Change the Default Behavior of the Threshold Limiter

In some situations, CA Technologies may recommend that you modify the behavior of the threshold limiter.

Follow these steps:

1. Navigate to the following URL:
   `http://DA_host:port/rest/thresholdmonitoring/config`
2. Note of the ID value of the ThresholdMonitoringConfiguration item.
3. Open a REST client editor or HTTP tool that sends requests and gets responses.
4. Set the Content-type to application/xml.
5. Enter the following filter criteria:
   - URL: http://*DA_host*:*port*/rest/thresholdmonitoring/config/*ID*
     - ***ID***
       The identification number that is assigned to the ThresholdMonitoringConfiguration item.
   - HTTP method = PUT
   - Increase the Percentage of Poll Cycle Threshold value on the Body tab of the HTTP Request pane.

     ```
     <ThresholdMonitoringConfiguration version="1.0.0">
          <PercentOfPollCycleThreshold> percent </PercentOfPollCycleThreshold>
          <RecoveryIntervalInMinutes> minutes </RecoveryIntervalInMinutes >
     </ThresholdMonitoringConfiguration>
     ```

     - **percent** specifies the percentage value.
     - **minutes** specifies the number of minutes to wait before reassessing the threshold monitoring engine.

   The threshold limiter runs with the updated values.

Disable the Limiter

In rare situations, CA Technologies may recommend that you disable the threshold limiter.

Follow these steps:

1. Navigate to the following URL:
   `http://DA_host:port/rest/thresholdmonitoring/config`
2. Note of the ID value of the ThresholdMonitoringConfiguration item.
3. Open a REST client editor or HTTP tool that sends requests and gets responses.
4. Set the Content-type to application/xml.
5. Enter the following filter criteria:
   - URL: http://*DA_host*:*port*/rest/thresholdmonitoring/config/*ID*
     - ***ID***
       The identification number that is assigned to the ThresholdMonitoringConfiguration item.
   - HTTP method = PUT
   - Disable the limiter on the Body tab of the HTTP Request pane:

     ```
     <ThresholdMonitoringConfiguration version="1.0.0">
          <ThresholdMonitoringLimiterEnabled>false</ThresholdMonitoringLimiterEnabled>
     </ThresholdMonitoringConfiguration>
     ```

   The limiter is disabled and the evaluation engine cannot enter a DEGRADED state.

Threshold Event Processing Self-Monitoring Metrics

To determine if you are doing too much eventing, monitor the key performance indicators in Data Aggregator. Eventing in Data Aggregator is performed in batches, for example, events are simultaneously evaluated and generated for large groups of items. Several metrics provide self-monitoring to assess the health of the Data Aggregator system.

To view these metrics, add a custom IM Device MultiTrend view to a dashboard. Edit the dashboard, using the following metrics from the metric family **Data Aggregator Event Calculation Times**:

- **Event Process Queue Size** shows the size of the event processing queue. An increase in queue size without a subsequent recovery (trending downward) indicates that eventing is backed up.
- **Count of Cleared Events** indicates the number of cleared events that are in the reporting resolution window.
- **Count of Created Events** indicates the number of raised events that are in the reporting resolution window.
  A continuously large number of events that are raised or cleared can affect the Event Manager database. These metrics can indicate when your system has exceeded the recommended event generation rate. Event generation/clear bursts are acceptable.
- **Count of Processed Event Rule Evaluations** indicates the sum of event rules multiplied by the number of items those rules are applied to. The higher the number of evaluations, the more work your system is doing. Some evaluations are more expensive than others. For example, evaluations with more conditions, more standard deviation conditions, or longer duration and window are more expensive. The total acceptable number of evaluations depends on your event rules.
- **Total Time to Calculate Events** indicates the total amount of time that was spent processing events for this metric family. If the value of this metric exceeds the number of seconds in the reporting resolution window, the eventing was delayed or backlogged at that point in time.

In general, steady values for these self-monitored metrics indicate a healthy system. Some intensive database jobs cause fluctuation in these self-monitoring metrics. Typically, these jobs run between 2 AM and 4 AM UTC. Turn on eventing slowly and judge the system health before moving forward with different rules. Monitor the health of the system over 24 hours after each subsequent change.

Errors in the Karaf log on the Data Aggregator system may also indicate that your system is under stress.

Modern Network Monitoring

DX NetOps Virtual Network Assurance (VNA) provides modern network monitoring for software-defined architectures and hybrid cloud platforms. DX NetOps Virtual Network Assurance enables comprehensive coverage with monitoring that is scalable and heterogeneous across the greatest number of technology stacks in the following architectures:

- Traditional
- SDN
- SDDC
- SD-WAN
- NFV
- Hybrid-cloud

To show you both virtual network data and traditional physical infrastructure performance information, DX NetOps Virtual Network Assurance collects data from these architectures and delivers that information to DX NetOps Performance Management.

For more information, see the DX NetOps Virtual Network Assurance documentation.

Load VNA Data

To populate dashboards with VNA data, load the inventory and performance data. For more information, see Manage Data from CA Virtual Network Assurance.

VNA Domains

DX NetOps Performance Management uses DX NetOps Virtual Network Assurance domains as reporting groups.

Modern Network Monitoring Dashboards

NetOps Portal provides various dashboards for viewing VNA data related to the relevant technologies and architectures of your environment.

Manage Data from Virtual Network Assurance

To display data for software-defined networking (SDN) and network functions virtualization (NFV) controllers and orchestrators, connect to DX NetOps Virtual Network Assurance (VNA). VNA includes the Gateway component, which delivers SDN/NFV data to DX NetOps Performance Management. After you configure the connection, VNA sends the inventory data to DX NetOps Performance Management. VNA sends updates to inventory information and performance data as that information is collected from the virtual network.

> **NOTE**
> To load performance data, VNA must be configured to collected data from the virtual network. For more information, see the DX NetOps Virtual Network Assurance documentation.

The data collector can support multiple Gateways, however, only one Gateway can be active at a time. You can delete a Gateway and its associated inventory. This action deletes the associated VNA items on DX NetOps Performance Management and deletes the historical performance data. Alternatively, to preserve the inventory and performance data, you can edit the lifecycle state of the Gateway. With a lifecycle state set to `Retired` , the data collector does not process data from the Gateway.

The following video shows the configuration process:

Configure the Connection in NetOps Portal

Follow these steps:

1. Hover over **Administration**, and then click **Monitored Items Management: VNA Gateways**.
2. Click **New** or **Edit**.
3. Specify the VNA host and port, and then select the data collector that receives VNA data.
   Default Port: 8080
4. Select the **Administrative Status**.
   With the Administrative Status set to `Up` , the data collector registers VNA to receive inventory and performance data.
5. Select the **Life Cycle State**.
6. Click **Save**.

Configure the Connection Through REST

Use the following REST information configure the connection:

URL: `http://`*`da_host`*`:8581/rest/tenant/`*`tenantID`*`/sdnGateways`

- **tenantID** is the ID of the tenant to which the data collector belongs. For the default tenant, the tenant ID is usually 1.

  > **TIP**
  > To get the tenant ID, use the following REST URL: `http://`*`da_host`*`:8581/rest/tenant`

Method: POST

Body:

```
<SDNGateway version="1.0.0" >
   <Item version="1.0.0">
     <Name>GatewayName</Name>
     <MDRItemID>DC_ItemID</MDRItemID>
   </Item>
   <AdminStatus>status</AdminStatus>
   <LifeCycleState>state</LifeCycleState>
   <Hostname>host</Hostname>
   <Port>8080</Port>
 </SDNGateway>
```

- **Name** assigns a name to the Gateway item.
- **MDRItemID** specifies which Data Collector the SDN Gateway connects to.
- **AdminStatus** determines whether the data collector receives inventory and performance data from the Gateway.
  Values: `UP` or `DOWN`
- **LifeCycleState** determines whether the data collector processes data from the Gateway. Only one Gateway can be active at a time.
  Values: `ACTIVE` or `RETIRED`
- **Hostname** is the hostname of the SDN Gateway.
- **Port** is the port for the SDN Gateway.
  Default: 8080

Example:

```
<SDNGateway version="1.0.0" >
  <Item version="1.0.0">
    <Name>SDN GW 2</Name>
    <MDRItemID>762</MDRItemID>
  </Item>
  <AdminStatus>UP</AdminStatus>
  <LifeCycleState>ACTIVE</LifeCycleState>
  <Hostname>sdn-gw2</Hostname>
  <Port>8080</Port>
</SDNGateway>
```

If the AdminStatus set to `UP` , the data collector registers the Gateway to send it inventory and performance data.

Monitor Virtual Inventory

In SDN and NFV environments, your virtual inventory changes according to the requirements of your services. Modern network monitoring provides insight into the trends in virtual inventory. Inventory metrics provide assurance that the programmatic processes that manage the network are functioning properly. These metrics also highlight anomalies in provisioning.

The SDN/NFV Virtual Inventory Overview dashboard shows your virtual inventory.

The SDN/NFV Virtual Inventory Overview dashboard provides the following information:

- VNF count by type over time as a trend chart and a stacked chart
- Service chain and virtual network inventory trends
- VNF inventory trends by type

NOTE

The Other VNFs Count view shows VNFs where DX NetOps Performance Management cannot determine the type.

Monitor SDN/NFV Virtual Resource Usage

To investigate performance degradation and for capacity planning, monitor the resource usage of your virtual environment.

DX NetOps Performance Management provides two dashboards that show resource utilization:

SDN/NFV Virtual Compute Usage Overview Dashboard

This dashboard shows virtual CPU and memory usage for VNFs. The views use different visualizations to show aggregated and individual resource utilization. By default, views show the VNFs with the highest utilization.

SDN/NFV Virtual Storage Usage Overview Dashboard

This dashboard shows virtual storage, reads, and writes for VNFs. The views use different visualizations to show aggregated and individual resource utilization. By default, views show the VNFs with the highest utilization.

The following video highlights the key capabilities of this dashboards:

Monitor SDN/NFV Physical Host Resource Usage

To investigate performance degradation and for capacity planning, monitor the physical resource usage of your virtual environment. In a virtual network, all virtual network functions (VNF) run on physical hosts. If the load on the physical hosts is too high, the VNFs compete for resources, leading to performance degradation.

DX NetOps Performance Management provides the SDN/NFV Physical Server Usage Overview dashboard which shows the status of the physical infrastructure that supports the virtual environment.

This dashboard includes the following information:

- Aggregated CPU, memory, and disk utilization for physical servers
- Availability and Reachability information
- Individual servers with the highest CPU, memory, and disk utilization
- Inventory of all servers in the SDN/NFV environment

The following video highlights the key features of this dashboard:

Monitor Service Chains

In an SDN/NFV environment, services are delivered dynamically through virtual network functions (VNF). The service chain represents the required VNFs and the stack of physical and virtual building blocks that support the VNF.

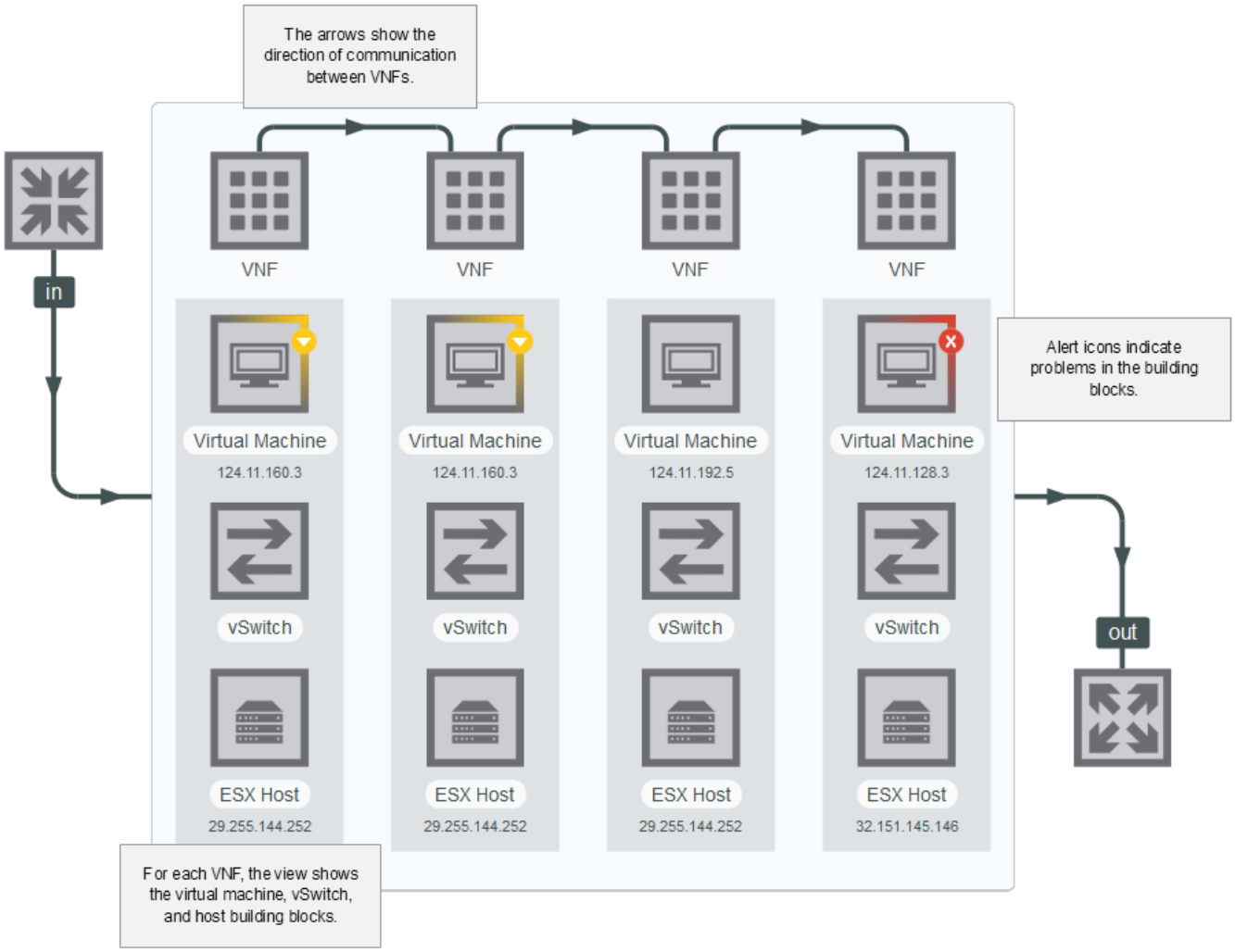
The Service Chain page provides information about the status of service chains. The main view shows the VNFs in the service chain and includes the virtual and physical building blocks. The view indicates the type of each VNF with an abbreviation. The arrows show whether communication is one-way or bidirectional.

To access the Service Chain page, go to **Inventory**, **Service Chains**, and click a service chain in the list. The view provides the following functionality:

- To show name, type, IP address, and performance information, hover over a building block in the service chain.
- Alert icons indicate problems in the building blocks. Hover over the icon for information about the violated threshold.
- To load the detailed context page for a building block, right-click a building block, and select a context.

The following image shows the important elements of the service chain visualization:

Figure 53: Service Chain Details Elements



Service Chain Thresholds

To configure the thresholds for the alert icons, click the **View Settings** (gear) icon, and click **Edit**. Assign Critical (red), Major (orange), and Minor (yellow) threshold for the following metrics:

- CPU utilization
- Memory utilization

VNF Types

The view uses the following abbreviations to identify the currently defined VNF types:

- **vFW** Firewall
- **vLB** Load Balancer
- **vNAT** NAT
- **vOPT** Optimizer
- **vCMF** Content Filter
- **vDPI** DPI
- **vWOL** WAN Accelerator
- **vADC** ADC
- **vCACHE** Cache
- **vROUTER** Router
- **Other**

Monitor vSwitch Performance

To identify bottlenecks in your SDN or NFV environment, monitor vSwitch performance. Because vSwitches connect VNFs to each other and to physical interfaces, the performance of the vSwitch determines the overall bandwidth of the connected VNFs. With the transition from physical to virtual networking, the generalized architecture of the x86 box replaces the highly specialized hardware of the physical switch. Because this architecture is not optimized for switch performance, vSwitch performance is an important indicator of overall virtual network health.

The following video highlights this feature:

NOTE

In DX NetOps Performance Management, vRouters appear as vSwitches.

SDN/NFV vSwitch Performance Overview Dashboard

The SDN/NFV vSwitch Performance Overview dashboard lets you analyze the overall performance of vSwitches in your environment. This dashboard uses different visualizations to show the following information about vSwitches:

- The inventory of vSwitches in the virtual network
- The vSwitch throughput in bits and packets
- The minimum, maximum, and average throughput in bits and packets
- The vSwitch utilization and throughput in bits and packets aggregated to the device level
- The CPU and memory utilization of vSwitch service

TIP

The views that show vSwitch network performance, such as throughput, errors, and discards, show the top vSwitches for these metrics. Use these views to drill in to the vSwitch context page for more details.

vSwitch Context Page

The vSwitch context page provides detailed information about a specific vSwitch. To access the vSwitch context page, click a vSwitch in the inventory, in another view, or in the service chain visualization.

Details Tab

The Details tab shows basic information about the vSwitch, a list of events, and availability information. Use this tab for basic troubleshooting.

Network Tab

The Network tab shows network performance information for the vSwitch. Use this tab to identify network performance bottlenecks that are related to the vSwitch. The tab includes the following information:

- Average interface utilization
- Errors and discards for interfaces on the vSwitch
- Aggregated throughput in bits and packets
- Interface throughput in bits and packets
- Inventory of the virtual interfaces

High dropped packets might indicate that the vSwitch service is overburdened.

Resource Tab

The Resource tab shows resource use for the vSwitch service. Use this tab to ensure that the vSwitch is not consuming too much of the host resources. The tab includes the following information:

- Current CPU and memory utilization

  NOTE

  CPU and memory utilization for the vSwitch service are compared against the total resources available to the host.
- Trend chart for CPU and memory utilization

High utilization for the service means that other VMs on the host might have insufficient resources.

Virtual Interface Context Page

The virtual interface context page provides performance information for interfaces on your vSwitches. To access the virtual interface context page, click an interface in the inventory or in another view.

Details Tab

The Details tab shows basic information about the interface and a list of events for to the interface.

Health Tab

The Health tab shows performance information for the virtual interface, such as the following information:

- Discards
- Errors
- Interface utilization
- Throughput in bits and packets

Monitor Cisco ACI

Cisco Application Centric Infrastructure (ACI) uses Nexus 9000 switches to create a dynamic virtual environment that hosts and serves applications. Monitor the Cisco ACI environment to verify that everything is operating as expected and to highlight problem areas. DX NetOps Performance Management provides dashboards and context pages that support Cisco ACI monitoring.

ACI Console

The ACI Console provides a searchable inventory list that shows relationships between items in the Cisco ACI environment. Where relevant, the inventory shows the most recent ACI health score.

- To limit the list to a particular node in the hierarchy, double-click that node.
- To return to a higher level of the hierarchy, click the breadcrumb links.
- To filter the inventory by a health-score threshold, use the slider at the top of the pane. The inventory shows nodes with children with health scores equal to or lower than the threshold.

The ACI Console also provides relationship diagrams for various aspects of the ACI inventory. The diagrams show alert icons on items that exceed customizable thresholds. Use the diagrams to understand how particular problems with the ACI infrastructure affect your applications. The ACI Console includes diagrams for the following inventory item types:

- **Application Profile Diagram**
  Shows the relationships between endpoint groups (EPGs) that support the application profiles (APs). Provider EPGs are connected to contracts connected to consumer EPGs. The diagram also shows the underlying leafs that support the endpoints in the EPGs.
- **EPG Diagram**
  Shows the AP parent and each end point. For each endpoint, the diagram shows the associated leaf.
- **Endpoint Diagram**
  Shows the EPG parent. If the endpoint is a vLAN, the diagram shows the connection to the leaf. If the endpoint is an application, the diagram shows the full supporting technology stack.
- **Leaf Diagram**
  Shows application profiles that contain endpoints which are connected to the leaf.
- **APIC Diagram**
  Shows leafs that are connected to the APIC controller.

  NOTE

  Because each spine is connected to each leaf and has no other relationships, the ACI console does not include a diagram for spines.

All diagrams share common functionality:

- To open a diagram, click an item in the inventory.
- To show more details for an item in the relationship map, hover over the icon for the item.
- To show which thresholds are exceeded, hover over the alert icon in the relationship map.
- To open the relationship map for another item, click the icon for an item in the current relationship map. The previous relationship map becomes a breadcrumb link on the right side of the diagram.
- To open the context page for an item, click the item name in the relationship map.
- To go to the ACI Health dashboard or the Switches Overview dashboard, click links in the upper left corner of the dashboard.

ACI Console Thresholds

To configure the thresholds for the alert icons, click the **View Settings** (gear) icon, and click **Edit**. Assign Critical (red), Major (orange), and Minor (yellow) threshold for the following metrics:

- CPU utilization
- Health score
- Interface utilization
- Memory utilization

The following image shows the important elements of the ACI Console:

Figure 54: ACI Console Elements

ACI Health Dashboard

The ACI Health dashboard provides focused information about your Cisco ACI environment. To highlight problem areas, the dashboard focuses on health scores and faults. To narrow the area of troubleshooting, change the context of the dashboard:

- To find problematic switches, scope to the fabric.
- To find a problem in a tenant, scope to a tenant.
- To find a problem in an application, scope to an AP.

This dashboard shows the following information:

- Counts of items in the Cisco ACI environment
- Count of critical faults
- ACI health scores for items in the environment
- Aggregated fault trends by severity
- AP and EPG inventory

Switches Overview Dashboard

The Switches Overview dashboard provides information about the status of switches. If you select the appropriate group of Nexus 9000 switches, you can monitor the infrastructure behind an ACI environment. For example, you can analyze critical faults, CPU, memory, and interface utilization for these switches.

NOTE

Most metrics on this dashboard require SNMP data collection from the Nexus 9000 switches through DX NetOps Performance Management.

This dashboard shows the following information:

- Switches with the highest CPU and memory utilization
- Switches with the highest interface utilization
- Switches with the highest policy CAM utilization
- Switches with the most critical faults

ACI Workflows

The ACI dashboards are designed with the following role-based workflows:

- An **ACI System Administrator** is responsible for maintaining the Cisco ACI environment.
  a. Use the ACI Console to look at the fabric and VMs:
     - View the relationship of the fabric to APs and EPGs.
     - View the relationship of fabric nodes to compute resources.
     - View the relationship of VMs to APs and EPGs.
     - View the relationship of VMs to fabric nodes and compute resources.
  b. Use the ACI inventor to track the number of entities in the ACI environment.
  c. Track the performance of APs and EPGs on context pages.
- A **Tenant Administrator** is responsible for managing tenants in the ACI environment.
  a. Scope the ACI Console to the tenant, and look at VM relationships to APs, EPGs, compute resources, and the fabric.
  b. Use the ACI Health dashboard to look at the health score and faults within the tenant.
  c. Track the performance of APs and EPGs on context pages.
- An **Application Owner** is responsible for a single application that is hosted in the ACI environment.
  a. Scope the ACI Console to the AP, and look at VM relationships to the AP, EPGs, compute resources, and the fabric.
  b. Use the AP context page to monitor computer and storage utilization by VM and in aggregate and top VM utilization.
- A **Fabric Administrator** is responsible for maintaining the health of the network in the ACI environment.
  a. Scope the ACI Console to the fabric, and look at fabric relationships to APs and EPGs and the relationships of fabric nodes to compute resources.
  b. Use the ACI Health dashboard to look at the health score and faults within the fabric.
  c. Track the performance of the Nexus 9000 switches on the Switches Overview dashboard and the context pages for the switches.

Monitor SD-WAN

SD-WAN solutions, like Viptela and Cisco IWAN, route traffic according to predefined performance requirements. DX NetOps Performance Management provides dashboards and context pages that support SD-WAN monitoring. Dashboards and context pages allow you to monitor the following items:

- **Tunnel**
  Represents the connection between two devices and has polled statistics (jitter, latency, and packet loss).
- **Application/SLA path**
  Represents a tunnel in relation to service level agreements (SLAs) thresholds. The measured SLA metrics (jitter, latency, packet loss) are reported as percentages of the related SLA thresholds. Rather than reporting tunnel traffic, application/SLA paths show the ability of a tunnel to meet the SLAs for different types of traffic (for example, voice or video). These traffic types are referred to as SLA classes.

  NOTE

  Viptela sites map to site groups in NetOps Portal. If desired, you can manage the site groups in NetOps Portal to update the site name. For more information, see Manage Groups.

For information about the metrics that Viptela collects including the VNA metrics, SNMP metric families, and NFA metrics, see the CA Virtual Network Assurance documentation.

The following video examines how DX NetOps Performance Management supports the Viptela SD-WAN technology:

SD-WAN Tunnel and Application Paths Dashboards

View the SD-WAN Tunnel Statistics dashboard or the SD-WAN App Path Statistics dashboard to view the health of your SD-WAN tunnels or application/SLA paths. The SD-WAN App Path Statistics dashboard lets you easily identify issues impacting your delivering of services and applications according to service level agreements (SLAs).

The SD-WAN Tunnel Statistics dashboard reports the following metrics:

- Jitter
- Latency
- Packet loss

The SD-WAN App Path Statistics dashboard reports the following metrics:

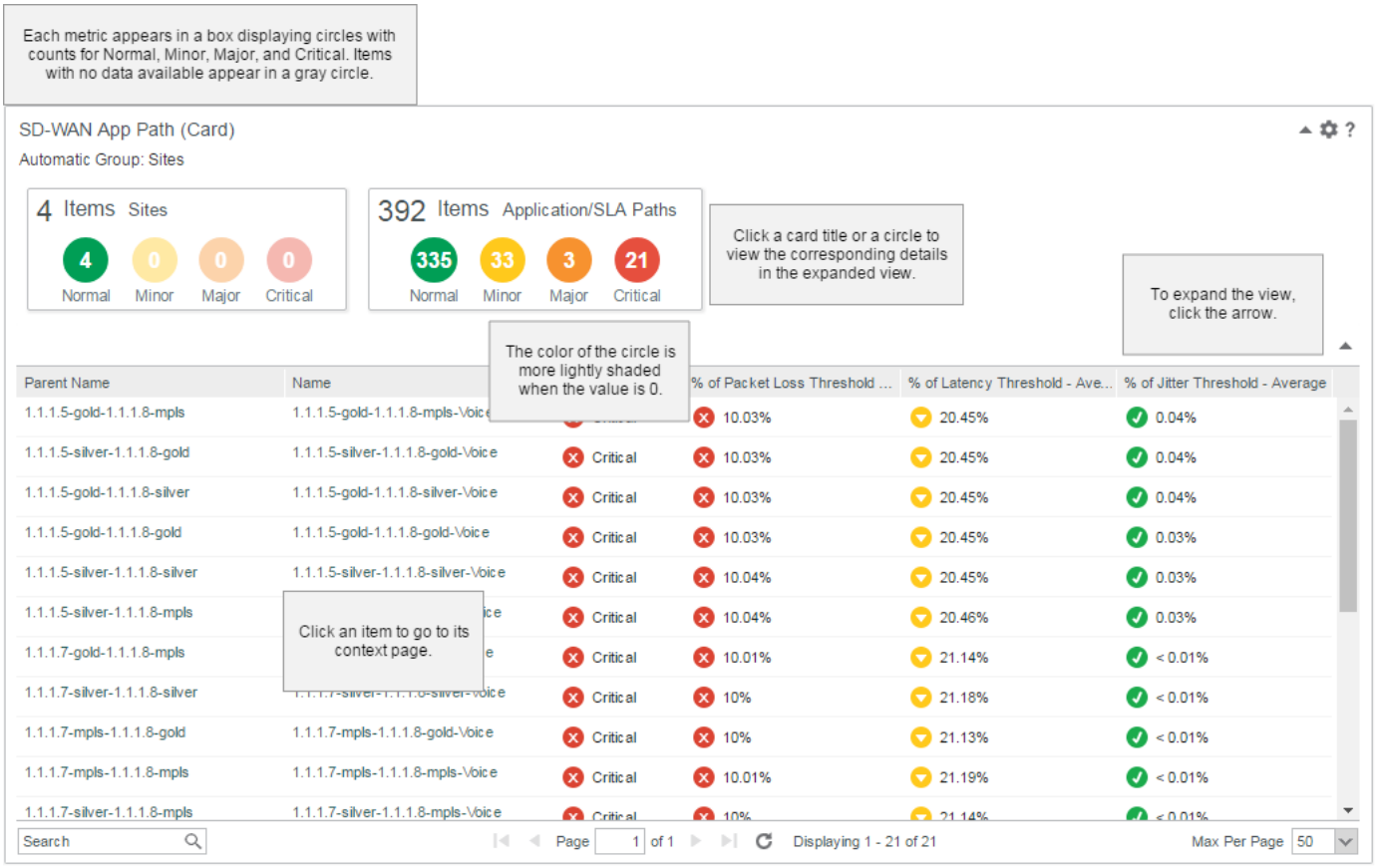
- Percentage of Jitter SLA Threshold
- Percentage of Packet Loss SLA Threshold
- Percentage of Latency SLA Threshold

The following details appear in these dashboards:

- **Health Counts**
  View the aggregated health counts of the following items:
  - Sites
  - Edge Routers
  - Application Paths
  - Applications
  - Tunnels

  Counts appear in color-coded circles for each threshold range. To expand the view, click the arrow in the lower-right corner. Click a card title or a circle to view the corresponding details in the expanded view.

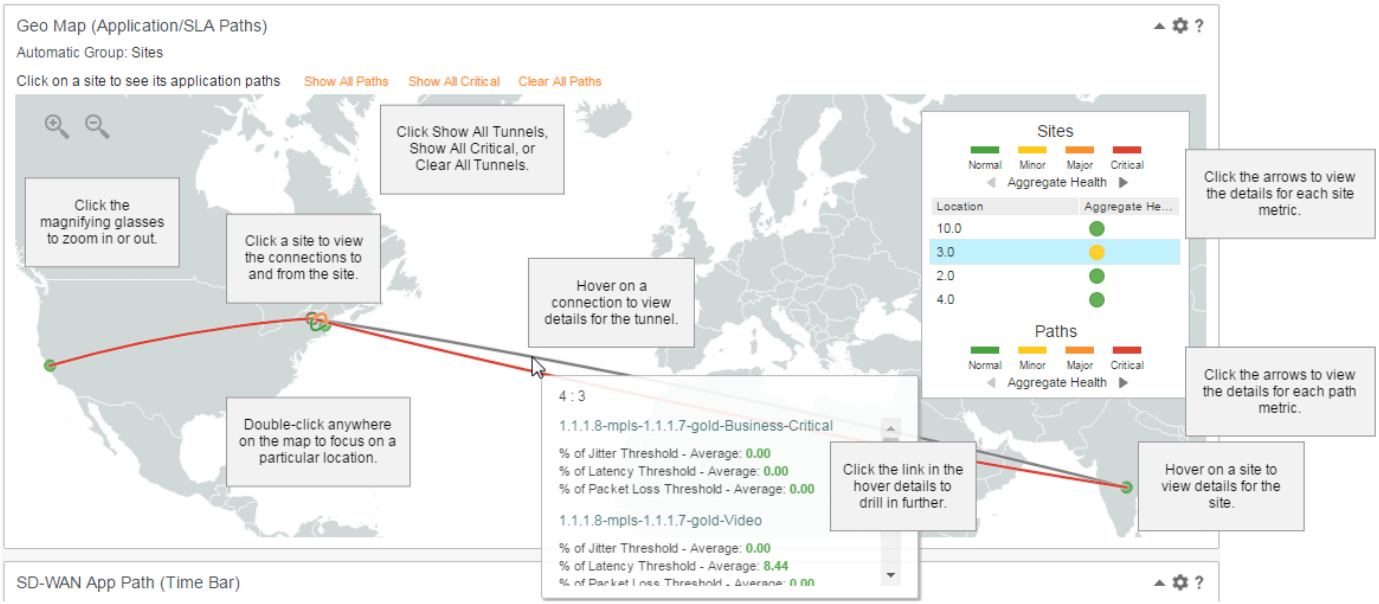
Figure 55: SD-WAN Card



- **Geographic Map**
  View the location of each site. When a site is selected, the connections to and from other sites appear. The connection lines are color-coded based on health metrics. Site router details appear when you hover over a site. Tunnel or application/SLA path details appear when you hover over a connection.

  NOTE
  Only sites within the page-level Group context are shown even when a tunnel or application path runs between an included and excluded site.

Figure 56: SD-WAN Map



- **Timeline**
  View time bar charts aggregating packet loss, latency, and jitter metrics and time bar charts for each metric.

  NOTE
  Only sites within the page-level Group context are shown even when a tunnel or application path runs between an included and excluded site. The selected site in the Map view filters this view.

Figure 57: SD-WAN Timebar

- **Scorecard**
  View tunnel or application/SLA path metrics by subgroup and component for the selected group. Subgroups and the items in those groups appear in a hierarchical format. Colored icons indicate performance levels for metrics and an overall health indicator.

Figure 58: SD-WAN Scorecard

- **Statistics**
  View tunnel or application/SLA metrics in a table with color-coded bars for percentage metrics.
- **Gauge**
  In the SD-WAN Tunnel Statistics dashboard, view packet loss in a gauge/table view.
- **Trend**
  View packet loss, latency, and jitter in a trend/table view.
- **Traffic**
  View latency and jitter metrics side by side for comparison and in a trend chart.
- **Inventory**
  View the source and destination end points of your tunnels or application/SLA paths.
- **Errors and Discards**
  In the SD-WAN Tunnel Statistics dashboard, view SD-WAN virtual interface errors in one table and SD-WAN virtual interface discards in another table.

SD-WAN Context Pages

From the inventory or the SD-WAN dashboards, you can navigate to context pages containing more details for the following SD-WAN items:

- **Devices**
  View the typical device context page for any of your SD-WAN devices.
- **Application/SLA Paths**
  On the Details tab, view the details for the selected application/SLA path. View its statistics, packet loss in a gauge, heat chart, and trend chart, and its traffic.
  On the Class Health tab, view the percentage of packet loss, percentage of jitter, and the percentage of latency. These metrics appear in gauges, heat charts, side-by-side bar charts, trend charts, and a bar chart table.
- **Tunnels**
  On the Details tab, view the details for the selected tunnel. View its statistics, packet loss in a gauge, heat chart, and trend chart, and its traffic.
- **Virtual Interfaces**
  On the Details tab, view the details for the selected virtual interface with its event list.
  On the Health tab, view trend charts for discards, utilization, errors, packet rate, and bit rate.

SD-WAN Workflows

The following video examines SD-WAN application/SLA paths and why CA Network Operations Analytics monitoring is so important to delivering a reliable digital experience:

The following workflows illustrate how an Operations Engineer can use the SD-WAN dashboards and context pages for SD-WAN monitoring:

1. One of the following events drives you to the relevant SD-WAN dashboard:
   – For an event impacting site connectivity, go to the SD-WAN Tunnel Statistics dashboard.
   – For an event that could impact service level agreements (SLAs), go to the SD-WAN App Path Statistics dashboard.
2. Use the Health Counts to view the site connections or application/SLA paths with issues.
3. Use the Geographic Map and Scorecard to view site details.
4. Review the remaining trend data.
5. Drill into edge devices.

Monitor AWS

CA Virtual Network Assurance monitors the AWS Cloud environment to verify that everything is operating as expected and to highlight problem areas. DX NetOps Performance Management provides custom dashboards and context pages that support AWS cloud monitoring. the dashboards and context pages allow you to monitor the following items:

- **Tunnel**
  Represents the connection between two devices and has the following polled statistics:
- – TunnelState
  – TunnelDataIn
  – TunnelDataOut
- **Ec2**
  Represents a virtual machine and has the following polled metrics:
  CPU
- – Utilization
  – Aggregated Network Incoming Bytes
  – Aggregated Network Outgoing Bytes
  – Aggregated Network Incoming Packets
  – Aggregated Network Outgoing Packets
  Disk
- – Disk IOPS
  – Disk Read Bytes
  – Disk Write Bytes
  – Disk Capacity

AWS Context Pages

From the inventory, you can navigate to context pages for more information on the the following AWS items:

- **Devices**
  View the typical device context page for any of your AWS devices.
- **Tunnels**

Context page for AWS tunnel does not include any AWS cloud metrics. You must create custom dashboards to view tunnel performance metrics.

- **EC2s**
  In the Summary tab, you can view the details for the selected EC2 with the corresponding event list.
  In the VMware Virtual machine tab, you can view trend charts for CPU Usage, Active Memory Usage, Power State Connection state, and so on. You must create custom dashboards to view all supported EC2 performance metrics.
- **Virtual Interfaces**
  In the Details tab, you can view the details for the selected virtual interface with its event list.

Configure Notifications

Configure notifications for events that come from a data source to Event Manager. The incoming events are evaluated against the conditions that you configure for the notification criteria. Only when the criteria are met does Event Manager take a notification action. If an event does not trigger a notification, you can still display the event in the Event List.

A user only configures and receives notifications for events for an item in a group to which the user has access.

Consider the following information:

- Users cannot see the notifications of other users.
- The action to delete event notifications does not affect the actual or future events.

Notification Actions

When you configure notifications, you can specify the following actions.

Email

Send email notifications to one or more recipients when an event is raised or cleared. The email provides a link to the context page for the device or component that triggered the alarm.

> **NOTE**
> To use the hostname in the URL instead of the IP address of the NetOps Portal host, configure the **Web Site Host**. For more information, see Update Performance Center Website Settings.

Supported roles: Users with a role that contains the Create Notifications role right and Event Manager access can configure email notifications. However, the Administrator role must first specify an SMTP server.
In the Email tab, select Enable, and configure the email notification settings.

Trap

Send trap notifications to fault or network management system (NMS) in your environment, such as DX NetOps Spectrum.

> **NOTE**
> While DX NetOps Spectrum can receive traps, this method is not the preferred method to integrate with DX NetOps Spectrum. For more information, see the CA Spectrum documentation.
>
> **IMPORTANT**
> Create an SNMP profile with the outgoing trap port (typically 162) before creating the notification.

Supported roles: Users with the Administrator role (global administrators) can configure trap notifications. Administrators must also have product privileges to Event Manager and data sources that create events.

The trap receivers must be preconfigured to receive traps. Each destination can have its own configuration regarding SNMP community and IPV4 destination. To receive and decrypt SNMPv3 traps, the SNMP profile for the trap receiver

should match this notification configuration. For more information, see SNMP Profiles. For more information about trap formats, see the corresponding NMS documentation for your trap receiver.

Script

Define a script. Scripts can store events to a database, forward notifications to multiple systems, send specific types of notifications to some specific system, and so on. You can log output from scripts that you own. Script return codes are logged in the standard PC log file. Script notification actions are executed serially to ensure that sets are processed before clears. If the queue of unprocessed script notification actions exceeds a certain size (5000 by default), any new incoming script notification actions are dropped. An event is generated when events start dropping and when the processing of new script notification actions restarts. If a script takes too long to complete execution (300 seconds by default), it is killed and an event is generated. The script notification action events are generated on the Data Aggregator item in NetOps Portal.

For moreinformation about how to change event properties that are related to script notification actions, see Change Event Properties.

> **NOTE**
> Store scripts in the `/opt/CA/PerformanceCenter/NotificationScripts` directory.

The following video examines automatic script execution:

The following parameters are automatically passed to script notification actions:

- `CAPM_EventDataSource`
- `CAPM_EventCategory`
- `CAPM_EventType`
- `CAPM_EventSubType`
- `CAPM_EventState`
- `CAPM_EventSeverity`
- `CAPM_EventOccurredOn`
- `CAPM_EventDesc`
- `CAPM_ItemParentName`
- `CAPM_ItemName`
- `CAPM_ItemNameAlias`
- `CAPM_ItemDesc`
- `CAPM_IPAddress`
- `CAPM_ItemUrl`
- `CAPM_ItemType`
- CAPM_ItemSubtype
- CAPM_ItemId
- CAPM_ItemParentId

> **NOTE**
> Event-specific properties are also available and prefixed with "`CAPM_EvProp_` ". The `CAPM_ItemId` and `CAPM_ItemParentId` parameters are NetOps Portal IDs. The script parameters and notifications are always in English regardless of the NetOps Portal language.

Supported roles: Users with a role that contains the Create Notifications role right can create or edit script notification actions.

Examples:

You can define a script that prints environment variables to a file and a CPU utilization threshold event can trigger the script.

1. Define the following `printenv.sh` script in the `/opt/CA/PerformanceCenter/NotificationScripts/` directory by running the following command:
   ```
   printenv > /opt/CA/PerformanceCenter/NotificationScripts/out.txt
   ```
2. Make the script executable by running the following command:
   ```
   chmod u+x printenv.sh
   ```
3. Configure notifications so that a CPU utilization threshold event triggers the `printenv.sh` script.
   The script creates the following output:
   ```
   CAPM_EventCategory=PERFORMANCE
   CAPM_EvProp__Severity=1
   CAPM_IPAddress=10.253.223.1
   CAPM_ItemNameAlias=cisco2621-10.253.223.1
   CAPM_EvProp_AlarmRuleID=5,313
   CAPM_EvProp_AlarmAggregationMethod=No Aggregation
   CAPM_ItemId=118
   CAPM_EvProp_AlarmProfileName=Test cpu
   CAPM_ItemDesc=Cisco Internetwork Operating System Software ^M
   IOS (tm) C2600 Software (C2600-IK9O3S3-M), Version 12.3(9), RELEASE SOFTWARE (fc2)^M
   Copyright (c) 1986-2004 by cisco Systems, Inc.^M
   Compiled Fri 14-May-04 14:37 by dchih
   CAPM_EventState=OPENED
   CAPM_EventType=ThresholdViolation
   PWD=/opt/CA/PerformanceCenter/EM/bin
   CAPM_EventDesc=A Threshold Violation event has been raised. (Profile Name: Test cpu,
    Rule Name: test cpu)
   CAPM_ItemName=CPU 2
   CAPM_ItemParentId=118
   CAPM_EvProp_ThresholdProfileFolderId=5,311
   CAPM_EvProp_AlarmDuration=60
   CAPM_EvProp_AlarmProfileId=5,314
   CAPM_EvProp_AlarmViolationRuleDetail=Utilization > 50.0
   SHLVL=1
   CAPM_EvProp__Alarm_ID=1500
   CAPM_ItemType=DEVICE
   CAPM_ItemParentName=cisco2621-10.253.223.1
   CAPM_EventSeverity=MAJOR
   CAPM_EvProp_AlarmMetricFamilyName=CPU
   CAPM_EventOccurredOn=Thu May 24 10:39:00 EDT 2018
   CAPM_EvProp_AlarmRuleName=test cpu
   CAPM_EventDataSource=Data Aggregator@fergi04-dev-da
   CAPM_ItemUrl=http://10.237.15.180:8181/pc/desktop/page?
   pg=r&DeviceID=118&timeRange=-1&startTime=2018-05-24+10%3A09+America
   %2FNew_York&endTime=2018-05-24+11%3A09+America%2FNew_York
   CAPM_ItemSubtype=router
   CAPM_EvProp_AlarmWindow=60
   ```

```
CAPM_EventSubType=Raised
_=/usr/bin/printenv
```

NOTE
If an error occurs during the script execution, the errors are logged in the *PC_Install_Directory*/PerformanceCenter/EM/logs/EMService.log file. For example, if the script does not exist or the script is not executable, an error is logged. The exit code of the script is also logged.

You can define a script that prints environment variables to a file and any device life cycle change can trigger the script.

1. Define the following `printenv.sh` script in the `/opt/CA/PerformanceCenter/NotificationScripts/` directory by running the following command:
   ```
   printenv > /opt/CA/PerformanceCenter/NotificationScripts/env.txt
   ```
2. Make the script executable by running the following command:
   ```
   chmod u+x printenv.sh
   ```
3. Configure notifications so that any device life cycle change event triggers the `printenv.sh` script.
   The script creates the following output:
   ```
   CAPM_EventCategory=CONFIG
   CAPM_EvProp__Severity=Unknown
   CAPM_IPAddress=138.42.96.2
   CAPM_ItemNameAlias=138.42.96.2 - alias
   CAPM_EvProp_User=admin
   CAPM_ItemId=110
   CAPM_EvProp_CurrState=RETIRED
   CAPM_EvProp_PrevState=ACTIVE
   CAPM_ItemDesc=RS 38000 - Riverstone Networks, Inc. Firmware Version: 9.4.1.1 PROM
    Version: prom-2.0.1.8
   CAPM_EventState=CLOSED
   CAPM_EventType=LifeCycle
    PWD=/opt/CA/PerformanceCenter/EM/bin
   CAPM_EventDesc=LifeCycle - Change
   CAPM_ItemName=rs38000-96.2
   CAPM_ItemParentId=110
   SHLVL=1
   CAPM_ItemType=DEVICE
   CAPM_ItemParentName=rs38000-96.2
   CAPM_EventSeverity=
   CAPM_EventOccurredOn=Mon Dec 18 16:20:58 EST 2017
   CAPM_EventDataSource=CA Performance Center
   CAPM_ItemUrl=http://10.237.10.219:8181/pc/desktop/page?
   pg=r&DeviceID=110&timeRange=-1&startTime=2017-12-18+15%3A50+America
   %2FNew_York&endTime=2017-12-18+16%3A50+America%2FNew_York
   CAPM_ItemSubtype=router
   CAPM_EventSubType=Change
   _=/usr/bin/printenv
   ```

NOTE
If an error occurs during the script execution, the errors are logged in the *PC_Install_Directory*/PerformanceCenter/EM/logs/EMService.log file. For example, if the script does not exist or the script is not executable, an error is logged. The exit code of the script is also logged.

Configure Notifications

To send a message that is related to events automatically, configure a notification.

Follow these steps:

1. Do one of the following tasks:
   - Hover over **Administration**, and then click **Configuration Settings: Notifications.**
   - Click the name of your user account in the upper-right corner and click **Manage Notifications**.
2. Click **New**.
3. Specify a name and description, and then click **Next**.
4. Select the groups that generate events to trigger the notification, and then click **Next**.
5. Select conditions for the notification, and then click **Next**.
6. Specify the notification actions:
   - **Email**
     Send email notifications.

     NOTE
     To include the device name for events that are triggered on components, use the `Item Parent Name` property.

     TIP
     To create or update a notification email template, select **Save** or **Update Email Template**. Changes to templates do not affect existing messages.
   - **Trap**
     Send trap notifications. Multiple destinations are supported, but the first destination is required.
     Two MIB choices are available in the Notifications wizard to provide compatibility for existing customers.
   - **Script**
     Specify the script file name. All scripts must be executable.
     You can use a script in another language like Python or Perl.

     NOTE
     Store scripts in the `/opt/CA/ PerformanceCenter/NotificationScripts` directory.
7. Click **Next**.
   DX NetOps Performance Management saves the notification and sends messages when the selected conditions occur.

Manage Notifications

Administrators can view, create, or delete notifications from **Administration**, **Configuration Settings: Notifications**. The **Notifications** option only appears when Event Manager is enabled and in the Available synchronized state.

NOTE
As a default tenant administrator, you can work in a real-user context to create a notification for a tenant administrator or tenant user. Log in as a tenant administrator or tenant user. The default tenant administrator can also administrate the tenant, and then create a tenant-scoped notification by proxying to the user.

Users can create email notifications by clicking the name of their user account in the upper-right corner, and then clicking **Manage Notifications**.

Traps Usage

The follow information describes traps usage information in DX NetOps Performance Management:

Get the SNMPengineID

The trap receiver (for example, DX NetOps Spectrum), using SNMPv3, uses the `SNMPengineID` to allow encrypted traps from a specific sender for decryption. Use the following REST endpoint to get the `SNMPengineID` :

URL: `http://<pc_host>:<port>/EventManager/webservice/notifications/engineId`

- **port** is the port for the Event Manager.
  Default: 8281

Method: GET

Return:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<EngineID>

   <ID>
80:00:13:70:01:8a:2a:f8:41
</ID>

</EngineID>
```

The ID string is the `SNMPengineID` .

EventManager Format Usage

The EventManager MIB is supported for trap notifications. If needed, the MIB files can be found in:

```
InstallLocation/PerformanceCenter/PC/webapps/pc/mibs/netqos-em-mib
```

- ***InstallLocation***
  Is the directory where CA NetOps Portal was installed.

When the EventManager format choice is selected, the trap will be sent out with the following variables:

- **netQosEventId**
  Specifies an identifier that Event Manager assigned to the event.
- **netQoSEventType**
  Specifies the type of event.
- **netQoSEventCategory**
  Categorizes the event.
  Values: 0 Unknown, 1 Fault, 2 Config, 3 Accounting, 4 Performance, 5 Security
- **netQoSEventSeverity**
  Specifies the severity of the event.
  Values: 0 Normal, 1 Unknown, 2 Minor, 3 Major, 4 Critical, 5 Unavailable
- **netQoSEventDescription**
  Describes the event.
- **netQoSEventState**

Specifies the current state of the event. Each state has its own notification.
Values: 0 opened, 1 acknowledged, 2 closed, 3 cleared

- **netQoSEventOpenTime**
  Specifies the UTC timestamp (from the eventState timestamp).
- **netQoSEventMapURL**
  No value is available. The "" string will be sent.
- **netQoSEventDetailsURL**
  No value is available. The "" string will be sent.
- **netQoSEventAssociatedItemURL**
  Specifies the URL to the item web page.
- **netQoSEventItemName**
  Specifies the item name. There is one notification per item.
  Maximum length: 127 bytes
- **netQoSEventItemType**
  Specifies the item type.
  Maximum length: 32 bytes
- **netQoSEventItemSubtype**
  Specifies the item subtype.
  Maximum length: 32 bytes
- **netQoSEventItemIpAddress**
  Specifies an IP address for the item or an empty string.
- **netQoSEventPropertyName**
  Specifies one name set for each property. A PropertyName exists for each property in the event. (The properties vary by the event type.)
  Maximum length: 128 bytes
- **netQoSEventPropertyValue**
  Specifies the property value for the event. A PropertyValue exists for each property in the event. (The properties vary by the event type.)

nhLiveAlarm Format Usage

The nhLiveAlarm MIB is supported for trap notifications. If needed, the MIB files can be found in:

```
InstallLocation/PerformanceCenter/PC/webapps/pc/mibs/concord-diagmon.mib
```

- ***InstallLocation***
  Is the directory where CA NetOps Portal was installed.

When using the nhLiveAlarm format for trap notifications, be aware of the following restrictions. Many of the variable values described by the CA eHealth trap MIB have changed from integrations with earlier versions of CA PC.

- **nhServerIp**
  No value is available. The "" string is sent.
- **nhServerName**
  No value is available. The "" string is sent.
- **nhServerPort**
  No value is available. The "" string is sent.
- **nhElementIp**
  Specifies the IP address of the item or "" if no IP address exists.
- **nhElementName**
  Specifies the item name.
- **nhElementId**

Specifies the item CA PC ID (global ID).
- **nhStartTime**
  Specifies the timestamp from the event.
- **nhDisplayStr**
  Specifies the value for the MaxThresholdValue variable from the event.
- **nhGroup**
  No value is available. The "" string is sent.
- **nhGroupList**
  No value is available. The "" string is sent.
- **nhExceptionType**
  No value is available. The "" string is sent.
- **nhVariable**
  Specifies variables in the event profile rule.
- **nhSeverity**
  Specifies the severity of the event.
- **nhOpenViewSeverity**
  No value is available. The "" string is sent.
- **nhProfile**
  Specifies the event profile name.
- **nhExceptionId**
  Specifies the event ID.
- **nhTechType**
  No value is available. The "" string is sent.
- **nhEventCarrier**
  No value is available. The "" string is sent.
- **nhElementAlias**
  No value is available. The "" string is sent.
- **nhComponent**
  No value is available. The "" string is sent.
- **nhDescription**
  Contains the event description.
- **nhAlarmOccurId**
  Specifies the alarm ID.
- **profileId**
  Specifies the event profile ID.
- **nhElementBaseType**
  Specifies the item type.

Administrating

Administrating includes information about monitoring and maintaining the system components, managing users, roles, and tenants, and system configuration options. This section also includes information about the APIs for the product.

For information about how to configure data collection, see Building.

Onboard a New Product Operator

As an administrator, you want to give a coworker with a unique role in the IT organization permissions to use the product.

The administrator creates a custom user account for each person who uses the product. Usually, predefined menus and roles are assigned to the new user accounts. However, to create a new user with a unique organizational role, custom menus, and roles are also required.

Custom user accounts are best deployed in a well-planned system that includes custom groups. Custom groups are assigned as permissions to let product operators view only the data, menus, and dashboards that they use to perform their daily tasks. Before you onboard a new product operator, set up custom groups for your organization.

Log in to Test the New User Account

To test a new user account, log in to the account. The Proxy feature lets you test user accounts while logged in as the administrator. However, user account proxying does not let you test roles or role rights.

Follow these steps:

1. Log in using the username and password that you assigned to the new user account.
2. Click the Inventory tab.
3. Click item links to verify that the user can see monitored items in the inventory.
4. Select a few dashboards, and verify that the views are populated with data.
5. Test your ability to select a new group or item context for a dashboard by taking the following steps:
   a. Click the [change] link above the time period selectors.
      A dialog opens with filtering options.
   b. Click to select another managed item.
   c. Expand nodes in the Groups tree to select a group context.
   d. Click OK.
6. Verify that the user can see data from the new item or group.
7. Test any special role rights that you assigned to the user, such as saving changes to views.
8. Log out when you are satisfied that the user account meets the requirements of the intended operator.

You are now ready to contact the new product operator and provide the username and password.

Set up a Unique Operator

As an administrator, you want to give a coworker in the IT organization permissions to use the product. You want to create a custom menu for this new operator to reflect this person's organizational responsibilities.

Onboarding a unique operator involves creating a menu and populating the menu with either predefined or custom dashboards. You also create a custom role for the new user, and assign the role to a new user account.

Finally, you log in to the new user account to test account permissions and determine whether adjustments are required.

Add a Menu for a New Product Operator

Custom menus let you organize dashboards and make them available to users with selected roles. Administrators and designers can create custom menus, and can select dashboards for each menu. A custom menu is available to any whose role to includes the menu.

Create a Custom Dashboard for a New Operator

Users with the necessary role right can create a custom dashboard. They can select views for the dashboard and their location on the page. They can also select the menus in which the dashboard appears so that it can be shared with other operators.

The views in a custom dashboard can also be customized. For example, you can select a group context, or you can specify a custom view title.

You can customize the predefined dashboard pages, or you can add new dashboards. You can select the views and data context for custom dashboards.

Create a Dashboard

Follow these steps:

1. Log in as a user with the required administrative role rights.
2. Click the **Dashboards** tab.
   A list of available dashboards appears. Each view on the page corresponds to a menu.
3. Click **Add Dashboard** next to the menu where you want the new dashboard to appear.
   The Add Dashboard page opens.
4. Complete the following fields:
   - **Dashboard Menu**
     The menu where you want the dashboard to appear.
   - **Menu Item**
     The name of the dashboard as you want it to appear in the menu.
   - **Dashboard Title**
     The name that you want to appear at the top of the new dashboard.
5. (Optional) Select a layout template for the dashboard.
   Each layout treats the page as a table with rows and columns for views. The Layout buttons indicate the number of views in each column and row on the page. We recommend selecting a layout before adding views.
6. (Optional) Apply a group or context filter to the views. Views with a selected context always display data for that context; they do not inherit the context of the dashboard.
   For example, if you set the context filter to Group A and add a view to the dashboard, that view will always display data for Group A, even after you change the dashboard context to Group B.

   NOTE

   By default, the context is Summary. With the Summary context setting, the available views display summary data for the current group context of the dashboard. The Summary setting does not require you to select a specific group or item. Summary views dynamically update the context when you change the context of the page.

Select a Context

Follow these steps:

1. Click **Select Context**.
2. Select a Context Type, such as a type of managed item. Select **Group** to see the Groups tree.
   By default, the list is filtered to show only items and item types to which you have access. For example, if you are not monitoring any servers, the Context Type list does not include the Servers option. Select **Show All Context Types** to see all context options.
3. Select a specific context item or a group context.
4. Click **OK**.
   The new context filter is saved.
   - Expand the categories of views in the left pane. Select the 'Display suppressed views in list' option only if you want to see views from data sources that you have not registered. This check box disables View Suppression.
   - Select a view that you want to add to the page from one of the expanded lists.

Select a View

Follow these steps:

1. Click and drag a view to the page layout, and drop it where you want it to appear.
2. (Optional) Click **Revert** to discard your changes.
   The layout returns to the settings that you last saved.
3. Click **Save**.
   The dashboard is saved, and is added to the selected menu.
   The dashboard page refreshes to reflect your changes. The changes persist across login sessions.

   NOTE

   : The maximum number of views per dashboard is 25.

Edit the Menu to Add the New Dashboard

Administrators and designers can customize menus to meet the requirements of each operator. When you edit custom or factory menus, you can add new dashboards, remove dashboards, and change the order of the dashboards in the menu.

Add a Custom Role

You can add a custom user role for each product operator. The user account roles let product operators perform their job responsibilities. Assign new roles to any menus that you have customized for the intended product operator. Roles are disabled until they are assigned to user accounts.

Follow these steps:

1. Log in as a user with the required administrative role rights.
2. Select **Administration**, **User Settings**, and click **Roles**.
   The Manage Roles page displays the current list of roles.
3. Click **New**.
   The Add Role dialog opens.
4. Enter the required information and make selections in the fields provided. The default value for Role Status is Enabled.
   A table indicates that no role rights have been selected for the role.
5. Select **Menu Set**, and click **Edit**.
   The Edit Role dialog opens.
6. Select the new menu in the **Available Rights** list, and click the right arrow.
   The menu moves to the **Selected Rights** list.
7. Select **NetOps Portal** , and click **Edit**.
   The Edit Role Rights dialog opens. Role rights that are listed under **Available Rights** can be added to the role.
8. Click an item on the left that you want to add to the role, and then click the right arrow.
   Use Shift + Click or Ctrl + Click to select multiple items in the list.
   The selected item moves from the **Available Rights** list to the **Selected Rights** list.
9. (Optional) Use the Up and Down arrows to move items around in the list. The order of role rights determines their priority when rights overlap.
10. Click **Save**.
    The Add Role page appears.
11. Click **Save**.
    The new role is created and appears in the Role List.

Add a Custom User Account

Add a user account for each operator. For security purposes, do not share user accounts with multiple people.

NOTE
You can create user accounts with basic parameters, and then edit them as a separate step to assign permissions. This workflow lets you carefully consider the groups that each operator must access.

Follow these steps:

1. Log in as a user with the required administrative role rights.
2. Navigate to the Manage Users page.
3. Click New.
   The Add User wizard opens.
4. Specify information for the account parameters.
   The Authentication Type field identifies the authentication method that applies to this user account. The method must match Single Sign-On configuration. Select one of the following options:
   – **NetOps Portal:** The default authentication scheme
   – **External:** A third-party authentication scheme, such as LDAP or SAML
5. Advance the wizard to the Permission Groups page.
6. Assign permissions to the user account.
7. Advance the wizard to the Product Privileges pages.
8. Click Save.
   The new user account appears on the Manage Users page.

Assign Permissions to the User Account

Individual operators require data access permissions to monitor data, which are based on groups. You can assign access permissions according to your plan for custom groups.

NOTE

Do not assign the 'Collections' group as part of a user's access permissions. Do not use this group for reporting.

To assign permissions, edit user accounts. Make sure that all operators see only the data that they require for their role.

Follow these steps:

1. Log in as a user with administrative privileges.
2. Select **Administration**, **User Settings**, and click **Users**.
   The Manage Users page opens.
3. Select a user account that you want to change, and click **Edit**.
   The Add User wizard opens.
4. Click **Access Permissions**.
   The Access Permissions page appears.
5. Add permission groups to the user account, as follows:
   a. Expand the groups in the **Available** groups tree in the left pane.
   b. Select a group or subgroup.
   c. Click the right arrow button to add it to **Selected** groups tree on the right.
6. Select a group from the drop-down list.
   By default, My Assigned Groups is selected.
   When the user logs in, data from the default group appears in dashboards by default.
7. Click **Save**.

Manage Data Sources

Data sources are the products and components that provide performance and configuration data.

The Data Aggregator data source administration is integrated into NetOps Portal. CA Application Delivery Analysis, Network Flow Analysis, and CA Unified Communications Monitor collect and aggregate data independently. However, once these data sources are registered in NetOps Portal, DX NetOps Performance Management controls many administrative functions. DX NetOps Performance Management does not administer DX NetOps Spectrum.

Synchronization status is included in the table on the Manage Data Sources page. Failures and detailed statuses are included in the Data Source Log.

The System Health icon at the top of the page indicates when a synchronization failure occurs. To view more information about the failure, click the icon. Review the information in the Global Synchronization Status section and the Data Sources section. For more information, see Synchronize Data Sources.

Maximum Data Source Instances Supported

The following limits apply to the number of data sources for each type that you can register:

- 10 CA Application Delivery Analysis data sources
- 10 Network Flow Analysis data sources
- 4 CA Unified Communications Monitor data sources
- 1 CA Application Performance Management data source
- 1 CA Catalyst Connector data source
- 1 Event Manager data source
- 1 DX NetOps Spectrum data source
- 1 CA Business Intelligence data source

Configuration Data from Data Sources

For CA Application Delivery Analysis, Network Flow Analysis, and CA Unified Communications Monitor, the monitoring parameters, user accounts, and other definitions are shared with DX NetOps Performance Management and other registered data sources after registration.

During registration, DX NetOps Performance Management imports shared monitoring data, such as user accounts, SNMP profiles, and other administrative data from the data sources. DX NetOps Performance Management resolves conflicts and eliminates duplication. At the next synchronization, DX NetOps Performance Management sends updated administrative data to registered data sources.

The registration process includes a "binding" step that prevents further modifications to shared administrative data in individual data sources. As a result, the data source administrator can only modify shared monitoring parameters after registration.

For user accounts and SNMP profiles, NetOps Portal uses the following processes to handle conflicts or duplicates:

Redundant User Accounts

A user can have two different accounts with the same name in different data source products. The resulting user account retains the password of the first account that is synchronized. The unique role rights and permissions from other accounts are added to the account as you register more data sources.

While multiple user accounts sometimes share a username in different data sources, some account parameters differ. Manual editing is required in this situation.

Example

A user who is named Robert uses Network Flow Analysis, and a different user, also named Robert, uses CA Application Delivery Analysis. In this case, DX NetOps Performance Management creates one account that is named Robert. The role

rights and permissions from both data sources are merged into a single new account. To preserve the distinct role rights of the two accounts, create an account with a unique username.

Redundant SNMP Profiles

Registering a data source that contains SNMP profile definitions automatically adds the profiles to DX NetOps Performance Management. The profiles are distributed to the other registered data sources during the next synchronization.

When a data source is added, DX NetOps Performance Management minimizes duplication of SNMP profiles by comparing the following values to existing profiles:

- User for SNMPv3
- Community String for SNMPv1 and SNMPv2

If the compared values match, the SNMP profile with the most recent timestamp is retained.

If the compared values do not match, DX NetOps Performance Management appends a number to duplicate profile names. For example, the first profile that is named Boston remains Boston. The second profile becomes Boston(1).

Synchronization with DX NetOps Spectrum

After an integration with DX NetOps Spectrum is configured, consider the following guideline whenever you restore the SpectroSERVER or OneClick database:

- If DX NetOps Spectrum exists as a data source in NetOps Portal, whenever you restore another data source, restart Tomcat.

  WARNING
  You must follow the correct procedures to enable data to synchronize correctly between DX NetOps Spectrum and NetOps Portal. For the correct procedures, see Integrate CA Spectrum with CA Performance Management .

Configure a Data Source

NetOps Portal represents each source of information as a registered data source. To load performance data, configure the data sources.

These procedures require the Administrator role:

Register a Data Source

To view collected data on the NetOps Portal dashboards, register your data sources.

Follow these steps:

1. Go to **Administration**, **Data Sources**, **Data Sources**.
2. Click **Add**.
3. Select the type of data source from the **Source Type** list.
   This list includes all possible data sources. The list does not show data sources that already have reached the registration limit.
4. (Optional) To delay synchronization with the data source, select **Disabled** for the **Status**.
5. Specify the hostname and port for the data source:
   - For the Data Aggregator data source, specify the IP address or hostname of the Data Aggregator host.
   - For other data sources, specify the IP address or hostname of the management console.
6. Select the communication protocol. For secure communication, select **https**.

NOTE

: Before you select HTTPS, verify that your system is configured for SSL. Single Sign-On supports secure communications between NetOps Portal and the data source products. For more information, see Single Sign-On .

7. (Optional) Specify a display name for the data source.
   By default, the data source type and the hostname are combined to create the display name. You can specify another name here. For example, instead of NetworkFlowAnalysis@192.168.10.22, you could name the data source NFA_NewYork.
8. (Optional) Select **Contribute inventory to the Data Aggregator**. If selected, all new data source inventory contributes to the Data Aggregator. If unselected, the existing inventory remains the same, and no new data source inventory contributes to the Data Aggregator.
   Automatic synchronization includes only devices that are discovered after you select this option. To discover previously discovered devices, perform a full synchronization of the Data Aggregator. DX NetOps Performance Management creates a discovery profile that includes the IP addresses of the devices. This discovery profile attempts discovery once per day.
9. (Optional) If the Web Console is on a different system than the data source, clear the **Same as data source** check box. Specify another hostname and port for the data source console.

   NOTE
   Use this parameter in cases when network address translation is deployed.
10. (Data Aggregator only) Configure the Data Aggregator settings:
    – **Synchronize component items that are not currently present on the monitored device**
    When the Data Aggregator finds a device component that is no longer present in the environment, that status of the component is set to **Not Present**. By default, the Data Aggregator does not synchronize these items because data can no longer be collected for the item. Historical data that has not reached the data retention limit is still available for these items.

      WARNING

      If the properties of an active component match the identifying properties of the not present component, the components are indistinguishable. Data from the old component might contribute to group based dashboards instead of data from the active component. Enable this feature only under the following circumstances:

      - You want to report on historical data for items that are no longer present in the environment.
      - You can ensure that the not present item does not conflict with an actively monitored item.
      - You can identify the not present items so that you can exclude the items from groups.
11. (DX NetOps Spectrum only) To enable DX NetOps Spectrum to control the life cycle state of items in DX NetOps Performance Management, select **Synchronize device life cycle state from Spectrum**. For more information, see Manage Device Life Cycles.
12. Click **Save**.
    If the data source is enabled, data appears in NetOps Portal after the next synchronization.

Test Data Source Connections

If data source registration does not complete successfully, use the test feature to determine the reason for the failure. The test validates version compatibility, and verifies that the data source is not registered with a different instance of the NetOps Portal.

To confirm the proper registration and connection of a data source, select the data source, and click **Test** .

If the test fails, verify that the server name or IP address is accurate for the source type. For more information, see Data Source Test Fails.

Edit a Data Source

When you edit a data source, changes apply to the system after the next synchronization.

Delete a Data Source

WARNING

When you delete a data source, the deletion does not remove everything from the system. If you register the same data source, unexpected behavior might occur. To remove a data source temporarily, edit the data source, and set the status to disabled.

A data source that you delete can be registered to another NetOps Portal instance.

Only users with the Administrator role and the Delete Data Sources role right can delete a data source. This role right is not granted by default. To delete a data source, assign this role right to the Administrator role.

NOTE

If View Suppression is enabled, views that are associated with the deleted data source are suppressed. As a result, deleting a data source might cause some menus and dashboards to become unavailable. A dashboard, context tab, or custom menu must contain at least one view that is not suppressed. Otherwise, the menu item does not appear in NetOps Portal.

Synchronize Data Sources

NetOps Portal periodically synchronizes with registered data sources to send configuration information and to retrieve data. NetOps Portal incrementally replicates information to the data sources. Data sources receive group configuration, authentication settings, SNMP profiles, users, and roles. The data sources send inventory to NetOps Portal. NetOps Portal uses the inventory data to request performance metrics from the data sources.

To determine whether items from multiple data sources are the same item, NetOps Portal runs global synchronization. During global synchronization, NetOps Portal reconciles devices and interfaces.

Devices are reconciled in the following scenarios:

NOTE

The primary IP address is the IP address that DX NetOps Performance Management uses to monitor a device. When a device is first discovered with the IP ranges discovery profile, DX NetOps Performance Management tries to use the IP address that maps to the hostname as the primary IP address.

- The primary IP addresses match.
- The name of a device without an IP address matches another device regardless of whether the other device has an IP address.
- The primary IP address for a new device is in the IP address list of an existing device, and the primary IP address of the existing device is in the IP address list of the new device.

After devices are consolidated, interfaces are reconciled in the following scenario:

- The parent device and the `ifIndex` match.

Reconciled items are a single item in NetOps Portal. If the reconciled item has different properties in different data sources, NetOps Portal uses the value from the data source with the highest priority. The Data Aggregator has the highest priority by default. First, the item type priority determines priority. Then the data source priority determines priority. Last, the data source assigned `SourceID` determines priority.

Incremental synchronization occurs 5 minutes after the previous synchronization completes. Incremental synchronization also occurs each time an SNMP profile is created. Only the records that are new or changed as of the last synchronization timestamp are included. The Last Polled On column shows the completion time of the last successful

synchronization for each data source. If this timestamp shows a time more than 5-10 minutes, synchronization for that data source has not completed successfully. Changes might not be reflected in the system.

A full synchronization occurs when a data source is first registered to NetOps Portal. NetOps Portal receives information about all managed items in that data source. This type of synchronization does not recur automatically. After the initial data source registration, this type of synchronization is not typically required.

These procedures require the Administrator role:

Synchronize a Data Source

Synchronization occurs regularly every 5 minutes with all registered data sources. To propagate a configuration change immediately, manually run synchronization. For example, if you add a group, you can send the change to the data sources immediately. If synchronization is in progress, the process is not interrupted and new changes are not applied until the next synchronization cycle. A new synchronization begins immediately once the in-progress synchronization is complete.

WARNING

Initiate full synchronization, only when requested by CA Support.

Follow these steps:

1. Go to **Administration**, **Data Sources**, **Data Sources**.
2. Select the data source.
3. Click **Resync**.
4. (Optional) Select **Perform a full resynchronization**.
5. In the dialog, click **Resync**.NetOps Portal collects inventory data and sends configuration changes to the selected data source. Only the records that are new as of the last synchronization timestamp are included.

View Synchronization Status

Synchronization status is included in the table on the Manage Data Sources page. Failures and detailed statuses are included in the Data Source Log.

The System Health icon at the top of the page indicates when a synchronization failure occurs. To view more information about the failure, click the icon. Review the information in the Global Synchronization Status section and the Data Sources section.

WARNING
If the Last Run Status in the Global Synchronization Status section displays Failed, contact CA Support.

The Data Sources section displays the status of all registered data sources. The following messages describe possible data source status conditions:

- **Awaiting Poll**
  The data source has never been contacted, and is waiting for the Device Manager to poll it. The data source is polled quickly unless the Device Manager is busy performing another poll.
- **Awaiting Bind**
  Inventory data has been retrieved from the data source. The data source is waiting for NetOps Portal to transmit configuration information, and to lock corresponding administrative features.
- **Available**
  The data source is available for reporting. Registration has succeeded.
- **Polling**
  The Device Manager is in the process of retrieving inventory data.
- **Registering**
  The Device Manager is in the process of registering the data source.
- **Binding**

The device manager is locking the users, roles, and groups that are defined in the data source. For some data sources, binding prevents further changes to configuration within the data source.

- **Synchronizing**
  The device manager is in the process of synchronizing with the data source.
- **Polling Failure**
  A failure occurred during polling.
- **Synchronization Failure**
  A failure occurred during synchronization.
- **Registration Failure**
  A failure occurred during registration.
- **Bind Failure**
  A failure occurred during the binding of users, groups, and roles.
- **Unable to Contact**
  NetOps Portal is unable to contact the data source due to communication problems.
- **Version Incompatible**
  The version of the data source is not compatible with NetOps Portal.
- **Requires Upgrade**
  The data source requires a software upgrade. Contact CA Support.
- **Requires Registration**
  The data source is waiting for registration.
- **Requires Migration**
  The data source requires migration and is waiting for the Device Manager.
- **Under Maintenance**
  The data source is under maintenance.
- **Disabled**
  The administrator has disabled the data source.

View the Data Source Log

Only the initial synchronization and any failures that occur during subsequent synchronizations are recorded in the Data Source Log. To determine when the last synchronization occurred, view the Last Polled On date on the Manage Data Sources page.

Use the Data Source Log to investigate suspected errors with the data source synchronization. Use this information to find events in the device manager logs. For more information, see Logs.

Follow these steps:

1. Go to **Administration**, **Data Sources**, **Data Sources**.
2. Select the data source.
3. Click **Log**.
   The Data Source Log page opens. The log shows only events that are related to synchronization for the selected data source.

Manage Roles and User Accounts

Manage user access with rights assigned to roles. Then refine access by assigning roles, access permissions, administer groups, and product privileges to each specific user account.

Roles

Roles are assigned to user accounts to control user access to the product features and dashboard pages. Based on job functions, roles grant administrative access to product configuration using role rights. Roles let users access data and product features that they require to perform their duties. Roles restrict access to features that they do not require.

Roles are shared with registered data sources. Roles determine what users can access in the data source interface when following a drilldown path to a data source. When you add a user, you select a role for the user account. You can edit roles to include new role rights. You can also disable roles to prevent users with those role assignments from using NetOps Portal. A set of predefined, or "factory," roles help you to add new users quickly while determining the required customizations.

User Accounts

Custom user accounts let operators view the data, menus, and dashboards that they require to perform their daily tasks. Operators with the administrator role rights can create user accounts and can manage existing accounts. Tenant administrators can manage user accounts only for their own tenant.

Before you create or edit user accounts, create the custom groups and roles that you require. Groups and roles are among the required parameters for each user account.

Create and Configure a User Account

Place managed items in custom groups before creating user accounts. Assign custom groups to user accounts as "permission groups," which determine the data that each user can view. You can grant selected ownership of a single branch of the Groups tree to a user account with administered groups.

Create any custom roles that you require before creating user accounts. Typically, the predefined roles provide starting points for customization.

We recommend the following process for creating a user account:

1. Log in as an administrator user.
2. Confirm that the appropriate groups exist, or create them if necessary.

   NOTE
   User account parameters include all the groups that the user can *view.* They also include one or more groups that the user can *manage*. The Administer Groups role right lets users without full administrative rights manage a specific branch of the Groups tree.
3. Confirm that the appropriate roles exist, or create them if necessary.
4. Add a user, and specify its user information.
5. Assign a role.
6. Assign permission groups.

   NOTE
   New user accounts have access to no groups by default. Their dashboards contain no data until you assign at least one permission group.
7. Assign group ownership so that the user can create and modify groups in one branch of the Groups tree.

   NOTE
   Only user accounts with the Administer Groups role right are eligible for this selective group ownership.
8. Assign product privileges to grant access to the data sources you have registered.
9. Test the user account by temporarily proxying it.

New User Account Example

To understand user account parameters, consider an example. A Data Center Manager at your company is responsible for data centers, staff, and infrastructure in the Southwest region.

Follow these steps:

1. Create a group named 'Southwest'.
2. Add managed items to this group.
   Include all the routers, switches, applications, and servers that comprise the Southwest region.
3. Create a custom role that includes the product features and menus that the Data Center Manager requires.
   The Data Center Manager is not a network engineer. This user does not drill in to detailed data in the data sources. To manage the team, this user wants to create dashboards and assign them to the roles of the team members.
   This user only wants to see menus containing high-level Management and Operations dashboards.
4. Add the user account.
5. When adding the user account, select the following items:
   - The custom 'Data Center Manager' role
   - The permission groups containing the managed items within the Southwest group that the Data Center Manager wants to monitor.

Role Rights

Role rights determine the types of views that users can see, the administrative features that they can change, and whether they can export data.

Administrators can grant rights to users by editing their role. The Edit Role dialog lists role rights that are currently assigned to roles. The Manage Users page shows the role that is assigned to each user.

Role rights also include menus. You can grant access to selected custom and predefined menus by editing role rights.

> **NOTE**
>
> Do not remove the administrative role rights from your primary administrator account. Administrative access to the console is required.

Administrative Role Rights

The following role rights give users access to administrative features. Limit the number of users with these role rights for increased security.

> **WARNING**
>
> Several role rights are limited to the Administrator role. Copying the Administrator role does not give the same role rights to the new role.

- **Create DA Threshold Profiles**
  Lets users define and configure threshold profiles. Users can only edit the profiles that they created. Unlike the Administer DA Threshold Profiles role right, this role right does not allow users to configure profiles that other users created, or to transfer ownership of profiles.
- **Administer Data Sources**
  Lets users register new data sources, test data source connections, view data source status, change data source parameters, and remove data sources. Also lets users view the data source log.
- **Administer Groups Owned by You**
  Lets users without full administrative rights manage a specific branch of the Groups tree. With this role right, users can create, change, and delete groups only in the specified branch. The Administrator role and the Tenant Administrator role have this role right by default, allowing administration of All Groups and the Tenant root group, respectively.

Only the Administrator and the owner (creator) of the groups in the administered branch can delete and modify groups in that branch. When an administered group is a child of another group, the administered group is deleted when the parent group is deleted. Administered groups are not deleted when the user account of the owner is deleted.

NOTE

Assign this role right to users who should not have full administrator rights to the Groups tree, but instead require limited, branch-specific administrator rights. In some organizations, this user is a 'power user' or a 'super user.'

Do not confuse the 'Administer Groups Owned by You' role right with the 'My Custom Groups' feature, which is simply a tool that lets users organize the groups to which an administrator has granted them access. 'My Custom Groups' does not provide administrative rights to a specific branch of the Groups tree.

- **Administer Groups Owned by You and Others**
  Lets users without full administrative rights manage the branch of the Groups tree that they have the rights to administer. With this role right, users can create, change, and delete groups in the branch that they or other users created. The Administrator role and the Tenant Administrator role have this role right by default, allowing administration of All Groups and the Tenant root group, respectively.
- **Administer IP Domains**
  Lets the user manage IP domains.
- **Administer Life Cycle**
  Lets the user change the state of devices.
- **Administer Maintenance Indicators**
  Lets users manage maintenance indicators.
- **Administer Menus**
  Lets users create, edit, and delete menus. This role right is required to assign new dashboards to menus. To assign menus to user accounts, the 'Administer Roles' role right is required.
- **Administer Roles**
  Lets users create, edit, and delete user account roles. Lets users assign new menus to user accounts by editing roles.
- **Administer Shared Dashboards**
  Lets users manage their own dashboards and the dashboards of other users. Users with this role right can edit an existing dashboard page and can save changes that are visible to other users.
  – To create a dashboard, the 'Create a Dashboard' role right is required.
  – To assign a dashboard to a menu, the 'Administer Menus' role right is required.
- **Administer SNMP Profiles**
  Lets the user manage SNMP profiles.
- **Administer Tenants**
  Grants users administrative rights over the tenants that are selected in the user wizard. Users with this role have the rights to administer certain tenants, but have limited access to the default tenant. This role is only used in multi-tenant environments. Tenant administration includes the ability to manage:
  – Users
  – Menus
  – Dashboards
  – Views
- **Administer Users**
  Lets users create, edit, and delete user accounts. Lets users assign new roles to user accounts.
- **Create DA Threshold Profiles**
  Lets users define and configure threshold profiles. Users can only edit the profiles that they created. Unlike the Administer DA Threshold Profiles role right, this role right does not allow users to configure profiles that other users created, or to transfer ownership of profiles.
- **Create a Dashboard**

Lets users create new dashboards and populate them with views. Other users cannot see these dashboards. To create dashboards for other users, the 'Administer Shared Dashboards' role right is required.

- **Create Notifications**
  Lets users configure email notifications using the Create/Edit Notifications wizard from the Administration, Notifications menu. Notifications are not supported for all data sources.

  NOTE

  To create notifications, the user also requires access to the Event Manager data source.
- **Create On-Demand Report Templates**
  Lets users create, edit, and delete on-demand report templates. This role right is always assigned together with the Run On-Demand Report Templates right. Users can save on-demand report templates at the user level, which allows only the user to view the templates.
- **Save On-Demand Report Templates for All Users**
  Lets the user save On-Demand report templates that are visible to all users. This role right is always assigned together with the Create On-Demand Report Templates and Run On-Demand Report Templates right.
- **Delete Data Sources**
  Lets a user with the Administrator role delete (unregister) a data source. This role right is not assigned to any user or role by default, and can only be assigned to the Administrator role.
- **Drill from Views into DA Admin Page**
  Lets a user access the Data Aggregator administrator page directly from a page that is associated with the Data Aggregator. For this role right to work properly, the user must also have the Administer Data Sources right. The ability to access the Data Aggregator administrator page is limited to views for Data Aggregator devices, interfaces, and components. Selecting a Data Aggregator interface or component causes the administrator page for the associated parent device to appear when clicking the gear button and Device Admin.
- **Edit Context Pages**
  Lets users edit, delete, add, or reorder tabs on context pages. A context is a managed item, such as a device, router, switch, or an interface. A context page resembles a dashboard with a fixed context. Only the Designer and Administrator roles have this right by default.
- **Modify Device Alias**
  Lets users modify the alias property for devices.
- **Modify Interface Alias**
  Lets users modify the alias property for interfaces.
- **Modify Device IP Address**
  Lets users modify the IP address property for devices.
- **Modify Interface Speed Overrides**
  Lets users modify the speed override properties for interfaces.
- **Proxy Users**
  Lets users log in as a selected user to view and verify user account settings.
- **Save Changes to Shared Views**
  Lets users save edits that they have made to the views on a shared page. Other users who can see these views can see the changes if they are applied as a 'Default for All Users'. The changes can also be saved to the user account so that they persist after logout.
- **SNMP Clear Text**
  Lets users troubleshoot SNMP profiles and view security information that is typically masked in clear text.

Role Rights for Dashboard and View Access

The following role rights give users access to reporting features. Most user accounts require these rights.

- **Drill into Data Sources**
  Lets users navigate to the data source interface during drilldown to see detailed data from a selected item.
- **Drill into Views**

Lets users drill in to a context view to see detailed data from a selected item. This role right is required to enable the 'Edit Context Pages' role right.

- **Edit Shared Views**
  Lets users edit the views on a shared page. Other users can see these views, but cannot see the changes. The changes can only be applied to the current login session or saved to the current user account.
- **Edit Time Zone**
  Lets users edit their own time zone setting for data that are displayed in dashboards.
- **Run On-Demand Report Templates**
  Lets users run on-demand report templates. This role right is always given together with the Create On-Demand Report Templates right. However, if the Create On-Demand Report Templates right is taken away, users do not lose the ability to edit and delete their on-demand dashboards. Users who have this right without the Create On-Demand Templates right can run on-demand report templates on the tenant level.
- **Run Dashboards at Higher Resolution**
  Lets users select higher resolutions when viewing dashboards. No roles are given this role right by default. Users with this role right can set and save the resolution to higher values than are typically allowed when reporting for longer time ranges. When users save the higher resolution at the tenant level, it is only visible to users with this role right. To view the current report resolution settings, select **Administration**, **Report Resolution**. The resolution settings cannot be modified on the Report Resolution page.
  The following resolutions apply when a user has the Run Dashboards at Higher Resolution role right:
  – **As Polled Data** Less than, or equal to, 31 days.
  – **Hourly Roll-Up** More than 31 days.
  – **Daily Roll-Up** More than 3 months.
  The following resolutions apply to a user without the Run Dashboards at Higher Resolution role right:
  – **As Polled Data** Less than 24 hours.
  – **Hourly Roll-Up** Less than 30 days, and more than, or equal to, 24 hours.
  – **Daily Roll-Up** More than, or equal to, 30 days.
- **View Conversations**
  Lets users see specific client conversations.
- **View Groups Change Log**
  Lets users see the table view that shows changes that are made to groups.
- **View Hosts**
  Lets users see specific client host information.
- **View Item Display Name or Name Alias**
  Lets users see the display names or the aliases for items.

  NOTE
  Users who are given this role right can select the name that appears in their dashboards and views in the My Settings, Display Settings menu item.
- **View Item Name Alias Only**
  Lets users see only the aliases for items.
- **View Inventory and Search**
  Determines whether users can access the Inventory tab and Search field to find items.
- **View Protocols**
  Lets users view protocol information, where available.
- **View System Health Dashboards**
  Lets users view system health dashboards that show information about the performance of DX NetOps Performance Management.
- **View ToS**
  Lets users view the Type of Service information in applicable views.

Role Rights to Export and Print

The following role rights allow users to export dashboard data in various formats:

- **Export to CSV**
  Lets users export the contents of a selected view to a file in comma-separated values (CSV) format.
- **Generate URLs for views**
  Lets users share views externally with a URL.
- **Print a Dashboard**
  Lets users export a dashboard page as a PDF, and to send it to a selected printer. This role right also lets users export a dashboard page to a CSV file.
- **Send Reports by Email**
  Lets users export dashboards as reports, and to send them to other users in email messages from the console.
- **Send Reports on a Schedule**
  Lets users set up schedules to export dashboards as reports, and to send them by email on a recurring basis automatically.

  NOTE
  This right also requires the 'Send Reports by Email' role right.

Data Source Role Rights

Each registered data source has its own set of roles with unique rights that let users access features and data within that interface. Administrators can assign rights for a role within that data source. These data source rights apply when users follow a drilldown path from a data view to that particular data source. However, any rights that are granted in this manner are specific to a data source instance. For example, if more than one CA Application Delivery Analysis (CA ADA) data source is registered, the rights for each management console are managed separately.

For example, an administrator can grant the right to generate reports in a CA Network Flow Analysis (CA NFA) data source, but withhold the right to edit dashboards in CA NetOps Portal. The individual data source Administrator Guides provide detailed information about the application of role rights.

Individual data source administrators can create user accounts and grant users role rights to access features within that data source. After registration, those rights are synchronized with CA NetOps Portal. The rights are then displayed in NetOps Portal when data source administrators edit a role.

NOTE

Role rights to individual data sources are distinct from rights to access DX NetOps Performance Management features, but they frequently have the same names.

Data Aggregator Role Rights

- **Drill from Views into DA Admin Page**
  Drill down from Data Aggregator views to the Monitored Devices Admin page to troubleshoot a view that does not display data.
- **Administer Tenants**
  Administer tenants, including user accounts, discover and delete devices for Data Aggregator.
- **Administer DA Threshold Profiles**
  Administer Data Aggregator threshold profiles, including creating threshold profiles, editing any threshold profiles, and changing the ownership of all threshold profiles.
- **Create DA Threshold Profiles**
  Create Data Aggregator threshold profiles, where you can create and manage event profiles. Event profiles contain event rules, and are associated with groups. You can create reports on the events that are generated with these profiles. This role allows you to create threshold profiles, edit your own profiles, and view all threshold profiles.

CA Network Flow Analysis Role Rights

The following role rights are applicable to the CA NFA (formerly CA ReporterAnalyzer) console:

- **View ToS**
  View Type of Service data
- **Manage Reports**
  Create, modify, delete, and execute reports
- **Run Reports**
  Execute defined reports
- **View Conversations**
  View conversation data
- **View Hosts**
  View host data
- **View Protocols**
  View protocol data

CA Application Delivery Analysis Role Rights

The following role rights are applicable to the CA ADA (formerly NetQoS SuperAgent) management console:

- **Engineering**
  Navigate the Engineering section and create Engineering reports
- **Operations**
  Navigate to the Operations section and create Operations reports
- **Management**
  Navigate the Management section and create Management reports
- **Incidents**
  Navigate the Incidents section and view Incidents reports
- **Investigations**
  Launch Investigations and drill into data from Investigations

Role rights do not give a CA ADA user:

- Permission to access the Administration page of the CA ADA management console.
  To give a user access to the Administration page, give the user the Administrator or Power User product privilege on the CA ADA data source.
- Access to actual report data in the CA ADA management console.
  To enable a user to see report data, assign the appropriate groups to the user.

CA UC Monitor Role Rights

The following role rights are applicable to the CA UC Monitor management console:

- **Call Details**
  Export call details to a CSV file
- **Call Performance**
  Access Call Performance reports
- **Call Quality and Volume**
  Access Call Quality and Volume reports
- **Call Watch**
  Access Call Watch reports
- **Call Watch Setup**

Set up and launch a Call Watch on a selected phone
- **Collector Incidents**
  Access Collector Incident reports
- **Incidents**
  Access Incident reports
- **Investigations**
  Access Investigation reports
- **Launch Investigation**
  Launch an investigation and view the resulting data
- **Phone Details**
  Access Phone Details reports
- **Quality**
  Access Quality reports
- **Trunk Groups**
  Access Trunk Group reports
- **Voice Interface**
  Access Voice Interface reports
- **Midstream Devices**
  Access midstream device and midstream legs reports

Manage Roles

NetOps Portal includes a set of predefined roles that you can assign to a custom user account. You can access summary information about these roles on the Manage Roles page.
For more information, see Role Rights and Data Source Role Rights.

Any custom roles that you create are also listed on this page.

If the predefined user roles that come with NetOps Portal do not fit your requirements, add custom roles. When you have finished creating a role, assign it to a user account as a separate step. Roles are inactive until they are assigned to user accounts. Only users with the 'Administer Users' and 'Administer Roles' role rights can assign roles to the user accounts. For more information, see Manage User Accounts.

You can view, edit, and delete predefined roles and also any custom roles previously added. You can also manage the user accounts associated with a specific role.

Global administrators and users with the required role rights can modify both predefined and custom roles. Tenant administrators only have access to the roles associated with their tenant.

Add or Edit a User Role

You can add custom roles. Create the roles that each unique product operator requires to perform job responsibilities.

Custom roles work best within a system of custom groups. Custom groups let you precisely grant access to dashboards and product features while restricting access to sensitive data. The groups that you create to organize data can serve as “permission groups” when you set up user account permissions.

Follow these steps:

1. Hover over **Administration**, and click **User Settings: Roles**.
2. Click **New**, or select the role that you want to edit and click **Edit**.
3. Enter or edit the required information, and make selections in the provided fields:
   – **Name**

Maximum Characters: 45
- **Description**
- **Enable Role**
  Specify whether to make the role active. Enabling this setting gives users with this role the access granted by its role rights. A role can be disabled for security purposes. When a role is disabled, users who are assigned that role are no longer allowed to log in.

4. Select **Menu Set**, and click **Edit**.
   Move the desired menus from the **Available Menus** list to the **Selected Menus** list.
5. Order the menus in the **Selected Menus** list to determine their order on the **Dashboards** tab and click **OK**.
6. Select **NetOps Portal** , and click **Edit**.
7. Move the desired access rights from the **Available Rights** list to the **Selected Rights** list.
8. Order the role rights to determine their priority in cases where rights overlap and click **OK**.
9. Click **Save** or **Save and Add Another**.

Manage the Users Associated with a Role

Select a role and click **Users** to open the User List page, which is filtered to show only users who are assigned to the selected role. Then manage the users accounts associated with the selected role. Click **Roles** to return to the Manage Roles page. For more information, see Manage User Accounts.

Delete a Role

You can delete any custom role that you have created. The Administrator role cannot be deleted or disabled.

Before you delete a custom role, review the Users column to see whether any user accounts are using this role. Manage the users who are associated with the role to remove all associations with the role you want to delete.

Product Privilege

The user account role is used to grant or restrict user access, such as administration.

Individual data sources allocate product access differently. The 'product privilege' setting for data sources can be applied to create users with administrative capabilities. For example, a user who has no access to administration can have an Administrator product privilege to a specific instance of CA Network Flow Analysis (CA NFA). That user has full administrative privileges to that data source when following a drilldown path for a CA NFA managed item.

The following types of product privilege may be available in the data sources and synchronized to CA NetOps Portal:

- **Administrator**
  Performs all functions, including creating and editing SNMP profiles and other configuration.
- **Power User**
  Creates menus and dashboards, and edits and create roles.
- **User**
  Views menus and dashboards designated by an administrator or power user.
- **None**
  Has no access to a data source. This setting prevents the user from following a drilldown path from a view to the data source's user interface. By default, all users have this product privilege setting for all data sources.
  A user can be denied access to a particular data source while having access to others.

Administrators can customize a user’s access levels by selecting the appropriate role rights.

Coordinate the product privilege setting with the role rights settings. To follow a drilldown path to a data source, a user requires the appropriate role right and a product privilege for that data source.

The predefined administrator account, 'admin', has administrative privileges for any data sources that are registered. The predefined user account, 'user', has limited (user-level) privileges for registered data sources.

Data Source Product Privileges

Each registered data source has its own product privilege with unique privileges within that interface. Administrators can assign a product privilege to a data source. The data source product privilege applies when users follow a drilldown path from a data view to that particular data source. However, any privileges that are granted in this manner are specific to a data source instance. For example, if more than one CA Application Delivery Analysis (CA ADA) data source is registered, the product privileges for each management console are managed separately.

The default administrator account, admin, is locked to prevent changes to product privileges. This account is required to have Administrator privileges for all registered data sources. Selecting a group of accounts that includes the admin account prevents you from editing the product privileges for any of the selected accounts.

CA Application Delivery Analysis Product Privileges

The following list summarizes product privileges applicable to the CA ADA (formerly CA SuperAgent) management console:

A user must have product privileges on the CA ADA data source to log in to the management console. Product privileges also specify access to the Administration page:

- **User**
  Gives access to all pages of the management console, except the Administration page.
- **Administrator**
  Gives access to all pages of the management console, including the Administration page.
- **Power User**
  Gives user-level product privilege, and Show Me menu access to the SNMP Profiles, Network Devices, and Device Groups on the Administration page.

  NOTE

  If a user cannot log in to the management console user interface, verify that the user has been given a product privilege on the CA ADA data source.

CA Network Flow Analysis Product Privileges

A user must have product privileges for the CA Network Flow Analysis (NFA) data source to log in to the NFA console. Product privileges also determine whether a user can access the Administration page, and can perform certain functions:

- **Administrator**
  Gives access to the Administration page in the NFA console, and to all functions. Functions include creating and managing user accounts, roles, groups, SNMP profiles, and scheduling for reports.
- **Power User**
  Gives user-level access and any additional abilities that the Role setting grants. For CA NFA, the Power User privilege is equivalent to the Administrator privilege.
- **User**
  Gives access to Top Interfaces reports and Interface Utilization reports on the Enterprise Overview page.
  A User with the appropriate Permission Group settings also has access to the following reports:
  - Top Hosts and Top Protocols reports on the Enterprise Overview page, if the user also has access to All Groups
  - Interfaces page reports for the interfaces that are accessible to the user
  - Existing reports on the Custom Reporting, Flow Forensics, and Analysis pages
  - Menus that an administrator has assigned to the User role

The Role and Permission Group settings determine whether the user can also run existing reports, create reports, and manage reports. To create reports, a user must have access to All Groups.

- **None**
  Has no access to a data source. The user who has this product privilege cannot log in to the NFA console or drill down from a CA NetOps Portal view to the NFA console. By default, all users have this product privilege setting for all data sources.

  NOTE
  The same user account can have different privileges for different data sources.

CA UC Monitor Product Privileges

The following list summarizes the product privileges applicable to the CA UC Monitor management console:

- **Administrator**
  Gives access to all functions, including administrative tasks, such as creating and editing:
  - Locations
  - Media devices
  - Thresholds
  - Call Watch definitions
  - Incident responses
  - Roles
  - User accounts
- **User**
  Gives access to report pages, and the ability to perform basic functions that the administrator selects. The User permission does not give access to administrative functions.

Manage Product Access

Grant access to product features and data as you create each user account. Use the following method to verify and modify the role rights for a specific user.

Follow these steps:

1. Log in as a user with the required administrative role rights.
2. Select **Administration**, **User Settings**, and click **Users**.
   The Manage Users page opens.
3. Select the user account that you want to edit.

   NOTE

   : The rights and privileges that are assigned to the predefined administrator account, 'admin', cannot be modified. This user account must have administrator access to all registered data sources.

   The Create New User wizard opens.
4. Click **Product Privileges**.

   NOTE
   All registered data sources appear on the Product Privileges page.
5. Click the values that are shown in the Product Privileges column to enable drop-down lists.
6. Select one of the following product privileges from the drop-down lists:
   - **Administrator**
     Performs all functions, including creating and editing groups, menus, dashboards, roles, and user accounts.
   - **Power User**

Creates menus and dashboards, and edits and creates roles.
- **User**
  Views menus and dashboards that are designated by an administrator or power user.
- **None**
  Has no access to a data source. This setting prevents the user from following a drilldown path from a view to the data source's user interface. By default, all users have this product privilege setting for all data sources.

7. Click **Save**.
   The changes to product privileges are saved to the selected user account.

Manage User Accounts

You can see high-level settings for the user accounts on the Manage Users page. In a multi-tenant environment, the global administrator sees a list of user accounts that are unassociated with a tenant. Tenant administrators only see user accounts for their tenant.

Before you create any custom user accounts, only the two factory user accounts are available.

You can add custom user accounts in addition to the predefined administration and user accounts that come with NetOps Portal. You can view, edit, clone, delete, and proxy existing user accounts.

> **NOTE**
> You cannot delete the two predefined user accounts (**admin** and **user**).

Permission Groups and User Accounts

The predefined groups (or system groups) help you quickly organize performance data and allocate operator access to that data. However, a more secure and better managed system is based on custom groups that are assigned to users as permissions.

Permission groups comprise the scope of the managed items that each user can monitor. Administrators can create custom groups of managed items, such as applications, servers, networks, routers, and interfaces, to reflect each user's area of responsibility. When they are assigned to a user account as permissions, custom groups are called permission groups.

You can assign multiple permission groups to each user during user account creation. For example, assign the permission groups 'North American Core Routers' and 'North American Critical Applications' to the same user account.

> **NOTE**
> As a best practice, do not assign the 'Collections' group as part of a user's permission groups. This group should not be used for reporting.

We recommend speaking with a CA technical representative to plan a strategy for creating a grouping and role structure. The best configuration meets your current requirements and is flexible enough to accommodate changes to your system.

Predefined User Accounts

NetOps Portal provides two predefined ("factory") user accounts. These accounts are useful for performing initial setup. You can use them to allocate LDAP access with minimal role rights, or as templates for custom user accounts. But because they are common to all NetOps Portal installations, they are less secure.

> **IMPORTANT**
> The factory user accounts are not substitutes for custom user accounts. We recommend changing the default passwords immediately after installation for improved security.

The factory user accounts have the following parameters:

- **admin**

Grants all administrative privileges.
Role: Administrator
Special Role Rights: All (the "global administrator" or Default Tenant administrator)
Permission Groups: Can view data from all groups
Default password: admin

- **user**
  Specifies typical operator privileges, such as viewing data.
  Role: IT Operator
  Special Role Rights: None
  Permission Groups: Can view data from all groups
  Default password: user

User account status is "Enabled" or "Disabled". Disable an account to prevent a user from accessing the product.

User Account Parameters

User accounts have the following required associations:

- **Role**
  The *role* is a parameter assigned to a user account that controls user access to product features and dashboard pages. Based on user job functions, the role grants administrative access to product configuration using *role rights*. Roles let users access data and product features that they require to perform their duties and restrict access to features that they do not require.
  NetOps Portal provides multiple predefined roles, with different role rights. A user with the required role rights can create additional roles and assign them to user accounts.
- **Permission Groups**
  Permission groups comprise the scope of the managed items that each user can monitor. Administrators can create custom groups of managed items, such as applications, servers, networks, routers, and interfaces, to reflect each user's area of responsibility. When they are assigned to a user account as permissions, custom groups are called permission groups.
  By default, new user accounts have no group assignment. If you want new users to see managed items, you must assign one or more groups to their user accounts. The predefined 'admin' and 'user' accounts have access to all groups. For user accounts that you create, limit the groups users can see based on their responsibilities.
- **Product Privilege**
  The *product privilege* is a type of permission set associated with a user account. The product privilege grants user access to features in selected data sources and does not apply to NetOps Portal functionality.

  NOTE
  In previous versions of NetQoS NetOps Portal, the product privilege referred to administrative access to product configuration, such as the ability to create custom groups. The role rights assigned to the user account now determine access to these features in CA NetOps Portal.

Add, Clone, or Edit a User Account

Add a user account for each person who operates NetOps Portal. For security purposes, do not share user accounts.

To create new user accounts quickly, use the Clone feature. You use existing accounts, such as the predefined user account, "user", as templates for the new user accounts.

Administrators can also create user account templates that are based on job function. These templates can be cloned to create individual accounts more easily. For security reasons, we recommend disabling the 'Enable user account' setting for templates. Disabling this setting helps prevent unintentional access to NetOps Portal. Instead, enable NetOps Portal access as needed for the user accounts you create by cloning templates.

NOTE
Before you create a user account, confirm that the required roles and groups exist.

Modify a user account when its job responsibilities change, or when new permission groups or roles are created. Edit a user account to assign any new roles that you create.

We recommend checking the permissions that are associated with each user account periodically. Checking permissions helps to ensure that all items are being monitored. Each time a new data source is registered, new system groups are added to the Groups tree. Sometimes, no NetOps Portal operators monitor these new groups until you explicitly add them to user accounts.

Follow these steps:

1. Hover over **Administration**, and click **User Settings: Users**.
2. Click **New**, **Clone**, or select the account that you want to change, and then click **Edit**.

   NOTE
   The rights and privileges that are assigned to the predefined administrator account, 'admin', cannot be modified. This user account must have administrator access to all registered data sources. If you select a group of accounts that includes the 'admin' account, you cannot modify any of the selected accounts.
3. Enter or edit the required information, make selections in the provided fields, and advance the wizard:
   - **Name**
   - **Description**
   - **Preferred Language**
     Specify a language for the NetOps Portal user interface. NetOps Portal displays the selected language regardless of the language selected for the operating system or for the browser language.

     NOTE
     For a language to display appropriately, the relevant fonts must be installed on the NetOps Portal server and the client.
   - **Email Address**
   - **Password**
   - **Time Zone**
     Specify the time zone that is applied to all dashboards viewed while the user is logged in to NetOps Portal. Typically, the time zone matches the locale of the computer that the operator uses to access NetOps Portal.
     Default: UTC (Coordinated Universal Time)
   - **Role**
   - **Account Status**
4. Specify the access permissions settings, and advance the wizard:
   - Move the desired permissions groups or subgroups from the **Available** list to the **Selected** list.
     Map each user profile to a group that makes sense for their initial view.

     IMPORTANT
     As a best practice, do not assign the Collections group as part of a user's permission groups. Do not use this group for reporting.
   - **Enable My Custom Groups Functionality (Optional)**
     Allow the user to create custom groups to organize managed items for troubleshooting and analysis. These groups are only available to this user on the My Custom Groups page. They do not appear in the main Groups tree.
   - **Default Group**
     A default group is selected for the user automatically. When the user logs in, data from the default group appears in dashboards by default. The group selections on the Access Permissions dialog filter the Available Groups tree. This filtering prevents users from having administrative rights to a part of the tree from which they are prohibited.

     NOTE
     Do not set the default group to “All Groups.” Using "All Groups" as the default group can cause performance issues and result in slow dashboard performance overall.

5. Move the desired administer groups from the **Available Groups** list to the **Selected Groups** list for a user with the **Administer Groups** role right, and advance the wizard.
   Lock icons appear next to read-only groups, which cannot be administered.
   Adding administered groups for an Administrator is not required.
   A user with **Administer Groups** role right can create groups *under* the selected group or subgroups. They can then modify or delete only those administered groups. Users cannot modify or delete groups that another user owns. For more information, see Role Rights.
6. For each data source product, specify one of the following product privileges, and click **Save**:
   – **Administrator**
     Allow the user to perform all functions, including creating and editing groups, menus, dashboards, roles, and user accounts.
   – **Power User**
     Allow the user to create menus and dashboards and also edit and create roles.
   – **User**
     Allow the user to view menus and dashboards that are designated by an administrator or power user.
   – **None**
     Restrict the user so that it has no access to a data source. Prevent the user from following a drill-down path from a view in NetOps Portal to the data source user interface. By default, all users have this product privilege setting for all data sources.
     The same user account can have different privileges for different data sources.
   For more information, see Product Privilege and Data Source Product Privileges.

Proxy Users and Tenants

The proxy feature lets administrators see and interact with NetOps Portal as another user or tenant. When you proxy a user, the menus, dashboards, permission groups, and user settings are proxied. When you proxy a tenant, you can see the users, items, and settings of that tenant.

> **NOTE**
>
> Role rights and product privileges to other data sources are not proxied. The proxied user account only persists within NetOps Portal. If you follow a drilldown path to a data source, the user account defaults to the original user account settings.

The proxy feature is useful for the following tasks:

- Verify permission groups and role rights.
- Configure and test menus and dashboards for a user.
- Configure a tenant environment.

Proxy a Tenant

Hover over **Administration**, and click **Group Settings: Tenants.** Select the tenant, and click **Administer**.

Proxy a User

> **TIP**
>
> To proxy a user account from another tenant, first proxy that tenant.

Hover over **Administration**, and click **User Settings: Users**. Select the user account, and click **Proxy**.

To stop proxying, click the **X** next to the **Proxy User** or **Proxy Tenant** indicator.

Multi-tenancy

To create separate monitoring environments that you administer from a single user interface, add custom tenants. The multi-tenancy support lets you monitor discrete customer environments separately and securely. A tenant represents a customer environment that a managed service provider (MSP) administers. Each tenant environment is independent and effectively functions as a separate instance of DX NetOps Performance Management. Each instance can contain multiple users and roles that are not shared among tenants.

By default, all managed items and their data are associated with the Default Tenant.

The basic tenant definition contains a few parameters to identify the MSP customer. Each tenant is associated with one or more IP domains. All items in the associated IP domains are associated with that tenant. The global administrator or the tenant administrator sets up the monitoring environment for the tenant. The following configuration items are scoped to the tenant:

- SNMP profiles
- User accounts
- Roles
- Custom groups
- Custom dashboards
- Custom menus
- Discovery profiles
- Threshold profiles

Administrator Roles for Multi-tenancy

When multi-tenancy is deployed, DX NetOps Performance Management supports the Global Administrator and Tenant Administrator roles:

- **Global Administrator**
  The Default Tenant administrator, usually representing an MSP. Product settings and data are not shared among tenants, but the Default Tenant Administrator can access and modify all settings. This user must have the predefined "Administrator" role.
- **Tenant Administrator**
  A limited administrator who is associated with a single tenant. This operator cannot access shared infrastructure or configuration belonging to the host (usually, the MSP). Tenant user accounts can include one or more of these administrator accounts. When you create a tenant, the user interface prompts you to create a tenant administrator.

Multi-tenancy Deployment Considerations

Consider the following factors when you create tenants or IP domain definitions:

- The size, scope, and organization of your monitoring environment
- The CA data sources that you plan to install and register
- Data source support for the multi-tenancy features

All of these factors work together to determine your strategy. For example, some data sources do not detect IP domains that are created within tenants.

WARNING

Once data collection has started, changes IP domain or tenant definitions are difficult. Data sources collect and aggregate data that retains a database association with the original IP domain or tenant.

Domain Monitoring Considerations

The IP domains feature supports environments where multiple enterprise systems must be monitored separately. For example, a managed services provider wants to monitor the systems and networks of different customers separately. The MSP administrator creates a tenant in CA NetOps Portal for each customer enterprise. The data and the configuration for each tenant are hidden from all other tenant users.

However, in other situations, you can deploy multiple IP domains in CA NetOps Portal without multi-tenancy. In other words, some deployment models consist of *multiple IP domains within the Default Tenant*.

The IP domain lets you control data collection parameters. Use custom IP domains to determine which collection devices monitor the managed items in your infrastructure. Each collection device, such as a Data Collector or CA Unified Communications Monitor Collector, operates within a single IP domain.

The following list provides some examples of environments where you can deploy multiple IP domains within the Default Tenant:

- A deployment that includes a CA Application Delivery Analysis data source.
  CA Application Delivery Analysis monitors IP domains without a concept of tenants. The IP domains that you create within custom tenants are not detected. When the Data Aggregator or Network Flow Analysis monitors items in those domains, they appear as duplicates in CA Application Delivery Analysis. The duplicate data is not aggregated.
- A large deployment that requires load balancing.
  For example, your enterprise includes ten routers with many interfaces, IP SLA testing, and QoS policies in place. Such a deployment would have a polling load similar to an environment with hundreds of servers being monitored for CPU and memory statistics only.
  To monitor the busy routers, you can create an IP domain and can deploy a powerful system for the Data Collector within that domain. And you can monitor the servers in another IP domain, using a less powerful system for the Data Collector. By running discoveries in the appropriate IP domains, you can determine the devices that each Data Collector is polling.
- A method to minimize the potential network impact of bulk statistics collection.
  For example, you can deploy a Data Collector close to the devices that it is monitoring. Data Collectors can process a massive amount of bulk statistics and can reduce them to a much smaller set of monitored metrics, which are sent to the Data Aggregator. As a result, less data passes across the network between the two components.
- Isolation of potentially sensitive SNMP traffic to a specific area, such as a DMZ.
  For example, security policies do not let SNMP traffic travel across the router that limits an area of network. One option is to deploy a Data Collector behind the router. A path to return the processed metrics back to the Data Aggregator must be open.
  The metric data traveling between the components is not encrypted. However, it is packaged and compressed in a way that makes it less "sniffable." As a result, the data is more secure than raw SNMP flows. To accomplish this setup, create an IP domain for the DMZ and deploy a Data Collector within that IP domain.

Deployment Process

The global administrator who is associated with the default Tenant space performs the initial steps to create a multi-tenant environment.

We recommend the following process for setting up a multi-tenant deployment:

1. Collect data about MSP customer virtual and physical systems.
2. Make a list of IP domains and SNMP versions, communities, or passwords for each MSP customer.
3. Create tenants. The tenant definition consists of a few simple parameters to identify the associated customer.
   The tenant definition also includes tenant administrator and user accounts.
4. Set the scope to a tenant to administer tenant configuration while logged in as a global administrator.
5. Create at least one IP domain to represent customer networks.
6. Create at least one SNMP profile to enable SNMP polling of devices supporting customer infrastructure.

7. Exit tenant administration. Repeat the previous steps for each tenant.

If data sources are already registered and are collecting data, wait a few minutes. CA NetOps Portal creates system groups based on items that are discovered during monitoring. These groups are useful for creating custom groups that you can then allocate to users as permissions.

When system groups are available, take the following steps:

1. Set the scope to a tenant to administer tenant configuration, or log in as the tenant administrator.
2. Create any custom groups that are required to represent the customer networks and systems.
3. Edit the default tenant user account to add permission groups.
   Consider the likely role of this user, and the managed items that this user manages.
4. Create any other custom roles, user accounts, SNMP profiles, dashboards, and menus that are required for this customer.

Work with the IT staff of each customer to designate a user to act as the tenant administrator. The tenant administrator can complete the tenant configuration by creating custom groups and additional user accounts, if desired.

Data Source Support for Multi-tenancy

DX NetOps Performance Management offers full support for multi-tenant monitoring. Integrated environments that include the following data sources offer limited multi-tenancy support:

Network Flow Analysis

Supported Features

Full support for multi-tenancy.

Assign a tenant and IP domain to each Harvester and router. Tenant assignment determines the configuration items that are available (such as tenant SNMP profiles).

A few limitations are noted in Multi-tenancy and CA Network Flow Analysis.

CA Application Delivery Analysis

Supported Features

- IP Domains
- Custom Tenant configuration

Does not segregate data within the data source interface.

CA Application Delivery Analysis monitors IP domains without a concept of tenants. As a result, CA NetOps Portal receives all items from CA Application Delivery Analysis in the Default Tenant. However, CA Application Delivery Analysis does support IP domains. CA NetOps Portal can thus associate these items with tenants according to their IP domain. Be aware that some managed items are duplicated between the Default Tenant and custom tenants.

DX NetOps Spectrum

Supported Features

Full support for multi-tenancy. However, tenancy is only visible in DX NetOps Performance Management, not in OneClick.

OneClick receives IP domains from CA NetOps Portal. Models in IP domains are synchronized them with CA NetOps Portal (and thus, associated with custom tenants).

Tenants are not visible in CA Spectrum OneClick. However, tenants in CA NetOps Portal have associated CA Spectrum devices based on IP domain. The global administrator can also make these items available for monitoring by tenant users by placing them into the Global Tenant Items group.

CA Unified Communications Monitor

Supported Features

Full support for multi-tenancy.

Locations are automatically associated with IP domains by their subnets.

CA eHealth

Supported Features

No support for multi-tenancy.

All items from these data sources are associated with the Default Tenant and Default IP Domain. Add items from these data sources to Service Provider groups to grant tenant access to them.

CA Application Performance Management

Supported Features

No support for multi-tenancy.

All items from these data sources are associated with the Default Tenant and Default IP Domain. Add items from these data sources to Service Provider groups to grant tenant access to them.

When multi-tenancy is deployed, DX NetOps Performance Management supports the Global Administrator and Tenant Administrator roles:

- **Global Administrator** - The Default Tenant administrator, usually representing an MSP. Product settings and data are not shared among tenants, but the Default Tenant Administrator can access and modify all settings. This user must have the predefined "Administrator" role.
- **Tenant Administrator** - A limited administrator associated with a single tenant. This operator cannot access shared infrastructure or configuration belonging to the host (usually, the MSP). Tenant user accounts can include one or more of these administrator accounts.

When you create a tenant, the user interface prompts you to create a tenant administrator and a tenant user account. Operators who use these accounts can only perform monitoring or administrative tasks within this tenant. They cannot access the managed items and parameters associated with other tenants.

Configure a Tenant Environment

Configuration tasks in a multi-tenant environment are split between the global administrator and the tenant administrator. The global administrator creates the tenant and configures monitoring profiles. The tenant administrator configures discovery and reporting.

Gather Prerequisite Information

Work closely with the tenant customer to plan the tenant deployment. Collect basic information about the customer environment, such as IP domains and SNMP profiles. Knowledge of physical and virtual system topology is useful for creating a custom grouping structure to represent the customer environment.

Select a user to act as the tenant administrator. The tenant administrator can then complete the tenant configuration by creating custom groups, roles, users, SNMP profiles, menus, and dashboards.

Before you begin the tenant setup, gather the following information:

- List of IP domains for the tenant
  Each IP domain requires a dedicated Data Collector
- List of SNMP communities for the tenant networks
- The desired username of the tenant administrator
  For example, select a representative of the customer site that is monitored.
- A dedicated Data Collector or a multi-tenant Data Collector with available capacity
- Configured monitoring profiles
  Monitoring profiles are global. The global administrator creates the required monitoring profiles that are used across all tenants. For more information, see Configure Monitoring Profiles.

Set Up the Tenant

Global Administrator

Create the Tenant Definition

The global administrator creates the tenant definition and tenant administrator. The tenant administrator can only see data and configuration for a single tenant. Data from other tenants is not accessible to a tenant administrator.

Follow these steps:

1. Hover over **Administration**, and click **Group Settings: Tenants**.
2. Click **New**.
3. Specify the required information.
4. Create user accounts for the tenant administrator and default tenant user.
5. Click **Save**.
   The tenant and the tenant administrator are created. The tenant administrator can complete the following procedures. Alternatively, the global administrator can administer the tenant environment to complete configuration.

   TIP

   To administer a tenant as the global administrator, hover over **Administration: Custom Settings**, and click **Tenants**. Select the tenant, and click **Administer**.

Define IP Domains

Each tenant requires an associated IP domain.

Follow these steps:

1. Select **Administration**, **IP Domains**.
2. Click **New**.
3. Specify a name and description.
4. (Optional) To assign a primary and secondary DNS address for the domain in Network Flow Analysis, select **DNS Settings**, and specify the required values.

   NOTE

   Only Network Flow Analysis uses the DNS settings. DX NetOps Performance Management does not directly use the DNS settings.
5. Click **Save**.

Assign the IP Address to a Data Collector

For a tenant environment with a dedicated data collector, assign the tenant and the IP domain to the Data Collector. For a multi-tenant Data Collector environment, configure the association through REST. For more information, see Tenant-Agnostic Data Collectors.

Follow these steps:

1. Hover over **Administration**, and click **Monitored Items Management: Data Collectors**.
2. Select the Data Collector, and click **Assign**.
3. Select the IP domain and tenant, and click **Save**.
   The IP domain and tenant are bound to the Data Collector.

Discover Devices

Tenant Administrator

Discovery is the process that DX NetOps Performance Management uses to build an inventory of devices in your network.

For more information, see Discovery.

Configure SNMP Profiles

SNMP profiles provide authentication credentials to communicate with devices in your network.

Follow these steps:

1. Hover over **Administration**, and click **Configuration Settings: SNMP Profiles**.
2. Click **New**.
3. Complete the fields, and change any default settings. Some fields apply only to SNMPv3.
   For complete details, see SNMP Profiles.
4. Click **Save**.
   The SNMP profile is added to the system and used for discovery and polling.

Create Discovery Profiles

Discovery profiles specify which devices DX NetOps Performance Management discovers. Create granular discovery profiles for devices with different SNMP credentials or different rediscovery schedule. Granular discovery profiles reduce unnecessary SNMP requests.

For more information, see Discovery Profiles.

Follow these steps:

1. Hover over **Administration**, and click **Monitored Items Management: Discovery Profiles List**.
2. Click **New**.
3. Specify a name for the profile.
4. Specify the IP address, IP ranges, or hostnames to target for discovery.
5. (Optional) Open the **SNMP** tab, select **Use specific list of assigned SNMP profiles**, and select the SNMP profiles to include in discovery.
   Using a specific list of SNMP profiles reduces unnecessary SNMP requests.
6. (Optional) Open the Schedule tab, and define a schedule.
   During normal operation, discovery runs on a schedule basis to discover new devices in the target range.
7. Click **Save**.
   DX NetOps Performance Management uses the discovery profile to find devices in your network.

Run Discovery

To build your inventory, use the discovery profiles to run discovery.

Follow these steps:

1. Select the discovery profile.
2. Click **Run**. You can run discovery only if the State is Ready.
   DX NetOps Performance Management discovers the devices within the specified IP addresses and hostnames, and adds the devices to the system inventory.
   To view a list of the discovered devices, select the discovery profile, and click History.

Configure Monitoring Collections

Tenant Administrator

In a tenant environment, the tenant administrator defines monitoring by associating collections with monitoring profiles. Monitoring profiles control how often to poll devices and which information to collect. Assigned metric families determine which metrics the system collects. Collections are system groups that group devices for monitoring. Associating a collection with a monitoring profile causes DX NetOps Performance Management to monitor the devices according to the parameters in that profile. The global administrator defines monitoring profiles for all tenants.

For more information, see Configure Monitoring Profiles.

Configure a Collection

Consider the following best practices for organizing devices into collections for monitoring:

- Create custom collections that match the monitoring requirements in the environment.
  - Consider the different layers of the network, access, distribution, and core. Devices in different layers might require different levels of monitoring.
  - Consider which technologies and metric families are required. Metric families that would be applied to all devices, such as CPU and memory, apply to broad collections. Targeted monitoring, such as QoS and IPSLA, apply to limited collections.
- Create collections that enable the flexibility to break out monitoring.
  - Some devices are included in multiple collections so that specific metric families are polled at different rates.
  - Devices in different collections have different filtering criteria.
  - Different monitoring requirements depending on importance of device

Follow these steps:

1. Hover over **Administration**, and click **Group Settings: Groups**.
2. Select the **Collections** folder in the left pane.
3. Click **Add Group**.
4. Specify a name, and click **Save**.
   The collection is created. To add devices to the collection, add rules.
5. Click the **Rules** tab, and click **+ Add Rule**.
6. Specify a rule name, select devices for the item type, add conditions as required, and click **OK**.
7. Click **Save and Run Rules**.
   DX NetOps Performance Management adds the items to the collection.

Assign Monitoring Profiles to Collections

Follow these steps:

1. From the **Collections** page, select a collection, and click the **Monitoring Profiles** tab.

2. Click **Manage**.
3. Select monitoring profiles for the **Assigned Monitoring Profiles** list, and click **Save**.
   The monitoring behavior that is defined in the assigned monitoring profiles is applied to the device collection.

Reporting, dashboards, and threshold monitoring function as normal in a tenant environment. For more information, see Configure Reporting in a New Environment.

Manage Tenants

Tenants provide you with separate monitoring environments that you administer from a single user interface. Tenants are not required for all deployments. The multi-tenancy feature lets an MSP monitor discrete customer networks and systems from a single instance of DX NetOps Performance Management.

NOTE

To edit the SNMP profiles, IP domains, or other configuration for a tenant, see Administer Tenants.

The global administrator manages tenants. View and manage tenants from the Manage Tenants page, which lists all the tenants in the system. To access the page, hover over **Administration**, and click **Group Settings: Tenants**.

These procedures required the default Administrator role:

Create a Tenant

During tenant creation, you can also create a tenant administrator and a tenant user. Unlike the global administrator, the tenant administrator can only see data and configuration for a single tenant. Data from other MSP customers is not accessible to a tenant administrator.

Follow these steps:

1. On the Manage Tenants page, click **New**.
2. Supply the required information, and make selections in the fields provided:
   - **Name**
   - **Account ID**
     This ID identifies the tenant and often corresponds to the MSP account number.
   - **Description**
   - **Status**
     Select whether the tenant is enabled or disabled.
   - **Theme**
     The theme is applied to all user accounts that are associated with this tenant.
   - **Language**
3. Specify the credentials for the tenant administrator.
   This user has administrative privileges in the tenant.
4. Specify the credentials for the default tenant user.
   This user account can access tenant-specific dashboards, but cannot access any administrative functions.
5. Click **Save**.
   The new tenant definition is created. To configure monitoring in the tenant, Administer the tenant.

Clone a Tenant

To use status, theme, and language of an existing tenant as a template for a new tenant, select the existing tenant, and click **Clone**.

NOTE

Tenant-specific information, such as users, groups, and SNMP profiles, is not copied. Configure monitoring in a tenant clone as if the clone is a new tenant.

Edit a Tenant

To modify any of the parameters that are defined during tenant creation, select the tenant, and click Edit. To charge monitoring configuration in the tenant, administer the tenant.

Disable and Enable a Tenant

To stop active monitoring of the infrastructure of a tenant, edit the tenant, and change the status to Disabled.

After you disable the tenant, the following results occur after the next synchronization with the Data Aggregator:

- The Data Aggregator system stops all Data Collectors that are associated with the disabled tenant. Data Collectors have the **Not Collecting Data** status. For any new Data Collector installations for a disabled tenant, the status is **Not Collecting Data**.

  NOTE
  If you reenable the tenant, manually restart the Data Collectors.
- Polling is stopped for any devices and components that are being monitored for the tenant.
- Historical data on the devices and components for the tenant remains accessible.
- Discovery profiles that are associated with the tenant cannot be run. The discovery profiles have the **Tenant Disabled** state.
- If a discovery profile is invalidated while a discovery is running, the discovery is aborted.
- An audit event is generated on the Data Aggregator device for the disabled tenant.

When you enable a tenant, the following results occur after the next synchronization with the Data Aggregator:

- Discovery profiles that are associated with the enabled tenant are validated. The discovery profiles display their current state.
- An audit event is generated on the Data Aggregator device for the tenant.

To restart polling, restart the Data Collectors that are associated with the tenant. After the restart, the following results occur:

- Polling is restarted for any devices and components that are being monitored on for the tenant.
- Discovery profiles that are associated with the enabled tenant can be run. If the discovery profile is schedule, the system runs discovery at the next scheduled time.

Delete a Tenant

To remove a tenant and all associated data from the system, delete the tenant.

WARNING

Information in a deleted tenant is not recoverable. To stop monitoring in a tenant, disable the tenant.

When you delete a tenant, the following definitions are deleted and cannot be recovered:

- Data sources
- SNMP profiles
- IP domains
- Threshold profiles
- User accounts
- Roles
- Groups
- Custom dashboards
- Custom menus

After you delete the tenant, the following results occur after the next synchronization with the Data Aggregator:

- Polling on the deleted devices and device components is stopped.
- Historical data on the deleted devices and device components is no longer accessible.
- An audit event is generated on the Data Aggregator device for the deleted tenant.

> **NOTE**
> If you delete a tenant when an associated Data Collector is down, the Data collector immediately stops when it restarts. An error message is logged in the following file:*DC_installation_directory*/apache-karaf-<*vers*>/shutdown.log

Administer Tenants

After you add a tenant, set up the required monitoring parameters and user access. To set up a tenant environment, log in as a tenant administrator that is associated with that tenant. If you are a global administrator, you can use the Administer Tenant feature to access NetOps Portal from the perspective of the tenant.

When you set the tenant scope to a selected tenant, only the configuration items available to that tenant appear. You can then administer the tenant, creating the required IP domains, user accounts, and more. They will only be available to users with permission to see the items that belong to that tenant.

Tenant Administrator

As a global administrator or a tenant administrator, you can modify the monitoring parameters of a tenant. Custom definitions that you create while administering a tenant are specific to that tenant, and are not shared among tenants.

To modify the IP domain, SNMP profile, user, role, and group definitions for a tenant, the tenant administrator simply logs in. The global administrator (the administrator for the Default Tenant) must set the tenant scope to the selected tenant to gain access to these definitions.

> **NOTE**
> The global administrator can create tenant administrator user accounts for each tenant.

When the tenant scope has been set, the procedures for administering a tenant are identical to the procedures to perform in a single-tenant environment.

Follow these steps:

1. Log in as a tenant administrator associated with this tenant.
   You can also set the tenant scope to access tenant configuration while logged in as the global administrator.
   The Administering Tenant indicator appears to show that you are administering the selected tenant environment.
   You can now see and modify only definitions that are associated with this tenant.
2. Click the Admininstration tab, and select an item to modify:

- IP Domains
- SNMP Profiles
- Groups
- Menus
- Roles
- Users

3. Follow the procedures specific to the selected item.
4. Save your changes.
   The modifications are only visible to administrators and operators whose user accounts were created within this tenant environment.

Global Administrator

To set up the environment for a tenant that you have already created, use the Administer Tenant feature. For example, you can add custom IP domains, user accounts, or groups to the tenant. Set the scope to the tenant to access NetOps Portal from the perspective of the tenant.

Follow these steps:

1. Log in as an Administrator.
2. Select **Administration**, **Group Settings**, and click **Tenants**.
3. Select the tenant that you want to administer, and click **Administer**.
   You are only able to see the configuration that is associated with the selected tenant.
   You can now create the IP domains, SNMP profiles, roles, users, menus, and groups that are required to represent and monitor this tenant environment. Use the menus under the **Administration** tab to configure the tenant.
4. (Optional) Change the tenant scope to another tenant by clicking the [change] link.
   The Manage Tenants page appears, and you can select another tenant.

As a global administrator or a tenant administrator, you can modify the monitoring parameters of a tenant. Custom definitions that you create while administering a tenant are specific to that tenant, and are not shared among tenants.

To modify the IP domain, SNMP profile, user, role, and group definitions for a tenant, the tenant administrator simply logs in. The global administrator (the administrator for the Default Tenant) must set the tenant scope to the selected tenant to gain access to these definitions.

> **NOTE**
> The global administrator can create tenant administrator user accounts for each tenant.

When the tenant scope has been set, the procedures for administering a tenant are identical to the procedures to perform in a single-tenant environment.

Follow these steps:

1. Log in as a tenant administrator associated with this tenant.
   You can also set the tenant scope to access tenant configuration while logged in as the global administrator.
   The Administering Tenant indicator appears to show that you are administering the selected tenant environment.
   You can now see and modify only definitions that are associated with this tenant.
2. Click the Administration tab, and select an item to modify:
   - IP Domains
   - SNMP Profiles
   - Groups
   - Menus
   - Roles
   - Users
3. Follow the procedures specific to the selected item.

4. Save your changes.
   The modifications are only visible to administrators and operators whose user accounts were created within this tenant environment.

Automate Tenant Configuration with REST Web Services

As an MSP administrator, you use a third-party or proprietary tool to manage tenants. You want to automate the configuration of your tenant information for monitoring and reporting purposes. This scenario shows how REST web services can be used to configure tenant information. For automated tenant configuration, you would write an application or script that leverages the web services in this scenario.

To configure tenant information using REST web services, follow these steps:

> **NOTE**
> Perform steps 1-3 using NetOps Portal REST web services. Perform steps 4-6 using the DX NetOps Performance Management Data Aggregator REST web services.

This process requires making calls to two different web servers:

- The NetOps Portal web server is used to create tenants, IP domains, and SNMP profiles.
- The Data Aggregator web server is used to assign Data Collector to an IP domain, to create discovery profiles, and to review discovery results.

> **NOTE**
> Use either a REST client editor or an HTTP tool that sends requests and gets responses to perform these procedures manually. In this scenario, we simply refer to the REST client editor.

Considerations Before You Begin

Before you configure the web services, consider the following information:

- Identifiers from NetOps Portal REST web services and DX NetOps Performance Management Data Aggregator REST web services are not interchangeable.
- When you upgrade, the version of the web services you were using might no longer be compatible. Upon upgrade, review the Release Notes to see if the REST web services have changed. If the REST web services changed, update your application or script.

Create a Tenant

As an MSP administrator, you can create tenants with specific parameters. The basic tenant definition contains a few parameters to identify your customer and let other operators access managed items and configuration for the customer. An administrator account is a required component of the tenant definition so that the customer can perform some tenant setup.

Follow these steps:

1. Enter a URL for the NetOps Portal REST web services API in the REST client. Use the following format:

   ```
   http://pc_host:8181/pc/center/webservice/tenants/
   ```

2. Select POST for "HTTP" Method.
3. Provide a valid username and password for a user account that has administrator access to NetOps Portal.
4. Select 'application/xml' as the 'Body Content-type' in the Body settings.
5. Enter the following information within the "Body" text section:
   – **tenantName**

Provides a name for the tenant you are creating.

- **tenantDesc**
  Describes the tenant. The tenantDesc tag is required, but an actual description is optional.
- **accountIdentifier**
  (Optional) Identifies this tenant. This value usually corresponds to the MSP account number. If a value is supplied as input, it must be unique across all defined tenants.
  Default: null
- **status**
  (Optional) Defines the status of this tenant. Select one of the following values:
  - Activated: Enables tenant user accounts for use.
  - Disabled: Prevents any actions by user accounts that are associated with this tenant.

  Default: ACTIVATED
- **removable**
  (Optional) States whether the item can be deleted (removed from the database).
  Values: true or false.
  Default: true
- **theme**
  (Optional) Specifies the theme that controls the appearance of NetOps Portal pages for this tenant. All operators whose user account is associated with this tenant see this same theme. Two themes are available: CA-Blue and CA-Gray.
  Default: CA-Blue.
- **defaultCulture**
  (Optional) Specifies the language (locale) for this tenant. Supply a language identifier from the following list:
  - en_US (English, United States)
  - ja_JP (Japanese)
  - zh_CN (Simplified Chinese)
  - zh_TW (Traditional Chinese)
  - fr_FR (French, France)

  Default: en_US

For example:

```
<tenant>

  <tenantName>John Doe</tenantName>

  <tenantDesc>John Doe Corporation tenant</tenantDesc>

  <accountIdentifier>JD1234</accountIdentifier>

  <status>Enabled</status>

  <removable>false</removable>

  <theme>CA-Blue</theme>

  <defaultCulture>en-US</defaultCulture>

</tenant>
```

6. Execute the web service call.
   The tenant is created.

Create an IP Domain

A tenant definition must contain at least one IP domain that identifies the IP addresses of managed items in the tenant environment. Create an IP domain and add it to the tenant you created previously.

Follow these steps:

1. Enter a URL for the NetOps Portal REST web services API in the REST client. Use the following format:

   ```
   http://pc_host:8181/pc/center/webservice/domains/
   ```

2. Select POST for "HTTP" Method.
3. Provide a valid username and password for a user account that has administrator access to NetOps Portal.
4. Select 'application/xml' as the 'Body Content-type' in the Body settings.
5. Enter the following information within the "Body" text section:
   - **groupItemID**
     (Optional) Defines an internal (database) identifier for the group definition that is associated with a domain. You provide this number. To specify a valid groupItemID, make a web service call to the groups webservice:

     ```
     http://pc_host:8181/pc/center/webservice/groups
     ```

   - **itemDesc**
     Describes the IP domain. The itemDesc tag is required, but an actual description is optional.
   - **itemName**
     Defines a name for the IP domain.
   - **TenantID**
     Defines an internal (database) identifier for the tenant definition. The tenant ID was assigned when you created the tenant and the tenant information was synchronized with Data Aggregator. Determine what your tenant ID is by making a web service call to the tenant web service:

     ```
     http://pc_host:8181/pc/center/webservice/tenants/tenantName/tenant_name
     ```

   - **primaryDNSAddress**
     The IP address of the primary name server for this domain. This tag is required, but an actual IP address is optional.
   - **primaryDNSPort**
     The port number that the primary name server uses. This tag is required, but an actual port number is optional.
   - **secondaryDNSAddress**
     The IP address of the secondary name server for this domain. This tag is required, but an actual IP address is optional.
   - **secondaryDNSPort**
     The port number that the secondary name server uses. This tag is required, but an actual port number is optional.
   - **isDnsProxyEnabled**
     (Optional) Indicates whether the proxy address is enabled for this IP domain.
     Default: false
   - **dnsProxyAddress**
     (Optional) Indicates the IP address of the DNS proxy server.

Default: null

For example:

```
<domain>

  <groupItemID>7</groupItemID>

  <itemDesc>{Description }</itemDesc>

  <itemName>{IP Domain Item Name}</itemName>

  <TenantID>8</TenantID>

  <primaryDNSAddress>0.0.0.0</primaryDNSAddress>

  <primaryDNSPort>0.0.0.0</primaryDNSPort>

  <secondaryDNSAddress>0.0.0.0</secondaryDNSAddress>

  <secondaryDNSPort>0.0.0.0</secondaryDNSPort>

  <isDnsProxyEnabled>false</isDnsProxyEnabled>

</domain>
```

6. Execute the web service call.
   The IP domain is created.

Provide SNMP Credentials

To enable SNMP polling of devices, provide SNMP credentials in an SNMPv1/SNMPv2c or SNMPv3 profile.

During inventory discovery, Data Collector uses SNMP profiles to determine what credentials it uses when accessing a device. NetOps Portal maintains a ranked list of profiles. During inventory discovery, each profile is tried for device access. The profile with the highest rank that can access a device is used.

For this scenario, you create an SNMPv3 profile.

> **NOTE**
> Community strings and credentials are encrypted when they are stored in NetOps Portal and when they are sent to Data Aggregator and Data Collector. NetOps Portal REST web services use the HTTP protocol instead of the HTTPS protocol. Therefore, the community strings and passwords are sent over the Internet in clear text, and, can be compromised. We recommend that you invoke this method only on the NetOps Portal host.

Follow these steps:

1. Enter a URL for the NetOps Portal REST web services API in the REST client. Use the following format:

   ```
   http://pc_host:8181/pc/center/webservice/profiles/
   ```

2. Set the Content-type to application/xml.
3. Provide a valid username and password for a user account that has administrator access to NetOps Portal.
4. Enter the following information within the "Body" text section:

- **name**
  Defines a name for the SNMP profile.

  NOTE
  Profile names must be unique, cannot be duplicated across SNMP versions, and are not case-sensitive.
- **port**
  (Optional) Identifies the port that is used to make SNMP connections to devices associated with this profile.
  Default: 161
- **userName**
  (Optional) Identifies the user for the profile, whose secret keys were used potentially to authenticate and encrypt the SNMPv3 packets. The user name is a character string.
  Default: the encrypted value of the name
- **context**
  (Optional) Specifies a collection of management information that is accessible by an SNMP entity. The context name is necessary for providing end-to-end identification and for retrieving data from an SNMPv3 agent. The context name is an octet string.
  Default: an empty string

  NOTE

  The Data Aggregator does not use Context Name on the SNMPv3 profiles to communicate with the device.
- **version**
  Specifies the version of SNMP that the profile uses. Specify Version3 for this scenario.
  Default: Version1or2C
- **securityLevel**
  (Optional) Specifies the security level to use. Enter one of the following values:
  - `NoAuthNoPriv`
  - `AuthNoPriv`
  - `AuthAndPriv`

  Default: `NoAuthNoPriv`
- **authProtocol**
  (Optional) Specifies the authentication protocol to use when contacting devices associated with this profile. Enter one of the following algorithms for authenticating SNMPv3 packets:
  - None (Do not attempt authentication.)
  - MD5 (Message Digest 5)
  - SHA (Secure Hash Algorithm)

  Default: None
- **authPassword**
  Specifies the password for authentication using SNMPv3 and the selected authentication protocol. The password must contain a minimum of eight characters. This value is required for the MD5 and the SHA authentication protocol values.
- **privProtocol**
  (Optional) Specifies the encryption protocol to use for data flows sent to any devices or servers that are associated with this profile, as follows:
  - None (Does not encrypt communications. Use only with NoPriv option.)
  - DES
  - AES (128-bit encryption)
  - Triple DES

  Default: None

Note: The privacy protocol option is only enabled when authentication is enabled for this profile.

- **privPassword**
  Defines the password that is used when exchanging encryption keys. This value is required for DES, AES, and Triple DES privacy protocols.
- **rank**
  (Optional) Specifies the rank of the profile in the global list of SNMP profiles.
  Default: 0
- **enabled**
  (Optional) Indicates whether the information in this profile is used when not explicitly assigned to a device. Select true.
  Default: true
- **tenantID**
  The internal (database) identifier for the tenant definition.
  Default: The ID of the Default Tenant

  NOTE
  This number must be the same number that you chose when you created the IP domain.

Example: No authentication and no privacy

```
<SnmpProfile>

    <name>Tokyo</name>

    <port>161</port>

    <userName>myuser</userName>

    <context></context>

    <version>Version3</version>

    <securityLevel>NoAuthNoPriv</securityLevel>

    <authProtocol>None</authProtocol>

    <authPassword>None</authPassword>

    <privProtocol>None</privProtocol>

    <privPassword>None</privPassword>

    <rank>4</rank>

    <enabled>true</enabled>

    <tenantID>7</tenantID>

</SnmpProfile>
```

Example: Authentication and no privacy

```
<SnmpProfile>
    <name>Brasil</name>
    <port>161</port>
    <userName>myuser</userName>
    <context></context>
    <version>Version3</version>
    <securityLevel>AuthNoPriv</securityLevel>
    <authProtocol>MD5</authProtocol>
    <authPassword>test</authPassword>
    <privProtocol>None</privProtocol>
    <privPassword>None</privPassword>
    <rank>3</rank>
    <enabled>true</enabled>
    <tenantID>7</tenantID>
</SnmpProfile>
```

Example: Authentication and privacy

```
<SnmpProfile>
    <name>Boston</name>
    <port>161</port>
    <userName>myuser</userName>
    <context></context>
    <version>Version3</version>
    <securityLevel>AuthAndPriv</securityLevel>
    <authProtocol>MD5</authProtocol>
```

```
        <authPassword>test</authPassword>

        <privProtocol>TripleDES</privProtocol>

        <privPassword>test</privPassword>

        <rank>1</rank>

        <enabled>true</enabled>

        <tenantID>7</tenantID>

    </SnmpProfile>
```

5. POST the following URL to create the profile:

```
POST http://pc_hostname:8181/pc/center/webservice/profiles/saveProfile/{true|false}
```

   – **{true|false}**
   Specifies a Boolean value for the rankTiesAscendingByDate parameter. **True** indicates that the profile you are adding is the last in rank order (as determined by the creation date of the SNMP profile).

   The XML returns true when the operation succeeds.
   The SNMP profile is automatically synchronized with Data Aggregator and is available for inventory discovery to use it.
   SNMP credentials are provided.

Assign Data Collector to the Tenant

Each Data Collector installation is assigned to a tenant and IP domain. Assign a Data Collector installation to the tenant you created previously.

> **NOTE**
> This scenario assumes that Data Collector is installed already. This scenario also assumes that you did not already run a discovery against this Data Collector. Once a discovery has occurred on a Data Collector instance, the Data Collector instance cannot be assigned to another tenant.

Follow these steps:

1. Determine what your tenant ID number is by making a web service call to the Data Aggregator tenant web service:

   ```
   http://da_host:8581/rest/tenants
   ```

   Note the tenant ID of the tenant that you want to assign to Data Collector.
2. Determine what your IP domain ID is by making a web service call to the Data Aggregator IP domain web service:

   ```
   http://da_host:8581/rest/tenant/tenantID/ipdomains
   ```

   – **tenantID**
   The ID of the tenant that you want to assign to the Data Collector instance. You determined this number in the previous step.

   Note the IP domain ID of the IP domain that you want to assign to Data Collector.
3. Determine the Data Collector ID number by making a web service call to the Data Aggregator Data Collector web service:

```
http://da_host:8581/rest/dcms
```

Note the ID number of the Data Collector installation you want to assign your tenant and IP domain to.

4. Make a web service call to the Data Aggregator Generic Data Collector web service:

```
http://da_host:8581/genericWS/dcm/collectorID
```

   – ***collectorID***
     The ID of the Data Collector installation you want to assign to your tenant and IP domain. You determined this ID in the previous step.
5. Select PUT for the "HTTP" Method.
6. Enter the following information within the "Body" text section:

```
<DCM>

  <DCM>

    <IP_DOMAIN_ID>IPdomainID</IP_DOMAIN_ID>

  </DCM>

</DCM>
```

   – ***IPdomainID***
     The IP domain ID for the tenant you want to assign Data Collector to. You determined this ID in step 2.

   Data Collector is assigned to your tenant.

Create a Discovery Profile and Run a Discovery

Discovery profiles specify how discovery operates in your Data Aggregator environment. Within a discovery profile, specify the IP addresses, IP address ranges, or host names you want to discover devices for.

You create a discovery profile and run a discovery in one operation when using the REST web services. When discovery is run, devices are discovered based on the discovery profile you create.

Follow these steps:

1. Access the Data Aggregator web server.
2. Open a REST client editor and select 'application/xml' as the 'Body Content-type' in the Body settings.
3. Find and make a note of the default tenant ID using the following URL:

```
GET http://da_host:8581/rest/tenants
```

4. View the XSD schema XML (which defines the structure of the discovery profile) by going to the following URL:

```
http://da_host:8581/rest/discoveryprofiles/XSD/getlist.xsd
```

5. Include IP addresses and host names so that devices in your network are discovered. Do one or more of the following actions:

- Type individual IP addresses for which you want to discover devices in the IP Address List field.
- Type the host names for which you want to discover devices in the Host List field.

  NOTE
  These fields accept comma-delimited values. Single quotes, double quotes, backward slashes, forward slashes, and ampersands are not permitted.

6. Set the following required attributes:
   - **RunStatus**
     Specifies whether to run the discovery. For this scenario, set the attribute to START.

     NOTE
     To rerun discovery later, update (PUT) the run status to START.
   - **Name**
     Specifies a descriptive name for the discovery profile. This field cannot contain single quotes, double quotes, backward slashes, forward slashes, and ampersands.
   - **IPDomainID**
     An internal ID that is used to identify IP domains. The Data Aggregator generates this number. Determine what your IP domain ID is by making a web service call to the IP domain web service:

     ```
     http://da_host:8581/rest/tenant/tenantID/ipdomains
     ```

7. (Optional) Limit the discovery to specific SNMP profiles using the following attributes:
   - **SNMPProfileIDList**
     Specifies a ranked list of SNMP Profile ItemIDs for use during discovery. This list is used only if UseListOfSnmpProfiles equals true.
   - **UseListOfSnmpProfiles**
     A boolean indicating whether a specific list of SNMP profiles is used. If false, the global list is used.
8. (Optional) Change the priority in which you want the discovered device to be named, using the following attribute:
   - **DeviceNameRankingList**
     Specifies the priority in which you want discovered devices to be named.
9. (Optional) Set the IcmpDiscoveryEnabled attribute to true to enable ICMP discovery.

Example: Discovery profile XML that includes an optional list of specific SNMP profiles.

```
<?xml version="1.0" encoding="UTF-8"?>

<DiscoveryProfile version="1.0.0">

  <ActivationStatus>true</ActivationStatus>

  <SNMPProfileIDList>

    <SNMPProfileID>478</SNMPProfileID>

    <SNMPProfileID>2239</SNMPProfileID>

  </SNMPProfileIDList>

  <UseListOfSnmpProfiles>true</UseListOfSnmpProfiles>

  <IPRangesList>
```

```
      <IPRanges>10.0.64.202-10.0.64.206</IPRanges>
    </IPRangesList>
    <HostNamesList>
      <HostNames>ahost</HostNames>
    </HostNamesList>
    <IPListList>
      <IPList>10.10.10.10</IPList>
    </IPListList>
    <RunStatus>START</RunStatus>
    <Item version="1.0.0">
      <Name>BigRange_TestProfileViaAPI</Name>
    </Item>
    <IPDomainMember version="1.0.0">
      <IPDomainID>285</IPDomainID>
    </IPDomainMember>
    <DeviceNameRankingList>
      <DeviceNameRanking>{http://im.ca.com/inventory}
       ManageableDevice.SystemName</DeviceNameRanking>
      <DeviceNameRanking>{http://im.ca.com/inventory}
        Device.HostName</DeviceNameRanking>
      <DeviceNameRanking>{http://im.ca.com/inventory}
       Device.PrimaryIPAddress</DeviceNameRanking>
    </DeviceNameRankingList>
  <IcmpDiscoveryEnabled>true</IcmpDiscoveryEnabled>
  </DiscoveryProfile>
```

10. Save and run the new discovery profile by entering the following URL:

```
POST http://da_host:8581/rest/tenant/tenantID/discoveryprofiles
```

a. **tenantID**
The internal database identifier for the tenant definition. The tenant ID was assigned when you created the tenant and the tenant information was synchronized with Data Aggregator. Determine what your tenant ID is by making a web service call to the tenant web service:
`http://da_host:8581/rest/tenants`

The discovery profile is created and discovery runs.
All discovered devices are automatically added to the appropriate out-of-the-box device collection or another user-created device collection.

Review the Discovered Devices and Instances

After you create and run a discovery profile, verify your results. View the summary of the number of new pingable and manageable devices that were discovered. The summary also includes details about the discovered devices, including IP address, type, vendor association, SNMP profiles, and model type.

Follow these steps:

1. View a list of discovery profiles on the Data Aggregator web server using the following URL:

   ```
   GET http://da_host:8581/rest/discoveryprofiles
   ```

   A list of discovery profiles and their instance IDs is returned.
2. Find the discovery instance ID for the discovery profile that you created previously.
3. View the discovery instance details, using the following URL:

   ```
   GET http://da_host:8581/rest/discoveryinstances/instance_ID
   ```

   The instance XML returns information about new and existing devices and SNMP profiles that were tested.
   Example:

```
<?xml version="1.0" encoding="UTF-8"?>

<DiscoveryInstance version="1.0.0">

    <ID>236</ID>

   <IPSweepTotalSuccess>5</IPSweepTotalSuccess>

   <CompletionTime>Thu Apr 12 12:36:35 CDT 2012</CompletionTime>

  <ExistingFoundDevicesList>

      <ExistingFoundDevices>241</ExistingFoundDevices>

      <ExistingFoundDevices>239</ExistingFoundDevices>

      <ExistingFoundDevices>240</ExistingFoundDevices>
```

```
    <ExistingFoundDevices>238</ExistingFoundDevices>
</ExistingFoundDevicesList>
<IPSweepCompletionTime>Thu Apr 12 12:36:35 CDT 2012</IPSweepCompletionTime>
<ExistingFoundManageableDevicesList>
    <ExistingFoundManageableDevices>241</ExistingFoundManageableDevices>
    <ExistingFoundManageableDevices>239</ExistingFoundManageableDevices>
    <ExistingFoundManageableDevices>240</ExistingFoundManageableDevices>
    <ExistingFoundManageableDevices>238</ExistingFoundManageableDevices>
</ExistingFoundManageableDevicesList>
<StartTime>Thu Apr 12 12:36:31 CDT 2012</StartTime>
<NewlyCreatedDevicesList>
    <NewlyCreatedDevices>383</NewlyCreatedDevices>
 </NewlyCreatedDevicesList>
 <TestedCommProfilesList>
     <TestedCommProfiles>199</TestedCommProfiles>
      <TestedCommProfiles>198</TestedCommProfiles>
  </TestedCommProfilesList>
  <IPSweepStartTime>Thu Apr 12 12:36:31 CDT 2012</IPSweepStartTime>
  <ProgressPercentage>100</ProgressPercentage>
  <IPSweepTotal>7</IPSweepTotal>
  <ProfileID>235</ProfileID>
  <NewlyCreatedManageableDevicesList>
       <NewlyCreatedManageableDevices>383</NewlyCreatedManageableDevices>
   </NewlyCreatedManageableDevicesList>
   <PingResponseDeviceCount>5</PingResponseDeviceCount>
```

```
        <CompletionStatus>SUCCESS</CompletionStatus>

        <Item version="1.0.0">

            <CreateTime>Thu Apr 12 12:36:31 CDT 2012</CreateTime>

         </Item>

    </DiscoveryInstance>
```

4. (Optional) Use the following URL to review information about specific devices (new or existing):

   ```
   GET http://da_host:8581/rest/devices/device_ID
   ```

   – ***device_ID***
   Specifies the ID of the referenced datapoint, such as devices. Use the ID numbers that were returned when you viewed the discovery instance details.
5. (Optional) Review information about the SNMP profiles that were tested during the discovery, using the following URL:

   ```
   GET http://da_host:8581/rest/profiles/profile_ID
   ```

   – ***profile_ID***
   Specifies the ID of the SNMP profile. The <TestedCommProfilesList> section in the XML displays the SNMP profile IDs.

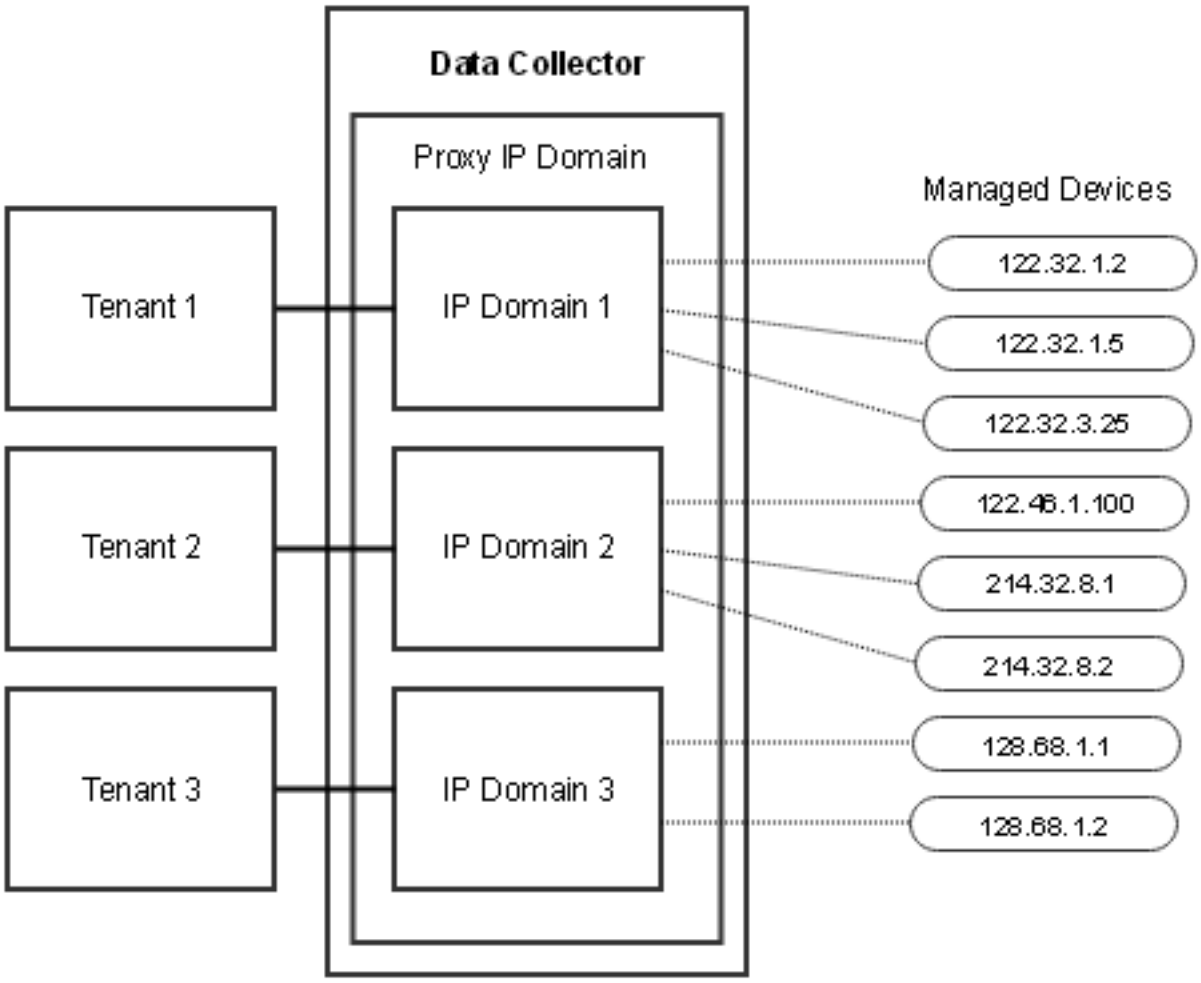
You have successfully used REST web services to configure tenant information. For automated tenant configuration, you would write an application or script that leverages the web services in this scenario.

Tenant-Agnostic Data Collectors

In a standard tenant deployment, each tenant has a dedicated Data Collector. For multiple tenants that reside in the same IP routing space, DX NetOps Performance Management can be configured to use fewer Data Collectors.

The following diagram illustrates the basic architecture of a tenant-agnostic Data Collector:

Figure 59: Multi-tenant Data Collector Architecture



- The Data Collector is associated with the proxy IP domain. This domain represents the IP domain of the managed environment, meaning that IP addresses must be unique across all tenants in the environment.

  TIP

  The IP proxy domain is the only new object to create this feature. This object must be created through REST.
- Multiple IP domains are associated with the Data Collector within the proxy IP domain.
- Each IP domain is associated with a single tenant.
- The Data Collector sends requests to the managed devices defined in each tenant.

Limitations

A tenant-agnostic Data Collector configuration has the following limitations:

- IP addresses in tenant IP domains cannot overlap. IP addresses must be unique across all tenants.

WARNING

The owner of the environment is responsible for ensuring IP address uniqueness. Without unique IP addresses, tenants can monitor and manage devices that belong to other tenants. To ensure IP address uniqueness, enforce limited IP ranges in discovery profiles.

- Devices cannot be migrated between tenants.
- Existing tenant deployments cannot be migrated to tenant-agnostic Data Collectors.
- Tenant-agnostic Data Collectors cannot host DX NetOps Mediation Manager (CAMM) or DX NetOps Virtual Network Assurance (CA VNA). For each tenant, a separate Data Collector is required for CAMM and CA VNA.

Configure the Data Collector Though REST

To configure a tenant-agnostic Data Collector, use REST to create the required items and associations. This process uses NetOps Portal and Data Aggregator REST.

WARNING

If you script these steps, account for synchronization time between steps. Synchronization can take up to 5 minutes.

Get Item IDs

Several item IDs are required to configure the Data Collector for multiple tenants. Get the item IDs before you begin configuration. Use the GET method with your preferred REST client for the following item IDs:

- Data Collector IDhttp://*da_host*:8581/rest/dcms
- Pseudo Tenant ID

```
http:// da_host :8581/rest/pseudotenants
```

NOTE

The pseudo tenant proxy (Pseudo Tenant Proxy) is predefined. The pseudo tenant works with the proxy IP domain to support multiple tenants and IP domains on a single Data Collector.

Configure the Data Collector

Configure the Data Collector to handle multiple tenants. Complete this procedure for each tenant-agnostic Data Collector.

For the bold IDs, use the IDs from the previous step.

Follow these steps:

1. Create a proxy IP domain:
   - **Method:** POST
   - **Endpoint:** `http://da_host:8581/rest/tenant/ {pseudo_Tenant_ID} /proxyipdomains/`
   - **Body:**

```
<ProxyIPDomain version="0.0.0">
    <IPDomain version="0.0.0">
        <Name>Proxy IP Domain</Name>
        <Description>Description for Proxy IP Domain</Description>
        <DNSProxyEnabled>false</DNSProxyEnabled>
        <DNS1Address>0.0.0.0</DNS1Address>
        <DNS2Address>0.0.0.0</DNS2Address>
        <DNS1Port>0</DNS1Port>
        <DNS2Port>0</DNS2Port>
```

```
        </IPDomain>
    </ProxyIPDomain>
```

2. Get and note the proxy IP domain ID:
   - **Method:** GET
   - **Endpoint:** *http://da_host:8581/rest/proxyipdomains*
3. Associate the proxy IP domain with the Data Collector:
   - **Method:** PUT
   - **Endpoint:** `http://da_host:8581/genericWS/dcm/` ***{DataCollectorID}***
   - **Body:**

```
<DCM>
    <IP_DOMAIN_ID> {proxyDomainId} </IP_DOMAIN_ID>
</DCM>
```

Add Tenants to the Proxy IP Domain

Create tenants and assign those tenants to the Data Collector. Complete this procedure once for each tenant.

For more information about the tenant and IP domain XML, see Automate Tenant Configuration with REST Web Services.

> **TIP**
>
> You can create the required tenants and the IP domains in those tenants, through the NetOps Portal UI. For more information see Manage Tenants and IP Domains. If you create these items in the UI, start this procedure from step 4.

Follow these steps:

1. Create the tenant:
   - **Method:** POST
   - **Endpoint:** `http://pc_host:8181/pc/center/webservice/tenants/`
   - **Body:**

```
<tenant>
  <tenantName>Tenant name</tenantName>
  <tenantDesc>Tenant description</tenantDesc>
  <accountIdentifier>AccountNumber</accountIdentifier>
  <status>Enabled</status>
  <removable>false</removable>
  <theme>CA-Blue</theme>
  <defaultCulture>en-US</defaultCulture>
</tenant>
```

2. Get and note the tenant ID:
   - **Method:** GET
   - **Endpoint:** `http://pc_host:8181/pc/center/webservice/tenants/tenantName/tenant_name`
3. Create the tenant IP domain:
   - **Method:** POST
   - **Endpoint:** `http://pc_host:8181/pc/center/webservice/domains/`
   - **Body:**

```
<domain>
```

```
  <itemDesc>IP domain description</itemDesc>
  <itemName>IP domain name</itemName>
  <TenantID> {tenantID} </TenantID>
  <primaryDNSAddress>0.0.0.0</primaryDNSAddress>
  <primaryDNSPort>0.0.0.0</primaryDNSPort>
  <secondaryDNSAddress>0.0.0.0</secondaryDNSAddress>
  <secondaryDNSPort>0.0.0.0</secondaryDNSPort>
  <isDnsProxyEnabled>false</isDnsProxyEnabled>
</domain>
```

4. Get the tenant IP domain ID:
   - **Method:** GET
   - **Endpoint:** `http://da_host:8581/rest/ipdomains`
5. Associate the proxy IP domain with the tenant IP domain:
   - **Method:** PUT
   - **Endpoint:** `http://da_host:8581/rest/ipdomains/{tenantIPDomainID}`
   - **Body:**

```
<IPDomain version="0.0.0">
    <ProxyIPDomain> {proxyIpDomainID} </ProxyIPDomain>
</IPDomain>
```

Configure the Tenant

Use the standard methods to configure monitoring in the tenant space.

Multi-tenancy and CA Application Delivery Analysis

CA Application Delivery Analysis (CA ADA) does not offer a native multi-tenancy feature. However, it does support IP domains.

CA ADA also supports data segregation per user account, so tenant users only see the data within their associated tenant.

CA ADA can observe duplicate IP traffic, which occurs in a managed service provider (MSP) environment. The provider can host an application on a single server for multiple customers whose environments contain overlapping client IP addresses.

You enable CA ADA to identify separate IP traffic during data collection setup. As you verify and modify data collection parameters, assign the same IP domain to the appropriate:

- Monitor feeds
- Client networks
- Servers or server subnets

With the same IP domain assignments for these feeds, CA ADA reports on the application traffic between a client and a server by domain.

Since applications are domain-independent, you are not required to define the same application twice to report on application performance across domains. However, to set different thresholds for application performance, performance OLAs, and availability OLAs, create an application for each IP domain.

If you do not need to separate duplicate IP traffic, you can use the DNS settings in the Default Domain to query DNS and resolve the hostname of a CA ADA server. Otherwise, CA ADA uses the monitor feed that is assigned to the server to resolve the hostname.

View a List of Domains in CA Application Delivery Analysis

You can view a list of domain definitions and current domain associations in the Administration section of the CA ADA management console.

NOTE

Any items that are not assigned to a specific domain in a data source are included in the Default Domains group. In the data source, they appear to be associated with the Default Domain.

Follow these steps:

1. Click the Administration tab in the management console.
2. Click Data Monitoring, Domains in the Show Me menu.
   The Domains page opens.
3. (Optional) View the DNS settings for a domain by clicking the magnifying glass symbol in the View column.
   The Domain Properties page opens.
4. Verify the properties.
5. Click OK when you have finished.
   The Domains page appears.

Assign a Domain to a Monitor Feed

You can instruct each Standard Monitor to associate the items that it monitors with a custom domain as part of CA ADA collection device setup.

NOTE

Any managed items that are not associated with a custom IP domain by a data source are associated with the Default Domain. This assignment is transparent to users who are not deploying custom IP domains.

Follow these steps:

1. Click the Administration tab in the management console.
2. Click Data Monitoring, Monitoring Devices in the Show Me menu.
   The CA ADA Monitors list page opens.
3. Click Edit to edit a multifunction monitoring device, such as a Standard or Multi-Port Monitor.
   The Monitor Properties page opens.
4. Scroll down to the Monitor Feeds list.
5. Click to edit a monitor feed.
6. Select a custom IP domain.
7. Click Update.
   All items detected by this monitor feed are automatically associated with the selected IP domain.

Assign a Domain to a Client Network

After you add a client network, you cannot change its IP domain association. If you need to change the assigned IP domain, you must delete the network, and then add it to the correct domain.

NOTE

Any managed items that are not associated with a custom IP domain by a data source are associated with the Default Domain. This assignment is transparent to users who are not deploying custom IP domains.

Follow these steps:

1. Click the Administration tab in the management console.
2. Click Data Monitoring, Networks in the Show Me menu.
   The Networks List page opens.
3. Select the IP domain from the list.
4. Click Add Network.
5. Enter the required information to add the network.
6. Click OK.

Assign a Domain to a Server or Server Subnet

After you add a server or server subnet, you cannot change its IP domain. If you add a server or server subnet to the wrong IP domain, you must delete it, and then add it to the correct domain.

> **NOTE**
>
> Any managed items that are not associated with a custom IP domain by a data source are associated with the Default Domain. This assignment is transparent to users who are not deploying custom IP domains.

Follow these steps:

1. Click the Administration tab in the management console.
2. Click Data Monitoring, Servers in the Show Me menu.
   The Server Subnet List and Server List page opens.
3. Select the IP domain from the list.
4. Supply the information to add the Server or Server Subnet.
5. Click OK.

Other Deployment Considerations

CA ADA supports the multi-tenancy features in CA NetOps Portal. However, take care when selecting user account product privileges. The product privilege to a data source enables a user to drill down from a view back to the source of the data.

Assuming that you have carefully segregated data from different customer environments into separate tenants, you probably want to prevent users from returning to the CA ADA interface. In that interface, tenant separation is not applied, and all data is available for viewing in reports. Product privileges are set in the Add or Edit User Account wizard.

> **WARNING**
> Assign the 'User' product privilege to any user who does not require access to all data in the CA ADA data source.

Multi-tenancy and CA Network Flow Analysis

The Network Flow Analysis (NFA) data source does not offer a native multi-tenancy feature. However, the harvesters and routers can associate monitored items with the appropriate tenant by means of the IP domain.

Interfaces and CVIs inherit their initial tenant-domain setting from the parent router and harvester. The setting is inherited when the parent harvester is added, and the router and interfaces first become active. If the harvester is not associated with a custom domain, the routers and interfaces are assigned to the Default Domain as they become active.

You can edit the settings for interfaces and CVIs to associate them with any tenant and domain at any time. The setting does not have to match the parent router or harvester.

Changing this setting can affect which operators have access to the interface's data. The setting does not affect which SNMP profiles are used for polling. The router tenant determines the set of SNMP profiles for polling.

Follow these steps:

1. Open the Active Interfaces page:
   a. Select Administration from the NFA console menu.
      The Administration page opens.
   b. Select Interfaces: Physical & Virtual from the Administration menu.
      The Active Interfaces page opens.
2. Select the check box next to one or more interfaces that you want to associate with a tenant and domain.
   – To search for parent routers, interfaces, or CVIs, enter all or part of a router IP address, a router or interface name, or an interface description in the Search field, and then click Search. Expand the router details.
   – To navigate to an interface or CVI manually, go to the page that contains the parent router and click the arrow next to the router name. The router details appear, showing the interfaces and CVIs.
3. Click **Edit**.
   The editing dialog opens. The Domain selection list is included in the dialog only if multiple domains exist.
4. Select a tenant/domain option from the Domain list.
5. Click **Save**.
   The dialog closes. The changes are shown on the Active Interfaces page.

   NOTE
   You can also change the tenant-domain setting for harvesters and routers.

Interface Domain Changes

When an administrator allocates the interfaces from a single router to multiple MSP customer domains, interfaces are often switched to different IP domains. Network Flow Analysis detects interface domain changes. These changes are not applied to data being sent to Data Aggregator, however, because Data Aggregator only identifies interfaces at the device level.

As a result, when interfaces are switched to a different domain after monitoring and data collection have begun, the global administrator sees two different interface items for each affected interface. The two interfaces continue to be monitored in separate IP domains, and their data is not aggregated. In this situation, individual tenant users do not see any problem. Device-level statistics for the routers or switches that contain these interfaces are not affected. Only the global administrator sees two interfaces to represent a single monitored interface.

Multi-tenancy and CA Unified Communications Monitor

CA Unified Communications Monitor (CA UCM) supports the multi-tenancy features of CA NetOps Portal (CAPC). You perform all tenant and IP domain configuration in CAPC. The collector then associates monitored items with the appropriate tenant by means of the IP domain.

In the CA UCM management console, you can instruct collectors to associate the items that they discover with custom domains in CAPC. The act of creating a single custom domain in CAPC enables domain associations for Locations, voice gateways, and call servers in any registered data sources.

Items appear with domain designations as soon as they are discovered from call traffic. Items discovered previously do not receive retroactive associations.

Locations are automatically associated with IP domains by the subnets that they contain. To preserve the flow of data collection and the appropriate association of data with IP domains, take care when moving Locations to new IP domains. Follow the procedure provided in the CA UCM Online Help to change IP domain assignments.

NOTE

: Any managed items that are not associated with a custom IP domain by a data source are associated with the Default Domain. This assignment is transparent to users who are not deploying custom IP domains.

Instruct collectors to associate items with custom IP domains.

Follow these steps:

1. Click Administration, Data Collection, Collectors.
2. Edit each collector to select its domain for the IP Domain parameter.
3. Reload the collectors to send them the domain information.
   Domains are populated with managed items after the next product synchronization.

Multi-tenancy and CA Spectrum

When you register the DX NetOps Spectrum data source, database synchronization occurs. DX NetOps Spectrum retrieves a list of IP domains. All IP domain definitions are sent over, regardless of their association with individual tenants. OneClick displays these NetOps Portal IP Domain models in the same area as DX NetOps Spectrum Global Collections in the OneClick Navigation panel. The IP Domain models have the same names as the IP domain definitions:

- Use these IP domains to determine which models are synchronized with DX NetOps Performance Management. To include a device model in DX NetOps Performance Management monitoring and make it available in NetOps Portal dashboards, add it to an IP domain in OneClick.
- Add only device models that should be synchronized with NetOps Portal. When the device models are synchronized, they are associated with the corresponding IP domain in NetOps Portal. The NetOps Portal IP domain may belong to the Default Tenant, or to any custom tenant. Do not add interface models. Device interfaces are automatically added to the IP domain with which their device is associated.
- DX NetOps Spectrum devices can be members of only a single IPDomain model type. If you attempt to add a model to multiple IP domains, you see an error message.

Performance Center Administration

To maximize the value of reporting and to make DX NetOps Performance Management fit the needs of your business, customize NetOps Portal.

To customize NetOps Portal, do any of the following tasks:

- To let users send reports by email, either manually or on a schedule, configure an email server. For more information, see Set the Email Server.
- To change the appearance of exported reports, apply a new theme to a NetOps Portal tenant. For more information, see Customize a Theme.
- To change the display settings, such as view suppression and item name display names and aliases, customize the display settings. For more information, see Customize Your User Settings.
- To let the NetOps Portal daemons run effectively, modify the maximum memory usage. For more information, see Modify Maximum Memory Usage for Each Performance Center Service.
- To avoid losing valuable data, back up the NetOps Portal database before you upgrade to a new release. Also create a backup archive of the database on a weekly basis. For more information, see Back Up Performance Center.

Back Up Performance Center

Back up NetOps Portal before an upgrade. To avoid losing valuable data, back up and archive NetOps Portal regularly.

Back Up the NetOps Portal Database

Create a backup archive of the current database anytime that you plan to reinstall or upgrade the software. We recommend that you create a backup archive on a weekly basis.

Follow these steps:

1. Log in to the NetOps Portal host as 'root', or use the 'sudo' account that you configured for the installation.
2. Estimate the size of the backup:

   ```
   du –hs MySql/data
   ```

3. Stop all the NetOps Portal services, using the following commands:

   ```
   service caperfcenter_console stop

   service caperfcenter_devicemanager stop

   service caperfcenter_eventmanager stop

   service caperfcenter_sso stop
   ```

4. Change to a directory where you want to save the database archive:

   ```
   cd /$backupDir
   ```

   Use any secure location with sufficient space for the backup directory.
5. Create a MySQL dump of the database using the following command:

   NOTE

   If you leave out the optional password syntax from the following command, you are prompted for the password.

   ```
   /opt/CA/MySql/bin/mysqldump --routines -u root -ppassword  netqosportal > $backupDir/
   netqosportal.sql
   ```

   NOTE
   During the installation, you are prompted to 'Select a Location for the MySQL Data Directory'. The default location is /opt/CA/MySql/data.
6. Create a MySQL dump of Event Manager data:

   ```
   /opt/CA/MySql/bin/mysqldump -u root -ppassword em > $backupDir/em.sql
   ```

7. Compress these backup files to save space using the following commands:

   ```
   tar czvf netqosportal.tgz netqosportal.sql

   tar czvf em.tgz em.sql
   ```

8. Remove the uncompressed MySQL dump files using the following commands:

```
rm netqosportal.sql

rm em.sql
```

9. Start the NetOps Portal services on the new system as follows:
   a. Start the SSO service:

```
service caperfcenter_sso start
```

   b. Wait one minute, then start the event manager and device manager:

```
service caperfcenter_eventmanager start


service caperfcenter_devicemanager start
```

   c. Wait one minute, then start the console service:

```
service caperfcenter_console start
```

Back Up Single Sign-On Configuration Files

If you configure single sign-on, backup the configuration settings. Use rsync or another preferred method, such as a script, to back these files up automatically.

Back up the following files:

- `/opt/CA/PerformanceCenter/sso/start.ini`
- `/opt/CA/PerformanceCenter/PC/start.ini`

Back up the following directories:

- `/opt/CA/PerformanceCenter/sso/webapps/sso/configuration`
- `/opt/CA/PerformanceCenter/sso/etc`
- `/opt/CA/PerformanceCenter/sso/conf`
- `/opt/CA/PerformanceCenter/PC/etc`
- `/opt/CA/PerformanceCenter/PC/conf`

If you have enabled SSL, back up the following files:

- `/opt/CA/PerformanceCenter/sso/start.d/ssl.ini`
- `/opt/CA/PerformanceCenter/PC/start.d/ssl.ini`

Back Up Custom Settings

Custom OpenAPI applications reside on the NetOps Portal host. If your deployment includes custom OpenAPI applications, back up the application folder:

- `/opt/CA/PerformanceCenter/PC/webapps/pc/apps`

If you use custom logos for your themes, back up the following files:

- /opt/CA/PerformanceCenter/PC/webapps/pc/css/CA-Blue/images/customLogo
- /opt/CA/PerformanceCenter/PC/webapps/pc/css/CA-Gray/images/customLogo

Back Up Reports

Full PDF reports are temporarily saved for download.

Back up the following directory:

- /opt/CA/PerformanceCenter/DM/repository

Back Up Script Notification Actions

If you have script notification actions, back up the following directory:

- /opt/CA/PerformanceCenter/NotificationScripts

Customize a Theme

DX NetOps Performance Management supports user interface themes. These themes primarily describe the colors that are used to draw the user interface and when generating PDF files. Several themes are shipped with the product (CA-Blue, CA-Gray, CA-White). You can define more site-specific themes, which provide different color schemes.

WARNING

Custom themes require reconfiguration every time a new version of DX NetOps Performance Management is installed. Carefully consider whether a new theme is necessary. Only experienced administrators should customize themes.

NOTE

If you only want to apply a unique logo, simply manage your theme settings and apply a new logo.

Create a New Theme

The theme directories that are shipped with the product are CA-Blue, CA-Gray, and CA-White. To define more site-specific themes, duplicate one of the existing theme directories using an appropriate naming convention. Then deploy the theme in NetOps Portal.

NOTE

Define the custom theme directory name as "XXXXXXX-Theme" where XXXXXXX is some identifier that the user can define. For example, "CA-Blue-Modified-Theme". The name should exclude any spaces or special characters.

If the theme directory name is appended with "-Theme," it is available for deployment and it is preserved during an upgrade. Custom theme directories are not preserved after an uninstall. Custom themes require reconfiguration every time a new version of DX NetOps Performance Management is installed.

NOTE

You can use the following procedure to change the user interface coloring, icon coloring, or the PDF font and coloring.

Follow these steps:

1. Go to the following location:
   /PC InstallDirectory/PerformanceCenter/PC/webapps/pc/css
2. Package one of the existing standard theme directories. For example, zip up the CA-Blue theme.

3. Copy the ZIP file to your local computer, unzip it, and name it appropriately.

   NOTE

   Define the custom theme directory name as "XXXXXXX-Theme" where XXXXXXX is some identifier that the user can define. For example, "CA-Blue-Modified-Theme". The name should exclude any spaces or special characters.
4. Do any of the following tasks:
   - To change the user interface coloring, go to the `/ThemeDirectory/includes` directory and edit the `capc_theme.css` file. When you are done editing `capc_theme.css` , copy and replace `capc_theme.css` to `capc_theme-min.css` .
     Example:
     To change the background color, locate the comment "page background color" and replace the color #f7f7f7 with "pink". After you save the file, users using your theme have a pink page background color instead of a pale gray one.
   - To change the icon coloring, go to the `/ThemeDirectory /images` directory and edit the relevant files in a graphics editor.
     Example:
     To change all the gray icons to be white icons, edit the `IconsSecondary.png` file.
   - To change the PDF font and font coloring, go to the `/ThemeDirectory /xsl/pdf` directory and edit the `defaultPageContent.xml` file.
5. Package the customized theme directory and name it appropriately.
6. From the NetOps Portal UI, manage your theme settings, and deploy the new theme.

Manage Your Theme Settings

To change the logo that appears in PDF report headers or apply a different existing theme, edit your theme settings. By default, all themes use the CA Technologies corporate logo. As a global administrator, you apply a theme to a specific tenant. If you do not deploy multi-tenancy, the theme applies to the default tenant. For more information, see Manage Tenants.

Apply a Logo

You can upload and apply a unique logo to the PDF output.

Follow these steps:

1. Log in as a user with the Administrator role.
2. Hover over **Administration**, and click **Group Settings: Themes**.
   The Theme Settings page opens.
3. To upload a unique logo, click **Browse** and select the image file for your logo.
4. To apply the logo the logo to a theme, select the theme from the **Apply to theme** drop-down list.
   If you do not deploy multi-tenancy, select **All Themes**.
5. Click **Upload Image**.
6. Edit the tenant to select the theme that you modified.

   NOTE

   To restore the default theme settings, click **Reset Images**.

Deploy a Theme

After you create a theme, you must deploy the theme to use use it.

NOTE
Theme deployment occasionally hangs when you use the Chrome browser. If this known limitation occurs, deploy the theme using Firefox or Internet Explorer as your browser.

Follow these steps:

1. Log in as a user with the Administrator role.
2. Hover over **Administration**, and click **Group Settings: Themes**.
   The Theme Settings page opens.
3. In the Deploy a Theme section, click **Browse**, and select the new theme.

   NOTE

   The maximum filesize is 100 MB. The file must have a ZIP file extension. One top-level theme directory should exist in the ZIP file. The name of the theme directory must end with "-Theme".
4. To replace an existing theme, select **Replace existing themes**.
5. Click **Add**.

Manage Authentication Requirements

By default, user passwords must meet the following requirements:

- Be different from the username
- Minimum length of 8 characters
- Maximum length of 30 characters
- Contain at least 3 of the following types of characters:
  - Special characters
  - Uppercase letters (A-Z)
  - Lowercase letters (a-z)
  - Numbers (0-9)

For more information, see Customize Your User Settings.

By default, non-LDAP passwords expire after 105 days. When passwords expire, access to REST and the OpenAPI is blocked. The next time users log in, they must change their passwords.

You can enable a setting that disables users after multiple failed login attempts within a timeframe. By default, 6 failed login attempts within 3 minutes disable the user. In addition to disabling the user, you can also block the IP address of the user for a specified amount of time. This functionality is *not* an intrusion detection system.

Configure Authentication Settings

If desired, you can change authentication requirements using the Single Sign-On Configuration/Performance Center Tool.

Follow these steps:

1. Log in to the Performance Center host.
2. Navigate to the Performance Center directory:
   ```
   cd PC_Install_Directory/PerformanceCenter
   ```
3. Launch the SSO Configuration utility:
   ```
   ./SsoConfig
   ```
4. Select CA Performance Center.
5. Select and run option 8: Performance Center Local Password Authentication.

6. Complete the following prompts:
   - **1. Enforce password requirements**
     Specify whether to enforce the password requirements.
     Default: Enabled
   - **2. Allow REST to create users with usernames and passwords that match**
     Specify whether to allow the REST web services to create users with usernames and passwords that match. If disabled, users are created with randomized passwords.
     Default: Enabled
   - **3. Minimum password length**
     Specify the minimum password length.
     Default: 8
   - **4. Password lifespan**
     Specify when passwords expire. To disable password expiration, specify 0. If disabled, passwords never expire.
     Default: 105 days
   - **5. Disable password expiration for a specific user**
     Specify the user to disable password expiration for. The password for the specified user never expires.
   - **6. Enable password expiration for a specific user**
     Specify the user to enable password expiration for. The password lifespan is applied.
   - **7. Expire password for a specific user**
     Specify the user to expire the password for. The specified user must change their password the next time they log in.
   - **8. Expire the password for all users**
     Confirm whether to expire the passwords for all users. If yes, all users must change their passwords the next time they log in.
   - **9. Failed login attempts before blocking**
     Specify the number of failed login attempts before blocking the user.
     Default: 6
   - **10. Timeframe for failed login attempts**
     Specify the timeframe that the failed login attempts must occur before blocking the user.
     Default: 3 minutes
   - **11. Disable user after failed login attempts**
     Specify whether multiple failed login attempts within a timeframe disable the user.
     Default: Disabled
   - **12. Number of minutes to block IP address after failed login attempts**Specify the number of minutes to block an IP address after multiple failed login attempts within a timeframe.
     Default: 0
7. Enter q to close the Single Sign-On Configuration Tool.
   The Single Sign-On Configuration Tool closes.

Enable a Disabled User

If a user is disabled after multiple failed login attempts, you can enable them using the Single Sign-On Configuration/ Performance Center Tool.

You can also manage user account status in the Performance Center administration UI. For more information, see Manage User Accounts.

Follow these steps:

1. Log in to the Performance Center host.
2. Navigate to the Performance Center directory:
   ```
   cd PC_Install_Directory/PerformanceCenter
   ```

3. Launch the SSO Configuration utility:
   ```
   ./SsoConfig
   ```
4. Select CA Performance Center.
5. Select and run option 9: Enable or Disable a user account.
6. Complete the prompts.
7. Enter q to close the Single Sign-On Configuration Tool.
   The Single Sign-On Configuration Tool closes.

Migrate Performance Center

If you upgrade the Linux OS release, you can use the following procedure to migrate NetOps Portal to a new host with a new IP address and hostname.

> **NOTE**
>
> You must upgrade the existing system to the product version you are migrating to before migrating. This migration procedure copies over NetOps Portal and does not require you to install NetOps Portal on the new host.

Prepare for the Migration

Follow these steps:

1. Verify that the new host meets the system requirements. For more information, see Review Installation Requirements and Considerations and the DX NetOps Performance Management Sizing Tool
2. Prepare the new host. For more information, see Prepare to Install CA Performance Management.
3. If you installed NetOps Portal on the original system as *sudo user,* add the command alias on the new system:
   a. Locate the following file on the NetOps Portal host:
      ```
      /etc/sudoers
      ```
   b. Add a command alias with the following permissions to the file:

- /tmp/CAPerfCenterSetup.bin
- /etc/init.d/caperfcenter_console **(for RHEL 6.x)**
- /etc/init.d/caperfcenter_devicemanager **(for RHEL 6.x)**
- /etc/init.d/caperfcenter_eventmanager **(for RHEL 6.x)**
- /etc/init.d/caperfcenter_sso **(for RHEL 6.x)**
- /etc/init.d/mysql
- /opt/CA/PerformanceCenter/PC/bin/caperfcenter_console **(for RHEL 7.x, SLES, or OL)**
- /opt/CA/PerformanceCenter/DM/bin/caperfcenter_devicemanager **(for RHEL 7.x, SLES, or OL)**
- /opt/CA/PerformanceCenter/EM/bin/caperfcenter_eventmanager **(for RHEL 7.x, SLES, or OL)**
- /opt/CA/PerformanceCenter/sso/bin/caperfcenter_sso **(for RHEL 7.x, SLES, or OL)**
- /opt/CA/PerformanceCenter/Tools/bin/npcshell.sh
- /opt/CA/PerformanceCenter/SsoConfig
- /opt/CA/PerformanceCenter/Uninstall_MySql
- /opt/CA/PerformanceCenter/Uninstall_PerformanceCenter
- /opt/CA/PerformanceCenter/Uninstall_SSO
- /sbin/service
- /opt/CA/MySql/bin/mysql
- /opt/CA/MySql/bin/mysqldump
- /opt/CA/PerformanceCenter/sso
- /opt/CA/PerformanceCenter/PC
- /opt/CA/PerformanceCenter/PC/webapps/pc/apps
- /opt/CA/PerformanceCenter/PC/webapps/pc/css/CA-Blue/images
- /opt/CA/PerformanceCenter/PC/webapps/pc/css/CA-Gray/images
- /usr/bin/vim
- /opt/CA/jre/bin/keytool
- /opt/CA/PerformanceCenter/DM/repository
- /opt/CA/PerformanceCenter/NotificationScripts

Separate the permissions with commas and place all permissions on a single line.

Example:

```
Cmnd_Alias CAPERFCENTER = /tmp/CAPerfCenterSetup.bin,/opt/CA/
PerformanceCenter/PC/bin/caperfcenter_console,/opt/CA/PerformanceCenter/
DM/bin/caperfcenter_devicemanager,/opt/CA/PerformanceCenter/EM/bin/
caperfcenter_eventmanager,/opt/CA/PerformanceCenter/sso/bin/caperfcenter_sso,/
etc/init.d/mysql,/opt/CA/PerformanceCenter/Tools/bin/npcshell.sh,/opt/CA/
PerformanceCenter/SsoConfig,/opt/CA/PerformanceCenter/Uninstall_MySql,/opt/
CA/PerformanceCenter/Uninstall_PerformanceCenter,/opt/CA/PerformanceCenter/
Uninstall_SSO,/sbin/service,/opt/CA/MySql/bin/mysql,/opt/CA/MySql/bin/
mysqldump,/opt/CA/PerformanceCenter/sso,/opt/CA/PerformanceCenter/PC,/opt/CA/
PerformanceCenter/PC/webapps/pc/apps,/opt/CA/PerformanceCenter/PC/webapps/pc/css/
CA-Blue/images,/opt/CA/PerformanceCenter/PC/webapps/pc/css/CA-Gray/images,/usr/
bin/vim, /opt/CA/jre/bin/keytool, /opt/CA/PerformanceCenter/DM/repository, /opt/
CA/PerformanceCenter/NotificationScripts
caadmin ALL = CAPERFCENTER
```

4. Verify that the new host has network access from the original host.

Copy the Files from the Existing System

Follow these steps:

1. Log in to the original NetOps Portal host.
2. Stop the NetOps Portal services on the original system:
   ```
   service caperfcenter_console stop
   service caperfcenter_devicemanager stop
   service caperfcenter_eventmanager stop
   service caperfcenter_sso stop
   ```
3. Create a dump of the MySQL `netqosportal` database from the original system:
   ```
   /opt/CA/MySql/bin/mysqldump --routines netqosportal -unetqos -ppassword > $backupDir/netqosportal.sql
   ```
4. Create a dump of MySQL Event Manager data from the original system:
   ```
   /opt/CA/MySql/bin/mysqldump em -unetqos -ppassword > $backupDir/em.sql
   ```
5. Copy the database backups:
   ```
   scp $backupDir/netqosportal.sql <new_PC_host>:/tmp/netqosportal.sql
   scp $backupDir/em.sql <new_PC_host>:/tmp/em.sql
   ```
6. Stop the mysql service on the original system:
   ```
   service mysql stop
   ```
7. Package the files in the installation directory:
   ```
   cd /opt
   tar czf PC.tgz ./CA
   ```

   NOTE

   The following path is the default installation directory: `/opt/CA`
8. Copy the files to the new NetOps Portal host:
   ```
   scp /opt/PC.tgz <new_PC_host>:/opt
   ```
9. Copy the registry file that stores the installation and version information:
   ```
   scp /var/.com.zerog.registry.xml <new_PC_host>:/var
   ```
10. If you are migrating to a RHEL 6.x system, copy the NetOps Portal service startup scripts:
    ```
    scp /etc/init.d/caperfcenter_console <new_PC_host>:/etc/init.d/caperfcenter_console
    scp /etc/init.d/caperfcenter_devicemanager <new_PC_host>:/etc/init.d/
    caperfcenter_devicemanager
    scp /etc/init.d/caperfcenter_eventmanager <new_PC_host>:/etc/init.d/
    caperfcenter_eventmanager
    scp /etc/init.d/caperfcenter_sso <new_PC_host>:/etc/init.d/caperfcenter_sso
    ```
11. Copy the MySQL startup script:
    ```
    scp /etc/init.d/mysql <new_PC_host>:/etc/init.d/mysql
    ```
12. Copy the MySQL configuration file:
    ```
    scp /etc/my.cnf <new_PC_host>:/etc/
    ```

Configure the New System

NOTE
If SSL is configured on the source system, reconfigure it on the new system after setup.

Follow these steps:

1. Log in to the new NetOps Portal host.

2. Change the directory:
   ```
   cd /opt
   ```
3. Extract PC.tgz file:
   ```
   tar -xvf PC.tgz
   ```
4. Verify that the paths in
   ```
   /etc/profile
    point /opt/CA
   ```
   to the root where NetOps Portal is installed. Verify the following PATHs:
   ```
   PATH="${PATH}:/opt/CA/jre/bin"
   export PATH
   PATH="${PATH}:/opt/CA/MySql/bin"
   export PATH
   PATH="${PATH}:/opt/CA/PerformanceCenter/Tools/bin"
   export PATH
   ```
5. Do one of the following steps:
   - If you are migrating to a RHEL 6.x system, add the following services to add runlevel information for the NetOps Portal system services:
     ```
     chkconfig --add mysql
     chkconfig --add caperfcenter_console
     chkconfig --add caperfcenter_devicemanager
     chkconfig --add caperfcenter_eventmanager
     chkconfig --add caperfcenter_sso
     ```
   - If you are migrating to a RHEL 7.x, SLES, or OL system, add the following services to add runlevel information for the NetOps Portal system services:
     ```
     chkconfig --add mysql
     /opt/CA/PerformanceCenter/PC/bin/caperfcenter_console install
     /opt/CA/PerformanceCenter/DM/bin/caperfcenter_devicemanager install
     /opt/CA/PerformanceCenter/EM/bin/caperfcenter_eventmanager install
     /opt/CA/PerformanceCenter/sso/bin/caperfcenter_sso install
     ```
6. Configure MySQL in the `/etc/group` and `/etc/passwd` files:
   a. Add the following command to /etc/group:
      ```
      mysql:x:27:
      ```
   b. Add the following command to `/etc/passwd` file:
      ```
      mysql:x:27:27:MySQL Server:/var/lib/mysql:/bin/bash
      ```
   c. To ensure that the edits take effect, log out from the new NetOps Portal host, and log in again.
   d. Start the mysql service on the new system:
      ```
      service mysql start
      ```
7. Restore the databases:
   ```
   cd /opt/CA/MySql/bin
   mysql netqosportal -unetqos -ppassword -e 'source /tmp/netqosportal.sql'
   mysql em -unetqos -ppassword -e 'source /tmp/em.sql'
   ```
8. Start the NetOps Portal services on the new system as follows:
   a. Start the SSO service:
      ```
      service caperfcenter_sso start
      ```
   b. Wait one minute, then start the Event Manager and Device Manager:

```
service caperfcenter_eventmanager start
service caperfcenter_devicemanager start
```

c. Wait one minute, then start the console service:

```
service caperfcenter_console start
```

Update the Event Manager Data Source

If the new NetOps Portal hostname is different from the original NetOps Portal host, update the Event Manager data source. You must run a full Event Manager data source sync after you update the host name in the NetOps Portal UI.

Follow these steps:

1. Log in to the NetOps Portal UI as the Administrator user. Use the new host name:
   `http://new_PC_host:8181`

   NOTE

   If you cannot log in, run the following commands to check for database errors:

   ```
   mysqlcheck mysql
   mysqlcheck em
   mysqlcheck netqosportal
   ```

   If you see missing functions or corrupt table issues, see the following procedure for repairing the database.
2. Hover over **Administration**, and click **Data Sources: Data Sources**.
3. Select the Event Manager data source, and click **Edit**.
4. Edit the Host Name and Display Name fields:

   NOTE

   If SSL is configured, specify HTTPS for the communication protocol.
5. To validate communication, click **Test**.
6. Click **Save**.
7. Select the Event Manager data source, and click **Resync**.
8. Select **Perform a Full Resynchronization**, and click **Resync**.
   NetOps Portal resynchronizes with the Event Manager data source.
9. After the Event Manager has completed a full resynchronization, re-register the Event Manager with the data sources:
   a. Run the following command to log in to MySQL:

      ```
      /opt/CA/MySql/bin/mysql em -unetqos -ppassword
      ```

   b. Run the following command:

      ```
      UPDATE em.data_sources SET LastEvent = 0, ConsumerID = 0;
      ```

   The new Event Manager IP address is pushed to the Data Aggregator.

Repair the Database

If you cannot log in to to the NetOps Portal UI, check for database errors. If there are errors, stop the NetOps Portal services. Repair the database with the appropriate command. Then restart the NetOps Portal services.

Follow this procedure:

1. Stop the NetOps Portal services:

   ```
   service caperfcenter_console stop
   service caperfcenter_devicemanager stop
   service caperfcenter_eventmanager stop
   ```

```
service caperfcenter_sso stop
```

2. Repair the database with the appropriate command:

```
mysqlcheck -r mysql
mysqlcheck -r em
mysqlcheck -r netqosportal
```

3. Start the NetOps Portal services on the new system as follows:
   a. Start the SSO service:

```
service caperfcenter_sso start
```

   b. Wait one minute, then start the event manager and Device Manager:

```
service caperfcenter_eventmanager start
service caperfcenter_devicemanager start
```

   c. Wait one minute, then start the console service:

```
service caperfcenter_console start
```

Configure the Data Source Connector for a Report

If you have the *Unified Dashboards and Reporting for Infrastructure Management* solution set up, and the new NetOps Portal hostname has changed, reconfigure the data source connector for the reports.

Follow these steps:

1. Ensure that the CA Business Intelligence data source is installed.
2. Open the CA Business Intelligence JasperReports login page in a browser. Log in to the CA Business Intelligence JasperReports Server as a user with an Administration role for the **Public** organization.
3. Select **View**, **Repository**.
4. In the Folders panel on left, expand **Public**, **ca**, **Performance Management**, **datasources**.
5. Select the NetOps Portal data source and click **Edit**.
6. Update the data source parameters and click **Save**:
   - **Schema**
     Specify HTTP or HTTPS.
   - **Host**
     Specify the NetOps Portal host name. Do not specify the Data Aggregator host name.
   - **Port**
     Specify the NetOps Portal port (default: 8181).
   - **NetOps Portal GUID**
     The NetOps Portal GUID (unique) is used when the password for the CA Business Intelligence web user is not available. This information passes to the Data Aggregator to ensure that the request is going to the correct NetOps Portal instance. A web service call on NetOps Portal retrieves this information.
     GUID Endpoint URL: `http://hostname:8181/pc/center/webservice/datasources/performanceCenterGUID`
7. Click **Save** to apply the changes.

Verify the Migration

You must run Resync All to update the IP addresses for the NetOps Portal `netqosportal` tables.

Follow these steps:

1. Log in to the NetOps Portal UI as the Administrator user. Use the new host name:
   `http://new_PC_host:8181`
2. Hover over **Administration**, and click **Data Sources: Data Sources**.

3. Click **Resync All**.
   NetOps Portal synchronizes with all registered data sources and updates the required contact information.
4. To verify that the NetOps Portal host name is correct on the Data Aggregator, use the following REST call:
   `http://<DA_host>:8581/rest/dataaggregator`
   The value for `NpcHostName` is the IP address of the new NetOps Portal host.

Configure Authentication

NetOps Portal supports external authentication schemes, such as LDAP, SAML 2.0, SSL.

- For LDAP, no additional configuration is required with a NetOps Portal migration.
- For SAML 2.0, reconfigure SAML 2.0 on the new system. For more information, see Set Up SAML 2.0 Support.
- For SSL, set up new SSL certificates with the new hostname and alias. For more information, see Set Up SSL Certificates for Performance Center.

NOTE

If you have overridden the NPC hostname that Data Aggregator contacts, update Web Service Host in SSO Config. For more information, see Update Performance Center Website Settings.

Modify Maximum Memory Usage for Each Performance Center Service

Modify the maximum memory usage for the NetOps Portal daemons to let the daemons run effectively. Configure memory allocation during or after the installation.

Follow these steps:

1. On the server where you installed NetOps Portal, run the following command:
   ```
   more /proc/meminfo
   ```
   The total memory usage of the server is displayed.
2. Make a note of the total memory.
3. Modify the maximum memory for a daemon as follows:
   a. Edit the following file:
      `/Installation Directory/PerformanceCenter/Service Subdirectory/conf/wrapper.conf`

      NOTE

      The Service subdirectory is one of the following options:

      - PC (Console daemon)
      - DM (Device Manager daemon)
      - EM (Event Manager daemon)

   b. Search for the parameter 'wrapper.java.maxmemory'.
   c. Change the current value. For example, in a small deployment, set it to '3072' (the units are MB).
   d. Save the file.
4. Stop and restart each daemon by entering the following commands:
   ```
   service service name stop
   service service name start
   ```

   NOTE

   The service name is one of the following options:

- caperfcenter_console
- caperfcenter_devicemanager
- caperfcenter_eventmanager

The maximum amount of memory is configured for a deployment of your scalability requirements.

Restore Performance Center

Restore an existing backup of NetOps Portal.

Restore the Database After a Reinstallation

Restore the NetOps Portal database from a backup archive after you reinstall the software. Restoring the database from a backup preserves data continuity and enables most historical reporting after a failure occurs.

Database restoration is required only if a failure occurs. Take the cleanup steps that are described in Clean Up After a Failed Installation before you attempt the installation again. Then, take the steps in this procedure.

NOTE

For upgrade failure situations, follow the steps in Recover from an Upgrade Failure.

Follow these steps:

1. Log in to the server as 'root', or use the 'sudo' account that you configured for the installation.
2. Stop all the NetOps Portal services using the following commands:

```
service caperfcenter_eventmanager stop

service caperfcenter_devicemanager stop

service caperfcenter_sso stop

service caperfcenter_console stop
```

3. Change to the directory where you saved the backup archive:

```
cd backup_Directory
```

4. Uncompress the database backup archives for NetOps Portal and Event Manager by executing the following commands:

```
tar zxvf netqosportal.tgz

tar zxvf em.tgz
```

5. Import the uncompressed NetOps Portal backup file:

```
mysql netqosportal -u root -ppassword -e 'source $backupDir/netqosportal.sql'
```

6. Import the uncompressed Event Manager backup file:

```
mysql em -u root -ppassword -e 'source $backupDir/em.sql'
```

7. Start the NetOps Portal services on the new system as follows:
   a. Start the SSO service:

      ```
      service caperfcenter_sso start
      ```

   b. Wait one minute, then start the event manager and device manager:

      ```
      service caperfcenter_eventmanager start
      ```

      ```
      service caperfcenter_devicemanager start
      ```

   c. Wait one minute, then start the console service:

      ```
      service caperfcenter_console start
      ```

8. Delete the uncompressed archive files to save space:

   ```
   rm netqosportal.sql
   ```

   ```
   rm em.sql
   ```

9. Log in to NetOps Portal as an administrator.
10. Verify that your configuration data appears in the Admin pages.

Recover from an Upgrade Failure

Follow these steps to restore the NetOps Portal database and verify the database schema version. If the schema version matches the current product version, try the upgrade again. If not, see *Recover from an Upgrade Failure with an Error* for the steps to update the schema version.

> **NOTE**
> Database restoration is required after an upgrade only if a failure occurs. Take the cleanup steps that are described in *Clean Up After a Failed Installation* before you attempt the upgrade again. Then, take the steps in this procedure.

Follow these steps:

1. Log in to the server as 'root', or use the 'sudo' account that you configured for the installation.
2. Stop all the NetOps Portal services, using the following commands:

   ```
   service caperfcenter_eventmanager stop
   ```

   ```
   service caperfcenter_devicemanager stop
   ```

   ```
   service caperfcenter_sso stop
   ```

   ```
   service caperfcenter_console stop
   ```

3. Change to the directory where you saved the backup archive:

   ```
   cd backup_Directory
   ```

4. Uncompress the database backup archives for NetOps Portal and Event Manager by executing the following commands:

   ```
   tar zxvf netqosportal.tgz

   tar zxvf em.tgz
   ```

5. Import the uncompressed NetOps Portal backup file:

   NOTE

   If you leave out the optional password syntax from the following command, you are prompted for the password.

   ```
   mysql netqosportal -u root -ppassword -e 'source $backupDir/netqosportal.sql'
   ```

6. Import the uncompressed Event Manager backup file:

   ```
   mysql em -u root -ppassword -e 'source $backupDir/em.sql'
   ```

7. Change to the following installation directory:

   ```
   cd /opt/CA/PerformanceCenter/Tools/bin
   ```

8. Run the following command to verify the database version:

   ```
   mysql -P3306 -D netqosportal -u root -ppassword


   mysql> select InstallDate, version, dbschemaversion from revision_info order by
    InstallDate asc;
   ```

   The output lists installation dates and versions of the software and database schema.
   If the database version does not match the current product version, follow the steps in *Recover from an Upgrade Failure with an Error*.

Recover from an Upgrade Failure with an Error

If an error occurs during an upgrade of the NetOps Portal software, restore the database and update the schema. See *Recover from an Upgrade Failure* for the steps to restore the database and verify the database schema version. If the schema version does not match the current product version, update the schema version.

Follow these steps:

1. Upgrade the database schema. From the Tools/bin directory, run the npcshell database utility to upgrade the schema to the current version:

   ```
   ./npcshell.sh upgradedb
   ```

2. Run the following commands to import database translation files:

   ```
   /opt/CA/jre/bin/java -jar /opt/CA/PerformanceCenter/SQL/seedlu/bin/seedlu.jar -
   resfile "/opt/CA/PerformanceCenter/SQL/messages_en_US.properties" -ctrlfile "/opt/
   CA/PerformanceCenter/SQL/control.sdlctrl" -connection "jdbc:mysql://localhost:3306/
   netqosportal?useUnicode=true&characterEncoding=UTF-8" -user netqos -pwd password -
   lang en-US
   /opt/CA/jre/bin/java -jar /opt/CA/PerformanceCenter/SQL/seedlu/bin/seedlu.jar -
   resfile "/opt/CA/PerformanceCenter/SQL/messages_zh_CN.properties" -ctrlfile "/opt/
   CA/PerformanceCenter/SQL/control.sdlctrl" -connection "jdbc:mysql://localhost:3306/
   netqosportal?useUnicode=true&characterEncoding=UTF-8" -user netqos -pwd password -
   lang zh-CN
   /opt/CA/jre/bin/java -jar /opt/CA/PerformanceCenter/SQL/seedlu/bin/seedlu.jar -
   resfile "/opt/CA/PerformanceCenter/SQL/messages_zh_TW.properties" -ctrlfile "/opt/
   CA/PerformanceCenter/SQL/control.sdlctrl" -connection "jdbc:mysql://localhost:3306/
   netqosportal?useUnicode=true&characterEncoding=UTF-8" -user netqos -pwd password -
   lang zh-TW
   /opt/CA/jre/bin/java -jar /opt/CA/PerformanceCenter/SQL/seedlu/bin/seedlu.jar -
   resfile "/opt/CA/PerformanceCenter/SQL/messages_fr_FR.properties" -ctrlfile "/opt/
   CA/PerformanceCenter/SQL/control.sdlctrl" -connection "jdbc:mysql://localhost:3306/
   netqosportal?useUnicode=true&characterEncoding=UTF-8" -user netqos -pwd password -
   lang fr-FR
   /opt/CA/jre/bin/java -jar /opt/CA/PerformanceCenter/SQL/seedlu/bin/seedlu.jar -
   resfile "/opt/CA/PerformanceCenter/SQL/messages_ja_JP.properties" -ctrlfile "/opt/
   CA/PerformanceCenter/SQL/control.sdlctrl" -connection "jdbc:mysql://localhost:3306/
   netqosportal?useUnicode=true&characterEncoding=UTF-8" -user netqos -pwd password -
   lang ja-JP
   ```

   NOTE

   Replace the '*password*' variable with the password.

3. Update the information that NetOps Portal uses to display administration pages and views. Run the following commands:
   - **CA Infrastructure Management Administration Pages:**

     ```
     ./npcshell.sh dbmigrate -package com.ca.im.plugin.pc -path ../../SQL/plugins/pc/
     ```

   - **Event-Related Views:**

     ```
     ./npcshell.sh dbmigrate -package com.ca.im.plugin.em -path ../../SQL/plugins/
     eventmanager/
     ```

- **Data Aggregator Administration Pages and Views:**

```
./npcshell.sh dbmigrate -package com.ca.im.plugin.da -path ../../SQL/plugins/
polaris/
```

4. Run the following command again to verify the database version after you have upgraded it:

```
mysql -P3306 -D netqosportal -u root -ppassword


mysql> select InstallDate, version, dbschemaversion from revision_info order by
 InstallDate asc;
```

5. Start all the NetOps Portal services:

```
service caperfcenter_eventmanager start

service caperfcenter_devicemanager start

service caperfcenter_sso start

service caperfcenter_console start
```

6. Delete the uncompressed archive files to save space:
```
rm netqosportal.sql
rm em.sql
```
7. Log in to NetOps Portal as an administrator.
8. Verify that your configuration data appears in Admin pages.

Restore Single Sign-On Settings

If your NetOps Portal deployment uses single sign-on, restore the configuration settings.

Restore the following files:

- `/opt/CA/CA/PerformanceCenter/sso/start.ini`
- `/opt/CA/CA/PerformanceCenter/PC/start.ini`

Restore the follow directories:

- `/opt/CA/CA/PerformanceCenter/sso/webapps/sso/configuration`
- `/opt/CA/CA/PerformanceCenter/sso/etc`
- `/opt/CA/CA/PerformanceCenter/sso/conf`
- `/opt/CA/CA/PerformanceCenter/PC/etc`
- `/opt/CA/CA/PerformanceCenter/PC/conf`

If you have enabled SSL, restore the following files:

- `/opt/CA/CA/PerformanceCenter/sso/start.d/ssl.ini`
- `/opt/CA/CA/PerformanceCenter/PC/start.d/ssl.ini`

Restore Custom Settings

If your deployment includes custom OpenAPI applications, restore the application folder:

- `/opt/CA/PerformanceCenter/PC/webapps/pc/apps`

If you use custom logos for your themes, restore the following files:

- `/opt/CA/PerformanceCenter/PC/webapps/pc/css/CA-Blue/images/customLogo`
- `/opt/CA/PerformanceCenter/PC/webapps/pc/css/CA-Gray/images/customLogo`

Restore Reports

Full PDF reports are temporarily saved for download.

Restore the following directory:

- `/opt/CA/PerformanceCenter/DM/repository`

Restore Script Notification Actions

If you have script notification actions, restore the following directory:

- `/opt/CA/PerformanceCenter/NotificationScripts`

Update the Data Aggregator and Event Manager Data Sources

If the new NetOps Portal hostname is different from the original NetOps Portal host, update the the Data Aggregator and Event Manager data sources.

NetOps Portal Disaster Recovery Script

Location: `/opt/CA/PerformanceCenter/Tools/bin/update_pc_da_database_references.sh`

On the NetOps Portal host in the recovery system, update the bold sections of the script to match your system:

```
...

#####################################################################

# UPDATE THE FOLLOWING PC/DA VARIABLES TO REFLECT NEW ENVIRONMENT

#####################################################################

NEW_PC_IP_ADDRESS="<Recovery/New PC IP Address>"

NEW_PC_HOSTNAME="<Recovery/New PC Hostname>"

NEW_PC_EVENT_PRODUCER_PORT=8181


NEW_PC_EVENT_PRODUCER_PROTOCOL="http" # change to "https" if using SSL
```

```
NEW_DA_IP_ADDRESS="<Recovery/New DA IP Address>"

NEW_DA_HOSTNAME="<Recovery/New DA Hostname>"

NEW_DA_PORT_NUMBER=8581


...
```

Start NetOps Portal

Follow these steps:

1. Restore the NetOps Portal backups.
2. Run the NetOps Portal disaster recovery script:

   ```
   /opt/CA/PerformanceCenter/Tools/bin/your_update_pc_da_database_references.sh
   ```

3. Start the SSO service:

   ```
   service caperfcenter_sso start
   ```

4. Wait one minute, then start the event manager and device manager:

   ```
   service caperfcenter_eventmanager start

   service caperfcenter_devicemanager start
   ```

5. Wait one minute, then start the console service:

   ```
   service caperfcenter_console start
   ```

Resynchronize the Event Manager Database

The recommended backup and restore procedures include an instruction to back up the Event Manager database.

If you neglect this step, problems can occur when the Event Manager tries to synchronize with NetOps Portal. The synchronization can fail because NetOps Portal has outdated Event Manager information. The newly installed Event Manager has the new information.

If this problem occurs, resynchronize these two databases.

> **WARNING**
> Since this procedure does not include a step to restore the Event Manager database, notifications are not preserved, and must be recreated. Otherwise, the Event Manager runs normally.

Follow these steps:

1. Log in as a user with the Administrator role.

2. Navigate to the Manage Data Sources page.
   The current list of registered data sources appears on the Manage Data Sources.
3. Select the data source that you want to remove (unregister).
4. Click Remove, and then click Yes to confirm the deletion.
   The data source is removed from the list.
5. Remove all properties that are related to NetOps Portal from the em.general database table using the following command:

```
DELETE from em.general where Attribute LIKE 'NPC.%';
```

6. Restart Event Manager using the following command:

```
service caperfcenter_eventmanager restart
```

7. Return to the Manage Data Sources page.
8. Register the Event Manager data source.

Set the Email Server

To let users send reports by email, configure an email server. NetOps Portal sends reports on a schedule or as needed. Select a server that the NetOps Portal server has network access to.

Follow these steps:

1. Log in as a user with administrative role rights.
2. Select **Administration**, **Configuration Settings**, and click **Email Server**.
   The Email Server Settings page opens.
3. Select the **Enable Email** check box.
   The page refreshes to highlight the required field.
4. Complete the following fields as necessary:
   - **SMTP Server Address**
     Is the IP address or hostname of the server to use to send reports by email.
   - **SMTP Server Port**
     Is the port on the email server that is used to send messages.
     Default: Port 25.
   - **Email Reply Address**
     Is the email address from which NetOps Portal sends reports. An administrator monitors this address for responses to email messages sent by the product.
5. (3.7.6 and lower only) Enable SSL encryption. This parameter is required if you want to use a secure connection to send email from NetOps Portal.

   NOTE

   To enable this feature, you must enable NetOps Portal to use SSL. For more information, see Enable Performance Center to use SSL.

   This parameter is unsupported for 3.7.7 and higher. For 3.7.7 and higher, if email server supports STARTTLS, Performance Center automatically tries to send all emails securely.

6. (Optional) Take the following steps to enable SMTP authentication:
   a. Select **Enable Authentication**.
   b. Type the username for SMTP authentication in the Username field.
   c. Type the authentication password in the Password field.
   d. Type the authentication password again in the Confirm Password field.

      NOTE
      SMTP authentication is disabled by default.
7. Click **Save**.
   The email server is set.

Import the Email Server Certificate

Import the email server certificate so that the email server is a trusted connection.

Follow these steps:

1. Import the email server certificate into the java trusted certificate keystore using the following steps:
   ```
   keytool -importcert -keystore InstallDirectory/jre/lib/security/cacerts
   -storepass cacertspassword -alias alias_name -file filename.cer
   ```
   - ***cacertspassword***
     Specify the password for the Certificate Authority keystore.

     NOTE
     The default password for the Certificate Authority keystore is **changeit**.
   - ***alias_name***
     Specify an alias that can be used to refer to the keystore entry that is created for the email server certificate.
   - ***filename*.cer**
     Specify the file to which the certificate is exported. We recommend using a full pathname that does not place the file in the current directory.
     Example: /tmp/capcCert.cer
2. Import the root and intermediate certificates into the java trusted certificate keystore for each certificate:
   ```
   keytool -importcert -keystore InstallDirectory/jre/lib/security/cacerts
   -storepass cacertspassword -alias alias_name -file filename.cer
   ```
   - ***cacertspassword***
     Specify the password for the Certificate Authority keystore.

     NOTE
     The default password for the Certificate Authority keystore is **changeit**.
   - ***alias_name***
     Specify an alias that can be used to refer to the keystore entry that is created for the root or intermediate certificate.
   - ***filename*.cer**
     Specify the file to which the certificate is exported. We recommend using a full pathname that does not place the file in the current directory.
     Example: /tmp/capcCert.cer
3. Confirm that you trust the certificate.
4. Verify that your imported keystore is available:
   ```
   keytool -list -keystore InstallDirectory/jre/lib/security/cacerts
   ```

Update Performance Center Website Settings

The Single Sign-On Configuration Tool lets you change the default settings for the NetOps Portal website and web service. For example, you can specify a different host or port number for the NetOps Portal web service. These settings instruct the Single Sign-On application how to connect to NetOps Portal.

Follow these steps:

1. Log in to the NetOps Portal host as root or with the 'sudo' command.
2. Launch the Single Sign-On Configuration Tool by running the './SsoConfig' command in the following directory:
   InstallDirectory/PerformanceCenter

   NOTE
   /opt/CA is the default installation directory.

   You are prompted to select an option. The available options correspond to CA applications running on the local server.
3. Use the following commands as needed while you are selecting settings:
   - q (quit)
   - b (go back to the previous menu)
   - u (update)
   - r (reset)
4. Enter 1 to configure NetOps Portal.
   You are prompted to select a configuration option.
5. Enter 3 for NetOps Portal.
   You are prompted to specify the priority.
   The Priority parameter only applies to NetOps Portal.
6. Enter one of the following options:
   - **1. Remote Value**These settings are propagated to all other CA products and data sources that are registered to this instance of NetOps Portal. This includes the Event Manager in NetOps Portal, which embeds the URL of NetOps Portal. NetOps Portal uses Remote Value settings only if a corresponding Local Override value is not present.
   - **2. Local Override**
     Overrides a setting on this NetOps Portal instance, which does not propagate to other CA products and data sources (including Event Manager) registered to this instance of NetOps Portal. Local Override takes precedence over both the Remote Value and default settings.

   You are prompted to select a property to configure.

   NOTE
   Configure the scheme or port using Remote Value to include the correct CAPC URL in threshold event email messages.
7. Enter one or more of the following properties. When prompted, enter u to update the value and supply a new value:
   - **1. Web Service Scheme**
     Specifies the URL scheme for Device Manager service.
   - **2. Web Service Host**
     Specifies the URL host for the Device Manager service.

     NOTE
     OpenAPI SSO Authentication must be able to access NetOps Portal console and SSO ports. If these ports are blocked for the listed hostname or IP address, use a specified hostname or IP address with access to console and SSO ports.
   - **3. Web Service Port**
     Specifies the URL port for the Device Manager service.
   - **4. Web Service Inventory**

Specifies the URL path for the Device Manager inventory web service (version 1).

- **5. Web Service Data Source Admin**
  Specifies the URL path for the Device Manager data source web service.
- **6. Web Site Scheme**
  Specifies the URL scheme for all access to the NetOps Portal website. If you have set up HTTPS, use https.
- **7. Web Site Host**
  Specifies the URL host for all access to the NetOps Portal website.

  NOTE
  To use a specified hostname in emails instead of the IP address for the NetOps Portal host, set the Remote Value to that hostname.
- **8. Web Site Port**
  Specifies the URL port for all access to the NetOps Portal website.
- **9. Web Site Path**
  Specifies the URL path for all access to the NetOps Portal website.
- **10. SMTP Enabled**
  Specifies whether the Simple Mail Transfer Protocol (SMTP) is enabled to allow NetOps Portal operators to email reports and event notifications.
  Default: Disabled

  NOTE

  We recommend you update SMTP settings on the Email Server Settings page. For more information, see Set the Email Server.
- **11. SMTP Server Address**
  Is the IP address of the SMTP server.
  Default: Disabled

  NOTE

  We recommend you update SMTP settings on the Email Server Settings page. For more information, see Set the Email Server.
- **12. SMTP Ports:**
  Specifies the port to use for SMTP requests.
  Default: Port 25.

  NOTE

  We recommend you update SMTP settings on the Email Server Settings page. For more information, see Set the Email Server.
- **13. SMTP SSL**
  Specifies whether to use SSL encryption when sending email from NetOps Portal or other CA data source products. Verify that SSL has been properly set up on your system before you enable this option.
  Default: Disabled

  NOTE

  We recommend you update SMTP settings on the Email Server Settings page. For more information, see Set the Email Server.
- **14. Email Reply Address**
  Specifies the return address to use for email messages that are generated by NetOps Portal. Enter u to update the value, and supply an email address. Use the format username@mydomain.com.

NOTE

We recommend you update SMTP settings on the Email Server Settings page. For more information, see Set the Email Server.

- **15. Email Format**
  Specifies the format to use for email messages that are sent by NetOps Portal. Enter u to update the value, and supply either HTML or text.

  NOTE

  We recommend you update SMTP settings on the Email Server Settings page. For more information, see Set the Email Server.
- **16. SMTP Username**
  Specifies a username to use when the email server challenges an SMTP request. Supply a username, or supply an empty string to disable client-side authentication.

  NOTE

  We recommend you update SMTP settings on the Email Server Settings page. For more information, see Set the Email Server.
- **17. SMTP Password**
  Specifies a username to use when the email server challenges an SMTP request. Supply any valid password. The SMTP Username parameter is required.

  NOTE

  We recommend you update SMTP settings on the Email Server Settings page. For more information, see Set the Email Server.
- **18. Web Service Inventory (Version 2) (Local Override)**
  Specifies the URL path for the Device Manager inventory web service (version 2).
- **19. SMTP Authentication (Local Override)**
  Specifies whether to use authentication when sending email from CA products.

  NOTE

  We recommend you update SMTP settings on the Email Server Settings page. For more information, see Set the Email Server.
- **20. Allow NetOps Portal in a frame (Local Override)**
  Determines whether NetOps Portal is allowed to display within a frame in a web page.
  Default: Enabled

8. Enter b when you have finished changing the default settings.
   You return to the previous set of options.
9. Enter b again to go back to the first set of options.
10. Enter q to quit.
    The Single Sign-On Configuration Tool closes. NetOps Portal directs all users to the Single Sign-On website using the new values that you supplied.

Data Aggregator Administration

The following articles relate to administration tasks for data aggregators and data collectors:

Authenticate and Encrypt ActiveMQ Communication

By default, the communication between the Data Aggregator and Data Collector is unencrypted and unauthenticated. To secure communications, secure the communication between the ActiveMQ brokers on these servers.

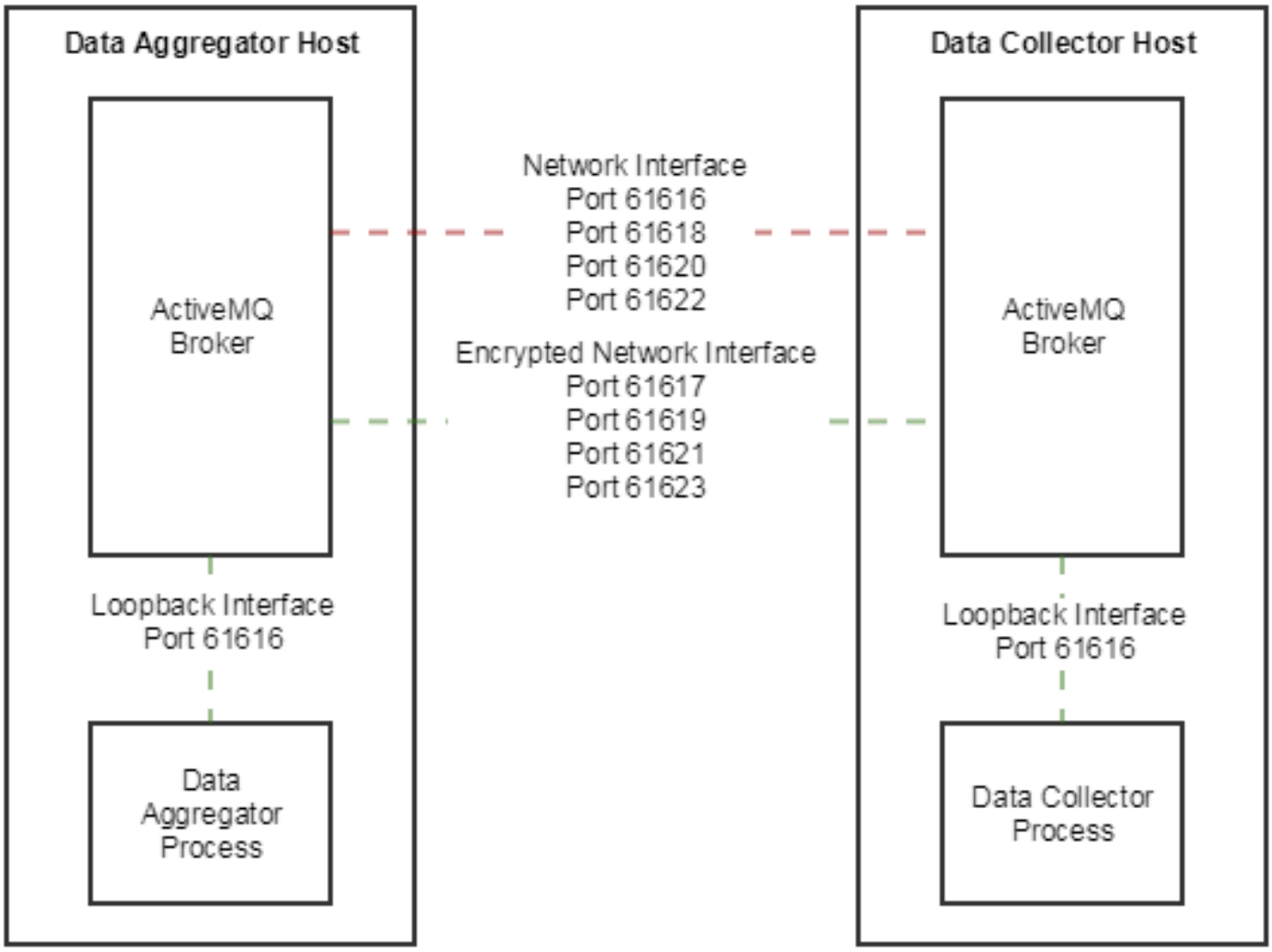
The following ports enable ActiveMQ communication between the Data Aggregator and the Data Collectors:

- **TCP 61616**
  Enables only ActiveMQ traffic
- **TCP 61618**
  Enables poll response delivery traffic
- **TCP 61620**
  Enables distributed IREP traffic
- **TCP 61622**
  Enables large data transfers
  This port also enables the simplified upgrade for Data Collectors. For more information, see Upgrade the Data Collectors.

The Data Aggregator does not communicate directly with the Data Collector. Each host server has an ActiveMQ broker. The Data Aggregator and Data Collector each communicate to the local broker over the loopback interface on port 61616. In the default configuration, the ActiveMQ brokers communicate on the same port over a network interface.

The following diagram shows the communication between the services, brokers, and hosts:

Figure 60: Diagram that shows the communication architecture between the Data Aggregator and Data Collectors.



To secure the communication between the brokers, use TLS and communicate on a different port. This procedure uses 61617, 61619, 61621, and 61623. Leave ports 61616, 61618, 616120, and 61622 unencrypted and restrict communication on these ports to the loopback interface.

NOTE

Throughout the documentation 8182, 8382, 61617, 61619, 61621, and 61623 appear as suggested port numbers for secured communications. In the instances where these ports appear, you are free to use any value you want as long as no other processes are using it.

Communication between the ActiveMQ broker and the Data Aggregator or Data Collector java process is not encrypted or authenticated. Because this traffic occurs only on the loopback interface, this communication is not vulnerable to sniffing.

WARNING

If DX NetOps Performance Management was installed to be run as a sudo user, run these commands as that sudo user.

To secure communications between the Data Aggregator and Data Collectors, complete the following procedures:

TIP

Save a backup copy of the activemq.xml file from the Data Aggregator and each Data Collector. To revert the authentication configuration, replace the updated XML files with the backups.

NOTE

A restart of the Data Aggregator and Data collectors, is not required during this process. The configuration changes take effect after restarting the ActiveMQ brokers. Restart the brokers after the configuration changes are complete.

WARNING

When you configure the Data Aggregator broker for TLS, configure all the Data Collector brokers for TLS too. Set up the configuration on each host, shut down all brokers. Then, restart the brokers, starting with the Data Aggregator.

Open Ports on Firewalls

On all relevant firewalls between the Data Aggregator host and Data Collector hosts, open port 61617 for TLS communications.

Generate Keys and Establish Trust

To establish a trusted connection, generate private/public key pairs for the Data Aggregator and each Data Collector and set up trust stores. Each Data Collector must trust itself and the Data Aggregator. The Data Aggregator must trust itself and all the Data Collectors.

Each system needs two private keys: one for the ActiveMQ broker, and one for the client, which is the Data Aggregator or Data Collector process. On each system, you replace two **.ks** files and one **.ts** files. Each file has a nonsecure password that is stored in clear text in activemq.xml. Because of the passwords, and the general sensitivity of encryption keys, the files activemq.xml ***.ts *.ks** require 400 permission. The user that runs the ActiveMQ broker must own these files.

The local local security policy dictates how to generate the key pairs. After you generate the keys, copy the public keys to the other hosts. All the Data Collectors need the Data Aggregator key, and the Data Aggregator needs all the Data Collector keys.

Example

This example procedure uses the JDK keytool. The following command generates a self-signed key using the JDK keytool:

```
keytool -genkey -alias KEY_ALIAS -keyalg RSA -keystore broker.ks
```

- **KEY_ALIAS** is a string that identifies the key.

The keytool is interactive and requires a series of inputs. The following example shows the interaction for the keytool:

NOTE
For the first and last name prompt, you must enter the host name of the system where you are creating the certification.

```
[root@hostname-dc conf]# keytool -genkey -alias dc1 -keyalg RSA -keystore broker.ks
Enter keystore password: 123456
Re-enter new password: 123456
What is your first and last name?
  [Unknown]:  Host_Name
What is the name of your organizational unit?
  [Unknown]:  Team1
What is the name of your organization?
```

```
  [Unknown]:  CGPM
What is the name of your City or Locality?
  [Unknown]:  Framingham
What is the name of your State or Province?
  [Unknown]:  MA
What is the two-letter country code for this unit?
  [Unknown]:  US
Is CN=Host_Name, OU=Team1, O=CGPM, L=Framingham, ST=MA, C=US correct?
  [no]:  yes
Enter key password for <dc1>
        (RETURN if same as keystore password):
```

Generate Keys and Establish Trust on the Data Collectors

Complete this procedure on each Data Collector host. Each keytool command is interactive and requests a password for each **.ks** and **.ts** file.

DC_ALIAS is a user defined unique identifier for each Data Collector host and can be any string as long as it is unique for each Data Collector.

Follow these steps:

1. Change directories:
   `cd /opt/IMDataCollector/broker/apache-activemq-version/conf`
2. Remove existing security files:
   `rm -f *.ks *.ts *.cert`
3. Generate the broker keystore and private key:
   `keytool -genkey -alias DC_ALIAS -keyalg RSA -keystore broker.ks`
4. Export the broker key for the Data Collector:
   `keytool -export -alias DC_ALIAS -keystore broker.ks -file DC_ALIAS.cert`
5. Import the client key for the Data Collector:
   `keytool -import -alias DC_ALIAS -keystore client.ts -file DC_ALIAS.cert`
6. Copy the Data Collector key to the Data Aggregator for import when you establish trust on the Data Aggregator:
   `scp DC_ALIAS.cert root@$DA_HOST:/tmp/DC_ALIAS.cert`

Generate Keys and Establish Trust on the Data Aggregator

Follow these steps:

1. Change directories:
   `cd /opt/IMDataAggregator/broker/apache-activemq-version/conf`
2. Remove existing security files:
   `rm -f *.ks *.ts *.cert`
3. Generate the broker keystore and private key:
   `keytool -genkey -alias DA_ALIAS -keyalg RSA -keystore broker.ks`
4. Export the broker key for the Data Aggregator:
   `keytool -export -alias DA_ALIAS -keystore broker.ks -file DA_ALIAS.cert`
5. Import the client key for the Data Aggregator:
   `keytool -import -alias DA_ALIAS -keystore client.ts -file DA_ALIAS.cert`
6. Import the client keys for *each* Data Collector:
   `keytool -import -alias DC_ALIAS -keystore client.ts -file /tmp/DC_ALIAS.cert`

   Repeat the keytool import command for each Data Collector with each DC_ALIAS.

7. Remove CERT files from the /tmp directory:
   ```
   rm /tmp/*.cert
   ```
8. Copy the broker key for the Data Aggregator to the /tmp directory:
   ```
   cp DA_ALIAS.cert /tmp
   ```
9. Grant the appropriate permissions to the security files:
   ```
   chmod 400 *.ks *.ts *.cert
   ```

Establish Trust from the Data Collectors to the Data Aggregator

Complete this procedure on *each* Data Collector host.

Follow these steps:

1. Change directories:
   ```
   cd /opt/IMDataCollector/broker/apache-activemq-version/conf
   ```
2. Copy the Data Aggregator key to the Data Collector host:
   ```
   scp root@$DA_HOST:/tmp/DA_ALIAS.cert .
   ```
3. Import the Data Aggregator key to the Data Collector keystore:
   ```
   keytool -import -alias DA_ALIAS -keystore client.ts -file DA_ALIAS.cert
   ```
4. Grant the appropriate permissions to the security files:
   ```
   chmod 400 *.ks *.ts *.cert
   ```

Configure ActiveMQ on the Data Aggregator

On the Data Aggregator, enable TLS with client authentication, and restrict OpenWire to localhost only.

Follow these steps:

1. On the Data Aggregator host, edit the following file:
   /opt/IMDataAggregator/broker/apache-activemq-version/conf/activemq.xml
2. Add the following XML section before <transportConnectors> parameter:
   ```
   <sslContext>
      <sslContext
         keyStore="broker.ks" keyStorePassword="123456"
         trustStore="client.ts" trustStorePassword="123456"/>
   </sslContext>
   ```
3. Restrict the existing OpenWire transport connector to the local host only:
   ```
   <transportConnector name="openwire" uri="tcp://127.0.0.1:61616"/>
   ```
4. Change the permissions for the file:
   ```
   chmod 400 activemq.xml
   ```

Configure ActiveMQ on the Data Collectors

On the Data Collectors, enable TLS, update the URL for the Data Aggregator, and restrict OpenWire to localhost only.

Complete the procedure on *each* Data Collector host.

Follow these steps:

1. On the Data Aggregator host, edit the following file:
   `/opt/IMDataCollector/broker/apache-activemq-version/conf/activemq.xml`
2. Add the following XML section before <transportConnectors> parameter:

```
<sslContext>
   <sslContext
      keyStore="broker.ks" keyStorePassword="123456"
      trustStore="client.ts" trustStorePassword="123456"/>
</sslContext>
```

3. Restrict the existing OpenWire transport connector to the local host only:
   ```
   <transportConnector name="openwire" uri="tcp://127.0.0.1:61616?
   maximumConnections=100&amp;wireFormat.maxFrameSize=104857600"/>
   ```
4. For all `<networkConnector>` entries, change `tcp://dahostname` to `ssl://dahostname` and update the ports.
   Example:
   The following example is a `<networkConnector>` entry that you might see:
   ```
   <networkConnector name="da_manager" uri="static:(tcp://dahostname:61616)"
    duplex="true" suppressDuplicateTopicSubscriptions="false">
   ```
   Replace `tcp` with `ssl` as shown in the following example:
   ```
   <networkConnector name="da_manager" uri="static:(ssl://dahostname:61617)"
    duplex="true" suppressDuplicateTopicSubscriptions="false">
   ```
5. Change the permissions for the file:
   ```
   chmod 400 activemq.xml
   ```

   NOTE

   The user running the ActiveMQ service must own this file.

Restart ActiveMQ Brokers

The ActiveMQ brokers reread the configuration when the broker restarts. Restart all the brokers simultaneously. Do not restart the Data Aggregator or Data Collector processes.

During the shutdown, the Data Collectors cache incoming traffic. To minimize data loss, perform the shutdowns and restarts in this order:

1. Shut down the ActiveMQ broker on each Data Collector:
   ```
   service activemq stop
   ```
2. Shut down the ActiveMQ broker on the Data Aggregator:
   ```
   service activemq stop
   ```
3. Start the ActiveMQ broker on the Data Aggregator:
   ```
   service activemq start
   ```
   If you do not, the Data Aggregator starts the broker automatically.
4. The Data Collectors automatically restart the ActiveMQ brokers. Use the following command to restart the brokers manually:
   ```
   service activemq start
   ```

Block Port 61616

On all relevant firewalls, restrict the communication on port 61616 to the loopback interface.

Verify Communication and Polling

After you secure communications, confirm that the system is polling.

Follow these steps:

1. Log in to NetOps Portal.
2. Select **Administration**, **Data Collector List**.
3. Verify that the status for each Data Collector is **Collecting Data**.
4. Wait 5 minutes, then click the **Refresh** button.
5. Verify that the status for each Data Collector is still **Collecting Data**.
6. (Optional) For further validation, look at a device on each Data Collector, and confirm that polled data is being loaded.

If any Data Collector does not have the **Collecting Data** status, use the following options to troubleshoot the Data Collector:

- On the Data Collector host, verify the ActiveMQ status:
  ```
  service activemq status
  ```
- Check for errors in Data Collector **Karaf** log: `/opt/IMDatacollector/apache-karaf-<vers>/karaf.log`
- Check for errors in Data Collector **ActiveMQ** log: `/opt/IMDataCollector/broker/apache-activemq-version/data/activemq.log`
- Check the permissions on activemq.conf and the **.ks** and **.ts** files. These files must be readable by the user that is attempting to run them.
- Verify the contents of the **.ks** and **.ts** files:
  ```
  keytool -list -keystore client.ts
  ```

Automate Device Inventory Synchronization

You want to automate the maintenance of the Data Aggregator device inventory through automated synchronization with a CMDB source. You use a configuration management database to maintain an inventory of all devices within your environment. This information is kept in one place, where it is then provisioned out to various monitoring tools. You want to build an automated means of keeping Data Aggregator device inventory in synchronization with your CMDB. To do so, configure NetOps Portal and Data Aggregator before you write your integration script.

To manually execute web service calls, use either a REST client editor or an HHTP tool that sends requests and gets responses to perform these procedures. In this scenario, we simply refer to the REST client editor.

> **NOTE**
> For automated inventory control, you would write an application or script that leverages the web services described in this documentation.

This process requires making calls to two different web servers:

- The NetOps Portal web server is used to create SNMP profiles.
- The Data Aggregator web server is used to create discovery profiles and review discovery results.

To discover devices and monitor components in your network using REST web services, follow these steps:

> **NOTE**
> Perform step 1 using NetOps Portal REST web services. Perform steps 2-3 using the Data Aggregator REST web services.

Provide SNMP Profile Credentials

SNMP profiles are definitions that contain the information necessary to enable secure queries of device MIBs using SNMP. These definitions provide SNMP parameters from NetOps Portal to the Data Aggregator data source, as needed, while ensuring data security.

For this scenario, you will create an SMPv3 profile.

NOTE
The methods that return SNMP profile definitions do not include community strings, user names, or passwords. The save method takes a community string, user name, or passwords that are not encrypted. We recommend that you only invoke this method on the computer where NetOps Portal is installed.

Follow these steps:

1. Access the NetOps Portal web server.
2. Open a REST client that is configured to access the NetOps Portal server. Log in using administrator credentials.

   NOTE
   For information about configuring your REST client, see the REST client documentation.
3. Set the Content-type to application/xml. Enter the Body text and modify the attributes as needed:
   - **Profile Name**
     Defines a name for the SNMP profile.

     NOTE
     Profile names must be unique, cannot be duplicated across SNMP versions, and are not case-sensitive.
   - **SNMP Version**
     Specifies the version of SNMP that the profile uses. Enter SNMPv3 for this scenario.
   - **Port**
     Identifies the port that is used to make SNMP connections to devices associated with this profile.
     Default: 161
   - **User Name**
     Identifies the user for the profile, whose secret keys were used potentially to authenticate and encrypt the SNMPv3 packets. The User Name is a character string.
   - **Context Name**
     Specifies a collection of management information that is accessible by an SNMP entity. The Context Name is necessary for providing end-to-end identification and for retrieving data from an SNMPv3 agent. The Context Name is an octet string.

     NOTE

     The Data Aggregator does not use Context Name on the SNMPv3 profiles to communicate with the device.
   - **Security Level**
     Specifies the security level to use. Enter one of the following values:
     - NoAuthNoPriv
     - AuthNoPriv
     - AuthAndPriv
   - **Authentication Protocol**
     Specifies the authentication protocol to use when contacting devices associated with this profile. Enter one of the following algorithms for authenticating SNMPv3 packets:
     - None (Do not attempt authentication.)
     - MD5 (Message Digest 5)
     - SHA (Secure Hash Algorithm)
   - **Authentication Password**
     Specifies the password for authentication using SNMPv3 and the selected authentication protocol. The password must contain a minimum of eight characters.
   - **Verify Authentication Password**
     Confirms the authentication password.
   - **Privacy Protocol**
     (Optional) Specifies the encryption protocol to use for data flows sent to any devices or servers that are associated with this profile, as follows:

- None (Does not encrypt communications. Use only with NoPriv option.)
- DES
- AES (128-bit encryption)
- Triple DES

Note: The privacy protocol option is only enabled when authentication is enabled for this profile.

– **Privacy Password**
(SNMPv3 only) Defines the password that is used when exchanging encryption keys.
– **Verify Privacy Password**
(SNMPv3 only) Confirms the password that is used when exchanging encryption keys.
– **Rank**
Specifies the rank of the profile in the global list of SNMP profiles.
– **Enabled**
For NetOps Portal, this indicates whether the information in this profile is used when not explicitly assigned to a device. For Data Aggregator, this value must be set to **true**.
– **TenantID**
Specifies the tenant ID. The default tenant ID is 8. Enterprise customers must enter the default tenant ID.

Example: No authentication and no privacy

```
<SnmpProfile>
    <name>Tokyo</name>
    <port>161</port>
    <userName>myuser</userName>
    <context></context>
    <version>Version3</version>
    <securityLevel>NoAuthNoPriv</securityLevel>
    <authProtocol>None</authProtocol>
    <authPassword>None</authPassword>
    <privProtocol>None</privProtocol>
    <privPassword>None</privPassword>
    <rank>4</rank>
    <enabled>true</enabled>
    <tenantID>8</tenantID>
</SnmpProfile>
```

Example: Authentication and no privacy

```
<SnmpProfile>
    <name>Brasil</name>
    <port>161</port>
    <userName>myuser</userName>
    <context></context>
    <version>Version3</version>
    <securityLevel>AuthNoPriv</securityLevel>
    <authProtocol>MD5</authProtocol>
    <authPassword>test</authPassword>
    <privProtocol>None</privProtocol>
    <privPassword>None</privPassword>
    <rank>3</rank>
    <enabled>true</enabled>
```

```
    <tenantID>8</tenantID>
</SnmpProfile>
```

Example: Authentication and privacy

```
<SnmpProfile>
    <name>Boston</name>
    <port>161</port>
    <userName>myuser</userName>
    <context></context>
    <version>Version3</version>
    <securityLevel>AuthAndPriv</securityLevel>
    <authProtocol>MD5</authProtocol>
    <authPassword>test</authPassword>
    <privProtocol>TripleDES</privProtocol>
    <privPassword>test</privPassword>
    <rank>1</rank>
    <enabled>true</enabled>
    <tenantID>8</tenantID>
</SnmpProfile>
```

4. POST the following URL to create the profile:

```
POST http://pc_hostname:8181/pc/center/webservice/profiles/saveProfile/{true|false}
```

   – ***pc_hostname***
     Specifies the NetOps Portal host name. (8181 is the required port.)
   – **{true|false}**
     Specifies a Boolean value for the rankTiesAscendingByDate parameter. **True** indicates that the profile you are adding will be the last in rank order (as determined by the creation date of the SNMP profile).

   The XML returns true when the operation succeeds.
   The SNMP profile is automatically synchronized with Data Aggregator and is available for inventory discovery to use it.

Create a Discovery Profile and Run a Discovery

Discovery profiles specify how discovery operates in your Data Aggregator environment. Within a discovery profile, you can specify the IP addresses, IP address ranges, and host names you want to discover devices for.

> **NOTE**
> This scenario is for an enterprise environment and discovers devices using the SNMP protocol. However, MSPs may also want to limit discovery to specific SNMP profiles. A full list of attributes is available by accessing the URL: http://*da_hostname port*/rest/discoveryprofiles/documentation.

You create a discovery profile and run a discovery in one operation when using the REST web services. When discovery is run, devices are discovered based on the discovery profile you create.

Follow these steps:

1. Access the Data Aggregator web server.
2. Open a REST client editor and set the Content-type to application/xml.
3. Find and make a note of the default tenant ID using the following URL:

```
GET http://da_hostname:port/rest/tenants
```

- ***da_hostname*:*port***
  Specifies the Data Aggregator host name and the port number.
  Default port: 8581

4. Enter the Body text for the discovery profile in a REST client and modify the attributes as needed.

   NOTE
   You can view the XSD schema which defines the structure of the discovery profile XML by going to the URL:

   `http://da_hostname:port/rest/discoveryprofiles/XSD/getlist.xsd`

5. Include IP addresses and host names so that the end-user switches in your network are discovered. Do one or more of the following actions:
   - Type individual IP addresses for which you want to discover devices in the IP Address List field.
   - Type the host names for which you want to discover devices in the Host List field.

   NOTE
   These fields accept comma-delimited values. Single quotes, double quotes, backward slashes, forward slashes, and ampersands are not permitted.

6. Set the following attributes:
   - **Name**
     Specifies a descriptive name for the discovery profile. This field cannot contain single quotes, double quotes, backward slashes, forward slashes, and ampersands.
   - **RunStatus**
     Specifies whether to run the discovery. For this scenario, set the attribute to START.

     NOTE
     To rerun discovery later, update (PUT) the run status to START.

   Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<DiscoveryProfile version="1.0.0">
  <ActivationStatus>true</ActivationStatus>
  <IPRangesList>
    <IPRanges>10.0.64.202-10.0.64.206</IPRanges>
  </IPRangesList>
  <HostNamesList>
    <HostNames>ahost</HostNames>
  </HostNamesList>
  <IPListList>
    <IPList>10.10.10.10</IPList>
  </IPListList>
  <RunStatus>READY</RunStatus>
  <Item version="1.0.0">
    <Name>BigRange_TestProfileViaAPI</Name>
  </Item>
  <IPDomainMember version="1.0.0">
    <IPDomainID>285</IPDomainID>
  </IPDomainMember>
</DiscoveryProfile>
```

7. Save and run the new discovery profile by entering the following URL:

```
POST http://da_hostname:port/rest/tenant/default_tenant_ID/discoveryprofiles
```

– **tenant/*default_tenant_ID***
Specifies the ID number for the default tenant workspace. Enterprise customers typically only use the default tenant workspace. For the POST operation, you *must* enter the default tenant ID when there are no other tenants.

The discovery profile is created and discovery runs.
All discovered devices are automatically added to the appropriate out-of-the-box device collection or another user-created device collection.

8. Verify the discovery profile displays in the discovery profiles list. Note the ID of the new discovery profile because it is used to review the discovery results.
```
GET http://da_hostname:port/rest/discoveryprofiles
```

Review the Discovered Devices and Instances

Verify that your initial discovery was successful by reviewing the discovered devices and instances.

Follow these steps:

1. To view a list of discovery profiles, access the Data Aggregator web server. Type the following URL:
```
GET http://da_hostname:port/rest/discoveryprofiles
```

– ***da_hostname*:*port***
Specifies the Data Aggregator host name and the port number.
Default port: 8581

A list of discovery profiles and their instance IDs is returned.

2. Find the discovery instance ID for the discovery profile that you created:

NOTE
Example:

```
<DiscoveryInstanceIDList relatesURL="relatesto/instances"
            rootURL="discoveryinstances">
                <ID>236</ID>
 </DiscoveryInstanceIDList>
```

3. View the discovery instance details using the following URL:
```
GET http://da_hostname:port/rest/discoveryinstances/instance_ID
```

The instance XML returns information about new and existing devices and SNMP profiles that were tested.
Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<DiscoveryInstance version="1.0.0">
    <ID>236</ID>
    <IPSweepTotalSuccess>5</IPSweepTotalSuccess>
    <CompletionTime>Thu Apr 12 12:36:35 CDT 2012</CompletionTime>
    <ExistingFoundDevicesList>
        <ExistingFoundDevices>241</ExistingFoundDevices>
        <ExistingFoundDevices>239</ExistingFoundDevices>
        <ExistingFoundDevices>240</ExistingFoundDevices>
        <ExistingFoundDevices>238</ExistingFoundDevices>
```

```
        </ExistingFoundDevicesList>
        <IPSweepCompletionTime>Thu Apr 12 12:36:35 CDT 2012</IPSweepCompletionTime>
        <ExistingFoundManageableDevicesList>
            <ExistingFoundManageableDevices>241</ExistingFoundManageableDevices>
            <ExistingFoundManageableDevices>239</ExistingFoundManageableDevices>
            <ExistingFoundManageableDevices>240</ExistingFoundManageableDevices>
            <ExistingFoundManageableDevices>238</ExistingFoundManageableDevices>
        </ExistingFoundManageableDevicesList>
        <StartTime>Thu Apr 12 12:36:31 CDT 2012</StartTime>
        <NewlyCreatedDevicesList>
            <NewlyCreatedDevices>383</NewlyCreatedDevices>
        </NewlyCreatedDevicesList>
        <TestedCommProfilesList>
            <TestedCommProfiles>199</TestedCommProfiles>
            <TestedCommProfiles>198</TestedCommProfiles>
        </TestedCommProfilesList>
        <IPSweepStartTime>Thu Apr 12 12:36:31 CDT 2012</IPSweepStartTime>
        <ProgressPercentage>100</ProgressPercentage>
        <IPSweepTotal>7</IPSweepTotal>
        <ProfileID>235</ProfileID>
        <NewlyCreatedManageableDevicesList>
            <NewlyCreatedManageableDevices>383</NewlyCreatedManageableDevices>
        </NewlyCreatedManageableDevicesList>
        <PingResponseDeviceCount>5</PingResponseDeviceCount>
        <CompletionStatus>SUCCESS</CompletionStatus>
        <Item version="1.0.0">
            <CreateTime>Thu Apr 12 12:36:31 CDT 2012</CreateTime>
        </Item>
    </DiscoveryInstance>
```

4. (Optional) Review information about specific devices (new or existing) using the following URL:
   `GET http://da_hostname:port/rest/devices/device_ID`

   – ***endpoint_ID***
   Specifies the ID of the referenced endpoint, such as devices. Use the IDs you found in the previously.
5. (Optional) Review information about the SNMP profiles that were tested during discovery using the following URL:
   `GET http://da_hostname:port/rest/profiles/profile_ID`

All discovered devices and instances appear.

Back Up the Data Aggregator

To avoid losing your settings and custom certifications due to an unexpected failure, back up the data aggregator. Back up the data aggregator and the data repository during specific events, such as before an upgrade.

You do not need to stop the data repository, the data collector, or the data aggregator services when you back up the data aggregator. Backups are stored in the location that you specify, which can be on the data aggregator system or on a different backup host system.

Prerequisite: You have root or sudo privileges.

Follow these steps:

1. From a command prompt, create a backup a backup directory in a secure location on the same or different backup host system:
   `mkdir DA_Backup`
   – ***DA_Backup***
   Specifies the directory path and name of the backup directory.
2. Create subdirectories within DA_Backup using all the following commands:
   ```
   mkdir DA_Backup/deploy_backup
   mkdir DA_Backup/cert_backup
   mkdir DA_Backup/MIBDepot_backup
   mkdir DA_Backup/CustomDeviceType_backup
   mkdir DA_Backup/etc
   mkdir DA_Backup/data
   ```
3. Run the following commands to back up the files on the DA:
   – This command backs up the deploy directory:
   `cp -r DA_install_directory/apache-karaf-version/deploy/* DA_Backup/deploy_backup`
   - ***DA_install_directory***
   Specifies the Data Aggregator install directory.
   Default: `/opt/IMDataAggregator`
   – This command backs up your certifications:
   IMPORTANT
   Do not back up the `local-jms-broker.xml` file and the README files from that directory.
   `cp -r DA_install_directory/data/certifications DA_Backup/cert_backup`
   In a fault tolerant environment, a shared directory (example: `/DASharedRepo`) is defined to help limit data loss. Therefore, in a fault tolerant environment the files would be located in the following directory:
   `cp -r DASharedRepo/certifications DA_Backup/cert_backup`
   For more information, see Fault Tolerance.
   – This command backs up all the custom MIBs in the MIBDepot directory:
   `cp -r DA_install_directory/apache-karaf-version/MIBDepot/ DA_Backup/MIBDepot_backup`
   NOTE
   If there are no customizations, this directory might be empty.
   – This command backs up all the custom device subtype xml files:
   `cp DA_install_directory/data/custom/devicetypes/DeviceTypes.xml DA_Backup/CustomDeviceType_backup/`
   In a fault tolerant environment, the files would be located in the following directory:
   `DASharedRepo/custom/devicetypes/DeviceTypes.xml`
   – This command backs up the etc directory files:
   `cp -r DA_install_directory/apache-karaf-version/etc DA_Backup/`
   – This command backs up the data/log directory:
   `cp -r DA_install_directory/apache-karaf-version/data/log DA_Backup/data/`

NOTE
By default, a `caVerticaUtility.sh` script is located in the `/opt/CA/IMDataRepository*/` directory. The script is a support tool available for exporting the iRep data from an existing system and importing it into a different schema.You can then query the iRep data to determine how the original system was configured.

Choose Another Host in a Cluster When Selected Host Fails

If the database host that is specified during Data Aggregator installation fails at runtime, Data Aggregator shuts down automatically. If you installed Data Repository in a cluster, point database connections to another host in the cluster before you restart Data Aggregator.

If more than one host in the Data Repository cluster fails, Data Repository and Data Aggregator shuts down automatically. The Data Repository cluster is only capable of losing one host.

If a single host in the cluster that is *not* specified during the Data Aggregator installation disconnects from the network (for example, because a firewall was put in place, or the Ethernet cable was removed), Data Aggregator shuts down. Data Aggregator restarts automatically if you set up the automatic recovery of the Data Aggregator process during the Data Aggregator installation. Once the host that is offline becomes available, return that host to the cluster. Select the “Restart Vertica on Host” option on the main menu of the admintools utility and follow the prompts.

If a single host in the cluster that is *not* specified during the Data Aggregator installation is stopped through the “Kill Vertica Process on Host” option on the Advanced Menu of the admintools utility, Data Aggregator continues to function. Once the host that is offline becomes available, return that host to the cluster. Select the “Restart Vertica on Host” option on the main menu of the admintools utility and follow the prompts.

Follow these steps:

1. Open the *DA_install_directory*`/apache-karaf-<`*vers*`>/etc/dbconnection.cfg` file on the Data Aggregator host.
2. Modify the following line in the dbconnection.cfg file. Modify the line to reference a hostname or IP address of one of the Data Repository cluster hosts that is still up and running:

   ```
   dbUrl=jdbc:vertica://database server hostname:database server port/databasename?
   use35CopyFormat=true&BinaryDataTransfer=false
   ```

   – ***database server hostname*:*database server port***
   Indicates the hostname or IP address of Data Repository and the Data Repository port number that you entered during the Data Aggregator installation.
   Default port number: 5433

   Example:
   If host2 is up and running in the cluster and you choose database connections to point to host2, your updated dbUrl entry could look like the following line:

   ```
   dbUrl=jdbc:vertica://host2:5433/mydatabasename?
   use35CopyFormat=true&BinaryDataTransfer=false
   ```

3. Save the dbconnection.cfg file.
4. Do one of the following steps:
   – Start the Data Aggregator service:

     ```
     service dadaemon start
     ```

NOTE

For RHEL 7.x or OL, `service` invokes `systemctl` . You can use `systemctl` instead.

– (Fault tolerant environment) Run one the following commands to enable the fault tolerant Data Aggregator so that it can start when necessary:

- **RHEL 6.x:**

```
service dadaemon activate
```

- **RHEL 7.x, SLES, or OL:**

```
DA_Install_Directory/scripts/dadaemon activate
```

5. To help ensure that Data Aggregator is not still running, type the following command:

```
Ps  - ef | grep java | grep  - v grep
```

Data Aggregator processes are not returned when Data Aggregator is not running.
Database connections point to the specified host in the cluster going forward.

Configure Data Collector When the Data Aggregator IP Address Changes

If the IP address of the Data Aggregator host changes, update the Data Collectors to point to the new IP. If the Data Collector uses the hostname to communicate with the Data Aggregator, restart the Data Collector. If the Data Collector uses the IP address to communicate with the Data Aggregator, complete the following procedure:

Follow these steps:

1. Log in to the Data Collector host.
2. Stop the Data Collector service:

```
service dcmd stop
```

3. Locate and edit the IP address in the following file:
`/opt/IMDataCollector/apache-karaf-<vers>/etc/com.ca.im.dm.core.collector.cfg`

NOTE

is the default installation directory.

4. Edit the following line:

```
...

collector-manager-da-hostname=
DA_host_IP
```

```
...
```

Save the file.

5. Locate and edit the following file:
   `/opt/IMDataCollector/broker/apache-activemq-<vers>/conf/activemq.xml`
6. Edit the IP address in the following lines:

```
...

<networkConnector name="da_manager" uri="static:(tcp://
DA_host_IP
:61616)" duplex="true" suppressDuplicateTopicSubscriptions="false"/>

<networkConnector name="da_manager-PRQ" uri="static:(tcp://
DA_host_IP
:61618)" duplex="true" suppressDuplicateTopicSubscriptions="false"/>

<networkConnector name="da_manager-IREP" uri="static:(tcp://
DA_host_IP
:61620)" duplex="true" suppressDuplicateTopicSubscriptions="false"/>

<networkConnector name="da_manager-blob" uri="static:(tcp://
DA_host_IP
:61622)" duplex="true" suppressDuplicateTopicSubscriptions="false"/>

...
```

In a fault tolerant tolerant environment, the content of the activemq.xml has the following format:

```
...

<networkConnector name="da_manager" uri="static:(failover:(tcp://
DA1_host_IP
:61616,tcp://
DA2_host_IP
:61616)?maxReconnectAttempts=3)" duplex="true"
 suppressDuplicateTopicSubscriptions="false"/>

<networkConnector name="da_manager-PRQ" uri="static:(failover:(tcp://
DA1_host_IP
:61618,tcp://
DA2_host_IP
:61618)?maxReconnectAttempts=3)" duplex="true"
 suppressDuplicateTopicSubscriptions="false"/>
```

```
<networkConnector name="da_manager-IREP" uri="static:(failover:(tcp://
DA1_host_IP
:61620,tcp://
DA2_host_IP
:61620)?maxReconnectAttempts=3)" duplex="true"
 suppressDuplicateTopicSubscriptions="false"/>

<networkConnector name="da_manager-blob" uri="static:(failover:(tcp://
DA1_host_IP
:61622,tcp://
DA2_host_IP
:61622)?maxReconnectAttempts=3)" duplex="true"
 suppressDuplicateTopicSubscriptions="false"/>

...
```

7. Start the Data Collector:

```
service dcmd start
```

8. Verify that the correct address appears in the Data Collector List.
   a. Open NetOps Portal as an administrator.
   b. Select Admin, Data Sources, and click a Data Aggregator data source.
   c. Click Data Collectors from the System Status menu.
      The IP address of each Data Collector appears under the 'Address' column. The status of each Data Collector is "Collecting Data".

Data Aggregator Configuration Changes During Network Disconnects to a Data Collector Host

Occasionally, the connection between a Data Aggregator host and a Data Collector host breaks, such as, when a network disconnect occurs. If the Data Aggregator and Data Collector processes are running during a disconnect, you can make configuration changes to the Data Aggregator installation. In this case, polling continues on the Data Collector host according to the configuration that existed before the network disconnect. Once the connection between the Data Aggregator and the Data Collector hosts reestablishes, Data Collector downloads the new configuration and adjusts polling accordingly.

For example, you make one of the following configuration changes:

- Change the expression an SNMP vendor certification uses to calculate a value on a metric family.
- Change the metric family to poll a new operational metric.

When the connection between the Data Aggregator and the Data Collector hosts is broken, the changes cannot take effect. After reconnection, Data Collector begins polling the new SNMP MIB objects used in the new expression or in calculating the new operational metric.

Manage Data Collector Installations

Select an IP domain and a tenant for each Data Collector installation. Each Data Collector instance that does discovery requests can be associated with only one IP domain.

IP domains are logical groupings that identify data from different devices and networks. IP addresses with associated interfaces or applications that belong to separate customer networks are monitored separately. When combined with appropriate permissions, IP domains are monitored from a single console, but users view data only for the domains that they monitor.

A tenant represents a customer environment that a managed service provider administers. Each tenant environment is independent and effectively functions as a separate instance of NetOps Portal. Each instance can contain multiple users and roles that are not shared among tenants.

The Default Tenant represents the tenant space for the managed service provider within the managed infrastructure. Assign the Default Tenant if you are not deploying multi-tenancy. In a single-tenant environment, the Default Tenant is the space used for monitoring the entire infrastructure.

Follow these steps:

1. Open CA NetOps Portal as an administrator.
2. Select **Administration**, **Data Sources**, and click a Data Aggregator data source.
3. Select a Data Collector instance from the list.
4. Verify that Data Collector is available for assignment. The Polled Items column lists the number of polled devices and components that are assigned to this Data Collector instance.

   WARNING
   If the number of polled devices and components is greater than one, you cannot change the tenant or IP domain assignment for the Data Collector instance.
5. Click **Assign**.
   The Assign Data Collector dialog opens.
6. Select the tenant that you want to assign to this Data Collector instance from the drop-down list.
   All monitored devices and components that this Data Collector instance discovers are automatically associated with this tenant.
   Select 'Default Tenant' if you want to use the default tenant.
7. Select the IP domain that you want to associate with this Data Collector instance.
   All managed devices and components that this Data Collector instance discovers are automatically associated with this IP domain.
8. Click **Save**.
   The tenant and IP domain are assigned to the Data Collector installation.

Migrate the Data Aggregator

To migrate the Data Aggregator, install the Data Aggregator on a new host with a new IP address and hostname and copy over your files. The following situations might require migration:

- You are moving to new hosts for a major OS upgrade (for example, RHEL 6.9 to RHEL 7.3).
- The current database hardware no longer meets sizing requirements.
- You are moving from virtual machines to physical hardware for the database.

NOTE

You must upgrade the existing system to the product version you are migrating to before migrating.

Install the Data Aggregator on the New Host

Prepare the host and install the Data Aggregator on the new host. During installation, use the details for the Data Repository for the new Data Aggregator host.

You can find the Data Repository details on the current Data Aggregator host in the following location:

- *DA_Install_Directory*/apache-karaf-<*version*>/etc/dbconnection.cfg

For more information, see Prepare to Install the Data Aggregator and Install the Data Aggregator.

WARNING

If the Data Repository was migrated, follow the Data Aggregator installation instructions in the console, and answer NO when prompted to drop the schema and the installation continues. If you answer YES, you lose your migrated data.

Migrate the Data Aggregator to the New Host

Create a backup and copy it over to the new host.

NOTE

To avoid corrupting the Vertica database, ensure the old Data Aggregator host is not pointing to the new Data Repository host.

Follow these steps:

1. Back up the Data Aggregator. For more information, see Back Up Data Aggregator.
2. Package the files:
   ```
   tar czf DA.tgz backup_dir
   ```
3. Copy the backup directory from the old Data Aggregator host to the new Data Aggregator host.
   ```
   scp -r backup_dir/DA.tgz DA_user@backuphost:/tmp
   ```
   - ***DA_user***
     User who owns the DA process (root or sudo user)

     NOTE

     If /tmp is not a preferred location, specify another directory location.
4. Restore the Data Aggregator. For more information, see Restore Data Aggregator.
5. Do one of the following steps:
   - Stop the Data Aggregator service:
     ```
     service dadaemon stop
     ```

     NOTE

     For RHEL 7.x or OL, `service` invokes `systemctl` . You can use `systemctl` instead.
   - (Fault tolerant environment) If the local Data Aggregator is running, run one the following commands to shut it down and prevent it from restarting until maintenance is complete:
     - **RHEL 6.x:**
       ```
       service dadaemon maintenance
       ```
     - **RHEL 7.x, SLES, or OL:**
       ```
       DA_Install_Directory/scripts/dadaemon maintenance
       ```
6. Do one of the following steps:
   - Start the Data Aggregator service:
     ```
     service dadaemon start
     ```

- (Fault tolerant environment) Run one the following commands to enable the fault tolerant Data Aggregator so that it can start when necessary:
  - **RHEL 6.x:**

    ```
    service dadaemon activate
    ```

  - **RHEL 7.x, SLES, or OL:**

    ```
    DA_Install_Directory/scripts/dadaemon activate
    ```

Update the Data Repository Configuration

On the Data Aggregator host, ensure the `dbconnection.cfg` file points to the correct Data Repository hosts.

Follow these steps:

1. Open the *DA_installation_directory*`/apache-karaf-<`*version*`>/etc/dbconnection.cfg` file on the Data Aggregator host.
2. Modify the following line in the `dbconnection.cfg` file. Modify the line to reference the hostname or IP address of each Data Repository host:

   `dbHostNames=`*dbNode1Hostname*`,`*dbNode2Hostname*`,`*dbNode3Hostname*

3. Save the `dbconnection.cfg` file.
4. Do one of the following steps:
   - Restart the Data Aggregator service:

     ```
     service dadaemon start
     ```

   - (Fault tolerant environment) Run one the following commands to enable the fault tolerant Data Aggregator so that it can start when necessary:
     - **RHEL 6.x:**

       ```
       service dadaemon activate
       ```

     - **RHEL 7.x, SLES, or OL:**

       ```
       DA_Install_Directory/scripts/dadaemon activate
       ```

Update Data Collector Configurations

Update the Data Collectors to point to the new Data Aggregator host. On each Data Collector host, edit the Data Aggregator host information in following files:

NOTE

If you are upgrading or migrating the Data Collectors, do this procedure after the upgrade or migration.

- `/opt/IMDataCollector/apache-karaf-<`*version*`>/etc/com.ca.im.dm.core.collector.cfg`
- `/opt/IMDataCollector/broker/apache-activemq-<`*version*`>/conf/activemq.xml`

Run the following commands to restart the Data Collector Karaf and Data Collector ActiveMQ services:

```
 service activemq stop
service activemq start
service dcmd stop
service dcmd start
```

For more information, see Configure Data Collector When the Data Aggregator IP Address Changes.

Update the Data Aggregator Data Source in NetOps Portal

Update the Data Aggregator Data Source in NetOps Portal to point to the new Data Aggregator host. For more information, see Configure a Data Source.

Follow these steps:

1. Go to **Administration**, and click **Data Sources**.
2. Click **Edit** and specify the IP address or hostname of the new Data Aggregator host.
3. Configure the Data Aggregator settings:

   NOTE

   If SSL is configured, specify HTTPS for the communication protocol.

   – **Synchronize component items that are not currently present on the monitored device**
   When the Data Aggregator finds a device component that is no longer present in the environment, that status of the component is set to **Not Present**. By default, the Data Aggregator does not synchronize these items because data can no longer be collected for the item. Historical data that has not reached the data retention limit is still available for these items.

   WARNING

   If the properties of an active component match the identifying properties of the not present component, the components are indistinguishable. Data from the old component might contribute to group based dashboards instead of data from the active component. Enable this feature only under the following circumstances:

   - You want to report on historical data for items that are no longer present in the environment.
   - You can ensure that the not present item does not conflict with an actively monitored item.
   - You can identify the not present items so that you can exclude the items from groups.

   – **Discover devices from other data sources**
   This option specifies whether synchronization includes devices that are reported by other data sources. Automatic synchronization includes only devices that are discovered after you select this option. To discover previously discovered devices, perform a full synchronization of the Data Aggregator.
   DX NetOps Performance Management creates a discovery profile that includes the IP addresses of the devices. This discovery profile attempts discovery once per day.
4. Click **Save**.
   If the Data Aggregator data source is enabled, data appears in NetOps Portal after the next synchronization.

Modify Maximum Memory Usage for Data Aggregator and Data Collector Components

For the Data Aggregator and the Data Collectors, the default maximum memory is 80% of the total system memory. AMQ uses 20% of total memory on both components. Both components reserve 2 GB of memory for the operating system. If you add more memory to the system, modify the maximum values to match the changes.

Follow these steps:

1. Open a console and type the following command:
   ```
   more /proc/meminfo
   ```
   The total memory usage is displayed.
2. Make a note of this total memory.

3. Modify the maximum memory for Data Aggregator by performing the following steps:
   a. Access the following file:
      DA_installation_directory/apache-karaf-<*vers*>/bin/setenv

   b. Modify the following line:
      IM_MAX_MEM=*numberUnit*

      - **numberUnit**
        Specifies the maximum amount of memory. *number* is a positive integer. *Unit* is “G” for GB or “M” for MB. Reserve 2 GB for operating system operations and reserve 20 percent for AMQ. Use the following formula to determine the value:
        total memory * 80% - 2 GB

      Example:
      ```
      33554432 KB * 80% - 2G = 24 GB
      IM_MAX_MEM=24G
      ```

   c. Save the file.
   d. Do one of the following steps:
      - Start the Data Aggregator service:
        ```
        service dadaemon start
        ```

        NOTE

        For RHEL 7.x or OL, `service` invokes `systemctl` . You can use `systemctl` instead.
      - (Fault tolerant environment) Run one the following commands to enable the fault tolerant Data Aggregator so that it can start when necessary:
        - **RHEL 6.x:**
          ```
          service dadaemon activate
          ```

        - **RHEL 7.x, SLES, or OL:**
          DA_Install_Directory/scripts/dadaemon activate

   e. In order for the memory setting change to persist during a Data Aggregator upgrade, modify the /etc/DA.cfg file, replacing the updated value for the property "da.memory".
      Example:
      ```
      da.memory=24G
      ```

4. Modify the maximum memory for all Data Collector hosts by performing the following steps:
   a. Access the following file:
      DC_installation_directory/apache-karaf-<*vers*>/bin/setenv

   b. Modify the following line:
      IM_MAX_MEM=*numberUnit*

      - **numberUnit**
        Specifies the maximum amount of memory. *number* is a positive integer. *Unit* is “G” for GB or “M” for MB. Reserve 2 GB for operating system operations and reserve 20 percent for AMQ. Use the following formula to determine the value:

total memory * 80% - 2 GB

c. Save the file.
d. Restart Data Collector hosts using the following commands:

```
service dcmd stop
service dcmd start
```

e. In order for the memory setting change to persist during a Data Collector upgrade, modify the /opt/DCM.cfg, replacing the updated value for the property "IM_MAX_MEM".

The maximum amount of memory is configured.

Example

The following example configures the maximum memory usage for Data Aggregator where the total memory is 48 GB:

1. Open a console and type the following command:

```
more /proc/meminfo
```

The following result appears:

```
MemTotal: 50331648KB
```

2. Calculate the maximum memory:
   Equation: total memory * 80% - 2 GB = maximum memory
   Solution: 50331648 KB * 80% - 2 GB≅ 38 GB
3. Access the following file:

```
DA_installation_directory/apache-karaf-<vers>/bin/setenv
```

4. Modify the following line:

```
IM_MAX_MEM=38G
```

5. Save the file.
6. Restart Data Aggregator.
7. Modify the da.memory value in the /etc/DA.cfg file:

```
da.memory=38G
```

The maximum amount of memory is modified.

Modify the External ActiveMQ Memory Limit

The Data Aggregator installer calculates the memory that is needed on your system to accommodate the ApacheMQ process. However, you can manually modify the memory limit settings to fine-tune ActiveMQ on your Data Aggregator system. For example, you can modify the settings under the following circumstances:

- When the system memory has changed.
- When the number of Data Collector systems have changed.
- To optimize the memory settings.
- When you have determined that ActiveMQs performance is degraded, by monitoring either the JConsole or the DX NetOps Performance Management custom chart with ActiveMQ metrics.

Follow these steps:

1. Calculate the amount of memory for the Data Aggregator ActiveMQ based on the following settings:
   – **Maximum java heap size**

This value is set to 20% system memory by default. The minimum value is 512M.
- **Initial minimum java heap size**
  This value should be 50% of maximum java heap size.
- **Memory limit for all messages**
  This value should be 50% of the maximum java heap size.
- **Memory limit per queue**
  This value should be calculated based on how many Data Collector installations you have.
  Example: The memory per queue
  (system memory for all messages)/5/(Data Collector count)

2. Calculate the amount of memory for the Data Collector ActiveMQ based on the following settings:
   - **Maximum java heap size (-Xmx)**This value is equal to 1GB, which is the recommended minimum.
   - **Initial java heap size (-Xms)**This value is equal to 50% of the maximum java heap size. By default, the value is equal to 512M.
   - **Default Disk Cache Size**This value is equal to 50% of the maximum Data Collector memory. By default, this value is equal to 45 minutes, or 500K, of data when you use a 5-minute poll rate.
   - **Default Drop Rate**This value is equal to 10% of the default disk cache size. The checker captures the amount of data in the cache every 30 seconds, and drops 10% of the data when the maximum cache size is reached.
3. Log in to the Data Aggregator host as the root user or the sudo user.
4. Stop the ActiveMQ broker:
   ```
   service activemq stop
   ```
5. Modify the java heap size for ActiveMQ:
   a. Access the following file:
      DA_installation_directory/scripts/activemq
   b. Modify the following line:
      ```
      ACTIVEMQ_OPTS_MEMORY=" -Xms788M -Xmx2575M -Xmn394M -server -XX:SurvivorRatio=6 -
      XX:+UseConcMarkSweepGC -XX:+UseParNewGC ...
      ```
      Change - **Xms** to be the Initial minimum java heap size.
      Change - **Xmx** to be the Maximum java heap size.
   c. Save the changes.
6. Modify the ActiveMQ memory limit for the producer flow control:
   a. Access the following file:
      DA_installation_directory/broker/apache-activemq-*vers*/conf/activemq.xml
   b. Locate the following line and change the value to Memory limit for all messages:
      ```
      <memoryUsage limit="value"/>
      ```
   c. Locate the following line, change the value to Memory limit per queue:
      ```
      <policyEntry queue=">" producerFlowControl="true" memoryLimit="value"/
      ```
   d. Save the changes.
7. In order for the memory setting change to persist during a Data Aggregator upgrade, modify the /etc/DA.cfg file, replacing the updated value for the property "da.activemq.memory".
   Example:
   ```
   da.activemq.memory=value
   ```
8. Start the ActiveMQ broker:
   ```
   service activemq start
   ```

ActiveMQ starts and uses the new memory settings.

Configure the Data Aggregator Cleanup

For 20.2.3 and higher, a cleanup runs daily by default to remove deleted items from the Data Aggregator. You can configure whether the cleanup runs. You can also disable the cleanup for attribute instance tables or relationships.

To configure when the cleanup occurs, see Schedule Data Purges.

Follow these steps:

1. Go to the following location:
   DA_install_directory/apache-karaf-2.4.3/etc
2. Create the following file in this location:
   com.ca.im.dm.core.database.dao.impl.DeletedItemCleanupDAO.cfg
3. Specify the following parameters and save the file:
   Example:
   ```
   cleanupEnabled=true
   attributeCleanupEnabled=true
   relationshipCleanupEnabled=true
   ```

Monitor System Health

System Health dashboards monitor system utilization and identify performance trends and degradation. In each of the suggested views, observe the value trend to detect abnormal changes.

To view the System Health dashboards, you need the View System Health Dashboards role right and the System Health - Data Aggregator group that is assigned to you. By default, only the Administrator has the View System Health Dashboards role right.

Use the following derived rollup metrics to identify changes:

- **Average**
  The average value across the resolution window
- **Maximum** The maximum value within the resolution window. Use this metric to visualize high peaks.
- **95th Percentile**
  95 percent of the samples within the resolution window have a lower value. Use this metric to smooth out spikes.

The following System Health dashboards are available:

Data Aggregator / Data Collector Health Dashboard

Data Aggregator Event Processing Dashboard

The Data Aggregator Event Processing dashboard monitors the health of event processing.

This dashboard has built-in thresholds, which can help you to identify issues with event processing. The following values indicate that the system is in good health:

- **Event Queue** - Less than two. An Event Queue Size of greater than two indicates that the system is behind, or that the system is at risk.
- **Poll Cycle Percent** - The Threshold Evaluation engine continues to run if the Poll Cycle Percent is less than 50%. When you use a 1-minute poll rate, the evaluation has to complete in less than 30 seconds. When you use a 5-minute poll rate, the evaluation has to complete in less than 150 seconds.
- **Events Created and Cleared** - The sum of Created and Cleared Events during a 5-minute interval is less than 900.
- **Event Evaluations** - Less than 150,000 during a 5-minute interval.

Event processing deteriorates if any of the following conditions occur:

- The Data Aggregator event processing is slow to process rules.
- You create and clear more than the recommended number of events.
- Event processing is backing up.

Data Aggregator General Processing Dashboard

The Data Aggregator General Processing dashboard monitors all Data Aggregator activities. To customize the views to fit the needs of your environment, specify thresholds on this dashboard. The following values or conditions indicate that the system is in good health:

- **Rate Data Loading Times**
  Less than 30 seconds
- **Roll-Up Calculation Times**
  Spikes in the Roll-Up Calculation Times after 00:30 UTC
  Daily and weekly processing causes these normal spikes.
- **Baseline Calculation Times**Spikes in the Baseline Calculation Times after 00:30 UTC
  Daily and weekly processing causes these normal spikes.
- **Data Repository Maintenance Times**Normal trend changes
  The maintenance times vary based on the size of the schema.
- **Reporting ETL Processing Times**
  Execution time does not exceed 15 minutes.
  Consistent processing time unless a significant number of new devices have been discovered.
- **Reporting ETL Count of Loaded Rows**The count is consistent with the number of items that the system is monitoring.
  This count tracks the number of dimension items, both device and component items, loaded into the reporting table.

Data Aggregator Polling Dashboard

The Data Aggregator Polling dashboard is used to determine the polling load on the Data Aggregator and the Data Collectors.

The following values help you to monitor the system health:

- **Polled Item Count** - To determine the threshold for the polled item count, use the DX NetOps Performance Management Sizing Tool.
- **Calculated Metrics per Second** - This value is equal to the value that you identified when setting up your environment, unless there is a change in the metric families or items in your environment.

Consider the following when using this dashboard:

- If the Data Aggregator is polling too many items, redistribute polled items to other Data Collectors.
- Observe the polling statistics to monitor the polling health, and to see whether the Data Aggregator is polling devices too quickly.
- The Stopped Polls by Device view helps you to identify problematic devices, which cause excessive polling, such as when the device is unavailable.

  TIP

  If you have multiple Data Collectors, hover over the graph for accurate information for each Data Collector.

Data Aggregator Queries Dashboard

The Data Aggregator Queries dashboard monitors the quantity and timing of RIB and OpenAPI queries. The RIB queries and OpenAPI queries views specify the number of corresponding queries for a specific time period. The following value indicates that the system is in good health:

RIB Query Processing TImes - Less that 120 seconds.

Update the Data Collector

You can migrate the Data Collector without having to rediscover network devices and components or lose historical data. For example, if you are a tool administrator, your server administrator can instruct you to relocate Data Collector to another host. Data Collector is polling 500,000 devices and components, and you do not want to lose data or perform rediscovery.

The Data Collector component can be moved even if you have device packs installed.

Note the following considerations:

- The amount of data loss is equal to the amount of time that has elapsed from the time the old Data Collector component is shut down to the time when the new Data Collector component has been deployed.
- If the old Data Collector component happens to start accidentally, the SNMP data is polled twice. A warning appears in the Data Aggregator karaf log:

  ```
  WARN  | Session Task-810 | 2013-01-02 13:52:09,062 | DCHeartBeatLog |
  ore.collector.interfaces |
        | HeartBeat message not received.  Expected: 93, Received: 255
  ```

  To fix this problem, stop or uninstall the old Data Collector component.

To move Data Collector to another system, follow this process:

NOTE

Backup and restore is not required for the Data Collector. For installations that use CA Mediation Manager, migrate the device packs before you install the Data Collector on the new host.

Determine the Unique Identifier for Data Collector

Determine the unique identifier for Data Collector before moving this component to another host.

Retrieve the Data Collector ID using *one* of the following methods:

- Log in to NetOps Portal as a user with the Administrator role, and do the following steps:
  a. Select **Administration**, and select a Data Aggregator data source from the menu.
     The Data Aggregator Admin UI opens.
  b. Select **System Status**, **Data Collectors** from the menu.

c. Find the Data Collector component that you want to move and notate its ID.
The format of the Data Collector component ID is HOSTNAME:UUID.

- Open a web browser and issue the following web service call:

```
http://DA_hostname:port/rest/dcms
```

DA_hostname:port
Specifies the Data Aggregator host name and the port number.
Default port: 8581
Find the <DataCollectionMgrInfo> section whose HostName and IPAddress match the one you want to move. Note the value for <DcmID>.

Stop Data Collector

Stop the Data Collector services on the current host before moving Data Collector to another host.

Follow these steps:

1. If you have installed device packs for this Data Collector, take the following steps. If no device packs are installed, proceed to Step 2.
   a. Log in to NetOps Portal as a user with the Administrator role.
   b. Select **Administration**, and then select a Data Aggregator data source from the menu.
   The Data Aggregator Admin UI opens.
   c. Select **EMS Integration Profiles** from the **Monitoring Configuration** menu.
   d. Right-click on a profile that is associated with this Data Collector host, and select **Stop**. Do this step for every EMS profile that is related to this Data Collector host.
   e. Archive CA Mediation Manager artifacts by running this command:

```
tar -zcvf filename /opt/IMDataCollector/apache-karaf-{n.n.n}/MediationCenter
```

   - ***filename***
     Specifies the name of the archive file, which is moved to the new Data Collector host.
2. Log on to the Data Collector host and run the following command:

```
service dcmd stop
```

3. Verify that Data Collector has stopped:
   a. Log in to NetOps Portal as a user with the Administrator role.
   b. Select **Administration**, and select a Data Aggregator data source from the menu.
   c. Select **System Status**, **Data Collectors** from the menu.
   d. Verify that the Data Collector status is "Not Connected".

Reinstall the Data Collector on a Clean Host

After you stop the Data Collector services on the original host, reinstall the Data Collector on a clean host. Use the same `DCM_ID` . Doing so allows the Data Collector to retrieve the original Data Collector configuration from the Data Aggregator when the new Data Collector process starts.

Verify that DX NetOps Mediation Manager and DX NetOps Virtual Network Assurance point to the correct Data Collectors:

- For DX NetOps Mediation Manager, see the CAMM documentation.
- For DX NetOps Virtual Network Assurance, see the CA VNA documentation.

Follow these steps:

1. (DX NetOps Mediation Manager only) Migrate your device packs. On the old Data Collector host, run the *$CAMM_HOME*/tools/migratePMtoCAMM script with the - t flag.
   This step assumes that you are running the script on a Data Collector server where a Local Controller is installed. You must also have the CA Mediation Manager Console running on another server. For more information, see the CAMM documentation.
2. (DX NetOps Virtual Network Assurance only) For DX NetOps Virtual Network Assurance, see the CA VNA documentation.
3. Log in to the new host system and open a command shell session.
4. Set an environment variable with the ID of the data collector by running this command:
   ```
   export DCM_ID=data collector id
   ```
   – ***data collector id*** The DcmID that we obtained as the unique identifier for the Data Collector in the previous section.
5. Install Data Collector from the same session by running the **install.bin** binary. For more information, see Install the Data Collectors.
6. Install the CA Mediation Manager LC on the same server.
7. If you have previously installed device packs for this Data Collector, perform these additional steps:
   a. Copy the zip files that you created previously with the migration script to local directories on this host.
   b. Use the CA Mediation Manager web console to deploy these device packs and start them.
      You do not need to redeploy the certification packs to the Data Aggregator host.
8. (Optional) After several polling cycles, verify that data is being collected on the new host.Uninstall the old Data Collector, and delete any associated EMS profiles.

Rebalance the Load on Data Collector

As a Data Collector instance monitors more devices, the capacity can be exceeded and the Data Collector can become overloaded. You can transfer the workload from one overloaded Data Collector instance to other Data Collector instances. You can rebalance the load on Data Collector in two ways:

- Select the Data Collector instances that you want to rebalance and click **Rebalance**. The product automatically rebalances the load among the selected Data Collector instances.

  NOTE

  When the load on the Data Collectors is rebalanced, DX NetOps Mediation Manager and DX NetOps Virtual Network Assurance devices are not rebalanced. Only SNMP devices are rebalanced.
- Move selected devices from one Data Collector instance to another.

  WARNING
  Do not rebalance or move many items during peak hours. These operations might negatively affect end-user performance.

Follow these steps:

1. Open CA NetOps Portal as an administrator.
2. Select **Administration**, **Data Sources**, and click a Data Aggregator data source.
3. Click **Data Collectors** from the **System Status** menu.
   You can see the number of devices and components that each Data Collector installation is polling. You can also see the total number of devices that are assigned to each Data Collector instance, including devices that are not currently polled.

Automatically Rebalance the Load on Data Collector

1. Select the Data Collector instances that you want to rebalance and click **Rebalance**.

   NOTE
   Be sure to select Data Collector instances within the same IP domain. Only Data Collector instances within the same IP domain can rebalance devices between one another.
2. A confirmation dialog displays the current device and polled item count for each selected Data Collector and the proposed resulting device and polled item counts.

   NOTE
   Devices can only be moved to Data Collector instances that can contact them.
3. Click **Yes**.

   NOTE
   Rebalancing polled items restarts the baseline average calculations for all rebalanced polled items.

Move Selected Devices to a Specific Data Collector Instance

1. Select the Data Collector instance that you want to move selected devices from.
2. In the Devices table, select the devices that you want to move to another Data Collector instance and then click Move Devices.
3. The Move Devices to Selected Data Collector dialog opens.
4. Select the Data Collector instance that you want to move your selected devices to from the drop-down list.

   NOTE
   Devices can only be moved to Data Collector instances that can contact them. Only Data Collector instances that are within the same IP domain are included for selection.
5. Click Yes.

   NOTE
   Moving devices restarts the baseline average calculations for the moved devices.

Restore Data Aggregator

You can restore the Data Aggregator files that you backed up. If Data Repository remains intact, you can restore only the Data Aggregator component.

Data Aggregator can remain running while you restore the backup. The files that are backed up can be dropped in the desired directories while Data Aggregator is running.

Backup and restore into the same product version.

NOTE
You must have root or sudo privileges to perform this task.

Follow these steps:

1. Open a command prompt.
2. (Optional) In the situations where the Data Aggregator karaf service is not running, uninstall the existing Data Aggregator and reinstall it.
3. Run all of the following commands:

```
cp -r DA_Backup/deploy_backup/* DA_install_directory/apache-karaf-version/deploy/
cp -r DA_Backup/cert_backup/* DA_install_directory/data/
cp -r DA_Backup/MIBDepot_backup/* DA_install_directory/apache-karaf-version/
MIBDepot/
```

```
cp DA_Backup/CustomDeviceType_backup/DeviceTypes.xml DA_install_directory/data/
custom/devicetypes/
cp -r DA_Backup/etc DA_install_directory/apache-karaf-version/
cp -r DA_Backup/data/log DA_install_directory/apache-karaf-version/data/
```

If prompted, overwrite the existing file.

- ***DA_Backup***
  Specifies the directory path and name of the backup directory.
- ***DA_install_directory***
  Specifies the Data Aggregator install directory.
  Default: /opt/IMDataAggregator

In a fault tolerant environment, a shared directory (example: `/DASharedRepo`) is defined to help limit data loss. Therefore, in a fault tolerant environment the `certifications` and `devicetype` files would be located in the following directories:

```
DASharedRepo/certifications
DASharedRepo/custom/devicetypes/DeviceTypes.xml
```

For more information, see Fault Tolerance.

4. Wait for a few minutes for Data Aggregator to synchronize automatically with CA NetOps Portal. When the connections between the Data Aggregator and the Data Collector hosts are established, the Data Collector hosts resume polling.
   Data Aggregator is restored.

   NOTE
   If you must restore Data Collector to a previous state, you can uninstall and reinstall Data Collector.

View the Health of the System

The System Status page provides the following health statuses:

- **Data Source Synchronization**
  View the overall health status of your data sources. To view details for each data source, click the expand arrow to the right.
- **Data Aggregator**
  View the overall health status of your Data Aggregators. To view details for each Data Aggregator, click the expand arrow to the right.
- **Data Collector**
  View the overall health status of your Data Collectors. To view details for each Data Collector, click the expand arrow to the right. The Data Collector List shows the tenant and IP domain to which each Data Collector installation is assigned, and the Data Collector status and version. You can also see the total number of devices and components that each Data Collector installation is polling, and that are assigned to the Data Collector instance, including devices that are not currently polled. The administrator can see a list of Data Collector installations for all tenants. Tenant administrators can see only the Data Collector installations that are assigned to their tenant.
- **VNA Gateway**
  View the overall health status of your VNA Gateway. To view details for the VNA Gateway, click the expand arrow to the right.
- **Scheduled Report Repository**
  View the overall health status of your Scheduled Report Repository. To view details, click the expand arrow to the right.
- **Report Generation Services**

View the overall health status of your Report Generation Services. To view details for reports on All Pages and reports for the First Page, click the expand arrow to the right. The following operational statuses apply to the Report Generation Services:
  - **Low Load**
    0-1 reports are waiting to run.
  - **Medium Load**
    2-5 reports are waiting to run.
  - **High Load**
    6 or more reports are waiting to run.

  You can also see the percentage of that last 12 hours that each operation status occurred.
- **Alarm Status Service**
  View the overall health status of your Alarm Status Service. To view the details for the Alarms Status Service, click the expand arrow to the right.
- **Usage Data**
  View your overall usage data. To view details, click the the expand arrow to the right.
- **Network Flow Analysis**
  View the overall health status of your Network Flow Analysis components. To view the details the Network Flow Analysis components, click the expand arrow to the right.

Follow these steps:

1. Log in as an Administrator.
2. Hover over **Administration**, and click **Data Sources: System Status**.
3. To view details, click the expand arrow to the right of the overall health status.

View Data Aggregator Details

You can view the total number of manageable and pingable devices that Data Aggregator monitors for all tenants. Individual device totals for each tenant are also displayed in a table. Tenant administrators can only view the total number of manageable and pingable devices that Data Aggregator monitors for their tenant.

You can also view the version and the build number of Data Aggregator.

Follow these steps:

1. Log in as an Administrator.
2. Select **Administration**, **Data Sources**, and click a Data Aggregator data source.
3. Click **Data Aggregator** from the System Status menu.
   The **Data Aggregator List** page opens.

Data Repository Administration

The Data Repository uses Vertica database software. The Data Repository does not support any customization.

- *Do not* deploy custom projections or modify database parameters.
- The Data Repository *does not support* a Vertica k-safety value greater than 1.
- The Data Repository *does not support* the use of Management Console, which is an Vertica management tool. This product can modify the Vertica database configuration.

The following topics provide information about configuring the Data Repository:

Configure Data Retention Rates

The default data retention rates for the Data Repository help to conserve disk space and improve reporting, and can be customized. Polled data is generated each poll cycle for all devices, and this data represents the most granular data available in the product. This raw, polled data is set to roll up at hourly, daily, and weekly aggregation levels.

Because higher-level aggregated data requires less disk space, you can keep this data for a longer period than polled data.

You can change how long the Data Repository retains the polled data, hourly rollup data, daily rollup data, and weekly rollup data. For example, you can change the polled data retention value to 30 days to conserve disk space. Find the balance that best suits your needs and environment.

By default, data is retained in Data Repository for the following number of days:

- Polled data: 45 days
- Hourly rollup data: 90 days
- Daily rollup data: 365 days
- Weekly rollup data: 730 days

The minimum number of days that Data Repository can retain data for is as follows:

- Polled data: 2 days
- Hourly rollup data: 8 days

  NOTE

  When the hourly retention rate is less than 32 days, the calendar heat chart views show daily data (only one sample per day). The charts appear sparse as a result.
- Daily rollup data: 31 days
- Weekly rollup data: 366 days

Follow these steps:

1. Enter the following information in a web browser:
   `http://hostname:port/rest/globalretentiondefinition`
   - ***hostname*:*port***
     Use this parameter to specify the Data Aggregator host name and the port number.
     Default port: 8581
2. Take note of the ID that is assigned to the globalretentiondefinition.
3. Look for the following elements:
   - `GtdRollupDataRetentionPeriod`
   - `DailyRollupDataRetentionPeriod`
   - `PolledDataRetentionPeriod`
   - `HourlyRollupDataRetentionPeriod`

   This information helps you determine which types of data you want to modify the retention period for.
4. Look for the elements that has the HistoricConfigurationDataRetentionPeriod parameter. This information helps you specify the number of days to keep item repository attribute values.
5. Open a REST client editor or HTTP tool that sends requests and gets responses and set the Content-type to application/xml.
6. Enter the following criteria:
   - URL: `http://hostname:port/rest/globalretentiondefinition/ID`
     - **ID**

A unique identification number that is assigned to the globalretentiondefinition parameter

- HTTP method = PUT
- Enter the retention periods that you want to change in the Body tab of the HTTP Request pane.
  For example:

```
<GlobalRetentionDefinition version="1.0.0">

<PolledDataRetentionPeriod>4</PolledDataRetentionPeriod>

</GlobalRetentionDefinition>
```

WARNING
Verify that there is no white space at the beginning of each of these lines, otherwise the PUT fails.

In this example, the polled data retention period has been changed to four days.

Results are returned in the Body tab of the HTTP Response pane.
For example:

```
<GlobalRetentionDefinitionList>

  <GlobalRetentionDefinition version="1.0.0">

    <ID>4</ID>

    <GtdRollupDataRetentionPeriod>730</GtdRollupDataRetentionPeriod>

    <HistoricConfigurationDataRetentionPeriod>7</
HistoricConfigurationDataRetentionPeriod>

    <DailyRollupDataRetentionPeriod>365</DailyRollupDataRetentionPeriod>

    <PolledDataRetentionPeriod>4</PolledDataRetentionPeriod>

    <HourlyRollupDataRetentionPeriod>90</HourlyRollupDataRetentionPeriod>

    <Item version="1.0.0">

      <CreateTime>Thu Dec 08 16:03:05 CST 2011</CreateTime>

      <Name>Global Retention Definition</Name>

    </Item>

  </GlobalRetentionDefinition>

</GlobalRetentionDefinitionList>
```

In this example, the polled data retention period has been changed to four days. The default retention periods for weekly rollup data, daily rollup data, and hourly rollup data remain.

Back Up the Data Repository

To protect your data, back up the Data Repository.

> **WARNING**
>
> The following procedure is the supported method for backing up the Data Repository. Taking a virtual machine snapshot is *not* a supported method for backing up the Data Repository.

The first backup is a full backup of all historical data. Subsequent backups are incremental and include all database activity that occurred since the snapshot at the start of the previous backup.

About Data Repository Backups

- The Data Repository and Data Aggregator continue to run during a Data Repository backup.
- Backup processing can be resource-intensive, but is prioritized below other processing. To let backups to proceed more quickly, and to minimize the impact to other processing, perform backups during non-peak hours.
- You can back up Data Repository to a remote host, or you can back it up to the same host. If you back up to the same host, save the backup to a different disk than the one that is used by the catalog and data directories.
- Perform full backups weekly. Perform incremental backups daily.
- Full backups occur only when the backup location is a new directory. The snapshotName can be the same as previous backups.
- The incremental snapshots store new files and hard links to unchanged files from the previous backup. Restoring to any incremental snapshot depends on the integrity of the files that are linked to in previous snapshots.
- For information about the size of the backup files, see the DX NetOps Performance Management Sizing Tool.

To ensure data integrity, back up each node of the data repository to a dedicated backup host. To prepare for the backup, perform the following procedure for each Data Repository node.

Before you begin, verify the following information about the Data Repository host and the remote backup host:

- Neither host is connected to LDAP.
- Neither host is connected to Network Information Service (NIS) and have the same Vertica Linux database administrator user.
- Port 50000 is open on any firewalls so that the Data Repository host can access the custom rsync/ssh port 50000 on the backup host.

> **NOTE**
>
> If you do not have a backup host, you can back up the Data Repository locally. For more information, see Configure the Data Repository Host for a Local Backup.

Follow these steps:

1. Log in to the backup host as the root user.
2. Create the Vertica Linux database administrator user on the remote backup host:
   ```
   useradd db_admin -s /bin/bash
   ```
   db_admin is the same Vertica Linux database administrator user that exists on the Data Repository hosts.
3. Set the Vertica Linux database administrator user password:
   ```
   passwd db_admin
   ```
4. Create the Vertica directories on the remote backup host:
   ```
   mkdir /opt/vertica/bin
   mkdir /opt/vertica/oss
   ```
5. Change the owner of the Vertica directories:
   ```
   chown -R db_admin /opt/vertica
   ```

6. Log out from the remote backup host.
7. Set up passwordless ssh on the Data Repository host for the remote backup host:
   a. Log in to the Data Repository host as the Vertica Linux database administrator user.
   b. Generate the keys for passwordless ssh:
      ```
      ssh-keygen -N "" -t rsa -f ~/.ssh/id_rsa
      cp ~/.ssh/id_rsa.pub ~/.ssh/authorized_keys2
      chmod 644 ~/.ssh/authorized_keys2
      ```
   c. Copy the Vertica Linux database administrator user public key into the list of authorized keys on the remote backup host:
      ```
      ssh-copy-id -i dradmin@backuphost
      ```
   d. Log in to the remote backup host as the Vertica Linux database administrator user.
   e. Copy the Vertica rsync and python tools from the Data Repository host to the remote backup host:
      ```
      scp dradmin@drhost:/opt/vertica/bin/rsync /opt/vertica/bin
      scp -r dradmin@drhost:/opt/vertica/oss/python /opt/vertica/oss
      ```
8. Verify that the remote backup host has the following directories:
   – /opt/vertica/bin/rsync
   – /opt/vertica/oss/python
9. Create the backup directory:
   ```
   mkdir backup_directory
   ```
   backup_directory specifies the directory where you want to save the backup files. Select a backup directory that is on a disk partition with a large amount of free space. If these directories are not writable by the database administrator user, give this user access to these directories.
   The remote host is ready for the backup configuration file to be created and for the backup directory to be initialized.

Configure the Data Repository Backup

To back up the Data Repository and configure automatic backups, create a configuration file for the backup. Vertica performs a full backup during the first backup into a new backup directory. All subsequent backups to the same directory are incremental backup, even if the snapshot name changes.

The node where you perform this procedure initiates the backup.

To configure the vbr utility, Vertica automatically installs sample configuration files at the following location:

`/opt/vertica/share/vbr/example_configs .`

For more information, see the Vertica documentation.

Follow these steps:

1. Log in to the Data Repository host as the database administrator user.
2. Create a password file:
   Example: `/opt/vertica/config/password.txt`

   NOTE

   You can choose a different location for the password file.

   ```
   [Passwords]
   ; Specified password for db admin account
   dbPassword = DBpassword
   ; Specifies password for rsync user account – if different than DB admin
   ; serviceAccessPass = rsyncpwd
   ```

```
; Specifies password for the dest_dbuser Vertica account. Used only for restoring to
 alternate cluster.
; dest_dbPassword = DestinationPwd
```

3. Go to the sample configuration files:
   `/opt/vertica/share/vbr/example_configs`
   The database administrator user requires write privileges for the directory.
4. Copy, edit, and deploy a configuration file for backup. See the following examples from Vertica.
   Example 1: `backup_restore_full_local.ini`
   Back up the Data Repository to a local area on the same machine. The backup can be a mount from an external shared drive or a local disk. You cannot use the same disk as `data/catalog`.

```
[Mapping]
; node_name = backup_host:backup_dir
; [] indicates backup to localhost
v_drdata_node0001 = []:/backups
v_drdata_node0002 = []:/backups
v_drdata_node0003 = []:/backups


[Misc]
; Backups with the same snapshotName form a time sequence limited by
 restorePointLimit.
; SnapshotName is used for naming archives in the backup directory, and for
 monitoring and troubleshooting.
; Valid values: a-z A-Z 0-9 - _
snapshotName = backup_snapshot


[Misc]
; The temp directory location on all database hosts.
; The directory must be readable and writeable by the dbadmin, and must implement
 POSIX style fcntl lockf locking.
; tempDir = /tmp/vbr


; Specifies the number of historical backups to retain in addition to the most recent
 backup.
; 1 current + n historical backups
restorePointLimit = 7


; Full path to the password configuration file
; Store this file in directory readable only by the dbadmin.
passwordFile = /opt/vertica/config/password.txt


; When enabled, Vertica confirms that the specified backup locations contain
; sufficient free space and inodes to allow a successful backup. If a backup
```

```
; location has insufficient resources, Vertica displays an error message explaining
 the shortage and
; cancels the backup. If Vertica cannot determine the amount of available space
; or number of inodes in the backupDir, it displays a warning and continues
; with the backup.
; enableFreeSpaceCheck = True


; When performing a backup, replication, or copycluster, specifies the maximum
; acceptable difference, in seconds, between the current epoch and the backup epoch.
; If the time between the current epoch and the backup epoch exceeds the value
; specified in this parameter, Vertica displays an error message.
; SnapshotEpochLagFailureThreshold = 3600
```

Example 2: `backup_restore_full_external.ini` Back up the Data Repository to a different machine. Replace the IP addresses with the IP address of the backup host(s).

```
[Mapping]
; node_name = backup_host:backup_dir
; In this "parallel backup" configuration, each node backs up to a distinct external
 host.
; To backup all database nodes to a single external host, use that single hostname/IP
 address in each entry below.
v_drdata_node0001 = 1.1.1.1:/backups
v_drdata_node0002 = 2.2.2.2:/backups
v_drdata_node0003 = 3.3.3.3:/
backups
[Misc]
; Backups with the same snapshotName form a time sequence limited by
 restorePointLimit.
; SnapshotName is used for naming archives in the backup directory, and for
 monitoring and troubleshooting.
; Valid characters: a-z A-Z 0-9 - _
snapshotName =
 backup_snapshot
[Misc]
; The temp directory location on all database hosts.
; The directory must be readable and writeable by the dbadmin, and must implement
 POSIX style fcntl lockf locking.
; tempDir = /tmp/
vbr
; Specifies the number of historical backups to retain in addition to the most recent
 backup.
; 1 current + n historical backups
restorePointLimit =
 7
; Full path to the password configuration file
; Store this file in directory readable only by the dbadmin
; (no default)
```

```
passwordFile = /opt/vertica/config/
password.txt
; When enabled, Vertica confirms that the specified backup locations contain
; sufficient free space and inodes to allow a successful backup. If a backup
; location has insufficient resources, Vertica displays an error message explaining
 the shortage and
; cancels the backup. If Vertica cannot determine the amount of available space
; or number of inodes in the backupDir, it displays a warning and continues
; with the backup.
; enableFreeSpaceCheck =
 True
; When performing a backup, replication, or copycluster, specifies the maximum
; acceptable difference, in seconds, between the current epoch and the backup epoch.
; If the time between the current epoch and the backup epoch exceeds the value
; specified in this parameter, Vertica displays an error message.
; SnapshotEpochLagFailureThreshold = 3600
```

5. (First Time Only) Initialize the backup directory before the first time you run the backup.

   ```
   /opt/vertica/bin/vbr.py --task init --config-
   file configuration_directory_path_filename
   ```

   – ***configuration_directory_path_filename*** indicates the directory path and filename of the configuration file that you will reference when you restore. This file is located where you ran the backup utility.

   Once initialized, multiple configuration files can use the directory if the files share the same backup directory location.
6. Back up Data Repository:

   ```
   /opt/vertica/bin/vbr.py --task backup --config-
   file configuration_directory_path_filename
   ```

   For example:

   ```
   /opt/vertica/bin/vbr.py --task backup --config-file /home/vertica/vert-db-
   production.ini
   ```
7. If you are prompted about the authenticity of the host, answer yes.
   The Data Repository starts the backup. This process can take a long time, especially for a full backup.
8. (Optional) If you do not want to retain the Data Repository password in clear text for future manual backups.

   WARNING
   The configuration file that is generated contains a clear text password. Automated backups require the password. This procedure prevents automated backups from this configuration file.

   a. Verify that the following line exists under the [Database] section:

      ```
      dbPromptForPassword = True
      ```

   b. Remove the following line from the [Database] section:

      ```
      dbPassword = password
      ```

Set Up an Automatic Backup

To ensure regular backups of the Data Repository, create a cron job to schedule automatic backups. The first backup is a full backup and the following backups are incremental. Run a full backup weekly or biweekly. Vertica performs a full backup only when you use a new backup directory.

TIP

Run a full backup weekly. If disk space is limited, retain only two to three weeks of data. Delete the oldest backup file at the beginning of each week. Use the vbr utility remove task to delete old backups. Vertica does not support removing backups through the file system.

Follow these Steps:

1. Create a wrapper shell script that contains the following line:
   ```
   /opt/vertica/bin/vbr.py --task backup --config-
   file configuration_directory_path_filename
   ```
   configuration_directory_path_filename indicates the directory path and filename of the configuration file that you will reference when you restore.
2. Save the contents to a new file named backup_script.sh in a location of your choice.
   For example:
   ```
   /home/vertica/backup_script.sh
   ```
3. Change permissions for running the script:

   WARNING

   `chmod 777` makes the file readable, writable, and executable by everyone. If you want only the script owner to run the file, use `chmod 700` . If you want only the root user to run the file, use `chmod 755` .

   ```
   chmod 777 location_backup_script.sh/backup_script.sh
   ```
   For example:
   ```
   chmod 777 /home/vertica/backup_script.sh
   ```
4. As the database administrator user, open the crontab to define a cron job:
   ```
   crontab -e
   ```
5. Add a cron job that runs the backup script.

   TIP

   Create a cron job to run the script daily at an off-peak time.

   For example:
   ```
   00 02 * * *   /home/vertica/backup_script.sh >/tmp/backup.log  2>&1
   ```
   This example cron job runs the backup script every day at 2:00 AM.
   The cron job runs a daily incremental backup.
6. Add a script to copy the configuration file, and change the snapshot name in the configuration file. Also use a new backup directory in the configuration file to cause Vertica to perform a full backup.

   WARNING

   Do not delete the previous configuration file. The original configuration file is required to remove a backup or restore from an older series of backups.
7. (Optional) Remove older backup sequences as required with the vbr utility with the remove task using the configuration that was used to create it.
   ```
   /opt/vertica/bin/vbr.py --task remove --archive=[<date>_<time>|"all"] --config-
   file configuration_directory_path_filename
   ```
   The remove command is destructive and removes the data and free space on the disk. The archive must be specified to remove a single restore point, a comma separated list, or "all". To display the list of backups, run `--task listbackup` .

VBR Utility Reference

The Vertica vbr utility lets you back up and restore either the full database, or one or more schema and table objects of interest. You can also copy a cluster and list backups you created previously.

For a full reference for the vbr utility, see the Vertica Documentation.

Configure the Data Repository Host for a Local Backup

If you do not have a dedicated backup host, you can back up the Data Repository locally. For best data integrity, we recommend a dedicated backup host.

For more information about Data Repository backups, see Back Up the Data Repository.

Follow these steps:

1. Log in to the Data Repository host as the database administrator user.
   In a cluster installation, you can log in to any host in the cluster.
2. Verify that the database administrator user is set up with a passwordless ssh key. In a cluster installation, ensure that passwordless ssh keys are set up for *each* host that is participating in the cluster.

   ```
   ssh hostname ls
   ```

   hostname is the name of the host where Data Repository is installed.
   If the passwordless ssh key is set up, you are *not* prompted for a password. If youare prompted for a password, set up passwordless ssh:
   a. Press Ctrl+C.
   b. Set up the Linux user account for the database administrator user with a passwordless ssh key:

      ```
      ssh-keygen -N "" -t rsa -f ~/.ssh/id_rsa

      cp ~/.ssh/id_rsa.pub ~/.ssh/authorized_keys2

      chmod 644 ~/.ssh/authorized_keys2
      ```

   c. Copy the SSH credentials to the other hosts in the cluster:

      ```
      ssh-copy-id -i dradmin@other-hostname-in-the-cluster
      ```

   d. Confirm that you are *not* prompted for a password:

      ```
      ssh hostname ls
      ```

3. Create a backup directory for each host in the cluster:

   ```
   mkdir backup_directory
   ```

   backup_directory specifies the directory where you want to save the backup files. Select a backup directory that is on a disk partition with a large amount of free space. If these directories are not writable by the database administrator user, give this user access to these directories.

The Data Repository Host is ready for the backup to run.

Restore Data Repository

You can restore Data Repository from an existing backup. To restore Data Repository, ensure that the database administrator user is part of the sudoers file.

WARNING

Do not restore a backup of an older release into a newer environment. The data from the older version might not be fully compatible with schema changes.

Usually, you restore Data Repository to the same computer where you backed it up from. However, you *can* restore Data Repository to a different computer. The computer that you restore to must be configured in the same way that the computer you backed up Data Repository from is. In a cluster environment, each computer you restore to must be configured in the same way that each computer you backed up each Data Repository node from is.

The following configurations must be the same:

- the IP address
- the hostname
- the catalog and data directories
- the catalog and data directory permissions
- the Vertica Linux database administrator user credentials
- the database administrator user account credentials
- the database user account credentials

Follow these steps:

1. Stop all Data Collector hosts that are associated with Data Aggregator by logging in to the computers where Data Collector is installed as the root user or a sudo user with access to a limited set of commands. Open a command prompt and type the following command:

   ```
   service dcmd stop
   ```

   NOTE

   For RHEL 7.x or OL, `service` invokes `systemctl` . You can use `systemctl` instead.

   Data Collector hosts stop.
2. Stop Data Aggregator by logging in to the computer where Data Aggregator is installed as the root user or a sudo user with access to a limited set of commands. Do one of the following steps:
   – Stop the Data Aggregator service:

     ```
     service dadaemon stop
     ```

   – (Fault tolerant environment) If the local Data Aggregator is running, run one the following commands to shut it down and prevent it from restarting until maintenance is complete:
     - **RHEL 6.x:**

       ```
       service dadaemon maintenance
       ```

     - **RHEL 7.x, SLES, or OL:**

```
DA_Install_Directory/scripts/dadaemon maintenance
```

Data Aggregator stops.

3. Log in to the database server you use for Data Repository as the database administrator user, *not* as the root user.
4. Type the following command:

```
/opt/vertica/bin/adminTools
```

The Administration Tools dialog opens.

5. Select (4) Stop Database.
6. Press the Space bar next to the database name, select OK, and press Enter.
   You are prompted for the database password.
7. Enter the database password and press Enter.
   Data Repository stops.

   NOTE
   If Data Repository does not stop, select (2) Stop Vertica on Host from the (7) Advanced Tools Menu.
8. Select Exit and press Enter.
9. To prepare to restore the Data Repository backup, log in as the Linux user account for the database administrator user to the database server you use for Data Repository.
   When you set up automatic backups of Data Repository, you configured the configuration file with a restore point of seven. Data Repository can be restored to the most recent backup or to any of the previous seven incremental backups.
10. Do one of the following steps:
    a. To restore Data Repository to the most recent backup, type the following command:

```
/opt/vertica/bin/vbr.py --task restore --config-
file configuration_directory_path_filename
```

- ***configuration_directory_path_filename***
  Indicates the filename and directory path of the configuration file you created when you ran the backup configuration procedure. This file is located where you ran the backup utility (/opt/vertica/bin/vbr.py).

For example:

```
/opt/vertica/bin/vbr.py --task restore --config-file /home/vertica/vert-db-
production.ini
```

NOTE
In a cluster installation, you can run the restore task from any of the hosts that are participating in the cluster.

TIP

For a list of available restore points, run the following command:

```
/opt/vertica/bin/vbr.py --task listbackup --config-
file configuration_directory_path_filename
```

    b. To restore Data Repository to any of the previous seven incremental backups, type the following command:

```
/opt/vertica/bin/vbr.py --task restore --config-
file configuration__directory_path_filename  --archive archive_name
```

- **configuration_directory_path_filename**
  Indicates the filename and directory path of the specific configuration file you want to restore a specific archive from. You created this configuration file when you ran the backup configuration procedure. This file is located where you ran the backup utility (/opt/vertica/bin/vbr.py).
- **archive_name**
  Indicates the name of the specific restore point that you want to restore to. Change to the backup directory that the configuration file for the restore point indicates. All of the restore points that are available are listed. Determine the archive name for the restore point that you want to restore to.

For example:

```
/opt/vertica/bin/vbr.py --task restore --config-file myconfig.ini  --
archive 20131020_170018
```

NOTE
In a cluster installation, you can run the restore task from any of the hosts that are participating in the cluster.

TIP

For a list of available restore points, run the following command:

```
/opt/vertica/bin/vbr.py --task listbackup --config-
file configuration_directory_path_filename
```

11. Restart Data Repository by logging in to the computer where Data Repository is installed as the database administrator user, *not* as the root user. Open a command prompt and do the following steps:
    a. Type the following command:

    ```
    /opt/vertica/bin/adminTools
    ```

    The Administration Tools dialog opens.
    b. Select (3) Start Database.
    c. Press the Space bar next to the database name, select OK, and press Enter.
    You are prompted for the database password.
    d. Enter the database password and press Enter.
    Data Repository starts.
    e. Select Exit and press Enter.
12. Restart the Data Aggregator. Log in to the Data Aggregator host as the root user or a sudo user, and start the dadaemon service. Do one of the following steps:
    – Start the Data Aggregator service:

    ```
    service dadaemon start
    ```

    – (Fault tolerant environment) If the local Data Aggregator is running, run one the following commands to shut it down and prevent it from restarting until maintenance is complete:
      - **RHEL 6.x:**

      ```
      service dadaemon maintenance
      ```

- **RHEL 7.x, SLES, or OL:**

```
DA_Install_Directory/scripts/dadaemon maintenance
```

Data Aggregator starts.

13. Restart the Data Collectors. Log in to each Data Collector host as the root user or a sudo user, and start the dcmd service:

    Data Collector hosts start.

Add a Node to the Data Repository Cluster

To increase system performance or replace a node, add a node to the Data Repository cluster.

> **NOTE**
>
> This process applies to expanding a node that was set up with a hostname. If a node was set up using localhost, the nodes table in Vertica has node_address or node_ip set to 127.0.0.1. To expand a node that was set up using localhost, contact CA Support.

For more details about this process, see the Vertica documentation.

Complete the Prerequisites

Before you add the new node to the Data Repository cluster, complete the prerequisites:

- Stop the Data Aggregator:

```
service dadaemon stop
```

  > **NOTE**
  > Keep the Data Aggregator stopped until you complete the entire process.

- Prepare the new host for the Data Repository installation. For more information see, Prepare to Install the Data Repository.
- Back up the Data Repository. For more information, see Back Up the Data Repository.

Prepare the New Host

Before you add the host to the cluster, configure and validate the drinstall.properties file.

Follow these steps:

1. Copy the installDR.bin file to /tmp on the new host.
2. On the new host, extract the installation files from the BIN file:

```
./installDR.bin
```

3. Configure the drinstall.properties file.

   > **WARNING**
   >
   > The parameters must match the properties file from the original cluster. We recommend that you copy the drinstall.properties from an active node, and modify only the necessary parameters.

   – DbAdminLinuxUser=*The Linux user that is created to serve as the Vertica database administrato*r
   Default: dradmin
   – DbAdminLinuxUserHome=*The Vertica Linux database administrator user home directory*
   Default: /export/dradmin

NOTE
This directory is created if the Vertical installer creates the user. Be sure that the directory leading up to the home account already exists on the system. For example, if you are using /export/dradmin, verify that /export exists.

- DbDataDir=*The location of the data directory*
  Default: /data
- DbCatalogDir=*The location of the catalog directory*
  Default: /catalog
- DbHostNames=*The names of the new hosts*
- DbName=*The database name*
  Default: drdata

  NOTE

  This parameter is case-sensitive.
- DbPwd=*The database password*
  Default: dbpass

  NOTE
  The database password that you define here is used during the installation of Data Aggregator.

4. Run the validation script:
   ```
   ./dr_validate.sh -p drinstall.properties
   ```

   The validation script might ask you to reboot.

Add a Host to the Cluster

After you have prepared the new host, add the host to the cluster.

WARNING

Use the same configuration for the new cluster as for the source cluster. For example, the vertica version, node count, database name, administrator, user, catalog directory, and data directory must be the same as the original Data Repository.

Follow these steps:

1. Log in to one of the existing Data Repository hosts.
2. To add the host to the cluster, do one of the following tasks:
   - If you have root access, enter the following command and specify the italic values:
     ```
     /opt/vertica/sbin/update_vertica --add-hosts hostname -u DbAdminLinuxUser
     -l DbAdminLinuxUserHome -L location/resources/vlicense.dat --rpm location/
     resources/vertica-release.rpm -T -S default
     ```

     - **hostname**
       Specify the name of the new host to add to the cluster.
     - **DbAdminLinuxUser**
       The database administrator user name for the cluster as specified in the properties file
     - **DbAdminLinuxUserHome**
       The database administrator user home directory
     - **Location** The location where you extracted installDR.bin file
     - **vertica-release.rpm**

The current RPM file that exists in the extracted installation directory

– If you do not have root access, run the command as the sudo user.

```
export SUDO_USER=root
/opt/vertica/sbin/update_vertica --add-hosts hostname -u DbAdminLinuxUser
-l DbAdminLinuxUserHome -L location/resources/vlicense.dat --rpm location/
resources/vertica-release.rpm -T -S default
```

The script installs Vertica on the new host and incorporates the host into the cluster.

NOTE

If the database administrator user does not already exist, the installation script creates the user. The script prompts you to assign a new password. If the database administrator user exists, but passwordless SSH is not set up, the script prompts for the password to set up.

3. Ensure the Vertica database administrator can write to the data and catalog directories. Run the following commands:

```
chown DbAdminLinuxUser.verticadba DbDataDir DbCatalogDir
chmod 755 DbDataDir DbCatalogDir
```

NOTE

If expanding from one node, segment the database after expansion is complete. For more information, see Segment Database Tables.

Add a Node to the Database

After you add the host to the cluster, add the node to the database. If you are adding multiple nodes to the cluster, prepare all nodes before you add any nodes to the database.

Follow these steps:

1. Log in to any host in the cluster as the Data Repository administrator.
2. Open the Administration Tools:

```
/opt/vertica/bin/adminTools
```

3. Select **Advanced Menu**, and press Enter.
4. Select **Cluster Management**, and press Enter.
5. Select **Add Host(s)**, and press Enter.
6. Select the database, and press Enter.
   The console displays a list of unused hosts.
7. Select the new host, and press Enter.
8. Confirm the selection, and follow the instructions in the console.
   The Data Repository adds the node to the cluster. This process can take a long time.

Create the Buddy Projection

Before you can make an expanded cluster k-safety of 1, you must create the buddy projection for the `migration_status table`.

1. Log into vsql as the `dradmin` user.
2. Run the following command:

```
select export_objects('/tmp/migration_status.sql', 'SCHEMA.migration_status');
```

– ***SCHEMA***
  Specify the schema, for example, `dauser` .

3. Edit the following file:
   `/tmp/migration_status.sql`
4. Remove the `CREATE TABLE` definition and everything above it.
5. Edit the `CREATE PROJECTION` as follows:
   a. Change the name from `migration_status_super` to `migration_status_super_b1`
   b. After `ALL NODES` and before the semi-colon (;) add `OFFSET 1`
   c. Before the `SELECT MARK_DESIGN_SAFE` line, add `SELECT START_REFRESH();`
   d. Change `MARK_DESIGN_KSAFE(0)` to `MARK_DESIGN_KSAFE(1)`
6. Save the `migration_status.sql` file.
7. Run the following command:
   `vsql -U dradmin -w dradminPassword -f /tmp/migration_status.sql`

   A prompt should confirm that the refresh background process has begun.

Update the Data Repository Configuration

On the Data Aggregator host, ensure the `dbconnection.cfg` file points to the correct Data Repository hosts.

Follow these steps:

1. Open the *DA_installation_directory*`/apache-karaf-<`*version*`>/etc/dbconnection.cfg` file on the Data Aggregator host.
2. Modify the following line in the `dbconnection.cfg` file. Modify the line to reference the hostname or IP address of each Data Repository host:
   `dbHostNames=`*dbNode1Hostname*`,`*dbNode2Hostname*`,`*dbNode3Hostname*
3. Save the `dbconnection.cfg` file.
4. Do one of the following steps:
   – Start the Data Aggregator service:
   `service dadaemon start`

   NOTE

   For RHEL 7.x or OL, `service` invokes `systemctl` . You can use `systemctl` instead.
   – (Fault tolerant environment) Run one the following commands to enable the fault tolerant Data Aggregator so that it can start when necessary:
   - **RHEL 6.x:**
     `service dadaemon activate`
   - **RHEL 7.x, SLES, or OL:**
     DA_Install_Directory`/scripts/dadaemon activate`

Migrate the Data Repository

To migrate the Data Repository, install Vertica on the new cluster and migrate the data. The following situations might require migration:

- You are moving to new hosts for a major OS upgrade (for example, RHEL 6.9 to RHEL 7.3).
- The current database hardware no longer meets sizing requirements.
- You are moving from virtual machines to physical hardware for the database.

NOTE

You must upgrade the existing system to the product version you are migrating to before migrating.

For more information about copying the database to another cluster, see the Vertica documentation.

NOTE

Depending on the amount of data, migrating the Data Repository can take a significant amount of time.

TIP

To minimize downtime, perform incremental backups after your initial backup.

Prepare to Install the Destination Data Repository

Before you install Vertica on the new cluster, prepare the environment for the installation. For more information about preparing the destination cluster, see Prepare to Install the Data Repository.

The target cluster must have the following things:

- The same number of nodes the source cluster
- A database with the same name as the source database

  NOTE

  The target database can be empty.
- The same node names as the source cluster

  NOTE

  The nodes names that are listed in the NODES system tables on both clusters must match. To change node names post install on the new system to match the existing one, contact Support.
- Be accessible from the source cluster
- The same database administrator account and all nodes must allow a database administrator of the source cluster to log in through SSH without a password

  NOTE

  Passwordless access within the cluster is not the same as passwordless access between clusters. The SSH ID of the administrator account on the source cluster and the target cluster are likely not the same. You must configure each host in the target cluster to accept the SSH authentication of the source cluster.
- Adequate disk space for the `vbr --task copycluster` command to complete.

Install Vertica on the New Cluster

The installation process is the same as a normal Data Repository installation, for more information, see Install the Data Repository.

WARNING

Use the same configuration for the new cluster as for the source cluster. For example, the vertica version, node count, database name, administrator, user, catalog directory, and data directory must be the same as the original Data Repository.

Migrate the Cluster

After you install the database on the new cluster, migrate the existing data. The migration uses the copy cluster command, which simultaneously backs up the existing database and restores the data to the new cluster. The copy cluster is configured and started and run on the source system to point to the new system. You can run copy cluster on a single node on the source system and it copies over to all nodes on the new system. Copy cluster copies all data in the Data

Repository from before you run the command. Because DX NetOps Performance Management continues to collect data during the migration, the process requires multiple runs of the copy cluster command.

Create a Configuration File for Copy Cluster

Before you can configure the target cluster, you need to know the exact names that `admintools` supplied to all nodes in the source database.

To see the node names, run a query similar to following example:

```
 VMart=> select node_name from nodes;
node_name
------------------
v_vmart_node0001
v_vmart_node0002
v_vmart_node0003
(3 rows)
```

The copy cluster command requires a configuration file that includes the necessary information. You can create the file in any desired location on the source system and give it any name with a `.ini` extension. In the configuration file, specify the host names of nodes in the target cluster as the backup hosts. Specify the nodes in a `[Mapping]` section as shown in the following example. Specify the full node name, not the short name (for example, `v_vmart_node0001` , not `node0001`).

Example:

The following example configuration file is set up to copy a database on a three node cluster (`v_vmart_node0001` , `v_vmart_node0002` , and `v_vmart_node0003`) to another cluster consisting of nodes: test-host01, test-host02, and test-host03:

> **NOTE**
>
> The dbName parameter is case-sensitive.

```
 [Misc]
snapshotName = CopyVmart
restorePointLimit = 5
 objectRestoreMode = createOrReplace
tempDir = /tmp/vbr
retryCount = 5
retryDelay = 1

[Database]
dbName = vmart
dbUser = dradmin
dbPassword =  password
dbPromptForPassword = False

[Transmission]
encrypt = False
checksum = False
port_rsync = 50000
```

```
[Mapping]
; backupDir is not used for cluster copy
v_vmart_node0001= test-host01
v_vmart_node0002= test-host02
v_vmart_node0003= test-host03
```

NOTE

Ensure the `/tmp/vbr` directory has read and write permissions.

Stop the Target Database

Before you start the migration, shut down the database on the target cluster.

Follow these steps:

1. Log in to the target database cluster as the database admin user.
2. Open Vertica admin Tools:
   ```
   /opt/vertica/bin/adminTools
   ```
3. Select (4) Stop Database. Wait for the shutdown to complete before you run copy cluster.

Copy Historical Data

After you install the database on the new cluster, copy the data from the existing database. The copy cluster command copies all information from before you initiate the command. New data continues to come in while the command is running, but the command does not copy this data. The target cluster must be stopped before you invoke copy cluster.

Follow these steps:

1. Log in to the source cluster with the database administrator account.
2. Ensure passwordless SSH is enabled for the database administrator account. In a cluster installation, ensure that passwordless ssh keys are set up for *each* host that is participating in the cluster.
   ```
   ssh hostname ls
   ```
   hostname is the name of the host where Data Repository is installed.
   If the passwordless ssh key is set up, you are *not* prompted for a password. If youare prompted for a password, set up passwordless ssh:
   a. Press Ctrl+C.
   b. Set up the Linux user account for the database administrator user with a passwordless ssh key:
      ```
      ssh-keygen -N "" -t rsa -f ~/.ssh/id_rsa
      cp ~/.ssh/id_rsa.pub ~/.ssh/authorized_keys2
      chmod 644 ~/.ssh/authorized_keys2
      ```
   c. Copy the SSH credentials to the other hosts in the cluster:
      ```
      ssh-copy-id -i dradmin@other-hostname-in-the-cluster
      ```
   d. Confirm that you are *not* prompted for a password:
      ```
      ssh hostname ls
      ```
3. Run the copy cluster command:

```
vbr.py --task copycluster --config-file CopyClusterConfigurationFile.ini
```

The command copies the historical data for the database and displays the following message:

```
 > vbr.py --config-file CopyVmart.ini --task copycluster
Preparing...
Copying...
 1871652633 out of 1871652633, 100%
 All child processes terminated successfully.
 copycluster done!
```

NOTE

If the `copycluster` command fails, ensure passwordless SSH is enabled for the database administrator account.

Verify the Copy of the Historical Data

After the copy cluster process completes, ensure the integrity of your data.

Follow these steps:

1. Log in to the target database cluster as the database admin user.
2. Open Vertica adminTools:
   ```
   /opt/vertica/bin/adminTools
   ```
3. Start the database.
4. From any node in the cluster, open open the Vertica SQL prompt:
   ```
   /opt/vertica/bin/vsql -U dauser
   ```
5. Run the following queries to verify the timestamp of these key database tables:
   ```
    SELECT to_timestamp(max(tstamp)) from dauser.reach_rate;
   SELECT to_timestamp(max(tstamp)) from dauser.ifstats_rate;
   ```

   The date and time must correspond to the time when you started the copy.
6. Open adminTools, and stop the database.

Stop the Data Aggregator

To maintain integrity of your data, stop the Data Aggregator before the final copy of the Data Repository. Polling continues on Data Collector if it is running and polling when Data Aggregator is stopped. Data Collector queues polled data for future delivery to Data Aggregator.

Follow these steps:

1. Log in to the Data Aggregator host as the root user or a sudo user.

   NOTE
   If you installed Data Aggregator as the sudo user, you set up a sudo command alias for the service dadaemon command. Use the sudo commands.
2. Use firewall rules to block traffic from all the Data Collectors. Run the following command for each Data Collector:
   ```
   iptables -A INPUT -s DC_IP -j DROP
   ```

DC_IP specifies the IP of the Data Collector.

3. Do one of the following steps:
   – Stop the Data Aggregator service:
     ```
     service dadaemon stop
     ```

     NOTE

     For RHEL 7.x or OL, `service` invokes `systemctl` . You can use `systemctl` instead.
   – (Fault tolerant environment) If the local Data Aggregator is running, run one the following commands to shut it down and prevent it from restarting until maintenance is complete:
     - **RHEL 6.x:**
       ```
       service dadaemon maintenance
       ```
     - **RHEL 7.x, SLES, or OL:**
       ```
       DA_Install_Directory/scripts/dadaemon maintenance
       ```

   The Data Aggregator stops.

Copy Recent Data

After you stop the Data Aggregator, run the copy cluster command again to copy recent data. The Data Repository copies only new data that arrive after the initial copy.

Follow these steps:

1. Log in to the source cluster with the database administrator account.
2. Run the copy cluster command:
   ```
   vbr.py --task copycluster --config-file CopyConfigurationFile.ini
   ```

   The command copies the recent data.

Verify the Copy of the Recent Data

To ensure the integrity of your data, verify the data.

Follow these steps:

1. Log in to the target database cluster as the database admin user.
2. Open Vertica admin Tools:
   ```
   /opt/vertica/bin/adminTools
   ```
3. Start the database.
4. From any node in the cluster, open the Vertica SQL prompt:
   ```
   /opt/vertica/bin/vsql -U dauser
   ```
5. Run the following queries to verify the timestamp of these key database tables:
   ```
   SELECT to_timestamp(max(tstamp)) from dauser.reach_rate;
   SELECT to_timestamp(max(tstamp)) from dauser.ifstats_rate;
   ```

   The date and time must correspond to the time when you started the copy.

Update the Database Connection Information

If you are not migrating the Data Aggregator, to enable communication between the Data Aggregator and the new Data Repository cluster, update the database connection information.

Follow these steps:

1. Log in to the Data Aggregator host.
2. Open the following file:
   ```
   vi /opt/IMDataAggregator/apache-karaf-version/etc/dbconnection.cfg
   ```
3. Update the following parameter with the hostnames of the new Data Repository cluster:
   ```
   dbHostNames=hostname1,hostname2,hostname3
   ```

Restart Data Aggregator

After the migration is complete, restart the Data Aggregator.

Follow these steps:

1. Log in to the Data Aggregator host as the root user or a sudo user.

   NOTE
   If you installed Data Aggregator as the sudo user, you set up a sudo command alias for the service dadaemon command. Use the sudo commands.
2. Do one of the following steps:
   - Start the Data Aggregator service:
     ```
     service dadaemon start
     ```
   - (Fault tolerant environment) Run one the following commands to enable the fault tolerant Data Aggregator so that it can start when necessary:
     - **RHEL 6.x:**
       ```
       service dadaemon activate
       ```
     - **RHEL 7.x, SLES, or OL:**
       ```
       DA_Install_Directory/scripts/dadaemon activate
       ```

   Data Aggregator starts and synchronizes with CA NetOps Portal and the Data Repository. When the iptables entries are removed, any queued, polled data on the Data Collectors is sent to the Data Aggregator. The oldest data is discarded if the queued data exceeds a disk space limit that is configured on the Data Collector system. As a result, there is a gap in the polled reporting data.
3. Monitor the Data Aggregator restart process:
   a. Log in to the Data Aggregator host and navigate to the following directory:
      ```
      /opt/IMDataAggregator/performance-spool
      ```
   b. Wait for few minutes after starting the Data Aggregator service. Verify that no DTO files exist with a size greater than zero.
   c. Enable traffic from the SNMP Data Collector with largest number of polled items:
      ```
      iptables -D INPUT -s DC_IP -j DROP
      ```

      Data Aggregator starts schema validation and processing of cached and new polled data from this Data Collector.

d. After the Data Aggregator system utilization decreases, enable traffic from the remaining SNMP Data Collectors.
e. After the Data Aggregator system utilization decreases, enable traffic from the CAMM Data Collectors.

Verify the Migration

After the Data Aggregator startup is complete, log in to NetOps Portal, and verify the following indicators:

- The system status is good and the Data Aggregator data source is available.
- Verify the Last Polled on data and time.
- Navigate to the Data Collector List and verify that all Data Collectors are up and collecting data.
- Open the Infrastructure Overview dashboard, and verify that data is available for the following time ranges:
  - Last hour
  - Last 7 days

Data Repository Heartbeat Monitor Process

The heartbeat monitor process checks whether Data Repository is running every 10 seconds. If the heartbeat process fails to confirm that the database is running after 5 minutes, Data Aggregator shuts down. An audit message is logged in the *DA_installation_directory*/apache-karaf-<*vers*>/shutdown.log file.

In a cluster environment, all nodes in the cluster are continuously checked for availability every 10 seconds. If a node cannot be contacted within 5 minutes, DX NetOps Performance Management generates and logs an event. An audit message is logged in the *DA_installation_directory*/apache-karaf-<*vers*>/shutdown.log file.

If the Data Repository node that failed is the primary node, Data Aggregator automatically switches to the next available node. DX NetOps Performance Management generates and logs an event.

> **WARNING**
> Certain administrative functions that are occurring during a high availability failover are interrupted and then fail. One poll cycle is lost. These functions will not resume after Data Repository connects to another node in the cluster environment. Administrative functions that you perform after Data Repository connects to another node in the cluster environment work as designed.

If more than one Data Repository node fails, the Data Aggregator shuts down.

Data Aggregator shuts down automatically if it fails to connect to Data Repository on start-up.

Data Repository Audit Process

The license agreement states that the total data stored in Data Repository cannot exceed 64 TB. The audit process audits the database daily at 3:00 AM to calculate the total space that Data Aggregator data occupies.

To view the most recent result of an audit, access the following URL in your browser:
`http://da_host:8581/rest/datarepositorymaintenance/audit`

This URL returns XML. The "Current Size" tag displays the current size of Data Repository in bytes.

> **WARNING**
> Review the audit results periodically. If you see a value greater than 64 TB, you are not in compliance with the license agreement. Contact CA Technical Support for further instructions.

Run Data Repository Diagnostic Utilities

Vertica provides a set of utilities that test the performance of your hardware for the Data Repository. To verify that the environment is ideal for the database, run these tests before you upgrade the Data Repository.

NOTE

If you have already installed the Data Repository, you can perform these tests at any time to verify performance. The utilities are available on each node in `/opt/vertica/bin` .

If the tests do not meet the recommendations, fix the issues before you continue the upgrade.

vcpuperf

This utility measures the CPU processing speed of the host and compares the speed against benchmarks for common server CPUs. The utility measures how long the server requires to complete the test, and determines whether CPU throttling is enabled.

Follow these steps:

1. Execute the following command on *each* Data Repository node:

```
./vcpuperf > /tmp/vcpuperf.out
```

2. Verify that the performance meets the following requirements:
   - The CPU time is consistent with the benchmark values in the output.
   - The low load time and high load time are within 10 microseconds. If the difference is grater than 50 microseconds, CPU throttling might be enabled on your system. Disable CPU throttling.

Example:

The following example shows the return from this utility:

```
$ /opt/vertica/bin/vcpuperf

Compiled with: 4.1.2 20080704 (Red Hat 4.1.2-52)

Expected time on Core 2, 2.53GHz: ~9.5s

Expected time on Nehalem, 2.67GHz: ~9.0s

Expected time on Xeon 5670, 2.93GHz: ~8.0s




This machine's time:

CPU Time: 7.740000s


Real Time:7.740000s
```

```
Some machines automatically throttle the CPU to save power.

This test can be done in <100 microseconds (60-70 on Xeon 5670, 2.93GHz).

Low load times much larger than 100-200us or much larger than the corresponding high
 load time

indicate low-load throttling, which can adversely affect small query / concurrent
 performance.


This machine's high load time: 67 microseconds.



This machine's low load time: 64 microseconds.
```

This test was performed on a system with 2.67-GHz processors, so the real time is acceptable. The difference between the high load time and low load time is within the expected tolerance.

For more information about this utility, see the Vertica documentation.

vioperf

This utility tests the performance of the disk input and output (I/O). The utility performs a series of reads and writes.

> **NOTE**
>
> To measure the read/write speeds when using the same SAN/NAS for the disk or VM disk, you must run vioperf on all nodes of the cluster at the same time for the data or catalog directory.

Follow these steps:

1. Execute the following commands on *each* Data Repository node:

   ```
   ./vioperf /data > /tmp/vioperf.out --duration=60sec
   ```

   /data is the full path of the data directory.

   ```
   ./vioperf /catalog > /tmp/vioperf.out --duration=60sec
   ```

   /catalog is the full path of the catalog directory.
2. Verify the Write and Read counter values at least 40 MB/s per core.
   The recommended I/O is 40 MB/s per physical core on each node. For example, the recommended I/O rate for a node with 2 hyper-threaded six-core CPUs (12 physical cores) is 480 MB/s.

   > **WARNING**
   >
   > If the **thread count** column shows a value of 1, the utility cannot determine the number of cores. Add the following arguement to the command to run the utility:

`--thread-count=`*`CORES`*

Cores defines the number of cores in the system as a fixed integer.

Example:

The following example shows the return from this utility for the data directory:

```
The minimum required I/O is 20 MB/s read and write per physical processor core on each
 node, in full duplex i.e. reading and writing at this rate simultaneously, concurrently
 on all nodes of the cluster. The recommended I/O is 40 MB/s per physical core on each
 node. For example, the I/O rate for a server node with 2 hyper-threaded six-core CPUs
 is 240 MB/s required minimum, 480 MB/s recommended.




Using direct io (buffer size=1048576, alignment=512) for directory "/drdata"




test      | directory                                    | counter name        |
 counter value       | counter value (10 sec avg)      | counter value/core  | counter
 value/core (10 sec avg) | thread count  | %CPU  | %IO Wait  | elapsed time (s)|
 remaining time (s)

-------------------------------------------------------------------------------------------------


Write     | /drdata                                      | MB/s
  | 873                | 873                            | 54.5625            |
 54.5625                         | 16            | 29    | 40        | 10              |
 5


Write     | /drdata                                      | MB/s
  | 868                | 865                            | 54.25              |
 54.0625                         | 16            | 28    | 30        | 15              |
 0

ReWrite   | /drdata                                      | (MB-read+MB-write)/
s| 275+275             | 275+275                        | 17.1875+17.1875     |
 17.1875+17.1875                 | 16            | 13    | 21        | 10              |
 5
```

```
ReWrite   | /drdata                                    | (MB-read+MB-write)/
s| 242+242            | 178+178                       | 15.125+15.125       |
 11.125+11.125                  | 16           | 7     | 17        | 15             |
 0

Read      | /drdata                                    | MB/s
  | 735                | 735                           | 45.9375             |
 45.9375                        | 16           | 11    | 23        | 10             |
 5

Read      | /drdata                                    | MB/s
  | 786                | 786                           | 49.125              |
 49.125                         | 16           | 26    | 25        | 15             |
 0

SkipRead  | /drdata                                    | seeks/s
  | 4511               | 4511                          | 281.938             |
 281.938                        | 16           | 14    | 19        | 10             |
 5

SkipRead  | /drdata                                    | seeks/s
  | 4477               | 4407                          | 279.812             |
 275.438                        | 16           | 3     | 15        | 15             |
 0
```

This server has 16 cores. The Read and Write counter values indicate the I/O is greater than 40 MB/s per core.

For more information about this utility, see the Vertica documentation.

vnetperf

This utility tests the network performance of the Data Repository hosts. The utility measures network latency and the throughput for the TCP and UDP protocols.

> **WARNING**
>
> This utility causes a high network load and affects database performance. Do not run this utility while the database is running.

Follow these steps:

1. Log in as a user that has passwordless ssh between the nodes.
2. Execute the following command on *one* of the Data Repository nodes:

   ```
   ./vnetperf --hosts DAhost,DRhost1,DRhost2,DRhost3 > /tmp/vnetperf.out
   ```

   Specify the hostname or IP address of the Data Aggregator host and each Data Repository host.
3. Verify that the network performance meets the following requirements:

- Round-trip time (RTT) latency of 200 microseconds or less
- Clock skew under 1 second
- Throughput of 800 MB/s or more
  The utility runs a series of throttled tests. Verify the throughput for the highest speed test.

For more information about this utility, see the Vertica documentation.

Segment Database Tables

If you received a table segmentation warning during the Data Aggregator upgrade, segment the database tables on the Data Repository. Table segmentation is a one-time task that is required for systems where the original installation was a release earlier than 2.3.3. Segmenting the tables reduces the amount of disk space that is required for the database. Segmenting the tables also improves general query performance.

TIP

If you are unsure whether your database requires segmentation, download the script and attempt to identify tables that require segmentation. If segmentation is already complete or not required, the script does not return any tables.

You can segment the database tables when the Data Aggregator and the Data Collectors are up or down.

WARNING

Segmentation is a resource-intensive process. We *strongly* recommend that you segment the database tables when the Data Aggregator and the Data Collectors are down.

Segmentation can take several hours to segment large tables in the database. During tests, migration of tables larger than 100 GB took over 10 hours. The segmentation time is not uniform to table size. Time depends on many factors including row count, column count, compression of the data, and system specifications.

No active monitoring of your infrastructure environment occurs when the Data Aggregator and Data Collectors are down.

If you segment the database tables while the Data Aggregator is up, the following restrictions apply:

- Do not perform any Data Aggregator administrative functions:
  - Modifying monitoring profiles
  - Associating collections to monitoring profiles
  - Increasing poll rates
  - Running new discoveries
- Minimize the report load.

Use the following process to segment the database tables:

Identify Tables That Require Segmentation

Before you start table segmentation, identify the tables that require segmentation.

Follow these steps:

1. Download the segment.py script from CA Support:
   - https://https://support.broadcom.com/enterprise-software
   - https://casupport.broadcom.com/phpdocs/7/8568/segment.py
   This procedure assumes that the segment.py script is in the home directory of the Vertica Linux database administrator user.
2. Log in to any host in the Data Repository cluster as the Vertica Linux database administrator user.
3. Run the script:

```
./segment.py --task tables --pass database_admin_user_password  [--
name database_name] [--port database_port]
```

- **database_admin_user_password**
  The Vertica Linux database administrator user password
- **database_name**
  The name of the database
  Defulat: drdata

  NOTE

  This parameter is case-sensitive.
- **database_port**
  The Vertica port
  Default: 5433

For example:

```
./segment.py --task tables --pass password --name mydatabase
```

Any currently unsegmented table projections, which are sorted from largest to smallest, are returned. If any tables require segmentation, continue this process.

Back up the Data Repository

Before you segment the tables, back up the Data Repository.

WARNING

After segmentation, the disk space for the backup increases by the amount of data in the new segmented table projections. Verify that the backup system has enough disk space available after segmentation is completed and before backups run.

The data for the old unsegmented table projections is removed from the backup data one day after the time of the restorePointLimit. To remove this data immediately, change the snapshot name in the backup configuration file, and do a full backup. You can then archive the older backup, and delete the backup from the backup disk. Use the presegmentation backup only if you cannot use the backup that was created after segmentation completed. If you use the presegmentation backup, the tables will require segmentation again.

For information, see Back Up the Data Repository.

Segment Tables with No Data

Tables with no date are segmented quickly and segmentation does not negatively affect performance. You can segment these tables without stopping the Data Aggregator.

Follow these steps:

1. Log in to any host in the Data Repository cluster as the Vertica Linux database administrator user.
2. Run the script:

```
./segment.py --task segment --zerotables --pass database_admin_user_password [--
name database_name] [--port database_port]
```

- **database_admin_user_password**
  The Vertica Linux database administrator user password
- **database_name**
  The name of the database

Default: drdata

NOTE

This parameter is case-sensitive.

– **database_port**
The Vertica port
Default: 5433

The script segments the database tables with no data.

Determine the Table Segmentation Time

To determine whether to segment the tables while the Data Aggregator is up or down, calculate the necessary time for segmentation.

Follow these steps:

1. Get a list of the tables that require segmentation:
```
./segment.py --task tables --pass database_admin_user_password [--
name database_name] [--port database_port]
```

NOTE

The `database_name` parameter is case-sensitive.

The list sorts the tables from largest to smallest.
2. Disable scheduled Data Repository backups until segmentation is complete. Backups can interfere with the segmentation process.
3. Select a table from step 1 that is about 5 GB in size, and segment that table:
```
./segment.py --task segment --table rate_table_name --
pass database_admin_user_password [--name database_name] [--port database_port]
```

NOTE

The `database_name` parameter is case-sensitive.

NOTE
You can run this command when Data Aggregator is running, but we recommend that you run the command during a 2-3 hour maintenance window.
4. Reenable the scheduled Data Repository backups.
5. Use the segmentation time for the 5-GB table to determine how long segmentation might take to segment all of the tables that are less than 100 GB.

NOTE
The actual time segmentation time for the database tables can vary based on the type and compression of the data in the tables. The values that are calculated here are rough estimates. When planning a scheduled maintenance window, add an extra hour of time for every 10 GB to 15 GB of database tables. For large databases, you might not be able to schedule a single maintenance window that is long enough to segment the entire database. In this case, you can segment the database tables over multiple maintenance windows.

Segment the Remaining Database Tables

Segment the remaining tables.

WARNING
When segmenting the tables in the database, if Data Aggregator is running, at least 40 percent of the available disk space must remain free for query processing and other database activities. When the Data Aggregator is not running, the total disk utilization during segmentation must not exceed 90 percent of available disk space. Tables that would cause the disk utilization to exceed these limits during segmentation are not segmented.

Follow these steps:

1. As the Vertica Linux database administrator user, log in to one of the computers in the cluster where Data Repository is installed.
2. During the table projection segmentation validation in the previous procedure, if more than ten zero-length table projections were seen during this verification, type the following command to segment them:
   ```
   ./segment.py --task segment --pass database_admin_user_password --zerotables [--name database_name] [--port database_port]
   ```
   - **database_admin_user_password**
     The Vertica Linux database administrator user password
   - **database_name**
     The name of the database
     Default: drdata

     NOTE

     This parameter is case-sensitive.
   - **database_port**
     The Vertica port
     Default: 5433

   For example:
   ```
   ./segment.py --task segment --pass password --zerotables --name mydatabase --port 1122
   ```
3. If there are table projections that are greater than 100 GB in size, type the following command to create a script to segment the table projections that are *less than* 100 GB first:
   ```
   ./segment.py --task script --pass database_admin_user_password --lt100G [--name database_name] [--port database_port]
   ```
   - **database_admin_user_password**
     The Vertica Linux database administrator user password
   - **database_name**
     The name of the database
     Default: drdata

     NOTE

     This parameter is case-sensitive.
   - **database_port**
     The Vertica port
     Default: 5433

   For example:
   ```
   ./segment.py --task script --pass password --lt100G --name mydatabase --port 1122
   ```
4. Disable scheduled backups until segmentation is complete. Backups can interfere with the segmentation process.
5. To execute the segment-script.sh script, type the following command:

```
nohup ./segment-script.sh
```

The script segments all unsegmented table projections that are less than 100 GB and sorts them from smallest to largest. The output is sent to nohup.out. If the shell is closed accidentally, the script continues to run.
Depending on your maintenance window size and the combined size of all of the tables under 100 GB, determine which tables can be segmented in the maintenance window. Modify the generated script by removing the tables that do not fit inside the maintenance window. Run the generated segment-script.sh during the maintenance window. If all of the tables under 100 GB could not be segmented in the maintenance window, regenerate the script, and run the segment-script.sh during the next maintenance window until all of the tables have been segmented.

WARNING

When you run the script, any tables that cause disk utilization to exceed 90 percent displays an error message. These tables are not segmented. To segment these tables, more available disk space is needed. You are prompted for each table that can cause disk utilization to exceed 60 percent. We strongly recommend that Data Aggregator be brought down before segmenting these tables.

This script can take several hours to execute. Do not interrupt the script execution once it begins to avoid corruption of the database.

6. Reenable scheduled backups only if more segmentation is needed and will be done in a future maintenance window.
7. Generate a script, segment-script.sh, that segments remaining table projections that are over 100 GB:

```
./segment.py --task script --pass database_admin_user_password [--name
database_name] [--port database_port]
```

   - **database_admin_user_password**
     The Vertica Linux database administrator user password
   - **database_name**
     The name of the database
     Default: drdata

     NOTE

     This parameter is case-sensitive.
   - **database_port**
     The Vertica port
     Default: 5433

   For example:

```
./segment.py --task script --pass password --name mydatabase --port 1122
```

   WARNING
   When the script is generated, any tables that might cause disk utilization to exceed 60 percent and 90 percent are indicated.
8. Disable scheduled backups, if they are not already disabled.
9. To execute the segment-script.sh script, type the following command:

```
nohup ./segment-script.sh
```

   The script segments all unsegmented tables and sorts them from smallest to largest.
10. Verify that all tables are now segmented:

```
./segment.py --task tables --pass database_admin_user_password [--
name database_name] [--port database_port]
```

NOTE

The `database_name` parameter is case-sensitive.

The following message appears:

```
No tables found with unsegmented projections.
```

11. Reenable scheduled backups.
12. If you segmented the database tables when Data Aggregator and Data Collector were down, start these components.

Move the Data Repository Data Directory

If necessary, you can move the Data Repository data directory from an existing location to another location on the same Vertica cluster.

This process involves the following steps:

1. Create the new location.
2. Move the data.
3. Drop the old location.

For more information, see the Vertica documentation.

You might need to move the Data Repository data directory for the following reasons:

- You want to add new storage to an existing server.
- You want to move the database from an old mount point to a new mount point because of an updated server build template.

The following process causes some downtime for DX NetOps Performance Management while the move occurs.

The following factors impact the total amount of downtime:

- Size of the database
- Storage input and output speed

The following process covers only moving the Data Repository data directory. For information about moving the Data Repository database from one server to another, see Disaster Recovery.

Example:

- You are moving the data directory for a three-node cluster:
  - node0001
  - node0002
  - node0003
- The current data directory is: `/spare/dbdata/data`
- The new data directory is: `/opt/application/CA/drdata`
- The Vertica database name is `drdata` .
- The `dbadmin` user for Veritca is `dradmin` .
- The following end goals apply to this scenario:

```
 /spare/dbdata/data/drdata/v_drdata_node0001_data >> /opt/application/CA/drdata/
drdata/v_drdata_node0001_data
 /spare/dbdata/data/drdata/v_drdata_node0002_data >> /opt/application/CA/drdata/
drdata/v_drdata_node0002_data
```

```
 /spare/dbdata/data/drdata/v_drdata_node0003_data >> /opt/application/CA/drdata/
drdata/v_drdata_node0003_data
```

Follow these steps:

1. On each node, create the new data directory:
   ```
   mkdir -p /new_data_directory/dbname/v_dbname_node000x_data
   ```
   - ***new_data_directory***
     Specify the new data directory.
   - ***dbname***
     Specify the name of the database.
   - ***x***
     Specify the existing node number.
2. Grant permissions to the new data directory:
   ```
   chown -R dradmin:verticadba /new_data_directory/dbname
   ```
3. For each parent in the path to /*new_data_directory*/dbname, grant permissions:
   ```
   chown dradmin:verticadba /parent
   ```
4. As `dradmin` , create the new storage location for each node in Vertica:
   ```
   create location '/new_data_directory/dbname/v_dbname_node000x_data' NODE
   'v_dbname_node000x' USAGE 'DATA,TEMP' LABEL 'TO_DATA_TEMP';
   ```
   - ***TO_DATA_TEMP***
     Specify a label for the new storage location. This label is required when you set the object policy.
5. Confirm the new storage locations:
   ```
    select * from storage_locations;
   ```
6. Set the object policy:
   ```
   select set_object_storage_policy('schema_name', 'TO_DATA_TEMP', true );
   ```
   - ***schema_name***
     Specify the name for the schema.
     Example: `dauser`
     To view a lists of the available schemas, run the following command:
     ```
     dradmin=> \dn
     ```
   - ***TO_DATA_TEMP***
     Specify the same label used when you created the new storage location.
7. Trigger the object to move:
   ```
   select enforce_object_storage_policy('schema_name');
   ```
8. Confirm that the policy is applied:
   ```
   select * from storage_policies;
   ```
9. Retire the old storage locations for each node:

```
select retire_location('/existing_data_directory/dbname/v_dbname_node000x_data',
'v_dbname_node000x', true);
```

- ***existing_data_directory***
  Specify the existing data location.

10. Confirm that the old storage locations are retired:
```
select is_retired, location_path from storage_locations;
```

11. Drop the old storage location for each node:
```
select drop_location('/existing_data_directory/dbname/v_dbname_node000x_data',
'v_dbname_node000x');
```

12. Clear the storage policy:
```
select clear_object_storage_policy( 'schema_name' );
```

13. Remove the label from the new storage location for each node:
```
select alter_location_label('/new_data_directory/dbname/v_dbname_node000x_data',
'v_dbname_node000x', '');
```

Flow Administration

You can configure flow monitoring from Network Flow Analysis in the NetOps Portal user interface. Configure the overall Application settings and the Watchdog settings. Watchdog Services let you monitor Network Flow Analysis components. The Watchdog Services poll each server in your Network Flow Analysis configuration once every two hours to determine the status of all components. You can establish thresholds, an email address for receiving messages, and other settings for the Watchdog Services to ensure that you are notified of issues with the components as soon as possible. You can also view, add, delete, and edit details about the Harvesters that supply flows to DX NetOps.

> **NOTE**
> Any Application settings or Watchdog settings left blank default to the values fetched from Network Flow Analysis.

You can also enable and disable network flow processing for your Network Flow Analysis interfaces. You can also delete interfaces. For more information, see Manage Network Flow Processing.

Configure Application Settings

You can configure a wide range of settings on the **Application Settings** pane.

Follow these steps:

1. Hover over **Administration**, and click **Data Sources: the Network Flow Analysis datasource**.
   The page for the data source opens.
2. Select **Application Settings** from the menu on the left side of the page.
   The **Application Settings** pane opens.
3. (*Optional*) Change any of the following settings as needed, then click **Save**.
   - **Interface Data Absence Limit**
     Specifies how long the program waits for an update before it flags an interface as missing.
     Default: 4 Hours
   - **TCP Rebase Port**

Specifies the target port for TCP traffic that is redirected by an application mapping rule. TCP traffic that you do not want to go to a target port goes to the TCP Rebase Port instead. Other settings that affect application mapping behavior are **UDP Rebase Port**, **ToS Mask**, and **Preserve ToS Map Proto**.
Default: 9000

- **ToS Mask**
  Specifies the number of bits that application mapping rules use for matching ToS values. The default value of 255 sets the program to look for matches throughout all ToS values. Other settings that affect application mapping behavior are **TCP Rebase Port**, **UDP Rebase Port**, and **Preserve ToS Map Proto**.
  Default: 255
- **UDP Rebase Port**
  Specifies the target port for UDP traffic that is redirected by an application mapping rule. UDP traffic that you do not want to go to a target port goes to the UDP Rebase Port instead. Other settings that affect application mapping behavior are **TCP Rebase Port**, **ToS Mask**, and **Preserve ToS Map Proto**.
  Default: 8000
- **Auto-Enable Interfaces**
  Specifies whether newly discovered interfaces are enabled automatically (True) or are disabled (False). If you want to control which interfaces are reported and consume licenses, set the value to False. This setting affects the **Enabled** status for new interfaces. Interfaces that have already been discovered are not affected by changes to this setting.
  Default: True
- **Default Time Zone**
  Specifies the time zone used for custom reports and analysis.
  Default: GMT
- **DNS Domains**
  Removes the specified suffixes from host names in NFA console views and reports. If you include `.my_company.com` , for example, this suffix is not shown in the host names that appear in any views or reports. To specify multiple entries, separate the entries with commas and without intervening spaces.
  Default: <no default>
- **Show Trendline Zeroes**
  Specifies whether trendlines drop to zero when there are gaps in data.
  Default: False
- **From Address**
  Specifies the email address of the Administrator, which is used as the **From** value when reports are emailed. If this setting is not configured properly, users cannot send scheduled or on-the-fly reports. These functions also require a properly configured **SMTP Server** value.
  Default: <no default>
- **SMTP Server**
  Specifies the IP address of the SMTP mail server that is used for emailing reports. If this setting is not configured properly, users cannot send scheduled or on-the-fly reports. These functions also require a properly configured **From Address** value.
  Default: <no default>
- **Licensed Devices**
  Records the total number of licenses that you purchased. This value is used to calculate the percentage of licenses in use. The **License Utilization** percentage is accurate only if the **Licensed Devices** value is accurate.
  Default: 50
- **Preserve ToS Map Proto**
  Specifies whether protocol traffic for ToS-based application-mapped data is combined (N) or is shown as separate data streams that are labeled with the original protocol designators (Y).
  For example, suppose the value is Y and you map TCP, UDP, and some other IP protocol traffic to one port. To continue the example, suppose you drill in to a link in the **Enterprise Overview** view in the NFA console for a host that has the mapped data. In this case, the **Stacked Protocol Trend** and **Protocol Trend** views show and label the protocol traffic separately, whether the protocol traffic is for TCP, UDP, or some other IP protocol.

- **Stacked Protocol Trend** and **Protocol Trend** views show protocol traffic that meets the following conditions: (1) The traffic passes the minimum threshold and (2) The protocol volume is high enough to place it in the Top N group.

If the **Preserve ToS Map Proto** value is N and the **Stacked Protocol Trend** views show related protocol traffic, all of the protocols for the mapped traffic are combined in a single traffic stream that has a TCP label.
Other settings that affect application mapping are **TCP Rebase Port**, **UDP Rebase Port**, and **ToS Mask**.
Default: Y

– **Pump Broadcast/Multicast**
Specifies whether interface views and reports include (True) or hide (False) broadcast/multicast traffic.
Default: True

– **Reporter IP**
Specifies the IP address of the NFA console.
Default:
- Stand-alone deployment: Loopback IP address of the stand-alone server
- Distributed 2-tier deployment: Loopback IP address of the NFA console

– **Report Service Polling Delay**
Specifies the number of seconds between checks to see if reports have finished running.
Default: 15

– **Router Domains**
Removes the specified suffixes from the flow router names. To specify multiple entries, separate the entries with commas and without intervening spaces.
Default: <no default>

– **Show Aggregations**
Specifies whether to include interface aggregations in the **Enterprise Overview**. If the value is True, interface aggregations are included in the views. To be included in the **Enterprise Overview**, the aggregations must have enough traffic to pass the minimum thresholds and to rank in the Top N group.
Default: False

– **Show Device Name**
Specifies whether the interface name format starts with the device name (True) or omits it (False).
Default: True

– **Display Notes Field**
Displays (True) or hides (False) the **Notes** icon for interfaces. If the **Notes** icon is visible, you can click it to add, edit, or view additional information about an interface.
Default: False

– **Trap Destination**
IP address or DNS name of the target server for sending the traps that are shown as events in the NetOps Portal. Traps can be displayed as events only when this setting is configured correctly. Set the **Trap Destination** value to match the IP address of one of the NFA console or stand-alone server that is registered as a data source for NetOps Portal
Default: IP address of the NFA console (distributed deployment) or stand-alone server (stand-alone deployment)

– **Traps Use Unique Domains**
Specifies whether all traps on a single domain system use the single domain.
Default: False

Edit Watchdog Service Settings

Edit the Watchdog settings to change configuration values such as thresholds, trap settings, polling settings, notification address, and community strings.

Follow these steps:

1. Display the **Watchdog Settings** pane:
   a. Hover over **Administration**, and click **Data Sources: the Network Flow Analysis datasource**.
      The page for the data source opens.
   b. Select **Watchdog Settings** from the menu on the left side of the page.
      The **Watchdog Settings** pane opens and displays the current settings.
2. Edit the **Watchdog Service** settings:
   - **Memory Threshold**
     Threshold for memory utilization. You are notified by email when the memory threshold on a server is exceeded, provided that the address and string are set.
     Default: 80 percent memory utilization
   - **SNMP Retries**
     Number of times the program attempts to poll an SNMP device. A high number of SNMP Retries can affect performance, depending on your network configuration.
     Default: 2
   - **Community String**
     SNMP string that the Watchdog Services use to verify the identity of components in a distributed deployment. The community string is used for gathering information from Harvesters. Use the same community name throughout the Network Flow Analysis deployment:
     - **Watchdog Settings** pane
     - SNMP service on each Windows server
     - `snmpd.conf` file on each Linux server

     Default: public
   - **Disk Threshold**
     Threshold for disk utilization. If the disk threshold on a server is exceeded, you are notified by email and an SNMP trap notification is generated, provided that the address and string are set.
     Default: 80 percent disk utilization
   - **SNMP Timeout**
     Number of seconds before an SNMP poll times out.
     Default: 5
   - **CPU Threshold**
     Threshold for CPU utilization. You are notified by email when the CPU threshold on a server is exceeded on any server and an SNMP trap notification is generated, provided that the address and string are set.
     Default: 80 percent CPU utilization
   - **Email Address**
     Destination email address to use for email notifications when thresholds are exceeded. To notify multiple recipients, separate the addresses with commas. The **Email Address** setting has no default value.
     Default: (none)
   - **Trap Community String**
     SNMP string to use for sending traps to a third-party trap receiver. Use one of the community names that the trap receiver is configured to accept.
     Default: public
   - **System Check Interval**
     Number of minutes between Watchdog system checks.
     Default: 60
   - **Trap Destination**
     IP address of the server that receives SNMP traps from the Watchdog Services. The traps are generated when thresholds for DX NetOps component performance are violated.
     Default: (none)
3. Click **Save** when you finish editing the settings.

Add a Harvester

Make sure the Harvesters you add have not been deleted from the **Harvester** pane previously. To add a Harvester instance successfully in DX NetOps after deleting it, the Harvester installation server must be re-imaged and the Harvester software must be re-installed.

Follow these steps:

1. Display the **Harvester** pane:
   a. Hover over **Administration**, and click **Data Sources: the Network Flow Analysis datasource**.
      The page for the data source opens.
   b. Select **Harvester** from the menu on the left side of the page.
      The **Harvester** pane opens and displays the current settings.
2. Click **Add**.
   The **Add Harvester** dialog opens.
3. Enter the following information:
   – **IP Address**
     Address of the Harvester server.
   – **Description**
     Identifying text about the Harvester, which appears in the **Harvester** page table.
   – **Domain**
     Parent tenant and domain combination for the Harvester and for any routers and interfaces that begin to supply data to the Harvester.
     Changing this setting affects the tenant-domain association for new routers that begin exporting flow data and any new interfaces that begin generating flow.
     In a multi-tenant environment, the Harvester tenant affects which SNMP profiles are available for routers to poll interfaces.
     The domain affects which operators and reports have access to the data from routers and interfaces.
     This option is visible only in an environment that contains multiple domains.
4. Click **Save**.
   The new Harvester is added and appears in the Harvester list.
   The usual process is to add one or more Harvesters, then configure the router interfaces to export flow to the Harvesters. If you configure the routers to export flow to the Harvesters first, the DX NetOps console immediately begins to collect data from the new Harvester. In this case, the domain for the routers is set at the time you add the parent Harvester.

   NOTE
   To validate the IP Address, the system checks the reachability of the Harvester on the IP. This process can take up to a minute when the IP is not a valid Harvester IP.

Edit the Details for Harvesters on the Harvester Page

You can edit the IP address, description, and tenant-domain setting.

NOTE
You can edit the IP address with Network Flow Analysis 10.0.4 and higher only.

Follow these steps:

1. Click **Edit** on the Harvester row that you want to edit.
   The **Edit Harvester** dialog opens.
2. Change any of the following settings:
   – **IP Address** (Network Flow Analysis 10.0.4 and higher only)
     Address of the Harvester server.
   – **Description**

(*Optional*) Additional information to help identify the Harvester

- **Domain**
  The tenant-domain association of the Harvester in a multi-domain deployment.
  Changing this setting affects the tenant-domain for any new routers that begin exporting flow data. Existing routers and interfaces retain their previous tenant-domain associations.
  The domain affects which operators and reports have access to the data from routers and interfaces.
  Default: Default Tenant\Default Domain

3. Click **Save**.
   Your changes are saved immediately and appear in the **Harvester** table.

   NOTE
   To validate the IP Address, the system checks the reachability of the Harvester on the IP. This process can take up to a minute when the IP is not a valid Harvester IP.

Delete a Harvester

Once you delete a Harvester, you cannot recover any of the data that the Harvester collected previously.

Follow these steps:

1. Click **Delete** in the row for the Harvester that you want to delete.
   A confirmation message opens.
2. Click **Yes** to confirm that you want to delete the Harvester.
   The Harvester is deleted and is not listed in the **Harvester** table. The NFA console no longer collects data from the routers that are associated with the deleted Harvester. The data that was collected from those routers previously is not available in reports.

Bulk Data Export

As a systems administrator or architect, you want to export polled rate data from Data Aggregator as a continuous CSV export to your own reporting tool for analysis and custom reporting purposes.

IMPORTANT
The bulk data export feature is unsupported in environments with fault tolerant Data Aggregators.

TIP

To export specific data, or to run a one-time export, the OpenAPI is a more flexible. For more information, see OpenAPI.

Data Aggregator provides data export capabilities that are done at the frequency of polling rate. When you start data export, all poll responses are written to CSV files. You can configure inclusive lists for metric families so that you only export the data you need.

The CSV files contain rate data and internal names, which are mapped to readable display names. If exported metric families include suppressed metrics, the CSV shows a null value for those metrics.

The size of CSV output files that are generated can come close to or can exceed the following limits:

Environment	Limit per hour (uncompressed)
Large scale	50 GB
Medium scale	25 GB
Small scale	5 GB

TIP

To avoid exhausting the available disk space, regularly process the output CSV files. For example, copy the files to another system and remove the files from the Data Aggregator host.

For performance reasons, create a separate disk partition on the Data Aggregator system. Select this partition for the output of the CSV rate data.

To export data, follow this process:

Configure the Export File Output Options

Configure the output file options using the **streamexport.csvoutwriter.cfg** file. This configuration file is copied to the directory during the Data Aggregator installation.

Example of the path:

```
/opt/IMDataAggregator/apache-karaf-2.4.3/etc
```

Follow these steps:

1. Change the directory to /opt/IMDataAggregator/apache-karaf-2.4.3/etc.
2. Edit the **streamexport.csvoutwriter.cfg** file as needed:
   - **output.filenameExtension**
     Specifies the extension or suffix of the CSV files.
   - **output.csvFileDelimiter**
     Specifies the column delimiter that is used in the output CSV file. For example, if the bundle is started and this parameter is changed, a new file is written immediately using this new column delimiter.
   - **output.filenameLocationPath**
     Specifies the file path and the prefix of the output file name. (The output file also consists of the date and time.)

     WARNING
     This file path must be on a different partition than the Data Aggregator installation. If DX NetOps Performance Management was installed with a sudo user, change the ownership of the directory to enable access to the sudo user.

     The syntax of the complete file name is:

     ```
     output.filenameLocationPath=<DC_host>_yyyy-MM-dd-
     Thh-mm-sec-ms.output.filenameextension
     ```

     Example:
     The file is written on April 2, 2013 at 8:42:04 a.m. and 123 ms. The Data Collector hostname is server.abc.com, and the following parameters are configured:

     ```
     output.filenameLocationPath=/opt/export_data/mydata
     output.filenameExtension=.csv
     ```

     The file name is:

     ```
     mydata_server.abc.com_2013-04-02T08-42-04-123.csv
     ```

     You can configure files with an absolute name, such as /myOutputDir/mydata. If the parent folders in the absolute path do not exist, the folders are created.
   - **output.filesize**
     Specifies the file size in bytes using a valid integer greater than 0. If the file size is exceeded, a new output file gets written.
     If the value is -1, this parameter is ignored and *all* the data is written into a single file (infinite).
   - **output.duration**
     Specifies the number of minutes using a valid integer greater than 1. If a file is older than *x* minutes, then a new file is written.

If the value is 1, then this parameter is ignored. If the value is -1, then *all* the data is written into a single file (infinite).

The output.filesize and output.duration parameters affect each other. If the output.filesize is exceeded *or* the file is older than output.duration, then a new file is written.

Example: streamexport.csvoutwriter.cfg

```
output.filesize=100000000
output.filenameLocationPath=/opt/data_export/ratedata_
feature.enabled=on
output.duration=60
output.csvFileDelimiter=,
output.filenameExtension=.csv
```

3. Save your changes.

Configure an Inclusive List of Metric Families

You can specify the metric families whose data you do want to export by configuring an inclusive list (whitelist). Data is collected only for the listed metric families. This configuration is optional.

Two files pertain to scoping by metric family:

- **streamexport.allMetricFamilies.out**
  This file is auto-generated at the start-up of the Data Aggregator system. The file includes all the available metric families.
  This file is periodically updated to include new metric families that are added to the system.
  Location: $KARAF_HOME/etc folder
  This file has the following format:

  ```
  metricFamilyInternalName=metricFamilyDisplayNameinEnglish
  ```

  Example:

  ```
  #Metric Family Name List for Customer Reference
  #Mon Jun 17 11:18:50 EDT 2013
  normalizedmemoryinfo=Memory
  normalizedavailabilityinfo=Availability
  normalizedcpuinfo=CPU
  normalizedportinfo=Interface
  ```

- **streamexport.metricFamilyWhiteList.cfg**
  Specifies all the metric families to export.
  Location: $KARAF_HOME/etc folder
  This file has the following format:

  ```
  whitelist.number=metricFamilyInternalName
  ```

  Example:

  ```
  #This is the whitelist file for the metric families.
  #Mon Jun 10 17:25:46 EDT 2013
  feature.enabled=on
  whitelist.1 = NormalizedMemoryInfo
  whitelist.2 = NormalizedPortInfo
  ```

Follow these steps:

1. On the local Data Aggregator system, open **streamexport.allMetricFamilies.out** using a text editor.
2. On the local Data Aggregator system, also open **streamexport.metricFamilyWhiteList.cfg** using a text editor.

3. Copy one or more metric family internal names from the **streamexport.allMetricFamilies.out** file. Paste the copied text into the **streamexport.metricFamilyWhiteList.cfg** file.
   Example:
   ```
   whitelist.1=<paste here>
   whitelist.2=<paste here next metric family name>
   Set feature.enabled=on
   ```
4. Save your changes to the **streamexport.metricFamilyWhiteList.cfg** file.

Export Extra Columns for Components

To include more information about component items, configure the export to include the ifAlias and ifDescr columns.

> **WARNING**
>
> If you configure this option after you start the data export, the output file might contain mixed data. Lines that are exported before you add the columns do not include the extra columns. Lines that are exported after the change include the columns. To avoid a mixed file, stop the Data Aggregator before you change the configuration.

Follow these steps:

1. Edit the following file:
   ```
   etc/streamexport.exportInfoResolver.cfg
   ```
2. Set the following parameter:
   ```
   enableInterfaceOutput=true
   ```
3. Save the changes.
   The Data Aggregator adds the ifAlias and ifDescr columns to the output file. This change occurs immediately.

Start the Rate Data Export Feature

Start the Data Aggregator Rate Data Export Service (export feature) using Data Aggregator REST web services. You can use any REST client tool or an HTTP tool that can send requests and can get responses. For this scenario, we use a REST client.

Follow these steps:

1. Set up a REST client with a connection to the Data Aggregator server.
2. Set the REST Content-type to application/xml.
3. Enter the following URL:
   ```
   GET http://da_host:8581/rest/dataexport/
   ```
4. Take note of the id of the data export profile you want to modify. By default, there is only one profile.
5. Enter text in the Body tab of the HTTP Request pane. At a minimum, set Enabled to true. For example:
   ```
   <DataExportInfo version="1.0.0">
               <Enabled>true</Enabled>
   </DataExportInfo>
   ```
6. Review other options that can be set at the following URL:
   ```
   http://da_host:8581/rest/dataexport/xsd/get.xsd
   ```
7. Save and start the rate data export feature by entering the following URL:
   ```
   PUT http://da_host:8581/rest/dataexport/id
   ```
   – ***id***
   Specifies the ID of the Data Export service.
   The following URL retrieves the ID of the Data Export service:
   ```
   GET http://da_host:8581/rest/dataexport
   ```

8. To verify that your changes took effect, enter the following URL:
   `GET http://da_host:8581/rest/dataexport/id`

   The data export starts automatically and temporary export files are created.
   When the export file is ready, the exporter automatically renames it to the previously configured file extension, such as .csv.
   You do not need to restart the services for a newly written file.
   After the data is exported, copy the data to your other system using the method required by that other system.

Stop the Bulk Data Export

To stop the data rate export at any time, set the Enabled attribute to **false**.
Example:

```
<DataExportInfo version="1.0.0">
 <Enabled>false</Enabled>
</DataExportInfo>
```

View Health Monitoring Information

A built-in mechanism monitors the health of the Data Aggregator and the Data Collector devices. Self-monitoring monitoring profiles determine the statistics that are discovered and polled for these devices. The discovered statistics are collected automatically.

> **WARNING**
> We recommend that you do not change or stop this self-monitoring.

As an administrator, you want to monitor your Data Aggregator and Data Collector items so that you can manage their performances proactively and you can perform capacity planning.

In this scenario, you will view the monitoring profiles that are associated with Data Aggregator and Data Collector self-monitoring. You will also view the components that are being monitored on the Data Aggregator device. This information helps you to understand how the health of these items is managed.

You will also create dynamic trend views, where you can view changes that are occurring on Data Aggregator over time. This information is useful when you are troubleshooting performance issues and performing capacity planning.

To view health monitoring information, follow these steps:

View Metric Families That Are Associated with Self-Monitoring

You can view the metric families that are related to the self-monitoring of Data Aggregator and Data Collector. You can also view the metrics within these metric families, which are used for polling the devices. This information helps you to gain an understanding of what types of data is collected and at what rates the data is collected. You can then understand how the health of these devices is managed. You will see this information in the view that you will create later in this scenario.

Follow these steps:

1. Log in to CA NetOps Portal as an administrator.
2. Select **Administration**, and click a Data Aggregator data source.
   The Data Aggregator Admin pages open.
3. Click **Monitoring Profiles** from the Monitoring Configuration menu.
4. Select each "DA Health (<Rate>)" monitoring profile individually and view the corresponding metric families on the Metric Families tab.

NOTE
To view a description for each metric family, hover over it.

5. Make a note of the metric families you see on the Metric Families tab.
6. Click **Metric Families** from the **Monitoring Configuration** menu.
7. Find and select each metric family individually and view the specific metrics that are collected on the Metrics tab.

View Monitored Components on Data Aggregator

You can view which components are being monitored on Data Aggregator. You can also view the polling status on the components. For example, for capacity planning purposes, you can see all of the Data Collector instances that are being monitored on the Data Aggregator instance.

NOTE
Only Data Collector instances for the tenant you are administering will be displayed. In an enterprise environment, you will likely only use the default tenant workspace and therefore will see all of the Data Collector instances.

Follow these steps:

1. Click **Monitored Devices** from the Monitored Inventory menu for a Data Aggregator data source.
2. Expand the "All Data Aggregators" folder and select the Data Aggregator device.
   The Polled Metric Families tab shows the metric families that are associated with the Data Aggregator device.
   The Components table for a given metric family shows the polling status on the discovered components.

NOTE
When new components are discovered on the self-monitoring metric family, the components are monitored automatically. Performing an "Update Metric Family" operation on a self-monitoring metric family will result in no change in the state of monitoring.

Create a Group

As an administrator, you can create a custom group to organize managed devices in CA NetOps Portal. In this scenario, you create a group for the Data Aggregator device. The Data Aggregator device will be the context for the view you will create in a later step.

Follow these steps:

1. Log in to CA NetOps Portal as a user with the Administrator role.
2. Select **Administration**, **Group Settings**, and click **Groups**.
   The Manage Groups dialog opens.
3. To find a location for the new group, expand nodes in the Groups tree. Create the group under the All Groups node in the Groups tree, or within an existing custom or site group. You cannot add groups to system groups, which appear "locked" in the Groups tree.

NOTE
We recommend that you create this group under a node where only you have access rights.

4. Right-click the node, and select **Add Group**.
   The Add Group window opens. The New tab is selected by default.
5. Type the words "Data Aggregators" for the name of the group and optionally provide a description.
6. Select **Custom** from the Group Type list.
7. Click **Save**.
   The new group appears in the Groups tree.
8. Select the "Data Aggregators" group that you created.
9. Click the Items tab in the right pane.
10. Click **Add Item** Type.

The Add Items dialog opens.
The list of items refreshes to show items of the selected type that are available to add to the group. The available items depend on the item type, the data sources that are registered, and the items discovered.

NOTE
To see more pages of items, click the links below the list. Or use the Search field to search for an item in the list.

11. Select the DataAggregator@*ip_address* item and click Add Items.
    – ***ip_address***
    Indicates the IP address of the Data Aggregator device.
12. Click **Close**.
    The Add Items dialog closes. The Items tab shows the DataAggregator@*ip_address* device that you added. Synchronization with Data Aggregator can take up to 5 minutes to begin.

Create a Dashboard and View Relevant Information

Custom dashboards are useful for displaying data from a particular item or group of items. (*Item* can be a device, component, or interface.) In this scenario, you want to display self-monitoring data on the Data Aggregator item. In particular, you want to look at the Polled Item Count metric, the SNMP Poll Failure Count metric, and the SNMP Request Count metric. By looking at this information, you can view any changes that are occurring over time. Use this information when troubleshooting performance issues and when you are performing capacity planning.

Follow these steps:

1. Click the Dashboards tab.
   The Available Dashboards page opens. Each view on the page corresponds to a menu.
2. Click **Add Dashboard** next to the menu where you want the new dashboard to appear.
   The Add Dashboard page opens.
3. Complete the following fields:
   a. Type "Data Aggregator Health' in the Menu Item and Dashboard Title fields.
   b. (Optional) Select a layout template for the dashboard.
   Each layout treats the page as a table with rows and columns for views. The layout buttons indicate the number of views in each column and row on the page.

   NOTE
   Select your layout before you add the Dynamic Trend View. If you change the layout after you add a view, the views will be removed.
4. Expand **Dynamic Views** in the left pane. Select and drag the Dynamic Trend View to the page layout, and drop the view where you want it to appear.
5. To create two more Dynamic Trend Views, select the view and click the **Copy** button twice.
6. Add the **Polled Item Count** metric to the first view. Add the **SNMP Poll Failure Count** metric to the second view. Add the **SNMP Request Count** metric to the third view. To add the metrics to the view, do the following steps:
   a. Select the Dynamic Trend View and click the **Edit** button.
   b. If it is not already selected, select **Device** from the **Context Type** drop-down box.
   c. For the title, type "Polled Item Count" for the first view, "SNMP Poll Failure Count" for the second view, and "SNMP Request Count" for the third view.
   d. Keep the default, **Multitrend**, as the View Type.
   e. Select the "DA Data Collector Polling Statistics" metric family for each view.

   NOTE
   It can take a few minutes for the available metric families to synchronize with CA NetOps Portal. If the self-monitor metric families are not immediately available, wait a few minutes and then try again.

f. Select the **Polled Item Count - Average** metric value for the first dynamic trend view. Select the **SNMP Poll Failure Count - Total** metric value for the second view. Select the **SNMP Request Count - Total** metric value for the third view.
g. Select the **Add Groups** radio button and then click the **Add/Remove Groups** button. Select the "Data Aggregators" group you created previously from the list of available groups and move it to the list of selected groups. Click OK.
h. Click **Save**.

Your report is generated automatically.

7. View any changes that are occurring over time.
   For example:
   – Polled item count increases when you were *not* expecting it to - Did you unintentionally start monitoring more items?
   – Polled item count remains the same, but you see spikes in the SNMP Request Count occasionally - Do you have change detection turned on in one of your monitoring profiles or perhaps scheduled discoveries?
   – Polled item count remains the same, but you see changes in the SNMP Poll Failure Count - Perhaps there has been a network outage.
   – Polled item count increases and the SNMP Poll Failure Count increases - Perhaps Data Collector is overloaded and cannot respond to all of the requests.

Restart Performance Management Component Services

To restart DX NetOps Performance Management, restart the component services on the host servers.

See the following pages for the procedures to restart the component services:

Restart the Data Aggregator

Restart the Data Aggregator service when required, such as when you upgrade the operating system on the Data Aggregator host. Stop Data Aggregator, complete the required actions, and restart Data Aggregator.

During a planned shutdown, the Data Aggregator continues processing for 5 minutes before the service stops:

- Data that has been received continues processing and is sent to the Data Repository. Data that is not processed is saved, and loaded when the Data Aggregator restarts.
- Incomplete rollup processing resumes when the Data Aggregator restarts.

Data Collectors queue poll data when the Data Aggregator is stopped. Data loading of queued data, threshold event processing, and rollup processing resume when Data Aggregator restarts. If queued data exceeds the disk space limit on the Data Collector, the oldest data is discarded.

If the Data Aggregator stops ungracefully, polled data and threshold event processing information might be lost.

Fault Tolerant Data Aggregators

If you are applying a patch to Data Aggregators in a fault tolerant environment, do the following restart steps in the following order:

1. Put the inactive Data Aggregator(B) into maintenance mode.
2. Apply the patch to the inactive Data Aggregator(B).
3. Activate the inactive Data Aggregator(B).
   This Data Aggregator(B) becomes available for failover.
4. Put the active Data Aggregator(A) into maintenance mode.

WARNING

This step causes failover. The previously inactive Data Aggregator(B) becomes the active Data Aggregator.

5. Apply the patch to the now inactive Data Aggregator(A).
6. Activate the now inactive Data Aggregator(A).
   This Data Aggregator(A) becomes available for failover. The other Data Aggregator(B) is the active Data Aggregator.

Stop the Data Aggregator

Shut down the Data Aggregator in preparation for any required tasks (for example, applying a security patch). In an environment with fault tolerant Data Aggregators, put the inactive Data Aggregator and then the active Data Aggregator into maintenance mode.

Follow these steps:

1. Log in to the Data Aggregator host as the root user or the sudo user.

   NOTE
   If you installed Data Aggregator as the sudo user, you set up a sudo command alias for the service dadaemon command. Use the sudo commands.
2. Do one of the following steps:
   – Stop the Data Aggregator service:
   `service dadaemon stop`

   NOTE

   For RHEL 7.x, `service` invokes `systemctl` . You can use `systemctl` instead.
   – (Fault tolerant environment) If the local Data Aggregator is running, run one the following commands to shut it down and prevent it from restarting until maintenance is complete:
   - **RHEL 6.x:**
     `service dadaemon maintenance`
   - **RHEL 7.x**, **SLES, or OL:**
     DA_Install_Directory`/scripts/dadaemon maintenance`

   The Data Aggregator completes processing and the service stops.

Start the Data Aggregator

After you complete any required tasks (for example, applying a security patch), restart the Data Aggregator. In an environment with fault tolerant Data Aggregators, activate each Data Aggregator.

Follow these steps:

1. Do one of the following steps:
   – Start the Data Aggregator service:
   `service dadaemon start`
   – (Fault tolerant environment) Run one the following commands to enable the fault tolerant Data Aggregator so that it can start when necessary:
   - **RHEL 6.x:**
     `service dadaemon activate`
   - **RHEL 7.x**, **SLES, or OL**:
     DA_Install_Directory`/scripts/dadaemon activate`

   Data Aggregator starts and synchronizes with CA NetOps Portal automatically. If the Data Repository is unavailable, the Data Aggregator shuts down. The Data Collectors send any queued data to the Data Aggregator.

Restart the Data Collector

Restart the Data Collector service when required, such as for the following reasons:

- The host loses power.
- You change the location of the host.
- You install an operating system patch.

When a Data Collector is stopped, polling stops. Discovery cannot run during this time. When Data Collector is restarted, scheduled polling and discovery resumes.

Follow these steps:

1. Log in to the Data Collector host as the root user or the sudo user.

   NOTE
   If you installed Data Collector as the sudo user, you set up a sudo command alias for the service dcmd command. Use the sudo commands.
2. Use one of the following commands to stop the Data Collector service:
   –
   ```
   service dcmd stop
   ```
   –
   ```
   sudo service dcmd stop
   ```
   The Data Collector stops.
3. Complete any required tasks.
4. Use one of the following commands to start the Data Collector service:
   –
   ```
   service dcmd start
   ```
   –
   ```
   sudo service dcmd start
   ```
   The Data Collector restarts and resynchronizes automatically. If the Data Aggregator is not available, the Data Collector shuts down. If the ActiveMQ Broker is unavailable, the Data Collector restarts that service.

Restart the Data Repository

Restart the data repository when required, such as for the following reasons:

- The data repository host loses power or locks up.
- The data repository host is relocated.
- You install an operating system patch.

NOTE
This procedure includes a restart of the data aggregator.
For more information, see Restart the Data Aggregator.

WARNING

If you try to restart one data repository node before another restarted node has completely rejoined the cluster, the whole cluster can go down. The cluster does not come back up until you manually start it.

Stop the Data Aggregator

Before you shut down the data repository, stop the data aggregator. In an environment with fault-tolerant data aggregators, put the inactive data aggregator and then the active data aggregator into maintenance mode.

Follow these steps:

1. Log in to the data aggregator host as the root user or a sudo user.
2. Do one of the following steps:
   – Stop the Data Aggregator service by issuing the following command:
   `service dadaemon stop`

   NOTE
   For RHEL 7.x, `service` invokes `systemctl` . You can use `systemctl` instead.
   – (Fault-tolerant environment) Issue one of the following commands to shut down the inactive data aggregator and prevent it from restarting. Then run the command on the active data aggregator:
   - **RHEL 6.x:**
     `service dadaemon maintenance`
   - **RHEL 7.x, SLES, OL:**
     DA_Install_Directory`/scripts/dadaemon maintenance`

Shut Down the Data Repository

Shut down the data repository in preparation for any required tasks (for example, applying a security patch).

WARNING

If you try to restart a data repository node before another restarted node has completely rejoined the cluster, the whole cluster can go down. The cluster does not come back up until you manually start it.

Stop the Database

Follow these steps:

1. Log in to the data repository host as the database administrator user.
2. Open Vertica adminTools by issuing the following command:
   `/opt/vertica/bin/adminTools`
3. Select **(4) Stop Database**.
4. Select the database, select **OK**, and then press **Enter** on your keyboard.
5. Provide the database password, and then press **Enter** on your keyboard.
   The data repository stops.

   TIP

   If the data repository does not stop, select **(2) Stop Vertica on Host** from **(7) Advanced Tools Menu**.
6. Select **(E) Exit**, and press **Enter** on your keyboard.

Stop Vertica on One Node in a Cluster

Follow these steps:

1. With the Vertica adminTools open, select **(7) Advanced Tools Menu**.
2. Select **(2) Stop Vertica on Host**.
3. Select the node to stop, and then press **Enter** on your keyboard.
4. Provide the database password, and then press **Enter** on your keyboard.
   The data repository on the node stops.

5. Select **(M) Main Menu**, and then press **Enter** on your keyboard.
6. Select **(E) Exit**, and then press **Enter** on your keyboard.

Start the Data Repository

After you complete required tasks, such as applying a security patch, restart the data repository.

WARNING

To avoid the cluster from going down and having to manually restart it, ensure that other restarted nodes have completely rejoined the cluster before restarting a data repository node.

Start the Database

Follow these steps:

1. With the Vertica adminTools open, select **(3) Start Database**.
2. Select the database, select **OK**, and then press **Enter** on your keyboard.
   You are prompted for the database password.
3. Provide the database password, and then press **Enter** on your keyboard.
   The Data Repository starts.

   TIP

   If Vertica fails to start normally, try running the following command:

   ```
   /opt/vertica/bin/admintools -t restart_node -d dbname --hosts host_ip_address --force
   ```
4. Select **(E) Exit** and press Enter.

Start a Node that is Down in a Cluster

Follow these steps:

1. With the Vertica adminTools open, select **(5) Restart Vertica on Host**.
2. Select the database, select **OK**. and then press **Enter** on your keyboard.
3. Select the node to start, and then press **Enter** on your keyboard.
4. Provide the database password, and then press **Enter** on your keyboard.
   The Data Repository on the node starts.
5. Select **(E) Exit**, and then press **Enter** on your keyboard.

Start the Data Aggregator

After you start the data repository, start the data aggregator. In an environment with fault-tolerant data aggregators, activate each data aggregator.

Follow these steps:

1. Do one of the following steps:
   – Start the Data Aggregator service:
   ```
   service dadaemon start
   ```
   – (Fault-tolerant environment) Run one the following commands to enable the fault-tolerant data aggregator so that it can start when necessary:
   - **RHEL 6.x:**
   ```
   service dadaemon activate
   ```
   - **RHEL 7.x, SLES, OL:**

```
DA_Install_Directory/scripts/dadaemon activate
```

The data aggregator starts and the system resumes normal operations.

Restart the ActiveMQ Broker

If a problem is detected, the Data Aggregator or Data Collector restarts the ActiveMQ broker automatically. If the restart is unsuccessful, restart the ActiveMQ broker manually. By default, the Data Aggregator and Data Collector determine the availability of the ActiveMQ broker every 30 seconds.

Follow these steps:

1. Open the ActiveMQ, which is located in the following location:

   ```
   da_install_dir/broker/apache-activemq-version/bin
   ```

   - ***da_install_dir*** specifies the location of the Data Aggregator or Data Collector installation directory.
   - **apache-activemq-*version*** specifies the version of Apache ActiveMQ.
     Example: apache-activemq-5.5.1b
2. Stop ActiveMQ:

   ```
   service activemq stop
   ```

3. Start ActiveMQ:

   ```
   service activemq start
   ```

Restart Performance Center

You can stop and restart NetOps Portal, such as when you are backing up your database. To restart NetOps Portal successfully, stop and restart all relevant services in the correct order.

Stop NetOps Portal

Run the following commands in the order in which they appear to stop NetOps Portal:

```
service caperfcenter_console stop

service caperfcenter_devicemanager stop

service caperfcenter_eventmanager stop

service caperfcenter_sso stop
```

(Optional) Restart the MySQL Database

Usually, you do not need to restart MySQL database. If you are experiencing issues with NetOps Portal, you can restart the MySQL database.

WARNING

Before you stop the MySQL database, verify that all NetOps Portal services are stopped.

Run the following commands to stop and restart MySQL:

```
service mysql stop

service mysql start
```

Start NetOps Portal

To start NetOps Portal, restart the component services.

Follow these steps:

1. Start the SSO service:

   ```
   service caperfcenter_sso start
   ```

2. Wait one minute, then start the event manager and device manager:

   ```
   service caperfcenter_eventmanager start

   service caperfcenter_devicemanager start
   ```

3. Wait one minute, then start the console service:

   ```
   service caperfcenter_console start
   ```

Verify Status of NetOps Portal Services

Run the following commands to verify the status of NetOps Portal services after you start NetOps Portal:

```
service caperfcenter_sso status

service caperfcenter_eventmanager status

service caperfcenter_devicemanager status

service caperfcenter_console status
```

The following sample message indicates that the NetOps Portal console is running:

```
Performance Center Console is running: PID:12993, Wrapper:STARTED, Java:STARTED
```

Disaster Recovery

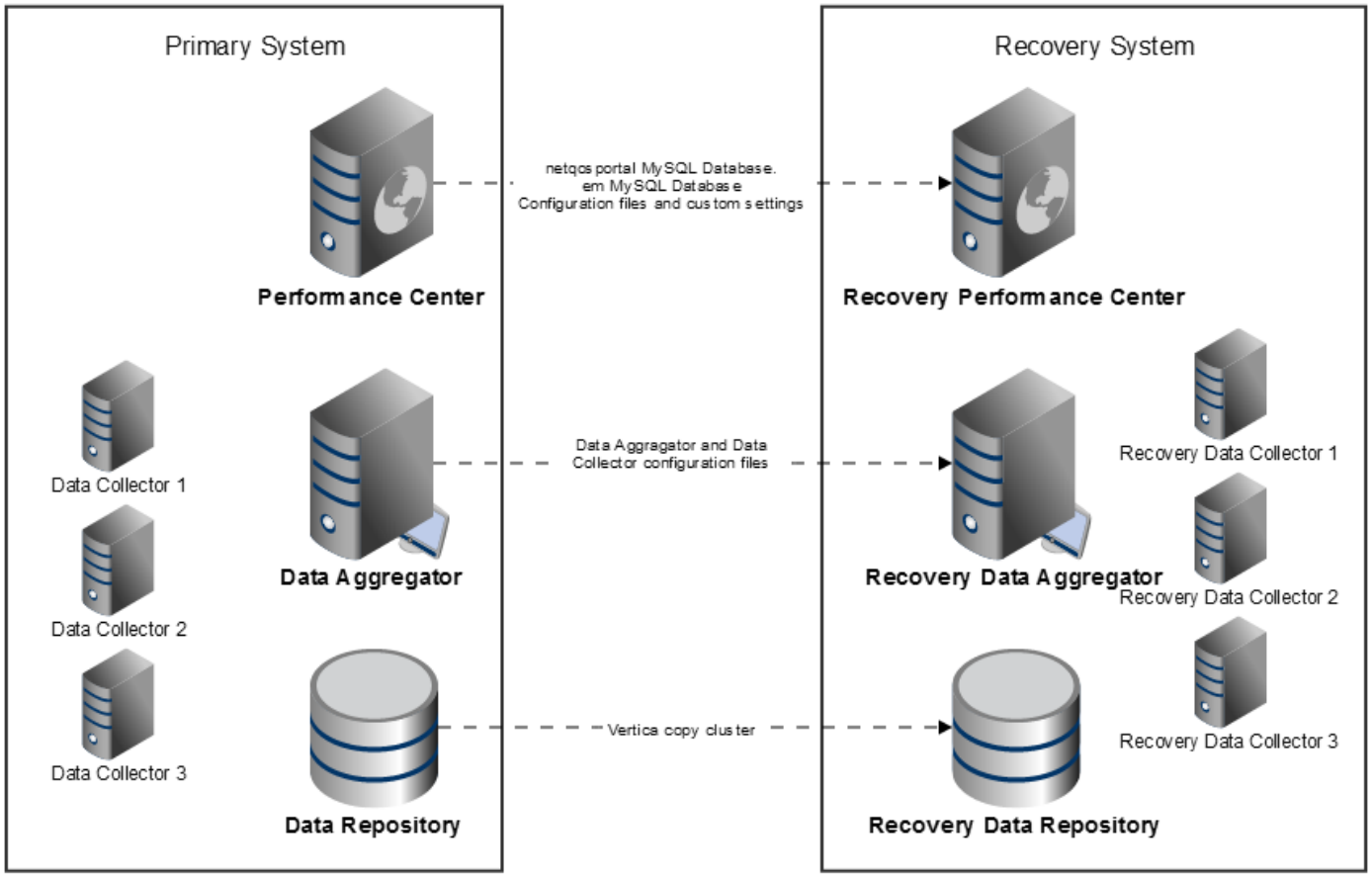
If a large-scale disaster occurs, the disaster recovery plan for DX NetOps Performance Management enables a switchover to a recovery system. The disaster recovery plan involves provisioning a secondary system as a recovery environment and regularly transferring data from the primary system. This scenario is not a temporary measure. The recovery system completely replaces the primary system.

> **NOTE**
> To reverse this process and return monitoring to the original site, go through the same process with the recovery and primary hosts switched. If new hardware is deployed due to the disaster, start by reinstalling the DX NetOps Performance Management components on the new hardware. If the original hardware is available and no upgrades have occurred on the active recovery system, start by configuring incremental data transfer.

The following video introduces how DX NetOps Performance Management establishes a detailed disaster recovery procedure meant to re-establish normal operations in the event of a major disruption such as a hurricane or fire:

The following diagram shows the primary and recovery systems, and the files that are copied regularly:

Figure 61: Disaster Recover Architecture



Install Components for the Recovery System

The recovery system is a secondary system that contains all the components for DX NetOps Performance Management. Under normal operations, the recovery system is offline. The recovery system has the same requirements as the primary system. For more information, see Review Installation Requirements and Considerations.

IMPORTANT
If you upgrade you primary system, upgrade the components in the recovery system. Each recovery component must be running the same version of the product as the primary system.

truetop

Install the Data Repository

Before you install Vertica on the recovery cluster, prepare the environment for the installation. For more information about how to prepare the recovery cluster, see Prepare to Install the Data Repository.

Ensure that the recovery system has the same settings as the primary cluster for the following settings:

- Database version
- Node names

  TIP
  To get the node name, run the following command on each node:
  ```
  /opt/vertica/bin/admintools -t list_allnodes
  ```
  This command also returns the installed Vertica version and the database name.
- Database name
- Database administrator
- Database user
- Catalog directory

  TIP
  To get the catalog directory configuration, run the following command on each node:
  ```
  /opt/vertica/bin/admintools -t list_db -d <database name>
  ```
- Data directory

The recovery cluster has the following requirements:

- Accessible from the primary cluster
- The same number of nodes as the primary cluster
- Passwordless SSH access to and from the recovery cluster for the Database Administrator account (default: dradmin) from the primary cluster.
  Provide passwordless SSH access for the Database Administrator account (default: dradmin) from each host in the primary cluster to each host in the recovery cluster. Then, provide passwordless SSH access for the dradmin account from each host in the recovery cluster back to each host in the primary cluster.

  TIP
  To configure passwordless SSH, run the following command from each host in the primary cluster to each node in the recovery cluster. Then run the command from each node in the recovery cluster to each node in the primary cluster:
  ```
  ssh-copy-id -i dradmin@target_host
  ```
- For each node pair, run the following command:
  ```
  ssh dradmin@<paired-node> '/opt/vertica/bin/admintools -t list_allnodes
  ```
  If you are prompted for a password, the ssh setup for the Database Administrator account (default: dradmin) is incomplete. Copy cluster will fail.
- Port 50000 must be open between all the Data Repository nodes and disaster recovery hosts.

The installation process is the same as a normal Data Repository installation, for more information, see Install the Data Repository.

WARNING

Use the same configuration for the new cluster as for the source cluster. For example, the vertica version, node count, database name, administrator, user, catalog directory, and data directory must be the same as the original Data Repository.

Install the Data Aggregator

Prepare the host and install the data aggregator for the recovery system.

During installation, use the details for the Data Repository for the recovery system.

For more information, see Prepare to Install the Data Aggregator and Install the Data Aggregator.

Install the Data Collectors

For each data collector in the primary system, install a data collector in the recovery system. The DCM ID identifies the data collector to the system.

TIP
This scenario assumes that the data collectors are centrally located with the other components. Some deployments use remote data collectors, which are deployed close to the monitored infrastructure in other data centers or geographical locations. To continue using a remote data collector if a disaster occurs, update it to communicate to the recovery data aggregator. For more information, see Configure Data Collector When the Data Aggregator IP Address Changes.

Use the following process:

1. Prepare the hosts for the recovery system data collectors. For more information, see Prepare to Install the Data Collectors.
2. Record the DCM ID values for each data collector in the primary system:
   a. Go to the following URL:
      `https://primary_da_host:8581/rest/dcms`
   b. For each data collector, note the value in the **<DcmID>** tag.
      The following XML shows an example:

```
<DataCollectionMgrInfoList>
   <DataCollectionMgrInfo version="1.0.0">
      <ID>4077</ID>
       <DcmID>primary-dc:69658898-a48c-44a6-9cba-963bb9c09684</DcmID>
      <Enabled>true</Enabled>
      <IPAddress>10.237.1.67</IPAddress>
      <RelatedDeviceItem>4078</RelatedDeviceItem>
...
```

3. For *each* recovery data collector, export the DCM ID for the corresponding primary data collector, and install the component in the recovery system:
   a. On the data collector host for the recovery system, export the DCM ID for the primary system data collector:
      `export DCM_ID=DATA_COLLECTOR_DCM_ID`

      Example:
      `export DCM_ID=primary-dc:69658898-a48c-44a6-9cba-963bb9c09684`

b. From the same session, install the data collector.
During installation, specify the details for the data aggregator for the recovery system.
For more information, see Install the Data Collectors.

4. To verify the installation, look at the DCM ID on the recovery data collector:
   a. Open the following file:
   `/opt/IMDataCollector/broker/apache-activemq-version/conf/activemq.xml`
   b. Find the broker name property and verify the DCM ID.
   The following example shows the section of the activemq.xml file that includes the broker name:

```
...    <broker
         xmlns="http://activemq.apache.org/schema/core"
         brokerName="dc_broker_69658898-a48c-44a6-9cba-963bb9c09684 "
         dataDirectory="${activemq.data}"
         useShutdownHook="false"
         useJmx="true">
 ...
```

   If the broker name does not match the expected DCM_ID UUID from the originating system, update the file with the correct DCM_ID UUID from the DCM_ID of the originating data collector.
5. Stop the data collector services:

```
service dcmd stop
```

NOTE

For RHEL 7.x or OL, `service` invokes `systemctl` . You can use `systemctl` instead.

```
service activemq stop
```

6. Stop the data aggregator services:
   a. Log in to the data aggregator host for the recovery system.
   b. Do one of the following steps:
   - Stop the data aggregator and ActiveMQ services:

```
service dadaemon stop

service activemq stop
```

   - (Fault tolerant environment) If the local data aggregator is running, run one of the following commands to shut it down and prevent it from restarting until maintenance is complete:
     - **RHEL 6.x:**

```
service dadaemon maintenance
```

     - **RHEL 7.x, SLES, or OL:**

```
DA_Install_Directory/scripts/dadaemon maintenance
```

Install NetOps Portal

Prepare the host and install NetOps Portal for the recovery system.

Do not configure LDAP integration or HTTPS on the recovery system. The settings are inherited from the primary system.

For more information, see Prepare to Install Performance Center and Install Performance Center.

Configure Incremental Data Transfer

Copy data provides the recovery system everything that is required to continue operation when the primary system is down. Devise a plan with a regular interval that occurs often enough to duplicate the required data. Use the same frequency for all components and start the transfer from each component simultaneously. We recommend a daily transfer of all components.

TIP
The data aggregator and NetOps Portal components require regular file copies between the primary system and the recovery system. You can use any file copy and schedule method as required in your system. In our lab environment, we configured SSH between the primary and recovery system, and used crontab to invoke the SCP command from the recovery system.

Example:

The following crontab example is configured on the secondary data aggregator host to copy a backup directory from the primary data aggregator. The copy occurs daily at 12:30 AM:

```
30 0 * * * scp -r root@primary_DA:/tmp/backup /tmp/backup
```

top

Data Repository

For the Data Repository, use copy cluster to duplicate the primary database to the recovery database. Copy cluster is an incremental backup that copies all updates to the database. Because this data transfer is the longest transfer, the backup frequency to the recovery system is limited by the runtime of the copy cluster command. Run the command multiple times before you schedule a regular transfer to verify the runtime. Ensure that the backup frequency is at lease twice the runtime of the copy cluster.

NOTE
For existing large databases, the copy cluster command takes as long as a full backup to complete. To minimize the performance impact to the system, restore a backup of the primary system to the recovery system, then configure and run copy cluster.

TIP
For a large database, an incremental copy cluster command for one day takes about one hour. We recommend you run an incremental copy cluster at least daily.

Follow these steps:

1. Create a configuration file for copy cluster on a host in the primary Data Repository cluster in the following directory: `/home/dradmin`
2. Use the example as a model to create the configuration file.
   Example: The following example configuration file is set up to copy a database on a three node cluster (v_drdata_node0001, v_drdata_node0002, and v_drdata_node0003) to another cluster consisting of nodes (recovery-host01, recovery-host02, and recovery-host03):

   NOTE
   The dbName parameter is case-sensitive.

```
[Misc]
snapshotName = Copydrdata
```

```
restorePointLimit = 5
tempDir = /tmp/vbr
retryCount = 5
retryDelay = 1

[Database]
dbName = drdata
dbUser = dradmin
dbPassword =  password
dbPromptForPassword = False

[Transmission]
encrypt = False
checksum = False
port_rsync = 50000

[Mapping]
; backupDir is not used for cluster copy
 v_drdata_node0001= recovery-host01:/data
 v_drdata_node0002= recovery-host02:/data
 v_drdata_node0003= recovery-host03:/data
```

3. Stop the database in the recovery system:
   a. Log in to the recovery database cluster as the database admin user.
   b. Open Vertica admin Tools:
      ```
      /opt/vertica/bin/adminTools
      ```
   c. Select (4) Stop Database. Wait for the shutdown to complete before you run copy cluster.
4. Copy historical data:
   a. Log in to the primary cluster as the database administrator account.
   b. Run the copy cluster command:
      ```
      vbr.py --task copycluster --config-file
      home/dradmin/CopyClusterConfigurationFile.ini
      ```

      The command copies the historical data for the database and displays the following message:
      ```
      > vbr.py --config-file home/dradmin/CopyClusterConfigurationFile.ini --task
      copycluster
      Preparing...
      Copying...
      1871652633 out of 1871652633, 100%
      All child processes terminated successfully.
      copycluster done!
      ```
5. Create a cron job to schedule copy cluster from the primary system on a regular interval. The following command initiates the transfer:
   ```
   vbr.py --task copycluster --config-file
   home/dradmin/CopyClusterConfigurationFile.ini
   ```

Data Aggregator

For the data aggregator, schedule a regular copy of the following files from the primary system to the recovery system:

- *DA_install_directory*/apache-karaf-*version*/deploy/*.xml

  NOTE
  Do not copy the following file from this directory: `local-jms-broker.xml` This directory might not initially contain other files.
- *DA_install_directory*/apache-karaf-*version*/etc/org.ops4j.pax.web.cfg
- *DA_install_directory*/data/custom/devicetypes/DeviceTypes.xml
  In a fault tolerant environment, a shared directory (example: `/DASharedRepo`) is defined to help limit data loss. Therefore, in a fault tolerant environment the file would be located in the following directory:
  DASharedRepo/custom/devicetypes/DeviceTypes.xml

  For more information, see Fault Tolerance.

Data Collectors

The data collectors do not require a regular backup. All relevant information is stored on the data aggregator and Data Repository.

NOTE
If the primary data collectors include custom memory settings, configure the recovery data collectors as required.

NetOps Portal

For NetOps Portal, create a database dump of the netqosportal and em databases, and back up custom settings. For more information, see Back Up and Restore the CA Performance Management Database.

Prepare the Disaster Recovery Scripts

The disaster recovery scripts replace hostname and IP address references to match the components in the recovery system.

For each script, create a copy, and provide the relevant information for your system.

Data Repository Disaster Recovery Script

Location: `/opt/CA/IMDataRepository_vertica`*Version*`/update_da_dc_database_references.sh`

On the Data Repository host in the recovery system, update the bold sections of the script to match your system:

```
############################################################
# UPDATE DAUSER/DAPASS BELOW TO REFLECT THE NON-ADMIN
# VERTICA USERNAME/PASSWORD FOR THIS SYSTEM
############################################################
DAUSER=dauser
DAPASS=dapass


#####################################################################
# UPDATE TO REFLECT THE NEW/RECOVERY DATA AGGREGATOR'S IP ADDRESS BELOW
```

```
#####################################################################
RECOVERY_DA_IP_ADDRESS="<Recovery/New IP Address for the Data Aggregator>"


#####################################################################
# UPDATE TO REFLECT THE NEW/RECOVERY DATA AGGREGATOR'S HOSTNAME BELOW
#####################################################################
SOURCE_DA_HOSTNAME="<Source/Original Hostname for the Data Aggregator>"
RECOVERY_DA_HOSTNAME="<Recovery/New Hostname for the Data Aggregator>"


#############################################################################################################
#
# UPDATE THE FOLLOWING ARRAYS TO REFLECT THE SOURCE DATA
# COLLECTOR HOSTNAMES, NEW RECOVERY HOSTNAMES, AND NEW RECOVERY
# IP ADDRESSES RESPECTIVELY.
#
# IMPORTANT: THE ORDER OF THE ENTRIES BELOW IS CRITICAL FOR
# MAPPING PURPOSES.  IN ADDITION, PLEASE NOTE THAT IF MULTIPLE VALUES
# ARE REQUIRED, PLEASE SEPARATE VALUES WITH A SINGLE SPACE.
#
#############################################################################################################
declare -a SOURCE_DC_HOSTNAMES=(<Source/Original DC Hostname 1> <Source/Original DC
Hostname 2>)
declare -a RECOVERY_DC_HOSTNAMES=(<New/Recovery DC Hostname 1> <New/Recovery DC
Hostname 2>)
declare -a RECOVERY_DC_IP_ADDRESSES=("<New/Recovery DC Hostname 1 IP Address>" "<New/
Recovery DC Hostname 2 IP Address>")
```

IMPORTANT
Ensure that the order of the data collectors for the source system and the recovery system is the same. The script uses the order of the list to map the primary system components to the recovery system.

NetOps Portal Disaster Recovery Script

Location: `/opt/CA/PerformanceCenter/Tools/bin/update_pc_da_database_references.sh`

On the NetOps Portal host in the recovery system, update the bold sections of the script to match your system:

```
...
#####################################################################
# UPDATE THE FOLLOWING PC/DA VARIABLES TO REFLECT NEW ENVIRONMENT
#####################################################################
NEW_PC_IP_ADDRESS="<Recovery/New PC IP Address>"
NEW_PC_HOSTNAME="<Recovery/New PC Hostname>"
NEW_PC_EVENT_PRODUCER_PORT=8181
NEW_PC_EVENT_PRODUCER_PROTOCOL="http" # change to "https" if using SSL
```

```
NEW_DA_IP_ADDRESS="<Recovery/New DA IP Address>"
NEW_DA_HOSTNAME="<Recovery/New DA Hostname>"
NEW_DA_PORT_NUMBER=8581
...
```

Activate the Recovery System

If a large-scale disaster occurs, and the primary system is unavailable, start the recovery system.

TIP
Startup time for the recovery system takes the same time that is required to start the data aggregator.

Start the Data Repository

Follow these steps:

1. Log in to the recovery database cluster as the database admin user.
2. Open Vertica admin Tools:
   `/opt/vertica/bin/adminTools`
3. Select (3) Start Database.
4. Press the Space bar next to the database name, select OK, and then **Enter**.
   You are prompted for the database password.
5. Enter the database password, and then press **Enter**.
   Data Repository starts.
6. Select **Exit**, and then **Enter**.
7. Run the Data Repository disaster recovery script:
   `/opt/CA/IMDataRepository_vertica/`*`your_update_da_dc_database_references.sh`*

Start the Data Aggregator

Do one of the following steps:

- Start the ActiveMQ and data aggregator services:
  ```
  service activemq start

  service dadaemon start
  ```
- (Fault tolerant environment) Run one of the following commands to enable the fault tolerant data aggregator so that it can start when necessary:
  - **RHEL 6.x:**
    ```
    service dadaemon activate
    ```
  - **RHEL 7.x, SLES, or OL:**
    `DA_Install_Directory``/scripts/dadaemon activate`

The data aggregator starts. If the Data Repository is unavailable, the data aggregator shuts down.

Start NetOps Portal

Follow these steps:

1. Restore the NetOps Portal backups. For more information, see Restore Performance Center.
2. Run the NetOps Portal disaster recovery script:
   `/opt/CA/PerformanceCenter/Tools/bin/your_update_pc_da_database_references.sh`
3. Start the SSO service:
   `service caperfcenter_sso start`
4. Wait one minute, then start the event manager and device manager:
   `service caperfcenter_eventmanager start`
   `service caperfcenter_devicemanager start`
5. Wait one minute, then start the console service:
   `service caperfcenter_console start`

Start the Data Collectors

Run the following command to start the data collector service:

```
service dcmd start
```

The data collector restarts. If the data aggregator is unavailable, the data collector shuts down.

> **TIP**
> For remote data collectors, update them to connect to the recovery data aggregator. For more information, see Configure Data Collector When the Data Aggregator IP Address Changes.

(Optional) Test the Recovery System

You can optionally test the recovery system manually.

Follow these steps:

1. Pause the incremental data transfer.
2. Start the Data Repository:
   a. Log in to the recovery database cluster as the database admin user.
   b. Open Vertica admin Tools:
      `/opt/vertica/bin/adminTools`
   c. Select (3) Start Database.
   d. Press the Space bar next to the database name, select OK, and press **Enter**.
      You are prompted for the database password.
   e. Enter the database password and press **Enter**.
      Data Repository starts.
   f. Select Exit, and press **Enter**.
   g. Run the Data Repository disaster recovery script:
      `/opt/CA/IMDataRepository_vertica/your_update_da_dc_database_references.sh`

3. Start the data aggregator:
   a. Do one of the following steps:
      - Start the ActiveMQ and data aggregator services:
        ```
        service activemq start

        service dadaemon start
        ```
      - (Fault tolerant environment) Run one of the following commands to enable the fault tolerant data aggregator so that it can start when necessary:
        - **RHEL 6.x:**
          ```
          service dadaemon activate
          ```
        - **RHEL 7.x, SLES, or OL:**
          ```
          DA_Install_Directory/scripts/dadaemon activate
          ```

   The data aggregator starts. If the Data Repository is unavailable, the data aggregator shuts down.
4. Start NetOps Portal:
   a. Restore the NetOps Portal backups. For more information, see Restore Performance Center.
   b. Run the NetOps Portal disaster recovery script:
      ```
      /opt/CA/PerformanceCenter/Tools/bin/your_update_pc_da_database_references.sh
      ```
   c. Start the SSO service:
      ```
      service caperfcenter_sso start
      ```
   d. Wait one minute, then start the event manager and device manager:
      ```
      service caperfcenter_eventmanager start
      service caperfcenter_devicemanager start
      ```
   e. Wait one minute, then start the console service:
      ```
      service caperfcenter_console start
      ```
5. Log in to the recovery NetOps Portal component and run reports against the recovery Data Repository and data aggregator.
6. Verify the data is available.
7. (Optional) If you have a set of recovery data collectors that you can double-poll for testing, start one or more data collectors. Verify polling occurs and the data is stored in the database.

   NOTE
   To prevent the recovery system from issuing duplicate notifications and reports during testing, disable them beforehand.

   a. Run the following command to start the data collector service:
      ```
      service dcmd start
      ```

   The data collector restarts. If the data aggregator is unavailable, the data collector shuts down.

   TIP
   For remote data collectors, update them to connect to the recovery data aggregator. For more information, see Configure Data Collector When the Data Aggregator IP Address Changes.

8. Shut down each component.
   After the next incremental data transfer, the database is in sync with the primary systems again. To fail over or test again, repeat these steps.

Fault Tolerance

Fault tolerance enables DX NetOps Performance Management to continue operating properly when a hardware failure or network issue occurs. In an environment where fault tolerance is configured, a secondary inactive data aggregator automatically becomes active. The newly-active data aggregator takes over to organize and feed data to NetOps Portal and to the data repository. The newly-active data aggregator retains state information from the previously-active data aggregator. The data aggregator host the network issue or hardware failure is available for failover when it becomes available.
For more information about how to view the health of your data aggregators, see View the Health of the System.

NOTE
If failover occurs, but the ActiveMQ process is still running, you can manually stop ActiveMQ by issuing the following command:

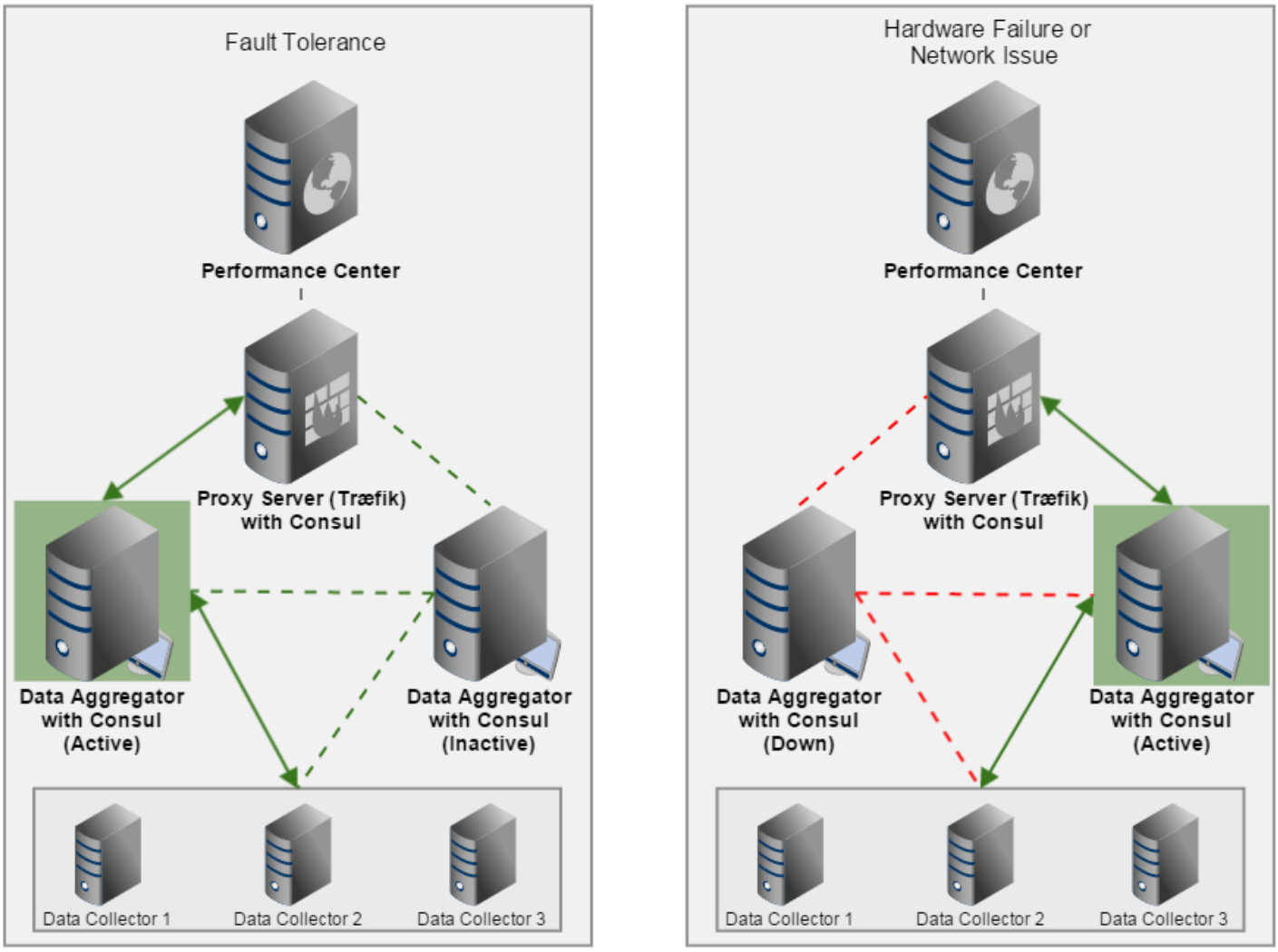
```
service activemq stop
```

Fault-Tolerant Architecture

The following diagram shows the system architecture of a fault-tolerant environment:

NOTE
Træfik is a modern HTTP reverse proxy and load balancer that you can use to deploy microservices with ease. Consul is a tool that you can use to manage services in the DX NetOps Performance Management deployment.

Figure 62: High Availability



Hardware Requirements

The following extra hardware is required for a fault-tolerant environment:

- An additional data aggregator server
- A proxy server. The proxy server works as the third node of the service management cluster for fault tolerance.
- Ensure that you have a new shared data directory (for example, `/DASharedRepo`) and that the same user ID is shared between data aggregator hosts. Data from whichever data aggregator is active is stored in this directory. For more information about the sizing requirements, see the DX NetOps Performance Management Sizing Tool.

> **NOTE**
> If you are using Network File System (NFS), DX NetOps Performance Management supports only NFSv4 and higher because of the ActiveMQ Kaha locking requirements.

> **WARNING**
> To avoid data loss and to prevent data from loading, the shared data directory must be accessible and up at all times.

Data Loss Comparison

In a fault-tolerant environment, some data loss might still occur when a hardware failure or network issue occurs. However, the amount of data loss is less than in an environment without fault tolerance configured. The following table compares the data loss from a hardware failure or network outage:

	Hardware Failure		Network Outage	
Is fault tolerance configured?	**No**	**Yes**	**No**	**Yes**
What happens to rollups?	Pending rollups are lost and never recovered.	The other available data aggregator consumes the pending rollups when it becomes active.	Pending backups are consumed when the network is restored.	The other available data aggregator consumes the pending rollups when it becomes active.
What is lost in memory?	For 10K polls in memory at scale, loss should not exceed 1 poll cycle. Max loss would be 10K items per metric family.	For 10K polls in memory at scale, loss should not exceed 1 poll cycle. Max loss would be 10K items per metric family.	For 10K polls in memory at scale, loss should not exceed 1 poll cycle. Max loss would be 10K items per metric family.	For 10K polls in memory at scale, loss should not exceed 1 poll cycle. Max loss would be 10K items per metric family.
What happens to data transfer object (DTO) files?	If the hardware failure is the disk, all files are lost. Otherwise, whole DTO files are consumed when the hardware is restarted after repair. Incomplete files are discarded.	Whole DTO files are processed and partially written DTO files are discarded. A DTO file is 1 metric family over 1 poll cycle.	Whole DTO files are processed and partially written DTO files are discarded. The data aggregator attempts to shut down gracefully and close any DTO file in flight.	Whole DTO files are processed and partially written DTO files are discarded. A DTO file is 1 metric family over 1 poll cycle.
What happens with the ActiveMQ Broker?	For 600-MB cache in memory and an average message size of 1.3K, approximately 470K messages could be lost.	For 600-MB cache in memory and an average message size of 1.3K, approximately 470K messages could be lost.	For 600-MB cache in memory and an average message size of 1.3K, approximately 470K messages could be lost.	For 600-MB cache in memory and an average message size of 1.3K, approximately 470K messages could be lost.
What happens with thresholding?	Data loss does not exceed 1 poll cycle.	Data loss does not exceed 1 poll cycle.	Data loss does not exceed 1 poll cycle.	Data loss does not exceed 1 poll cycle.

Configure the Failover Settings

During failover, the inactive data aggregator has 45 minutes to start by default. If the data aggregator does not start within 45 minutes, the fault-tolerant environment tries to start the other data aggregator host. This process repeats for each host every 45 minutes until one of the hosts start.

We recommend that you observe how much time passes between when you issue the command to start the data aggregator REST service and when the service is available. Adjust the `startwait` parameter as appropriate before configuring fault tolerance.

> **WARNING**
>
> Ensure that you configure enough time. Do not set the configurable start time to less than 45 minutes. A start time that is too low can result in data loss or system malfunction.

If it always takes longer than 20 to 30 minutes to start a data aggregator, the hardware might be under resourced. If the hardware is under resourced, DX NetOps Performance Management stops functioning.
For more information about the sizing requirements, see the DX NetOps Performance Management Sizing Tool.

A configurable failover wait time is set to 5 minutes by default. Failover only happens when the active data aggregator is unresponsive to the fault tolerance heartbeat for longer than the configure time (default: 5 minutes). If you have limited

network availability with periodical network outages or system thrashing that may last several minutes, you can increase the failover wait time.

WARNING
To avoid data corruption or data loss, do not set the configurable failover wait time to less than 5 minutes.

Follow these steps:

1. Edit the `config.json` file in the *`Data_Aggregator_Install_Directory`*`/consul-ext/conf/` directory.
2. Edit the `startime` and `failwait` parameters (**s** second, **m** minute, **h** hour).
3. Save your changes.

Configure a Fault-Tolerant Environment

When you first install or upgrade the DX NetOps Performance Management components to the 3.5 release or higher, you are prompted to configure a fault-tolerant environment. After the initial installation or upgrade to a fault-tolerant environment, the responses to the fault-tolerant environment prompts are saved and the prompts do not appear during future upgrades of the fault-tolerant environment. A fault-tolerant environment requires a new shared directory (example: `/DASharedRepo`) to help limit data loss. The shared drive stores customized metric families, DTO files, and the ActiveMQ Kaha database. When a hardware failure or network issue occurs, the newly-active data aggregator accesses the shared drive. The data aggregator picks up where the now inactive data aggregator left off. The user ID that the shared drive is created with must be synced to both of the data aggregators. Then both data aggregators have read and write permissions to that directory.

Follow these steps:

1. Follow the installation or upgrade procedure for the data repository:
   - Install the Data Repository
   - Upgrade the Data Repository
2. Ensure that you have a new shared data directory (example: `/DASharedRepo`) and that the same user ID is shared between data aggregator hosts. Data from the active data aggregator is stored in this directory.

   NOTE
   If you are using NFS, only NFSv4 and higher is supported because of the ActiveMQ Kaha locking requirements.

   WARNING
   The shared data directory must be accessible at all times. If the shared data directory is down and is inaccessible, no data is loaded and data loss occurs.
3. Install the proxy server.
   For more information, see Install or Uninstall the Proxy Server.
4. Follow the installation or upgrade procedure for the active data aggregator:
   - Install the Data Aggregator
   - Upgrade the Data Aggregator

   As you proceed through the data aggregator install or upgrade, you are prompted about configuring fault tolerance. In a fault-tolerant environment, the database user credentials for both data aggregators must match.
5. Complete the following prompts:

   WARNING
   The entries to the following prompts must match for both data aggregators.

   - **Configure Data Aggregator For Fault Tolerance**
     Specify 2 to configure fault tolerance.
     Default: 1

NOTE
The default is for a non-fault tolerant environment.

- **Data Aggregator Proxy Host**
  Specify the host name/IP address of the proxy server.

  NOTE
  You can specify only an IPv4 address.
- **Consul HTTP port:**
  Specify the port for communication with Consul.
  Default: 8500
- **Choose host IP address for Consul**

  NOTE
  This prompt appears only when multiple public IP addresses are configured.

  Specify the bind address that the Consul agents use to communicate with each other. The Consul agents include the proxy host and the data aggregators in the cluster. If prompted for an address, specify an address that the other two hosts in the Consul cluster can reach.

6. Install the secondary inactive data aggregator.
   One of the two available data aggregators becomes the active data aggregator. The other data aggregator is available for failover.
7. Follow the installation or upgrade procedure for each data collector:
   - Install the Data Collectors
   - Upgrade the Data Collectors

   As you proceed through the data collector install or upgrade, you are prompted for a failover location for fault tolerance. The Data Collector installer prompts for the inactive data aggregator host if fault tolerance is configured.
8. Follow the installation or upgrade procedure for NetOps Portal:
   - Install Performance Center
   - Upgrade Performance Center

   As you proceed through the NetOps Portal upgrade, you are prompted about fault tolerance. Follow the prompts to migrate the data aggregator data source from the original data aggregator to the proxy host.

Verify Communication Ports

Open the following ports to allow communications to function properly in a fault-tolerant environment:

- **TCP 8300**
  In a fault-tolerant environment, enables communication between the proxy server and the data aggregators.
- **TCP/UDP 8301**
  In a fault-tolerant environment, enables LAN communication between the proxy server and the data aggregators.
- **TCP 8500**
  In a fault-tolerant environment, enables communication between the proxy server and the data aggregators to the HTTP API.

Verify the Fault-Tolerant Environment Configuration

After you install each data aggregator and add it as a data source, the **System Status** page provides the overall health status of the data aggregators.
For more information about how to view this page, see View the Health of the System.

Next Steps

After setting up your environment, enable the fault-tolerant data aggregators to use HTTPS.

For more information, see Enable Fault Tolerant Data Aggregators to use HTTPS.

Install or Uninstall the Proxy Server

The proxy server sends traffic from NetOps Portal to the active data aggregator. The proxy server works as the third node of the service management cluster for fault tolerance. The service management cluster includes the proxy server, the active data aggregator, and the inactive data aggregator.
For more information about fault tolerance, see Fault Tolerance.

In this article:

(Optional) Configure the Sudo User Account for the Proxy Server

If you do not have root access to install and run the proxy server, configure the sudo user account.

Follow these steps:

1. Locate the `/etc/sudoers` file on the proxy server host.
2. Add a command alias with the following permissions to the file:
   ```
   Cmnd_Alias CA_DAPROXY = /tmp/installDAProxy.bin,/sbin/service daproxy *,/sbin/service
    consul *,/opt/CA/daproxy/Uninstall/Uninstall
    ## Allows the daproxy user to manage the proxy server
     dasudouser_name ALL = CA_DAPROXY
   ```
   This command alias details the commands that the sudo user must be able to run.

The sudo user account for the proxy server is configured.

Install the Proxy Server

Before you install and configure the fault-tolerant data aggregators, install the proxy server.

Follow these steps:

1. Log in to the proxy server host as the root or sudo user.
2. Copy the `installDAProxy.bin` file to the `/tmp` directory.
3. Change to the `/tmp` directory:
   ```
   cd /tmp
   ```
4. Change permissions for the installation file:
   ```
   chmod a+x installDAProxy.bin
   ```
5. Run the console installation using one of the following options:
   - If you have root access to the proxy sever, run the installation by issuing the following command:
     ```
     ./installDAProxy.bin -i console
     ```
   - If you have configured the sudo user account for the proxy server, run the installation by issuing the following command:
     ```
     sudo ./installDAProxy.bin -i console
     ```
6. Follow the instructions in the console.
7. When prompted, specify the following parameters for the proxy server:
   - **Install Set**
     Specify 1 for an installation set including both the proxy server and Consul.
   - **Install Folder**
     Specify where to install the proxy server.

Default: `/opt/CA/daproxy`

- **First DA Host**
  Specify the hostname of the first active Data Aggregator.
- **Second DA Host**
  Specify the hostname of the second inactive Data Aggregator.
- **DA HTTP API Port**
  Specify the port for communication with the Data Aggregator HTTP API.
  Default: 8581
- **Consul Port**
  Specify the port for communication with Consul.
  Default: 8500
- **Choose host IP address for Consul**

  NOTE
  This prompt appears only when multiple public IP addresses are configured.

  Specify the bind address that the Consul agents use to communicate with each other. The Consul agents include the proxy host and both Data Aggregators in the cluster. If prompted for an address, specify an address that the other two hosts in the Consul cluster can reach.

The proxy server is installed.

Uninstall the Proxy Server

If necessary, you can uninstall the proxy server.

Follow these steps:

1. Log in to the proxy server host as the root or sudo user.
2. Issue the following command:
   Proxy_Install_Directory`/daproxy/Uninstall/Uninstall`

The proxy server is uninstalled.

Single Sign-On

Single Sign-On is the authentication scheme for NetOps Portal and all supported data sources. Once they are authenticated to NetOps Portal, users can navigate the console and registered data sources without signing in again.

Enabling the navigation of multiple product interfaces ensures a seamless drilldown experience for operators analyzing performance and status data. For example, if a user logs in to NetOps Portal and follows a drilldown path to the data source interface, that user does not log in again.

NetOps Portal uses a distributed architecture. The Single Sign-On website is automatically installed on every server where a supported data source or NetOps Portal is installed. The distributed architecture lets users log in to data source products by logging in to the servers where these products are running.

Authentication and Security

Single Sign-On provides authentication services to NetOps Portal and supported data sources. It also supports external authentication schemes, such as LDAP and SAML 2.0. This support lets you integrate NetOps Portal and other CA data source products into the same authentication scheme, enterprise-wide.

When a user is required to enter a username and password, the Single Sign-On security auditing feature logs information about who is logging in, and at what time of day. On Linux servers, the log is saved in the following location:

```
[InstallationDirectory]/PerformanceCenter/sso/logs
```

Authentication Methods

The Single Sign-On login page supports user authentication in CA NetOps Portal and in the data source products. Single Sign-On supports the following authentication methods:

- Product authentication, which is based on user accounts
- LDAP
- Security Assertion Markup Language (SAML) 2.0

The CA NetOps Portal administrator can modify settings for an individual instance of Single Sign-On. For example, you can set up LDAP authentication in Single Sign-On. You can also configure optional encryption with Secure Sockets Layer (SSL) or change the default virtual directory.

> **NOTE**
> As a result of the distributed architecture, any updates to the Single Sign-On website affect only those data source products that are running on the same server.

Single Sign-On Configuration Tool

The Single Sign-On Configuration Tool is a command-line application. The application lets administrators adjust the settings for the Single Sign-On website and the associated CA data source products.

> **NOTE**
> The 'Remote Value' option in the Configuration Tool propagates the settings to each registered data source. Use the 'Local Override' option to override the propagated settings on the local server.

The Single Sign-On Configuration Tool was designed to run on Linux systems. However, you can also deploy it on the Windows servers where data sources are installed. If you launch the Configuration Tool from a Windows server, log in as an Administrator on that server.

Use the Single Sign-On Configuration Tool to perform the following tasks:

- Configure data source products to use LDAP authentication.
  All the LDAP settings for each product are updated using this tool. You can also test the current LDAP configuration to verify settings.
- Configure data source products to use SAML 2.0 authentication.
  In addition to using the Configuration Tool, the administrator must also take some steps on the Identity Provider to set up SAML 2.0 authentication.
- Update the Single Sign-On virtual directory that each product references.
  If you added an encryption scheme or you changed the Single Sign-On virtual directory, use this tool to synchronize the data source products. For example, data sources on the modified server need instructions on where to redirect users who do not successfully authenticate.
- Enable communications among servers running CA software products using HTTPS.
  This change affects the Single Sign-On URL scheme and port. The Single Sign-On Configuration Tool lets administrators easily update these values in all the necessary data source products.
- Enable FIPS-compliant encryption and hashing algorithms (where applicable).
  The Single Sign-On configuration tool configures the DX NetOps Performance Management to use FIPS-compliant encryption and hashing algorithms. For more information, see FIPS-Compliant Encryption.

  > **WARNING**
  >
  > DX NetOps Performance Management is not fully FIPS-compliant. This feature is for FIPS-compliant encryption only and does not meet full FIPS compliance.

Data Source Support

CA Single Sign-On supports the following data sources:

- Data Aggregator
- CA Network Flow Analysis
- CA Application Delivery Analysis
- CA UC Monitor

The Single Sign-On Configuration Tool was designed to run on Linux systems. However, you can also deploy it on the Windows servers where data sources are installed. If you launch the Configuration Tool from a Windows server, log in as an Administrator on that server.

You can also run the Configuration Tool on Linux and send configuration instructions to data sources running on Windows by using the Remote Value option.

The Configuration Tool is installed in the following directory location on Linux:

```
[InstallationDirectory]/CA/PerformanceCenter
```

On Windows servers where the data sources are installed, the Configuration Tool is installed in the following directory:

```
[InstallationDirectory]\Portal\SSO\bin\SsoConfig.exe
```

Set Up LDAP Authentication

Single Sign-On provides Lightweight Directory Access Protocol (LDAP) integration, which lets users authenticate to an LDAP server in your environment. Once authenticated, they are mapped to a predefined or a custom user account that the administrator specifies.

The Single Sign-On Configuration Tool lets you precisely specify how the Single Sign-On server connects to the LDAP server. You can also map individual NetOps Portal users to the user accounts that support their workflow while protecting sensitive data.

> **NOTE**
>
> : Changes made in the Single Sign-On Configuration Tool only affect newly created LDAP users. They do not apply to existing LDAP users registered within NetOps Portal.

The LDAP parameters available in the Single Sign-On Configuration Tool let you integrate DX NetOps Performance Management and all registered data sources into an existing authentication scheme. For example, the LDAP server can authorize groups of users who are mapped to a single custom user account in NetOps Portal. The actual account names and LDAP groups can be extensively customized. Search scope parameters let you determine how the directory search is conducted. And you can select the user account properties that are considered when validating users.

Enable LDAP Authentication with No Authentication Mechanism

Use the Single Sign-On Configuration Tool to instruct registered data sources to use the same LDAP scheme to authenticate users. The Single Sign-On Configuration Tool lets you supply parameters that enable the CA server to connect securely to the LDAP server. Using the Configuration Tool, you can also associate users in the LDAP catalog with either predefined or custom user accounts in CA NetOps Portal.

The steps to take to enable LDAP authentication are slightly different if you are using an authentication mechanism such as GSSAPI. Without an authentication mechanism, you must use a service account to bind to the LDAP server. This account requires read and search access to the LDAP server. You must supply the full DN (distinguished name) of the connection user, and you must also enable the User Bind parameter.

Single Sign-On binds to the LDAP server using the credentials that you supply for the Connection User and Connection Password parameters. Then Single Sign-On performs a directory search that is based on the string that you supply for the Search String parameter. The search results include the DN of the user. Single Sign-On performs a second bind to the LDAP server using this DN and password.

> **WARNING**
> In cases where no authentication mechanism is used, we strongly recommend establishing an SSL connection to the LDAP server. Otherwise, the passwords are transmitted to the LDAP server in cleartext.

Follow these steps:

1. Log in to the server where CA NetOps Portal or a CA data source product is installed.
   Log in as root or with the 'sudo' command.
2. Launch the Single Sign-On Configuration Tool by running the './SsoConfig' command in the following directory:
   `InstallationDirectory/CA/PerformanceCenter`
   You are prompted to select an option. The available options correspond to CA applications running on the local server.
3. Use the following commands as needed while you are selecting settings:
   - q (quit)
   - b (go back to the previous menu)
   - u (update)
   - r (reset)
4. Enter 1 to configure CA NetOps Portal.
   You are prompted to select an option.
5. Enter 1 for LDAP Authentication.
   You are prompted to specify the priority.
   The Priority parameter only applies to CA NetOps Portal.
6. Enter one of the following options:
   - **1. Remote Value**
     These settings are propagated to all other CA products and data sources that are registered to this instance of CA NetOps Portal. This includes the Event Manager in CA NetOps Portal, which embeds the URL of CA NetOps Portal. CA NetOps Portal uses Remote Value settings only if a corresponding Local Override value is not present.
   - **2. Local Override**
     Overrides a setting on this CA NetOps Portal instance, which does not propagate to other CA products and data sources (including Event Manager) registered to this instance of CA NetOps Portal. Local Override takes precedence over both the Remote Value and default settings.

   You are prompted to select a property to configure.

   > **NOTE**
   >
   > Configure the scheme or port using Remove Value to include the correct CAPC URL in threshold event email messages.
7. Enter one or more of the following properties. When prompted, enter u to update the value and supply a new value:
   - **1. Connection User**
     Defines the user ID (in this case, the user ID of the service account) that the login server uses to connect to the LDAP server. This LDAP username is used to bind to the server.

     > **WARNING**
     > A service account with read and search access to the LDAP server is required for this parameter if you are not using an authentication mechanism, such as GSSAPI.
   - **2. Connection Password**
     Defines the password for the login server to use to connect to the LDAP server.
     Example: If the login server uses a fixed account, enter text like the following example:

```
SomePassword
```

- **3. Search Domain**
  Identifies the LDAP server and port to which CA Single Sign-On connects. Also identifies the location in the directory tree where the search looks for users when verifying user account credentials. If you do not also supply a port number after the server in the string, Port 389 is used.
  Use the following format for the search domain:

  ```
  LDAP://ldap_server:port/path_to_search
  ```

  NOTE
  The search path is *required*.
- **4. Search String**
  Specifies the criteria that are used to locate the correct record for the user. Works with the Search Scope parameter. If only a subset of LDAP users is allowed to log in, the search string can be used to search the record for multiple properties. The value for this parameter can include any valid LDAP search criterion.
  Example:

  ```
  (saMAccountName={0})
  ```
- **5. Search Scope**
  Specifies the criteria that are used to locate the correct record for the user. Used with the Search String parameter. Determines the scope of the search that the LDAP server performs for the user account. Type one of the following values:
  - **onelevel**
    Includes the current directory in the search. Matches objects in the current directory and prevents unexpected matches deeper in the directory.
  - **subtree**
    Includes all subdirectories in the search. Recommended for most installations.
  - **base**
    Limits the search to the base object.
- **6. User Bind**
  Specifies whether to do an additional authentication step (bind) using the distinguished name (DN) and password of the user to validate the supplied credentials.

  WARNING
  This parameter must be set to Enabled if you entered a service account in Steps 1 and 2.

  Default: Disabled.
- **7. Encryption**
  Specifies the authentication mechanism to use when binding a second time to the LDAP server.
  Default: Simple.
  Accepted Values: Simple, GSSAPI, DIGEST-MD5.
- **8. Account User**
  Specifies the CA NetOps Portal default account to which to map validated LDAP users who lack a group membership. Works with the Account Password parameter. If a valid user does not match any group definitions, the user is logged in with the default user ID specified for this parameter.
  To allow all users to log in with their own username, enter:
  - {saMAccountName}
  - {saMAccountName} or {CN}

  NOTE
  The Account User parameter corresponds to a field from the directory entry for this user. Typically, the value matches your search filter.
- **9. Account User Default Clone**
  Specifies a user account to clone if validated LDAP users are members of a group that is not specified for the Group parameter.

Example: Enter 'user' if you want such users to have minimal privileges.

NOTE

An existing user account is required.

- **10. Group**
  Lets you determine the default account handling for selected user accounts or groups of accounts.
  Example: To enable all members of a group to log in using an administrator account, enter:
  ```
  <LDAPGroups><Group searchTag="memberOf" searchString="CN=SEC_All Employees,CN=Users,DC=company,DC=local"
   user="{saMAccountName}" passwd="" userClone="admin"/></LDAPGroups>
  ```
- **11. Timeout**
  Specifies the amount of time that CA NetOps Portal waits while making authorization checks to the LDAP server. When the authorization check times out, users who try to log in are denied access. To view the errors, open the SSOService.log file. The default timeout is 10000.

8. Verify that the LDAP Status is set to Enabled. If the LDAP status is set to Disabled, then authentication uses the internal NetOps Portal user database.
9. Enter q to quit.
   The Configuration Tool closes.

Example Configuration

1. Connection User: CN=********,OU=Role-Based,OU=North America,DC=ca,DC=com [the full DN of the service account]
2. Connection Password: ******* [the password of the service account]
3. Search Domain: LDAP://******.ca.com/DC=ca,DC=com
4. Search String: (sAMAccountName={0})
5. Search Scope: Subtree
6. User Bind: Enabled
7. Encryption: false
8. Account User: {sAMAccountName}
9. Account User Default Clone: user
10. Group: 'All Employees'
11. Krb5ConfigFile: krb5.conf

Encrypt the Connection to the LDAP Server Using GSSAPI

CA Single Sign-On supports encrypted connections using DIGEST-MD5 or GSSAPI. When you use an encrypted connection to the directory server, you do not have to use a service account to bind to the LDAP server (the UserBind parameter that you set in the Single Sign-On Configuration Tool).

To use GSSAPI for encryption, you must change some settings in a configuration file.

Follow these steps:

1. Log in to the server where CA NetOps Portal or a CA data source product is installed.

   NOTE

   If the settings for GSSAPI LDAP are pushed to a data source, configure the **krb** file on the data source system.

   Log in as root or with the 'sudo' command.
2. Change to the following directory:
   ```
   Installation Dir/webapps/sso/Configuration/
   ```
3. Open the krb5.conf file in that directory for editing.

4. Set the following required parameters:

```
[libdefaults]
        default_realm = CA.COM
[realms]
        CA.COM  = {
                kdc = EXAMPLE.CA.COM
                default_domain = CA.COM
}

[domain_realm]
       .CA.COM = CA.COM
}
```

   where:
   - **[libdefaults]**
     Contains default values for the Kerberos V5 library.
   - **default_realm**
     Maps subdomains and domain names to Kerberos realm names. Lets programs determine the realm for a host, based on its fully qualified domain name. In this example, the default realm is CA.COM.
   - **realms**
     Contains information about Kerberos realm names, which describe the location of Kerberos servers and include other realm-specific information.
   - **kdc**
     Is the Kerberos key distribution center to support authentication services. For example, EXAMPLE.CA.COM.
   - **default_domain**
     Is the default IP domain. For example, CA.COM.

     NOTE
     Your Active Directory or LDAP Administrator can probably provide you with a krb5.conf file or help you to create one.
5. Save your changes.
6. Now follow the steps in Enable LDAP Authentication Using an Encryption Mechanism to configure LDAP authentication with CA Single Sign-On.

Enable LDAP Authentication Using an Encryption Mechanism

Use the Single Sign-On Configuration Tool to instruct registered data sources to use the same LDAP scheme to authenticate users. The Single Sign-On Configuration Tool lets you supply parameters that enable the CA server to connect securely to the LDAP server. When you use Digest-MD5 or GSSAPI to encrypt the connection to the LDAP server, a single bind operation -- as the user you specify -- occurs.

Using the Configuration Tool, you can also associate users in the LDAP catalog with either predefined or custom user accounts in CA NetOps Portal.

Follow these steps:

1. Log in to the server where NetOps Portal or a CA data source product is installed.
   Log in as root or with the 'sudo' command.
2. Launch the Single Sign-On Configuration Tool by running the './SsoConfig' command in the following directory:

```
InstallationDirectory/CA/PerformanceCenter
```

   You are prompted to select an option. The available options correspond to CA applications running on the local server.
3. Use the following commands as needed while you are selecting settings:

- q (quit)
- b (go back to the previous menu)
- u (update)
- r (reset)

4. Enter 1 to configure CA NetOps Portal.
   You are prompted to select an option.
5. Enter 1 for LDAP Authentication.
   You are prompted to specify the priority.
   The Priority parameter only applies to CA NetOps Portal.
6. Enter one of the following options:
   - **1. Remote Value**
     These settings are propagated to all other CA products and data sources that are registered to this instance of CA NetOps Portal. This includes the Event Manager in CA NetOps Portal, which embeds the URL of CA NetOps Portal. CA NetOps Portal uses Remote Value settings only if a corresponding Local Override value is not present.
   - **2. Local Override**
     Overrides a setting on this CA NetOps Portal instance, which does not propagate to other CA products and data sources (including Event Manager) registered to this instance of CA NetOps Portal. Local Override takes precedence over both the Remote Value and default settings.

   You are prompted to select a property to configure.

   NOTE

   Configure the scheme or port using Remove Value to include the correct CAPC URL in threshold event email messages.
7. Enter one or more of the following properties. When prompted, enter u to update the value and supply a new value:
   - **1. Connection User**
     Defines the user ID that the login server uses to connect to the LDAP server. This LDAP user name is used to bind to the server. A service account is not typically required for a connection that uses an authentication mechanism, such as GSSAPI.
     Example: If the login server uses a fixed account, enter text with the following syntax:
     ```
     CN=The User,cn=Users,dc=domain,dc=com
     ```
     Or you can enter the following value because the connection is using an authentication mechanism:
     ```
     {0}
     ```
     Complex configurations need the user principal name to identify the user. Supply '{0}' and use their email address as the domain name. For example:
     ```
     {0}@domain.com
     ```
     The LDAP server typically does not require a full DN for an encrypted connection.

     NOTE
     For security reasons, do not make the connection user a static account. The LDAP authentication only checks the password when binding to the server. If you use a static account, any user that exists in the LDAP tree is able to log in with any password.
   - **2. Connection Password**
     Defines the password for the login server to use to connect to the LDAP server.
     Example: If the login server uses a fixed account, enter text like the following example:
     ```
     SomePassword
     ```
     Or you can enter the following value because the connection is using an authentication mechanism:
     ```
     {1}
     ```
   - **3. Search Domain**

Identifies the LDAP server and port to which CA Single Sign-On connects. Also identifies the location in the directory tree where the search looks for users when verifying user account credentials. If you do not also supply a port number after the server in the string, Port 389 is used.
Use the following format for the search domain:

```
LDAP://ldap_server:port/path_to_search
```

NOTE
The search path is *required*.

- **4. Search String**
  Specifies the criteria that are used to locate the correct user in the directory. Works with the Search Scope parameter. If only a subset of LDAP users is allowed to log in, the search string can be used to search a record for multiple properties. The value for this parameter can include any valid LDAP search criterion.
  Example:

```
(saMAccountName={0})
```

- **5. Search Scope**
  Specifies the criteria that are used to locate the correct record for the user. Used with the Search String parameter. Determines the scope of the search that the LDAP server performs for the user account. Type one of the following values:
  - **onelevel**
    Includes the current directory in the search. Matches objects in the current directory and prevents unexpected matches deeper in the directory.
  - **subtree**
    Includes all subdirectories in the search. Recommended for most installations.
  - **base**
    Limits the search to the base object.
- **6. User Bind**
  Specifies whether to do an additional authentication step (bind) using the distinguished name (DN) and password of the user to validate the supplied credentials.
  Default: Disabled. This value is acceptable with an encrypted connection.
- **7. Encryption**
  Specifies the authentication mechanism to use when binding again to the LDAP server.
  In this case (that is, using an authentication mechanism), enter 'GSSAPI' or 'DIGEST-MD5', based on the mechanisms of your LDAP server.
  Default: Simple.
  Accepted Values: Simple, GSSAPI, DIGEST-MD5.
- **8. Account User**
  Specifies the CA NetOps Portal default account to which to map validated LDAP users who lack a group membership. Works with the Account Password parameter. If a valid user does not match any group definitions, the user is logged in with the default user ID specified for this parameter.
  To allow all users to log in with their own username, enter:
  - {saMAccountName}
  - {saMAccountName} or {CN}

  NOTE
  The Account User parameter corresponds to a field from the directory entry for this user. Typically, the value matches your search filter.
- **9. Account User Default Clone**
  Specifies a user account to clone if validated LDAP users are members of a group that is not specified for the Groups parameter.
  Example: Enter 'user' if you want such users to have minimal privileges.

NOTE

An existing user account is required.

- **10. Group**
  Lets you determine the default account handling for selected user accounts or groups of accounts.
  Example: To enable all members of a group to log in using an administrator account, enter:
  ```
  <LDAPGroups><Group searchTag="memberOf" searchString="CN=SEC_All Employees,CN=Users,DC=company,DC=local"
   user="{saMAccountName}" passwd="" userClone="admin"/></LDAPGroups>
  ```
- **11. Krb5ConfigFile**
  Configure the **krb** file location.

NOTE

If another data source, such as Network Flow Analysis or CA Application Delivery Analysis, is getting the LDAP settings, configure the **krb** file location on the data source. Use a local override with the Single Sign-On Configuration Tool on the data source system.

8. Verify that the LDAP Status is set to Enabled. If the LDAP status is set to Disabled, then authentication uses the internal NetOps Portal user database.
9. Enter q to quit.
   The Configuration Tool closes.

Example Configuration

SSO Configuration/CA NetOps Portal/LDAP Authentication/Remote Value:

1. Connection User: {0}
2. Connection Password: {1}
3. Search Domain: LDAP://******.ca.com/DC=ca,DC=com
4. Search String: (sAMAccountName={0})
5. Search Scope: Subtree
6. User Bind: Disabled
7. Encryption: DIGEST-MD5
8. Account User: {sAMAccountName}
9. Account User Default Clone: user
10. Group: 'All Employees'
11. Krb5ConfigFile: /opt/CA/PerformanceCenter/sso/webapps/sso/configuration/krb5.conf

Enable LDAPS Authentication

Use the Single Sign-On Configuration Tool to instruct registered data sources to use LDAP over SSL (LDAPS) for secure user authentication. By default, LDAP traffic is transmitted unsecured. Enable LDAPS by installing a certificate from a certification authority (CA). With CA Single Sign-On, you must import your certificate into the Java trusted keystore.

The Single Sign-On Configuration Tool lets you supply parameters that enable the CA server to connect securely to the LDAP server. Using the Configuration Tool, you can also associate users in the LDAP catalog with either predefined or custom user accounts in CA NetOps Portal.

Follow these steps:

1. Log in to the server where CA NetOps Portal or a CA data source product is installed.
   Log in as root or with the 'sudo' command.
2. Follow the instructions in the topic titled Import the LDAP Certificate to obtain your certificate and import it into the Java keystore.
3. Launch the Single Sign-On Configuration Tool by running the './SsoConfig' command in the following directory:

```
InstallationDirectory/CA/PerformanceCenter
```

You are prompted to select an option. The available options correspond to CA applications running on the local server.

4. Use the following commands as needed while you are selecting settings:
   - q (quit)
   - b (go back to the previous menu)
   - u (update)
   - r (reset)
5. Enter 1 to configure CA NetOps Portal.
   You are prompted to select an option.
6. Enter 1 for LDAP Authentication.
   You are prompted to specify the priority.
   The Priority parameter only applies to CA NetOps Portal.
7. Enter one of the following options:
   - **1. Remote Value**These settings are propagated to all other CA products and data sources that are registered to this instance of CA NetOps Portal. This includes the Event Manager in CA NetOps Portal, which embeds the URL of CA NetOps Portal. CA NetOps Portal uses Remote Value settings only if a corresponding Local Override value is not present.
   - **2. Local Override**
     Overrides a setting on this CA NetOps Portal instance, which does not propagate to other CA products and data sources (including Event Manager) registered to this instance of CA NetOps Portal. Local Override takes precedence over both the Remote Value and default settings.

   You are prompted to select a property to configure.

   NOTE

   Configure the scheme or port using Remove Value to include the correct CAPC URL in threshold event email messages.
8. Enter one or more of the following properties. When prompted, enter u to update the value and supply a new value:
   - **1. Connection User**
     Defines the user ID that the login server uses to connect to the LDAP server. This LDAP user name is used to bind to the server. A service account is not typically required for a connection that uses an authentication mechanism, such as GSSAPI.
     Example: If the login server uses a fixed account, enter text with the following syntax:

     ```
     CN=The User,cn=Users,dc=domain,dc=com
     ```

     Or you can enter the following value because the connection is using an authentication mechanism:

     ```
     {0}
     ```

     Complex configurations need the user principal name to identify the user. Supply '{0}' and use their email address as the domain name. For example:

     ```
     {0}@domain.com
     ```

     The LDAP server typically does not require a full DN for an encrypted connection.

     NOTE
     For security reasons, do not make the connection user a static account. The LDAP authentication only checks the password when binding to the server. If you use a static account, any user that exists in the LDAP tree is able to log in with any password.
   - **2. Connection Password**
     Defines the password for the login server to use to connect to the LDAP server.
     Example: If the login server uses a fixed account, enter text like the following example:

     ```
     SomePassword
     ```

     Or you can enter the following value because the connection is using an authentication mechanism:

```
{1}
```

- **3. Search Domain**
  Identifies the LDAP server and port to which CA Single Sign-On connects. Also identifies the location in the directory tree where the search looks for users when verifying user account credentials. If you do not also supply a port number after the server in the string, Port 389 is used.
  Use the following format for the search domain:

  ```
  LDAPS://ldap_server:port/path_to_search
  ```

  NOTE
  The search path is *required*.

  To establish an SSL connection to the LDAP server, use 636 or another SSL connection port for your LDAP server:

  ```
  LDAPS://LDAP Server:636/OU=Users,OU=North America,DC=ca,DC=com
  ```

- **4. Search String**
  Specifies the criteria that are used to locate the correct user in the directory. Works with the Search Scope parameter. If only a subset of LDAP users is allowed to log in, the search string can be used to search a record for multiple properties. The value for this parameter can include any valid LDAP search criterion.
  Example:

  ```
  (saMAccountName={0})
  ```

- **5. Search Scope**
  Specifies the criteria that are used to locate the correct record for the user. Used with the Search String parameter. Determines the scope of the search that the LDAP server performs for the user account. Type one of the following values:
  - **onelevel**
    Includes the current directory in the search. Matches objects in the current directory and prevents unexpected matches deeper in the directory.
  - **subtree**
    Includes all subdirectories in the search. Recommended for most installations.
  - **base**
    Limits the search to the base object.
- **6. User Bind**
  Specifies whether to do an additional authentication step (bind) using the distinguished name (DN) and password of the user to validate the supplied credentials.
  Default: Disabled. This value is acceptable with an encrypted connection.
- **7. Encryption**
  (Optional) Specifies the authentication mechanism to use when binding again to the LDAP server.
  The default (Simple authentication) is acceptable with LDAPS.
- **8. Account User**
  Specifies the CA NetOps Portal default account to which to map validated LDAP users who lack a group membership. Works with the Account Password parameter. If a valid user does not match any group definitions, the user is logged in with the default user ID specified for this parameter.
  To allow all users to log in with their own username, enter:
  - {saMAccountName}
  - {saMAccountName} or {CN}

  NOTE
  The Account User parameter corresponds to a field from the directory entry for this user. Typically, the value matches your search filter.
- **9. Account User Default Clone**
  Specifies a user account to clone if validated LDAP users are members of a group that is not specified for the Groups parameter.
  Example: Enter 'user' if you want such users to have minimal privileges.

NOTE

: An existing user account is required.

- **10. Group**Lets you determine the default account handling for selected user accounts or groups of accounts.
  Example: To enable all members of a group to log in using an administrator account, enter:
  ```
  <LDAPGroups><Group searchTag="memberOf" searchString="CN=SEC_All Employees,CN=Users,DC=company,DC=local"
   user="{saMAccountName}" passwd="" userClone="admin"/></LDAPGroups>
  ```

9. Enter q to quit.
   The Configuration Tool closes.
10. Restart all NetOps Portal services:
    ```
    service caperfcenter_console restart
    service caperfcenter_devicemanager restart
    service caperfcenter_eventmanager restart
    service caperfcenter_sso restart
    ```

Example Configuration

1. SSO Configuration/CA NetOps Portal/LDAP Authentication/Remote Value
2. Connection User: {0}
3. Connection Password: {1}
4. Search Domain: LDAPS://******.ca.com:636/OU=Users,OU=North America,DC=ca,DC=com
5. Search String: (sAMAccountName={0})
6. Search Scope: Subtree
7. User Bind: Disabled
8. Encryption: Simple
9. Account User: {sAMAccountName}
10. Account User Default Clone: user
11. Group: 'All Employees'
12. Krb5ConfigFile: krb5.conf

Import the LDAP Certificate

To run with LDAPS, import an LDAP certificate into the Java keystore.

To generate an SSL certificate, use the keytool command. Import a certificate from a Certificate Authority and install it in the keystore.

Follow these steps:

1. Obtain the certificate from the LDAP server administrator.
2. Determine whether the root Certificate Authority certificate is part of the default java trusted authorities:
   ```
   keytool -list -v -keystore InstallDirectory/jre/lib/security/cacerts -
   storepass cacertspassword
   ```

   - ***cacertspassword***
     Specify the password for the Certificate Authority keystore.

     NOTE
     The default password for the Certificate Authority keystore is **changeit**.
3. Import the LDAP, intermediate, or root certificates into the java trusted certificate keystore using the following steps for each certificate:

```
keytool -importcert -keystore InstallDirectory/jre/lib/security/cacerts -
storepass cacertspassword -alias alias_name -file filename.cer
```

- ***cacertspassword***
  Specify the password for the Certificate Authority keystore.

  NOTE
  The default password for the Certificate Authority keystore is **changeit**.
- ***alias_name***
  Specify an alias that can be used to refer to the keystore entry that is created for the root or intermediate certificate.
- ***filename* .cer**
  Specify the file to which the certificate is exported. We recommend using a full pathname that does not place the file in the current directory.
  Example: /tmp/capcCert.cer

4. Create a backup of the cacerts file.
5. (Optional) Change the password of the java trusted certificates keystore:
   ```
   keytool -storepasswd -keystore installDirectory/
   jre/lib/security/cacerts
   ```
6. Verify that your imported certificate is available:
   ```
   keytool -list -keystore installDirectory/jre/
   lib/security/cacert
   ```

Validate LDAP Settings

The Single Sign-On Configuration Tool lets you test the LDAP settings that you have supplied. You can verify that LDAP authentication is set up correctly. An LDAP test script prompts you to specify a username and password combination to test, using the current settings for LDAP authentication. If you have not already used the Configuration Tool to change LDAP authentication settings, the defaults are used.

Follow these steps:

1. Log in to the server where CA NetOps Portal or a supported data source product is installed.
   Log in as root or with the 'sudo' command.
2. Launch the Single Sign-On Configuration Tool by running the './SsoConfig' command in the following directory:
   ```
   InstallationDirectory/CA/PerformanceCenter
   ```
   You are prompted to select an option. The available options correspond to CA applications running on the local server.
3. Use the following commands as needed while you are selecting settings:
   - q (quit)
   - b (go back to the previous menu)
   - u (update)
   - r (reset)
4. Enter 1 to configure CA NetOps Portal.
   You are prompted to select an option.
5. Enter 5 for the Test LDAP option.
   The prompt asks you to enter a username.
6. Enter a username and a password that you know can authenticate using LDAP.
   Single Sign-On attempts to use the parameters you supplied when you set up LDAP authentication to connect to the LDAP server and validate the user account. If the test succeeds, numerous steps are logged.
   A message reports whether the authentication succeeded or failed.
7. Enter q to quit.

Set Up SAML 2.0 Support

The Security Assertion Markup Language (SAML) is a security protocol that is based on XML. SAML allows the exchange of security assertions about a subject, such as a person or a computer, that is requesting access to a secure domain. Assertions include whether the subject can access certain resources, and whether an external data source, such as a policy store, is used.

A typical use of SAML-based authentication is in a federated environment, such as cloud-based services that require an extra layer of security in the corporate network. But any SAML implementation involves at least three component roles:

- **Relying Party**
  Uses identity information that is stored on another server to let authorized users gain access to a system. Also referred to as the 'service provider.' CA NetOps Portal has this role when Single Sign-On is configured to use SAML for authentication.
- **Asserting Party**
  Stores identity or security information and provides it when requested for authentication purposes. The SAML term for this component is the *Identity Provider* or *IdP*. The CA SiteMinder server has this role, for example.
- **Subject**
  Is the user (or computer) associated with the identity information that is stored by the IdP.

SAML 2.0 Support in Single Sign-On

CA Single Sign-On supports authentication with Security Assertion Markup Language (SAML) 2.0. A Single Sign-On service can accept and decode SAML 2.0 tokens and can present them to authentication agents that conform to the SAML standard.

Single Sign-On support for SAML 2.0 includes support for single logout. With this support, a user who is logged in to multiple user interfaces can log out of all of them simultaneously. For example, a user who logs in to CA NetOps Portal and later drills down into flow data in CA Network Flow Analysis can log out of one interface and be logged out of the other interface automatically.

Single Sign-On uses a standards-based SAML 2.0 library. As a result, it potentially supports many more products that rely on the SAML 2.0 standards. However, the following CA products are the only Identity Providers that we have tested with CA Single Sign-On:

- CA SiteMinder Federation Manager
- CA Arcot A-OK™ On-Demand

In a SAML environment, you can select from multiple authentication methods. CA NetOps Portal users can log in using the typical ('Product') authentication method in Single Sign-On, or they can use a SAML token. The Product method is enabled by default for all active user accounts. Users access the CA NetOps Portal user interface using the standard URL for CA Single Sign-On.

To let users authenticate using SAML 2.0, the administrator must change some Single Sign-On settings using the Configuration Tool. The administrator must also enable External Authentication for all user accounts, and for all registered data sources that support SAML 2.0.

Not all CA data source products support SAML 2.0. If you configure SAML 2.0 for external authentication in Single Sign-On and register a data source that lacks SAML support, CA NetOps Portal users must reauthenticate when they drill down into that data source.

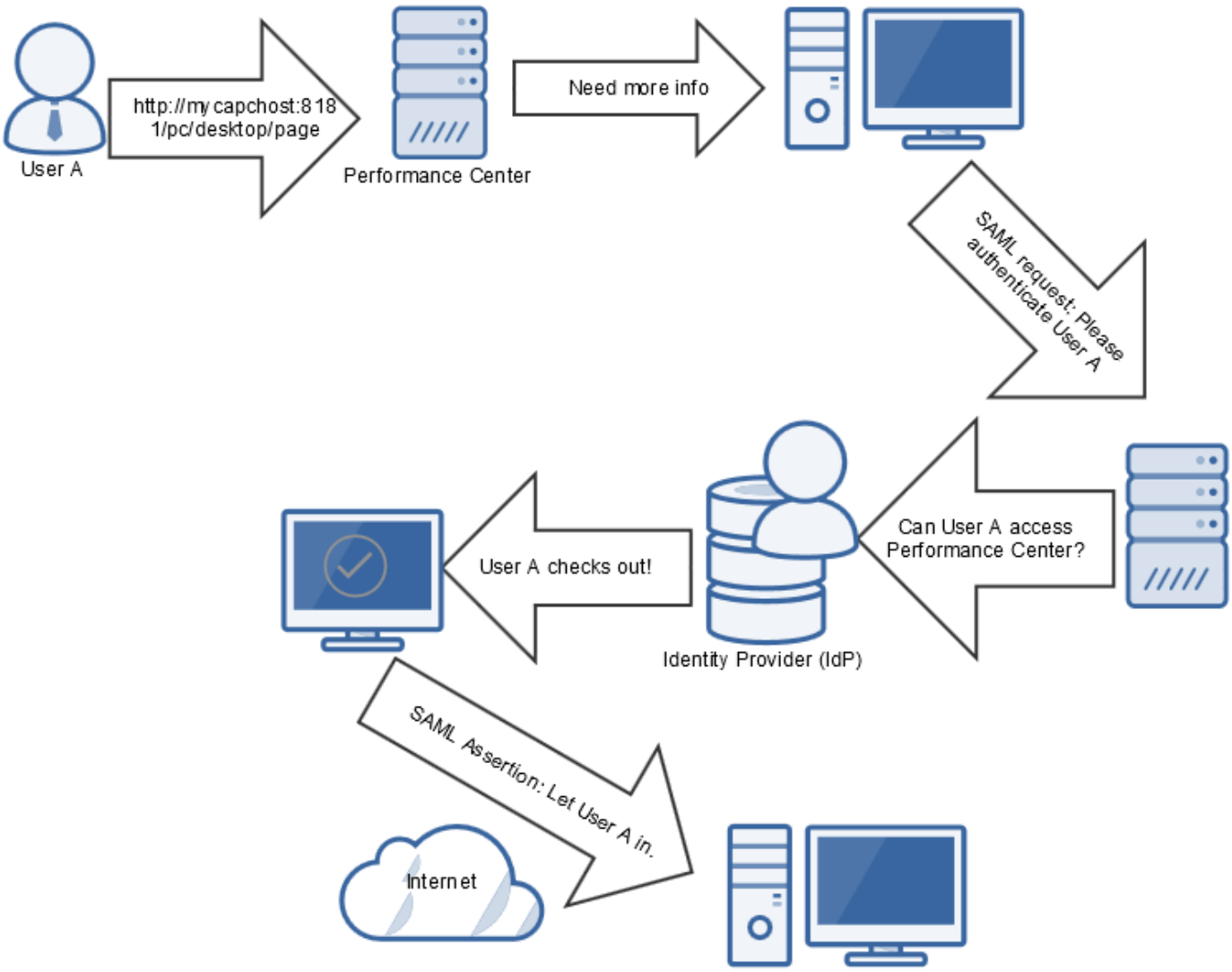
About Single Sign-On Support for SAML 2.0

The NetOps Portal authentication process that uses Single Sign-On differs from authentication that takes advantage of SAML 2.0 support. With SAML 2.0 authentication, users do not see the CA NetOps Portal Login page. They are instead

redirected to an interface that the IdP provides. For all other supported authentication methods, Single Sign-On provides the login page.

The following diagram illustrates the SAML 2.0 authentication process with Single Sign-On, CA NetOps Portal, and an IdP that supports the SAML 2.0 standard, such as CA SiteMinder:

Figure 63: SSO



The following generic process describes how CA NetOps Portal supports SAML 2.0 authentication. Implementation-specific options, such as digitally signed certificates and transport binding, have been omitted:

1. A user attempts to access CA NetOps Portal, by navigating to http://mycapchost:8181/pc/desktop/page, for example.
2. CA NetOps Portal responds with a SAML request for authentication from the Identity Provider (IdP).
3. The browser processes the request and contacts the authentication software running on the IdP server.
4. The IdP determines whether the user has an existing logon security context -- whether the user is already logged on.
5. If the user is not logged on, the IdP authenticates the user with an implementation-specific method.
   For example, the IdP might interact with the browser to challenge the user to provide credentials. This stage of the authentication is irrelevant to CA Single Sign-On.

6. The IdP builds and sends a SAML assertion representing the user's logon security context to the browser.
   The assertion includes a required attribute, subjectNameId, and an optional attribute, ClonedUser.
   The value of subjectNameId corresponds to the authorized user.
   You can include the name of the cloned user account in the assertion. This attribute defines the user account to which authorized SAML users are mapped.
7. The browser sends the SAML assertion to CA NetOps Portal.
8. CA NetOps Portal obtains the assertion and processes it.
9. If the assertion is valid, CA NetOps Portal establishes a session for the user. The browser redirects to the target page, the Home dashboard page for the user.

How to Set Up SAML Authentication

To enable SAML 2.0 authentication in Single Sign-On, the administrator performs the following procedures:

1. Following the guidelines specific to the Identity Provider (IdP), create a metadata file that establishes the agreement between the IdP and Single Sign-On.
   For more information, see Prepare the IdP Agreement.
2. (Optional) Create a properties file to enable digital signatures and encryption for communications between the IdP and servers running CA software.
   For more information, see Preparing the Security Properties File.
3. Use the Single Sign-On Configuration Tool to set parameters for SAML Authentication.
   For more information, see Configure SAML 2.0 Support in Single Sign-On.
4. Set parameters on the IdP server. For example, add all data source product websites that support SAML to the list of trusted sites.
   For more information, see Configure the IdP.
5. Update user accounts in CA NetOps Portal Administration to add an instruction to use external authentication.
   For more information, see Complete SAML 2.0 Setup.

Prepare the IdP Agreement

A metadata file in XML format is required to establish the agreement between the IdP and the Service Provider. In this case, CA NetOps Portal and all registered data sources that support SAML 2.0 require this agreement. The metadata file describes the IdP and contains information about the profiles it supports. This file also contains data about the services that it requires from the Service Provider.

Single Sign-On can import this file to set up the relationship with the IdP.

Some types of IdP, such as CA SiteMinder, provide utilities to help you create these files and export them. Or they create the agreement automatically, based on the parameters you set.

Consult the documentation for your IdP to perform this task.

Preparing the Security Properties File

To use encryption and digital certificates for communications between NetOps Portal and the IdP, a properties file is required. In this file, you specify the certificate to use for signing and encryption and other parameters to enable the encryption.

The SAML properties file is saved in the Single Sign-On configuration directory:

```
/opt/CA/PerformanceCenter/sso/webapps/sso/configuration
```

For example, a file like this is required:

```
/opt/CA/PerformanceCenter/sso/webapps/sso/configuration/
```

```
saml.properties
```

The properties file must include the following parameters:

- Directory location and filename of the signing certificate. For more information, see Set Up SAML Certificates.
- Verification certificate alias and password to access the certificate.
- Hostname of the CA NetOps Portal server.
- Directory location and filename of the agreement that you exported from the IdP.

Here is an example of the syntax:

```
# Location of the certificate used for signing SAML documents
saml.sp.certificate.location=/opt/CA/saml2configuration/[Certificate filename]
saml.sp.certificate.password=[password]
saml.sp.certificate.alias=[alias]

saml.sp.metadata.hostname=[Full Hostname of <npc> server]
saml.sp.metadata.entityId=[Name of the <npc> server without IP domain]
saml.sp.metadata.organizationName=[Name of your organization]
saml.sp.metadata.contactPerson=[First and last name of administrator]
saml.sp.metadata.email=[Email address of contact person]

# Location of the metadata file for the Login Site
saml.idp.metadata.file=/opt/CA/saml2configuration/[Filename].xml
```

Whenever you modify the saml.properties file, export the metadata file (which establishes the agreement with the IdP) again. For more information, see Configure SAML 2.0 Support in Single Sign-On. You must also restart Single Sign-On.

Set Up SAML Certificates

To configure the single sign-on website to use SAML over SSL, obtain and install a private key and an associated public certificate. SAML can be used with either a self-signed certificate or a certificate that a trusted Certificate Authority has signed. The procedures are typically specific to an organization and the policies of its security team. However, these procedures provide some guidance.

Select the appropriate procedure for your situation:

> **NOTE**
> For more information about the keytool command, see the Java documentation on the Oracle website.

Generate and Import a Certificate

To generate an SAML certificate, use the keytool command. Generate a self-signed certificate and install it in the keystore.

Follow these steps:

1. Change the directory:
   ```
   textcd InstallDirectory/PerformanceCenter/sso/webapps/sso/configuration
   ```

   > **NOTE**
   > /opt/CA is the default installation directory.
2. If a keystore file exists, rename the existing keystore file to create a backup of it:
   ```
   textmv InstallDirectory/PerformanceCenter/sso/webapps/sso/configuration/
   keystore InstallDirectory/PerformanceCenter/PerformanceCenter/sso/webapps/sso/
   configuration/keystore.bak
   ```

WARNING
Move the old keystore. If you do not, an error appears in later steps: "Keystore was tampered with, or password was incorrect."

3. Generate a private key and a public, self-signed certificate:
   `keytool -genkeypair -keystore keystore_file.ks -storepass password -keyalg RSA -keysize 2048 -keypass password -alias alias_name`

   Note your entries for the following variables:
   – ***keystore_file.*** **ks**
   Specify the name of the keystore file to create
   – ***password***
   Specify the password for the keystore and self-signed certificate. Specify a secure password.
   – ***alias_name***
   Specify an alias that can be used to refer to the keystore entry that is created for the self-signed certificate.

   NOTE
   N ote: When you are prompted for your first and last name, provide the fully qualified hostname of the server.
4. Proceed through the security prompt questions and confirm your responses.
   Your self-signed SAML certificate is generated and installed in the keystore.
5. Configure the `saml.properties` with the keystore location and filename, keystore password, and alias. For more informarion, see Preparing the Security Properties File.

Convert a Self-Signed Certificate to a Certification Authority SAML Certificate

A self-signed certificate is signed by the same entity whose identity it certifies rather than a Certification Authority. Therefore, a self-signed certificate is not trusted. The following procedure explains how to convert the self-signed certificate to a certificate that a trusted Certification Authority has signed.

Follow these steps:

1. Change the directory:
   `cd InstallDirectory/PerformanceCenter/sso/webapps/sso/configuration`

   NOTE
   /opt/CA is the default installation directory.
2. Export a certificate signature request:
   `keytool -certreq -keystore keystore_file.ks -storepass password -alias alias_name -keypass password -file RequestFileName.csr`

   – ***keystore_file.ks***
   Specify the same keystore file name previously created.
   – ***password***
   Specify the same password when creating the self-signed certificate.
   – ***alias_name***
   Specify the same alias when creating the self-signed certificate.
   – ***RequestFileName*****.csr**
   Specify the path and file name of the exported signature request.
3. Send the resulting file (***RequestFileName*** .csr) to a qualified signing authority with any other requested information.

The Certificate Authority sends you a signed certificate (***SignedCert*** .cer). They might also provide a root Certificate Authority certificate (***RootCA*** .cer) to authenticate the signed certificate.

4. (Optional) Determine whether the root Certificate Authority certificate is part of the default java trusted authorities:

   ```
   keytool -list -v -keystore InstallDirectory/jre/lib/security/cacerts -
   storepass password
   ```

   – ***password***
   Specify the same password when creating the self-signed certificate.

   NOTE
   The default password for the Certificate Authority keystore is **changeit**.
5. (Optional) Search the output for the Certificate Authority that signed your certificate. If the Certificate Authority is not listed, add it to the list of trusted authorities:

   ```
   keytool -importcert -keystore InstallDirectory/jre/lib/security/cacerts -
   storepass cacertspassword -alias alias_name -file RootCA.cer
   ```

   – ***cacertspassword*** Specify the password for the Certificate Authority keystore.
   – ***alias_name***
   Specify an alias that can be used to refer to the keystore entry that is created for the root or intermediate certificate.
   – ***RootCA*** **.cer**
   Specify the filename of the root certificate.

   NOTE
   Import the root certificate and any intermediate certificates between the certificate authority root certificate and the certificates authorities signed certificate.
6. Import the signed certificate:

   ```
   keytool -importcert -trustcacerts -keystore keystore_file.ks -storepass password -
   alias alias_name -keypass password -file SignedCert.cer
   ```

   – ***password***
   Specify the same password when creating the self-signed certificate.
   – ***alias_name***
   Specify the same alias when creating the self-signed certificate.
   – ***SignedCert*****.cer**
   Specify the certificate file from the Certificate Authority.
7. Confirm that you trust the certificate.
8. Validate the contents of the keystore:

   ```
   keytool -list -keystore InstallDirectory/PerformanceCenter/sso/webapps/sso/
   configuration/keystore_file.ks
   ```

   The single certificate that you imported appears in the list.
   The Certificate Authority SAML certificate replaces your self-signed certificate in the keystore.
9. Update the certificate in the IdP with the same certificate that you just imported.

Import a Key and an Existing Certificate

You can use a private key and public certificate (a self-signed or a Certificate Authority certificate) from a different source. For example, your security team provides an SAML certificate that is customized for your organization. To use this SAML certificate, import the private key and the signed certificate.

Follow these steps:

1. Change the directory:
   ```
   cd InstallDirectory/PerformanceCenter/sso/webapps/sso/configuration
   ```

   NOTE
   /opt/CA is the default installation directory.
2. If a keystore file exists, rename the existing keystore file to create a backup of it:
   ```
   textmv InstallDirectory/PerformanceCenter/sso/webapps/sso/configuration/
   keystore InstallDirectory/PerformanceCenter/sso/webapps/sso/configuration/
   keystore.bak
   ```

   WARNING
   Move the old keystore. If you do not, an error appears in later steps: "Keystore was tampered with, or password was incorrect."
3. Create a PKCS#12 keystore from the private key and certificate:
   ```
   openssl pkcs12 -export -in certificate.pem -inkey privatekey.pem -name alias_name -
   out keystore.pkcs12
   ```

   - ***certificate*.pem**
     Specify the certificate that is provided to you.
   - ***privatekey*.pem**
     Specify the private key that is provided to you.
   - **alias_name**
     Specify an alias that can be used to refer to the keystore entry that is created for the certificate
   - ***keystore*.pkcs12**
     Specify the keystore to create to store the keys provided.

     NOTE
     This command works on Linux only.
4. Import the key and certificate into the CA NetOps Portal keystore:
   ```
   keytool -importkeystore -destkeystore keystore_file.ks -deststorepass password
   -srckeystore keystore.pkcs12 -srcstoretype pkcs12 -srcalias src_alias_name -
   destalias dest_alias_name -destkeypass password
   ```

   - ***keystore_file.* ks** Specify the name of the keystore file to create.
   - ***password*** Specify the password for the keystore and imported certificate. Specify a secure password.
   - ***keystore.* pkcs12** Specify the PKCS#12 keystore previously created.
   - ***src_alias_name*** Specify the alias_name when importing the private key and certificate.
   - ***dest_alias_name*** Specify an alias that can be used to refer to the keystore entry that is created for the imported certificate.

   Your existing SAML certificate is imported into the keystore.
5. Determine whether the certificate includes a chain terminating at a certificate in the keystore. If the certificate is missing, import it into the Java keystore.
   ```
   keytool -printcert -file filename
   ```

   - ***filename*** specifies the name of the certificate.
6. Update the certificate in the IdP with the same certificate that you just imported.

Configure SAML 2.0 Support in Single Sign-On

The administrator uses the Single Sign-On Configuration Tool to set parameters for SAML authentication. Take these steps on all servers with a data source that has users who authenticate using SAML 2.0.

> **NOTE**
>
> : Multiple authentication schemes can be in use simultaneously. For example, users of a CA Network Flow Analysis data source can use LDAP to log in, while users of DX NetOps Performance Management are using SAML 2.0.

Follow these steps:

1. Log in to the server where CA NetOps Portal or a CA data source product is installed.
   Log in as root or with the 'sudo' command.
2. Launch the Single Sign-On Configuration Tool by running the './SsoConfig' command in the following directory:
   `InstallationDirectory/CA/PerformanceCenter`
   You are prompted to select an option. The available options correspond to CA applications running on the local server.
3. Use the following commands as needed while you are selecting settings:
   - q (quit)
   - b (go back to the previous menu)
   - u (update)
   - r (reset)
4. Enter the value that corresponds to the data source that you want to configure. For example, enter 1 to configure CA NetOps Portal.
   You are prompted to select an option.
5. Enter 2 for SAML Authentication.
   You are prompted to specify the priority.
   The Priority parameter only applies to CA NetOps Portal.
6. Enter one of the following options:
   - **1. Remote Value**These settings are propagated to all other CA products and data sources that are registered to this instance of CA NetOps Portal. This includes the Event Manager in CA NetOps Portal, which embeds the URL of CA NetOps Portal. CA NetOps Portal uses Remote Value settings only if a corresponding Local Override value is not present.
   - **2. Local Override**
     Overrides a setting on this CA NetOps Portal instance, which does not propagate to other CA products and data sources (including Event Manager) registered to this instance of CA NetOps Portal. Local Override takes precedence over both the Remote Value and default settings.

   You are prompted to select a property to configure.
   To supply values for the SAML2 properties, enter u to update the value and then enter a new value.

   > **NOTE**
   >
   > Configure the scheme or port using Remove Value to include the correct CAPC URL in threshold event email messages.
7. Enter 1 to select the 'Enable SAML2 Authentication' parameter.
   You are prompted to select an option.
8. Enter u to change the value, and enter 1 to enable SAML 2.0 authentication.
9. Enter 2 to set the 'Clone Default User Accounts' parameter.
   - **2. Clone Default User Accounts**
     Defines the user account to which authorized SAML users are mapped. The role and product privileges that are associated with the user account you specify are applied to all users who successfully authenticate.
     Default: Blank.
     Example: Enter 'user' if you want all users to log in with user-level privileges.

NOTE

An existing user account is required.

The user accounts configured on the IdP are sent to CA NetOps Portal when the agreement is established. They appear in the User List on the Manage Users Admin page, where they can be edited.

10. Enter 3 to enable security parameters.
    - **3. SAML2 Signature and Encryption Enabled**Enables security and encryption for communications between CA NetOps Portal and the IdP.**Default**: Disabled
11. You are prompted to choose an option.
12. Enter u to change the value, and enter 1 to enable it.

    NOTE
    This setting must match the setting on the IdP.
13. Enter 4 to enable automatic reauthentication.
    - **4. SAML2 Auto-Reauthentication**Specifies whether the system keeps the IdP session for the user active. Enable this parameter to allow DX NetOps Performance Management to perform a passive reauthentication ('auto-reauthentication').
      The next parameter lets you set the duration of the timeout period.**Default**: Disabled.
14. Enter u to change the value, and enter 1 to enable it.
15. Enter 5 to set the reauthentication timeout period.
    - **5. Auto-Reauthentication Time Period**Specifies the period of time before a passive reauthentication is performed. If the 'SAML2 Auto-Reauthentication' parameter is disabled, this parameter is ignored.
      Default: None.
16. Enter u to change the value, and enter a new value.
17. Enter b twice to go back to the initial prompt.
18. Enter 6 to export the metadata file that establishes the agreement with the IdP.The metadata file supplies the identity provider with the parameters to use when authenticating users.
19. You are asked to supply a directory path and filename.
20. Enter the filename. For example, enter the following:

    ```
    /tmp/CAPCMetadata.xml
    ```

    The file is generated automatically, based on the settings you selected in the Configuration Tool.
21. You see a printout of the XML if the export operation succeeds. If the operation fails, you see an error message.
22. Enter q to quit.The Configuration Tool closes.

Configure the IdP

To use SAML 2.0 for user authentication in NetOps Portal, set the appropriate parameters on the identity provider (IdP). Any IdP that supports the SAML 2.0 standard should work, but CA has tested only CA SiteMinder.

You can manually configure the IdP, or you can import the IdP agreement from the Single Sign-on server.

Manually Configure the IdP

Follow these steps:

1. Enable the SAML2 authentication mode on the IdP.
2. Provide a URL for the assertion consumer service, which is running on the servers where Single Sign-On is installed. For example:

   ```
   http://MyServerName:8381/sso/saml2/UserAssertionService
   ```

where 8381 is the port that Single Sign-On uses.

3. Set the binding method to 'HTTP-Redirect'.

   NOTE
   HTTP Redirect is the only binding method that Single Sign-On supports.

4. Provide URLs for the single logout service.
   The logout service and the response location are both required. These services are running on the server where Single Sign-On is installed.
   Use the following examples:

   ```
   http://MyServerName:8381/sso/saml2/LogoutService
   http://MyServerName:8381/sso/saml2/LogoutServiceResponse
   ```

5. Add all data source product websites that support SAML 2.0 to the list of trusted sites.
   This step can involve adding these websites to a list of federation partnership entities.
6. (*Optional*) Verify digital signature and encryption settings. You must also configure these settings in Single Sign-On.

Import the IdP Agreement File

Follow these steps:

1. Import the IdP agreement file from its location on the Single Sign-On server.
   You exported this file after you completed other setup steps using the Single Sign-On Configuration Tool. For more information, see Configure SAML Support in Single Sign-On.
2. Add all data source product websites that support SAML 2.0 to the list of trusted sites.
   This step can involve adding these websites to a list of federation partnership entities.
3. (*Optional*) Verify digital signature and encryption settings. You must also configure these settings in Single Sign-On.

Troubleshooting

Problem:

You see the following error message after configuring SAML:

```
RelayState is either null or a blank string. RelayState must be set for SSO to work correctly.
Invalid syntax, RelayState=<value>
RelayState does not have parameter SsoRedirectUrl, RelayState=<value>
```

Reason:

Some IdPs do not return the RelayState= value that CA NetOps Portal sends to the IdP during authentication verification.

Solution:

Manually configure RelayState for your IdP. Use the following syntax:

```
SsoProductCode=pc&SsoRedirectUrl=http://<capc>:8181/pc/desktop/page
```

NOTE
For secure communications, replace http: with https:, and replace the port number.

Complete SAML 2.0 Setup

To enable SAML 2.0 authentication, edit user accounts to use External Authentication. New user accounts in CA NetOps Portal are set to use NetOps Portal Authentication by default. The administrator must update the accounts of all operators who authenticate using SAML 2.0.

During SAML2.0 configuration, you specify an existing CA NetOps Portal user account to be 'cloned' in the IdP. Any users who are already defined on the IdP receive the same level of product privilege as the user account you designate. These

accounts are also propagated to CA NetOps Portal, where they appear as new users in the User List. In many cases, you must edit these accounts to make sure that these users can access only the data they require to do their jobs.

Follow these steps:

1. Log in to CA NetOps Portal as a user with administrative privileges.
2. Select **Administration**, **User Settings**, and click **Users**.
   The Manage Users page opens.
3. Select a user account to edit.
4. Click **Edit**.
   The Edit User wizard opens.
5. Select **External** as the Authentication Type.
6. Use the wizard to make any other desired changes to the user account.
7. Click **Save**.
   The changes to the user account are saved.

Set Up HTTPS

By default, Single Sign-On uses HTTP for communications between the browser and DX NetOps Performance Management. TLS (Transport Layer Security) and its predecessor, SSL (Secure Sockets Layer), are widely supported encryption protocols that secure data transmissions over the Internet. TLS and SSL can be used with HTTP to form HTTPS (HTTP-Secure). This guide uses *SSL* as a blanket term to mean "TLS and SSL."

> **NOTE**
>
> Older HTTPS ciphers and protocols (TLS v1.0, v1.1 and SSL v3) are no longer supported by default. TLS v1.2 is the only cipher suite offered by default.

You can enhance the security in your monitoring system by configuring Single Sign-On to use HTTPS instead of HTTP.

Configuring CA Single Sign-On to use HTTPS is optional. Before you can configure the Single Sign-On website to use HTTPS, you must obtain a server certificate. The team that creates and enforces security policies for your organization can probably assist you with these steps.

Enhance the security in your monitoring system by configuring Single Sign-On to use HTTPS instead of HTTP. To enable HTTPS for CA Single Sign-On, first enable SSL for NetOps Portal:

1. Install the certificates that validate the identity of the server.
2. Change the database to ensure that NetOps Portal properly redirects to the correct port and scheme for Single Sign-On, and the reverse.
3. Change the services for both NetOps Portal and Single Sign-On to reflect the new ports and schemes.

Two ports are important for these steps: the NetOps Portal port (which defaults to 8181) and the Single Sign-On port (which defaults to 8381). Port 8181 is the NetOps Portal connection port. If users require authentication, the server redirects them to Single Sign-On on port 8381, where they see the Login page. Once a user has successfully logged in, the server redirects that user back to the original URL at port 8181.

Therefore, you cannot use the same port in each configuration step. Otherwise, a conflict occurs between NetOps Portal and Single Sign-On.

You can enable HTTPS using the SSL configuration tool, or you can configure SSL manually.

To enable HTTPS for NetOps Portal and Single Sign-On with the SSL configuration tool, see Enable Performance Center to use SSL.

To enable HTTPS for NetOps Portal and Single Sign-On manually, complete the following steps:

1. Set Up SSL Certificates for Performance Center.
2. Configure the Port and Website for HTTPS.

3. Configure Performance Center to Use HTTPS.
4. Update Single Sign-On Configuration and Restart the Services.

To enable HTTPS for the Data Aggregator, complete the following steps:

1. Configure the Data Aggregator to Use HTTPS
2. Configure the Data Aggregator and Performance Center for HTTPS

Enable Performance Center to use SSL

You can enable HTTPS using the SSL configuration tool. Before you enable HTTPS using the SSL configuration tool, Set Up SSL Certificates for Performance Center. To configure SSL manually, see Enable Performance Center to use SSL Manually.

In addition to setting up SSL, you can secure the single sign-on to prevent possible redirection to other sites. For more information, see Limit Single Sign-On Redirection.

The SSL configuration tool also allows you to check for any configuration issues or revert to the default settings.

NOTE
The private and signed certificate should be in PEM format.

Configure SSL

Use the SSL configuration tool to configure NetOps Portal to use SSL.

To configure signed certificates, see Set Up SSL Certificates for Performance Center.

Follow these steps:

1. Launch the SSL configuration tool by running the '`./SslConfig`' command in the following directory:
   `PC_Install_Directory``/PerformanceCenter`

   NOTE
   `/opt/CA`
   is the default installation directory.
2. Complete the following prompts:
   - **Preferred Language**
     Specify a language for the configuration tool.
   - **Options**
     To configure NetOps Portal to use SSL, specify 2. Confirm your selection.
   - **Single Sign-On Port**
     Specify the port for Single Sign-On (for example: 8382).
   - **NetOps Portal Port**
     Specify the port for NetOps Portal (for example: 8381).
   - **Existing Certificate**
     Specify whether an SSL certificate exists.
     If you are generating a new certificate, specify No and complete the certificate prompts.
   - **Host Name**
     Specify the full host name for your NetOps Portal server.
   - **Password**
     Specify a newly created password for importing the certificate into the jetty keystore:

PC_Install_Directory/PerformanceCenter/jetty/etc/keystore

- **Obfuscate**
  Specify whether to obscure the password in the jetty configuration files.
- **Trust Store Password**
  Specify the password to the JRE trust store:
  PC_Install_Directory/jre/lib/security/cacerts
  Default: changeit

  NOTE

  In the rare event that you encounter issues with the SSL configuration tool, a debug log is available in the *InstallDirectory*/PerformanceCenter directory.

Check for Configuration Issues

You can also use the SSL configuration tool to check your SSL configuration for any potential issues. Common issues include certificate errors.

Follow these steps:

1. Launch the SSL configuration tool by running the '**.**/SslConfig' command in the following directory:
   PC_Install_Directory/PerformanceCenter

   NOTE

   /opt/CA is the default installation directory.
2. Complete the following prompts:
   - **Preferred Language**
     Specify a language for the configuration tool.
   - **Options**
     To check your configuration for any potential issues, specify 1. Confirm your selection.
3. Review output.

Revert to the Default Settings

If necessary, use the SSL configuration tool to revert to the default settings.

Follow these steps:

1. Launch the SSL configuration tool by running the '**.**/SslConfig' command in the following directory:
   PC_Install_Directory/PerformanceCenter

   NOTE

   /opt/CA is the default installation directory.
2. Complete the following prompts:
   - **Preferred Language**
     Specify a language for the configuration tool.
   - **Options**
     To revert to the default settings, specify 3. Confirm your selection.

Enable Performance Center to use SSL Manually

You can enable HTTPS using the SSL configuration tool, or you can configure SSL manually. To enable HTTPS using the SSL configuration tool, see Enable Performance Center to use SSL.

To configure the single sign-on website to use HTTPS, obtain and install a private key and an associated public certificate. SSL can be used with either a self-signed certificate or a certificate that a trusted Certificate Authority has signed. The procedures are typically specific to an organization and the policies of its security team.

To set up HTTPS, use the Single Sign-On Configuration Tool to update the default website scheme and port to match the encryption settings. By default, Single Sign-On uses Port 8381.

To configure NetOps Portal to use HTTPS, edit the configuration files with the website and port settings. Replace the HTTP connector with an HTTPS connector.

Edit startup files to support SSL encryption in Single Sign-On and restart all NetOps Portal and single sign-on services to update the settings. In addition to setting up SSL, you can secure the single sign-on to prevent possible redirection to other sites.

In summary, to enable HTTPS for CA NetOps Portal and Single Sign-On manually, complete the following steps:

1. Set Up SSL Certificates for Performance Center.
2. Configure the Port and Website for HTTPS.
3. Configure Performance Center to Use HTTPS.
4. Update Single Sign-On Configuration and Restart the Services.

Set Up SSL Certificates for Performance Center

To configure the single sign-on website to use HTTPS, obtain and install a private key and an associated public certificate. SSL can be used with either a self-signed certificate or a certificate that a trusted Certificate Authority has signed. The procedures are typically specific to an organization and the policies of its security team. However, these procedures provide some guidance.

Select the appropriate procedure for your situation:

> **NOTE**
> For more information about the keytool command, see the Java documentation on the Oracle website.

Generate and Import a Certificate

To generate an SSL certificate, use the keytool command. Generate a self-signed certificate and install it in the keystore.

Follow these steps:

1. Change the directory:
   ```
   cd InstallDirectory/PerformanceCenter/jetty/etc
   ```

   > **NOTE**
   > /opt/CA is the default installation directory.
2. If a jetty keystore file exists, rename the existing jetty keystore file to create a backup of it:
   ```
   mv InstallDirectory/PerformanceCenter/jetty/etc/keystore InstallDirectory/
   PerformanceCenter/jetty/etc/keystore.bak
   ```

WARNING
Move the old keystore. If you do not, an error appears in later steps: "Keystore was tampered with, or password was incorrect."

3. Generate a private key and a public, self-signed certificate:

```
keytool -genkeypair -ext SAN=dns:fully_qualified_hostname -keystore keystore_file.ks
-storepass password -keyalg RSA -keysize 2048 -keypass password -alias alias_name
```

Note your entries for the following variables:

- ***fully_qualified_hostname*** Specify the fully qualified host name of the server. Enter the same value when you are prompted for your first and last name.
- ***keystore_file.*** **ks**
  Specify the name of the keystore file to create.
- ***password***
  Specify the password for the keystore and self-signed certificate. Specify a secure password.
- ***alias_name***
  Specify an alias that can be used to refer to the keystore entry that is created for the self-signed certificate.

  NOTE
  N ote: When you are prompted for your first and last name, provide the fully qualified hostname of the server.

4. Proceed through the security prompt questions and confirm your responses.
5. Export the self-signed certificate from the keystore:

```
keytool -exportcert -keystore keystore_file.ks -storepass password -alias alias_name
-file filename.cer
```

- ***keystore_file*** **.ks** Specify the same keystore file name previously created.
- ***password***
  Specify the same password when creating the self-signed certificate.
- ***alias_name***
  Specify the same alias when creating the self-signed certificate.
- ***filename*****.cer**
  Specify the file to which the certificate is exported. We recommend using a full pathname that does not place the file in the current directory.
  Example: /tmp/capcCert.cer

  NOTE
  We recommend backing up any certificates that could be rewritten before continuing.

6. Import the self-signed certificate into the java trusted certificate keystore:

```
keytool -importcert -keystore InstallDirectory/jre/lib/security/cacerts
-storepass cacertspassword -alias alias_name -file filename.cer
```

- ***cacertspassword***
  Specify the password for the Certificate Authority keystore.

  NOTE
  The default password for the Certificate Authority keystore is **changeit**.
- ***alias_name***
  Specify the same alias when creating the self-signed certificate.
- ***filename*** **.cer**

Specify the file to which the certificate is exported. We recommend using a full pathname that does not place the file in the current directory.
Example: /tmp/capcCert.cer

7. Confirm that you trust the certificate.
8. Back up the certificate file:
   `cp filename.cer filename.cer.bak`
9. (Optional) For more security, change the password of the java trusted certificates keystore:
   `keytool -storepasswd -keystore InstallDirectory/jre/lib/security/cacerts`

   You are prompted to provide the existing keystore password and a new keystore password.
10. Verify that your imported keystore is available:
    `keytool -list -keystore InstallDirectory/jre/lib/security/cacerts`

WARNING
To enable the web services, the self-signed certificate must be in the Certificate Authority keystore. Otherwise, you see an error in the log that reports that PKIX did not find a certificate.

Your self-signed SSL certificate is generated and installed in the keystore.

Convert a Self-Signed Certificate to a Certification Authority SSL Certificate

A self-signed certificate causes a browser warning to appear when you open NetOps Portal. The warning does not appear if you use a certificate that a trusted Certification Authority has signed. The following procedure explains how to convert the self-signed certificate to a certificate that a trusted Certification Authority has signed.

Follow these steps:

1. Change the directory:
   `cd InstallDirectory/PerformanceCenter/jetty/etc`

   NOTE
   /opt/CA is the default installation directory.
2. Export a certificate signature request:
   `keytool -certreq -keystore keystore_file.ks -storepass password -ext SAN=dns:[FQHN] -alias alias_name -keypass password -file RequestFileName.csr`

   - ***keystore_file***
     Specify the same keystore file name previously created.
   - ***password***
     Specify the same password when creating the self-signed certificate.
   - ***alias_name***
     Specify an alias that can be used to refer to the keystore entry that is created for the root or intermediate certificate.
   - ***RequestFileName*.csr**
     Specify the path and file name of the exported signature request.
3. Send the resulting file (***RequestFileName*** .csr) to a qualified signing authority with any other requested information. The Certificate Authority sends you a signed certificate (***SignedCert*** .cer). They might also provide a root Certificate Authority certificate (***RootCA*** .cer) to authenticate the signed certificate.
4. Determine whether the root Certificate Authority certificate is part of the default java trusted authorities:

```
keytool -list -v -keystore InstallDirectory/jre/lib/security/cacerts -
storepass cacertspassword
```

- ***cacertspassword***
  Specify the password for the Certificate Authority keystore.

  NOTE
  The default password for the Certificate Authority keystore is **changeit**.

5. Import the intermediate or root certificates into the java trusted certificate keystore using the following steps for each certificate:

```
keytool -importcert -keystore InstallDirectory/jre/lib/security/cacerts
-storepass cacertspassword -alias alias_name -file filename.cer
```

- ***cacertspassword***
  Specify the password for the Certificate Authority keystore.

  NOTE
  The default password for the Certificate Authority keystore is **changeit**.
- ***alias_name***
  Specify an alias that can be used to refer to the keystore entry that is created for the root or intermediate certificate.
- ***filename* .cer**
  Specify the file to which the certificate is exported. We recommend using a full pathname that does not place the file in the current directory.
  Example: /tmp/capcCert.cer

6. Import the signed certificate into the jetty keystore:

```
keytool -importcert -trustcacerts -keystore keystore -storepass password -
alias alias_name -keypass password -file SignedCert.cer
```

- ***password***
  Specify the same password when creating the self-signed certificate.
- ***alias_name***
  Specify the same alias when creating the self-signed certificate.
- ***SignedCert*.cer**
  Specify the certificate file from the Certificate Authority.

7. Confirm that you trust the certificate.
8. Validate the contents of the jetty keystore:

```
keytool -list -keystore InstallDirectory/PerformanceCenter/jetty/etc/keystore
```

   The single certificate that you imported appears in the list.
   The Certificate Authority SSL certificate replaces your self-signed certificate in the keystore.
9. Import the signed certificate into the java trusted certificate keystore:

```
keytool -importcert -keystore InstallDirectory/jre/lib/security/cacerts
-storepass cacertspassword -alias alias_name -file filename.cer
```

- ***cacertspassword***
  Specify the password for the Certificate Authority keystore.

  NOTE
  The default password for the Certificate Authority keystore is **changeit**.
- ***alias_name***

Specify the same alias when creating the self-signed certificate.

- ***filename* .cer**
  Specify the file to which the certificate is exported. We recommend using a full pathname that does not place the file in the current directory.
  Example: /tmp/capcCert.cer

10. Confirm that you trust the certificate.
11. Verify that your imported keystore is available:
    `keytool -list -keystore InstallDirectory/jre/lib/security/cacerts`

Import a Key and an Existing Certificate

You can use a private key and public certificate (a self-signed or a Certificate Authority certificate) from a different source. For example, your security team provides an SSL certificate that is customized for your organization. To use this SSL certificate, import the private key and the signed certificate.

Follow these steps:

1. Change the directory:
   `cd InstallDirectory/PerformanceCenter/jetty/etc`

   NOTE
   /opt/CA is the default installation directory.
2. If a jetty keystore file exists, rename the existing jetty keystore file to create a backup of it:
   `textmv InstallDirectory/PerformanceCenter/jetty/etc/keystore InstallDirectory/PerformanceCenter/jetty/etc/keystore.bak`

   WARNING
   Move the old keystore. If you do not, an error appears in later steps: "Keystore was tampered with, or password was incorrect."
3. Create a PKCS#12 keystore from the private key and certificate:
   `openssl pkcs12 -export -in certificate.pem -inkey privatekey.pem -name alias_name -out keystore.pkcs12`

   - ***certificate*.pem**
     Specify the certificate that is provided to you.
   - ***privatekey*.pem**
     Specify the private key that is provided to you.
   - **alias_name**
     Specify an alias that can be used to refer to the keystore entry that is created for the certificate
   - ***keystore*.pkcs12**
     Specify the keystore to create to store the keys provided.

     NOTE
     This command works on Linux only.
4. Import the key and certificate into the CA NetOps Portal keystore:
   `keytool -importkeystore -destkeystore keystore_file.ks -deststorepass password -srckeystore keystore.pkcs12 -srcstoretype pkcs12 -srcalias src_alias_name -destalias dest_alias_name -destkeypass password`

- ***keystore_file.*** **ks** Specify the name of the keystore file to create.
- ***password*** Specify the password for the keystore and imported certificate. Specify a secure password.
- ***keystore.*** **pkcs12** Specify the PKCS#12 keystore previously created.
- ***src_alias_name*** Specify the alias_name when importing the private key and certificate.
- ***dest_alias_name*** Specify an alias that can be used to refer to the keystore entry that is created for the imported certificate.

Your existing SSL certificate is imported into the keystore.

5. Determine whether the certificate includes a chain terminating at a certificate in the keystore. If the certificate is missing, import it into the Java keystore.

   ```
   keytool -printcert -file filename
   ```

   - ***filename*** specifies the name of the certificate.
6. Import the signed certificate, and intermediate or root certificates into the java trusted certificate keystore using the following steps for each certificate:

   ```
   keytool -importcert -keystore InstallDirectory/jre/lib/security/cacerts
   -storepass cacertspassword -alias alias_name -file filename.cer
   ```

   - ***cacertspassword***
     Specify the password for the Certificate Authority keystore.

     NOTE
     The default password for the Certificate Authority keystore is **changeit**.
   - ***alias_name***
     Specify the same alias when importing the signed certificate into the NetOps Portal keystore. Or, specify an alias that can be used to refer to the keystore entry that is created for the root or intermediate certificate.
   - ***filename*** **.cer**
     Specify the file to which the certificate is exported. We recommend using a full pathname that does not place the file in the current directory.
     Example: /tmp/capcCert.cer
7. Confirm that you trust the certificate.
8. Verify that your imported keystore is available:

   ```
   keytool -list -keystore InstallDirectory/jre/lib/security/cacerts
   ```

Configure the Port and Website for HTTPS

To set up HTTPS, use the Single Sign-On Configuration Tool to update the default website scheme and port to match the encryption settings. By default, Single Sign-On uses Port 8381.

Perform the tasks in this procedure on every server where a data source is installed.

Follow these steps:

1. Launch the Single Sign-On Configuration Tool by running the './SsoConfig' command in the following directory:

   ```
   InstallDirectory/PerformanceCenter
   ```

   NOTE
   /opt/CA is the default installation directory.

   You are prompted to select an option.

2. Use the following commands as needed while you are changing settings:
   - q (quit)
   - b (go back to the previous menu)
   - u (update)
   - r (reset)
3. Enter 1 to select CA NetOps Portal (CAPC).
4. Enter 4 to configure Single Sign-On.
   You are prompted to specify the priority.
5. Enter one of the following options:
   - **1. Remote Value**
     These settings are propagated to all other CA products and data sources that are registered to this instance of CAPC. This includes the Event Manager in CAPC, which embeds the URL of CAPC. CAPC uses Remote Value settings only if a corresponding Local Override value is not present.
   - **2. Local Override**
     Overrides a setting on this CAPC instance, which does not propagate to other CA products and data sources (including Event Manager) registered to this instance of CAPC. Local Override takes precedence over both the Remote Value and default settings.

   You are prompted to select a property to configure.

   NOTE

   Configure the scheme or port using Remove Value to include the correct CAPC URL in threshold event email messages.
6. Enter 12 for the Scheme property.
7. Enter 'u' to update the value.
8. Supply 'https' for the value.
9. Enter 13 for the Port property.
10. Update the value to '8382'.
11. Enter 'b' twice to go back to the SSO Configuration/CA NetOps Portal menu.
12. Enter '3' to configure the NetOps Portal.
    You are prompted to specify the priority.
13. Enter either '1' for Remote Value or '2' for Local Override.
14. Enter '6' to select Web Site Scheme.
15. Update the value to 'https'.
16. Enter '8' to select Web Site Port.
17. Update the value to '8182'.
18. Enter q to quit.

Configure Performance Center to Use HTTPS

To configure NetOps Portal to use HTTPS, edit the configuration files with the website and port settings and replace the HTTP connector with an HTTPS connector.

NOTE
By default, TLS v1.2 is the only cipher suite offered. Older HTTPS ciphers and protocols, such as TLS v1.0, v1.1, and SSL v3, are not supported.

Follow these steps:

1. Change to the following directory:
   `cd /InstallDirectory/PerformanceCenter/PC`

NOTE
`/opt/CA` is the default installation directory.

2. Edit the `start.ini` file and apply the following changes:
   – Find the following lines and update them, as follows:
   Original Text
   ```
   # To enable ssl, modify this line to use module https
   # Module: http
   --module=http
   ```
   New Text
   ```
   # To enable ssl, modify this line to use module https
   # Module: https
   --module=https
   ```
   Original Text
   ```
   # To enable ssl, uncomment this line module
   # Module: ssl
   #--module=ssl
   #etc/ssl-lucky13.xml
   ```
   New Text
   ```
   # To Enable ssl, uncomment this line module
   # Module: ssl
   --module=ssl
   etc/ssl-lucky13.xml
   ```
   – If NetOps Portal is not installed in the default installation directory location, update the directory path.
3. Add the port and password information to the `PC/start.d/ssl.ini` file:
   ```
   # SSL
   # define the port to use for secure redirection
   jetty.ssl.port=8182
   jetty.https.port=8182
   jetty.httpConfig.securePort=8182
   # Set up a keystore and truststore
   jetty.sslContext.keyStoreType=JKS
   jetty.sslContext.keyStorePath=etc/keystore_file.ks
   jetty.sslContext.trustStorePath=etc/keystore_file.ks
   # Set up passwords
   jetty.sslContext.keyStorePassword=password
   jetty.sslContext.keyManagerPassword=password
   jetty.sslContext.trustStorePassword=password
   ```
4. Specify the following values from the SSL certificate setup:
   – ***keystore_file*.ks**
   Specify the name of the keystore file that is used to store the certificate.

   NOTE
   The keystore file must be in the `etc` directory.
   – ***password***
   Specify the password for the keystore that is selected when creating the certificate.

NOTE
By default, the password values for the `jetty.sslContext.keyStorePassword`, `jetty.sslContext.keyManagerPassword`, and `jetty.sslContext.trustStorePassword` parameters are stored in plain text. However, you can obfuscate them.
For more information, see Obfuscate Jetty Passwords.

5. Change to the following directory:
   `cd /InstallationDirectory/CA/PerformanceCenter/sso/webapps/sso/configuration`
6. Edit the `CAPerformanceCenter.xml` file.
7. Replace the `Scheme` and `Port` values with settings that are appropriate for SSL:

```
<?xml version="1.0" encoding="utf-8" ?>
<Configuration>
     <SingleSignOnEnabled>True</SingleSignOnEnabled>
     <SingleSignOnProductCode>pc</SingleSignOnProductCode>
     <SignInPageProductDefaultUrl>
          <Scheme>https</Scheme>
          <Port>8182</Port>
          <PathAndQuery>/pc/desktop/page</PathAndQuery>
     </SignInPageProductDefaultUrl>
     <SingleSignOnWebServiceUrl>
          <Scheme>https</Scheme>
          <Port>8182</Port>
          <PathAndQuery>/pc/center/webservice/sso</PathAndQuery>
     </SingleSignOnWebServiceUrl>
</Configuration>
```

8. Edit the `CADataAggregator.xml` file.
9. Replace the `Scheme` and `Port` values for `SingleSignOnWebServiceUrl` with settings that are appropriate for SSL. Do not change values for **SignInPageProductDefaultUrl**:

```
<?xml version="1.0" encoding="utf-8" ?>
<Configuration>
  <SingleSignOnEnabled>True</SingleSignOnEnabled>
  <SingleSignOnProductCode>da</SingleSignOnProductCode>
  <RemoteWebSite>True</RemoteWebSite>
  <SignInPageProductDefaultUrl>
    <Scheme>http</Scheme>
    <Port>8581</Port>
    <PathAndQuery>/</PathAndQuery>
  </SignInPageProductDefaultUrl>
  <SingleSignOnWebServiceUrl>
    <Scheme>https</Scheme>
    <Port>8182</Port>
    <PathAndQuery>/pc/center/webservice/sso</PathAndQuery>
  </SingleSignOnWebServiceUrl>
</Configuration>
```

Update Single Sign-On Configuration and Restart the Services

Edit startup files to support SSL encryption in Single Sign-On and restart all NetOps Portal and Single Sign-On services to update the settings. In addition to setting up SSL, you can secure the single sign-on to prevent possible redirection to other sites. For more information, seeLimit Single Sign-On Redirection.

Follow these steps:

1. Change to the following directory:
   `cd InstallDirectory/PerformanceCenter/sso`

   NOTE
   `/opt/CA` is the default installation directory.
2. Edit the `start.ini` file and apply the following changes:
   – Find the following lines and update them, as follows:
   Original Text
   ```
   # To enable ssl, modify this line to use module https
   # Module: http
   --module=http
   ```
   New Text
   ```
   # To enable ssl, modify this line to use module https
   # Module: https
   --module=https
   ```
   Original Text
   ```
   # To enable ssl, uncomment this line module
   # Module: ssl
   #--module=ssl
   #etc/ssl-lucky13.xml
   ```
   New Text
   ```
   # To Enable ssl, uncomment this line module
   # Module: ssl
   --module=ssl
   etc/ssl-lucky13.xml
   ```
   – If NetOps Portal is not installed in the default location, update the directory path.
3. Add the port and password information to the `sso/start.d/ssl.ini` file:
   ```
   # SSL
   # define the port to use for secure redirection
   jetty.ssl.port=8382
   jetty.https.port=8382
   jetty.httpConfig.securePort=8382
   # Set up a keystore and truststore
   jetty.sslContext.keyStoreType=JKS
   jetty.sslContext.keyStorePath=etc/keystore_file.ks
   jetty.sslContext.trustStorePath=etc/keystore_file.ks
   # Set up passwords
   jetty.sslContext.keyStorePassword=password
   jetty.sslContext.keyManagerPassword=password
   jetty.sslContext.trustStorePassword=password
   ```

   Specify the following values from the SSL certificate setup:
   – ***keystore_file*.ks**

Specify the name of the keystore file used to store the certificate.

– ***password***
Specify the password for the keystore selected when creating the certificate.

4. (Optional) To disable all protocols except TLSv1.2, modify the following file:
`PC_install_directory/PerformanceCenter/jetty/etc/jetty-ssl-context.xml`
Add the following code to the end of the file before `</Configure>` :

```
<Set name="IncludeProtocols">
     <Array type="java.lang.String">
          <Item>TLSv1.2</Item>
     </Array>
</Set>
```

5. Stop the console, device manager, and SSO services by entering the following commands:
```
service caperfcenter_console stop
service caperfcenter_devicemanager stop
service caperfcenter_sso stop
```
6. Restart the services by entering the following commands:
   a. Start the SSO service:
   ```
   service caperfcenter_sso start
   ```
   b. Wait one minute, then start the device manager:
   ```
   service caperfcenter_devicemanager start
   ```
   c. Wait one minute, then start the console service:
   ```
   service caperfcenter_console start
   ```

Enable the Data Aggregator to use SSL

In an environment with a single Data Aggregator, you can enable HTTPS using the Data Aggregator SSL configuration tool. To configure HTTPS manually, see Enable the Data Aggregator to use SSL Manually.

You can also enable HTTPS for the Data Aggregator in a fault tolerant environment with multiple Data Aggregators. For more information, see Enable Fault Tolerant Data Aggregators to Use SSL.

> **NOTE**
> The private and signed certificate should be in PEM format. 3.7.4 and higher supports the PKCS12 file format in addition to the PEM file format.

Configure HTTPS

Use the Data Aggregator SSL configuration tool to configure the Data Aggregator to use HTTPS.

Follow these steps:

1. Launch the SSL configuration tool by running the `./sslConfig.sh` command in the following directory:
`DA_Install_Directory/scripts`

   > **NOTE**
   > `/opt/IMDataAggregator` is the default installation directory.

2. Select `Enable HTTPS FIPS`.
3. Complete the following prompts:

- **Existing Signed Certificate**
  Specify whether an SSL certificate and private key exist.
  If you have an existing certificate and private key, specify Yes and the location and filename of the certificate and key.
  If you are generating a new certificate, specify No and complete the certificate prompts.
- **HTTPS Port**
  Specify the port for secure communication.
  Default: 8582

Import the Data Aggregator Certificate into the NetOps Portal TrustStore

Use the NetOps Portal SSL configuration tool to import the Data Aggregator certificate into the NetOps Portal trustStore.

Follow these steps:

1. Copy the Data Aggregator certificate to the NetOps Portal host.
2. Launch the SSL configuration tool by running the `./SslConfig` command in the following directory:
   PC_Install_Directory`/PerformanceCenter`

   NOTE
   `/opt/CA` is the default installation directory.
3. Complete the following prompts:
   - **Preferred Language**
     Specify a language for the configuration tool.
   - **Options**
     To import the Data Aggregator certificate, specify 4. Confirm your selection.
4. Specify the location and filename of the certificate.
5. Specify the password for the NetOps Portal trustStore.

Revert to the Default Settings

If necessary, use the Data Aggregator SSL configuration tool to revert to the default settings.

Follow these steps:

1. Launch the SSL configuration tool by running the `./sslConfig.sh` command in the following directory:
   DA_Install_Directory`/scripts`

   NOTE
   `/opt/IMDataAggregator`
   is the default installation directory.
2. Select `Disable HTTPS FIPS`.

Enable the Data Aggregator to use SSL Manually

In an environment with a single Data Aggregator, you can enable HTTPS for the Data Aggregator manually. SSL can be used with either a self-signed certificate or a certificate that a trusted Certificate Authority has signed. To set up HTTPS, update the default port and cipher suites.

You can also enable HTTPS for the Data Aggregator in a fault tolerant environment with multiple Data Aggregators. For more information, see Enable Fault Tolerant Data Aggregators to Use SSL.

After you configure the Data Aggregator to use HTTPS, configure the Data Aggregator and NetOps Portal to communicate with HTTPS. After you configure NetOps Portal to use HTTPS, import the Data Aggregator root and intermediate certificates into the NetOps Portal JRE trustStore. Then export the NetOps Portal certificate to the Data

Aggregator. Import the NetOps Portal certificate into the Data Aggregator trustStore. From the NetOps Portal UI, edit the data source scheme and port for the Data Aggregator.

To enable HTTPS for the Data Aggregator, complete the following steps:

1. Configure the Data Aggregator to Use HTTPS
2. Configure the Data Aggregator and Performance Center for HTTPS

Revert the Data Aggregator Back to Use HTTP

If necessary, you can revert the Data Aggregator back to use HTTP.

Follow these steps:

1. Remove the keyStore file from the following location:
   `$DA_install_directory/apache-karaf-version/etc/keystore`
2. Remove the trustStore file from the following location:
   `$DA_install_directory/apache-karaf-version/etc/truststore`
3. Revert the following file:
   `$DA_install_directory/apache-karaf-version/etc/org.ops4j.pax.web.cfg`
4. Revert the following file:
   `$DA_install_directory/apache-karaf-version/etc/jetty.xml`
5. Revert the following file:
   `$DA_install_directory/apache-karaf-version/etc/system.properties`
6. Restart the Data Aggregator service:

```
service dadaemon stop
service dadaemon start
```

Configure the Data Aggregator to Use HTTPS

In an environment with a single Data Aggregator, you can enable HTTPS for the Data Aggregator manually. You cannot enable HTTPS for the Data Aggregator in a fault tolerant environment. SSL can be used with either a self-signed certificate or a certificate that a trusted Certificate Authority has signed. To set up HTTPS, update the default port and cipher suites.

If necessary, you can revert the Data Aggregator back to use HTTP. For more information, see Enable the Data Aggregator to Use HTTPS.

Set Up SSL Certificates for the Data Aggregator

Generate a self-signed certificate or request a certificate from a trusted Certificate Authority.

> **NOTE**
>
> Provide all alternative Data Aggregator host names. If you generate an SSL certificate with the OpenSSL utility, add the alternative host names. Add each host name as a Subject Alternative Name (SAN) in the v3 certificate extension.

To generate an SSL certificate, use the keytool command. Generate a self-signed certificate and install it in the keystore.

Follow these steps:

1. Change the directory:
   `cd DA_install_directory/apache-karaf-version/etc`

2. If a keystore file exists, rename the existing keystore file to create a backup of it:
   ```
   mv DA_install_directory/apache-karaf-version/etc/keystore DA_install_directory/
   apache-karaf-version/etc/keystore.bak
   ```

   WARNING
   Move the old keystore. If you do not, an error appears in later steps: "Keystore was tampered with, or password was incorrect."
3. Generate a private key and a public, self-signed certificate:
   ```
   keytool -genkeypair -ext SAN=dns:fully_qualified_hostname -keystore keystore -
   storepass password -keyalg RSA -keysize 2048 -keypass password -alias alias_name
   ```

   Note your entries for the following variables:
   - ***fully_qualified_hostname***
     Specify the fully qualified host name of the server. Enter the same value when you are prompted for your first and last name.
   - ***keystore***
     Specify the name of the keystore file to create.
   - ***password***
     Specify the password for the keystore and self-signed certificate. Specify a secure password.
   - ***alias_name***
     Specify an alias that can be used to refer to the keystore entry that is created for the self-signed certificate.

     NOTE
     When you are prompted for your first and last name, provide the fully qualified hostname of the server.
4. Proceed through the security prompt questions and confirm your responses.
5. Export the self-signed certificate from the keystore:
   ```
   keytool -exportcert -keystore keystore -storepass password -alias alias_name -
   file filename.cer
   ```

   - ***keystore***
     Specify the same keystore file name previously created.
   - ***password***
     Specify the same password when creating the self-signed certificate.
   - ***alias_name***
     Specify the same alias when creating the self-signed certificate.
   - ***filename*.cer**
     Specify the file to which the certificate is exported. We recommend using a full pathname that does not place the file in the current directory.
     Example: /tmp/DA_Cert.cer

     NOTE
     We recommend backing up any certificates that could be rewritten before continuing.
6. Import the self-signed certificate into the java trusted certificate keystore:
   ```
   keytool -importcert -keystore InstallDirectory/jre/lib/security/cacerts
   -storepass cacertspassword -alias alias_name -file filename.cer
   ```

   - ***cacertspassword***
     Specify the password for the Certificate Authority keystore.

NOTE
The default password for the Certificate Authority keystore is **changeit**.

- ***alias_name***
  Specify the same alias when creating the self-signed certificate.
- ***filename*.cer**
  Specify the file to which the certificate is exported. We recommend using a full pathname that does not place the file in the current directory.
  Example: /tmp/DA_Cert.cer

7. Confirm that you trust the certificate.
8. Back up the certificate file:
   `cp filename.cer filename.cer.bak`
9. (Optional) For more security, change the password of the java trusted certificates keystore:
   `keytool -storepasswd -keystore InstallDirectory/jre/lib/security/cacerts`

   You are prompted to provide the existing keystore password and a new keystore password.
10. Verify that your imported keystore is available:
    `keytool -list -keystore InstallDirectory/jre/lib/security/cacerts`

    WARNING
    To enable the web services, the self-signed certificate must be in the Certificate Authority keystore. Otherwise, you see an error in the log that reports that PKIX did not find a certificate.

    Your self-signed SSL certificate is generated and installed in the keystore.

Import a Certificate

SSL can be used with either a self-signed certificate or a certificate that a trusted Certificate Authority has signed.

Follow these steps:

1. Import the public certificate for the Data Aggregator host to create a keystore:
   `keytool -import -file DA_public_cert -alias alias_name -keystore $DA_install_directory/apache-karaf-version/etc/keystore`

   - ***DA_public_cert***
     Specify the public certificate for the Data Aggregator host.
   - ***alias_name***
     Specify the alias that is used to refer to the certificate.
   - ***DA_install_directory***
     Specify the Data Aggregator installation directory.
   - ***version***
     Specify the version.
2. Specify the keystore password when prompted.
3. (Optional) If a Certificate Authority provided a root certificate, import the root certificate:
   `keytool -import -file root_cert -alias alias_name -keystore $DA_install_directory/apache-karaf-version/etc/keystore`

   - ***root_cert***
     Specify the root certificate for generating the public certificate.

4. (Optional) If a Certificate Authority provided intermediate certificates, import each intermediate certificate:
   `keytool -import -file intermed_cert -alias alias_name -keystore $DA_install_directory/apache-karaf-version/etc/keystore`

   – ***intermed_cert***
   Specify the intermediate certificate for generating the public certificate.

Configure the Data Aggregator Port and Cipher Suites

To set up HTTPS, update the default port and cipher suites.

Follow these steps:

1. Back up the following file:
   DA_install_directory/apache-karaf-*version*/etc/org.ops4j.pax.web.cfg
2. Replace the file contents with the following lines:
   ```
   org.osgi.service.http.enabled=false
   org.osgi.service.http.port.secure=8582
   org.osgi.service.http.secure.enabled=true
   org.ops4j.pax.web.config.file=${karaf.home}/etc/jetty.xml
   ```
3. Back up the following file:
   DA_install_directory/apache-karaf-*version*/etc/jetty.xml
4. Add the following XML within the <Configure> tag:

```
<Call name="addConnector">
   <Arg>
      <New class="org.eclipse.jetty.server.ssl.SslSelectChannelConnector">
         <Arg>
            <New class="org.eclipse.jetty.http.ssl.SslContextFactory">
               <Set name="keyStore"><Property name="jetty.home" default="." />/etc/keystore</Set>
               <Set name="keyStorePassword">changeit</Set>
               <Set name="keyManagerPassword">changeit</Set>
               <Set name="trustStore"><Property name="jetty.home" default="." />/etc/keystore</Set>
               <Set name="trustStorePassword">changeit</Set>
               <Set name="excludeProtocols">
                  <Array type="java.lang.String">
                     <Item>TLSv1</Item>
                     <Item>SSLv3</Item>
                     <Item>SSLv2</Item>
                     <Item>SSLv2Hello</Item>
                  </Array>
               </Set>
               <Set name="includeCipherSuites">
                  <Array type="String">
                     <Item>TLS_DH_RSA_WITH_AES_128_CBC_SHA</Item>
                     <Item>TLS_DH_RSA_WITH_AES_128_CBC_SHA256</Item>
                     <Item>TLS_DH_RSA_WITH_AES_128_GCM_SHA256</Item>
                     <Item>TLS_DH_RSA_WITH_AES_256_CBC_SHA</Item>
                     <Item>TLS_DH_RSA_WITH_AES_256_CBC_SHA256</Item>
                     <Item>TLS_DH_RSA_WITH_AES_256_GCM_SHA384</Item>
                     <Item>TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA</Item>
                     <Item>TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256</Item>
```

```
                        <Item>TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256</Item>
                        <Item>TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA</Item>
                        <Item>TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384</Item>
                        <Item>TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384</Item>
                        <Item>TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA</Item>
                        <Item>TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256</Item>
                        <Item>TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256</Item>
                        <Item>TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA</Item>
                        <Item>TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384</Item>
                        <Item>TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384</Item>
                        <Item>TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA</Item>
                        <Item>TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256</Item>
                        <Item>TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256</Item>
                        <Item>TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA</Item>
                        <Item>TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA384</Item>
                        <Item>TLS_ECDH_ECDSA_WITH_AES_256_GCM_SHA384</Item>
                        <Item>TLS_ECDH_RSA_WITH_AES_128_CBC_SHA</Item>
                        <Item>TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256</Item>
                        <Item>TLS_ECDH_RSA_WITH_AES_128_GCM_SHA256</Item>
                        <Item>TLS_ECDH_RSA_WITH_AES_256_CBC_SHA</Item>
                        <Item>TLS_ECDH_RSA_WITH_AES_256_CBC_SHA384</Item>
                        <Item>TLS_ECDH_RSA_WITH_AES_256_GCM_SHA384</Item>
                        <Item>TLS_PSK_WITH_AES_128_CBC_SHA</Item>
                        <Item>TLS_PSK_WITH_AES_128_CBC_SHA256</Item>
                        <Item>TLS_PSK_WITH_AES_128_GCM_SHA256</Item>
                        <Item>TLS_PSK_WITH_AES_256_CBC_SHA</Item>
                        <Item>TLS_PSK_WITH_AES_256_CBC_SHA384</Item>
                        <Item>TLS_PSK_WITH_AES_256_GCM_SHA384</Item>
                        <Item>TLS_RSA_PSK_WITH_AES_128_CBC_SHA</Item>
                        <Item>TLS_RSA_PSK_WITH_AES_128_CBC_SHA256</Item>
                        <Item>TLS_RSA_PSK_WITH_AES_128_GCM_SHA256</Item>
                        <Item>TLS_RSA_PSK_WITH_AES_256_CBC_SHA</Item>
                        <Item>TLS_RSA_PSK_WITH_AES_256_CBC_SHA384</Item>
                        <Item>TLS_RSA_PSK_WITH_AES_256_GCM_SHA384</Item>
                        <Item>TLS_RSA_WITH_AES_128_CBC_SHA</Item>
                        <Item>TLS_RSA_WITH_AES_128_CBC_SHA256</Item>
                        <Item>TLS_RSA_WITH_AES_128_GCM_SHA256</Item>
                        <Item>TLS_RSA_WITH_AES_256_CBC_SHA</Item>
                        <Item>TLS_RSA_WITH_AES_256_CBC_SHA256</Item>
                        <Item>TLS_RSA_WITH_AES_256_GCM_SHA384</Item>
                        <Item>TLS_AES_128_CCM_8_SHA256</Item>
                        <Item>TLS_AES_128_CCM_SHA256</Item>
                        <Item>TLS_AES_128_GCM_SHA256</Item>
                        <Item>TLS_AES_256_GCM_SHA384</Item>
                    </Array>
                </Set>
            </New>
        </Arg>
        <Set name="port">8582</Set>
        <Set name="maxIdleTime">30000</Set>
      </New>
    </Arg>
</Call>
```

NOTE

The following line specifies the secure port for the Data Aggregator configured to use HTTPS:

```
<Set name="port">8582</Set>
```

5. Edit the `keyStorePassword`, `keyManagerPassword`, and `trustStorePassword` attributes.

NOTE

The `keyManagerPassword` should be the same as the `keyStorePassword`.

6. Back up the following file:
   DA_install_directory/apache-karaf-*version*/etc/system.properties
7. Add the following lines at the end of the file:

```
javax.net.ssl.keyStore=${karaf.home}/etc/keystore
javax.net.ssl.keyStorePassword=changeit
javax.net.ssl.trustStore=${karaf.home}/etc/truststore
javax.net.ssl.trustStorePassword=changeit
```

8. Edit the keystorePassword and the trustStore password.
9. Restart the Data Aggregator service:

```
service dadaemon stop
service dadaemon start
```

Configure the Data Aggregator and Performance Center for HTTPS

After you configure the Data Aggregator to use HTTPS, configure the Data Aggregator and NetOps Portal to communicate with HTTPS.

Configure NetOps Portal Communication

After you configure NetOps Portal to use HTTPS, import the Data Aggregator root and intermediate certificates into the NetOps Portal JRE trustStore. Then export the NetOps Portal certificate to the Data Aggregator.

Follow these steps:

1. Copy the root certificate and the intermediate certificates from the Data Aggregator host to the NetOps Portal host.
2. Import the root certificate to the NetOps Portal JRE trustStore:

```
keytool -import -file root_cert -alias alias_name -keystore $PC_install_directory/
jre/lib/security/cacerts
```

   – ***PC_install_directory***
   Specify the NetOps Portal installation directory.
3. Import each intermediate certificate to the NetOps Portal JRE trustStore:

```
keytool -import -file intermed_cert -alias alias_name -keystore
 $PC_install_directory/jre/lib/security/cacerts
```

4. Restart the NetOps Portal services:

```
service caperfcenter_sso stop
```

```
service caperfcenter_console stop

service caperfcenter_sso start

service caperfcenter_console start
```

5. Export the NetOps Portal certificate:

```
keytool -export -keystore $PC_install_directory/jre/lib/security/cacerts -
alias alias_name -file PC_filename.cer
```

– ***PC_filename.cer***
Specify the filename of the NetOps Portal certificate.

Configure Data Aggregator Communication

Import the self-signed, or root and intermediate certificates that are used to generate the NetOps Portal public certificate into the Data Aggregator trustStore.

Follow these steps:

1. Copy the NetOps Portal certificate from the NetOps Portal host to the Data Aggregator host.
2. Import the NetOps Portal certificate into the Data Aggregator trustStore:

```
keytool -import -file PC_filename.cer -alias alias_name -keystore
 $DA_install_directory/apache-karaf-version/etc/truststore
```

3. Restart the Data Aggregator service:

```
service dadaemon stop

service dadaemon start
```

Edit the Data Aggregator Data Source

From the NetOps Portal UI, edit the data source scheme and port for the Data Aggregator.

Follow these steps:

1. Hover over **Administration**, and click **Data Sources: Data Sources**.
2. Select the Data Aggregator data source, and click **Edit**.
3. Specify 8582 as the secure port for the Data Aggregator.
4. Change the communication protocol to **https**.
5. Click **Save**.

Enable Fault Tolerant Data Aggregators to Use HTTPS

You can enable HTTPS for a data aggregator in a fault-tolerant environment with multiple data aggregators. For more information, see Fault Tolerance.

In an environment with a single data aggregator, you can enable HTTPS for the data aggregator. For more information, see Enable the Data Aggregator to use SSL and Enable the Data Aggregator to use SSL Manually.

SSL can be used with either a self-signed certificate or a certificate that a trusted Certificate Authority has signed.

Follow these steps:

- Configure the Traefik Connection as HTTPS from Performance Center to Proxy Server.
- Configure Consul as HTTPS.
- Configure the Traefik Connection as HTTPS from Proxy Server to Data Aggregator.

Configure the Traefik Connection as HTTPS from Performance Center to Proxy Server

You can enable HTTPS for the data aggregator in a fault tolerant environment with multiple data aggregators. For more information, see Enable Fault Tolerant Data Aggregators to Use HTTPS.

Configure the NetOps Portal to proxy server connection as HTTPS.

Configure the Proxy Server

Manage the keystore file, certificates, and configure the proxy server.

Follow these steps:

1. If a keystore file already exists, back it up:
   Example:
   ```
   cp /opt/CA/daproxy/conf/cacerts /opt/CA/daproxy/conf/cacerts.bak
   ```
2. Generate a private key and a public, self-signed certificate:
   ```
   keytool -genkeypair -ext SAN=dns:fully_qualified_hostname,dns:hostname -keystore keystore -
   storepass store_password -keyalg RSA -keysize 2048 -keypass key_password -alias alias_name
   ```
   NOTE
   Specify the proxy host name both with and without the domain within the `SAN=` argument.
   a. Note your entries for the following variables:
      - **fully_qualified_hostname**
        Specify the fully qualified host name of the server. Enter the same value when you are prompted for your first and last name.
      - **hostname**
        Specify the host name of the server without the domain.
      - **keystore**
        Specify the name of the keystore file to create.
      - **store_password**
        Specify the password for the keystore. Specify a secure password.
      - **key_password**
        Specify the password for the private key. Specify a secure password.
      - **alias_name**
        Specify an alias for the keystore entry created for the self-signed certificate.
        Example: `daproxy`

      Example:
      ```
      keytool -genkeypair -ext SAN=dns:myDaproxyHost.my.domain.net,dns:myDaproxyHost -keystore /opt/CA/
      daproxy/conf/cacerts -storepass changeit -keyalg RSA -keysize 2048 -keypass changeit -alias daproxykey
      ```
   b. Proceed through the security prompt questions and confirm your responses.
      Output:
      ```
      What is your first and last name?
      ```

```
   [Unknown]:  myDaproxyHost.my.domain.net
What is the name of your organizational unit?
   [Unknown]:  MyOrgUnit
What is the name of your organization?
   [Unknown]:  MyOrg
What is the name of your City or Locality?
   [Unknown]:  MyCity
What is the name of your State or Province?
   [Unknown]:  MyStateFullName
What is the two-letter country code for this unit?
   [Unknown]:  US
Is CN=myDaproxyHost.my.domain.net, OU=MyOrgUnit, O=MyOrg, L=MyCity, ST=MyStateFullName, C=US correct?
   [no]:  yes

Warning:
The JKS keystore uses a proprietary format. It is recommended to migrate to PKCS12 which is an
 industry standard format using "keytool -importkeystore -srckeystore /opt/CA/daproxy/conf/cacerts -
destkeystore /opt/CA/daproxy/conf/cacerts -deststoretype pkcs12".
```

3. Verify that the key generated:
   Example:
   ```
   keytool -list -keystore /opt/CA/daproxy/conf/cacerts -storepass changeit |grep daproxy
   ```
   Output:
   ```
   daproxykey, Jun 24, 2020, PrivateKeyEntry,
   ```
4. Export the self-signed certificate from the keystore:
   ```
   keytool -exportcert -keystore keystore -storepass store_password -alias alias_name -file /tmp/filename.cer
   ```
   - **keystore**
     Specify the name of the keystore file that you created previously.
   - **store_password**
     Specify the password for the keystore that you created previously.
   - **alias_name**
     Specify the alias for the keystore entry created previously for the self-signed certificate.
   - **filename.cer**
     Specify the file to which the certificate is exported.

     TIP
     We recommend that you specify a full path that places the file outside the current directory. Before you continue, we recommend that you back up any certificates that might get rewritten.

   Example:
   ```
   keytool -exportcert -keystore /opt/CA/daproxy/conf/cacerts -storepass changeit -alias daproxykey -file /
   tmp/daproxyss.cer
   ```
   Output:
   ```
   Certificate stored in file </tmp/daproxyss.cer>

   Warning:
   The JKS keystore uses a proprietary format. It is recommended to migrate to PKCS12 which is an industry
    standard format using
   "keytool -importkeystore -srckeystore /opt/CA/daproxy/conf/cacerts -destkeystore /opt/CA/daproxy/conf/
   cacerts -deststoretype pkcs12".

   ls -alrt /tmp
          -rw-r--r--.  1 root   root         945 Jun 24 13:56 daproxyss.cer
   ```

5. Extract the private certificate and key as .pem files:
   a. Convert the keystore to pkcs12:

      NOTE
      (Optional) If you have converted the keystore to pkcs12 before, remove or rename the pks12 copy of the keystore:

      ```
      rm -f /opt/CA/daproxy/conf/cacerts.p12
      ```

      ```
      keytool -importkeystore -srckeystore source_keystore -srcstorepass source_store_password
       -srckeypass source_key_password -srcalias source_alias -destalias destination_alias -
      destkeystore destination_keystore -deststoretype PKCS12 -deststorepass destination_store_password -
      destkeypass destination_key_password
      ```

   b. Note your entries for the following variables:
      - **source_keystore**
        Specify the name of the keystore file that you created previously.
      - **source_store_password**
        Specify the password for the keystore that you created previously.
      - **source_key_password**
        Specify the password for the private key that you created previously.
      - **source_alias**
        Specify the alias for the keystore entry created previously for the self-signed certificate.
        Example: `daproxy`
      - **destination_alias**
        Specify an alias for the resulting keystore entry created for the self-signed certificate.
        Example: `daproxy`
      - **destinaion_keystore**
        Specify the name of the resulting keystore file to create.

        TIP
        Specify a full path. Before continuing, back up any destination files that might get rewritten.
      - **destination_store_password**
        Specify the password for the resulting keystore. Specify a secure password.
      - **destination_key_password**
        Specify the password for the resulting private key. Specify a secure password.

      Example:

      ```
      keytool -importkeystore -srckeystore /opt/CA/daproxy/conf/cacerts -srcstorepass changeit -srckeypass
       changeit -srcalias daproxykey -destalias daproxykeyp12 -destkeystore /opt/CA/daproxy/conf/cacerts.p12
       -deststoretype PKCS12 -deststorepass changeit -destkeypass changeit
      ```

      Output:

      ```
      Importing keystore /opt/CA/daproxy/conf/cacerts to /opt/CA/daproxy/conf/cacerts.p12...
      ```

   c. Extract the key to a file:

      NOTE
      (Optional) If you have extracted the key to a file before, remove or rename the file:

      ```
      rm -f /tmp/daproxyss_key.pem
      ```

      Example:

      ```
      openssl pkcs12 -in /opt/CA/daproxy/conf/cacerts.p12 -nodes -nocerts -out /tmp/daproxyss_key.pem
      ```

      Output:

      ```
      Enter Import Password: <enter password value you passed in "-deststorepass" above>
      MAC verified OK
      ```

   d. Write the certificate as a .pem file:

      NOTE
      (Optional) If you have written the certificate as a .pem file before, remove or rename the file:

```
rm -f /tmp/daproxyss.pem
```

Example:

```
openssl x509 -inform der -in /tmp/daproxyss.cer -out /tmp/daproxyss.pem
```

e. Verify the list of files produced:

Example:

```
ls -alrt /tmp/daproxyss*
```

Output:

```
-rw-r--r--. 1 root root  961 Jul 23 11:06 /tmp/daproxyss.cer
-rw-r--r--. 1 root root 1853 Jul 23 11:09 /tmp/daproxyss_key.pem
-rw-r--r--. 1 root root 1359 Jul 23 11:09 /tmp/daproxyss.pem
```

6. (Optional) Request a CA-signed certificate for the proxy server host using the self-signed certificate produced here. Copy and use your CA-signed certificate file(s), including any intermediate and root certificates, in the following steps.
7. Copy the proxy server certificate files for the DAProxy/Traefik service and NetOps Portal:

Example:

```
// copy to daproxy config folder for use by traefik on DAProxy...
cp /tmp/daproxyss.cer /opt/CA/daproxy/conf/daproxyss.cer
cp /tmp/daproxyss.pem /opt/CA/daproxy/conf/daproxyss.pem
cp /tmp/daproxyss_key.pem /opt/CA/daproxy/conf/daproxyss_key.pem

// move the certificate file(s) to a temp folder on PC...
cp /opt/CA/daproxy/conf/daproxyss.cer <myPcHost>/tmp/daproxyss.cer
```

8. Configure the proxy server based on your NetOps Portal version:

20.2.3 and Lower:

a. Back up the following file:

Proxy_Server_Install_Directory/bin/conf/daproxy.cfg

Example:

```
cp /opt/CA/daproxy/conf/daproxy.cfg /opt/CA/daproxy/conf/daproxy.cfg.bak
```

b. Edit the `daproxy.cfg` file and set the following parameters:

Example:

```
vi /opt/CA/daproxy/conf/daproxy.cfg

[entryPoints]
   [entryPoints.da]
         address = ":8582"
         [entryPoints.da.tls]
            minVersion = "VersionTLS11"
            cipherSuites = [
                "TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256",
                "TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256",
                "TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384",
                "TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256",
                "TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256",
                "TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384",
                "TLS_RSA_WITH_AES_128_CBC_SHA256",
                "TLS_RSA_WITH_AES_128_GCM_SHA256",
                "TLS_RSA_WITH_AES_256_GCM_SHA384",
                "TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA",
                "TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA",
                "TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA",
                "TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA",
                "TLS_RSA_WITH_AES_128_CBC_SHA",
```

```
            "TLS_RSA_WITH_AES_256_CBC_SHA"
        ]
        [[entryPoints.da.tls.certificates]]
            certFile = "<da_proxy_server_cert>"
            keyFile = "<da_proxy_server_key>"
```

Example:

The following example shows the `tls.certificates` settings:

```
[[entryPoints.da.tls.certificates]]
    certFile = "/opt/CA/daproxy/conf/daproxyss.pem"
    keyFile = "/opt/CA/daproxy/conf/daproxyss_key.pem"
```

NOTE

Until you configure the fault-tolerant data aggregators for HTTPS, the backend definition in the `daproxy.cfg` file remains `http` .

Example:

```
[backends]
   [backends.da]
        [backends.da.healthcheck]
         path = "/"
         interval = "10s"
        [backends.da.servers.server1]
         url = "http://<da1_hostname>:8581"
        [backends.da.servers.server2]
         url = "http://<da2_hostname>:8581"
```

20.2.4 and Higher:

a. Back up the following file:

Proxy_Server_Install_Directory/bin/conf/daproxy.toml

Example:

```
cp /opt/CA/daproxy/conf/daproxy.toml /opt/CA/daproxy/conf/daproxy.toml.bak
```

b. Edit the `daproxy.toml` file and set the following parameters:

Example:

```
vi /opt/CA/daproxy/conf/daproxy.toml
```

a. Update the `entryPoints` section to set the port to reach the proxy server, usually 8582:

Before Example:

```
[entryPoints]
# Web entry point for traefik on daproxy host
   [entryPoints.web]
        # For HTTPS, change to '8582'.
           address = ":8581"
```

After Example:

```
[entryPoints]
# Web entry point for traefik on daproxy host
   [entryPoints.web]
        # For HTTPS, change to '8582'.
           address = ":8582"
```

b. Update the rule in `http.routers.DARouter` to include the domain name:

Before Example:

```
[http.routers]
     # Define which incoming requests are routed to which service
     [http.routers.DARouter]
          entryPoints = ["web"]
```

```
          # This rule routes all requests to DAService
          # For HTTPS, the hostname here will need to include domain name
          #  if your daproxy host certificate uses fully qualified hostname,
          #  for example:
          #    Host(`yourhost.yourdomain.net`}
          #    ...or to support urls with and without domain...
          Host(`yourhost.yourdomain.net`,`yourhost`}
          #  The hostname used in Performance Center's Data Source
          #   configuration should include domain if your certificate does.
          rule = "Host(`DaproxyHost`) && PathPrefix(`/`)"
          service = "DAService"
```

After Example:

```
[http.routers]
     # Define which incoming requests are routed to which service
     [http.routers.DARouter]
          entryPoints = ["web"]
          # This rule routes all requests to DAService
          # For HTTPS, the hostname here will need to include domain name
          #  if your daproxy host certificate uses fully qualified hostname,
          #  for example:
          #    Host(`yourhost.yourdomain.net`}
          #    ...or to support urls with and without domain...
          Host(`yourhost.yourdomain.net`,`yourhost`}
          #  The hostname used in Performance Center's Data Source
          #   configuration should include domain if your certificate does.
          rule = "Host(`myDaproxyHost.my.domain.net`) && PathPrefix(`/`)"
          service = "DAService"
```

c. Uncomment the following sections:

```
# For HTTPS, uncomment these.
#  [http.routers.DARouter.tls]
#    options = "DAOptions"

# For HTTPS, uncomment these.
#[tls.options]
#  [tls.options.DAOptions]
#    minVersion = "VersionTLS12"
#    cipherSuites = [
#                     "TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256",
#                     "TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256",
#                     "TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384",
#                     "TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256",
#                     "TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256",
#                     "TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384",
#                     "TLS_RSA_WITH_AES_128_CBC_SHA256",
#                     "TLS_RSA_WITH_AES_128_GCM_SHA256",
#                     "TLS_RSA_WITH_AES_256_GCM_SHA384",
#                     "TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA",
#                     "TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA",
#                     "TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA",
#                     "TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA",
#                     "TLS_RSA_WITH_AES_128_CBC_SHA",
#                     "TLS_RSA_WITH_AES_256_CBC_SHA"
```

```
#                ]

# For HTTPS, uncomment these and set path and filenames to
# your daproxy host's cert & key files.
#[[tls.certificates]]
#    certFile = "/opt/CA/daproxy/conf/daproxy_cert.pem"
#    keyFile = "/opt/CA/daproxy/conf/daproxy_key.pem"
#[tls.stores]
#  [tls.stores.default]
#    [tls.stores.default.defaultCertificate]
#      certFile = "/opt/CA/daproxy/conf/daproxy_cert.pem"
#      keyFile = "/opt/CA/daproxy/conf/daproxy_key.pem
```

Example:
The following example shows the `tls.certificates` settings:

```
[[tls.certificates]]
     certFile = "/opt/CA/daproxy/conf/daproxyss.pem"
     keyFile = "/opt/CA/daproxy/conf/daproxyss_key.pem"
[tls.stores]
     [tls.stores.default]
       [tls.stores.default.defaultCertificate]
        certFile = "/opt/CA/daproxy/conf/daproxyss.pem"
        keyFile = "/opt/CA/daproxy/conf/daproxyss_key.pem"
```

NOTE
Until you configure the fault-tolerant data aggregators for HTTPS, the `http.services` section remains the same.
Example:

```
[[http.services.DAService.loadBalancer.servers]]
     # For HTTPS, change to 'https' and '8582'.
     url = "http://myDaHost1:8581"
[http.services.DAService.loadBalancer.servers]]
     # For HTTPS, change to 'https' and '8582'.
     url = "http://myDaHost2:8581"
```

9. Restart the Proxy services:
   ```
   service daproxy stop
   service daproxy start
   ```
10. Test the HTTPS connection to the proxy server:
    `https://<daproxy_hostname>:8582/`

Configure NetOps Portal

Use the NetOps Portal SSL configuration tool to import the proxy server certificate into the NetOps Portal trustStore.

Follow these steps:

1. Copy the proxy server certificate to the NetOps Portal host:
   Example:
   ```
   cp /tmp/daproxyss.cer /opt/CA/daproxyss.cer
   ```
   NOTE
   If you imported a data aggregator or proxy server certificate to the NetOps Portal keystore before, find and remove the existing certificate:
   1. Find the existing certificate in NetOps Portal keystore:
      – **da-cert**

The alias that `SslConfig` uses when importing a data aggregator certificate to NetOps Portal.

Example:

```
keytool -list -keystore /opt/CA/jre/lib/security/cacerts -storepass changeit |grep -i da-cert
```

Output:

```
da-cert, Jul 23, 2020, trustedCertEntry,
```

2. If found, remove the certificate:
   Example:

```
keytool -delete -keystore /opt/CA/jre/lib/security/cacerts -alias da-cert -storepass changeit
```

2. Launch the SSL configuration tool by running the `./SslConfig` command in the following directory:

```
PC_Install_Directory/PerformanceCenter
./SslConfig
```

NOTE
`/opt/CA` is the default installation directory.

Example:

```
cd /opt/CA/PerformanceCenter
./SslConfig
```

NOTE
Launching the SSL configuration tool restarts the `caperfcenter_console` and `caperfcenter_devicemanager` services.

3. Complete the following prompts:
   - **Preferred Language**
     Specify a language for the configuration tool.
   - **Options**
     To import the data aggregator certificate, specify 4. Confirm your selection.
4. Specify the location and filename of the certificate.
   Example: `/opt/CA/daproxyss.cer`
5. Specify the password for the NetOps Portal trustStore.
   Default: `changeit`
6. (Optional) If you used a CA-signed certificate, import any included intermediate and root certificates to the to the NetOps Portal keystore:
   Example:

```
keytool -import -file /tmp/intermed_cert.pem -alias daproxyintermediate -keystore /opt/CA/conf/cacerts -
storepass changeit
```

7. Open NetOps Portal, hover over **Administation: Data Sources**, and then click **Data Sources**.
8. Edit the data aggregator data source:
   - Specify HTTPS.
   - Specify 8582 for the port.
   - Specify the fully qualified domain name for the host (for example, `myDaProxyHost.my.domain.net`).
9. Click **Test** to validate, and then click **Save**.

Troubleshooting

Symptom:

NetOps Portal cannot connect to the proxy server and the following error appears:

```
Caused by: javax.net.ssl.SSLHandshakeException: java.security.cert.CertificateException: No subject
 alternative DNS name matching <myDaproxyHost> found.
```

Solution:

Verify whether you specified the proxy server host name with and without the domain when you generated the key.

Example:

```
keytool -genkeypair -ext SAN=dns:myDaproxyHost.my.domain.net,dns:myDaproxyHost -keystore /opt/CA/daproxy/conf/
cacerts -storepass changeit -keyalg RSA -keysize 2048 -keypass changeit -alias daproxykey.
```

Configure Consul as HTTPS

You can enable HTTPS for the data aggregator in a fault-tolerant environment with multiple data aggregators. For more information, see Enable Fault Tolerant Data Aggregators to Use HTTPS.

Configure Consul on the proxy server to the data aggregators as HTTPs.

Generate the Consul Certificates

On the proxy server, run the following commands to generate the Consul certifications for all Consul hosts.

Follow these steps:

> **NOTE**
> The following steps generate files in the current working directory.

1. Go to the following location:
   ```
   cd Proxy_Install_Directory/daproxy/conf
   ```
   – **Proxy_Install_Directory**
   Specify the installation directory of the proxy server.
   Default:`opt/CA`

   Example:
   ```
   cd /opt/CA/daproxy/conf
   ```
2. Generate the Consul agent certificates:
   ```
   Proxy_Install_Directory/daproxy/bin/consul tls ca create -domain=my.domain.net
   ```
   – **my.domain.net**
   Specify the domain of your host.

   Example:
   ```
   /opt/CA/daproxy/bin/consul tls ca create -domain=my.domain.net
   ```
   Output:
   ```
   ==> Saved my.domain.net-agent-ca.pem
   ==> Saved my.domain.net-agent-ca-key.pem
   ```
3. Generate the Consul server certificates for the Proxy host:
   ```
   Proxy_Install_Directory/daproxy/bin/consul tls cert create -server --additional-dnsname=proxy_host --
   additional-dnsname=fully_qualified_proxy_host -dc=capm -domain=my.domain.net
   ```
   – **proxy_host**
   Specify the host name of the proxy server.
   – **fully_qualified_proxy_host**
   Specify the fully qualified host name of the proxy server.
   – **my.domain.net**
   Specify the domain of your host.

   Example:
   ```
   /opt/CA/daproxy/bin/consul tls cert create -server --additional-dnsname=myDaproxyHost --additional-
   dnsname=myDaproxyHost.my.domain.net -dc=capm -domain=my.domain.net
   ```
   Output:
   ```
   ==> WARNING: Server Certificates grants authority to become a
           server and access all state in the cluster including root keys
   ```

```
        and all ACL tokens. Do not distribute them to production hosts
        that are not server nodes. Store them as securely as CA keys.
==> Using my.domain.net-agent-ca.pem and my.domain.net-agent-ca-key.pem
==> Saved capm-server-my.domain.net-0.pem
==> Saved capm-server-my.domain.net-0-key.pem
```

4. Generate the Consul server certificates for the first data aggregator host:

   Proxy_Install_Directory/daproxy/bin/consul tls cert create -server --additional-dnsname=*DA1_host* --additional-dnsname=*fully_qualified_DA1_host* -dc=capm -domain=*my.domain.net*

   – **DA1_host**
   Specify the host name of the first data aggregator.
   – **fully_qualified_DA1_host**
   Specify the fully qualified host name of the first data aggregator.
   – **my.domain.net**
   Specify the domain of your host.

   Example:

   ```
   /opt/CA/daproxy/bin/consul tls cert create -server --additional-dnsname=myDa1Host --additional-dnsname=myDa1Host.my.domain.net -dc=capm -domain=my.domain.net
   ```

   Output:

   ```
   ==> WARNING: Server Certificates grants authority to become a
           server and access all state in the cluster including root keys
           and all ACL tokens. Do not distribute them to production hosts
           that are not server nodes. Store them as securely as CA keys.
   ==> Using my.domain.net-agent-ca.pem and my.domain.net-agent-ca-key.pem
   ==> Saved capm-server-my.domain.net-1.pem
   ==> Saved capm-server-my.domain.net-1-key.pem
   ```

5. Generate the Consul server certificates for the second data aggregator host:

   Proxy_Install_Directory/daproxy/bin/consul tls cert create -server --additional-dnsname=*DA2_host* --additional-dnsname=*fully_qualified_DA2_host* -dc=capm -domain=*my.domain.net*

   – **DA2_host**
   Specify the host name of the second data aggregator.
   – **fully_qualified_DA2_host**
   Specify the fully qualified host name of the second data aggregator.
   – **my.domain.net**
   Specify the domain of your host.

   Example:

   ```
   /opt/CA/daproxy/bin/consul tls cert create -server --additional-dnsname=myDa2Host --additional-dnsname=myDa2Host.my.domain.net -dc=capm -domain=my.domain.net
   ```

   Output:

   ```
   ==> WARNING: Server Certificates grants authority to become a
           server and access all state in the cluster including root keys
           and all ACL tokens. Do not distribute them to production hosts
           that are not server nodes. Store them as securely as CA keys.
   ==> Using my.domain.net-agent-ca.pem and my.domain.net-agent-ca-key.pem
   ==> Saved capm-server-my.domain.net-2.pem
   ==> Saved capm-server-my.domain.net-2-key.pem
   ```

Copy the Consul Certificate Files

Copy the Consul certificate files from the proxy server to the data aggregators.

Follow these steps:

1. Run the following command to get the shared folder name:

   NOTE
   You can use the home folder of your fault tolerant environment to store certificate files for transfer between the Data Aggregators and the proxy server. The home folder is defined in the `DA.cfg` file on the Data Aggregator as shown in following output example.

   ```
   cat /etc/DA.cfg |grep -i da.data.home=
   ```

   Output:

   ```
   da.data.home=/myDaDataHome
   ```

2. Copy the files from the proxy server to the shared folder:
   Example:

   ```
   cp /opt/CA/daproxy/conf/my.domain.net-agent-ca.pem /myDaDataHome/my.domain.net-agent-ca.pem
   cp /opt/CA/daproxy/conf/capm-server-my.domain.net-1.pem /myDaDataHome/capm-server-my.domain.net-1.pem
   cp /opt/CA/daproxy/conf/capm-server-my.domain.net-1-key.pem /myDaDataHome/capm-server-my.domain.net-1-key.pem
   cp /opt/CA/daproxy/conf/capm-server-my.domain.net-2.pem /myDaDataHome/capm-server-my.domain.net-2.pem
   cp /opt/CA/daproxy/conf/capm-server-my.domain.net-2-key.pem /myDaDataHome/capm-server-my.domain.net-2-key.pem
   ```

3. On the first data aggregator, copy the agent certificate of the proxy server, and the server certificate of this data aggregator from the shared folder:
   Example:

   ```
   cp /myDaDataHome/my.domain.net-agent-ca.pem /opt/IMDataAggregator/consul/conf/my.domain.net-agent-ca.pem
   cp /myDaDataHome/capm-server-my.domain.net-1.pem /opt/IMDataAggregator/consul/conf/capm-server-my.domain.net-1.pem
   cp /myDaDataHome/capm-server-my.domain.net-1-key.pem /opt/IMDataAggregator/consul/conf/capm-server-my.domain.net-1-key.pem
   ```

4. On the second data aggregator, copy the agent certificate of the proxy server, and the server certificate of this data aggregator from the shared folder:
   Example:

   ```
   cp /myDaDataHome/my.domain.net-agent-ca.pem /opt/IMDataAggregator/consul/conf/my.domain.net-agent-ca.pem
   cp /myDaDataHome/capm-server-my.domain.net-2.pem /opt/IMDataAggregator/consul/conf/capm-server-my.domain.net-2.pem
   cp /myDaDataHome/capm-server-my.domain.net-2-key.pem /opt/IMDataAggregator/consul/conf/capm-server-my.domain.net-2-key.pem
   ```

Configure Consul

Configure Consul on the proxy server and each data aggregator.

Follow these steps:

NOTE
Perform the following steps on the proxy server and each of the data aggregators.

1. Back up the following file:
   Proxy server:

   ```
   cp Proxy_install_directory/daproxy/conf/config.json Proxy_install_directory/daproxy/conf/config.json.orig
   ```

   Example:

   ```
   cp /opt/CA/daproxy/conf/config.json /opt/CA/daproxy/conf/config.json.orig
   ```

   Data aggregator:

   ```
   cp DA_install_directory/consul/conf/config.json DA_install_directory/consul/conf/config.json.orig
   ```

   Example:

```
cp /opt/IMDataAggregator/consul/conf/config.json /opt/IMDataAggregator/consul/conf/config.json.orig
```

2. Edit the `config.json` file to replace `"ports": {"http":8500}'` with the following parameters:
   Example:

```
"bind_addr": "x.x.x.x",
"ports": {
"http": -1,
"https": 8443
},
"verify_incoming": false,
"verify_outgoing": true,
"verify_server_hostname": true,
"ca_file": "/opt/CA/daproxy/conf/my.domain.net-agent-ca.pem",
"cert_file": "/opt/CA/daproxy/conf/capm-server-my.domain.net-0.pem",
"key_file": "/opt/CA/daproxy/conf/capm-server-my.domain.net-0-key.pem",
"auto_encrypt": {
"allow_tls": true
},
"tls_min_version": "tls11",
"tls_cipher_suites":
 "TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA38
```

   - **bind_addr**
     Specifies the IP address of this host, which is reachable from the data aggregator and proxy server hosts.
   - **ca_file**
     Specifies the Consul agent certificate file generated on the proxy server. Use the same file on all Consul hosts.
   - **cert_file**
     Specifies the Consul server certificate generated on the proxy server specifically for this host.
   - **key_file**
     Specifies the Consul server key generated on the proxy server specifically for this host.
3. Append the domain of the current host to the Consul service startup arguments:
   Example:

```
vi /etc/systemd/system/consul.service
          ExecStart=<existing args> -domain my.domain.net
systemctl daemon-reload
```

   - **my.domain.net**
     Specifies the domain of this host.
4. (Optional) To enable Consul logging, append logging arguments to the Consul service using the install path of the current host:
   Example:

```
vi /etc/systemd/system/consul.service
          ExecStart=<existing args> -log-file=/opt/IMDataAggregator/consul/log/ -log-rotate-duration=24h -
log-rotate-max-files=7
systemctl daemon-reload
```

Configure the Data Aggregators

The following steps apply only to the data aggregators, and not to the proxy server.

Follow these steps:

> **NOTE**
> Perform the following steps for each data aggregator.

1. Back up the following file:

`/etc/DA.cfg`

Example:

```
cp /etc/DA.cfg /etc/DA.cfg.orig
```

2. Edit the `DA.cfg` file, and set the following parameters:

```
da.consul.port=8443
da.consul.secure=true
```

3. Back up the following file:

```
DA_install_directory/consul-ext/bin/start-consul-ext.sh
```

NOTE
`/opt/IMDataAggregator` is the default installation directory.

Example:

```
cp /opt/IMDataAggregator/consul-ext/bin/start-consul-ext.sh /opt/IMDataAggregator/consul-ext/bin/start-consul-ext.sh.orig
```

4. Edit the `start-consul-ext.sh` file and set the following parameters:

```
CONSUL_PORT=8443
CONSUL_SECURE=true
```

5. Import the agent certificate of the proxy server into the data aggregator's Java keystore.

Example:

```
/opt/IMDataAggregator/jre/bin/keytool -import -file /opt/IMDataAggregator/consul/conf/my.domain.net-agent-ca.pem -alias daproxyagent -keystore /opt/IMDataAggregator/jre/lib/security/cacerts -storepass changeit
```

Restart Consul

Restart the Consul services on the proxy server and data aggregators.

Follow these steps:

1. Stop the Consul service on the proxy server:

```
service consul stop
```

2. Stop the Consul service on the data aggregators:

```
service consul stop
service consul-ext stop
```

3. Start the Consul service on the proxy server:

```
service consul start
```

4. Start the Consul service on the data aggregators:

NOTE
Start the service on the active data aggregator first.

```
service consul start
service consul-ext start
```

5. Verify that Consul is responding to HTTPS requests on port 8443.
The following URL should show Consul cluster members:

```
https://myDaproxyHost.my.domain.net:8443/v1/agent/members?segment=_all
```

Import the Consul Certificate

Import the Consul server certificate from the proxy server to the NetOps Portal server.

Follow these steps:

1. Copy the Consul server certificate file, `capm-server-my.domain.net-0.pem` from the proxy server to NetOps Portal:

Example:

```
cp capm-server-my.domain.net-0.pem /tmp/capm-server-my.domain.net-0.pem
```

2. Import the certificate into NetOps Portal:
   Example:
   ```
   keytool -import -file /tmp/capm-server-my.domain.net-0.pem -alias consulserverlvndaproxy -keystore /opt/CA/conf/cacerts -storepass changeit
   ```
3. Restart the NetOps Portal console service:
   ```
   service caperfcenter_console.service stop
   service caperfcenter_console.service start
   ```

Configure the Traefik Connection as HTTPS from Proxy Server to Data Aggregator

You can enable HTTPS for the data aggregator in a fault tolerant environment with multiple data aggregators.

For more information, Enable Fault Tolerant Data Aggregators to Use HTTPS.

Configure Traefik on the proxy server to the data aggregators as HTTPS.

> **IMPORTANT**
> The following configuration steps apply to 20.2.4 and higher. For more information, see Upgrade Fault Tolerant Data Aggregators.

Configure the Data Aggregators

Enable HTTPS on each data aggregator.

Follow these steps:

1. Run the following command:
   ```
   cd DA_install_directory/scripts
   ./sslConfig.sh
   ```
   > **NOTE**
   > `/opt/IMDataAggregator` is the default installation directory.

   Example:
   ```
   cd /opt/IMDataAggregator/scripts
   ./sslConfig.sh
   ```
   > **NOTE**
   > When prompted for the "common name for the certificate," use the fully qualified host name of your data aggregator (for example, "`myDaHost.my.domain.net` ").

   Output:
   ```
   ----------------------------------------
   Data Aggregator HTTPS FIPS Configurator
   ----------------------------------------

   Please choose an option:
   1 - Enable HTTPS FIPS
   2 - Disable HTTPS FIPS
   q - Quit

   ------------------------------------------
   HTTPS FIPS status                : DISABLED
   HTTP Connection status [port 8581] : SUCCESS
   ------------------------------------------

   Your choice: [Default=1]: 1
   ```

```
Would you like to proceed with enabling HTTPS FIPS on the Data Aggregator? [y/n, Default=n]: y
Do you have an existing signed certificate and key? [y/n, Default=n]: n
CA temporary files will be located in the /root/cadafips directory.
Specify the HTTPS port for secure communication [8582]:
Specify the name of your company [test_company]:MyCompany
Specify the name of your organizational unit [test_org_unit]:MyOrgUnit
Specify a common name for the certificate [test_common_name]:myDaHost.my.domain.net
Specify a two-letter country code [XX]:US
Specify your state or province [test_state]:MyStateFullName
Specify the name of your locality, city, or town [test_locality]:MyCity
Specify an email address for the certificate [test@test.abc]:my.email@mydomain.com
Generate new certificate.
Generating a 2048 bit RSA private key
.....................................................................................+++
.....................................+++
writing new private key to 'server.key'
-----
Signature ok
subject=/C=US/ST=MyStateFullName/L=MyCity/O=MyCompany/OU=MyOrgUnit/CN=myDaHost.my.domain.net/
emailAddress=my.email@mydomain.com
Getting Private key
Create PKCS12 keystore with the server certificate
Transform the keystore into JKS format
Importing keystore server.pkcs12 to keystore.jks...
Entry for alias 1 successfully imported.
Import command completed:  1 entries successfully imported, 0 entries failed or cancelled

Warning:
The JKS keystore uses a proprietary format. It is recommended to migrate to PKCS12 which is an industry
 standard format using "keytool -importkeystore -srckeystore keystore.jks -destkeystore keystore.jks -
deststoretype pkcs12".
Create truststore with server certificate
Certificate was added to keystore
Saving the original keystore to /opt/IMDataAggregator//apache-karaf-2.4.3/etc/keystore.bak
Saving the original truststore to /opt/IMDataAggregator//apache-karaf-2.4.3/etc/truststore.bak
Copy keystore and truststore to /opt/IMDataAggregator//apache-karaf-2.4.3/etc directory
Modify org.ops4j.pax.web.cfg
Modify jetty.xml
Modify system.properties.
Stopping DA daemon. This may take a while. Please be patient.
Redirecting to /bin/systemctl stop dadaemon.service

------------------------------------------------------------------
SSL is now enabled.
DataAggregator should be available in HTTPS at https://<da_hostname>:8582
------------------------------------------------------------------
Press enter to back to menu.
```

2. Verify that your key generated:
   Example:

   NOTE
   The following example uses "alias 1" because the `sslConfig.sh` script creates the key using that alias.

```
keytool -list -keystore /opt/IMDataAggregator/apache-karaf-2.4.3/etc/keystore -storepass changeit -alias 1
```

Output:

```
1, Sep 17, 2020, PrivateKeyEntry,
Certificate fingerprint (SHA1): 81:47:DE:CF:BC:15:DA:FB:57:1B:2A:01:C9:17:E7:F8:6E:AE:EE:EA

Warning:
The JKS keystore uses a proprietary format. It is recommended to migrate to PKCS12 which is an industry
 standard format using "keytool -importkeystore -srckeystore /opt/IMDataAggregator/apache-karaf-2.4.3/etc/
keystore -destkeystore /opt/IMDataAggregator/apache-karaf-2.4.3/etc/keystore -deststoretype pkcs12".
```

3. Copy the certificate files from the data aggregator for the DAProxy/Traefik service to use later:

   NOTE
   (Optional) Request a CA-signed certificate for this data aggregator using the self-signed certificate produced here. Copy and use your CA-signed certificate file(s), including any intermediate and root certificates, in the following steps.

   Example:

   NOTE
   In the following example, `server.cert` is the self-signed certificate that the `sslConfig.sh` script created. Change file names containing "da1" to "da2" for subsequent data aggregators.

   ```
   cp /root/cadafips/server.crt <myDAProxyHost>/tmp/da1_cert.pem
   ```

   Optional:

   NOTE
   If you requested a CA-signed certificate, include any provided intermediate and root cert files here.

   ```
   cp /tmp/IntermediateCA.crt <myDAProxyHost>/tmp/IntermediateCA.crt
   ```

4. Import the data aggregator certificate to the Java keystore for use by the consul-ext service.
   In the following example, `server.cert` is the self-signed certificate that the `sslConfig.sh` script created:

   ```
   /opt/IMDataAggregator/jre/bin/keytool -import -file /root/cadafips/server.crt -alias dacert -keystore /
   opt/IMDataAggregator/jre/lib/security/cacerts -storepass changeit
   ```

5. (Optional) If you used a CA-signed certificate for this DA, import any included intermediate and root certificates to the java keystore.

   Example:

   ```
   /opt/IMDataAggregator/jre/bin/keytool -import -file /tmp/IntermediateCA.crt -alias daintermediate -
   keystore /opt/IMDataAggregator/jre/lib/security/cacerts -storepass changeit
   ```

6. Back up the `checkDAStatus.groovy` file.

   Example:

   ```
   cp /opt/IMDataAggregator/consul-ext/scripts/checkDAStatus.groovy /opt/IMDataAggregator/consul-ext/scripts/
   checkDAStatus.groovy.orig
   ```

7. Edit the `checkDAStatus.groovy` file to use the fully qualified host name.

   NOTE
   This assumes you specified the fully qualified host name when generating the certificates for this data aggregator host.

   Example:

   ```
   vi /opt/IMDataAggregator/consul-ext/scripts/checkDAStatus.groovy
   ```

   Change this:

   ```
   hostname = InetAddress.getLocalHost().getHostName()
   ```

   To this:

   ```
   hostname = InetAddress.getLocalHost().getCanonicalHostName()
   ```

Configure the Proxy Server

Update the proxy server configuration.

Follow these steps:

1. Copy the data aggregator certificates to the trusted root certificate folder:
   Example:
   ```
   cp /tmp/da1_cert.pem /etc/pki/ca-trust/source/anchors/da1_cert.pem
   cp /tmp/da2_cert.pem /etc/pki/ca-trust/source/anchors/da2_cert.pem
   ```
   Optional:
   NOTE
   If you requested a CA-signed certificate, include any provided intermediate and root cert files here.
   ```
   cp /tmp/IntermediateCA.crt /etc/pki/ca-trust/source/anchors/IntermediateCA.crt
   ```
2. To update the trusted root certificates, run the following command:
   ```
   update-ca-trust extract
   ```
3. Update the proxy server configuration:
   NOTE
   The following steps configure the connection from the proxy server to the data aggregators only. Do not modify the other "HTTPS" sections of the `daproxy.toml` file at this time.
   Example:
   ```
   cd /opt/CA/daproxy/conf
   vi daproxy.toml
   ```
4. Update the "url" values to use "https", the host names that include the domain, and the port values for your data aggregators:
   Example:
   ```
   [http.services]
       [http.services.DAService.loadBalancer]
           [http.services.DAService.loadBalancer.healthcheck]
            path = "/"
            interval = "10s"
           [[http.services.DAService.loadBalancer.servers]]
             # Change to https and 8582 when configuring HTTPS
             url = "https://myDaHost1.my.domain.net:8582"
           [[http.services.DAService.loadBalancer.servers]]
             # Change to https and 8582 when configuring HTTPS
             url = "https://myDaHost2.my.domain.net:8582"
   ```
5. Restart the DAProxy service:
   ```
   service daproxy stop
   service daproxy start
   ```
6. Do one of the following steps to verify your connection:
   - If you have not configured NetOps Portal to the proxy server connection as HTTPS, access the following URL:
     Example:
     ```
     https://myDAProxyHost:8581
     ```
   - If you have configured NetOps Portal to the proxy server connection as HTTPS, access the following URL:
     Example:
     ```
     https://myDAProxyHost.my.domain.net:8582
     ```
   - Open NetOps Portal, hover over **Administation: Data Sources**, click **Data Sources**, **Data Aggregator**, and **Test**.

Enable Single Sign-On Spoofing Protection

For enhanced security, enable Single Sign-On URL spoofing protection. When this protection is enabled, users can log in using only authorized URLs. The Log In button is disabled for all other URLs. Otherwise, users get only a warning regarding suspicious URLs.

The list of available authorized URLs is populated based on successful log in attempts before protection is enabled.

Follow these steps:

1. Hover over **Administration**, and click **Configuration Settings: Single Sign-On Settings**.
2. Select **Enable SSO URL Protection**.
3. Select Authorized URLs.
4. Click **Save**.

Obfuscate Jetty Passwords

The password values for the `jetty.sslContext.keyStorePassword` , `jetty.sslContext.keyManagerPassword` , and `jetty.sslContext.trustStorePassword` parameters are stored in plain text by default in the `PC/start.d/ssl.ini` file and the `sso/start.d/ssl.ini` file. However, you can obfuscate them.

Follow these steps:

1. Run the following command to generate a secured version of the password:

```
$ cd /opt/CA/PerformanceCenter/jetty

$ java -cp lib/jetty-util-
version
.jar org.eclipse.jetty.util.security.Password
password
```

– ***version*** The installed version of jetty

Example output:

```
$ java -cp lib/jetty-util-9.4.7.v20170914.jar
 org.eclipse.jetty.util.security.Password password_to_encrypt

2018-03-16 17:37:01.212:INFO::main: Logging initialized @148ms to
 org.eclipse.jetty.util.log.StdErrLog

password_to_encrypt

OBF:1y7v1ugg1zsp1yf41w9f1wu81vnw1rpg1tv31z0f1tvz1rp61vn61wu61w8b1yf21zt11uha1y83

MD5:403956d7e303ee5f1c3714041a74e0fa
```

2. Open the `PC/start.d/ssl.ini` file.
3. Replace the plain text password with the entire obfuscated value including "`OBF` ":

```
# SSL
# define the port to use for secure redirection
```

```
jetty.ssl.port=8182
jetty.https.port=8182
jetty.httpConfig.securePort=8182
# Set up a keystore and truststore
jetty.sslContext.keyStoreType=JKS
jetty.sslContext.keyStorePath=etc/keystore_file.ks
jetty.sslContext.trustStorePath=etc/keystore_file.ks
# Set up passwords
jetty.sslContext.keyStorePassword=
password
jetty.sslContext.keyManagerPassword=
password
jetty.sslContext.trustStorePassword=
password
```

Example:

```
# SSL
# define the port to use for secure redirection
jetty.ssl.port=8182
jetty.https.port=8182
jetty.httpConfig.securePort=8182
# Set up a keystore and truststore
jetty.sslContext.keyStoreType=JKS
jetty.sslContext.keyStorePath=etc/keystore_file.ks
jetty.sslContext.trustStorePath=etc/keystore_file.ks
# Set up passwords

jetty.sslContext.keyStorePassword=OBF:1y7v1ugg1zsp1yf41w9f1wu81vnw1rpg1tv31z0f1tvz1rp61vn61wu
jetty.sslContext.keyManagerPassword=OBF:1y7v1ugg1zsp1yf41w9f1wu81vnw1rpg1tv31z0f1tvz1rp61vn61
jetty.sslContext.trustStorePassword=OBF:1y7v1ugg1zsp1yf41w9f1wu81vnw1rpg1tv31z0f1tvz1rp61vn61
```

Update Single Sign-On Website Settings

You can change the default settings for Single Sign-On using the Single Sign-On Configuration Tool. For example, you can change the virtual directory for the Single Sign-On website.

You can change other settings that affect Single Sign-On behavior when users attempt to log in. Some parameters also affect user interface (UI) behavior, such as the timeout period that logs the user out automatically in response to inactivity.

> **WARNING**
> Updates to the Single Sign-On website only affect CA data source products that are running on the same server because of the distributed architecture of the software.

Follow these steps:

1. Log in to the server where NetOps Portal or a CA data source product is installed.

Log in as root or with the `sudo` command.

2. Launch the Single Sign-On Configuration Tool by running the `./SsoConfig` command in the following directory:
   InstallDirectory`/PerformanceCenter`

   NOTE
   `/opt/CA` is the default installation directory.

   You are prompted to select an option. The available options correspond to CA applications running on the local server.
   Use the following commands as needed while you are selecting settings:
   - `q` (quit)
   - `b` (go back to the previous menu)
   - `u` (update)
   - `r` (reset)
3. Enter **1** to configure NetOps Portal.
4. Enter **4** for Single Sign-On.
   You are prompted to specify the priority.
   The `Priority` parameter only applies to NetOps Portal.
5. Enter one of the following options:
   - **1. Remote Value**
     These settings are propagated to all other CA applications and data sources that are registered to this instance of NetOps Portal. This includes the Event Manager in NetOps Portal, which embeds the URL of NetOps Portal. NetOps Portal uses `Remote Value` settings only if a corresponding `Local Override` value is not present.
   - **2. Local Override**
     Overrides a setting on this NetOps Portal instance, which does not propagate to other CA applications and data sources (including Event Manager) registered to this instance of NetOps Portal. `Local Override` takes precedence over both the `Remote Value` and default settings.

   You are prompted to select a property to configure.

   NOTE
   Configure the scheme or port using `Remote Value` to include the correct Performance Center URL in threshold event email messages.
6. Enter one or more of the following properties. When prompted, enter `u` to update the value and supply a new value:
   - **1. Anonymous User Enabled**
     Specifies whether the Sign-In page appears when users attempt to log in to a data source product interface. A value for the `Anonymous User ID` parameter is required if this parameter is enabled. Users do not see the Sign-In page when they attempt to log in. They are logged in as the user associated with the `Anonymous User ID` parameter.
     The `Localhost User Enabled` parameter takes precedence when the following conditions are met:
     - The user is logging in from the Single Sign-On server.
     - The `Localhost User Enabled` and `Anonymous User Enabled` parameters are enabled.

     Default: Disabled

     NOTE
     The Anonymous User login takes precedence over Windows Authentication.
   - **2. Anonymous User ID**
     Specifies the username that is used to authenticate the user automatically, bypassing the Sign-In page. This parameter is used only if the `Anonymous User Enabled` parameter is enabled. Select one of the following values:
     - **1 -** The username for the default administrator account (admin).
     - **2 -** The username for the default user account (user).
     - Another username that exists in the NetOps Portal database.
   - **3. Localhost User Sign-In Page Enabled**

Specifies whether the Sign-In page appears when users log in from the server where Single Sign-On is installed. If this parameter is enabled, the Sign-In page appears, even if users log in from the server where Single Sign-On is installed.
If this parameter is disabled, the following rules apply:

- The `Localhost User Enabled` parameter must be enabled.
- The value for the `Localhost User ID` parameter must contain a valid product username. This value is used to log in the user to the UI, bypassing the Sign-In page.

Default: Disabled

– **4. Localhost User Enabled**
Specifies whether users are automatically signed in--bypassing the Sign-In page--when they are logging in from the server where Single Sign-On is installed. A value for the `Localhost User ID` parameter is required if this parameter is enabled.

- If the `Localhost User Sign-In Page Enabled` parameter is enabled, this parameter is used in cases where users click **Sign In** without entering a username or password. Users are then logged in to the application as the user associated with the `Localhost User ID` parameter.
- If users do supply a username and password, those credentials are used for authentication.
- If this parameter is enabled but the `Localhost User Sign-In Page Enabled` parameter is disabled, users bypass the Sign-In page. Users are instead logged in to the UI using the value of the `Localhost User ID` parameter.
- If users log in from the server where Single Sign-On is installed and the `Localhost User Enabled` and `Anonymous User Enabled` parameters are enabled, the `Localhost User Enabled` parameter takes precedence.

Default: Disabled

– **5. Localhost User ID**
Specifies the user ID that is used to authenticate users automatically--bypassing the Sign-In page--when they log in to the server where Single Sign-On is installed. This parameter is used only if the `Localhost User Enabled` parameter is enabled. Enter one of the following values:
1 - The username for the default administrator account (admin).
2 - The username for the default user account (user).

– **6. Cookie Timeout Minutes**
Specifies the number of minutes that pass before a Single Sign-On cookie expires. Each time a user performs an action in a data source product interface, the cookie timeout resets. If the timeout expires, the user is logged out and must reauthenticate.
Default: 20 minutes

– **7. Encryption Decryption Key**
Specifies the key that is used to encrypt and decrypt a Single Sign-On cookie.

– **8. Encryption Algorithm**
Specifies the encryption algorithm that is used to encrypt and decrypt a Single Sign-On cookie. Enter **DES** or **AES** as the value.

> **WARNING**
> For FIPS-compliant encryption, use AES encryption. DX NetOps Performance Management is not fully FIPS-compliant. This feature is for FIPS-compliant encryption only and does not meet full FIPS compliance. CA Application Delivery Analysis, Network Flow Analysis, and CA Unified Communications Monitor do not support AES encryption.
> For more information, see FIPS-Compliant Encryption.

– **9. Failed Sleep Seconds**
Specifies the number of seconds Single Sign-On waits after a failed sign-in attempt.

– **10. Remember Me Enabled**
Specifies whether the **Remember Me** check box is displayed on the Sign-In page. When the **Remember Me** check box is selected, Single Sign-On uses the value for `Remember Me Timeout Days` to determine when the Single

Sign-On token expires. If the `Remember Me Enabled` parameter is disabled, a user is automatically logged out when the Single Sign-On token expires after `Cookie Timeout Minutes` . When using SAML, if the `Remember Me Enabled` parameter is enabled, the Single Sign-On token uses the value for `Remember Me Timeout Days` to set its expiration. If the `Remember Me Enabled` parameter is disabled, the Single Sign-On token uses `Cookie Timeout Minutes` to set its expiration.
Default: Enabled

- **11. Remember Me Timeout Days**
  Specifies the number of days that pass before a user who selected the **Remember Me** check box on the Sign-In page must re-authenticate. This parameter is only used if the `Remember Me Enabled` parameter is enabled. A value of **0** indicates that the **Remember Me** setting does not expire; the user must click the **Sign Out** link in a data source product interface.
- **12. Scheme**
  Specifies the URL scheme that data source products can use to access the Single Sign-On application. If you are using SSL, enter **https:** as the value.
- **13. Port**
  Specifies the URL port that data source products can use to access the Single Sign-On application.
- **14. Virtual Directory**
  Specifies the name of the virtual directory for Single Sign-On.

  NOTE
  The virtual directory is required to use an encryption scheme for communications among CA servers.

  Default: SingleSignOn

  NOTE
  Changing the value for any of the previous parameters does not replace the default value, but the new value now takes precedence. The new value is actually a Local Override.
- **15. Allow Single Sign-On in a frame (Local Override)**
  Determines whether Single Sign-On can display within a frame in a web page.
  Default: Disabled

  NOTE
  With single sign-on in frames set to disabled, the browser view is functional only for pages that are also on the server where Single Sign-On is installed.

7. Enter **b** after you have finished changing the default settings.
   You return to the previous set of options.
8. Enter **b** again to go back to the first set of options.
9. Enter **q** to close the Single Sign-On Configuration Tool.
   The Single Sign-On Configuration Tool closes.
   NetOps Portal directs all unauthenticated users to the Single Sign-On website using the new values that you supplied.

Add Custom HTTP Headers

For enhanced security, in 3.7.9 and higher only, you can add custom HTTP headers with the SSO configuration tool.

Follow these steps:

1. Launch the SSO configuration tool by running the '`./SSOConfig`' command in the following directory:
   PC_Install_Directory`/PerformanceCenter`

   NOTE
   `/opt/CA`
   is the default installation directory.
2. Specify `1` for **CA Performance Center**.

3. Specify `3` for **Performance Center**.
4. Specify `2` for **Local Override**.
5. Specify `21` for **Custom HTTP Headers**.
6. Specify `u` to update the setting.
7. Specify the following code:

```
X-XSS-Protection:1; mode=block|X-Content-Type-Options:nosniff|Strict-Transport-Security:max-age=31536000;
 includeSubDomains
```

   NOTE
   A vertical bar(|) separates the headers. The headers contain the header name and value separated by a colon(:).
8. Specify `q` to exit the SSO configuration tool.
   Responses now include your custom headers and these settings remain after upgrade.

Logs

To investigate issues, review any log messages that occurred around the time of the issue. Each DX NetOps Performance Management component includes separate log files:

Support often requests log files or changes the log configuration. CARE collects the log files.

Data Aggregator Logs

The Data Aggregator supports standard log4j configuration and logging levels. Use the following file to configure the logs:
`/opt/IMDataAggregator/apache-karaf-2.4.3/etc/org.ops4j.pax.logging.cfg`

Data Aggregator logs are available in the following directory:
`/opt/IMDataAggregator/apache-karaf-2.4.3/data/log`

The following logs provide potentially useful information.

- `Exception.log`
  This log includes exceptions and warnings. When the system is operating normally, this log contains no messages. Exceptions in this log require attention from CA Support.
  If an exception is repeated, recurring entries in the log do not include the stack trace. A recurrence count indicates how many times the exception has occurred.
- `karaf.log`
  This log is a catch all for Data Aggregator information.
- `LongRunningAndFailedQueries.log`
  This log lists errors and information messages related to database queries.
  – Long running queries are requests that take longer than 60 seconds to complete.
  – Queries time out and fail after 110 seconds.
- `PollSummary.log`
  This log shows poll responses that the data aggregator receives from the Data Collectors. Use this log to verify that Data Aggregator is receiving poll responses.

By default, the Data Aggregator supports 1-10 backups with a max files size of 100 MB.

Undocumented logs in the log directory are for internal use only.

Performance Center Logs

NetOps Portal log filenames include the relevant date and time. New log files are generated automatically each day. Older log files are removed automatically after 14 days to avoid consuming excessive disk space.

Access the most recent log file to find errors that are associated with the database or data source synchronization. You can start by opening the Events dashboard from the Dashboards tab and sorting by Status. If you want to look at the related log file, note the event type and failure date and time. In the log directory, open the log file with the corresponding date in the filename.

The following logs are stored in subfolders that correspond to a service (or daemon). Find the following log files in the following path:
`CA/PerformanceCenter/servicename/logs`

Replace the ***servicename*** parameter with one of the following service names:

- **DM**
  The Device Manager
  - **DMService.log**
    Output from the Device Manager, primarily related to synchronization.
  - **wrapper.log**
    caperfcenter_devicemanager process logging.
- **EM**
  The Event Manager
  - **EMService.log**
    Output from the Event Manager; includes details of events and alarms.
  - **wrapper.log**
    caperfcenter_eventmanager process logging.
- **PC**
  The main console program
  - **PCService.log**
    NetOps Portal-related logging; comprises user interface and view components.
  - **wrapper.log**
    caperfcenter_console process logging.
- **SSO**
  The Single Sign-On authentication software
  - **SSOService.log**
    Single Sign-On logging, including HTTPS (Secure Sockets Layer) information where HTTPS has been configured.
  - **wrapper.log**
    caperfcenter_sso process logging.

  For problems with the Single Sign-On Configuration Tool, view the following application log:
  `/opt/CA/PerformanceCenter/sso/logs/application.log`

The MySQL error log is stored by default in the following path:

`/opt/CA/MySql/data`

- **MySQL**
  - ***hostname*.err**
    The hostname is the name of the system. This file contains errors related to MySQL.

SSO Audit Log

To support security auditing, Single Sign-On logs details about user login activity to a file. Each time a user attempts to log in to the NetOps Portal login page, SSO logs an entry in the audit log. If SSO redirects the user to a SAML2 server to log

in or re-authenticate, SSO only logs successful logins. Check the log to verify user activity. The log contains one line per login.

The following details are written to the log per login request:

- Time and date stamp when the user logged in
- Product code (for example, pc for NetOps Portal)
- Username
- Whether the Remember Me option was selected
- Single Sign-On version
- The remote host IP address

Follow these steps:

1. Log in to the server where a CA data source product is installed.
2. Open a command prompt, and cd to the following directory:
   `[InstallationDirectory]/PerformanceCenter/sso/logs`

   NOTE
   The audit log is saved in the following location on Windows servers:

   `[InstallationDirectory]\Portal\SSO\logs.`
3. Enter dir to see the contents of the directory.
   The filename of the log file is SingleSignOnAuditLogyyyy-mm-dd.log.
4. Enter the name of the audit file you want to view.
   The file opens in the local text editor application.

FIPS-Compliant Encryption

By default, when DX NetOps Performance Management synchronizes the Simple Network Management Protocol (SNMP) Profiles to Federal Information Processing Standards (FIPS)-compatible data sources, it encrypts the following parameters using a FIPS-compliant algorithm:

SNMPv1/v2c:

- Community Name

SNMPv3:

- User Name
- Authentication Password
- Privacy Password

FIPS-compatible data sources include the data aggregator, the Event Manager, and DX NetOps Spectrum. For other data sources, DX NetOps Performance Management synchronizes these parameters using a non-FIPS-compliant algorithm. For more information, see SNMP Profiles.

You can also configure DX NetOps Performance Management to use FIPS-compliant encryption and hashing algorithms (where applicable) for user passwords and Single Sign-On. By default, FIPS-compliant encryption is not enabled.

IMPORTANT
DX NetOps Performance Management is not fully FIPS-compliant. This feature is for FIPS-compliant encryption only and does not meet full FIPS compliance.

Enable FIPS-Compliant Encryption

If you enable FIPS-compliant encryption and you upgrade NetOps Portal before the data aggregator as recommended, temporary FIPS compatibility synchronization errors occur. This temporary condition is resolved when you upgrade the data aggregator.

> **WARNING**
> To avoid disabling the data aggregator data source, do not enable FIPS when a product upgrade is in progress and different versions of NetOps Portal and the data aggregator might be in place.

After you enable FIPS-compliant encryption, you cannot register or use data sources which do not support FIPS. Registered data sources that do not support FIPS are disabled when you enable this feature.

> **WARNING**
> Enabling FIPS-compliant encryption is not reversible. The only way to roll back the configuration is to restore the **netqosportal** and **em** database for NetOps Portal.
> For more information, see Restore Performance Center.

> **TIP**
> Enable FIPS-compliant encryption during non-business hours. Any active user sessions are invalidated when FIPS is enabled. Users need to log back in. The logs might also temporarily show encryption errors when these user sessions are invalidated.

Follow these steps:

1. Back up the **netqosportal** NetOps Portal database.
   For more information, see Back Up Performance Center.
2. Log in to the NetOps Portal host.
3. Navigate to the NetOps Portal directory by issuing the following command:
   ```
   cd /opt/CA/PerformanceCenter
   ```
4. Launch the SSO Configuration utility by issuing the following command:
   ```
   ./SsoConfig
   ```
5. Select CA NetOps Portal.
6. Select and run **option 7: Enable FIPS**.
7. Follow the prompts in the console.

The utility configures DX NetOps Performance Management to use FIPS-compliant encryption and hashing algorithms.

Next Steps

After you enable FIPS-compliant encryption, verify that the system is working. Store the most recent NetOps Portal backup in a secure location in case you have to roll back the configuration. Remove previous NetOps Portal backups or store the backups in a secure location. Passwords in backups from before you enable FIPS-compliant encryption do not use FIPS-compliant encryption.

Integrating

DX NetOps Performance Management integrates with other CA Technologies monitoring software. Each integration enriches the available data and provides more information about your infrastructure.

Many integrations, such as CA Application Delivery Analysis and Network Flow Analysis, use the data source model. The integrated product collects data and sends it to NetOps Portal for visualization. NetOps Portal controls administrative functions. For more information, see Manage Data Sources.

Other integrations, such as DX NetOps Virtual Network Assurance and DX NetOps Mediation Manager, connect to a Data Collector, and inject the data into the Data Aggregator data source. With these integrations, data appears to be native to the DX NetOps Performance Management environment.

Table 1: Performance Monitoring Integrations

Product Integrations with DX NetOps Performance Management	Integration Business Value
DX NetOps Virtual Network Assurance	DX NetOps Virtual Network Assurance enables existing infrastructure management solutions to monitor software-defined networking (SDN) and network functions virtualization (NFV). For more information, see Modern Network Monitoring.
CA Application Delivery Analysis	CA Application Delivery Analysis provides end-to-end performance monitoring through dashboards and views that show historically normal performance for users and metrics that cross acceptable performance thresholds. For more information, see Application Delivery Analysis Views.
Network Flow Analysis	Network Flow Analysis gives you an enterprise-wide view into the composition of traffic on every link and helps you detect threatening traffic patterns in the making. For more information, see Register and Configure Network Flow Analysis.
CA Unified Communications Monitor	CA Unified Communications Monitor passively monitors the performance of your unified communications systems to maintain and report on the quality of audio and video calls. For more information, see UC Monitor Views in NetOps Portal.
CA Application Performance Management	The NetOps Portal, Data Aggregator, and Data Collector components are supported for instrumentation with CA Application Performance Management (APM). APM receives performance metrics about the component services and hosts. For more information, see Monitor Server Performance with DX Application Performance Management.
DX NetOps Mediation Manager	DX NetOps Mediation Manager monitors the performance for non-SNMP based devices, such as mobile wireless, fiber-optic switch, radio access, and 3G or 4G voice data. DX NetOps Mediation Manager supports a wide range of protocols to access data, for example, SOAP, SSH, XML, SQL, JMS, SFTP, and HTTP. DX NetOps Mediation Manager is portable across all platforms. For more information, see Generate MM Device Packs.
DX NetOps Spectrum	DX NetOps Spectrum is a services and infrastructure management system that monitors the state of managed elements including the following: • Devices • Applications • Host systems • Connections For more information, see Integrate DX Spectrum for Fault Management.

Product Integrations with DX NetOps Performance Management	Integration Business Value
CA Business Intelligence	The CA Business Intelligence integration with DX NetOps Performance Management offers access to CA Business Intelligence reports and dashboards through NetOps Portal. For more information, see Integrate CA Business Intelligence.

Application Delivery Analysis Views

CA Application Delivery Analysis provides end-to-end performance monitoring through dashboards and views that show historically normal performance for users and metrics that cross acceptable performance thresholds. CA Application Delivery Analysis gathers troubleshooting information, and helps you to determine the origin of an application, network, or server performance problem. A CA Application Delivery Analysis user with the Administrator product privilege can configure the product to automatically initiate notifications or investigative actions for performance anomalies.

The CA Application Delivery Analysis views available in NetOps Portal are different from the views available in CA Application Delivery Analysis. Each view has a column titled Data Source when multiple data sources are available. Also, each view offers a context that filters the results. For example, CA Application Delivery Analysis views can be configured with a server context, which allows you to report on a particular server.

Metrics

When registered as a data source, CA Application Delivery Analysis views report on the following metrics:

Active Sessions

Measures the number of active TCP sessions reported by a monitor feed that match an application/server/network combination on the management console.

Byte Loss Percentage

Measures the ratio of retransmitted data to total data, percentage of data lost on the monitored network, and loss rate in bytes per second.

Completed Sessions

Measures the number of sessions completed during a 5-minute monitoring period.

Connection Setup Time

Measures the amount of time that it takes to establish a TCP session between the client and server before data transfer can begin.

Data Transfer Time

A Combined metric that measures the time that it takes to transmit a complete application response from the first response (the end of the Server Response Time) to the last packet sent in that request. Data Transfer Time excludes the initial server response time and includes NRTT if there is no more data to send than fits in the TCP window. The response

time can be impacted by the design of the application or the performance of the server or network. The CA Application Delivery Analysis management console does not open an incident when the Data Transfer Time threshold is crossed.

Effective Network Round Trip Time

Is a Network metric that consists of Network Round Trip Time plus Retransmission Delay. Note that Retransmission Delay is not the delay due to any retransmissions; it is the average amount of retransmission delay per round trip. It is important to note that the management console adds two averages, and combines two metrics. The CA Application Delivery Analysis management console opens a Network incident when the incident threshold for this metric is crossed.

Estimated Client Transaction TIme

Provides an approximation of Total Transaction Time in the absence of a Client segment. This metric is the summation of the Server segment Transaction Time and the WAN segment Network Round Trip Time. This is not an “engineering” level metric, but it can provide an indicator of the order of magnitude of response time for a given location or time of day. This metric is only available with a CA ADA 9.3 data source.

Expired Sessions

Measures the number of TCP sessions where the CA ADA Monitor service did not see the TCP session tear down (FIN or RST packet). Sessions which are inactive for a period of time are cleared out of memory and marked as Expired. The management console classifies a session as Expired if it does not observe any packets in a 15-minute period. Too many expired sessions left open can cause servers to become unresponsive.

From Server Bytes

Measures the number of bytes that a server sent to a client.

From Server Packets

Measures the number of packets that a server sent to a client.

Network Connection Time (NCT or NSCT)

Is a Network metric that measures the amount of time between the Syn-Ack sent by the server and the Ack received back from the client. When a network is uncongested, it is a measurement of network latency that represents the minimum latency due to distance and serialization, and is the best possible round trip time for your network architecture. Sudden spikes in this value are commonly attributed to congestion, while a plateau (which goes up and stays up) typically indicates a path change. The CA Application Delivery Analysis management console opens a Network incident when the incident threshold for this metric is crossed.

Network Round Trip Time

A Network metric that measures the time that a packet takes to travel across the network in both directions between the server and clients on a network, excluding loss. Application, server, and client processing time are excluded. The CA Application Delivery Analysis management console opens a Network incident when the incident threshold for this metric is crossed.

Observations

The observation count measures the number of times during a 5-minute monitoring interval that a monitoring device calculated a performance metric for a particular application/server/network combination. Within a TCP transaction there can be different numbers of observations of different metrics. For example, there may be more observation counts for Network Round Trip Time than Server Response Time. Other metrics are links, and always have the same number of observations. For example, each TCP transaction has one Server Response Time observation and one Data Transfer

Time observation. To rate a metric as Normal, Minor (yellow), or Major (orange), the metric must have a minimum number of observations.

Open Sessions

Measures the number of sessions still open at the end of the data collection period. Open sessions might become Expired or Completed during subsequent reporting intervals.

Packet Loss Percentage

A Network metric that measures the ratio of retransmitted data to total data within the network. This is measured from the vantage point of the monitoring device, which is next to the server. The monitoring device can identify packets retransmitted by the server because of data losses in the server-to-client direction along the network path. When data loss occurs in the client-to-server direction (in the network path before reaching the server, for example), the monitoring device cannot observe such packet loss, and that delay is not included in the Packet Loss Percentage. On the Engineering page of the CA Application Delivery Analysis management console, Packet Loss Percentage is part of the QoS report. The CA Application Delivery Analysis management console opens a Network incident when the incident threshold for this metric is crossed.

Rate from Server in Bytes

Measures the rate of data from the server in bytes.

Rate from Server in Packets

Measures the rate of data from the server in packets.

Rate to Server in Bytes

Measures the rate of data to the server in bytes.

Rate to Server in Packets

Measures the rate of data to the server in packets.

Refused Session Percentage

A Server metric that measures the percentage of connection requests that the server explicitly rejected during the reporting interval. This metric is part of the Unfulfilled TCP/IP Session Requests report in the CA Application Delivery Analysis management console. The CA Application Delivery Analysis management console opens a Server incident when the incident threshold for this metric is crossed.

Refused Sessions

Measures the number of connection requests that were explicitly rejected by the server during the three-way handshake.

Retransmission Delay

A Network metric that measures the elapsed time between sending the original packet and sending the last duplicate packet. The management console reports Retransmission Delay as an average across observations, and not just for the retransmitted packets. For example, if one packet in a set of 10 requires 300 ms of retransmission time, the Retransmission Delay is reported as 30 ms (300 ms/10 packets). The CA Application Delivery Analysis management console opens a Network incident when the incident threshold for this metric is crossed.

Retransmitted Bytes

Measures the amount of bandwidth used by retransmitted data.

Retransmitted Packets

Measures the increased load on the network infrastructure due to retransmitted packets.

Server Connection Time (SCT)

A Server metric that measures the amount of time that a server takes to acknowledge the initial client connection request by sending a Syn-Ack in response to the client's SYN packet. The CA Application Delivery Analysis management console opens a Server incident when the incident threshold for this metric is crossed.

Server Response Time

A Server metric that measures the time that it takes for a server to send an initial response to a client request or the initial server "think time." Increases in the Server Response Time generally indicate a lack of server resources such as CPU, memory, disk, or I/O, a poorly written application, or a poorly-performing tier in a multi-tier application. The CA Application Delivery Analysis management console opens a Server incident when the incident threshold for this metric is crossed.

Session Duration

Measures the duration of each TCP session.

To Server Bytes

Measures the number of bytes that a client sent to a server.

To Server Packets

Measures the number of packets that a client sent to a server.

Total Bytes

Measures the total bytes transmitted in and out.

Total Packets

Measures the total packets transmitted in and out.

Total Sessions

Indicates the total number of sessions that Completed or Expired in the sampling period. The sum of completed sessions and expired sessions is equal to the number of total sessions. It does not include Open, Unresponsive, or Refused sessions.

Transaction Time

A Combined metric that measures the amount of time elapsed from when the client sends the request to when it receives the last packet in the response. Transaction Time is the sum of Server Response Time, Network Round Trip Time, Retransmission Delay, and Data Transfer Time. The CA Application Delivery Analysis management console does not open an incident when the Transaction Time threshold is crossed.

Unresponsive Session Percentage

A Server metric that measures the percentage of sessions where a connection request was sent, but the server never responded. Part of the Unfulfilled TCP/IP Session Requests view. The CA Application Delivery Analysis management console opens a Server incident when the incident threshold for this metric is crossed.

Unresponsive Sessions

Indicates the number of sessions where a connection request was sent, but the server never responded.

User Count

Indicates the number of unique clients observed during the sample period.

User Throughput

Indicates the number of bytes transmitted divided by the time required to transmit the information.

ADA Dashboards

DX NetOps Performance Management offers CA Application Delivery Analysis (ADA) dashboards. ADA views are also included in NetOps Portal dashboards. The following CA ADA dashboards are available:

- Application Performance Dashboard
- Network Overview Dashboard (CA ADA)
- Network Performance Dashboard (CA ADA)
- Performance Events Dashboard
- Server Overview Dashboard (CA ADA)
- Server Performance Dashboard (CA ADA)

Application Performance Dashboard

The Application Performance Dashboard provides the following application performance and incident views:

Incident Count by Application (CA ADA)

The Incident Count by Application view lists the applications that are impacted with the most incidents. Use this view to help you start troubleshooting reported issues by matching Network and Server incidents to trouble tickets.

By default, the incident count includes Open and Closed incidents where the performance threshold for a Server metric or Network metric was exceeded during more than one 5-minute reporting interval.

> **NOTE**
>
> This view does **not** include Combined Metrics (Data Transfer Time and Transaction Time) because CA Application Delivery Analysis (CA ADA) opens a Network or Server incident when the threshold for a Combined metric is exceeded.

Click an application to view its incidents in the CA ADA management console.

You may also edit the view to change the filter criteria for the view. Filter criteria include:

- **Scheduled Maintenance**
  Choose whether to include incidents that are opened or closed during a scheduled maintenance period:
  - **Excluded**

Choose this option to exclude incidents that are opened or closed during a server's scheduled maintenance period. This is the default.

- **Included**
  Choose this option to include incidents that are opened or closed during a server's maintenance period. This option increases the number of open and closed incidents because when a scheduled maintenance period begins, CA ADA:
  - Closes the existing incidents
  - Opens new incidents, if performance degrades

- **Metric Type**
  Filter the incident count by Server metric, Network metric, or Combined metric.
  - **Minimum Elapsed Time (Minutes)**
    Choose a time period to specify a threshold for the total duration required to include an incident in the Incident Count. To include all incidents with the specified Minimum Severity, choose a Duration of Any. The default is 10 minutes.
  - **Minimum Severity**
    From least severe to most severe, include:
    - **Minor:** Includes all incident severities in the Incident count. This is the default.
    - **Major:** Includes incidents that have a Major or Unavailable severity.
    - **Unavailable:** Includes Unavailable incidents.
  - **Incident State**
    For the metrics that you have selected, specify whether to filter open or closed incidents in the Incident Count. By default, open and closed incidents are included based on the specified Minimum Severity and Duration.

    NOTE

    Closed incidents help you to identify incident trends that can lead to problem resolution. By default, CA ADA retains incidents for 3 months.

    Choose one of the following options:
    - Open
    - Closed
    - Open and Closed. This is the default.
  - **Context Settings**
    Displays the view filter. A context type of:
    - **Summary**
      Filters the view based on the selected group.
    - **Server**
      Filters the view based on the selected server.
  - **Apply Changes**
    Select the scope of your changes from the **Apply Changes** drop-down.

Performance by Application (CA ADA)

The Performance by Server view provides information about application performance. This view displays performance data from the last 24 hours. If the time period for the report is longer than 24 hours, the view only displays performance data for the most recent 24 hours.

If you have enough observations, and if a threshold is available for each combination of application, server, network, and 5-minute period, CA Application Delivery Analysis (CA ADA) classifies the metric using one of these ratings:

- **Acknowledged**

Displays a severity state (diagonal stripes) to indicate a CA ADA or CA NetOps Portal (CAPC) user acknowledged an incident. When this happens, CAPC marks data that the incident covers as Acknowledged. CA ADA automatically marks future data covered by an acknowledged incident as Acknowledged.

- **No Data**
  Displays a severity state (blank) to indicate that no data is available.
- **Unrated**
  Displays a severity state (gray) to indicate that either there is insufficient past data (two full business days of data are needed) to establish a threshold, or there were not enough observations to exceed the minimum observations threshold.
- **Normal**
  Displays a severity state (green) to indicate that the metric value is between zero and the Minor threshold.
- **Minor**
  Displays a severity state (yellow) to indicate that the metric value exceeds the Minor threshold as defined in the CA ADA management console.
- **Major**
  Displays a severity state (orange) to indicate that the metric value exceeds the Major threshold as defined in the CA ADA management console.
- **Unavailable**
  Displays a severity state (red) to indicate that the application on a server is not running (unavailable). This rating appears when you click All Server Metrics in Settings. This rating is only applicable to a user-defined application where the CA ADA administrator has assigned servers to the application.

To isolate the component and metric contributing to a performance problem, click one of the worst-performing items at the top of the graph.

You may also the filter criteria for the view. By default, all data is included. Filter criteria include:

- **Metric Type**
  Select the Server metrics, Network metrics or Combined metrics you want to use to filter the list of networks.
- **Context Settings**
  Displays the view filter. A context type of:
  - **Summary**
    Filters the view based on the selected group.
  - **Server**
    Filters the view based on the selected server.
- **Apply Changes**
  Select the scope of your changes from the **Apply Changes** drop-down.

Performance Map by Application (CA ADA)

The Performance Map by Application view provides a Top N view into the worst performing applications based on a particular metric. Configure this view to measure application performance using any of the available CA Application Delivery Analysis (CA ADA) metrics. If you receive an incident notification, or notice degraded performance in the Performance By Network view, use this view to drill into the Components reports on the Engineering report of the CA ADA management console. Use the Components reports to begin troubleshooting the metric details for the selected application.

To isolate the component and metric contributing to a performance problem, click one of the worst-performing networks at the top of the graph.

You may also edit the filter criteria for the view to display performance information for a different metric. Filter criteria include:

- **Metric Type**

Choose a metric to filter the list of networks.

- **Context Settings**
  Display the view filter. A context type of:
  - **Summary**
    Filters the view based on the selected group.
  - **Server**
    Filters the view based on the selected server.
- **Apply Changes**
  Select the scope of your changes from the **Apply Changes** drop-down.

Network Overview Dashboard (CA ADA)

Use the Network Overview Dashboard to determine the network device status. This dashboard gives you a broad view of network performance, and helps you to track discards and errors associated with networking infrastructure.

By default, the Network Overview Dashboard contains the following views. Note that some views require a CA Application Delivery Analysis data source:

- Incident Count By Network
- Performance Scorecard
- Top CPU Utilization Routers/Switches Gauge/Table
- Top Enterprise Hosts by Volume
- Top Enterprise Protocols by Volume
- Top Interface Errors - Discards
- Top Interface Utilization - In - Trend/Table
- Top Interface Utilization - Out - Trend/Table
- Top Memory Utilization Routers/Switches - Gauge/Table

Network Performance Dashboard (CA ADA)

The Network Performance dashboard provides network performance and incident views, and views of interface and router health and status. This dashboard emphasizes interface utilization, and includes lists of devices for easy drilldown into device performance and availability data.

By default, the Network Performance dashboard contains the following views. Note that some views require a CA Application Delivery Analysis data source:

- Groups and Sites
- Incident Count By Network
- Incident List by Network
- Performance Maps
- Top Interface Errors - Discards
- Top Interface Utilization - In - Trend/Table
- Top Interface Utilization - Out - Trend/Table
- Top Performance by Network
- Top Performance Map by Network

Performance Events Dashboard

The Performance Events dashboard provides performance and incident views of network, server, and applications.

By default, the Performance Events dashboard contains the following views:

- Incident Count by Application
- Incident Count By Network
- Incident Count by Server
- Incident List by Network
- Incident List by Server
- Performance by Application
- Performance by Network
- Performance by Server

Server Overview Dashboard (CA ADA)

The Server Overview Dashboard shows an overview of server and application performance. The views on this dashboard help you to track memory and CPU utilization levels on critical servers.

The following topics are covered in this section:

- Performance Scorecard
- Top CPU Utilization
- Top Disk Storage
- Top Disk Utilization
- Top Least Available Servers
- Top Least Reachable Servers
- Top Memory Utilization

Server Performance Dashboard (CA ADA)

The Server Performance dashboard provides server performance and incident views, and health- and status-related data from servers and groups of servers.

- Groups and Sites
- Incident Count by Server
- Performance by Server
- Performance Map by Server
- Servers (Universal List)

ADA Views

The following CA Application Delivery Analysis (ADA) views are available in NetOps Portal:

- Engineering Trend
- Incident Counts
- Incident Lists
- Performance Maps
- Performance Scorecard (ADA View)
- Performance Views

Engineering Trend

The Engineering Trend view provides an engineering chart for a particular Server metric, Network metric or Combined metric.

If baseline data is applicable, the grey line indicates the number of observations. Not every metric supports baseline or observation data.

> **NOTE**
> This view does not support data from multiple data sources. If you registered multiple instances of CA ADA, edit the view to filter by one data source. You may also change the dashboard group context by clicking the Group link. Using either method, you can select one data source in the Groups tree to serve as the view context.

Configure this view to measure application performance using any of the available CA ADA metrics. If you receive an incident notification, or notice degraded performance, use this view to quickly drill to the Components reports on the Engineering report of the CA ADA management console. Use the Components reports to begin troubleshooting the metric details for the selected application.

Edit the filter criteria for the view to display performance information for a different metric:

- **Metric Type**
  Choose a CA ADA metric to filter the list of applications.
- **Context Settings**
  Displays the view filter. A context type of:
  - **Summary**
    Filters the view based on the selected group.
  - **Server**
    Filters the view based on the selected server.
- **Apply Changes**
  Select the scope of your changes from the **Apply Changes** drop-down.

Incident Count by Application

The Incident Count by Application view lists the applications that are impacted with the most incidents. Use this view to help you start troubleshooting reported issues by matching Network and Server incidents to trouble tickets.

By default, the incident count includes Open and Closed incidents where the performance threshold for a Server metric or Network metric was exceeded during more than one 5-minute reporting interval.

> **NOTE**
>
> This view does **not** include Combined Metrics (Data Transfer Time and Transaction Time) because CA Application Delivery Analysis (CA ADA) opens a Network or Server incident when the threshold for a Combined metric is exceeded.

Click an application to view its incidents in the CA ADA management console.

You may also edit the view to change the filter criteria for the view. Filter criteria include:

- **Scheduled Maintenance**
  Choose whether to include incidents that are opened or closed during a scheduled maintenance period:
  - **Excluded**
    Choose this option to exclude incidents that are opened or closed during a server's scheduled maintenance period. This is the default.
  - **Included**

Choose this option to include incidents that are opened or closed during a server's maintenance period. This option increases the number of open and closed incidents because when a scheduled maintenance period begins, CA ADA:
  - Closes the existing incidents
  - Opens new incidents, if performance degrades
- **Metric Type**
  Filter the incident count by Server metric, Network metric, or Combined metric.
  - **Minimum Elapsed Time (Minutes)**
    Choose a time period to specify a threshold for the total duration required to include an incident in the Incident Count. To include all incidents with the specified Minimum Severity, choose a Duration of Any. The default is 10 minutes.
  - **Minimum Severity**
    From least severe to most severe, include:
    - **Minor:** Includes all incident severities in the Incident count. This is the default.
    - **Major:** Includes incidents that have a Major or Unavailable severity.
    - **Unavailable:** Includes Unavailable incidents.
  - **Incident State**
    For the metrics that you have selected, specify whether to filter open or closed incidents in the Incident Count. By default, open and closed incidents are included based on the specified Minimum Severity and Duration.

    NOTE

    Closed incidents help you to identify incident trends that can lead to problem resolution. By default, CA ADA retains incidents for 3 months.

    Choose one of the following options:
    - Open
    - Closed
    - Open and Closed. This is the default.
  - **Context Settings**
    Displays the view filter. A context type of:
    - **Summary**
      Filters the view based on the selected group.
    - **Server**
      Filters the view based on the selected server.
  - **Apply Changes**Select the scope of your changes from the **Apply Changes** drop-down.

Incident Count By Network

The Incident Count by Network view lists the networks that are impacted with the most incidents. Use this view to help you start troubleshooting reported issues by matching network incidents to trouble tickets. By default, the incident count includes open and closed incidents where the performance threshold for a Network metric was exceeded during more than one 5-minute reporting interval.

Click a network to view its incidents in the CA Application Delivery Analysis (CA ADA) management console.

You may also edit the view to change the filter criteria for the view. Filter criteria include:

- **Scheduled Maintenance**
  Choose whether to include incidents that are opened or closed during a scheduled maintenance period:
  - **Excluded**

Choose this option to exclude incidents that are opened or closed during a server's scheduled maintenance period. This is the default.

- **Included**
  Choose this option to include incidents that are opened or closed during a server's maintenance period. This option increases the number of open and closed incidents because when a scheduled maintenance period begins, CA ADA:
  - Closes the existing incidents
  - Opens new incidents, if performance degrades

- **Metric Type**
  Filter the incident count by the Network metric.
- **Minimum Elapsed Time (Minutes)**
  Choose a time period to specify a threshold for the total duration required to include an incident in the Incident Count. To include all incidents with the specified Minimum Severity, choose a Duration of Any. The default is 10 minutes.
- **Minimum Severity**
  From least severe to most severe, include:
  - **Minor:** Includes all incident severities in the Incident count. This is the default.
  - **Major:** Includes incidents that have a Major or Unavailable severity.
  - **Unavailable:** Includes Unavailable incidents.
- **Incident State**
  For the metrics that you have selected, specify whether to filter open or closed incidents in the Incident Count. By default, open and closed incidents are included based on the specified Minimum Severity and Duration.

  NOTE

  Closed incidents help you to identify incident trends that can lead to problem resolution. By default, CA ADA retains incidents for 3 months.

  Choose one of the following options:
  - Open
  - Closed
  - Open and Closed. This is the default.
- **Context Settings**
  Displays the view filter. A context type of:
  - **Summary**
    Filters the view based on the selected group.
  - **Server**
    Filters the view based on the selected server.
- **Apply Changes**
  Select the scope of your changes from the **Apply Changes** drop-down.

Incident Count by Server

The Incident Count by Server view lists the servers that are impacted with the most incidents. Use this view to help you start troubleshooting reported issues by matching server incidents to trouble tickets. By default, the incident count includes open and closed incidents where the performance threshold for a Server metric was exceeded during more than one 5-minute reporting interval.

Click a server to view its incidents in the CA Application Delivery Analysis (CA ADA) management console.

You may also edit the view to change the filter criteria for the view. Filter criteria include:

- **Scheduled Maintenance**
  Choose whether to include incidents that are opened or closed during a scheduled maintenance period:

  - **Excluded**
    Choose this option to exclude incidents that are opened or closed during a server's scheduled maintenance period. This is the default.
  - **Included**
    Choose this option to include incidents that are opened or closed during a server's maintenance period. This option increases the number of open and closed incidents because when a scheduled maintenance period begins, CA Application Delivery Analysis:
    - • Closes the existing incidents
      - Opens new incidents, if performance degrades
- **Metric Type**
  Filter the incident count by the Server metric.
- **Minimum Elapsed Time (Minutes)**
  Choose a time period to specify a threshold for the total duration required to include an incident in the Incident Count. To include all incidents with the specified Minimum Severity, choose a Duration of Any. The default is 10 minutes.
- **Minimum Severity**
  From least severe to most severe, include:
  - **Minor:** Includes all incident severities in the Incident count. This is the default.
  - **Major:** Includes incidents that have a Major or Unavailable severity.
  - **Unavailable:** Includes Unavailable incidents.
- **Incident State**
  For the metrics that you have selected, specify whether to filter open or closed incidents in the Incident Count. By default, open and closed incidents are included based on the specified Minimum Severity and Duration.

    NOTE

    Closed incidents help you to identify incident trends that can lead to problem resolution. By default, CA ADA retains incidents for 3 months.

  Choose one of the following options:
  - Open
  - Closed
  - Open and Closed. This is the default.
- **Context Settings**
  Displays the view filter. A context type of:
  - **Summary**
    Filters the view based on the selected group.
  - **Server**
    Filters the view based on the selected server.
- **Apply Changes**
  Select the scope of your changes from the **Apply Changes** drop-down.

Incident Counts

The Incident Count views show the networks, servers, and applications with the most incidents. The following Incident Count views are available:

Incident Count by Network

The Incident Count by Network view lists the networks that are impacted with the most incidents. Use this view to help you start troubleshooting reported issues by matching network incidents to trouble tickets. By default, the incident count

includes open and closed incidents where the performance threshold for a Network metric was exceeded during more than one 5-minute reporting interval.

Click a network to view its incidents in the CA Application Delivery Analysis (CA ADA) management console.

You may also edit the view to change the filter criteria for the view. Filter criteria include:

- **Scheduled Maintenance**
  Choose whether to include incidents that are opened or closed during a scheduled maintenance period:
  - **Excluded**
    Choose this option to exclude incidents that are opened or closed during a server's scheduled maintenance period. This is the default.
  - **Included**
    Choose this option to include incidents that are opened or closed during a server's maintenance period. This option increases the number of open and closed incidents because when a scheduled maintenance period begins, CA ADA:
    - Closes the existing incidents
    - Opens new incidents, if performance degrades
- **Metric Type**
  Filter the incident count by the Network metric.
- **Minimum Elapsed Time (Minutes)**
  Choose a time period to specify a threshold for the total duration required to include an incident in the Incident Count. To include all incidents with the specified Minimum Severity, choose a Duration of Any. The default is 10 minutes.
- **Minimum Severity**
  From least severe to most severe, include:
  - **Minor:** Includes all incident severities in the Incident count. This is the default.
  - **Major:** Includes incidents that have a Major or Unavailable severity.
  - **Unavailable:** Includes Unavailable incidents.
- **Incident State**
  For the metrics that you have selected, specify whether to filter open or closed incidents in the Incident Count. By default, open and closed incidents are included based on the specified Minimum Severity and Duration.

    NOTE

    Closed incidents help you to identify incident trends that can lead to problem resolution. By default, CA ADA retains incidents for 3 months.

  Choose one of the following options:
  - Open
  - Closed
  - Open and Closed. This is the default.
- **Context Settings**
  Displays the view filter. A context type of:
  - **Summary**
    Filters the view based on the selected group.
  - **Server**
    Filters the view based on the selected server.
- **Apply Changes**
  Select the scope of your changes from the **Apply Changes** drop-down.

Incident Count by Server

The Incident Count by Server view lists the servers that are impacted with the most incidents. Use this view to help you start troubleshooting reported issues by matching server incidents to trouble tickets. By default, the incident count includes

open and closed incidents where the performance threshold for a Server metric was exceeded during more than one 5-minute reporting interval.

Click a server to view its incidents in the CA Application Delivery Analysis (CA ADA) management console.

You may also edit the view to change the filter criteria for the view. Filter criteria include:

- **Scheduled Maintenance**
  Choose whether to include incidents that are opened or closed during a scheduled maintenance period:
  - **Excluded**
    Choose this option to exclude incidents that are opened or closed during a server's scheduled maintenance period. This is the default.
  - **Included**
    Choose this option to include incidents that are opened or closed during a server's maintenance period. This option increases the number of open and closed incidents because when a scheduled maintenance period begins, CA Application Delivery Analysis:
    - • Closes the existing incidents
      - Opens new incidents, if performance degrades
- **Metric Type**
  Filter the incident count by the Server metric.
- **Minimum Elapsed Time (Minutes)**
  Choose a time period to specify a threshold for the total duration required to include an incident in the Incident Count. To include all incidents with the specified Minimum Severity, choose a Duration of Any. The default is 10 minutes.
- **Minimum Severity**
  From least severe to most severe, include:
  - **Minor:** Includes all incident severities in the Incident count. This is the default.
  - **Major:** Includes incidents that have a Major or Unavailable severity.
  - **Unavailable:** Includes Unavailable incidents.
- **Incident State**
  For the metrics that you have selected, specify whether to filter open or closed incidents in the Incident Count. By default, open and closed incidents are included based on the specified Minimum Severity and Duration.

    NOTE

    Closed incidents help you to identify incident trends that can lead to problem resolution. By default, CA ADA retains incidents for 3 months.

  Choose one of the following options:
  - Open
  - Closed
  - Open and Closed. This is the default.
- **Context Settings**
  Displays the view filter. A context type of:
  - **Summary**
    Filters the view based on the selected group.
  - **Server**
    Filters the view based on the selected server.
- **Apply Changes**
  Select the scope of your changes from the **Apply Changes** drop-down.

Incident Count by Application

The Incident Count by Application view lists the applications that are impacted with the most incidents. Use this view to help you start troubleshooting reported issues by matching Network and Server incidents to trouble tickets.

By default, the incident count includes Open and Closed incidents where the performance threshold for a Server metric or Network metric was exceeded during more than one 5-minute reporting interval.

NOTE

This view does **not** include Combined Metrics (Data Transfer Time and Transaction Time) because CA Application Delivery Analysis (CA ADA) opens a Network or Server incident when the threshold for a Combined metric is exceeded.

Click an application to view its incidents in the CA ADA management console.

You may also edit the view to change the filter criteria for the view. Filter criteria include:

- **Scheduled Maintenance**
  Choose whether to include incidents that are opened or closed during a scheduled maintenance period:
  - **Excluded**
    Choose this option to exclude incidents that are opened or closed during a server's scheduled maintenance period. This is the default.
  - **Included**
    Choose this option to include incidents that are opened or closed during a server's maintenance period. This option increases the number of open and closed incidents because when a scheduled maintenance period begins, CA ADA:
    - Closes the existing incidents
    - Opens new incidents, if performance degrades
- **Metric Type**
  Filter the incident count by Server metric, Network metric, or Combined metric.
  - **Minimum Elapsed Time (Minutes)**
    Choose a time period to specify a threshold for the total duration required to include an incident in the Incident Count. To include all incidents with the specified Minimum Severity, choose a Duration of Any. The default is 10 minutes.
  - **Minimum Severity**
    From least severe to most severe, include:
    - **Minor:** Includes all incident severities in the Incident count. This is the default.
    - **Major:** Includes incidents that have a Major or Unavailable severity.
    - **Unavailable:** Includes Unavailable incidents.
  - **Incident State**
    For the metrics that you have selected, specify whether to filter open or closed incidents in the Incident Count. By default, open and closed incidents are included based on the specified Minimum Severity and Duration.

    NOTE

    Closed incidents help you to identify incident trends that can lead to problem resolution. By default, CA ADA retains incidents for 3 months.

    Choose one of the following options:
    - Open
    - Closed
    - Open and Closed. This is the default.
  - **Context Settings**
    Displays the view filter. A context type of:
    - **Summary**
      Filters the view based on the selected group.
    - **Server**
      Filters the view based on the selected server.
  - **Apply Changes**

Select the scope of your changes from the **Apply Changes** drop-down.

Incident List by Network

The Incident List by Network view displays a summary of Network incidents. Use this view to drill into details on an incident in the CA Application Delivery Analysis (CA ADA) management console. By default, the incident list includes Open and Closed incidents where the performance threshold for a Network metric was exceeded during more than one 5-minute reporting interval.

Click an incident to view its details in the CA ADA management console.

You may also edit the view to change the filter criteria for the list of incidents. Filter criteria include:

- **Scheduled Maintenance**
  Choose whether to include incidents that are opened or closed during a scheduled maintenance period:
  - **Excluded**
    Choose this option to exclude incidents that are opened or closed during a server's scheduled maintenance period. This is the default.
  - **Included**
    Choose this option to include incidents that are opened or closed during a server's maintenance period. This option increases the number of open and closed incidents because when a scheduled maintenance period begins, CA ADA:
    - Closes the existing incidents
    - Opens new incidents, if performance degrades
- **Metric Type**
  Filter the incident count by the Network metric.
- **Minimum Elapsed Time (Minutes)**
  Choose a time period to specify a threshold for the total duration required to include an incident in the Incident Count. To include all incidents with the specified Minimum Severity, choose a Duration of Any. The default is 10 minutes.
- **Minimum Severity**
  From least severe to most severe, include:
  - **Minor:** Includes all incident severities in the Incident count. This is the default.
  - **Major:** Includes incidents that have a Major or Unavailable severity.
  - **Unavailable:** Includes Unavailable incidents.
- **Incident State**
  For the metrics that you have selected, specify whether to filter open or closed incidents in the Incident Count. By default, open and closed incidents are included based on the specified Minimum Severity and Duration.

  NOTE

  Closed incidents help you to identify incident trends that can lead to problem resolution. By default, CA ADA retains incidents for 3 months.

  Choose one of the following options:
  - Open
  - Closed
  - Open and Closed. This is the default.
- **Context Settings**
  Displays the view filter. A context type of:
  - **Summary**
    Filters the view based on the selected group.
  - **Server**

Filters the view based on the selected server.

- **Apply Changes**
  Select the scope of your changes from the **Apply Changes** drop-down.

Incident List by Server

The Incident List by Server view shows a summary of Server incidents, and lets you drill into details on an incident in the CA Application Delivery Analysis (CA ADA) management console. By default, the incident list includes Open and Closed incidents where the performance threshold for a Server metric was exceeded during more than one 5-minute reporting interval.

Click an incident to view its details in the CA ADA management console.

You may also edit the view to change the filter criteria for the list of incidents. Filter criteria include:

- **Context Settings**
  Displays the view filter. A context type of:
  - **Summary**
    Filters the view based on the selected group.
  - **Server**
    Filters the view based on the selected server.
- **Metric Type**
  Filter the list of incidents by Server metric.
- **Minimum Elapsed Time (Minutes)**
  Choose a time period to specify a threshold for the total duration required to include an incident in the Incident Count. To include all incidents with the specified Minimum Severity, choose a Duration of Any. The default is 10 minutes.
- **Minimum Severity**
  From least severe to most severe, include:
  - **Minor:** Includes all incident severities in the Incident count. This is the default.
  - **Major:** Includes incidents that have a Major or Unavailable severity.
  - **Unavailable:** Includes Unavailable incidents.
- **Incident State**
  For the metrics that you have selected, specify whether to filter open or closed incidents in the Incident Count. By default, open and closed incidents are included based on the specified Minimum Severity and Duration.

  NOTE

  Closed incidents help you to identify incident trends that can lead to problem resolution. By default, CA ADA retains incidents for 3 months.

  Choose one of the following options:
  - Open
  - Closed
  - Open and Closed. This is the default.
- **Apply Changes**
  Select the scope of your changes from the **Apply Changes** drop-down.

Incident Lists

The Incident List views show summaries of network and server incidents. The following Incident List views are available:

Incident List by Network

The Incident List by Network view displays a summary of Network incidents. Use this view to drill into details on an incident in the CA Application Delivery Analysis (CA ADA) management console. By default, the incident list includes Open and Closed incidents where the performance threshold for a Network metric was exceeded during more than one 5-minute reporting interval.

Click an incident to view its details in the CA ADA management console.

You may also edit the view to change the filter criteria for the list of incidents. Filter criteria include:

- **Scheduled Maintenance**
  Choose whether to include incidents that are opened or closed during a scheduled maintenance period:
  - **Excluded**
    Choose this option to exclude incidents that are opened or closed during a server's scheduled maintenance period. This is the default.
  - **Included**
    Choose this option to include incidents that are opened or closed during a server's maintenance period. This option increases the number of open and closed incidents because when a scheduled maintenance period begins, CA ADA:
    - Closes the existing incidents
    - Opens new incidents, if performance degrades
- **Metric Type**
  Filter the incident count by the Network metric.
- **Minimum Elapsed Time (Minutes)**
  Choose a time period to specify a threshold for the total duration required to include an incident in the Incident Count. To include all incidents with the specified Minimum Severity, choose a Duration of Any. The default is 10 minutes.
- **Minimum Severity**
  From least severe to most severe, include:
  - **Minor:** Includes all incident severities in the Incident count. This is the default.
  - **Major:** Includes incidents that have a Major or Unavailable severity.
  - **Unavailable:** Includes Unavailable incidents.
- **Incident State**
  For the metrics that you have selected, specify whether to filter open or closed incidents in the Incident Count. By default, open and closed incidents are included based on the specified Minimum Severity and Duration.

  NOTE

  Closed incidents help you to identify incident trends that can lead to problem resolution. By default, CA ADA retains incidents for 3 months.

  Choose one of the following options:
  - Open
  - Closed
  - Open and Closed. This is the default.
- **Context Settings**
  Displays the view filter. A context type of:
  - **Summary**
    Filters the view based on the selected group.
  - **Server**
    Filters the view based on the selected server.
- **Apply Changes**
  Select the scope of your changes from the **Apply Changes** drop-down.

Incident List by Server

The Incident List by Network view displays a summary of Server incidents. Use this view to drill into details on an incident in the CA Application Delivery Analysis (CA ADA) management console. By default, the incident list includes Open and Closed incidents where the performance threshold for a Server metric was exceeded during more than one 5-minute reporting interval.

Click an incident to view its details in the CA ADA management console.

You may also edit the view to change the filter criteria for the list of incidents. Filter criteria include:

- **Context Settings**
  Displays the view filter. A context type of:
  - **Summary**
    Filters the view based on the selected group.
  - **Server**
    Filters the view based on the selected server.
- **Metric Type**
  Filter the list of incidents by Server metric.
- **Minimum Elapsed Time (Minutes)**
  Choose a time period to specify a threshold for the total duration required to include an incident in the Incident Count. To include all incidents with the specified Minimum Severity, choose a Duration of Any. The default is 10 minutes.
- **Minimum Severity**
  From least severe to most severe, include:
  - **Minor:** Includes all incident severities in the Incident count. This is the default.
  - **Major:** Includes incidents that have a Major or Unavailable severity.
  - **Unavailable:** Includes Unavailable incidents.
- **Incident State**
  For the metrics that you have selected, specify whether to filter open or closed incidents in the Incident Count. By default, open and closed incidents are included based on the specified Minimum Severity and Duration.

    NOTE

    Closed incidents help you to identify incident trends that can lead to problem resolution. By default, CA ADA retains incidents for 3 months.

  Choose one of the following options:
  - Open
  - Closed
  - Open and Closed. This is the default.
- **Apply Changes**
  Select the scope of your changes from the **Apply Changes** drop-down.

Performance by Application

The Performance by Application view shows application performance data from the last 24 hours. If the time period for the report is longer than 24 hours, the view only displays performance data for the most recent 24 hours.

If you have enough observations, and if a threshold is available for each combination of application, server, network, and 5-minute period, CA Application Delivery Analysis (CA ADA) classifies the metric using one of these ratings:

- **Acknowledged**

Displays a severity state (diagonal stripes) to indicate a CA ADA or CA NetOps Portal (CAPC) user acknowledged an incident. When this happens, CAPC marks data that the incident covers as Acknowledged. CA ADA automatically marks future data covered by an acknowledged incident as Acknowledged.

- **No Data**
  Displays a severity state (blank) to indicate that no data is available.
- **Unrated**
  Displays a severity state (gray) to indicate that either there is insufficient past data (two full business days of data are needed) to establish a threshold, or there were not enough observations to exceed the minimum observations threshold.
- **Normal**
  Displays a severity state (green) to indicate that the metric value is between zero and the Minor threshold.
- **Minor**
  Displays a severity state (yellow) to indicate that the metric value exceeds the Minor threshold as defined in the CA ADA management console.
- **Major**
  Displays a severity state (orange) to indicate that the metric value exceeds the Major threshold as defined in the CA ADA management console.
- **Unavailable**
  Displays a severity state (red) to indicate that the application on a server is not running (unavailable). This rating appears when you click All Server Metrics in Settings. This rating is only applicable to a user-defined application where the CA ADA administrator has assigned servers to the application.

To isolate the component and metric contributing to a performance problem, click one of the worst-performing items at the top of the graph.

You may also the filter criteria for the view. By default, all data is included. Filter criteria include:

- **Metric Type**
  Select the Server metrics, Network metrics or Combined metrics you want to use to filter the list of networks.
- **Context Settings**
  Displays the view filter. A context type of:
  - **Summary**
    Filters the view based on the selected group.
  - **Server**
    Filters the view based on the selected server.
- **Apply Changes**Select the scope of your changes from the **Apply Changes** drop-down.

Performance by Network

The Performance by Network view shows network performance data from the last 24 hours. If the time period for the report is longer than 24 hours, the view only displays performance data for the most recent 24 hours.

If you have enough observations, and if a threshold is available for each combination of application, server, network, and 5-minute period, CA Application Delivery Analysis (CA ADA) classifies the metric using one of these ratings:

- **Acknowledged**
  Displays a severity state (diagonal stripes) to indicate a CA ADA or CA NetOps Portal user acknowledged an incident. When this happens, the CA NetOps Portal (CAPC) marks data that the incident covers as Acknowledged. CA ADA automatically marks future data covered by an acknowledged incident as Acknowledged.
- **No Data**
  Displays a severity state (blank) to indicate that no data is available.
- **Unrated**

Displays a severity state (gray) to indicate that either there is insufficient past data (two full business days of data are needed) to establish a threshold, or there were not enough observations to exceed the minimum observations threshold.

- **Normal**
  Displays a severity state (green) to indicate that the metric value is between zero and the Minor threshold.
- **Minor**
  Displays a severity state (yellow) to indicate that the metric value exceeds the Minor threshold as defined in the CA ADA management console.
- **Major**
  Displays a severity state (orange) to indicate that the metric value exceeds the Major threshold as defined in the CA ADA management console.
- **Unavailable**
  Displays a severity state (red) to indicate that the application on a server is not running (unavailable). This rating appears when you click All Server Metrics in Settings. This rating is only applicable to a user-defined application where the CA ADA administrator has assigned servers to the application.

To isolate the component and metric contributing to a performance problem, click one of the worst-performing items at the top of the graph.

You may also edit the filter criteria for the view. By default, all data is included. Filter criteria include:

- **Metric Type**
  Select the Server metrics, Network metrics or Combined metrics that you want to use to filter the list of networks.
- **Context Settings**
  Displays the view filter. A context type of:
  - **Summary**
    Filters the view based on the selected group.
  - **Server**
    Filters the view based on the selected server.
- **Apply Changes**
  Select the scope of your changes from the **Apply Changes** drop-down.

Performance by Server

The Performance by Server view provides server performance data for the last 24 hours. If the time period for the report is longer than 24 hours, the view displays performance data only for the most recent 24 hours.

If you have enough observations, and if a threshold is available for each combination of application, server, network, and 5-minute period, CA Application Delivery Analysis (CA ADA) classifies the metric using one of these ratings:

- **Acknowledged**
  Displays a severity state (diagonal stripes) to indicate a CA ADA or CA NetOps Portal (CAPC) user acknowledged an incident. When this happens, CAPC marks data that the incident covers as Acknowledged. CA ADA automatically marks future data covered by an acknowledged incident as Acknowledged.
- **No Data**
  Displays a severity state (blank) to indicate that no data is available.
- **Unrated**
  Displays a severity state (gray) to indicate that either there is insufficient past data (two full business days of data are needed) to establish a threshold, or there were not enough observations to exceed the minimum observations threshold.
- **Normal**
  Displays a severity state (green) to indicate that the metric value is between zero and the Minor threshold.
- **Minor**

Displays a severity state (yellow) to indicate that the metric value exceeds the Minor threshold as defined in the CA ADA management console.
- **Major**
  Displays a severity state (orange) to indicate that the metric value exceeds the Major threshold as defined in the CA ADA management console.
- **Unavailable**
  Displays a severity state (red) to indicate that the application on a server is not running (unavailable). This rating appears when you click All Server Metrics in Settings. This rating is only applicable to a user-defined application where the CA ADA administrator has assigned servers to the application.

To isolate the component and metric contributing to a performance problem, click one of the worst-performing items at the top of the graph.

You may also the filter criteria for the view. By default, all data is included. Filter criteria include:

- **Metric Type**
  Select the Server metrics, Network metrics or Combined metrics you want to use to filter the list of networks.
- **Context Settings**
  Displays the view filter. A context type of:
  - **Summary**
    Filters the view based on the selected group.
  - **Server**
    Filters the view based on the selected server.
- **Apply Changes**
  Select the scope of your changes from the **Apply Changes** drop-down.

Performance Map by Server

The Performance Map by Server view provides a Top N view into the worst performing servers based on a particular metric. Configure this view to measure server performance using any of the available CA Application Delivery Analysis (CA ADA) metrics. If you receive an incident notification, or notice degraded performance in the Performance By Server view, use this view to drill into the Components reports on the Engineering report of the CA ADA management console. Use the Components reports to begin troubleshooting the metric details for the selected server.

To isolate the component and metric contributing to a performance problem, click one of the worst-performing servers at the top of the graph.

You may also edit the filter criteria for the view to display performance information for a different metric. Filter criteria include:

- **Metric Type**
  Choose a metric to filter the list of servers.
- **Context Settings**
  Displays the view filter. A context type of:
  - **Summary**
    Filters the view based on the selected group.
  - **Server**
    Filters the view based on the selected server.
- **Apply Changes**
  Select the scope of your changes from the **Apply Changes** drop-down.

Performance Maps

A performance map provides a Top N view into the worst performing networks, servers, or applications based on a particular metric.

If you receive an incident notification, or notice degraded performance in a Performance view, use this view to drill into the Components reports on the Engineering report of the CA Application Delivery Analysis (CA ADA) management console. Use the Components reports to begin troubleshooting the metric details for the selected application.

To isolate the component and metric contributing to a performance problem, click one of the worst-performing networks at the top of the graph.

Top Performance Map by Network

The Top Performance Map by Network view provides a Top N view into the worst performing networks based on a particular metric. Configure this view to measure application performance using any of the available CA Application Delivery Analysis (CA ADA) metrics. If you receive an incident notification, or notice degraded performance in the Performance By Network view, use this view to drill into the Components reports on the Engineering report of the CA ADA management console. Use the Components reports to begin troubleshooting the metric details for the selected application.

To isolate the component and metric contributing to a performance problem, click one of the worst-performing networks at the top of the graph.

You may also edit the filter criteria for the view to display performance information for a different metric. Filter criteria include:

- **Metric Type**
  Choose a metric to filter the list of networks.
- **Context Settings**
  Display the view filter. A context type of:
  - **Summary**
    Filters the view based on the selected group.
  - **Server**
    Filters the view based on the selected server.
- **Apply Changes**
  Select the scope of your changes from the **Apply Changes** drop-down.

Performance Map by Server

The Performance Map by Server view provides a Top N view into the worst performing servers based on a particular metric. Configure this view to measure server performance using any of the available CA Application Delivery Analysis (CA ADA) metrics. If you receive an incident notification, or notice degraded performance in the Performance By Server view, use this view to drill into the Components reports on the Engineering report of the CA ADA management console. Use the Components reports to begin troubleshooting the metric details for the selected server.

To isolate the component and metric contributing to a performance problem, click one of the worst-performing servers at the top of the graph.

You may also edit the filter criteria for the view to display performance information for a different metric. Filter criteria include:

- **Metric Type**
  Choose a metric to filter the list of servers.
- **Context Settings**
  Displays the view filter. A context type of:

  - **Summary**
    Filters the view based on the selected group.
  - **Server**
    Filters the view based on the selected server.
- **Apply Changes**
  Select the scope of your changes from the **Apply Changes** drop-down.

Performance Map by Application

The Performance Map by Application view provides a Top N view into the worst performing applications based on a particular metric. Configure this view to measure application performance using any of the available CA Application Delivery Analysis (CA ADA) metrics. If you receive an incident notification, or notice degraded performance in the Performance By Network view, use this view to drill into the Components reports on the Engineering report of the CA ADA management console. Use the Components reports to begin troubleshooting the metric details for the selected application.

To isolate the component and metric contributing to a performance problem, click one of the worst-performing networks at the top of the graph.

You may also edit the filter criteria for the view to display performance information for a different metric. Filter criteria include:

- **Metric Type**
  Choose a metric to filter the list of networks.
- **Context Settings**
  Display the view filter. A context type of:
  - **Summary**
    Filters the view based on the selected group.
  - **Server**
    Filters the view based on the selected server.
- **Apply Changes**
  Select the scope of your changes from the **Apply Changes** drop-down.

Performance Scorecard

The Performance Scorecard view provides a high-level view into the worst-performing applications by summarizing the percentage of time that an application is operating at different levels of performance, such as Normal, Minor, or Major.

If you have enough observations, and if a threshold is available for each combination of application, server, network, and 5-minute period, CA Application Delivery Analysis (CA ADA) classifies the metric using one of these ratings:

- **Acknowledged**
  Displays a severity state (diagonal stripes) to indicate that a CA ADA or CA NetOps Portal (CAPC) user acknowledged an incident. When this happens, CAPC marks data that the incident covers as Acknowledged. CA ADA automatically marks future data covered by an acknowledged incident as Acknowledged.
- **No Data**
  Displays a severity state (blank) to indicate that no data is available.
- **Unrated**
  Displays a severity state (gray) to indicate that either there is insufficient past data (two full business days of data are needed) to establish a threshold, or there were not enough observations to exceed the minimum observations threshold.
- **Normal**
  Displays a severity state (green) to indicate that the metric value is between zero and the Minor threshold.
- **Minor**

Displays a severity state (yellow) to indicate that the metric value exceeds the Minor threshold as defined in the CA ADA management console.

- **Major**
  Displays a severity state (orange) to indicate that the metric value exceeds the Major threshold as defined in the CA ADA management console.
- **Unavailable**
  Displays a severity state (red) to indicate that the application on a server is not running (unavailable). This rating appears when you click All Server Metrics in Settings. This rating is only applicable to a user-defined application where the CA ADA administrator has assigned servers to the application.

To isolate the component and metric contributing to a performance problem, click one of the worst-performing items at the top of the graph.

You may also edit the filter criteria for the view. By default, all data is included. Filter criteria include:

- **Metric Type**
  Choose a metric to filter the list.
- **Context Settings**
  Displays the view filter. A context type of:
  - **Summary**
    Filters the view based on the selected group.
  - **Server**
    Filters the view based on the selected server.
- **Apply Changes**
  Select the scope of your changes from the **Apply Changes** drop-down.

Performance Views

Performance views show the performance data for servers, networks, and applications from the last 24 hours. If the time period for the report is longer than 24 hours, the view only displays performance data for the most recent 24 hours.

If there are enough observations, and if a threshold is available for each combination of application, server, network, and 5-minute period, CA Application Delivery Analysis classifies the metric.

Contents

Performance by Network

The Performance by Network view provides information about network performance. This view displays performance data from the last 24 hours. If the time period for the report is longer than 24 hours, the view only displays performance data for the most recent 24 hours.

If you have enough observations, and if a threshold is available for each combination of application, server, network, and 5-minute period, CA Application Delivery Analysis (CA ADA) classifies the metric using one of these ratings:

- **Acknowledged**
  Displays a severity state (diagonal stripes) to indicate a CA ADA or CA NetOps Portal user acknowledged an incident. When this happens, the CA NetOps Portal (CAPC) marks data that the incident covers as Acknowledged. CA ADA automatically marks future data covered by an acknowledged incident as Acknowledged.
- **No Data**
  Displays a severity state (blank) to indicate that no data is available.
- **Unrated**

Displays a severity state (gray) to indicate that either there is insufficient past data (two full business days of data are needed) to establish a threshold, or there were not enough observations to exceed the minimum observations threshold.

- **Normal**
  Displays a severity state (green) to indicate that the metric value is between zero and the Minor threshold.
- **Minor**
  Displays a severity state (yellow) to indicate that the metric value exceeds the Minor threshold as defined in the CA ADA management console.
- **Major**
  Displays a severity state (orange) to indicate that the metric value exceeds the Major threshold as defined in the CA ADA management console.
- **Unavailable**
  Displays a severity state (red) to indicate that the application on a server is not running (unavailable). This rating appears when you click All Server Metrics in Settings. This rating is only applicable to a user-defined application where the CA ADA administrator has assigned servers to the application.

To isolate the component and metric contributing to a performance problem, click one of the worst-performing items at the top of the graph.

You may also edit the filter criteria for the view. By default, all data is included. Filter criteria include:

- **Metric Type**
  Select the Server metrics, Network metrics or Combined metrics that you want to use to filter the list of networks.
- **Context Settings**
  Displays the view filter. A context type of:
  - **Summary**
    Filters the view based on the selected group.
  - **Server**
    Filters the view based on the selected server.
- **Apply Changes**
  Select the scope of your changes from the **Apply Changes** drop-down.

Performance by Server

The Performance by Server view provides information about server performance. This view displays performance data from the last 24 hours. If the time period for the report is longer than 24 hours, the view only displays performance data for the most recent 24 hours.

If you have enough observations, and if a threshold is available for each combination of application, server, network, and 5-minute period, CA Application Delivery Analysis (CA ADA) classifies the metric using one of these ratings:

- **Acknowledged**
  Displays a severity state (diagonal stripes) to indicate a CA ADA or CA NetOps Portal (CAPC) user acknowledged an incident. When this happens, CAPC marks data that the incident covers as Acknowledged. CA ADA automatically marks future data covered by an acknowledged incident as Acknowledged.
- **No Data**
  Displays a severity state (blank) to indicate that no data is available.
- **Unrated**
  Displays a severity state (gray) to indicate that either there is insufficient past data (two full business days of data are needed) to establish a threshold, or there were not enough observations to exceed the minimum observations threshold.
- **Normal**
  Displays a severity state (green) to indicate that the metric value is between zero and the Minor threshold.
- **Minor**

Displays a severity state (yellow) to indicate that the metric value exceeds the Minor threshold as defined in the CA ADA management console.

- **Major**
  Displays a severity state (orange) to indicate that the metric value exceeds the Major threshold as defined in the CA ADA management console.
- **Unavailable**
  Displays a severity state (red) to indicate that the application on a server is not running (unavailable). This rating appears when you click All Server Metrics in Settings. This rating is only applicable to a user-defined application where the CA ADA administrator has assigned servers to the application.

To isolate the component and metric contributing to a performance problem, click one of the worst-performing items at the top of the graph.

You may also the filter criteria for the view. By default, all data is included. Filter criteria include:

- **Metric Type**
  Select the Server metrics, Network metrics or Combined metrics you want to use to filter the list of networks.
- **Context Settings**
  Displays the view filter. A context type of:
  - **Summary**
    Filters the view based on the selected group.
  - **Server**
    Filters the view based on the selected server.
- **Apply Changes**
  Select the scope of your changes from the **Apply Changes** drop-down.

Performance by Application

The Performance by Server view provides information about application performance. This view displays performance data from the last 24 hours. If the time period for the report is longer than 24 hours, the view only displays performance data for the most recent 24 hours.

If you have enough observations, and if a threshold is available for each combination of application, server, network, and 5-minute period, CA Application Delivery Analysis (CA ADA) classifies the metric using one of these ratings:

- **Acknowledged**
  Displays a severity state (diagonal stripes) to indicate a CA ADA or CA NetOps Portal (CAPC) user acknowledged an incident. When this happens, CAPC marks data that the incident covers as Acknowledged. CA ADA automatically marks future data covered by an acknowledged incident as Acknowledged.
- **No Data**
  Displays a severity state (blank) to indicate that no data is available.
- **Unrated**
  Displays a severity state (gray) to indicate that either there is insufficient past data (two full business days of data are needed) to establish a threshold, or there were not enough observations to exceed the minimum observations threshold.
- **Normal**
  Displays a severity state (green) to indicate that the metric value is between zero and the Minor threshold.
- **Minor**
  Displays a severity state (yellow) to indicate that the metric value exceeds the Minor threshold as defined in the CA ADA management console.
- **Major**
  Displays a severity state (orange) to indicate that the metric value exceeds the Major threshold as defined in the CA ADA management console.
- **Unavailable**

Displays a severity state (red) to indicate that the application on a server is not running (unavailable). This rating appears when you click All Server Metrics in Settings. This rating is only applicable to a user-defined application where the CA ADA administrator has assigned servers to the application.

To isolate the component and metric contributing to a performance problem, click one of the worst-performing items at the top of the graph.

You may also the filter criteria for the view. By default, all data is included. Filter criteria include:

- **Metric Type**
  Select the Server metrics, Network metrics or Combined metrics you want to use to filter the list of networks.
- **Context Settings**
  Displays the view filter. A context type of:
  - **Summary**
    Filters the view based on the selected group.
  - **Server**
    Filters the view based on the selected server.
- **Apply Changes**
  Select the scope of your changes from the **Apply Changes** drop-down.

Top Performance by Network

The Top Performance Map by Network view provides a Top N view into the worst performing networks based on a particular metric. Configure this view to measure application performance using any of the available CA Application Delivery Analysis (CA ADA) metrics. If you receive an incident notification, or notice degraded performance in the Performance By Network view, use this view to drill into the Components reports on the Engineering report of the CA ADA management console. Use the Components reports to begin troubleshooting the metric details for the selected application.

To isolate the component and metric contributing to a performance problem, click one of the worst-performing networks at the top of the graph.

You may also edit the filter criteria for the view to display performance information for a different metric. Filter criteria include:

- **Metric Type**
  Choose a metric to filter the list of networks.
- **Context Settings**
  Display the view filter. A context type of:
  - **Summary**
    Filters the view based on the selected group.
  - **Server**
    Filters the view based on the selected server.
- **Apply Changes**
  Select the scope of your changes from the **Apply Changes** drop-down.

Top Performance Map by Network

The Top Performance Map by Network view provides a Top N view into the worst performing networks based on a particular metric. Configure this view to measure application performance using any of the available CA Application Delivery Analysis (CA ADA) metrics. If you receive an incident notification, or notice degraded performance in the Performance By Network view, use this view to drill into the Components reports on the Engineering report of the CA ADA management console. Use the Components reports to begin troubleshooting the metric details for the selected application.

To isolate the component and metric contributing to a performance problem, click one of the worst-performing networks at the top of the graph.

You may also edit the filter criteria for the view to display performance information for a different metric. Filter criteria include:

- **Metric Type**
  Choose a metric to filter the list of networks.
- **Context Settings**
  Display the view filter. A context type of:
  - **Summary**
    Filters the view based on the selected group.
  - **Server**
    Filters the view based on the selected server.
- **Apply Changes**
  Select the scope of your changes from the **Apply Changes** drop-down.

Register and Configure Network Flow Analysis

This use case describes how an administrator can register CA Network Flow Analysis (CA NFA) as a data source for an instance of CA NetOps Portal (CAPC) or CA NetQoS NetOps Portal (CA NPC), and perform initial configuration tasks. As soon as you register CA Network Flow Analysis as a data source, you can perform some key administrative tasks, such as setting up SNMP profiles, user accounts, user permissions, and groups. You perform these tasks in the Console for CAPC or CA NPC. The settings that you specify are synchronized down to CA NFA.

You must have the Administrator role in both product interfaces to perform the steps that are outlined in the accompanying procedures.

Change the Domain of Interfaces and CVIs in NFA

Interfaces and CVIs inherit their initial tenant-domain setting from the parent router and Harvester. The setting is inherited when the parent Harvester is added and the router and interfaces first become active. If the Harvester is not associated with a custom domain, the routers and interfaces are assigned to the Default Domain as they become active.

You can edit the settings for interfaces and CVIs to associate them with any tenant and domain at any time. The setting does not have to match the parent router or Harvester.

Changing this setting can affect which operators have access to the interface's data. The setting does not affect which SNMP profiles are used for polling. The router tenant determines the set of SNMP profiles for polling.

Follow these steps:

1. Open the Active Interfaces page.
   a. Select **Administration** from the NFA console menu.
      The Administration page opens.
   b. Select Interfaces: **Physical & Virtual** from the Administration menu.
      The Active Interfaces page opens.
2. Select check boxes next to one or more interfaces that you want to associate with a tenant and domain.
   - To search for parent routers, interfaces, or CVIs, type all or part of a router IP address, a router or interface name, or an interface description in the Search field, and then click **Search**. Expand the router details.
   - To navigate to an interface or CVI manually, go to the page that contains the parent router, and click the arrow next to the router name. The router details expand to show the interfaces and CVIs.
3. Click **Edit**.
   The editing dialog opens. The Domain selection list is included in the dialog only if multiple domains exist.
4. Select a tenant / domain option from the Domain list.

5. Click **Save**.
   The dialog closes. The changes are shown on the Active Interfaces page.

   NOTE
   You can also change the tenant-domain setting for harvesters and routers.

Configure Network Flow Analysis in Performance Center

As an IT engineer or tools administrator, you can start monitoring network traffic composition and utilization statistics by deploying Network Flow Analysis (NFA).

This use case walks you through the steps to configure NFA as a data source for NetOps Portal (CAPC) or CA NetQoS NetOps Portal (CA NPC). This use case also describes how to configure CA NFA to receive flow data, and to configure the following elements, which are administered in CAPC or CA NPC:

- SNMP profiles
- IP domains
- User accounts, permissions, and roles
- Groups

Before you perform these procedures, install NFA and CAPC or CA NPC in your environment. You must have administrative role rights for both products to perform the tasks that are described here.

Configure Flow Collection

Configure the routers in CA Network Flow Analysis (CA NFA) to verify that they are sending data to the Harvesters.

CA NFA can begin to collect flows as soon as you complete the following tasks:

Add One or More Harvesters

Add one or more Harvesters to enable data to be processed and displayed.

Prerequisite:

Recommended: If you have not already done so, register CA Network Flow Analysis (CA NFA), and set up the domains before you add any Harvesters.

Follow these steps:

1. Open the NFA console, logged in with Administrator rights. For example, enter the following address in a browser:
   http://<ipaddress>/ra/
   User name: *admin*
   Password: *admin*
2. Open the Harvester page:
   a. Select Administration from the NFA console menu.
      The Administration page opens.
   b. Select System: Harvester from the Administration page menu.
      The Harvester page opens and displays the current list of Harvesters.
3. Click Add.
   The Add Harvester dialog opens.
4. Enter the following information:
   – **IP Address**
     Address of the Harvester server.
   – **Description**

Identifying text about the Harvester, which appears in the Harvester page table.

– **Domain**
Parent tenant and domain combination for the Harvester and for any routers and interfaces that begin to supply data to the Harvester.
Changing this setting affects the tenant-domain association for new routers that begin exporting flow data and any new interfaces that begin generating flow.
In a multi-tenant environment, the Harvester tenant affects which SNMP profiles are available for routers to poll interfaces.
The domain affects which operators and reports have access to the data from routers and interfaces.

NOTE
This option is visible only in an environment that contains multiple domains.

5. Click Save.
The new Harvester is added and appears in the Harvester list, provided that the IP address passes the connection test. If the test connection to the web service fails, an error message opens.
The usual process is to add one or more Harvesters, then configure the router interfaces to export flow to the Harvesters. If you configure the routers to export flow to the Harvesters first, the NFA console immediately begins to collect data from the new Harvester. In this case, the domain for the routers is set at the time you add the parent Harvester.

NOTE
Make sure that the Harvesters that you add have not been deleted from the Harvester page previously. To add a Harvester instance successfully in CA NFA 2.4 after deleting it, the Harvester installation server must be re-imaged and the Harvester software must be re-installed.

Verify the Harvester Domain

Verify that each Harvester is associated with the appropriate domain before you set up routers to export flow data. If you have not already done so, set up any needed custom tenants and domains before you proceed.

NOTE
The tenant feature is applicable only to deployments that include CA NetOps Portal. If your deployment uses CA NetQoS NetOps Portal, the tenant setting is always Default Tenant.

Follow these steps:

1. Open the Harvester page:
   a. Select Administration from the NFA console menu.
   The Administration page opens.
   b. Select System: Harvester from the Administration page menu.
   The current list of Harvesters appears.
2. Click Edit next to the Harvester that you want to edit.
The Edit Harvester dialog opens.
3. (Optional) Change the Domain setting (tenant-domain combination) as needed.
Default: Default Tenant \ Default Domain.
You can also change the IP Address and Description.

NOTE
If no custom IP domains have been created, the Harvester table includes only the IP Address and Description columns.

4. Click Save when your changes are complete.
Your changes are saved immediately.

Set Up the Routers

Enable NetFlow on each CA Network Flow Analysis router by completing the steps in this topic. You can configure routers to export any of the following flow protocols:

- NetFlow v5, v7, v9, and Sampled NetFlow
- sFlow version 5
- IPFIX, J-Flow, cFlow, and NetStream flow that complies with the standards for NetFlow v5, v7, or v9

Configure flow from each source to be exported to a single Harvester. If flow from one source is exported to multiple Harvesters, a number of problems result. If this occurs, contact CA Support for help.

NetFlow provides a broad view of your network packet streams by creating flow records for all packets. The data from these flow records represents all packets. Sampled NetFlow/IPFIX and sFlow take samples from your packet streams, producing fewer flow records and lessening the impact to a collector. The lower your sampling rate, the less precise the data is likely to be.

In order for data from non-sampled flows to appear in reports of 15-minute (historical) data, the following minimum fields are required:

- One of the following: 1 - IN_BYTES, 85 - IN_PERMANENT_BYTES, 231 - FW_INITIATOR_OCTETS, or 232 - FW_RESPONDER_OCTETS
- 4 - PROTOCOL
- 7 - L4_SRC_PORT
- 8 - IPV4_SRC_ADDR
- 10 - INPUT_SNMP
- 11 - L4_DST_PORT
- 12 - IPV4_DST_ADDR
- 14 - OUTPUT_SNMP

Complete these tasks:

1. Back up the current router configuration.
2. Configure NetFlow export for each interface individually:
   a. Set the flow export version.
   b. Set the flow source IP address. Cisco recommends that you configure a loopback source interface. The IP addresses of non-loopbacked interfaces can change.
   c. Set the flow destination IP address and set the destination port to 9995. If you are using a custom value for the harvester listening port, use that value as the destination port. The port values must match or the Harvester does not receive flow data.
   d. Set the flow expiration timeout to 1 minute.
3. Enable flow for each interface.
   – NetFlow v5 or v5-compatible flow:
     - Monitoring multiple interfaces on a router: Use either all ingress or all egress. Use the same option for all of the interfaces. Ingress and egress values may vary slightly due to routers dropping packets and changing ToS values as traffic travels between interfaces.
     - Monitoring a single known interface on a router: Use ingress and egress. This option results in fewer total flows from the router to the Harvester and puts less load on the network and the Harvester.
   – NetFlow v9 or v9-compatible flow:
     The Harvester identifies and deduplicates multiple flows on a single router, so you can use ingress and egress on multiple interfaces. You may find it most efficient to use this option for two or three interfaces. You have the option to enable ingress and egress across all interfaces, but this configuration may put an unnecessary burden on the Harvester.
4. Configure SNMP index persistence on each router that supports this feature.

Add DSAs (Three-Tier Deployment)

We recommend that you add at least one DSA within 30 minutes of starting flow collection. Until you add a DSA to your three-tier deployment, 15-minute data is not available for reports. Reports that show a time range of more than 2 hours do not show any data.

NOTE

Do not add a DSA instance instead of editing the IP address of a retired DSA. In this case, routers continue to send data to the retired DSA--and that data is not available in reports. If you have to delete a DSA, contact CA Support for assistance.

Add a DSA

Follow these steps:

1. Open the NFA console, logged in with Administrator rights.
2. Display the DSA page in the NFA console user interface:
   a. Select Administration from the NFA console menu.
      The Administration page opens.
   b. Select System: DSA from the Administration menu.
      The DSA page opens and shows a list of the current DSAs.
3. Click Add.
   The Add DSA dialog opens.
4. Enter the IP address for the DSA server.
5. Click Test Connection.
   A connection test is performed to determine whether the NFA console server can contact the DSA server and can verify that MySQL is installed. If the test succeeds, a "Test success" message opens.
6. Respond to the test results:
   a. If the Test success message opens, click OK to close it.
   b. If an error message opens, close the error message and respond as described in Troubleshoot DSA Addition.
7. Once the test completes successfully, click Save in the Add DSA dialog.
   The connection test is performed, followed by a test to locate DSA settings on the target server.
8. Note the test results:
   – If no error message opens, the tests have succeeded. The following events result:
     • The dialog closes and the DSA is added to the DSA list.
     • The Harvesters begin to include the new DSA in the destinations for new enabled interfaces that report 15-minute data.
     • The NFA console pushes settings down to the new DSA.
     • The DSA is configured to retrieve 15-minute data files from the NFA console.
     • Data from the DSA begins to be available for reports in approximately 30 minutes.
   – If an error message opens, close the error message and respond as described in Troubleshoot DSA Addition.

NOTE

- If a DSA does not begin to collect the 15-minute data within 30 minutes after flow collection begins, problems can result. The processed data accumulates on the NFA console server and processing slows. If the problem continues, the NFA console stops collecting the 15-minute data and unprocessed data accumulates on the Harvesters. If Watchdog traps are configured, the Watchdog sends out alerts that the Harvester or Reaper

is falling behind. If the problem is left unchecked, the CA Network Flow Analysis services on the Harvesters may stop running.
- Each DSA is enabled to collect data for a maximum of 5,000 enabled interfaces that have reported data.
- For information about the data types, storage lifespan, minimum thresholds, and report types for the 15-minute data that is stored on DSA servers, see the topic 15-Minute Data in the *CA Network Flow Analysis Administrator Guide.*

Test Connection Errors

Use the following tips for troubleshooting error messages that may open when you click Test Connection in the Add DSA dialog

- "An invalid server IP address was entered":
  You entered an IP address in an invalid format. Make sure that you enter the IP address correctly.
- "System.Web.Services.Protocols.SoapException...'":
  Verify that the NetQoS MySql service is running on the NFA console server. If the service is not running, start it.
- "Unable to connect to any of the specified MySQL hosts":
  Start the NetQoS MySql service on the DSA server. Verify that the NetQoS MySql service is running on the DSA server. If the service is not running, start it.
- "Unknown database 'nqrptr'":
  The DSA database nqrptr was not found on the target server. Verify that the DSA software installation was successful.

Save Errors

Use the following tips for troubleshooting error messages that may open when you click Save in the Add DSA dialog

- "An existing record is already in use":
  Enter the IP address of a DSA that is not already in the DSA list.
- "Connection must be valid and open'":
  Verify that the NetQoS MySql service is running on the DSA server. If the service is not running, start it.
- "System.Web.Services.Protocols.SoapException...'":
  Verify that the target server is running and can be reached by the NFA console server. Verify that the NetQoS MySql service is running on the NFA console server. If the service is not running, start it.
- "Table 'nqrptr.settings' doesn't exist"
  The DSA settings table was not found. The DSA software was not installed successfully on the target server.
  - Verify that you entered the correct IP address for the DSA--not the IP address for a Harvester or for the NFA console.
  - Verify that the DSA software installation was successful.

Configure Traps

Trap configuration is complete when you have finished the following tasks:

- Create the traps that you need, as described in the *CA Network Flow Analysis Administrator Guide "*Create Traps" topic.
- Enable traps to be displayed as events in the NetOps Portal Console:
  a. Open the Application Settings page in the NFA console.
  b. Set the Trap Destination value to match the IP address of one of the following servers:
     - (CA PC) NFA console or stand-alone server that is registered as a data source
     - (NPC) Event Manager server
- (Optional) Enable Watchdog trap notifications to be sent to your trap receiver: Open the Watchdog Settings page in the NFA console. Configure values for the Trap Destination, Email Address, and other Watchdog settings.
- (Optional) Verify that the events are displayed on the NetOps Portal Console--on the Events page (CA PC) or the Event List page (NPC). If the events are not shown as expected, verify that the following conditions are met:

- The logs show that events have been generated and have been forwarded to the Event Manager.
- The Event Manager host name is resolvable by the DNS server for CA Network Flow Analysis.
- The Trap Destination value on the Application Settings page in the NFA console matches the IP address of one of the following servers:
  - (CA PC) NFA console or stand-alone server that is registered as a data source
  - (NPC) Event Manager server
- (NPC Only): The Event Manager is installed.

Set Up User Accounts

One predefined user account, admin, is included with the installation of CA Network Flow Analysis. The admin account has full administrative privileges.

The administrator must create a user account for each person who will use the product--administrators and operators. Custom user accounts enhance security and take advantage of the narrowly defined role rights that determine access to product features and data.

Custom user accounts are best deployed in a well-planned system that includes custom groups. Custom groups are assigned as permissions to let product operators view only the data, menus, and dashboards that they need to perform their daily tasks.

View a List of User Accounts in NFA

The Manage Users page lets you see high-level settings for user accounts. In a multi-tenant environment, the global administrator sees a list of user accounts that are not explicitly associated with a tenant. Tenant administrators only see user accounts for their tenant.

Before you create any custom user accounts, only the two factory user accounts are available.

Follow these steps:

1. Log in as a user with the required administrative role rights.
2. Select **Administration**, **User Settings**, and click **Users**.
   The current list of user accounts appears.

   NOTE
   Tenant administrators only see the items that are associated with their tenant.

   The table includes the following information about each user account:
   - **Name**
     Login name for the user account.
   - **Role**
     Role assigned to the user account.
   - **CAPC Privilege**
     Identifies the level of access to registered data sources.
   - **Permission**
     Lists the permission groups that are assigned to this account. Permission groups are shown as nested locations within the Groups tree. If this user is able to create custom groups that are not visible to other users, "My Custom Groups" are indicated.
     Default: '/All Groups'.
   - **Status**
     Indicates whether the user account is enabled or disabled.

Add User Accounts

Add a user account for each person who will operate the products. For security purposes, operators should not share user accounts.

> **NOTE**
> If you register CA Network Flow Analysis as a data source for CA NetQoS NetOps Portal, the steps are slightly different.

Follow these steps:

1. Log in to the NetOps Portal Console as a user with the required administrative role rights.
2. Confirm that the required roles and groups exist.
3. Select **Administration**, **User Settings**, **Users**.
   The current list of user accounts is displayed.
4. Click **New**.
   The Create New User wizard (CA PC) or Add User page (NPC) opens.
5. Enter the appropriate information for the account parameters.
6. Add permission groups to the user account, described in Assign Product Privileges.
7. Click **Save**.
   The new user appears in the list of user accounts.

Add Role Rights for Users

If the predefined user roles do not fit your requirements, you can add custom user roles. Ideally, you create the roles that each unique product operator needs to be able to perform his or her job responsibilities.

Custom roles work best within a system of custom groups. Custom groups let you precisely grant access to dashboards and product features while restricting access to sensitive data. The same groups that you create to organize data can serve as “permission groups” when you set up user account permissions.

A new role has no rights until you add them. The following graphic shows the Add Role dialog in the CA NetOps Portal Console with a role that is beginning to be defined.

Follow these steps:

1. Log in to the NetOps Portal Console as a user with the required administrative role rights.
2. Navigate to the Manage Roles or Roles List page.
   The page displays the current list of roles.
3. Click New.
   The Add Role dialog opens.
4. Supply the required information and make selections in the fields provided.
5. Specify the menus that will be visible to users with the new role:
   a. Select Menu Set (CA PC) or select a menu or product from the list at the bottom of the dialog (NPC).
   b. Click Edit.
      The Edit Menu Set dialog opens. Menus in the 'Available Menus' list can be added to the role.
   c. Click an item on the left that you want to add to the role, then click the right arrow.
      Use Shift + Click or Ctrl + Click to select multiple items.
      Each selected item moves to the Selected Menus list.
   d. (Optional) Use the Up and Down arrows to move items around in the list. The order of menus in the list determines their order on the Dashboards tab.
   e. Click OK.
      You return to the Add Role page.

6. Set the NetOps Portal rights for the role:
   a. Select NetOps Portal (CA PC) or NetQoS NetOps Portal (NPC).
   b. Click Edit.
      A dialog opens, which you use to select NetOps Portal access rights.
   c. Click an item on the left that you want to add to the role, then click the right arrow.
      The access right moves to the Selected Rights list.
   d. (Optional) Use the Up and Down arrows to move items around in the list. The order of role rights determines their priority in cases where rights overlap.
   e. Click OK.
      You return to the Add Role page.
7. Set the CA Network Flow Analysis rights for the role:
   a. Select the name of the registered CA Network Flow Analysis instance.
   b. Click Edit.
      A dialog opens, which you use to select access rights for CA Network Flow Analysis in the same way you selected access rights for NetOps Portal.
   c. When the access rights are set up correctly, click OK.
      The new role is created and appears in the Role List.
8. Repeat the previous step to set the rights for any additional data source that you want to include.
9. Click Save on the Add Role page.
   You return to the Manage Roles page (CA PC) or Roles List page (NPC).

NOTE

When you finish creating a role, assign it to a user account as a separate step. Roles are inoperative until they are assigned to user accounts. Only users with the 'Administer Users' and 'Administer Roles' role rights can assign roles to user accounts.

Assign Permission Groups to User Accounts

Individual operators require data access permissions to monitor data in the products. Access permissions are based on groups. You can assign access permissions according to your plan for custom groups. Your goal as the administrator is to make sure that all operators see only the data they require to do their job.

For example, suppose you create custom groups and assign them as permissions to IT staff. When staff members log in to NetOps Portal, they can view data from the systems that are assigned to them.

Follow these steps:

1. Log in to the NetOps Portal Console as a user with administrative privileges.
2. Click **Administration**, **User Settings: Users**. The Manage Users page opens.
3. Select a user account that you want to change, and click Edit.
   The Edit User wizard or dialog opens.
4. Display the permission groups:
   – (CAPC) Click the **Access Permissions** button.
   – (NPC) Locate the **Permission Groups** pane in the middle of the page.

   The group settings are displayed.
5. Add permission groups to the user account
   – Expand the groups in the **Available Groups** tree on the left so that subgroups are shown.
   – Select a group or subgroup.
   – Click the right arrow or **Add** button to add the group.
   – Repeat as necessary.

The selected permission groups appear in the Selected Groups pane.

6. Select the default group for the user--the data that appears by default in the dashboards for the user:
   – (CAPC) Right-click the target group and select **Make Default**.
   – (NPC) Select the target group and click **Make Default**.
7. Click **Save**.
   The changes are saved to the user account, and you return to the Manage Users page.
   When the user logs in, data from the default group appears in dashboards by default.

Assign Product Privileges

Each registered data source has its own product privilege setting, which grants unique privileges within that product interface. Administrators give users product privileges for each data source. For example, the product privilege determines whether a user can log in to CA Network Flow Analysis or drill down from a NetOps Portal view to details in the NFA console. Privileges are specific to the data source instance.

The default administrator account, admin, is locked to prevent changes to product privileges. This account must have Administrator privileges for all registered data sources. If you select a group of accounts that includes the admin account, you cannot edit the product privileges for any of the selected accounts.

CA Network Flow Analysis Product Privileges

A user must have product privileges for the CA Network Flow Analysis data source to log in to the NFA console. Product privileges also determine whether a user can access the Administration page, and can perform certain functions:

- **Administrator**
  Gives access to the Administration page in the NFA console and to all functions. Functions include creating and managing user accounts, roles, groups, SNMP profiles, and scheduling for reports.
- **Power User**
  Gives user-level access and any additional abilities that the Role setting grants. For CA Network Flow Analysis, the Power User privilege is equivalent to the Administrator privilege.
- **User**
  Gives access to Top Interfaces reports and Interface Utilization reports on the Enterprise Overview page.
  A User with the appropriate Permission Group settings also has access to the following reports:
  – Top Hosts and Top Protocols reports on the Enterprise Overview page, if the user also has access to All Groups
  – Interfaces page reports for the interfaces that are accessible to the user
  – Existing reports on the Custom Reporting, Flow Forensics, and Analysis pages
  – Menus that an administrator has assigned to the User role
  The Role and Permission Group settings determine whether the User also can run existing reports, create reports, and manage reports. To create reports, a User must have access to All Groups.
- **None**
  Has no access to a data source. The user who has this product privilege cannot log in to the NFA console or drill down from a NetOps Portal view to the NFA console. By default, all users have this product privilege setting for all data sources.

  NOTE
  The same user account can have different privileges for different data sources.

Set Up Groups

We recommend that you create custom groups to help manage items in the NetOps Portal Console. Custom groups are required to let operators see performance data from the routers they manage.

Properly configured, groups can prevent operators from viewing particular types of data for security reasons. The administrator can selectively grant users access to data in their area of responsibility, such as a physical location or subnet.

Before you start creating groups, plan a strategy and a structure. Consider the types of access permissions that operators require to perform their monitoring duties. If necessary, you can discuss your organizational and monitoring goals with a CA technical representative.

Create groups under the All Groups node in the Groups tree, or within an existing custom or site group. You cannot add groups to system groups, which appear "locked" in the Groups tree.

You can add a maximum of 2000 child groups to a parent group.

Follow these steps:

1. Log in to the NetOps Portal Console as a user with the required administrative role rights.
2. Navigate to the Manage Groups page.
   The page displays current groups in a tree structure.
3. Expand nodes in the Groups tree to find a location for the new group.
4. Right-click the node, and select Add Group (CA PC) or Add New Group (NPC).
   The Add Group window opens with the New tab selected by default.
5. Supply values for the following parameters:
   - **Group Name**
     Specifies a name for the group. Do not use the following special characters in group names: /&\,%.
   - **Description**
     (Optional) Helps you identify the group.
6. Confirm the setting for the following parameter:
   - **Include the children of managed items**
     Adds the children of managed items automatically when the items are added to this group. If this option is disabled and you add a router, the router interfaces are not included. As a result, the data from those interfaces is not visible in drilldown views.
     Default: Selected (CA PC) or Not Selected (NPC).
     Icon
     Note: Clear this option for a custom group that contains routers or the group will not be usable in in the NFA console.
7. Select Custom or Site from the Group Type list.
   If you selected Site as the type, specify values for the additional parameters that appear, including Location.
8. Click Save.
   The new group appears in the Groups tree.
   The group contains no items until you add them. You have two options for adding items to a custom group:
   - Manually populate the group by adding items in the Manage Groups interface.
   - Create rules to manage group membership.

Results of Unregistering

On rare occasions, you may want to unregister CA Network Flow Analysis. For example, you would unregister a CA Network Flow Analysis instance before registering it with a different NetOps Portal instance. Unregister only if it is really necessary.

If you unregister CA Network Flow Analysis, the following rules apply:

- Users: Users without product privileges to CA Network Flow Analysis in NetOps Portal are deleted from the CA Network Flow Analysis database. Existing User IDs remain unchanged. You cannot add new users or edit user account settings while unregistered.
- Roles: Roles are not deleted. Users continue to have their previous roles. Existing Role IDs remain unchanged. You cannot change the roles or permissions for existing users while unregistered.
- Groups:
  - Groups that do not exist in CA Network Flow Analysis are deleted. You cannot add or change groups while unregistered.
  - Nested groups that are associated with an interface are displayed as interface groups in the NFA console.
  - Groups that are not associated with an interface are displayed as permissions.
- Single Sign-On and LDAP: Single Sign-On and LDAP values remain unchanged.

NOTE
For a more detailed description of the results of unregistering during an upgrade of CA Network Flow Analysis, see the CA Network Flow Analysis documentation.

Register Network Flow Analysis

Register the product in the NetOps Portal Console--on the Manage Data Sources page (CA PC) or the Data Source List page (NPC).

NOTE
For information about the number of data sources that you can use, see the *Release Notes* for your NetOps Portal version.

Follow these steps:

1. Verify that no one else is running a session of Network Flow Analysis. If multiple users write to the database simultaneously, problems can result.
2. Log in to the NetOps Portal Console as a user who has the Administrator role.
3. Click Admin, Data Sources.
   The current list of registered data sources are shown on the Manage Data Sources page (CA PC) or the Data Source List page (NPC).
4. Click Add (CA PC) or New (NPC).
   The Add Data Source dialog opens.
5. Select the type of data source you want to add from the Source Type list.

   NOTE
   All CA products that can be registered as data sources are shown in the Source Type list. The list is not filtered to hide products that are installed already.
6. Enter the Host Name of the data source.
   The hostname is the IP address or DNS hostname of the server that hosts the database for the data source. For a distributed deployment of the product, enter the hostname of the NFA console or stand-alone server.
7. Enter the port to use for contacting CA Network Flow Analysis.
   For more information about the port setting, see the *CA Single Sign-On User Guide*.
8. Select the protocol to use for contacting the data source. Select **https** if your network uses SSL for communications. Verify that you have configured the system correctly before you select the **https** option.

   NOTE
   For more information about using SSL for communication between the product and NetOps Portal, see the *CA Single Sign-On User Guide.*
9. (Optional) Enter a Display Name for the CA Network Flow Analysis data source.

If you do not enter a name, a default name is created by combining the data source type and hostname. For example, you can use a default name like NetworkFlowAnalysis@xxx.x.x.xx or you can use a name like NetworkFlowAnalysis_NewYork.

10. Confirm whether the web console address is the same as the Host Name. If it is not, take the following steps:
    - Clear the 'Same as Data Source' check box.
    - Provide the web console hostname, port, and protocol.
11. Click Save.
    The updated list shows the data sources that are registered.

Test Data Source Connections (Register and Configure NFA Use Case)

In most cases, the status indicates that data source registration has completed successfully. If the status indicates an error, use the test feature on the Manage Data Sources page.

Click the Test button to run a test that confirms the proper registration and connection of a new data source. The test checks for version compatibility, and verifies that the data source is not registered with a different instance of the CA NetOps Portal software.

If the test fails, verify that the server name or IP address is accurate for the source type. For more information, see Data Source Test Fails.

Verify IP Domains

The IP domains feature helps to address potential IP address conflicts. Domain identifiers indicate that two managed items that otherwise appear as duplicate IP addresses are actually two different managed items.

IP domains also let a global administrator in NetOps Portal control which managed items are visible to and accessible by a particular administrator or user.

Information about custom IP domains is sent down to the data sources during synchronization. Domains are available for use during configuration. You can use the NFA console Administration functions to add interfaces, custom virtual interfaces (CVIs), routers, and Harvesters to the custom domains that you create.

During the initial setup, verify that the existing IP domains are adequate to monitor your environment. To see the domains, complete one of the following actions in the NetOps Portal Console:

- (CA PC) Select **Administration**, **IP Domains**, and review the domains on the Manage IP Domains page.
- (NPC) Select **Administration**, **Groups**, and expand the All Groups tree on the Manage Groups page.

View the IP Domains List

IP domains are required for monitoring multiple environments with overlapping IP addresses. Set up all the domains that you need before you begin to export flow data.

Follow these steps:

1. Log in to the NetOps Portal Console as a user with the required administrative role rights.
2. Display the current domains:
   - (CA PC) Select **Administration**, **Configuration Settings: IP Domains**.
     The Manage IP Domains page opens and displays the current domains.
   - (NPC) Select **Administration**, **NetQoS Settings: Groups**, and expand the All Groups tree on the Manage Groups page.
     The current domains are shown under **All Domains**. To display the parameters for a domain in CA NetQoS NetOps Portal, select the domain and click the **Properties** tab.

If you have not created any custom IP domains, only the Default Domain appears in the list. This predefined domain has a 'null' setting for all parameters.

Any custom domains that you have created include values for the following parameters:

- **Name**
  Identifies the domain.
- **Description**
  (Optional) Describes this domain namespace, such as naming the enterprise that owns it.
- **Primary DNS Address**
  Is the IP address of the primary name server for this domain.
- **Primary DNS Port**
  Is the port number that the primary name server uses.
- **Secondary DNS Address**
  Is the IP address of the secondary name server for this domain. Can be the same as the primary address.
- **Secondary DNS Port**
  Is the port number that the secondary name server uses.

Add Custom IP Domains

Administrators can set the domain assignment for Harvesters, routers, interfaces, and custom virtual interfaces (CVIs). We recommend that you set up any custom tenants and domains that you need before you add the Harvesters.

Having the appropriate IP domains set up helps to achieve the following goals:

- Assign the correct tenant-domain when you add Harvesters so that their routers and interfaces inherit the correct associations. The routers have the appropriate SNMP profiles available to poll their interfaces.
- Make specific content accessible only to the operators who monitor the content.
- Enable Administrators to create domain-specific ToS labels, protocol groups, and Autonomous System (AS) names in CA Network Flow Analysis.
- Avoid IP address conflicts.
  For example, suppose that a router with a single IP address has interfaces that belong to different enterprises. The domain identifiers clarify that the interfaces are different managed items, even though they have the same IP address.

The Default Domain is created automatically. The Default Domain includes any items that are not assigned to a custom domain.

Follow these steps:

1. Log in to the NetOps Portal Console as a user with the required administrative role rights.
2. Display the current domains by completing one of the following actions:
   – (CA PC) Select **Configuration Settings: IP Domains**.
     The Manage IP Domains page opens and displays the current domains.
   – (NPC) Select **Administration**, **NetQoS Settings: Groups**, then expand the **All Groups** tree on the **Manage Groups** page.
     The current domains are shown under All Domains.

   If you have not created any custom IP domains, only the Default Domain appears in the list.
3. Create a domain:
   – (CA PC) Click **New**.
     The IP Domains Administration dialog opens.
   – (NPC) Right-click **All Domains** and select **Add New Domain**.
     The Add Domain dialog opens.
4. Specify the appropriate parameters in the provided fields.

The Device Name Alias (CAPC only) field indicates the alias to use for a managed device. A device alias is a user-configured name that is applied to the associated managed item in NetOps Portal. Click **Browse** to navigate to and import a CSV file of aliases. The CSV file contains a list of IP address-to-device alias mappings.
Aliases that are associated with the primary IP address of a device take precedence over aliases that are associated with any secondary IP addresses. Look for the primary IP address in the Address column of the Inventory Devices list. We recommend using the primary IP address of the device in the CSV file.
For example:

```
172.24.36.107,Austin Router
```

Browse to select the file and click Open.
If you include aliases for devices that you are managing, it can take up to 5 minutes to begin synchronizing these aliases with NetOps Portal.

> **NOTE**
> To remove an alias, import a CSV file that includes the IP address for the device and a blank alias column. To change an alias, modify the alias entry in the CSV file and reimport the file.

5. **Interface Description Override**
   (CA PC only) Indicates the alternate description to use for an interface. Interface descriptions appear in CA NetOps Portal already, but you can provide an alternate description. Click Browse to navigate to and import a CSV or TXT file of alternate descriptions. The file contains a comma-separated list of values that include the device IP address, interface name, interface description, and alternate interface description (alias) mappings.
   For example:

```
172.24.36.107,ethernet_7,vmxnet3 Ethernet Adapter,Connection to Dallas
```

> **NOTE**
> Use the primary IP address of the associated device in the CSV or TXT file. Secondary IP addresses are not supported. Look for the primary IP address in the Address column of the Inventory Devices list.

   Browse to select the file and click Open.
   If you include alternate descriptions for interfaces you are managing already, it can take up to 5 minutes to begin synchronizing these descriptions with CA NetOps Portal.

> **NOTE**
> You can use the same alternate interface descriptions for more than one interface. To remove an alternate description, import a CSV or TXT file that includes the IP address for the device, the interface name, the interface description, and a blank alias column. When you remove an alternate description, the original interface description reappears in NetOps Portal views.

> **WARNING**
> If you use a spreadsheet program to remove all the alternate descriptions from a CSV file, include a column heading for the interface description override column in the imported file. If you do not include this column heading, the original interface descriptions do not reappear in NetOps Portal views.

   To change a description, modify the alias entry in the CSV or TXT file and re-import the file.
   - **DNS Settings check box**
     (CA PC only) If selected, displays the Primary DNS/Port and Secondary DNS/Port options.
   - **Primary DNS Address**
     Is the IP address of the primary name server for this domain.
   - **Primary DNS Port**
     Is the port number that the primary name server uses.
   - **Secondary DNS Address**
     Is the IP address of the secondary name server for this domain. Can be the same as the primary address.
   - **Secondary DNS Port**
     Is the port number that the secondary name server uses.
   - **Enable DNS Proxy Address**

(NPC only) Indicates whether the proxy address is enabled for this IP domain.
- **DNS Proxy Address**
  (NPC only) Is the IP address of the DNS proxy server.
  This setting is required only if your network is located behind a DNS proxy server.

6. Click **Save**.The new IP domain appears on the page.
7. Repeat the steps as required to add more IP domains.

Verify SNMP Profiles for NFA

CA Network Flow Analysis sends secure SNMP information to NetOps Portal at registration. This information is transformed into SNMP profiles that contain the information necessary to enable secure queries of device MIBs using SNMP. Profile information is updated at each synchronization.

During initial product setup, verify that the available SNMP profiles are adequate to monitor your environment. The available SNMP profiles are listed on the Manage SNMP Profiles page (CAPC) or SNMP Profile List page (NPC).

View SNMP Profiles

You can view a list of SNMP profiles that are defined. The list includes high-level information about the contents of each profile.

If no tenant definitions have been created, the SNMP profiles are shared among all registered data sources. The global administrator sees a list of SNMP profiles that are not explicitly associated with a tenant. Tenant administrators see only the items that are associated with their tenant.

Follow these steps:

1. Log in as a user with the Administrator role.
2. Select **Administration**, **Configuration Settings**, and click **SNMP Profiles**.
   The Manage SNMP Profiles page opens, and the current list of SNMP profiles appears.

Add SNMP Profiles

Administrators can create SNMP profiles in the NetOps Portal Console.

Follow these steps:

1. Log in to the NetOps Portal Console as a user with the Administrator role.
2. Display the list of SNMP profiles:
   - CAPC: Select **Administration**, **Configuration Settings: SNMP Profiles**.
   - NPC: Select **Administration**, **NetQoS Settings: SNMP Profiles**.

   The page displays the current list of SNMP profiles.
3. Click **New**.
   The Add SNMP Profile dialog opens.
4. Complete the fields and change the default settings as needed.
5. Click **Save**.
   You return to the list of SNMP profiles. The new profile appears in the list.
   NetOps Portal performs a global synchronization to send the profile information to CA Network Flow Analysis.

Verify That Data Is Received

To verify that data is received, complete the following tasks:

Verify That the Interfaces Are Enabled

Once you configure CA Network Flow Analysis to receive flow data, verify that the expected interfaces are monitored.

Follow these steps:

1. Open the NFA console, logged in with Administrator rights.
2. Open the Available Interfaces page:
   a. Select Administration from the NFA console menu.
      The Administration page opens.
   b. Select System: Enable Interfaces in the Administration menu.
      The Available Interfaces page opens.
3. Expand the router details to display the interface list: Click the arrow to the left of the router name.
4. Review the list to review whether the interfaces are enabled.
5. Change the interface status: Select the check box next to one or more interfaces, then click Enable or Disable.
   The selected interfaces are immediately enabled or disabled.
6. Repeat these steps for each router.

Verify That the Interfaces Are Visible in the NFA Console

Make sure that the configured interfaces are visible in the NFA console.

1. Verify that interfaces are visible on the Enterprise Overview page:
   a. Log in to the NFA console.
   b. Click Enterprise Overview in the main menu.
   c. Make sure the report views show interfaces.
2. Verify that interfaces are visible in the Interface Index:
   a. Click Interfaces in the NFA console menu.
      The Interface Index opens.
   b. Make sure that the Interface Index includes the interfaces that you expect to see.
      You can locate interfaces by expanding router details to view interfaces. Alternatively, you can use the Search box to find routers or interfaces by entering all or part of a name or description.
3. If interfaces are not visible, perform the following preliminary troubleshooting tasks:
   – Verify that the CA Network Flow Analysis services are running on the NFA console server.
   – Review the logs. View the logs in the NFA console or open the logs from the <install_path>\reporter\Logs directory.

Network Flow Analysis Views in NetOps Portal

You can view Network Flow Analysis data in NetOps Portal in the following ways:

Built-in CAPC Dashboards with Enterprise-Wide Data

- **Infrastructure Overview dashboard**:
  – Interfaces Over Threshold
  – Routers with the Most Flow Traffic
  – Top Enterprise Hosts by Volume
  – Top Enterprise Protocols by Volume
  – Top Flows by Interface
  – Top IP Interface Utilization (Flow)
- **Management: Management Overview dashboard**:

  - Top Flows by Interface
  - Top IP Interface Utilization (Flow)
- **Management: Network Overview dashboard**:
  - Top Enterprise Hosts by Volume
  - Top Enterprise Protocols by Volume
- **Capacity Planning: Router/Switch Capacity Watch Lists dashboard**:
  - Routers with the Most Flow Traffic

Custom CAPC Dashboard Views with Interface-Specific Data

Display interface-specific CA Network Flow Analysis data by adding the following views with an interface context type:

- Calendar Heat Chart (Flow)
- Stacked Protocol Trend
- Stacked ToS Trend
- Top Conversations (Bar)
- Top Conversations (Pie)
- Top Conversations (Table)
- Top Hosts (Bar)
- Top Hosts (Pie)
- Top Hosts (Table)
- Top Protocols (Bar)
- Top Protocols (Pie)
- Top Protocols (Table)
- ToS Summary (Pie)
- ToS Summary (Table)

For more information about customizing and adding views to a dashboard, see Manage Dashboards.

Built-In CAPC Interface Page Views with CA Network Flow Analysis Data

- **IP Performance tab**:
  - Stacked Protocol Trend
  - Top Conversations (Pie)
  - Top Hosts (Pie)
  - ToS Summary (Pie)
- **CBQoS tab**:
  - Stacked Protocol Trend
  - Stacked ToS Trend

Enterprise-Level Views

You can view enterprise-wide data from CA Network Flow Analysis in several NetOps Portal dashboard views, which are described in the following topics.

- Interfaces Over Threshold
- Routers with the Most Flow Traffic
- Top Enterprise Hosts by Volume
- Top Enterprise Protocols by Volume
- Top Flows by Volume
- Top IP Interface Utilization

The top interfaces, hosts, protocols, or ToS have the highest traffic volume during the reporting period.

Interfaces Over Threshold

The Interfaces Over Threshold view lists the most heavily used interfaces throughout the enterprise. A table summary shows the interfaces with utilization that exceeds the configured thresholds.

The Interfaces Over Threshold view shows the interfaces whose traffic exceeded the configured thresholds during the reporting period. The view includes the following information for up to ten top interfaces:

- **Status**
  Identifies the interface status as Critical (Red - Meets or exceeds the user-defined Critical threshold) or Warning (Orange - Meets or exceeds the user-defined Warning threshold).
- **Interface Name**
  Identifies the interface by its name. (Depending on the application setting for the name format, the name may be prefixed by the device name.)
- **Traffic Direction**
  Shows whether the data was inbound or outbound on the interface.
- **Speed**
  (CAPC) Records the data speed that is defined for the interface.
- **Average Utilization**
  Measures the average percentage of interface capacity that was used.
- **Percent Time Critical**
  Shows the percentage of the reporting period that the interface met or exceeded the Critical threshold.
- **Percent Time Warning**
  Shows the percentage of the reporting period that the interface met or exceeded the Warning threshold.

NetOps Portal views show the data from the time range that is defined for the page.

Contents

Opening the View

To see this view, go to one of the following locations:

- (CAPC) Infrastructure Overview dashboard; Summary context view in a custom dashboard
- (NPC) Enterprise, Traffic Analysis, Routers/Switches Overview, or custom dashboard

Available Actions

You can perform several actions in this view:

- Change the thresholds, view name, and utilization settings as described in this topic.
- (CAPC) Change the columns that are shown in the table: click near a column border, click Columns, then choose the columns to display.
- Click an interface name to open the Interface context pages. You can review details or open additional views of interface data.

Change the View Settings

Follow these steps:

1. Open the dialog for editing the view:

- (CAPC) Click the Edit icon in the view title bar, and then click Edit.
- (NPC) Click the arrow next to the title name, and select Edit from the menu.

The dialog opens.

2. (Optional) Edit the text in the Title field to change the name in the view title bar.
3. (Optional) Edit the thresholds by changing any of the following values in the Interfaces Over Threshold Settings section:
   - **Critical - % Utilization**: Specify the utilization percentage for flagging interfaces with a status of Critical, the highest level of concern. If the utilization for an interface has met or exceeded this percentage, it is marked with a red (Critical) status symbol.
   - **Warning - % Utilization**: Specify the utilization percentage for flagging interfaces with a status of Warning. If the utilization for an interface has met or exceeded this percentage, but has not met the Critical threshold, the interface is marked with an orange (Warning) status symbol.
   - **Affected % of reporting period**: Specify the percentage of the reporting period that a utilization percentage must be violated in order for the threshold to be met.
     For example, if the 'Affected % of reporting period' value is 25, the threshold is met for the interfaces that have a utilization level at or above the threshold level during 25% of the reporting period. With the default reporting period of 24 hours, the list includes interfaces at or above the threshold value for six hours or more during the previous 24 hours.
4. (Optional) (NPC) Define a new context to filter the interfaces that can appear in the view: select the Filter by value, and select a context type and setting in the Select Context dialog.
   Interfaces that are not in the selected group do not appear in the view, even if they violate a threshold. If you select a group, the defined context appears under the view title.
5. Select the scope of your changes from the **Apply Changes** drop-down.
6. Click **Save** to save your changes.

Find the Comparable View in the Network Flow Analysis (NFA) Console

The Interfaces Over Threshold view is similar to the Interface Utilization view on the Enterprise Overview page in the NFA console.

Routers with the Most Flow Traffic

The Routers with the Most Flow Traffic view displays the routers in your network that have the highest traffic. Traffic use is measured for both inbound and outbound traffic during the reporting period, as reported by CA Network Flow Analysis.

The view includes the following information for a maximum of 10 routers:

- **Name**
  Consists of the router's IP address and device name (Y-Axis). If an administrator defined an alias for the device item, the alias is displayed. Otherwise, the discovered device name is displayed.
- **Volume**
  Measures the total amount of traffic for the router expressed in megabytes, for example (X-Axis).

NetOps Portal views show the data from the time range that is defined for the page.

Opening the View

To see this view, go to the following location:

- Infrastructure Overview and Router/Switch Capacity Watch Lists dashboards; Summary-type view in a custom dashboard

Available Actions

You can perform several actions in this view:

- Change the view name by editing the view settings.
- Display details in a tooltip by hovering over a bar.
- Click a router bar to view details in the NetOps Portal Router pages.

Top Enterprise Hosts by Volume

The Top Enterprise Hosts by Volume view shows the enterprise hosts that have the highest traffic volume, as reported by CA Network Flow Analysis.

The view shows a bar for each of a maximum of 10 hosts that have the highest traffic volume. The bar chart includes the following information:

- **Host**
  Identifies the host server by its name and IP address (Y-Axis). If an administrator has defined an alias for the device, the alias is displayed. Otherwise, the discovered device name is displayed.
- **Volume**
  Measures the total amount of data sent to or from the host, expressed in a scale that is appropriate for the highest-volume host (X-Axis).

Data appears from the time range that is defined for the page.

Opening the View

To see this view, go to one of the following locations:

- (CAPC) Infrastructure Overview, Network Overview, or Summary context custom dashboard

Available Actions

You can perform several actions in this view:

- Change the view name by editing the view settings.
- Display details in a tooltip by hovering over a bar.
- Click a name or bar to open the related views in the NFA console.

Find the Comparable View in the NFA Console

The Top Enterprise Hosts by Volume view is similar to the Top Hosts view on the Enterprise Overview page in the NFA console.

Top Enterprise Protocols by Volume

The **Top Enterprise Protocols by Volume** view shows the protocols with the highest volume of network traffic across the enterprise.

The view includes the following information for a maximum of ten protocols that are associated with the highest traffic during the reporting period:

- **Protocol**
  Identifies the protocol by its keyword (Y-axis).
- **Volume**

Measures the total amount of data associated with the protocol expressed in a scale that is appropriate for the highest-volume protocol (X-axis).

Data appears from the time range that is defined for the page.

Opening the View

To see this view, go to one of the following locations:

- (CA PC) Infrastructure Overview or Network Overview dashboard; Summary context view in a custom dashboard
- (NPC) Enterprise, Traffic Analysis, Network Overview, and custom dashboards

Available Actions

You can perform several actions in this view:

- Change the view name in the view settings.
- Display details in a Tooltip by holding your cursor over a bar.
- Click a name or bar to open related views in the NFA console.

Find the Comparable View in the NFA Console

The Top Enterprise Protocols by Volume view is similar to the Top Protocols view on the Enterprise Overview page in the NFA console.

Top Flows by Volume

The Top Flows by Volume views show the interfaces across the enterprise that have the highest volume of inbound or outbound traffic.

The view shows the following information for a maximum of 10 top interfaces:

- **Name**
  Identifies the interface by its device name (such as its router name), followed by a colon (:) and the interface name (Y-Axis).
- **Volume**
  Measures the volume of flow data on the interface (X-Axis) expressed in a scale that is appropriate for the highest-volume interface.

NetOps Portal views show the data from the time range that is defined for the page.

Opening the View

To see this view, go to one of the following locations:

- (CAPC) Infrastructure Overview and Management Overview dashboards; Summary context view in a custom dashboard
- (NPC) Custom dashboard

Available Actions

You can perform several actions in this view, including the following ones:

- Change the data direction or the view name by editing the view settings.
- Display details in a tooltip by hovering over a bar. The tooltip identifies the interface position among the top 10, with Interface 0 as the one with the highest traffic volume.
- Click a name or bar to open related information on the interface context pages.

Find Flow Volume Data in the Network Flow Analysis (NFA) Console

To see the flow volume of multiple top interfaces in the NFA console, create and run a Custom report. For example, you can view flow volume for the top interfaces in summary pie charts, summary tables, trend charts, and stacked trend charts. For instructions, see the topic "Set Up Custom Reports" in the *CA Network Flow Analysis Operator Guide.*

To see the flow volume of a single top interface in the NFA console, drill into details from the Enterprise Overview page:

1. Click an interface name or bar in one of the Top Interfaces views on the NFA console Enterprise Overview page.
2. Select Flows from the list labeled "For this interface, show me" on the Interface page that opens.
3. Click the gray bar on the left side of the page to change the presentation mode.
4. Click Volume in the Presentation menu that opens.
   The Flows views display a trend chart of inbound flow volume and outbound flow volume.
   To jump to the NetOps Portal Interface Pages data for the selected interface, click the arrow next to the Flows title, and select CAPC/NPC Interface Performance.

Top IP Interface Utilization

The Top IP Interface Utilization (Flow) views show the high-utilization interfaces from across the enterprise.

The view includes the following information for a maximum of 10 top interfaces during the reporting period:

- **Name**
  Identifies the interface by its device name/interface name (Y-Axis).
- **Percent (Utilization)**
  Measures the percentage of interface capacity that was used (X-Axis). The view shows the utilization of either inbound or outbound capacity.

NetOps Portal views show the data from the time range that is defined for the page.

Opening the View

To see this view, go to one of the following locations:

- (CAPC) Infrastructure Overview and Management Overview dashboards; Summary-type view in a custom dashboard
- (NPC) Traffic Analysis and custom dashboards

Available Actions

You can perform several actions in this view:

- Change the data direction, view name, and context (the interfaces that are used) by editing the view settings.
- (NPC) Change the utilization thresholds.
- Display details in a tooltip by holding your cursor over a bar.
- Click a name or bar to open related information on the Interface context pages.

Find the Comparable View in the Network Flow Analysis (NFA) Console

The Top IP Interface Utilization (Flows) view is similar to the Interface Utilization view on the Enterprise Overview page in the NFA console.

Interface Stacked Trend View

The Stacked Trend views show the top protocol or Type of Service (ToS) values that DX NetOps Performance Management uses for traffic on the currently selected interface. The following Stacked Trend views are available:

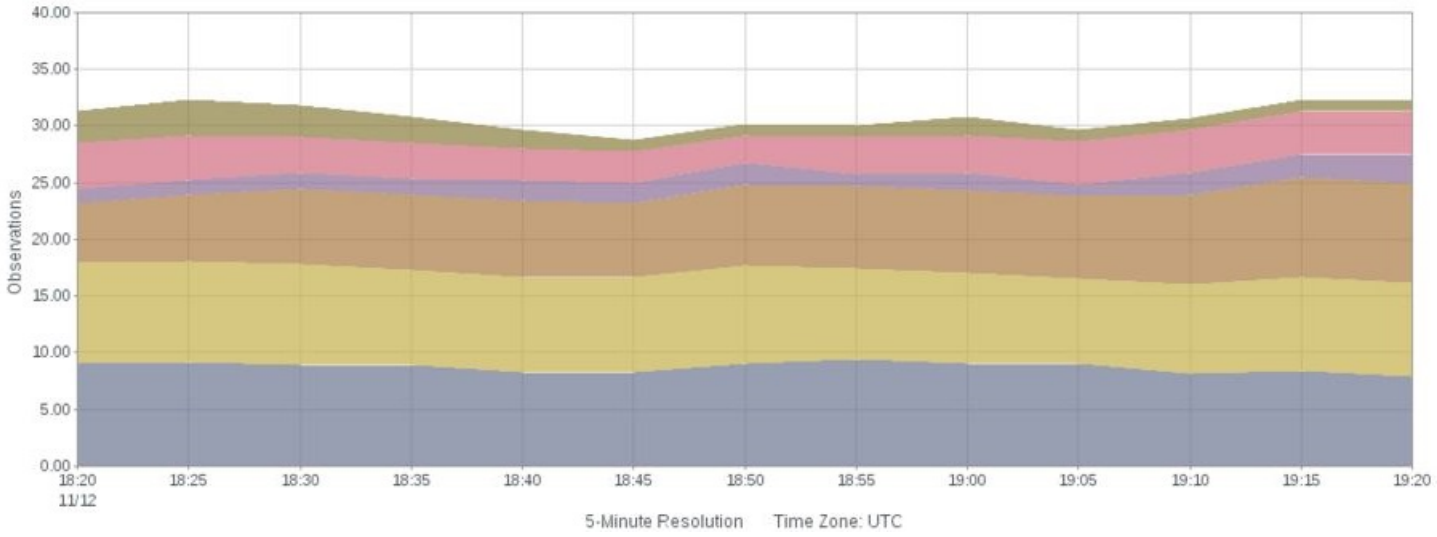
- Stacked Protocol Trend
- Stacked ToS Trend

Stacked ToS Trend

The Stacked Typy of Service (ToS) Trend views show the interface traffic for the top ToS, including the time that the traffic occurred. A timeline of rates is included for each ToS value. You can configure the view to display rate, utilization, or volume information.

The stacked chart shows the value of rate, utilization, and volume as stacked lines. The value of the first metric is measured from the X axis. The value of each of the other metrics is measured from the top of the stacked line to the previous metric. For example, if the top of the first trend line indicates 10, and the second line indicates 25, the value of the second metric is 15.

The following chart shows an example of a stacked trend chart. At 18:55, the value of the metric that is represented by the blue color is approximately ten. At the same time, the value of the metric that is represented by the yellow color is approximately seven. The value of the metric in yellow is calculated by subtracting the value at the bottom of the yellow trend line from the value at the top.



The view includes the following information:

- **Identifier**
  Identifies the interface that is used for the report. The identifier string consists of the router name, interface name, and interface description (under the view title).
  (NPC) The identifier line also includes the interface speed.
- **ToS Bands**
  Show the data rate, data volume, or interface capacity utilization for each top ToS that is associated with traffic on the interface.
- **Time**
  Point in time during data transmission, expressed in hours and minutes (X-Axis).
- **Measurement Setting:**

  - **Rate:** Data transfer rate at each point in time, expressed in kilobits per second or a rate that is appropriate for the highest-volume ToS (Y-Axis). The rate is calculated by dividing the data volume by the elapsed transmission time.
  - **Bytes** (Volume): Data volume at each point in time, expressed in a scale that is appropriate for the highest-volume ToS (Y-Axis).
  - **Percent** (Utilization): Percentage of the total interface capacity that the ToS traffic uses (Y-Axis). The utilization percentage is calculated by dividing the data rate by the data speed.

Depending on the data direction, the view shows inbound, outbound, or total data on the interface:

- **Legend**
  Identifies the ToS for each color band by ToS number and label (bottom of the view).

NetOps Portal views show the data from the time range that is defined for the page.

In this article:

Open the View

To see this view in console, go to one of the following locations:

- (CAPC) Custom dashboard; Interface Pages (with an interface selected): CBQoS tab
- (NPC) Interface Pages (with an interface selected): Interface QoS and custom tabs

> **NOTE**
> You can add Multi-Interface Stacked ToS Trend views to a custom dashboard or to a custom tab in the Interface pages of the console. This view consists of a group of interface-specific stacked ToS trend charts.

Available Actions

You can perform several actions in this view, including:

- Change the traffic direction (In, Out, or Total), the type of measurement (Rate, Volume, or Utilization), and the view name by editing the view settings. If the view is on a custom interface context dashboard in the console, you can change the interface.
- (CAPC) Zoom in to narrow the time frame.
- (CAPC) Display only the data for a single ToS: Right-click a ToS in the legend at the bottom of the view, and then click Focus. This menu is available for a view that has multiple ToS values. (This option is active when the view contains multiple ToS values.)
- (CAPC) Hide data for one of multiple ToS values: Right-click a ToS in the legend at the bottom of the view, and then click Hide.
- (CAPC) Position your cursor over legend items to display explanatory Tooltips.
- Jump to details on an NFA console Interface page by double-clicking a ToS value in the legend.

Find ToS Trend Data in the NFA Console

You can display ToS volume in trend charts or stacked trend charts in the NFA console for a selected interface:

- **Overview**
  - **Report type:** Overview
  - **Presentation menu options:** Mixed Chart or Mixed Trend; Volume.
  - **Views:** Stacked ToS Trend (In and Out) for the Top N ToS, plus other overview views.
- **Top N ToS, Stacked Trends**

  - **Report type:** ToS
  - **Filter:** Top N ToS
  - **Presentation menu options:** Stacked Trend Chart; Volume.
  - **Views:** Stacked Trend for the Top N ToS (In, Out, and Total).
- **Top N ToS, Trends**
  - **Report type:** ToS
  - **Filter:** Top N ToS
  - **Presentation menu options:** Trend Chart; Volume
  - **Views:** Trend (In, Out, and Total) for each of the Top N ToS
- **Single ToS, Stacked Trends/Trends**
  - **Report type:** ToS
  - **Filter:** Single ToS value
  - **Presentation menu options:** Mixed Trend; Volume
  - **Views:** Trend (In and Out with baselines), Stacked ToS Trend (In and Out).
- **Single ToS, Trends**
  - **Report type:** ToS
  - **Filter:** Single ToS value. Presentation menu options: Mixed Chart; Volume.
  - **Views:** Stacked ToS Trend (In and Out).
- **Conversation**
  - **Report type:** Conversations
  - **Filter:** Single conversation source and destination.
  - **Report subtype:** Protocols
  - **Presentation menu options:** Volume.
  - **Views:** Conversation Trend (maximum of seven views for different timespans).

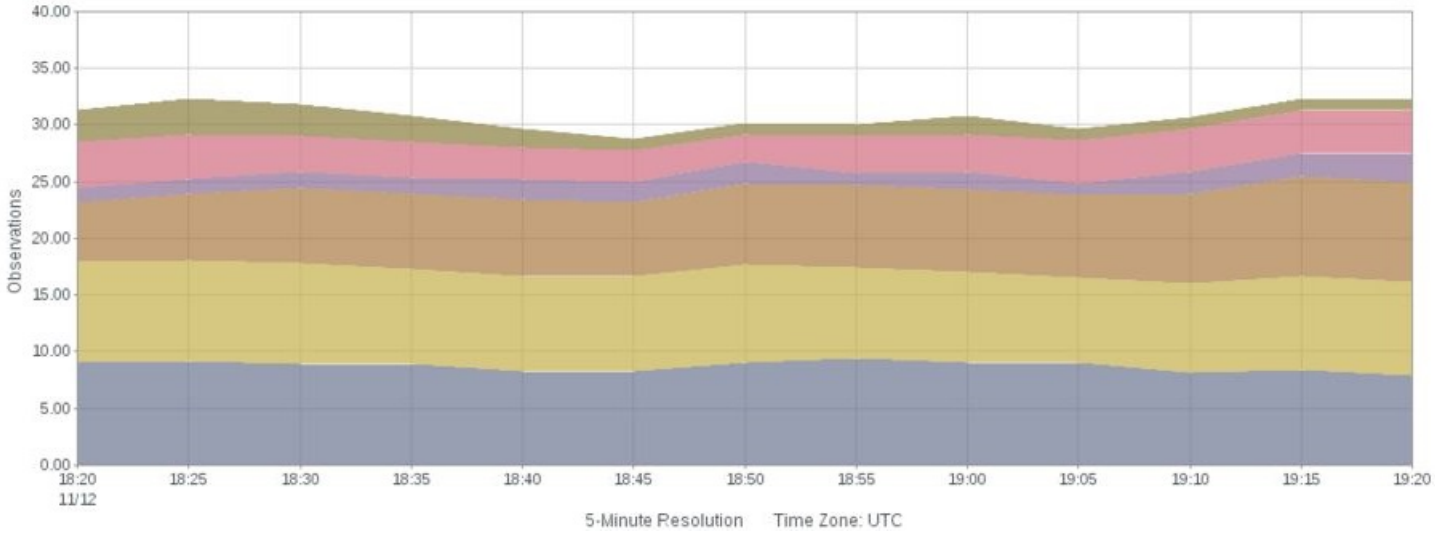
To display Flow Forensics-level detail, click the Flow Forensics link and run a Flow Forensics report.

Stacked Protocol Trend

The Stacked Protocol Trend views show the protocols that are used the most heavily for traffic on the selected interface. The views also show when the traffic occurred. A timeline of rates is included for each listed ToS value. You can configure the view to display rate, utilization, or volume information.

The stacked chart shows the value of rate, utilization, and volume as stacked lines. The value of the first metric is measured from the X axis. The value of each of the other metrics is measured from the top of the stacked line to the previous metric. For example, if the top of the first trend line indicates 10, and the second line indicates 25, the value of the second metric is 15.

The following chart shows an example of a stacked trend chart. At 18:55, the value of the metric that is represented by the blue color is approximately ten. At the same time, the value of the metric that is represented by the yellow color is approximately seven. The value of the metric in yellow is calculated by subtracting the value at the bottom of the yellow trend line from the value at the top.



The views include the following information:

- **Identifier**
  Identifies the interface that is used for the report. The identifier string consists of the router name, interface name, and interface description (under the view title).
  (NPC) The identifier line also includes the interface speed.
- **Protocol Bands**
  Show the data rate, the data volume, or the interface capacity utilization for each top protocol that is associated with traffic on the interface.
- **Time (All Views)**
  Point in time during data transmission--expressed in hours and minutes (X-Axis).
- **Measurement Setting:**
  - **Rate:** Data rate at each point in time. For example, the rate can be expressed in kilobits per second (Y-Axis). The rate is calculated by dividing the data volume by the elapsed transmission time.
  - **Bytes** (Volume): Data volume at each point in time. For example, the volume can be expressed in kilobytes, (Y-Axis).
  - **Percent** (Utilization): Percentage of the total interface capacity that the protocol uses (Y-Axis). The utilization percentage is calculated by dividing the data rate by the data speed.

Depending on the data direction, the view shows inbound, outbound, or total data on the interface.

- **Legend**
  Identifies the protocol for each color band by protocol keyword and tcp/udp port (bottom of the view).

NetOps Portal views show the data from the time range that is defined for the page.

Contents

Opening the Views

To see these views in the console, go to one of the following locations:

- (CAPC) Interface Pages (with an interface selected): Custom dashboard, IP Performance, and CBQoS tabs
- (NPC) Interface Pages (with an interface selected): Interface Capacity, Interface QoS, and custom tabs

> **NOTE**
> You can add Multi-Interface Stacked Protocol Trend views to a custom dashboard or to a custom tab in the Interface pages in the console. This view consists of a group of interface-specific stacked protocol trend charts

Available Actions

You can perform several actions in this view, including the following ones:

- Change the traffic direction, the type of measurement (Rate, Volume, or Utilization), and the view name by editing the view settings. If the view is on a custom interface context dashboard in the console, you can change the interface.
- (CAPC) Zoom in to narrow the time frame.
- (CAPC) Display only the data for a single protocol: Right-click a protocol in the legend at the bottom of the view, and click Focus. This menu is available for a view that has multiple protocols. (This option is active when the legend contains multiple protocols.)
- (CAPC) Hide data for one of multiple protocols: Right-click a protocol in the legend at the bottom of the view and, click Hide.
- (CAPC) Position your cursor over legend items to display explanatory Tooltips.
- Jump to details on an NFA console Interface page by double-clicking a protocol in the legend.
- (NPC) Jump to details on the corresponding Interface page by double-clicking a protocol band in the view. To select a destination tab on the Interface page, right-click the protocol band and select a tab from the menu.

Find Protocol Trend Data in the NFA Console

You can display protocol volume in the NFA console in trend charts or stacked trend charts for a selected interface:

- **Overview**
  - Report type: Overview
  - Presentation menu options: Mixed Chart; Volume
  - Views: Stacked Protocol Trend (In and Out) for the Top N Protocols, plus other overview views.
- **Top N Protocols, Stacked Trends**
  - Report type: Protocols.
  - Filter: Top N Protocols.
  - Presentation menu options: Stacked Trend Chart; Volume.
  - Views: Stacked Trend for the Top N Protocols (In, Out, and Total).
- **Top N Protocols, Trends**
  - Report type: Protocols. Filter: Top N ToS. Presentation menu options: Trend Chart; Volume. Views: Trend (In, Out, and Total) for each of the Top N Protocols.
- **Single Protocol**
  - Report type: Protocols.
  - Filter: Single protocol.
  - Views: (Depending on the selected report subtype): trends, stacked trends, trend summaries, and multi-trend summaries for protocols, protocol hosts, and protocols in conversations.

To display Flow Forensics-level detail, click the Flow Forensics link, and run a Flow Forensics report.

Interface ToS Summaries

The ToS Summary views show the Type of Service (ToS) values for traffic on the selected interface. The views are described in the following topics:

ToS Summary (Pie)

The ToS Summary (Pie) view shows an overview of the Type of Service (ToS) values for traffic on the selected interface.

The view includes a pie chart and table of information about the high-volume ToS values in use on the selected interface. The table includes the following information by default:

- **Identifier**
  Identifies the interface that is used for the report. The identifier string consists of the router name, interface name, and interface description (under the view title).
  (NPC) The identifier line also includes the interface speed.
- **Type of Service**
  Name of the ToS values associated with high-volume traffic, identified by number and label.
- **Total**
  Shows the total data volume for the reporting period.
- **Percent**
  (CAPC) Lists the percentage of the total data volume for the Top N ToS.

NetOps Portal views show the data from the time range that is defined for the page.

Opening the Views

To see these views in the NetOps Portal Console, go to one of the following locations:

- (CAPC) Interface Pages (with an interface selected): IP Performance tab; Custom dashboard
- (NPC) Interface Pages (with an interface selected): Custom tab

Available Actions

You can perform several actions in this view, including the following ones:

- Change the traffic direction and view name by editing the view settings. If the view is on a custom interface context dashboard in the console, you can change the interface.
- (CAPC) Change the type of measurement.
- Jump to details on an NFA console Interface page by double-clicking a ToS name.

Find ToS Summary Pie Charts in the NFA Console

You can display pie charts of ToS summary data in the NFA console for a selected interface:

- **Overview**
  - **Report type:** Overview
  - **Presentation menu option:** Pie Chart
  - **View:** ToS Summary (In and Out) for the Top N ToS
- **Top N ToS Summary**
  - **Report type:** ToS
  - **Filter:** Top N ToS
  - **Presentation menu option:** Pie Chart
  - **View:** ToS Summary (In, Out, and Total) for the Top N ToS
- **Single ToS Summaries**
  - **Report type:** ToS
  - **Filter:** Single ToS
  - **Report subtype:** Overview
  - **Presentation menu option:** Pie Chart
  - **Views:** ToS Protocol Summary (In and Out) for the single ToS; ToS Hosts Summary (From and To) for the single ToS; ToS Conversations Summary (Total) for the single ToS.

    NOTE
    You can view additional versions of the summary pie charts by selecting Protocols, Hosts, or Conversations as the report subtype.

To display Flow Forensics-level detail, click the Flow Forensics link and run a Flow Forensics report.

ToS Summary (Table)

The ToS Summary (Table) views show rate, volume, or utilization for the top ToS values of the traffic on a particular interface. You can use this information to compare traffic for each of the top ToS values.

The example graphic shows the view in the console. The table shows the rate for each listed ToS value. You can configure the view to display rate, utilization, or volume information.

An interface identification string is shown under the view title. The table contains a row for each ToS with the Type of Service identifier (EF/AF, DSCP, and ToS values) and the following rate, volume, or utilization information:

- **Rate:**
  - (CAPC/NPC) Average rate of total, inbound, and outbound data for each ToS (Average Total. Average Out, and Average In)
  - (CAPC) Maximum rate of data that is outbound or inbound on the interface for each ToS (Maximum Out and Maximum In)

  The rate is calculated by dividing the data volume by the elapsed transmission time.
- **Volume:** Volume of outbound, inbound, and all data for each ToS (Out, In, and Total), expressed in a scale that is appropriate for the highest-volume ToS.
- Utilization:
  - (CA PC/NPC) Average utilization of total, outbound, and inbound data that each ToS consumes (Average Total. Average In, and Average Out)
  - (CAPC) Maximum percentage of interface capacity that the outbound or inbound utilizes for each ToS (Maximum Out and Maximum In)

  The utilization percentage is calculated by dividing the data rate by the data speed.

NetOps Portal views show the data from the time range that is defined for the page.

Opening the View

To see this view in the console, go to one of the following locations:

- (CA PC) Custom dashboard
- (NPC) Interface Pages (with an interface selected): Interface QoS or custom tab

Available Actions

You can perform several actions in this view:

- Change the type of measurement (Rate, Volume, or Utilization) and the view name by editing the view settings.
- (CAPC) Change the interface.
- Re-sort the table data by clicking a column heading. Click again to toggle between descending and ascending order.
- Change the Max Per Page value to show more or fewer items on each table page.
- (CAPC) Change the columns that are shown in the table: Click near a column border, click Columns, and then choose the columns to display.
- Click a Type of Service link to display more information about the ToS on Interface report pages in the NFA console.

Find ToS Summary Tables in the NFA Console

You can display ToS summary tables in the NFA console for a selected interface:

- **Top N ToS Summary**

  - **Report type:** ToS.
  - **Filter:** Top N ToS.
  - **Presentation menu options:** Summary Table; Volume.
  - **View:** ToS Summary Table for the Top N ToS.
- **Protocol Summary for a Single ToS**
  - **Report type:** ToS.
  - **Filter:** Single ToS.
  - **Report subtype:** Protocols.
  - **Subtype filter:** Top N Protocols.
  - **Presentation menu options:** Summary Table; Volume.
  - **Views:** ToS Protocol Summary Table for the single ToS.
- **Host Summary for a Single ToS**
  - **Report type:** ToS.
  - **Filter:** Single ToS.
  - **Report subtype:** Hosts.
  - **Subtype filter:** Top N Hosts.
  - **Presentation menu options:** Summary Table; Volume.
  - **Views:** ToS Hosts Summary Table for the single ToS.
- **Conversation Summary for a Single ToS**
  - **Report type:** ToS.
  - **Filter:** Single ToS.
  - **Report subtype:** Conversations.
  - **Subtype filter:** Top N Conversations.
  - **Presentation menu options:** Summary Table; Volume.
  - **Views:** ToS Conversations Summary Table for the single ToS.

To display Flow Forensics-level detail, click the Flow Forensics link and run a Flow Forensics report.

Interface Top Conversations

The following Top Conversations views show the conversations that generate the highest traffic on the currently selected interface:

Top Conversations (Bar)

The Top Conversations (Bar) views show the conversations that have the highest traffic on the selected interface. A bar graph shows the volume for each conversation.

For example, use conversation information to determine the IP addresses of high-volume hosts. Contact the host owners or users to investigate the nature and purpose of the traffic.

You can view the conversations for incoming data, outgoing data, or all data.

The view includes a bar for each top conversation on the selected interface. A maximum of 10 conversations are shown. The view includes the following information:

- **Identifier**
  Identifies the interface that is used for the report. The identifier string consists of the router name, interface name, and interface description (under the view title).
  (NPC) The identifier line also includes the interface speed.
- **Conversation Pair**

Identifies the conversation source and destination servers by their names (the fully qualified DNS names, if they are available), followed by the IP addresses (Y-Axis).

- **Volume**
  Measures the total amount of data that was exchanged in the conversation expressed in a scale that is appropriate for the highest-volume conversation (X-Axis).

NetOps Portal views show the data from the time range that is defined for the page.

Opening the View

To see this view in the NetOps Portal Console, go to one of the following locations:

- (CAPC) Custom dashboard
- (NPC) Interface Pages (with an interface selected): Interface Capacity or custom tab

Available Actions

You can perform several actions in this view:

- Change the view name by editing the view settings.
- (CAPC) Change the traffic direction and the interface.
- Display details in a Tooltip by holding your cursor over a bar.
- Jump to conversation details on an NFA console Interface report page by clicking a bar or name.

Find Conversation Data in the NFA Console

You can view conversation volume trend charts in the NFA console for any interface that you have selected:

- **Overview Multi-Trend**
  - **Report type:** Overview.
  - **Presentation menu options:** Mixed Trend; Volume.
  - **View:** Conversations Multi Trend Summary (Total) for the Top N Conversations, plus other views.
- **Top N Conversations Trend**
  - **Report type:** Conversations.
  - **Filter:** Top N Conversations.
  - **Presentation menu options:** Trend Chart; Volume.
  - **View:** Conversations Trend for the Top N Conversations.
- **Conversations for a Single Protocol**
  - **Report type:** Protocols.
  - **Filter:** Single protocol.
  - **Report subtype:** Conversations.
  - **Conversation Filter:** Top N Conversations.
  - **Presentation menu option:** Trend Chart.
  - **View:** Protocol Conversations Summary (Total) for a single protocol.
- **Conversations for a Single ToS**
  - **Report type:** ToS.
  - **Filter:** Single ToS.
  - **Report subtype:** Conversations.
  - **Conversation Filter:** Top N Conversations.
  - **Presentation menu options:** Trend Chart; Volume.
  - **View:** ToS Trend view for each conversation that uses the single ToS.

NOTE
To see trend charts for a single conversation, click Top N Conversations and select a single conversation as the filter.

To display Flow Forensics-level detail, click the Flow Forensics link and run a Flow Forensics report.

Top Conversations (Pie)

The Top Conversations (Pie) view includes a pie chart of the conversations that account for the most traffic on the selected interface.

The view includes a pie chart and a table of information about the high-volume conversations on the selected interface. A text string near the top of the view identifies the interface whose data is displayed. The table includes the following information by default:

- **Identifier**
  Identifies the interface that is used for the report. The identifier string consists of the router name, interface name, and interface description (under the view title).
  (NPC) The identifier line also includes the interface speed.
- **Source - Destination Name**
  Identifies the host servers that initiated and received the conversation data by their fully qualified DNS names (if available) and IP addresses.
- **Total**
  Shows the total amount of data in the conversation expressed in a scale that is appropriate for the conversation with the highest volume.
- **Percent**
  (CAPC) Records how much the conversation consumes out of the total traffic that is displayed.

NetOps Portal views show the data from the time range that is defined for the page.

Opening the Views

To see these views in the NetOps Portal Console, go to one of the following locations:

- (CAPC) Custom dashboard; Interface Pages (with an interface selected): IP Performance tab
- (NPC) Interface Pages (with an interface selected): Interface QoS or custom tab

Available Actions

You can perform several actions in this view, including the following ones:

- Change the view name by editing the view settings.
- (CAPC) Change the traffic direction and the type of measurement. If the view is on a custom interface context dashboard in the CA NetOps Portal Console, you can change the interface.
- (CAPC) Change the columns that are shown in the table: Click near a column border, click Columns, and then choose the columns to display.
- Jump to details on an NFA console Interface page by clicking a link.

Find Conversation Pie Charts in the NFA Console

You can display conversation pie charts in the NFA console for any interface that you have selected:

- **Overview**

  - **Report type:** Overview
  - **Presentation menu option:** Pie Chart
  - **View:** Conversations Summary (Total) for the Top N Conversations, plus other overview views
- **Top N Conversations**
  - **Report type:** Conversations
  - **Filter:** Top N Conversations
  - **Presentation menu option:** Pie Chart.
  - **View:** Conversations Summary (Total) for the Top N Conversations
- **Conversations for a Single Protocol**
  - **Report type:** Protocols
  - **Filter:** Single protocol
  - **Report subtype:** Conversations or Overview
  - **Conversations Filter:** Top N Conversations. Presentation menu option: Pie Chart or Mixed Chart
  - **View:** Protocol Conversations Summary (Total) for a single protocol.
- **Conversations for a Single ToS**
- **Report type:** ToS
- **Filter:** Single ToS
- **Report subtype:** Conversations
- **Conversation Filter:** Top N Conversations
- **Presentation menu option:** Pie Chart
- **View:** ToS Conversations Summary (Total) for a single ToS

To display Flow Forensics-level detail, click the Flow Forensics link and run a Flow Forensics report.

Top Conversations (Table)

The Top Conversations (Table) views show data for the top highest volume conversations on a particular interface. The maximum number of top conversations that are shown is 10. You can configure the view to show rate, utilization, or volume information.

The table contains a row for each conversation with the source and destination. Each conversation contains the fully qualified DNS host name (if available) and IP address of the servers that initiated and received the conversation data. The table also contains the following rate, volume, or utilization information:

- **Rate:**
  - (CAPC/NPC) For each conversation, the average rate of total data (Average Total), data that goes to the destination host (Average To), and data that comes from the source host (Average From).
  - (CAPC) For each conversation, the maximum rate of data that comes from the source host (Maximum From) and goes to the destination host (Maximum To).

  The rate is calculated by dividing the data volume by the elapsed transmission time.
- **Volume:** For each conversation, the total amount of data (Total), data that comes from the source host (From), and data that goes to the destination host (To), expressed in a scale that is appropriate for the highest-volume conversation.
- **Utilization:**
  - (CAPC/NPC) For each conversation, the average utilization by data that comes from the source host (Average From), data that goes to the destination host (Average To), and total data (Average Total).
  - (CA PC) For each conversation, the maximum percentage of interface capacity that is used by the data that comes from the source host (Maximum From) or that goes to the destination host (Maximum To).

  The utilization percentage is calculated by dividing the data rate by the data speed.

NetOps Portal views show the data from the time range that is defined for the page.

Opening the View

To see this view in the NetOps Portal Console, go to one of the following locations:

- (CAPC) Custom dashboard
- (NPC) Interface Pages (with an interface selected): Custom tab

Available Actions

You can perform several actions in this view:

- Change the type of measurement (Rate, Volume, or Utilization) and the view name by editing the view settings.
- (CAPC) Change the interface.
- (CAPC) Change the columns that are shown in the table: Click near a column border, click Columns, and then choose the columns to display.
- Click one of the links to jump to a pre-filtered Interface page report in the NFA console.

Find Conversation Tables in the NFA Console

You can display tables with conversation volumes in the NFA console for a selected interface:

- **Top N Conversations**
  - **Report type:** Conversations.
  - **Filter:** Top N Conversations.
  - **Presentation menu option:** Summary Table; Volume.
  - **View:** Conversation Summary Table for the Top N Conversations.
- **Conversations for a Single Protocol**
  - **Report type:** Protocols.
  - **Filter:** Single protocol.
  - **Report subtype:** Conversations.
  - **Subtype filter:** Top N Conversations.
  - **Presentation menu option:** Summary Table; Volume.
  - **View:** Protocol Conversation Summary Table for a single protocol.
- **Conversations for a Single ToS**
  - **Report type:** ToS.
  - **Filter:** Single ToS.
  - **Report subtype:** Conversations.
  - **Subtype filter:** Top N Conversations.
  - **Presentation menu options:** Summary Table; Volume.
  - **View:** ToS Conversations Summary Table for a single ToS.

To display Flow Forensics-level detail, click the Flow Forensics link and run a Flow Forensics report.

Interface Top Hosts

The Top Hosts views show the hosts that generate the highest traffic on the currently selected interface. The views are described in the following topics:

Top Hosts (Bar)

The Top Hosts (Bar) views show the top high-volume hosts for a particular interface. You can use this view to determine the IP addresses of hosts that are responsible for high volumes of network traffic. You can then contact the owner or user of each host to investigate the nature and purpose of the traffic. You can view the hosts for incoming flows, outgoing flows, or all flows.

The bar chart includes the following information for a maximum of 10 hosts:

- **Identifier**
  Identifies the interface that is used for the report. The identifier string consists of the router name, interface name, and interface description (under the view title).
- **Host Name**
  Identifies the host server by its fully qualified DNS name (if available) and IP address (Y-Axis).
- **Volume**
  Measures the total amount of data for the host on the interface, expressed in a scale that is appropriate for the highest-volume host (X-Axis).

NetOps Portal views show the data from the time range that is defined for the page.

Opening the View

To see this view in the console, go to one of the following locations:

- (CA PC) Custom dashboard

Available Actions

You can perform several actions in this view:

- Change the traffic direction and the view name by editing the view settings.
- (CAPC) Change the interface.
- Display details in a tooltip by holding your cursor over a bar.
- Jump to details for a specific host on an NFA console Interface page by clicking a bar.

Find Host Trend Views in the NFA Console

The Enterprise Overview page in the NFA console shows traffic volume for the top hosts in a bar chart.

You also can display host volume in trend charts for a selected interface:

- **Overview**
  - **Report type:** Overview.
  - **Presentation menu options:** Mixed Trend; Volume.
  - **View:** Hosts Multi Trend Summary (From and To) for the Top N Hosts, plus other overview views.
- **Top N Host Trend**
  - **Report type:** Hosts.
  - **Filter:** Top N Hosts.
  - **Presentation menu options:** Trend Chart; Volume.
  - **View:** Host Trend for each of the Top N Hosts.

    NOTE

    To see trend charts for a single host, click Top N Hosts and select a host as the filter.

To display Flow Forensics-level detail, click the Flow Forensics link and run a Flow Forensics report.

Top Hosts (Pie)

The Top Hosts (Pie) views show the hosts that account for the highest volumes of network traffic on the selected interface.

The table includes the following information by default:

- **Identifier**
  Identifies the interface that is used for the report. The identifier string consists of the router name, interface name, and interface description (under the view title).
- **Host Name**
  Identifies the host server by its fully qualified DNS name (if available) and IP address.
- **Total**
  Records the total amount of data for the host on the interface, expressed in a scale that is appropriate for the host with the highest volume.
- **Percent**
  (CAPC) Records the amount of traffic the host consumes out of the total traffic that is displayed.

NetOps Portal views show the data from the time range that is defined for the page.

Opening the Views

To see these views in the NetOps Portal Console, go to one of the following locations:

- (CAPC) Custom dashboard; Interface Pages (with an interface selected): IP Performance tab

Available Actions

You can perform several actions in this view, including the following ones:

- Change the traffic direction and the view name by editing the view settings. If the view is on a custom interface context dashboard in the console, you can change the interface.
- (CAPC) Change the columns that are shown in the table: Click near a column border, click Columns, and then choose the columns to display.
- Jump to details on an Interface report page in the NFA console by clicking a host link in the view.

Find Host Pie Charts in the NFA Console

You can display pie charts with host volumes in the NFA console for a selected interface:

- **Overview**
  - **Report type:** Overview
  - **Presentation menu option:** Pie Chart
  - **View:** Host Summary (From and To) for the Top N Hosts, plus other overview views.
- **Top N Hosts Summary**
  - **Report type:** Hosts
  - **Filter:** Top N Hosts
  - **Presentation menu option:** Pie Chart
  - **View:** Host Summary (From, To, and Total) for the Top N Hosts
- **Hosts for a Single Protocol**

  - **Report type:** Protocols.
  - **Filter:** Single protocol.
  - **Report subtype:** Overview.
  - **Presentation menu option:** Pie Chart.
  - **View:** Protocol Hosts Summary (From, To, and Total) for the single protocol.

To display Flow Forensics-level detail, click the Flow Forensics link and run a Flow Forensics report.

Top Hosts (Table)

The Top Hosts (Table) views show rate, volume, or utilization for the hosts that exchange the highest volume of data on a particular interface. The view is configured to show the rate for each listed host. You can configure the view to display rate, utilization, or volume information.

The view contains an interface identification string and a table. The table contains a row for each host with the fully qualified DNS host name (if available) and IP address, as well as the following rate, volume, or utilization information (by default):

- **Rate:**
  - (CAPC/NPC) For each host, the average rate of total data (Average Total), data that goes to the host (Average To), and data that comes from the host (Average From).
  - (CAPC) For each host, the maximum rate of data that comes from the host (Maximum From) and goes to the host (Maximum To).

  The rate is calculated by dividing the data volume by the elapsed transmission time.
- **Volume:** For each host, the total amount of data (Total), data from the host (From), and data to the host (To), expressed in a scale that is appropriate for the highest-volume host.
- **Utilization:**
  - (CAPC/NPC) For each host, the average utilization by data that comes from the host (Average From), data that goes to the host (Average To), and total data (Average Total).
  - (CAPC) For each host, the maximum percentage of interface capacity that is used by the data from the host (Maximum From) or that goes to the host (Maximum To).

  The utilization percentage is calculated by dividing the data rate by the data speed.

NetOps Portal views show the data from the time range that is defined for the page.

Opening the View

To see this view in the NetOps Portal Console, go to one of the following locations:

- (CAPC) Custom dashboard
- (NPC) Interface Pages (with an interface selected): Custom tab

Available Actions

You can perform several actions in this view:

- Change the type of measurement and the view name by editing the view settings.
- (CAPC) Change the interface.
- (CAPC) Change the columns that are shown in the table: Click near a column border, click Columns, then choose the columns to display.
- Click a name to jump to a pre-filtered Interface page report in the NFA console.

Find Host Tables in the NFA Console

You can display tables with host volumes in the NFA console for a selected interface:

- **Top N Hosts Summary**
  - **Report type:** Hosts.
  - **Filter:** Top N Hosts.
  - **Presentation menu option:** Summary Table; Volume.
  - **View:** Host Summary Table for the Top N Hosts.
- **Hosts for a Single Protocol**
  - **Report type:** Protocols.
  - **Filter:** Single protocol.
  - **Report subtype:** Overview.
  - **Presentation menu option:** Summary Table; Volume.
  - **View:** Protocol Host Summary Table for the single protocol.
- **Hosts for a Single ToS**
  - **Report type:** ToS.
  - **Filter:** Single ToS.
  - **Report subtype:** Hosts.
  - **Subtype filter:** Top N Hosts.
  - **Presentation menu options:** Summary Table; Volume.
  - **View:** ToS Hosts Summary Table for the single ToS.

To display Flow Forensics-level detail, click the Flow Forensics link and run a Flow Forensics report.

Interface Top Protocols

The Top Protocols views show the protocols that are associated with the highest traffic on the interface that you have selected. The views are described in the following topics:

Top Protocols (Bar)

The Top Protocols (Bar) views show the top high-volume IP protocols for traffic on a particular interface. A bar chart shows which protocols account for the most traffic on the selected interface.

This view gives you an overall picture of the amount of data that is associated with particular protocols--and, therefore, with applications--on the interface. The view also lets you determine whether the application protocols are related to business-critical processes, or are related to low-priority or non-business related processes such as unauthorized web use. You can view protocol traffic for incoming flows, outgoing flows, or all flows.

The bar chart includes the following information for a maximum of 10 protocols:

- **Identifier**
  Identifies the interface that is used for the report. The identifier string consists of the router name, interface name, and interface description (under the view title).
  (NPC) The identifier line also includes the interface speed.
- **Protocol**
  Identifies the protocol by its descriptor (Y-Axis).
- **Volume**
  Measures the total amount of protocol data expressed in a scale that is appropriate for the highest-volume protocol (X-Axis).

NetOps Portal views show the data from the time range that is defined for the page.

Opening the View

To see this view in the console, go to one of the following locations:

- (CAPC) Custom dashboard
- (NPC) Interface Pages (with an interface selected): Custom tab

Available Actions

You can perform several actions in this view, including the following ones:

- Change the traffic direction and the view name by editing the view settings.
- (CAPC) Change the interface.
- Display details in a Tooltip by holding your cursor over a bar.
- Jump to details for a specific protocol on an NFA console Interface page by clicking a bar or name.

Find the Comparable View in the NFA Console

The Top Protocols bar charts in the console are similar to the Top Protocol view on the Enterprise Overview page in the NFA console.

Top Protocols (Pie)

The Top Protocols (Pie) views show the protocols that are associated with the highest traffic volumes on the selected interface.

The table includes the following information by default:

- **Identifier**
  Identifies the interface that is used for the report. The identifier string consists of the router name, interface name, and interface description (under the view title).
  (NPC) The identifier line also includes the interface speed.
- **Protocol Name**
  Identifies the protocol by its keyword and TCP/UDP port assignment.
- **Total**
  Records the total volume of network traffic on the interface that is associated with the protocol
- **Percent**
  (CAPC) Records how much the protocol consumes out of the total traffic that is displayed.

NetOps Portal views show the data from the time range that is defined for the page.

Opening the Views

To see these views in the console, go to one of the following locations:

- (CAPC) Custom dashboard
- (NPC) Interface Pages (with an interface selected): Interface QoS or custom tab

Available Actions

You can perform several actions in this view:

- Change the traffic direction and the view name by editing the view settings.
- (CAPC) Change the interface.
- (CAPC) Change the columns that are shown in the table: Click near a column border, click Columns, and then choose the columns to display.
- Jump to details on an NFA console Interface page by clicking a protocol name.

Find Protocol Pie Charts in the NFA Console

You can display pie charts with protocol traffic volumes in the NFA console for a selected interface:

- **Overview**
  - **Report type:** Overview.
  - **Presentation menu option:** Pie Chart.
  - **View:** Protocol Summary (In and Out) for the Top N Protocols, plus other overview views.
- **Top N Protocol Summaries**
  - **Report type:** Protocols.
  - **Filter:** Top N Protocols.
  - **Presentation menu option:** Pie Chart.
  - **View:** Protocol Summary (In, Out, and Total) for the Top N Protocols.
- **Hosts or Conversations for Single Protocol**
  - **Report type:** Protocols.
  - **Filter:** Single protocol.
  - **Presentation menu option:** Pie Chart.
  - **Views:** Protocol Hosts Summary (From and To) for the single protocol; Protocol Conversations Summary (Total) for the single protocol.

To display Flow Forensics-level detail, click the Flow Forensics link and run a Flow Forensics report.

Top Protocols (Table)

The Top Protocols (Table) views show the rate, volume, or utilization for the protocol traffic with the highest volume on a particular interface. For example, you can use this information to compare the data volume or utilization for particular protocols.

The view is set to show the traffic volume for each listed protocol. You can configure the view to display rate, utilization, or volume information.

The view contains an interface identification string and table. The table has a row for each protocol with the protocol name (keyword and TCP/UDP port assignment) and the following rate, volume, or utilization information (by default):

- **Rate:**
  - (CAPC/NPC) Average rate of total (Average Total), inbound (Average In), and outbound data (Average Out) for each protocol.
  - (CAPC) Maximum rate of data that is outbound, inbound, or both outbound and inbound (Maximum Out, Maximum In, and Maximum Total) on the interface for each protocol

  The rate is calculated by dividing the data volume by the elapsed transmission time.
- **Volume:** Number of bytes/megabytes of outbound data (Out), inbound data (To), and all data (Total) for each protocol.
- **Utilization:**
  - (CAPC/NPC) Average utilization by inbound data (Average In), outbound data (Average Out), and total data (Average Total) for each protocol.
  - (CAPC) Maximum percentage of interface capacity that the outbound (Maximum Out) or inbound protocol data utilizes (Maximum In)

The utilization percentage is calculated by dividing the data rate by the data speed.

NetOps Portal views show the data from the time range that is defined for the page.

Opening the View

To see this view in the NetOps Portal Console, go to one of the following locations:

- (CAPC) Custom dashboard
- (NPC) Interface Pages (with an interface selected): Custom tab

Available Actions

You can perform several actions in this view:

- Change the type of measurement (Rate, Volume, or Utilization) and the view name by editing the view settings.
- (CAPC) Change the interface.
- Re-sort the table data by clicking a column heading. Click again to toggle between descending and ascending order.
- Change the Max Per Page value to show more or fewer items on each table page.
- (CAPC) Change the columns that are shown in the table: Click near a column border, click Columns, and then choose the columns to display.
- Click a name to jump to a pre-filtered Interfaces report in CA Network Flow Analysis.

Find Protocol Tables in the NFA Console

You can use these ways to display tables of protocol volume data in the NFA console for a selected interface:

- **Top N Protocols**
  - **Report type:** Protocols.
  - **Filter:** Top N Protocols.
  - **Presentation menu options:** Summary Table; Volume.
  - **View:** Protocol Summary Table for the Top N Protocols, plus other overview views.
- **Protocols for a Single Host**
  - **Report type:** Hosts.
  - **Filter:** Single host.
  - **Report subtype:** Protocols.
  - **Presentation menu options:** Summary Table; Volume.
  - **View:** Host Protocol Summary Table for the single host.
- **Protocols for a Single Conversation**
  - **Report type:** Conversations.
  - **Filter:** Single conversation.
  - **Report subtype:** Protocols.
  - **Presentation menu options:** Summary Table; Volume.
  - **View:** Conversation Protocol Summary Table for the single conversation.

To display Flow Forensics-level detail, click the Flow Forensics link and run a Flow Forensics report.

Anomaly Detector Dashboard

The Anomaly Detector dashboard provides data from the Anomaly Detector component for Network Flow Analysis. The Anomaly Detector monitors baselines and analyzes the flow patterns that can indicate misconfiguration, malicious attacks, or poor application delivery. For more information, see the Network Flow Analysis documentation.

By default, this dashboard contains the following views:

Anomaly Activity

The Anomaly Activity view shows the number of anomalies and anomaly clusters that occurred in relation to all the processed records.

The number of records is shown in the hundreds of thousands (Y-Axis) over a 24-hour period. Activity is shown along the X-Axis each time that the program runs (usually at 15-minute intervals).

You can edit the following view settings:

- Time frame for all views on the page, as described in Set a Custom Time Frame.
  By changing the time frame for the page, you can discover when the issue began, and look for patterns.
- View title and context.

Anomaly Detector Overall Status

The Anomaly Detector Overall Status table shows the number of records that were processed during the selected time frame, the number of anomalies, and the number of anomaly clusters.

You can edit the following view settings:

- Time frame for all views on the page, as described in Set a Custom Time Frame
- View title and context

This view is included by default on the Anomaly Detector page in the NetOps Portal Console.

Anomaly Drill-In

The Anomaly Drill-In table lists the following information for each anomaly:

- Probability
- Value
- Originating router and interface
- Time that the anomaly occurred

If you drill into an anomaly cluster from the Enterprise-wide Correlated Anomalies view, the Anomaly Drill-In table opens. You can use the Date link to drill into a trend chart that shows the value and probability over time.

The Anomaly Drill-In view provides the following information about each anomaly:

- **Anomaly Type**
  The type of anomalous behavior. For a description of each anomaly type that you can enable for monitoring, refer to Sensors Overview.
- **Host**
  The name or IP address of the host on which the anomalous behavior is detected. The host may be a client system, a server, a router, or an interface. The program attempts to resolve the hostname of any IP address and displays that name in this field.
- **Host Link**
  Click a Host link to go to more granular information about the device that has the anomaly. Clicking a Host link may be the first step in troubleshooting the anomaly.
  Host link destinations are based on the sensor type. For many CA Network Flow Analysis sensors, the Host link opens the page for defining a Flow Forensics report in the NFA console, which has pre-populated report filters.
- **Prob(%)**
  The calculated likelihood that flagged packet flows are truly anomalous.

Probability is expressed as a percentage. For example, if the probability for an anomaly type is 91%, the packet flows that triggered the reported anomalous behavior are calculated to have a 91% probability of being truly anomalous. In this case, the packet flows have a low probability of occurring normally on this network.
For more information about the probability algorithm, see Probability Thresholds.

- **Value**
  The value that triggered the report of anomalous behavior, expressed in the units of measure shown in the Unit column. For example, the value could be the number of gigabytes of data in the anomalous flow.
- **Metric/Unit**
  The unit of measurement that is used to express the Value, such as packets, flows, or destination hosts.
- **Discovered by**
  The router, interface, or data source that detected the anomalous data.
- **Discovered by Link**
  Click a Discovered By link to view details. The link destination is determined by the type of anomaly:
  – CA Network Flow Analysis Anomalies: router or interface page in the NetOps Portal Console
  – Anomalies from other data sources: main page for the originating product
- **Date**
  The date and time that the anomalous behavior is detected. The time may vary by up to 15 minutes from the exact time when the flows took place. Data is pulled from the harvesters for analysis at 15-minute polling intervals.
- **Date Link**
  Click the Date link to view the Anomaly Trend view. This view shows the value and probability of the anomaly over time.

You can edit the following view settings:

- Time frame, as described in Set a Custom Time Frame.
- View title.

NOTE
You can use the zoom feature to limit the time frame.

Anomaly Trend

The Anomaly Trend view shows the value and probability of the anomaly over time. To display this trend chart, click a link in the Date column in either the Anomaly Drill-in view or the Enterprise-Wide Anomalies view.

The view shows the pattern of deviation from normal network behavior. A longer term view can help to determine patterns over days, weeks, or months. The recorded values are shown as a blue trend line on the X-Axis. The probability that the behavior is a true anomaly is shown as a pink trend line on the Y-Axis.

You can edit the following view settings:

- Time frame, as described in Set a Custom Time Frame
- View title

NOTE

You can use the Zoom feature to limit the time frame.

Enterprise-Wide Anomalies

The Enterprise-Wide Anomalies view is a comprehensive summary of the anomalous behavior during the reporting time frame, with details about the anomaly type, location, and size.

This view is useful for examining problematic behavior, or for initiating troubleshooting procedures to stop an attack. The view provides more detailed information about the anomalies that you see in other views. This view also identifies network locations that you should investigate.

You can edit the following view settings:

- Time frame for all views on the page, as described in Set a Custom Time Frame
- View title and context

This view is not included by default on the Anomaly Detector page in the NetOps Portal Console. To see this view, add it to a page.

The view provides the following information about anomalous network behavior:

- **Anomaly Type**
  The type of anomalous behavior. For a description of each anomaly type that you can enable for monitoring, see Sensors Overview.
- **Host**
  The name or IP address of the host on which the anomalous behavior is detected. The host may be a client system, a server, a router, or an interface. The program attempts to resolve the hostname of any IP address and displays that name in this field.
- **Probability**
  The calculated likelihood that flagged packet flows are truly anomalous.
  Probability is expressed as a percentage. For example, if the probability for an anomaly type is 91%, the packet flows that triggered the reported anomalous behavior are calculated to have a 91% probability of being truly anomalous. In this case, the packet flows have a low probability of occurring normally on this network.
  For more information about the probability algorithm, see Probability Thresholds.
- **Date**
  The date and time that the anomalous behavior is detected. The time may vary by up to 15 minutes from the time when the flows actually took place. Data is pulled from the Harvesters for analysis at 15-minute polling intervals.

Enterprise-Wide Correlated Anomalies

The Enterprise-Wide Correlated Anomalies table summarizes the anomalous behaviors that can damage the network. This view identifies network locations that you can investigate if you suspect malicious activity.

Anomaly clusters are better indicators for problems than single anomalies. Many types of attacks involve multiple instances of anomalous network behavior. Instances are often clustered around a group of a few hosts at first. The behavior then spreads out, and seemingly unrelated devices are affected; unexpected traffic is produced from multiple sources.

Correlation is performed by using an algorithm that considers the typical patterns for each type of monitored network traffic.

An anomaly is *correlated* when the following requirements are met:

- Three or more anomaly instances exist
- Two different anomaly types are present, or have an Anomaly Index above 2.0
- One device is the source of the anomalies

You can change the time frame for this view and all views on the page, as described in Set a Custom Time Frame.

You can edit the following view settings:

- Time frame for all views on the page, as described in Set a Custom Time Frame
- View title and context

This view is included by default on the Anomaly Detector page in the NetOps Portal Console.

The view provides the following information about anomalous network behavior:

- **Host**
  The IP address of the host that displays the anomalous behavior. The host may be a client computer, a server, a router, or an interface. The program attempts to resolve the hostname of the IP address, and displays that name in the Host field.
- **Anomaly Index**
  The count of the anomalies in the cluster, weighted by their role as either primary or secondary. The anomaly correlation algorithm compares each particular behavior to the typical patterns for the network traffic type. A higher index number indicates a more severe issue.
- **Types**
  The number of different types of anomalous network behaviors that occurred during the reporting period.
- **Date**
  The date and time of the first correlated anomaly on the host.
  The time may vary by up to 15 minutes from the time when the flows actually took place. Data is pulled from the Harvesters for analysis at 15-minute intervals.
- **Date Link**
  Click the Date link In the Enterprise-Wide Correlated Anomalies view to view the Anomaly Detector Drill-In table.

Links and Detail Pages

Links are included in some views to help start anomaly troubleshooting. Use the links to view preconfigured reports or more details. The following views include links:

- Enterprise-Wide Correlated Anomalies
- Enterprise-wide Anomalies
- Anomaly Detector Drill-In

The following links are available in these views:

- **Date Link**
  Opens the Anomaly Trend view. This view shows the value and probability of an anomaly over time.
- **Discovered by Link**
  View the details of an anomaly. The link destination is determined by the type of anomaly:
  - CA Network Flow Analysis Anomalies: router or interface page in the NetOps Portal Console
  - Anomalies from other data sources: main page for the originating product
- **Host Link**
  View more granular information about the device that has the anomaly. Click this link to start troubleshooting the anomaly.
  Host link destinations are based on the sensor type. For many CA Network Flow Analysis sensors, the Host link opens the page for defining a Flow Forensics report in the NFA console, which has pre-populated report filters.

Top Anomalies by Host

The Top Anomalies by Host pie chart shows the top anomalous hosts, ranked by the number of anomalies for the reporting time frame. A maximum of ten hosts are included. The number of instances is shown next to each sector.

You can edit the following view settings:

- Time frame for all views on the page, as described in Set a Custom Time Frame
- View title and context

This view is included by default on the Anomaly Detector page in the NetOps Portal Console.

Top Anomalies by Interface

The Top Anomalies by Interface pie chart shows the anomalies for the top interfaces, ranked by number of anomalies. A maximum of ten anomalies are included. The number of instances is shown next to each sector.

You can edit the following view settings:

- Time frame for all views on the page, as described in Set a Custom Time Frame
- View title and context

This view is included by default on the Anomaly Detector page in the NetOps Portal Console.

Top Enterprise-Wide Network Anomalies

The Top Enterprise-Wide Network Anomalies pie chart shows the top anomaly types for the reporting time frame. This view shows the type of network traffic that had the highest proportion of anomalous traffic. This data may give you the first insight into poor network performance. The legend identifies the number of instances and the colors for each anomaly type. Anomaly types are named for the corresponding sensors. For a description of each sensor, see Sensors Overview.

The Top Enterprise-Wide Network Anomalies view is most useful for tracking sudden changes in network behavior. For example, suppose that the Enterprise-Wide Network Anomalies view shows that the Large DNS Packet Sources category accounts for 25% of all potentially anomalous behavior for the past week. If the summary indicates that Large ICMP Packets account for 50% of such traffic today, you can start to investigate.

You can edit the following view settings:

- Time frame for all views on the page, as described in Set a Custom Time Frame
- View title and context

This view is included by default on the Anomaly Detector page in the NetOps Portal Console.

Calendar Chart (Flow)

The Calendar Heat Chart (Flow) view maps the utilization percentage of the selected interface over time to help you find recurring data patterns. Finding a pattern can help you to identify the source of high traffic rates and potential performance issues. For example, the view can show the hour of each day when utilization is the highest.

Each color represents a severity range that is calculated as a percentage of total capacity. High utilization is shown in orange and red. Low utilization is shown in green and blue.

The view includes the following information:

- **Identifier**
  Consists of the router name, interface name, and interface description (under the view title). The interface description consists of the ifDescr value by default, and can differ from the interface description that is shown in the NFA console.
- **Month, Date, and Day of the Week**
  Identify the day that the traffic occurred (X-Axis columns).
- **Hour**
  Identifies the hour of the day that the traffic occurred (Y-Axis).

To see the Calendar Heat Chart view in the CA NetOps Portal (CAPC) Console, add it to a custom dashboard.

You can perform several actions in this view:

- Change the data direction and view name, as described in this topic.
- (CAPC) Display details in a Tooltip by holding your cursor over a cell.
- (CAPC) Click Show All and choose a pattern-matching filter. For example, select Busy Hour to show only the data for the busiest hour of each day.

Find the Comparable View in the NFA Console

To display Calendar Chart data for an interface in the NFA console, select an interface on the Interface page and select the following options:

- Report type: Utilization.
- Presentation menu option: Direction In or Direction Out.

To display Flow Forensics-level detail, click the Flow Forensics link and run a Flow Forensics report.

Change the View Settings

You can adjust several settings in the Calendar Heat Chart.

Follow these steps:

1. (CAPC) Click the Edit icon in the view title bar and click Edit.
2. (Optional) Edit any of the following settings in the Calendar Heat Chart (Flow) Settings section:
   - **Title:** Change the name that appears in the view title bar.
   - (**CAPC) Time Display Format:** Select the time format for the chart, either 12 hours or 24 hours.
   - **(CAPC) Zone Start:** Set the starting value of each heat zone. The defaults are based on IT industry standards for performance. For example, the default Red Zone Start value is 70 percent utilization.
     Defaults: Green Zone Start = 0, Yellow Zone Start = 50, Orange Zone Start = 60, Red Zone Start = 70.
   - **(CAPC) Business Week Start:** Select the day that starts the business week.
     Default: Monday.
   - **(CAPC) Direction Settings:** Select the direction of traffic on the selected interface to include in the report:
     - **Out:** Outbound on the interface.
     - **In:** Inbound on the interface.
     - **Total:** Combination of inbound and outbound traffic.
3. (Optional) (CAPC) Change the context for the view data: Select a different interface from the Context Settings table.
4. Select the scope of your changes from the **Apply Changes** drop-down.
5. Click **Save** to save your changes.
   The settings dialog closes. The view refreshes to reflect your updates.

UC Monitor Views in NetOps Portal

The following topics discuss the CA Unified Communications Monitor dashboards and views that are available in NetOps Portal:

Call Quality Breakdown

The Call Quality Breakdown pie chart shows a rollup of systemwide call quality for the selected time frame. The percentage of all call minutes that fell into each severity range is displayed in the chart.

The call quality data in this chart comes from the Mean Opinion Scores (MOS) of every call leg that was measured within that time frame. MOS values are evaluated according to the Call Quality performance thresholds that are assigned to the Locations where call activity occurred. The thresholds determine which MOS values are rated Normal, Minor, or Major. The threshold values in these severity categories can be customized per location or assigned automatically per codec.

MOS values do not apply to video streams. Network MOS is not included.

A unique color is assigned to each severity level shown in the pie chart. A legend explains the color assignments. Data that is unrated can indicate that the Minimum Observations threshold for that location was not met during the selected time frame.

Call Quality Service Level Agreement

The Call Quality Service Level Agreement (SLA) view lets Managed Service Providers (MSPs) prove to a customer that they are meeting the SLA commitment for audio call quality by codec. MSPs create one Call Quality SLA view for each customer.

The Call Quality SLA view provides the following information:

- **Calls Meeting SLA**
  The number of calls that met the SLA commitment during the selected interval, based on your selections on the Settings dialog.
  If both legs of a call have MOS or Network MOS, then both legs must meet the SLA commitment for the call to be included in the Calls Meeting SLA value.
  Calls are not included in the Calls Meeting SLA value when one leg meets the commitment, but the other leg does not.
- **Calls Not Meeting SLA**
  The number of calls that did not meet the SLA commitment during the selected interval.
- **Total Calls**
  Total number of calls observed during the selected interval. Only Calls that had MOS or Network MOS for at least one of the two legs in the call are included in the total. The total includes the number of calls that met the SLA.

  NOTE
  The total does not include calls that do not have MOS or Network MOS, such as short calls, abandoned calls, and calls with setup failures.

  The number in this column is a link to the Calls Overview report in the CA UC Monitor management console. The Calls Overview report can contain up to a week's worth of data, and can therefore take a long time to open.
- **Percent**
  The percentage of calls that met the SLA commitment during the selected interval.
- **Importance**
  Indicates whether the SLA commitment was met, or indicates the importance of a failure to meet the SLA, based on your selections in the Settings dialog.

Call Quality Trend

The Call Quality Trend view provides an enterprise-level view of Mean Opinion Score (MOS) data from all calls detected during the selected timeframe.

About the Call Quality Trend View

The trend chart does not include an indication of the number of call minutes used to derive the metrics. To determine the number of observations behind the trend, navigate to the UC Monitor data source and view the Call Performance Overview report. Click the Metric Details link at the top of the report. MOS values do not apply to video streams. Network MOS is not included.

NOTE
This view does not support data from multiple data sources. If you registered multiple instances of UC Monitor, edit the view to filter by one data source. You can also change the dashboard group context by clicking the Group link. Using either method, you can select one data source in the Groups tree to serve as the view context.

Call Quality Trend View Settings

The Settings dialog provides filtering and display options for the view.

- **Title**
  The default title is Call Quality Trend. You can change the title as necessary.
- **Context Settings**
  Select another managed item to change the source of the data in the view.
- **Apply Changes**
  Select the scope of your changes from the **Apply Changes** drop-down.

Performance Overview Dashboard

The Performance Overview dashboard provides a network manager or higher-level executive with a daily or weekly summary of overall VoIP and video call performance.

You can click a link in a dashboard view and access the related report in the CA UC Monitor management console, with the appropriate context selected. The management console provides more details for the data that is summarized in the dashboard.

Performance by Call Server

The call servers shown in the Performance by Call Server view handled calls during the selected time frame. Their performance ratings are derived from the Locations that they served. The name of each server is a link that allows further investigation.

The Call Performance category includes both call quality and call setup metrics. Where available, video metrics are included. The bar graph represents the number of originated calls or call minutes that were rated Normal, Minor, or Major.

- **Calls Originated:** Calls from endpoints that are registered to a call server.
- **Call Minutes:** Minutes of call activity that were reported by a call server.

The Calls Originated column provides the total number of calls that were set up by the indicated call servers. Calls not included in the Calls Originated total were routed by the call servers in this view, but were set up by a different call server.

Performance by Group

The Performance by Group view is available only when custom groups are defined in CA NetOps Portal (CAPC). You can drill down into individual group members and their data by clicking a group name.

The view rates call performance in the incoming direction to gauge the listening quality for VoIP and video users in that group. Groups are sorted by worst call performance.

Click a Location link to see ratings for performance metrics and for the components participating in calls to the Locations in this view.

The Performance by Group view displays calls between Locations in the selected group. The view also identifies all associated call servers, including calls servers that are not explicitly part of the group definition.

Performance by Location

The Performance by Location view lists all monitored Locations where endpoints had call activity. This view evaluates call performance in the incoming direction to gauge the listening quality for VoIP and video users.

Data that is unrated can indicate that the Minimum Observations threshold for that Location was not met during the selected time frame.

To see ratings for performance metrics and for the components participating in calls to the Locations listed in this view, click a Location link.

Performance by Media Device

The Performance by Media Device view shows incoming calls from the PSTN and outgoing calls from the IP network that the device handled.

The media device category includes voice gateways and other devices that support call routing and processing. The metrics available for each device vary according to the device type and the environment. In a Microsoft environment, SNMP polling of media devices is not possible. As a result, fewer metrics are available for Microsoft media devices than for Cisco voice gateways.

The Calls Originated column provides the total number of calls that originated at points in the PSTN.

Performance Overview Dashboard View Settings

The Settings dialog provides filtering and display options for each view in this dashboard:

- **Title**
  The default titles for the views in this dashboard are:
  - **Performance by Call Server**
  - **Performance by Group**
  - **Performance by Location**
  - **Performance By Media Device**

  You can change the titles as necessary.
- **Context Settings**
  Select another managed item to change the source of the data in the view.
- **Apply Changes**Select the scope of your changes from the **Apply Changes** drop-down.

Top Volume and Utilization Dashboard

The Top Volume and Utilization Dashboard helps network engineers plan for network growth and track usage statistics for their unified communications systems. Data from CA UC Monitor can help engineers plan for capacity needs by determining the current operating levels of key unified communications components.

You can click a link in a view to access the related report in the CA UC Monitor management console with the appropriate context selected. The management console provides more details for the data that is summarized in the dashboard.

Contents

Top Groups

The Top Groups Volume view compares the call volume of the groups of locations with the highest usage during the selected timeframe. The view provides a list of the top talkers at a particular point in time. All calls, including audio-only, video-only, and audio and video, are considered when compiling the list of top call volumes.

The Volume bar chart shows a comparison of call volumes among the groups of locations with the highest volumes. Each bar represents a relative activity level, which lets you easily compare call volumes among busy Locations.

The names in the Group column are links to the Top Volume report in the CA UC Monitor interface. You can see the volume statistics of individual group members in the context views included in this report.

By default, the Volume bar is calculated using the number of calls placed by locations in the indicated groups. You can select Call Minutes as the charted unit instead. Click the blue arrow to the left of the view name, and select Edit. In the dialog that opens, select Call Minutes from the “Calculate using” list.

Top Locations

The Top Locations Volume view compares the call volume of the locations with the highest usage during the selected timeframe. The view provides a list of the top talkers at a particular point in time. All calls, including audio-only, video-only, or audio and video, are considered when compiling the list of top call volumes.

The Volume bar chart shows a comparison of call volumes among the locations with the highest volumes. Each bar represents a relative activity level, which lets you easily compare call volumes among busy locations.

The names in the Location/Media Device column are links to the Top Volume report. A related view of the busiest endpoints from the selected Location is also included in the report.

By default, the Volume bar is calculated using the number of calls placed by Locations in the indicated groups. You can select Call Minutes as the charted unit instead. Click the blue arrow to the left of the view name, and select Edit. In the dialog that opens, you can select Call Minutes from the “Calculate using” list.

Top Phones

The Top Phones Volume view compares the call volume of the endpoints with the highest usage during the selected timeframe. The view provides a list of the top talkers at a particular point in time. All calls, including audio-only, video-only, and audio and video, are considered when compiling the list of top call volumes.

The Volume bar chart shows a comparison of call volumes among the endpoints with the highest volumes. Each bar represents a relative activity level so that you can compare call volumes among endpoints.

The numbers in the Phone Number column are links to the Phones report.

If no directory number is available for an endpoint, its IP address is provided in the IP Address column. The IP address can be a link to the endpoint web page. The endpoint name is also shown in the Name column.

Top Trunk Groups

The Top Trunk Groups view helps you understand trunk group usage, and provides individual trunk usage statistics. Use this page to find underutilized trunk groups and overburdened trunks, where performance can deteriorate.

Statistics for all known trunk groups are included. The most heavily used trunks are shown first.

- **Name**
  The name assigned to the trunk group. Click to drill down into an hourly breakdown of usage for this trunk group. The name of the trunk group is based on information from the monitored call traffic and the naming convention employed by the trunking equipment.
- **Utilization (%)**
  The average usage during the time period, expressed as a percentage of capacity. Average usage is based on the capacity divided by the timeframe.
  – (*Avaya only)* The trunk group capacity is discovered from SNMP polling of the Avaya Communication Manager. The usage for an Avaya trunk group is based on channel capacity, divided by the Centum Call Seconds **(**CCS) observed

for all active channels. Usage is a percentage of the trunk group channel capacity, divided by the observed CCS. The total is again divided by the selected timeframe.
  - (*Cisco only*) Usage is the percentage of total voice interface capacity in use during the selected timeframe. Total interface capacity is derived from the known capacity of each voice interface in the group.

  To identify underused trunk groups, sort this column to see the least used trunk group first.
- **Maximum Utilization**
  The highest recorded usage during the selected time period, expressed as a percentage of capacity. Maximum usage is a good indicator of potential capacity issues. Even when the average usage is low, the maximum usage is significant because it often indicates the usage during the “busy hour” of the day. Maximum usage is based on the capacity, multiplied by the timeframe.
  - (*Avaya only*) The capacity is based on information from the Avaya Communication Manager.
  - (*Cisco only*) The capacity is discovered when voice gateway devices are discovered.
- **Grade of Service (GoS)**
  (*Cisco only*) An estimation of the probability that a VoIP call will receive a busy signal. The GoS value (a decimal fraction) is always expressed with reference to the busy hour when the traffic intensity is the greatest. GoS is reported from the perspective of the origination Location or gateway device (the outgoing direction).
- **Call Minutes**
  The number of call minutes used to calculate usage statistics. This value provides a sense of the scope of activity, and helps to determine the significance of the data, based on sample size.
- **Call Minutes Capacity**
  The number of call minutes that were supported during the selected time period. The capacity is discovered during monitoring, or is manually supplied during configuration.

Top Voice Interfaces

The Top Voice Interfaces view helps you to understand voice gateway usage, and provides individual gateway voice interface statistics. Use this view to look for overburdened interfaces, where performance can deteriorate. The minimum timeframe for this view is one day; no data appears if a narrower timeframe is selected. This view contains:

- Statistics for all known gateway voice interfaces.
- The most heavily used interfaces, which are shown first.

Top Volume and Utilization View Settings

The Settings dialog provides filtering and display options for the views in this dashboard:

- **Title**
  The default titles for the views in this dashboard are:
  - **Top Groups**
  - **Top Locations**
  - **Top Phones**
  - **Top Trunk Groups**
  - **Top Voice Interfaces**

  You can change the title as necessary.
- **Context Settings**
  Select another managed item to change the source of the data in the view.
- **Apply Changes**
  Select the scope of your changes from the **Apply Changes** drop-down.

Worst Performance Dashboard

The Worst Performance Dashboard helps IT staff monitor the quality of unified communications system performance and summarize the performance data. By default, the Worst Performance dashboard provides the following views, which focus on the Locations and phones with the worst performance.

You can click a link in a view to access the related report in the CA UC Monitor management console, with the appropriate context selected. The management console provides more details of the data that is summarized in the dashboard.

Worst Locations

The Worst Locations view shows the pairs of Locations that had the lowest-quality metrics for the data traveling between them. The Worst Locations view provides the following information.

- **Name**
  The name of the location or media device that received the data stream with poor performance metrics. The Name could also refer to the name of the origination location for a data view filtered by a call setup metric.
- **Sending Name**
  The name of the location or media device that sent the low-performing call data to the other location in the pair.
- **Call Server**
  The call server that handled the calls for the shown severity breakdown.
- **Call Minutes**
  The number of minutes that calls were active between this pair of locations.
- **Calls**
  The number of distinct calls that ran between this pair of locations, and that also contributed to the “worst” performance metric displayed in the table.
- **Severity Breakdown**
  A stacked bar chart that shows applicable severity ratings as portions of the bars, which total 100 percent. Severity ratings are color-coded to match the severity indicators in other reports.
- **Unrated, Normal, Minor, Major**
  These columns show the actual severity percentages for the selected quality metric type. Percentages always total 100 percent.

  NOTE

  : An unrated metric indicates that a threshold is disabled, or that the threshold for minimum observations is too high for typical levels of call traffic.

Worst Locations Settings

The Settings dialog provides filtering and display options for the view:

- **Title**
  The default title is Worst Locations. You can change the title as necessary.
- **Metric Type**
  Select a different metric to change the type of data in the view.
- **Context Settings**
  Select another managed item to change the source of the data in the view.
- **Apply Changes**
  Select the scope of your changes from the **Apply Changes** drop-down.

Worst Phones

The Worst Phones view provides a list of endpoints with the lowest MOS for the selected timeframe. Endpoints are identified by the directory number or SIP URI, and by their IP address.

- **Phone Number**
  The directory number or SIP URI of the endpoint.
- **Name**
  The name, MAC address, or host name of the endpoint.
- **IP Address**
  The IP address of the endpoint.
- **Call Minutes**
  The number of minutes that calls were active on this endpoint during the selected timeframe.
- **Calls**
  The number of distinct calls placed or received.
- **Average Metric**
  The average MOS or Network MOS for all calls to and from this endpoint during the selected timeframe. Unlike the severity reflected in the bar chart, the average is not weighted by the duration of the calls. The contents of this column are determined by your selection in the Metric Type field in the Settings dialog.
- **Severity Breakdown**
  A stacked bar chart that shows each severity rating as a portion of the bar, which totals 100 percent. The severity ratings are color-coded to match the severity indicators in other reports.
- **Unrated, Normal, Minor, Major**
  Depending on your selection in the "Show breakdown values as" field on the Settings dialog, these columns display one of the following:
  – The percentage of calls in a severity category. Percentages always total 100 percent.
  – The number of calls in a severity category

  NOTE

  : An unrated metric indicates that a threshold is disabled, or that the threshold for minimum observations is too high for typical levels of call traffic. Set a lower minimum for observations, or assign different thresholds to the endpoints in question.

Worst Phones Settings

The Settings dialog provides filtering and display options for the view.

- **Title**
  The default title is Worst Phones. You can change the title as necessary.
- **Metric Type**
  Select MOS or Network MOS to determine the contents of the Average Metric and Severity Breakdown columns.
- **Show breakdown values as**
  Select Calls or Percent to determine the contents of the Unrated, Normal, Degraded, and Excessive columns.
- **Context Settings**
  Select another managed item to change the source of the data in the view.
- **Apply Changes**
  Select the scope of your changes from the **Apply Changes** drop-down.

Monitor Server Performance with DX Application Performance Management

The NetOps Portal, Data Aggregator, and Data Collector components are supported for instrumentation with CA Application Performance Management (APM). APM receives performance metrics about the component services and hosts. During installation of DX NetOps Performance Management components, the installer can configure the CA APM agent. You can also configure the APM agent for NetOps Portal, Data Aggregator, and Data Collector post-installation.

NOTE

After each configuration, restart the corresponding services to see that your changes take effect.

Follow these steps:

1. Log in to the host for the relevant component.
2. Change the working directory:

   ```
   cd Installation_Directory/wily
   ```

3. Run the following script:
   `setEMHost.sh`
   The script requests the following parameters:
   – CA APM hostname
   – CA APM port
   Default: 5001
4. On the NetOps Portal host, change the working directory:

   ```
   cd PC_Installation_Directory/PerformanceCenter/wily
   ```

   PC_Installation_Directory
   Specify the directory where NetOps Portal is installed.
5. Run the following commands:

   ```
   ./addWily.sh  PC_Installation_Directory/PerformanceCenter  PC_Installation_Directory/
   PerformanceCenter/PC/conf/wrapper-user.conf  PC

   ./addWily.sh  PC_Installation_Directory/PerformanceCenter  PC_Installation_Directory/
   PerformanceCenter/sso/conf/wrapper-user.conf  sso

   ./addWily.sh  PC_Installation_Directory/PerformanceCenter  PC_Installation_Directory/
   PerformanceCenter/DM/conf/wrapper-user.conf  DM

   ./addWily.sh  PC_Installation_Directory/PerformanceCenter  PC_Installation_Directory/
   PerformanceCenter/EM/conf/wrapper-user.conf  EM
   ```

6. If monitoring is configured for NetOps Portal, restart all NetOps Portal services:

   ```
   service caperfcenter_console restart
   service caperfcenter_devicemanager restart
   service caperfcenter_eventmanager restart
   service caperfcenter_sso restart
   ```

7. If monitoring is configured for the Data Aggregator, restart the ActiveMQ service for the Data Aggregator:

   ```
   service activemq stop

   service activemq start
   ```

8. If monitoring is configured for the Data Aggregator, do one of the following steps:
   – Stop the Data Aggregator service:

     ```
     service dadaemon stop
     ```

     NOTE

     For RHEL 7.x or OL, `service` invokes `systemctl` . You can use `systemctl` instead.
   – (Fault tolerant environment) If the local Data Aggregator is running, run one the following commands to shut it down and prevent it from restarting until maintenance is complete:
     - **RHEL 6.x:**

       ```
       service dadaemon maintenance
       ```

     - **RHEL 7.x, SLES, or OL:**

       ```
       DA_Install_Directory/scripts/dadaemon maintenance
       ```

9. If monitoring is configured for the Data Aggregator, do one of the following steps:
   – Start the Data Aggregator service:

     ```
     service dadaemon start
     ```

   – (Fault tolerant environment) Run one the following commands to enable the fault tolerant Data Aggregator so that it can start when necessary:
     - **RHEL 6.x:**

       ```
       service dadaemon activate
       ```

     - **RHEL 7.x, SLES, or OL:**

       DA_Install_Directory/scripts/dadaemon activate

10. If monitoring is configured for the Data Collector, restart all Data Collector services:

```
service activemq stop
service activemq start
service dcmd stop
service dcmd start
```

Generate MM Device Packs

DX NetOps Mediation Manager (MM) uses device packs to monitor data from non-SNMP devices or obtain data from the Element Management System (EMS). MM includes device packs that support many vendors and device types. To support a new device type or a new vendor, create and deploy device packs in your environment. Use the MM device pack generator application.

For more information, see the documentation for DX NetOps Mediation Manager.

Integrate DX Spectrum for Fault Management

The DX NetOps Spectrum integration with DX NetOps Performance Management shares models, Global Collections, and events between the two systems. DX NetOps Spectrum contributes devices, interfaces, and groups to the DX NetOps Performance Management inventory, which DX NetOps Performance Management can monitor. DX NetOps Performance Management contributes infrastructure performance events to DX NetOps Spectrum, so you can see performance events and fault alarms side by side in OneClick.

This following video examines the integration between these two products:

Integration Features

The following list identifies supported features in the integration between DX NetOps Spectrum and DX NetOps Performance Management:

Device Integration

From DX NetOps Spectrum, determine which devices to contribute to the DX NetOps Performance Management inventory. DX NetOps Performance Management creates discovery profiles to monitor those devices. DX NetOps Performance Management automatically discovers interfaces that are associated with those devices. If the devices exist in DX NetOps Performance Management, the devices are reconciled to a single item.

> **WARNING**
>
> DX NetOps Spectrum SNMP throttling does not apply to ongoing polling by Data Aggregator. This feature protects critical devices from failing in case too many polling flows are configured. The throttling mechanism applies to any monitoring or discovery activities. If you have configured throttling in DX NetOps Spectrum, apply the same setting in Data Aggregator. For more information, see Poll Sensitive and Critical Devices Without a Performance Impact.

Drill Down from OneClick to DX NetOps Performance Management Data

Access DX NetOps Performance Management data from DX NetOps Spectrum device and interface models, which provides rapid access to contextual information about device performance issues.

> **TIP**
>
> To navigate directly to NetOps Portal from OneClick, right-click a device or interface, and click **NetOps Portal** . A new window opens, and DX NetOps Performance Management loads the context page for the selected device or interface.

Model Classes and Device Subtypes

Synchronization uses the DX NetOps Spectrum model class to determine the DX NetOps Performance Management device subtype. The following list shows the model class followed by the device subtype:

- Router = Router
- Switch-Router = Router
- Switch = Switch
- Workstation-Server = Server

All other model classes are listed as **Other** in DX NetOps Performance Management.

WARNING

A device that is synchronized from DX NetOps Spectrum appears as pingable if any of the follow configuration issues occur:

- The SNMP profile in DX NetOps Performance Management does not have the correct contact information for the device.
- The SNMP profile is not assigned to the discovery profile.
- A local firewall is preventing communication from the Data Collector.

Synchronized Discovery

Synchronization reduces the administrative management that is required for device discovery. DX NetOps Performance Management adds the IP addresses of synchronized devices from DX NetOps Spectrum to a discovery profile. The discovery profile is specified for each IP domain. Run the discovery profile to identify the devices through SNMP. For more information, see Run Discovery.

To enable synchronized discovery, edit the Data Aggregator data source, and select **Discover devices from other data sources**. Synchronization occurs approximately every 5 minutes.

Life Cycle Status

You can configure DX NetOps Spectrum to control the life cycle state of devices in DX NetOps Performance Management. Changes in DX NetOps Spectrum trigger changes in DX NetOps Performance Management. If you change the state of a device in DX NetOps Performance Management, the state does not change again unless the state changes in DX NetOps Spectrum. With this option enabled, DX NetOps Performance Management uses the following behavior:

- Active devices in DX NetOps Spectrum are Active in DX NetOps Performance Management.
- Devices in Maintenance in DX NetOps Spectrum have the Maintenance state in DX NetOps Performance Management.
- The state of the device in DX NetOps Spectrum overwrites the state in DX NetOps Performance Management.

TIP

To avoid conflicts in device life cycle management, do not grant the Administer Life Cycle role to users in DX NetOps Performance Management.

To manage the life cycle state of a device from DX NetOps Spectrum, edit the DX NetOps Spectrum data source, and select **Synchronize device life cycle state from Spectrum**.

For more information about life cycle in DX NetOps Performance Management, see Manage Device Life Cycles.

Interface Synchronization

For each device that DX NetOps Spectrum sends to DX NetOps Performance Management, all the monitored interfaces from DX NetOps Spectrum are also synchronized. For each devices, DX NetOps Performance Management shows the following for interfaces:

- Interfaces that are synchronized from DX NetOps Spectrum.
- Interfaces that are discovered directly through SNMP
- Interfaces that are monitored by other data sources, such as CA Application Delivery Analysis or Network Flow Analysis.

NOTE

DX NetOps Performance Management does not show interfaces that are filtered out at the monitoring profile level. If you right-click an interface model in OneClick, the drill-down option is not available to you if that interface

is filtered out of the corresponding monitoring profile in DX NetOps Performance Management. For more information, see Configure Monitoring Profiles.

IP Domains

DX NetOps Performance Management IP domains are synchronized as CA NetOps Portal IP Domain models in OneClick. The IP domain models appear in the same area as Global Collections in the OneClick Navigation panel. The IP domain models have the same names as the DX NetOps Performance Management IP domain definitions. If you add a device model to the IP domain in OneClick, the device is synchronized to that IP domain in DX NetOps Performance Management.

NOTE

All IP domain definitions are synchronized, regardless of tenant associations.

Groups

DX NetOps Spectrum Global Collections and landscapes are synchronized to DX NetOps Performance Management as groups in the CA NetOps Portal Groups tree. The groups are created under Inventory/Data Sources in the group tree.

Use these groups for the following tasks:

- Create reporting groups.
- Define site membership.
- Drive the content of other custom groups and collections.

Multi-Tenancy

DX NetOps Spectrum devices that are synchronized to DX NetOps Performance Management belong to the tenant who owns the associated IP domain. For items that are shared among multiple tenants, add the items to the appropriate Service Provider groups in NetOps Portal. For more information, see Groups and Multi-tenancy.

Event Integration

Event integration converts DX NetOps Performance Management performance events to DX NetOps Spectrum alarm set or clear events that are asserted on models in each landscape. Polling for supported events begins when synchronization completes. Alarms that originate in DX NetOps Performance Management appear in the OneClick Console.

WARNING

When a device is modeled as a non-proxy model on multiple landscapes, event processing cannot map the event to the correct devices in DX NetOps Spectrum.

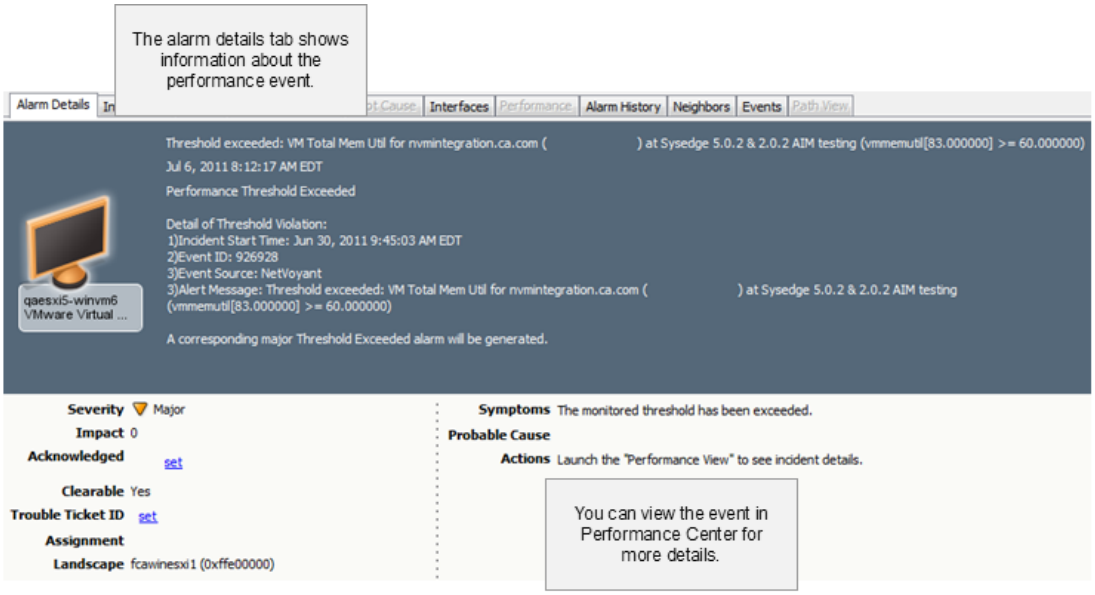
NOTE

When a device model is in maintenance mode, DX NetOps Spectrum does not process performance events that are synchronized from DX NetOps Performance Management.

By default, DX NetOps Spectrum supports events from Data Aggregator and Network Flow Analysis data sources. You can configure OneClick to handle other events from the Event Manager in DX NetOps Performance Management.

The following image shows the alarm details for a DX NetOps Performance Management performance event in the OneClick console. The alarm includes the alarm type, time of occurrence, event ID, and source:

Figure 64: Infrastructure Performance Events in CA Spectrum



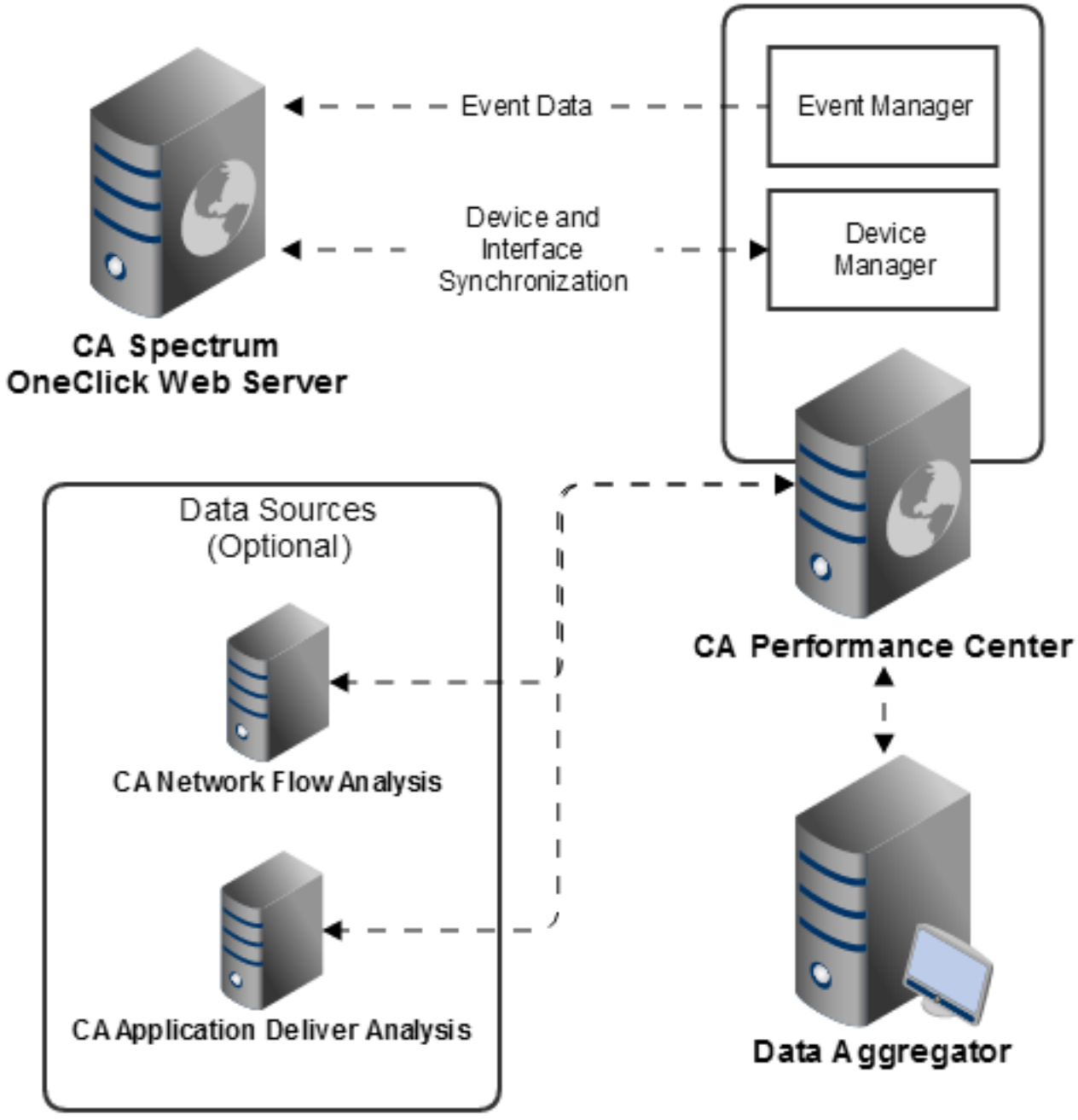
Alarm Integration

Alarms views in NetOps Portal let you view and manage DX NetOps Spectrum alarms. An alarms view provides a prioritized list of DX NetOps Spectrum alarms, which helps you quickly focus on resolving the most impactful problems. The alarms view also provides visibility into other, potentially related, issues on the same device, or connected to a device. For more information, see Alarms View.

Integration Architecture

The following image illustrates the architecture of the integration:

Figure 65: Solution Architecture with CA Spectrum and CA Performance Management



The OneClick server synchronizes devices with the device manager on the NetOps Portal host. The OneClick server polls the event manager for performance events.

- One SpectroSERVER or a Distributed SpectroSERVER (DSS) can be synchronized with DX NetOps Performance Management by specifying the OneClick web server as a data source.
- Each landscape in the DSS is defined as a NetOps Portal group.
- Devices and interfaces in the DSS are synchronized with DX NetOps Performance Management and added to the appropriate landscape group.
- OneClick polls the Event Manager for events that are relevant to a specified landscape group. This polling occurs every 60 seconds by default. Any retrieved events are then translated to DX NetOps Spectrum events, which can generate or clear alarms.
  DX NetOps Spectrum only processes alarms for devices and interfaces that are synchronized to DX NetOps Performance Management.

Integrate DX Spectrum with DX Performance Management

To monitor DX NetOps Spectrum devices and interfaces with DX NetOps Performance Management and process DX NetOps Performance Management events in DX NetOps Spectrum, configure the integration.

Verify the Prerequisites

To ensure a successful integration, confirm the following prerequisites:

- To verify that the product versions are compatible, see the Data Source Compatibility Matrix.

  NOTE

  To utilize DX NetOps Virtual Network Assurance 3.6 with DX NetOps Spectrum, install DX NetOps Virtual Network Assurance 3.6 or DX NetOps Virtual Network Assurance 3.5.1 (available from Support), and DX NetOps Spectrum 10.2.3 available from the Downloads area.
- To discover DX NetOps Spectrum devices with the Data Aggregator, verify that the Data Aggregator is configured correctly:
  a. Hover over **Administration**, and click **Data Sources: Data Sources**.
  b. Select the Data Aggregator data source, and click **Edit.**
  c. Select **Discover devices from other data sources**.
  d. Click **Save**.

  For more information, see Configure a Data Source.
- Create IP domains in NetOps Portal for each IP routing space that DX NetOps Spectrum monitors. For more information, see IP Domains.
- For each IP domain, install a Data Collector and assign it to that IP domain. For more information, see Install the Data Collectors.
- For event integration, verify that the DX NetOps Spectrum OneClick server can communicate to the NetOps Portal host on port 8281.

Configure DX NetOps Spectrum as a Data Source

To connect DX NetOps Performance Management and DX NetOps Spectrum, add DX NetOps Spectrum as a data source in NetOps Portal.

Follow these steps:

1. In NetOps Portal, hover over **Administration**, and click **Data Sources: Data Sources**.
2. Click **Add**.
3. Select **Spectrum Infrastructure Manager** in the **Source Type** field.
4. Specify the following information:
   – **Host Name**

The IP address or DNS host name of the OneClick server

- **Port**
  The communication port for the OneClick server
  Default: 8080 (Unix), 80 (Windows)

5. Select the communication protocol.
   Before you select **https**, ensure the following prerequisites:
   - SSL is enabled on the OneClick web server host by configuring the server.xml and axis2.xml files appropriately.
   - The OneClick SSL certificates and any intermediate certificates have been imported to the following file on the NetOps Portal host: `/opt/CA/jre/lib/security/cacerts`
     For more information, see the DX NetOps Spectrum documentation.
6. (Optional) Specify a display name for the data source.
   By default, the data source type and the hostname are combined to create the display name.
7. If the Web Console is *not* on the OneClick server, clear this option, and specify the host name, port, and protocol for the Web Console server.
8. (Optional) To enable DX NetOps Spectrum to control the life cycle state of items in DX NetOps Performance Management, select **Synchronize device life cycle state from Spectrum**.
9. Specify the username and password of an administrator account on DX NetOps Spectrum with access to the REST API.
10. Click **Test** to verify that DX NetOps Performance Management can contact the OneClick server and the Web Server.
11. Click **Save.**
    DX NetOps Performance Management begins synchronization with DX NetOps Spectrum. DX NetOps Performance Management IP domains appear in DX NetOps Spectrum.

Enable Monitoring in DX NetOps Performance Management

This integration configures DX NetOps Performance Management to monitor DX NetOps Spectrum devices for performance data.

Create SNMP Profiles

SNMP profiles provide authentication credentials to communicate with devices in the monitored network.

> **TIP**
>
> In a multi-tenant environment, create the SNMP profiles in the tenants that include the IP domain to which the DX NetOps Spectrum devices were added.

Follow these steps:

1. Hover over **Administration**, and click **Configuration Settings: SNMP Profiles**.
2. Click **New**.
3. Complete the fields, and change any default settings. Some fields apply only to SNMPv3.
   For complete details, see SNMP Profiles.
4. Click **Save**.
   The SNMP profile is added to the system and used for discovery and polling.

Add Device Models to IP Domain Models

To select which DX NetOps Spectrum devices are discovered in DX NetOps Performance Management, add those devices to the CA NetOps Portal IP domain Global Collections in OneClick. When synchronization occurs, all IP domains are sent to DX NetOps Spectrum. Use search rules to update IP domain membership dynamically in DX NetOps Spectrum.

DX NetOps Spectrum devices can only be members of a single IP domain. If you attempt to add a model to multiple IP domains, you see an error message.

IP domains are designated with the following icon in OneClick:

Add Devices to IP Domains Manually

Follow these steps:

1. In any topology, select a device model, right-click the model, and click **Add To, Global Collection(s)**.
   The Select Global Collections dialog opens. The IP Domain models appear in the list of Global Collections.
2. Select the IP domain model, and click **OK**.
   DX NetOps Spectrum adds the devices to the selected IP domain during synchronization.

Update IP Domain Membership Dynamically

Follow these steps:

1. In the **Explorer** tab of the **Navigation** panel, find the **Global Collections** node, and then locate the **NetOps Portal IP Domains**.
2. Right-click an IP domain, and then click **Edit Global Collection**.
   The Edit Global Collection dialog opens.
3. Click **Search Options**, and create search expressions according to the DX NetOps Spectrum Documentation.
4. To identify the landscapes to include when searching, click **Landscapes**.
5. Click **OK**.
   The search rules create dynamic membership for the NetOps Portal IP domain. Changes to the membership are automatically synchronized to DX NetOps Performance Management.

Discovery in DX NetOps Performance Management

During the next synchronization, DX NetOps Spectrum sends the device IP addresses to the selected IP domains in DX NetOps Performance Management. DX NetOps Performance Management reconciles the synchronized devices with existing devices. If the IP address does not belong to a device that the Data Aggregator already monitors, the IP address is added to a discovery profile with the following properties:

- The name of the discovery profiles is the same as the name of the IP domain.
- By default, the discovery profile runs daily.
- To avoid the buildup of IP addresses, the IP address is removed from the discovery profile after the device is discovered.

> **TIP**
> To enable rediscovery, add the IP address of the DX NetOps Spectrum devices to another discovery profile in DX NetOps Performance Management.

For more information about discovery in DX NetOps Performance Management, see Discovery.

Configure Monitoring

To control the performance metric polling from DX NetOps Performance Management, sort the DX NetOps Spectrum devices into collections and configure monitoring profiles. For more information, Configure Monitoring Profiles.

Configure Threshold Profiles

DX NetOps Performance Management analyzes configurable performance thresholds and creates events when performance metrics violate those thresholds. DX NetOps Spectrum process the events and creates alarms on the appropriate models. To create events, configure threshold profiles. For more information, see Configure Threshold Profiles.

Enable Event Polling in DX NetOps Spectrum

Configure event polling in DX NetOps Spectrum. The SpectroSERVER polls the NetOps Portal host for DX NetOps Performance Management threshold events on devices that are modeled in DX NetOps Spectrum. The alarms appear in DX NetOps Spectrum 1-2 minutes after the events occur in DX NetOps Performance Management.

> **TIP**
> You can customize event integration.
> For more information, see Customize Event Integration.

Follow these steps:

1. From the OneClick home page, click **Administration**.
2. In the left panel, click **NetOps Portal Integration Configuration**.
3. For Event Polling, specify the poll interval in seconds, and select **Enabled**.
   Default: 60 seconds
4. Click **Save**.
   DX NetOps Spectrum starts polling events on the next poll cycle.

Send Traps to DX NetOps Spectrum

To send traps to DX NetOps Spectrum, define a matching SNMP Profile in DX NetOps Spectrum and DX NetOps Performance Management. To map the traps from a device to specific DX NetOps Spectrum events, use the MIB Tools application in OneClick. Then complete event customization using Event Configuration. In Event Configuration, define event processing rules, create the event message to display to users, and set other parameters. For more information, see the CA Spectrum documentation.

Multi-tenancy and DX NetOps Spectrum

For more information about multi-tenancy and DX NetOps Spectrum, see Multi-tenancy and CA Spectrum.

Customize Event Integration

By default, DX NetOps Spectrum polls threshold violation and clear events only. You can also configure DX NetOps Spectrum to poll for any events in the DX NetOps Performance Management Event Manager database. To process other events from DX NetOps Performance Management, customize the events that DX NetOps Spectrum polls.

> **NOTE**
> For DX NetOps Spectrum to process an event, the device or port must be modeled in CA Spectrum and included in the synchronization process. To process events that are not associated with an item that is modeled in DX NetOps Spectrum, configure a DX NetOps Performance Management trap notification and use the DX NetOps Spectrum South Bound Gateway.

To determine the event types available to add to DX NetOps Spectrum, see the `em.event_types` table in the Event Manager database.

To configure DX NetOps Spectrum to poll for specific events, complete the following procedures:

Review the Integration Example

This example shows how to configure CA Spectrum to poll for a specific event in the Event Manager database. The event in this example identifies when a router device experiences high memory usage.

1. Identify a device or port for which you want DX NetOps Spectrum to poll for in the Event Manager database. If the device or port is not modeled in DX NetOps Spectrum, model the element. For example, to monitor specific events for a particular router, the router must be modeled in the DX NetOps Spectrum database.
2. Obtain a developer ID from CA Technical Support for use with DX NetOps Spectrum - DX NetOps Performance Management integration. This example uses the default developer ID value, 0xffff.
3. Identify the events for which DX NetOps Spectrum polls. For example, you can identify "Incident" events that originates from CA Application Delivery Analysis.
4. Define the event by modifying the XML file:
   a. Copy the following file to `<$SPECROOT> /custom/netqos/config/container/netqos-integration-application-config.xml:`
   `<$SPECROOT>\tomcat\webapps\spectrum\WEB-INF\netqos\config\container\netqos-integration-application-config.xml`
   b. Open the copied file for editing:
   `/custom/netqos/config/container/netqos-integration-application-config.xml`
   c. Define the custom event. Update the existing eventTypeManager element as follows: add the "Incident" event to the list of events for which to poll, establish an alarm map value, and specify a default alarm clear code.
   The following code shows these changes. Notice that the alarm clear code uses a developer ID.

```
<bean id="eventTypeManager"
   class="com.ca.im.netqos.integration.event.type.EventTypeManager">
   <property name="interestingEventTypes">
     <map>
       <entry key="ThresholdViolation" value-ref="thresholdViolationAlarmCodes" />
       <entry key="Incident" value-ref="IncidentAlarmCodes" />
     </map>
   </property>
   <property name="alarmClearCodes">
     <map>
       <entry key="ThresholdViolation" value="0x5c40009" />
       <entry key="Incident" value="0xffff0004" />
     </map>
   </property>
</bean>
```

   d. Define the alarm map by adding the following new bean element:

```
<bean id="IncidentAlarmCodes"
    class="org.springframework.beans.factory.config.MapFactoryBean">
    <property name="sourceMap">
      <map>
        <entry key="1" value="0xffff0001" />
        <entry key="2" value="0xffff0002" />
        <entry key="3" value="0xffff0003" />
      </map>
    </property>
</bean>
```

   e. Save and close the file.
5. Specify how DX NetOps Spectrum processes the encountered event by updating the Event Disposition file:
   a. Open the following file for editing:
   `<$SPECROOT>\SS\CsVendor\netqos\EventDisp`

b. Add the following map entries for the "Incident" event:

```
#Incident Event
0xffff0001E 50 A 1, 0xffff0001,107
0xffff0002E 50 A 2, 0xffff0002,107
0xffff0003E 50 A 3, 0xffff0003,107
0xffff0004E 50 C 0xffff0001,107 C 0xffff0002,107 C 0xffff0003,107
```

c. Save and close the file.

WARNING

Back up this file in case the contents are changed during a DX NetOps Spectrum upgrade.

6. Create an event format file for each of the alarm codes using the following naming convention (*AlarmCode* - *EventFormatFile*):
   - 0xffff0001 - Eventffff0001
   - 0xffff0002 - Eventffff0002
   - 0xffff0003 - Eventffff0003
   - 0xffff0004 - Eventffff0004
   - Create a text file containing content similar to the following text:

```
{d "%w- %d %m-, %Y - %T"} - {S 109} is reporting a minor threshold violation.
Detail of Threshold Violation:
        1) Incident Start Time: {D 111}
        2) Event ID: {S 107}
        3) Event Source: {S 113}
        4) Alert Message: {S 76620}
A corresponding minor Threshold Violation Alarm will be generated.
(event [{e}])
```

NOTE
When creating Eventffff0004, use appropriate wording for clearing an alarm.

   - Save the file to the following location:

```
<$SPECROOT>\SG-Support\CsEvFormat
```

   - Repeat steps a and b for each alarm code.
7. Create a probable cause file for each of the alarm codes using the following naming convention (*AlarmCode* - *ProbableCauseFile*):
   - 0xffff0001 - Probffff0001
   - 0xffff0002 - Probffff0002
   - 0xffff0003 - Probffff0003
   - 0xffff0004 - Probffff0004
   - Create a text file containing content similar to the following text:

```
A minor threshold violation has occurred.
SYMPTOMS:
The monitored threshold has been exceeded.
PROBABLE CAUSES:
RECOMMENDED ACTIONS:
Launch the "Performance View" to see incident details.
```

   - Save the file to the following location:

```
<$SPECROOT>\SG-Support\CsPCause
```

   - Repeat steps a and b for each alarm code.
8. Restart the SpectroSERVER and OneClick servers.
   When the integration is complete, DX NetOps Spectrum uses the updated files to poll for the "Incident" event, generating events and alarms as specified.

Obtain a Developer ID

When defining events for the DX NetOps Spectrum - DX NetOps Performance Management integration, you use identifying event codes. The first 2 bytes of any event code contain a developer ID. You can obtain a registered developer ID from CA so that you can specify unique codes for your events. Using a unique developer ID lets you easily recognize your new codes in OneClick. Doing so also prevents potential conflicts with other DX NetOps Spectrum event codes.

To obtain a developer ID from CA, contact CA Technical Support.

Update the netqos-integration-application-config.xml File

DX NetOps Spectrum uses the netqos-integration-application-config.xml file to determine the events for which to poll. DX NetOps Spectrum polls for ThresholdViolation events by default. To poll for more events, modify the netqos-integration-application-config.xml to define the event codes and associated alarms for each event.

The netqos-integration-application-config.xml file is located in the following directory:

```
$SPECROOT\tomcat\webapps\spectrum\WEB-INF\netqos\
config\container
```

Define Events

The eventTypeManager bean defines the events for which DX NetOps Spectrum polls. The entries for ThresholdViolation events appear in the file by default. You can manually add more events.

```
<bean id="eventTypeManager"
   class="com.ca.im.netqos.integration.event.type.EventTypeManager">
   <property name="interestingEventTypes">
     <map>
       <entry key="ThresholdViolation" value-ref="thresholdViolationAlarmCodes" />
       <entry key="TestEvent" value-ref="TestEventAlarmCodes" />
     </map>
   </property>
   <property name="alarmClearCodes">
     <map>
       <entry key="ThresholdViolation" value="0x5c40009" />
       <entry key="TestEvent" value="TestEventAlarmClearCode" />
     </map>
   </property>
</bean>
```

Update the following property elements to add events that DX NetOps Spectrum can include in polling:

- **interestingEventTypes**
  Specifies the types of events to include in polling. Each entry element identifies a specific event type and an alarm code map value. The ThresholdViolation entry is included by default. Add an entry element, as follows:
  ```
  <entry key="TestEvent" value-ref="TestEventAlarmCodes" />
  ```
  – ***TestEvent***
    Specifies the name of an event in the Event Manager database.
  – ***TestEventAlarmCodes***
    Specifies the value of the map that identifies the alarms for this event.

    NOTE
    The alarm code map is described in the next section.
- **alarmClearCodes**

Specifies the alarm clear codes for polled events. The default alarm clear code for the ThresholdViolation event is 0x5c40009. For each event, add an entry element, as follows:

```
<entry key="TestEvent" value="TestEventAlarmClearCode" />
```

– ***TestEvent***
Specifies the name of the event that was added for polling.
– ***TestEventAlarmClearCode***
Specifies the alarm clear code for the event.

Define Alarms

An alarm map defines the alarm code values that are associated with a particular event. For each polled event (or, each interestingEventTypes entry), a corresponding alarm map must be defined. The alarm map for the ThresholdViolation event appears in the file by default. An alarm map for each custom event must be added manually.

```
<bean id="thresholdViolationAlarmCodes"
    class="org.springframework.beans.factory.config.MapFactoryBean">
    <property name="sourceMap">
      <map>
        <entry key="1" value="0x5c40010" />
        <entry key="2" value="0x5c40011" />
        <entry key="3" value="0x5c40012" />
      </map>
    </property>
</bean>
<bean id="testEventAlarmCodes"
    class="org.springframework.beans.factory.config.MapFactoryBean">
    <property name="sourceMap">
      <map>
        <entry key="alarmSev1" value="alarmCode1" />
        <entry key="alarmSev2" value="alarmCode2" />
        <entry key="alarmSev3" value="alarmCode3" />
      </map>
    </property>
</bean>
```

To add alarm maps for custom events, add a bean element for each event and update the following values:

- ***testEventAlarmCodes***
Specifies the alarm code map value for a particular event. This value is established on the interestingEventTypes entry and must match that value.
- ***alarmSev1 - alarmCode1, alarmSev2 - alarmCode2, alarmSev3 - alarmCode3***
Specifies the *alarmSeverity* - *alarmCode* pairs for a particular event. For example, for the default ThresholdViolation event, the Minor (1), Major (2), and Critical (3) alarm codes are 0x5c40010, 0x5c40011, and 0x5c40012, respectively.

Update the Event Disposition File

The Event Disposition (EventDisp) file is used to determine how to process the events that are configured in the netqos-integration-application-config.xml file. Each event entry maps an event to a DX NetOps Spectrum event file.

The EventDisp file for DX NetOps Spectrum - DX NetOps Performance Management integration is located in:

```
<$SPECROOT>\SS\CsVendor\netqos
```

For the default ThresholdViolation event, the following entries map the alarm codes to individual DX NetOps Spectrum event files:

```
text#PC Threshold 0x5c40010 E 50 A 1,0x5c40010,107 0x5c40011 E 50 A 2,0x5c40011,107 0x5c40012 E 50 A
 3,0x5c40012,107 0x5c40009 E 50 C 0x5c40010,107 C 0x5c40011,107 C 0x5c40012,107
```

For each custom event, add new event map entries to the file. The following example shows syntax that generates or clears alarms that are based on the event code.

```
text#New Event alarmCode1E 50 A 1, alarmCode1_filename,107 alarmCode2E 50 A 2,
 alarmCode2_filename,107 alarmCode3E 50 A 3, alarmCode3_filename,107 alarmClearCode4E
 50 C alarmCode1,107 C alarmCode2,107 C alarmCode3,107
```

Create Event Format Files

An event format file contains the message about the event that is displayed to users on the Events tab in OneClick. Each new event that is defined in the netqos-integration-application-config.xml file requires an event format file. The file enables the event to display correctly in the OneClick Events view.

The file name must match the alarm code (for example, alarm code 0x5c40010 uses the file "Event05c40010"). And the file must exist in the following directory:

```
<$SPECROOT>\SG-Support\CsEvFormat
```

The following example shows the file format:

```
text{d "%w- %d %m-, %Y - %T"} - {S 109} is reporting a minor threshold violation. Detail of Threshold
 Violation: 1) Incident Start Time: {D 111} 2) Event ID: {S 107} 3) Event Source: {S 113} 4) Alert Message: {S
 76620} A corresponding minor Threshold Violation Alarm will be generated. (event [{e}])
```

Create Probable Cause Files

A probable cause file defines the symptoms, probable causes, and recommended corrective actions for an alarm. Each new alarm code requires a probable cause file so that the alarm displays correctly in the OneClick Alarms view.

The file name must match the alarm code (for example, alarm code 0x5c40010 uses the file "Prob05c40010"). And the file must exist in the following directory:

```
<$SPECROOT>\SG-Support\CsPCause
```

The following example shows the file format:

```
textA minor threshold violation has occurred. SYMPTOMS: The monitored threshold has been exceeded. PROBABLE
 CAUSES: RECOMMENDED ACTIONS: Launch the "Performance View" to see incident details.
```

Deploy the Changes

After you complete your configuration changes, restart the SpectroSERVER and OneClick servers.

Event polling now reflects your changes.

> **NOTE**
>
> If you have multiple SpectroSERVERs, distribute the event file configuration changes to all servers.

RouterHighMemor

Integrate CA Business Intelligence

The CA Business Intelligence integration with DX NetOps Performance Management offers access to CA Business Intelligence reports and dashboards through NetOps Portal.

CA Business Intelligence (CABI) reports for DX NetOps Performance Management offer portable graphical representations of DX NetOps Performance Management performance metrics. To run a CA Business Intelligence report for DX NetOps Performance Management, install the CA Business Intelligence JasperReports Server. DX NetOps Performance Management content is delivered through a Connector for the CA Business Intelligence JasperReports Server. For more information, see Install CA Business Intelligence Reports and Dashboards.

WARNING

CA Business Intelligence does not support FIPS-compliant encryption. If you enable FIPS-compliant encryption, you cannot register or use CA Business Intelligence. For more information, see FIPS-Compliant Encryption.

Run and access reports based on your DX NetOps Performance Management product privileges. View CA Business Intelligence dashboards in NetOps Portal to evaluate performance metrics including high-level representations of your sites and their monitored devices. For more information, see CA Business Intelligence Reports and Dashboards.

DX NetOps Performance Management integrates with CA Business Intelligence in two ways:

- DX NetOps Performance Management connects to a standalone CA Business Intelligence instance. In this deployment, CA Business Intelligence contains only DX NetOps Performance Management data.
- The *Unified Dashboards and Reporting for Infrastructure Management* solution lets you integrate a single instance of CA Business Intelligence with the following products:
- – DX NetOps Performance Management
  – CA Unified Infrastructure Management
  – DX NetOps Spectrum
  – CA Service Operations Insight

Sharing a single instance of CA Business Intelligence reduces the number of CA Business Intelligence instances that you are required to deploy and maintain. Sharing CA Business Intelligence also enables dashboards that span multiple products, giving you a new level of insight. For more information, see Unified Dashboards and Reporting for Infrastructure Management.

Verify the Prerequisites

For the following information, see the Compatibility Matrix and the CA Business Intelligence documentation:

NOTE

Java 8 is required on the machine where CA Business Intelligence is installed. Java 7 is not supported.

- The supported CA Business Intelligence JasperReports Server versions
- The required third-party tools for the CA Business Intelligence JasperReports Server installation
- The CA Business Intelligence JasperReports Server installation procedure

NOTE

Carefully review the CA Business Intelligence documentation for the list of RHEL Linux versions that support a GUI installation. Use the silent installation for all other versions.

Install CA Business Intelligence Reports and Dashboards

The following procedures apply to configuring the standalone CABI integration and the Unified Dashboards and Reporting for Infrastructure Management solution. For more information, see Unified Dashboards and Reporting for Infrastructure Management.

After you install the CA Business Intelligence JasperReports Server, use the following procedures to set up and run CA Business Intelligence reports in NetOps Portal.

WARNING
A Windows install is unsupported for CA Business Intelligence 7.1.1.

Install Reports on the CA Business Intelligence JasperReports Server

To make CA Business Intelligence reports that use DX NetOps Performance Management data available, install them on the CA Business Intelligence JasperReports Server.

The capm_reports-xxxx.zip package (where "xxxx" is the package version) is located in the *Installation_Directory*/PerformanceCenter/cabi folder after the DX NetOps Performance Management installation.

For CA Business Intelligence on Windows, before you install the reports, Powershell must be enabled.

The installation script can be used for a fresh installation or an upgrade. The script covers the following items:

- Data source connector files
- Data source connector configuration
- NetOps Portal host name, port, and schema definition
- Report files
- Dasboard/dashlets files
- Common folder structure to use in Shared CABI environment

User administration and authentication setting updates are excluded from the script and must be configured manually.

Follow these steps:

1. Verify that the CA Business Intelligence JasperReports Server is available.
2. Copy the capm_reports-xxxx.zip file to the CA Business Intelligence JasperReports Server host and unpack it into a folder for the reports.

   NOTE

   Ensure the report archive (capm_topn-x.x.x.zip) and the data source archive (capm_da-x.x.x.zip) are in the same folder on CA Business Intelligence JasperReports Server as the installation scripts.
3. Run the Install.bat (Windows) or Install.sh (Unix) installation script.

   WARNING

   The DX NetOps Performance Management CA Business Intelligence content installation requires a CA Business Intelligence tomcat instance restart. The CA Business Intelligence web interface is unavailable for approximately 10-15 minutes after the installation start. Ensure that the installation process is done during an appropriate maintenance window.
4. Complete the following prompts:
   - **Input a CA PC target host**
     Specify the NetOps Portal host name. Do not specify the Data Aggregator host name.
   - **Input the NetOps Portal GUID**

     NOTE

     This prompt appears only when the installer cannot connect to the NetOps Portal server and get GUID automatically.

   The NetOps Portal GUID (unique) is used as the default password, and must be configured correctly. The GUID passes to the Data Aggregator to ensure that the request is going to the correct NetOps Portal instance.
   The following web service call on the NetOps Portal host retrieves this information:
   `http://PC_host:8181/pc/center/webservice/datasources/performanceCenterGUID`
   Example Response:

```
<dataSourceGUID name="CA Performance Center" guid="66750a6b-57f6-440e-b501-
d79134d9bb61"/>
```

- **Input a target CA PC port [8181]**
  Specify the NetOps Portal port.
  Default: 8181
- **Input a target schema http/https [http]**
  Specify http or https.
  Default: http
- **CA PC admin user [admin]**
  Specify the NetOps Portal admin user.
  Default: admin
- **CA PC admin user password [admin]**
  Specify the password for the NetOps Portal admin user.
  Default: admin

The script can take up to 10 minutes to complete the installation.

5. To configure the values that are required for generating the OpenAPI token, go to the following location:
   CABIJasperReportsServer/WEB-INF/bundles
6. Edit the capc_config.properties file.

   NOTE

   The OpenAPI token setting values must match the parameter values on the Data Aggregator for (`cookieName` , `tokenEncryptionDecryptionKey` , and `tokenTimeoutInMinutes`) in the following location:

   ```
   /opt/IMDataAggregator/apache-karaf-2.4.3/etc/
   com.ca.im.odata.authservice.impl.AuthenticationService.cfg
   ```

   - **openAPIcookieName**
     The name of the cookie containing the token. This value should match the `cookieName` value.
   - **encryptionKey**
     The key that is used to decrypt incoming tokens. This value should match the `tokenEncryptionDecryptionKey` value.
   - **tokenTimeout**
     The length of time, after which, the token is invalidated. This value should match the `tokenTimeoutInMinutes` value.

Configure User Synchronization

When DX NetOps Performance Management integrates with CA Business Intelligence, you create and manage users in CA NetOps Portal. The users that you create in CA NetOps Portal are replicated in CA Business Intelligence. Replicated users can access reports and dashboards without having to log in to CABI.

NOTE

Note the following:

- User names that contain the following special characters are not synchronized with the CABI Server:
  – , \ | ` " ' ~ ! # $ % ^ & [ ] * + = ; : ? < > } {) (/
  – Spaces
- Synchronization between DX NetOps Performance Management and CA Business Intelligence occurs every 5 minutes. Users that you create in CA NetOps Portal may take up to 5 minutes to appear in CA Business Intelligence.

WARNING

LDAP support is planned for a future DX NetOps Performance Management release.

Follow these steps:

1. On the CA Business Intelligence system, copy the following files in <*CA Business Intelligence*> `/apache-tomcat/webapps/jasperserver-pro/WEB-INF/config`:
   - capm.properties
   - capm.jks
2. Add the copied files in the following location on the system where CA NetOps Portal is installed:
   <*CA NetOps Portal*>/PC/webapps/pc/WEB-INF/CABIKeystore
3. Add CA Business Intelligence Jaspersoft Server as a data source.

Add a Certificate for SSL-Enabled NetOps Portal

For reporting to work in an SSL-Enabled NetOps Portal instance, you must add the appropriate certificates to the CA Business Intelligence trust store. The appropriate certificates are the root or intermediate certificates comprising the chain of trust for the NetOps Portal public certificate. For more information, see Set Up SSL Certificates for Performance Center.

Follow these steps:

1. On the CA Business Intelligence server, enter the following URL in a browser to open CA NetOps Portal:
   https://<*capc_hostname*>:8182
   A warning about an un-trusted certificate appears in the browser.
   The browser prompts you to get the certificate.
2. Save the certificate with the following name in a local folder:
   capc_cert.cer
   The certificate that you saved is the CA NetOps Portal public certificate.

   WARNING

   If the CA NetOps Portal public certificate is not self-signed, you must import the root and intermediate certificates used for signing the CA NetOps Portal public certificate.

   To save the certificate:
   a. Click the certificate warning button near the URL.
   b. Click Details, Copy to File.
   c. Select (DER encoded binary X.509 (.CER)).
3. Open the command line by entering `cmd`.
4. CD to the folder `<CABI_INSTALL_FOLDER>/java/lib/security`
   The folder should contain the cacerts file. The cacerts file is the CA Business Intelligence trust store file.
   If the cacerts file is not in the security folder, search for the folder under the <*CABI_INSTALL_FOLDER*>.
5. Run the following command:
   ```
   <PATH_TO_JAVA>/bin/keytool -importcert -alias ALIAS -keystore cacerts -file <path>/capc_cert.cer
   ```

   NOTE

   In the -file parameter, <path> indicates the full path to the capc_cert.cer file.
6. Enter the password for the cacerts trust store. The default password is "changeit." Then, enter Yes.
7. CD to the folder `<CABI_INSTALL_FOLDER>/jre/lib/security`
   The folder should contain the cacerts file. The cacerts file is the CA Business Intelligence trust store file.
   If the cacerts file is not in the security folder, search for the folder under the <*CABI_INSTALL_FOLDER*>.
8. Run the following command:

```
<PATH_TO_JAVA>/bin/keytool -importcert -alias ALIAS -keystore cacerts -file <path>/
capc_cert.cer
```

NOTE

In the -file parameter, <path> indicates the full path to the capc_cert.cer file.

9. Enter the password for the cacerts trust store. The default password is "changeit." Then, enter Yes.
10. Restart the CA Business Intelligence service.

Configure the CA Business Intelligence Authentication

The following OpenAPI configuration file is available on the Data Aggregator in the /opt/IMDataAggregator/apache-karaf-2.4.3/etc directory:

- com.ca.im.odata.authservice.impl.AuthenticationService.cfg

Customize the following parameters to match the configuration on the CA Business Intelligence server:

- **cookieName**
  The name of the cookie containing the token. This value should match the `openAPIcookieName` value.
- **tokenEncryptionDecryptionKey**
  The key that is used to decrypt incoming tokens. This value should match the `encryptionKey` value.
- **tokenTimeoutInMinutes**
  The length of time, after which, the token is invalidated. This value should match the `tokenTimeout` value.

Verify Time Synchronization

Date and time settings in CA NetOps Portal and CA Business Intelligence must be synchronized. If the date and time settings are not synchronized, an error can occur when users access reports. The error occurs when the difference in time exceeds the Cookies Expiration Timeout set in CA NetOps Portal (Default: 20 minutes).

Silent Installation for CA Business Intelligence Reports and Dashboards

You can automate the installation of reports and dashboards by specifying installation responses in a configuration file, which is called by the install script. In this case, configuration is specified in a single file rather than as manual responses to the installation script.

The following procedures apply to configuring the standalone CABI integration and the Unified Dashboards and Reporting for Infrastructure Management solution. For more information, see Unified Dashboards and Reporting for Infrastructure Management.

After you install the CA Business Intelligence JasperReports Server, use the following procedures to set up and run CA Business Intelligence reports in NetOps Portal.

To perform a silent installation, follow these steps:

1. Complete the steps in Install CA Business Intelligence Reports and Dashboards. Instead of following the procedure in the section called *Install Reports on the CA Business Intelligence JasperReports Server*, complete steps 2 - 4 in this procedure.
2. Copy the capm_reports-xxxx.zip file to the CA Business Intelligence JasperReports Server host and unpack it into a folder for the reports.
   The capm_reports-xxxx.zip package (where "xxxx" is the package version) is located in the Installation_Directory/PerformanceCenter/cabi folder after the DX NetOps Performance Management installation.

NOTE

Ensure the report archive (capm_topn-x.x.x.zip) and the data source archive (capm_da-x.x.x.zip) are in the same folder on CA Business Intelligence JasperReports Server as the installation script

3. Create a Silent Installation configuration file to be used during the process. The file must have following structure:
   a.
   ```
   CABIPath="path_to_CABI"
   host=capc_hostname
   guid=capc_guid
   port=capc_port
   schema=[HTTP|HTTPS]
   JKSRECREATE=No
   ```

   Specify the following information in the configuration file:
   - **CABIPath**: Specify a path to the installed CA Business Intelligence JasperReports Server. If you specify a path with spaces in it, enclose the path in quotation marks.
     - **Default:** `/opt/CA/SharedComponents/CA Business Intelligence`

       WARNING

       Do not add the back slash character '\' or forward slash character "/" to the end of the CABIPath value.
   - **host:** Specify the NetOps Portal host name. Do not specify the Data Aggregator host name.
   - **guid**: Specify a PerformanceGUID for the data source. The NetOps Portal GUID (unique) is used when the password for the CA Business Intelligence web user is not available. This information passes to the Data Aggregator to ensure that the request is going to the correct NetOps Portal instance. A web service call on NetOps Portal retrieves this information.
     GUID Endpoint URL: `http://hostname:8181/pc/center/webservice/datasources/performanceCenterGUID`
   - **port**: Specify the CA NetOps Portal port that the data source uses for retrieval.
     Default: 8181
   - **schema**: Specify HTTP or HTTPS.
     Default: HTTP
     Note: On Linux, you must specify LF at the end of each line.

- Run install.bat on Windows or install.sh on Linux with –silent parameter and a path to a silent installation file. If path or file name contains spaces, enclose the complete path in quotes.
  Examples:
  Windows:
  ```
  install.bat –silent "D:\path to silent install file\silent_params.txt"
  install.bat –silent silentParams.txt
  ```
  Linux:
  ```
  ./install.sh –silent "/path to silent install file/silent_params.txt"
  ./install.sh –silent silentParams.txt
  ```

  WARNING

  Important! After you install the reports and dashboards, complete the steps in the following sections in Install CA Business Intelligence Reports and Dashboards:

  Manage Users

  Add a Certificate for SSL-Enabled NetOps Portal

  Configure the CA Business Intelligence Authentication

Troubleshooting

If the silent installation fails, review the install.log file that is located in the same directory as the installation script.

Symptom:

The install fails and the following error appears:

```
ERROR: The system was unable to find the specified registry key or value.
There is no java.exe in the directory
Installation failed
```

Solution:

For installing on Windows as a non-admin user, use the manual installation steps. For more information, see Install CA Business Intelligence Reports and Dashboards.

Silent Uninstall of the CA Business Intelligence Reports and Dashboards

You can specify uninstall responses in a configuration file to automate the uninstall of reports and dashboards. This silent uninstall removes all content and files related to CA Business Intelligence reports and dashboards.

Follow these steps:

1. Add the uninstall responses to a configuration file with the following structure:

```
CABIPath="CABI_path"

host=PC_host


port=PC_port


schema=Schema
CABIuser=CABI_user
CABIpwd=CABI_password
```

   - ***CABI_path***
     Specify a path to the CA Business Intelligence JasperReports Server. If you specify a path with spaces in it, enclose the path in quotation marks.
     Default: `/opt/CA/SharedComponents/CA Business Intelligence`
   - ***PC_host***
     Specify the local host name where CA Business Intelligence was installed.
     Default: `localhost`
   - ***PC_port***
     Specify the local tomcat port.
     Default: 8080
   - ***Schema***
     Specify HTTP or HTTPS.
     Default: http
   - ***CABI_user***
     Specify the user with administrative privilege.

Default: `superuser`

– ***CABI_password***
Specify the password for the user with administrative privilege.
Default: `superuser`

2. Run `install.bat` on Windows or `install.sh` on Linux with the `-silent` and `-uninstall` parameter and a path to the silent uninstall file. If the path or file name contains spaces, enclose the complete path in quotes.
Examples:
Windows:

```
install.bat -silent "D:\silent_uninstall_path\silent_uninstall_file.txt" -uninstall
install.bat -silent silent_uninstall_file.txt -uninstall
```

Linux:

```
./install.sh -silent "/silent_uninstall_path/silent_uninstall_file.txt" -uninstall
./install.sh -silent silent_uninstall_file.txt - uninstall
```

Troubleshooting

If the silent uninstall fails, review the install.log file that is located in the same directory as the installation script.

Configure High Availability with CA Business Intelligence

High Availability (HA) is the ability for the system to continue functioning after the failure of one or more of the servers. A part of High Availability is failover. Failover is the ability for user connections to migrate from one server to another. If there is a server failure, the user applications can continue to operate. Enabling High Availability (HA) helps achieve load balancing, failover, and scalability features. CA Business Intelligence JasperReports® Server uses an HTTP Server. As a result, the failover occurs for e-node and the load balancer or router is appropriate for HTTP Server-based application. For more information, see the CA Business Intelligence documentation.

Enable User Synchronization in a CABI HA Environment

If you have HA configured for CABI, the integration with DX NetOps Performance Management requires you to configure user synchronization.

Follow these steps:

1. Install Tomcat on the host where the load balancer installed.
2. Stop Tomcat and the Apache web server (httpd).
3. Add the following line to the *tomcat_home_directory*`/conf/server.xml` file to enable the AJP 1.3 connector on port 8009:

```
<Connector port="8009" protocol="AJP/1.3"/>
```

4. Copy the CABISync.war file to the *tomcat_home_directory*`/webapps` directory.
5. Modify the *apache_web_server_directory*`/conf/workers.properties` file as follows:

```
worker.list=loadbalancer,status,sync_listaner
worker.node1.port=8009
worker.node1.host=node1's server name/ip address
```

```
worker.node1.type=ajp13
worker.node1.lbfactor=1
worker.node2.port=8009
worker.node2.host=node2's server name/ip address
worker.node2.type=ajp13
worker.node2.lbfactor=1
worker.loadbalancer.type=lb
worker.loadbalancer.balance_workers=node1,node2
worker.loadbalancer.sticky_session=1
worker.status.type=status
worker.sync_listaner.type=ajp13
worker.sync_listaner.port=8009
worker.sync_listaner.host=localhost
```

6. Modify the *apache_web_server_directory*/conf/mod-jk.conf file as follows:

```
LoadModule jk_module modules/mod_jk.so
JkWorkersFile conf/workers.properties
JkLogFile logs/mod_jk.log
JkLogLevel info
JkLogStampFormat "[%a %b %d %H:%M:%S %Y]"
JkOptions +ForwardKeySize +ForwardURICompat -ForwardDirectories
JkRequestLogFormat "%w %V %T"
JkMount /<application-name> loadbalancer
JkMount /<application-name>/* loadbalancer
Jkmount /CABISync sync_listaner
Jkmount /CABISync/* sync_listaner
JkShmFile logs/jk.shm
<Location /jkstatus>
JkMount status
Order deny,allow
Allow from all
</Location>
```

– ***<application-name>***
  The name of CABI application, usually "`jasperserver-pro`"

8. Start Tomcat and the Apache web server.

Migrate Data to Unified Dashboards and Reporting for Infrastructure Management

The Unified Dashboards and Reporting for Infrastructure Management solution allows you to share a single CA Business Intelligence (CABI) server instance with multiple CA Agile Operations products. If you previously configured DX NetOps Performance Management to work CA Business Intelligence (CABI) server, you can migrate data from the standalone CABI to a shared CABI instance. For more information, see Unified Dashboards and Reporting for Infrastructure Management.

NOTE

The version of the standalone CABI instance must match the version of the shared CABI instance.

Follow these steps:

1. Export Content from the standalone CABI server as follows:
   a. Log in to the standalone CABI instance as a system administrator (for example, superuser).
   b. Navigate to **Manage**, **Server Settings**, and click **Export**.
   c. Specify the **Export Data File Name**.
   d. Select the content that you want to export, and then click **Export**.
2. Import the exported report content into the shared CABI server as follows:
   a. Log in to the shared CABI server as a system administrator.
   b. Navigate to **Manage**, **Server Settings**, then click **Import**.
   c. Select the file to import, then click **Import**.

Migrate CA Business Intelligence from Windows to Linux

Windows is unsupported with CA Business Intelligence 7.1.1. If you integrate with CA Business Intelligence on Windows, when you upgrade to 7.1.1, you must migrate your DX NetOps Performance Management and DX NetOps Spectrum content from the Windows CA Business Intelligence server to a Linux CA Business Intelligence server.

Follow these steps:

1. Export the content from the Windows CA Business Intelligence server:
   a. Log in to CA Business Intelligence on the Windows server as a system administrator (for example, superuser).
   b. Click **Manage**, **Server Settings**, and **Export**.
   c. Specify a file name for the export data.
   d. Select the content to export.
   e. Click **Export**
2. In NetOps Portal, delete the old Windows CA Business Intelligence data source.
3. Copy the `capm.jks` , `spectrum.jks` , `capm.properties` , and `spectrum.properties` files located in the following location from the old Windows CA Business Intelligence server to the new Linux CA Business Intelligence server:
   CABI_install_directory`/apache-tomcat/webapps/jasperserver-pro/WEB-INF/config`
4. Run the following command to install the DX NetOps Spectrum content on the Linux server:
   `java -jar spectrumConfigInstaller.jar -install`
5. Copy the `capm_reports-`*version*`.zip package` file located in the following location from the old Windows CA Business Intelligence server to the new Linux CA Business Intelligence server:
   PC_install_directory`/PerformanceCenter/cabi`
6. Run the following command to install the DX NetOps Performance Management reports on the Linux server:
   `Install.sh`
7. In NetOps Portal, add the new Linux CA Business Intelligence data source.
8. Wait for groups, users, and roles to synchronize.
9. Import the content to the Linux CA Business Intelligence server:
   a. Log in to CA Business Intelligence on the Linux server as a system administrator (for example, superuser).
   b. Click **Manage**, **Server Settings**, and **Import**.
   c. Select the content to import.
   d. Click **Import**

CA Business Intelligence Reports and Dashboards

DX NetOps Performance Management provides default reports and dashboards that you can use to view performance data.

CA Business Intelligence Reports

CA Business Intelligence (CABI) reports for DX NetOps Performance Management offer portable graphical representations of DX NetOps Performance Management performance metrics.

DX NetOps Performance Management includes the following default reports:

CA Business Intelligence Dashboards

Dashboards summarize data in a visual format. DX NetOps Performance Management includes the following dashboards:

- **Product Usage** - Provides a high-level representation of the network that is monitored by CA NetOps Portal. This dashboard provides the following information:
  - Total number of monitored devices
  - Number of monitored devices per site
  - Number of devices per type
  - Interfaces Inventory Capacity
- **Monitoring Status** - Shows the following information about DX NetOps Performance Management:
  - Status, host name, IP address, number of monitored devices, and version for data aggregators and data collectors. You click on a data collector name to view information about newly added monitored devices.
  - Number of new monitored devices for a selected data collector. You can view new devices added in the following time increments:
    - Last hour
    - Last 24 hours
    - Last 7 days
    - Last 30 days

Launch a Report or Dashboard from NetOps Portal

You can launch CA Business Intelligence reports from the NetOps Portal user interface. You must have the Administrator, Power User, or User product privilege set for the CA Business Intelligence data source and the **Drill Into Data Sources** role right. For more information, see Manage Product Access.

> **NOTE**
>
> You manage access to individual reports in CA Business Intelligence. Only users with user names in both NetOps Portal and CA Business Intelligence can access NetOps Portal reports.

Follow these steps:

1. Hover over **Reports**, and click **Report Management: CA Business Intelligence Reports**.
2. Click a report.
   The report opens in a new window.

You can also launch the CA Business Intelligence home page from CA NetOps Portal. Once you open the CABI home page, use the following instructions to open reports and dashboards.

> **NOTE**
>
> To view the version of installed Performance Management content, open the Top N report and scroll to the bottom of the input controls on the left side. The report version appears in a read-only text field.

Follow these steps:

1. Hover over **Inventory**, and click **Consoles: *Your_CABI_Data_Source.***

2. Select **View**, **Repository**.
3. In the Folders panel on the left, select **Public**, **ca**, **Performance Management**, **reports**.
4. Click **Report Name**.
5. Right-click a report (for example, **DX NetOps Performance Management Top N**) and select **Run** or **Run in New Tab**.
6. To select a report, click the appropriate row, and click **Run**.
7. Enter the required report parameters and click **Apply**.

Change the Logo

You can change the logo that appears in the CA Business Intelligence (CABI) reports for DX NetOps Performance Management.

Follow these steps:

1. Open the CA Business Intelligence JasperReports Server login page in a browser and log in with the CA Business Intelligence superuser account.
2. Select **View**, **Repository**.
3. In the Folders panel on the left, select **Public**, **ca**, **Performance Management**, **resources, images**.
4. Right-click **images**, export and back up the file.
5. In the right pane, select the **CA_Logo.png** file and click **Edit**.
6. Click **Choose File** and select your custom logo.

   NOTE

   The recommended logo size is approximately 52 x 48 pixels.
7. Click **Submit**.

Troubleshooting

If CABI reports and dashboards are missing from the NetOps Portal user interface, your reports and dashboards might exceed the maximum list limit. Update the `UniversalList.Limit` attribute, which defaults to 5000. For more information, see Cannot View More than 5000 Device Components in Inventory List.

Top N Report

Top N reports are tabular reports that list the top elements in a group that exceed or fall below the performance values that you specify. If you are troubleshooting infrastructure or planning for upgrades, Top N reports can identify the elements that need the most focus. You can also use these reports to specify a service goal against which you can compare the performance of the elements.

The following video shows how to generate multi-variable, large-scale Top N reports:

Configure the Top N Report

You can display a custom view of performance data by configuring the following options that DX NetOps Performance Management includes.

- **Time Interval**: Specify the period of time that the report reflects. You can select the following options:

  - Last Hour
  - Last 24 Hours
  - Last 7 Days
  - Last 30 Days
- **Time Zone**
- **Enable Business Hours**: Specify the days that you want DX NetOps Performance Management to use in the report. Specifying business days can limit data to high-traffic times.
- **Business Hour Start/End Time:** Specify the start and end of the business day to limit report data to that period of time.
- **Display Items As:** Select **Alias** or **Names** to determine how interfaces and devices are identified in the report. If you select **Alias**, DX NetOps Performance Management displays the alias as defined in NetOps Portal.
- **Select Group Name**: Specify the group that contains the devices that you want to view in the report.
- **Metric Family**: Select the metric family that contains the metric for the report.
- **Variable:** Select the metric that you want to view in the report.
- **Aggregation Function:** Select one of the following options:
  - **Average**
  - **Total**
  - **Maximum**
  - **Minimum**
  - **95th Percentile**
  - **96th Percentile**
  - **97th Percentile**
  - **98th Percentile**
  - **99th Percentile**
  - **Divide By Time**

    NOTE
    You can specify all non-percentile aggregation functions, or all percentile functions. You cannot mix an aggregation function, such as maximum, with a percentile.
- **Variable Filter Operation:** Specifies the variable's relation to the aggregation function. For example, you can specify Bits Per Second that are greater than or equal (>=) to the 95th percentile.
- **Top All:** Includes all data in the report.
- **Top N:** Limit the number of results to a specific number.
- **Sort Order:** Specify how the report displays the results.

Group Aggregate Trend Report

The Group Aggregate Trend (GAT) report aggregates data for a group of managed items for a specified period time. For example, you can generate a report that includes **bits per second in** for a group of interfaces in the previous month. The report provides this following data for up to four metrics:

- Number of items in the group
- The number reflects the count of active devices or components. The report does not include a device or component in the number in the following cases:

  - Polling is disabled for an item
  - The metric has no data for the selected time period
- Minimum value that is calculated for selected period and metric
- Maximum value that is calculated for selected period and metric
- 95-percentile value that is calculated for selected period and metric
- 98-percentile value that is calculated for selected period and metric

This article includes the following topics:

Configure the GAT Report

DX NetOps Performance Management includes the following options that you can configure to view aggregated group metrics.

- Time Period
- Time Granularity
- Custom Time Period
- Time Zone
- Group Name
- Metric Family
- Enable Variable *x*
  Select this option to add a metric to the report. You can specify up to 4 metrics.
- Variable *x*
  Specify the metric that the report will include.
- Rounding Pattern
- Locale
- Business Hours
  Specify the days that you want DX NetOps Performance Management to use in the report. Specifying business days can limit data to high-traffic times.

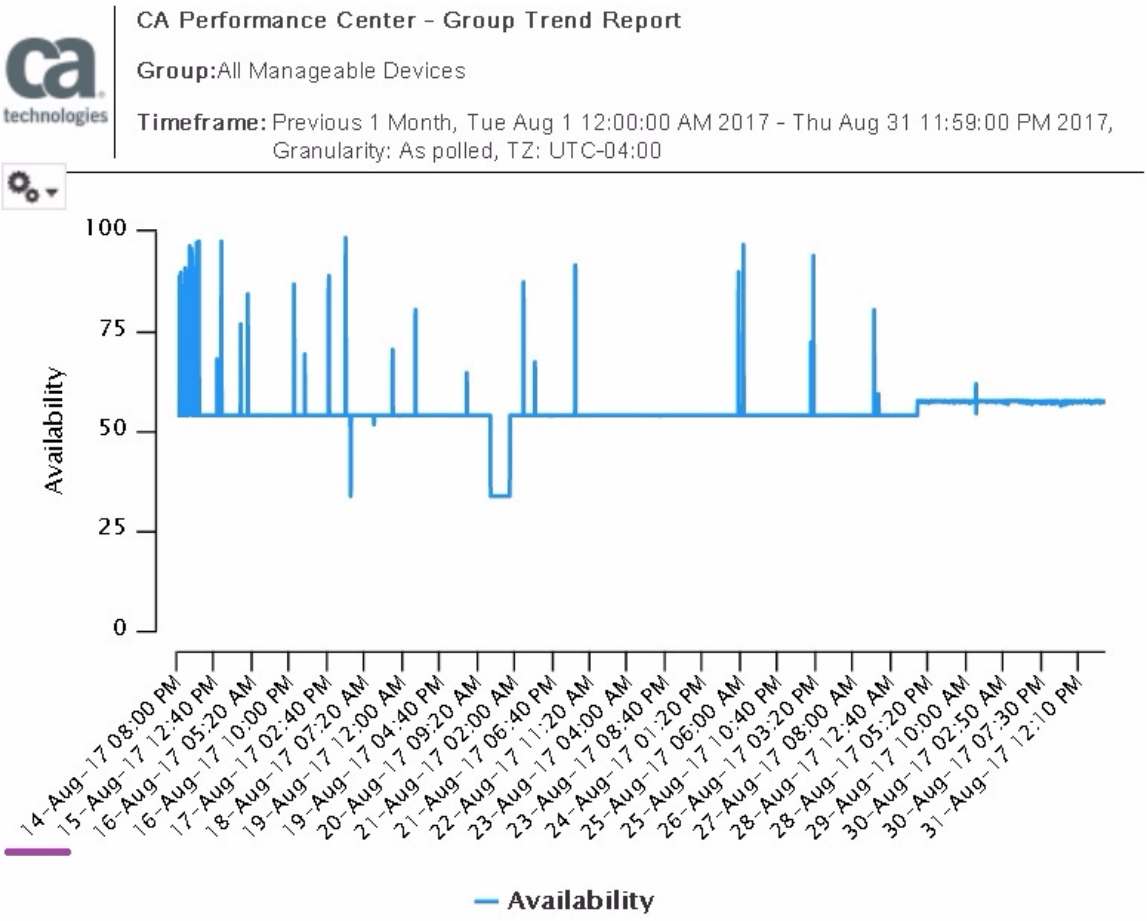
Understand the Time Granularity Selector

Sometimes, when a user selects a time period, the report shows data for only part of that period. For example, a report with the following settings is run on September 28:

- **Time Period:** Previous Month
- **Time Granularity:** As Polled

Instead of the report starting on August 1 as expected, it starts on August 14.

Figure 66: Group_Trend_Report_1



The report displays partial results for the time period because of the data retention settings.

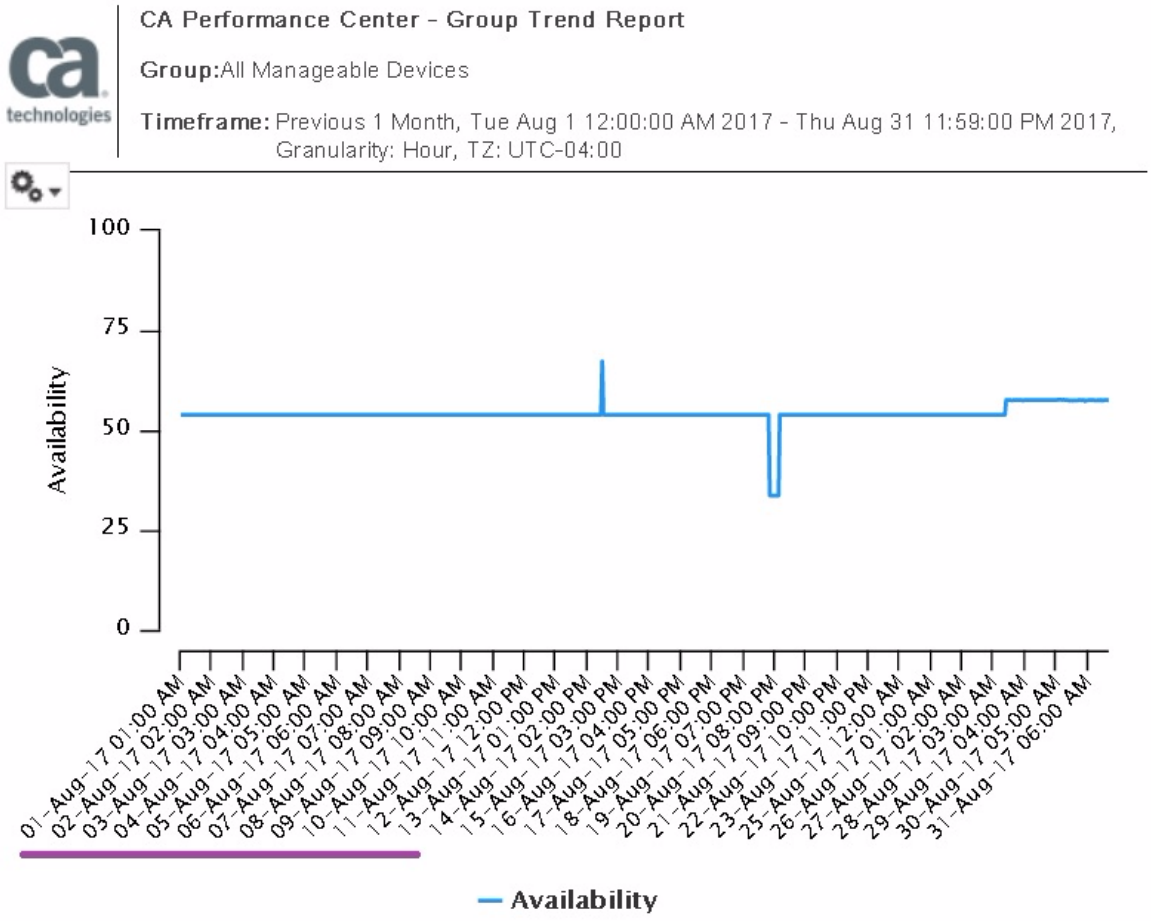
DX NetOps Performance Management uses the following default data retention settings:

- Polled data: 45 days
- Hourly rollup data: 90 days
- Daily rollup data: 365 days
- Weekly rollup data: 730 days

For more information, see Configure Data Retention Rates.
In the previous example, the report was configured to include polled data for August. However, because the report was run September 28, DX NetOps Performance Management only retained data from August 14 (September 28 - 45 days is August 14).

To see data for the entire month of August, change Time Granularity to Hourly. Because hourly data is retained for 90 days, DX NetOps Performance Management can display data for all of August.

Figure 67: Group_Trend_Report_2



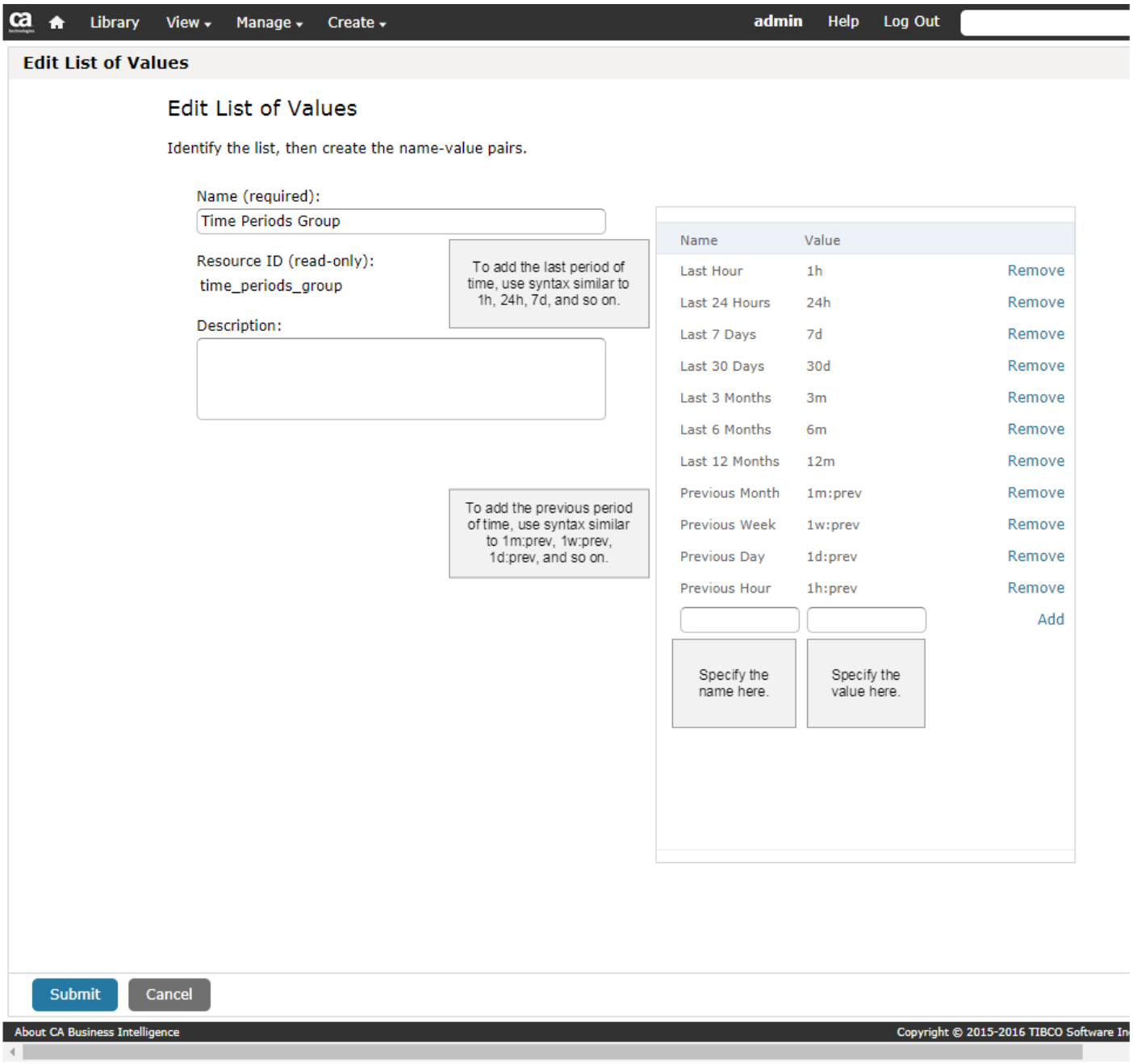
Configure a Custom Time Period

If the default time periods do not meet your needs, you can add a new Time Period .

Follow these steps:

1. Log in to the CABI Reports Server as a user with edit permissions.
2. Go to `Public/ca/Performance Management/inputcontrols/list of values`.
3. Edit the **Time Periods Group**.
4. Add any interval that the OpenAPI supports.
5. Specify the following information for the new time period:
   - **Name**
     Specify the name that appears in the Time Period options list.
   - **Value**
     Specify the custom timeframe.

Figure 68: List_of_Values



Troubleshoot the GAT Report

Report Timeout

Sometimes, the GAT report fails if you specify a long Time Period (month or more), *and* **As Polled** as the Time Granularity setting. The report can fail because of a timeout on the Data Aggregator. The default query timeout setting on the Data Aggregator is 30 seconds.

To resolve this issue, choose *one* of the following options:

- Set the Time Granularity to Hourly and rerun the GAT report.
- Increase the Data Aggregator timeout setting (defaultQueryTimeoutSecs - Default: 30)
  You can change the value of the defaultQueryTimoutSecs parameter during query execution (for current query only) or permanently in the following location:`/opt/IMDataAggregator/apache-karaf-<vers>/etc/com.ca.im.odata.beans.ODataLimiters.cfg`
  For more information, see Configure OpenAPI Defaults and Limits.

  NOTE

  In DX NetOps Performance Management tests, CA Technologies found that the appropriate Data Aggregator timeout setting for 10,000 interfaces is `defaultQueryTimeoutSecs=60` . However, you may need to increase the timeout setting if there are more items in the group the report is run against.

Situations to Watch Report

The Situations to Watch report predicts the health of your environment by providing the following information:

- When a given metric is expected to cross a specified threshold
- How quickly the metric will cross the threshold
- Anticipated future value based on a time period you choose (for example, 30, 60, and 90 days in the future)

You can use the Situations to Watch report to analyze resource allocation and prevent issues before they impact performance.

Configure the Situations to Watch Report

DX NetOps Performance Management includes a number of options that you can configure to see a custom view of metric thresholds.

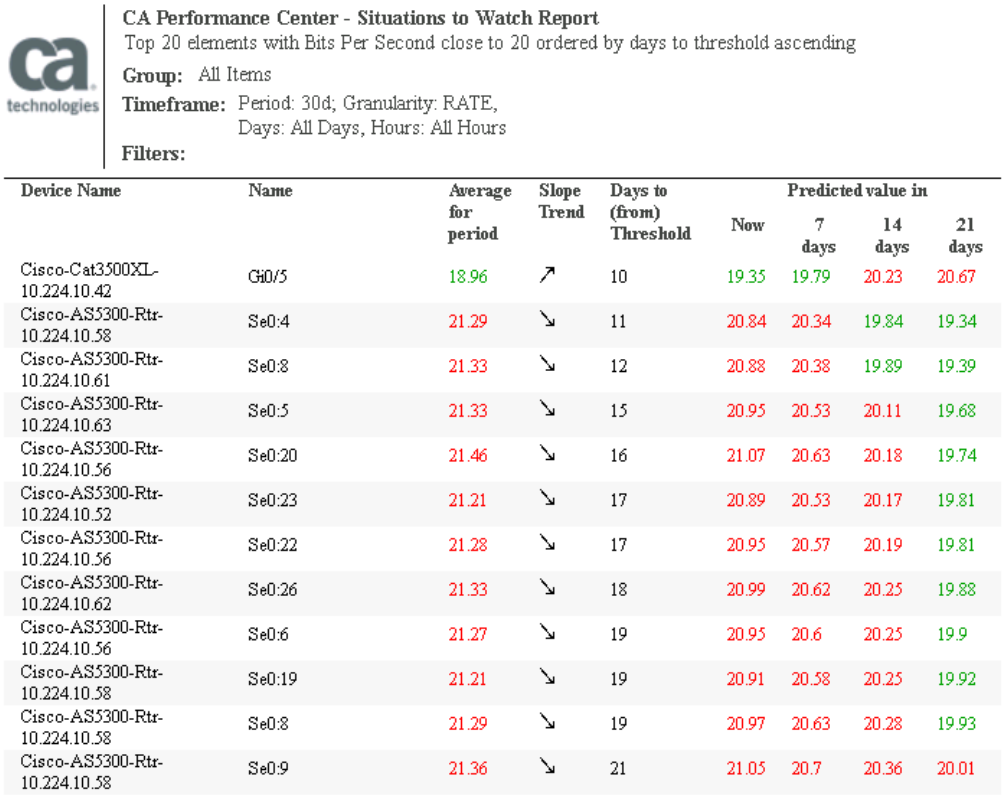
- **Time Period**: Specify the period time that DX NetOps Performance Management uses as the historical baseline when calculating a future metric value. For example, specify 30 days to use data from the previous month to determine the future behavior of the metric.
- **Time Granularity**: Specify how frequently DX NetOps Performance Managementcollects data for the report. You can select the following options:
  - As polled
  - Hour
  - Day
  - Week
- **Enable Business Hours**: Specify the days that you want DX NetOps Performance Management to use in the report. Specifying business days can limit data to high-traffic times.
- **Business Hour Start/End Time:** Specify the start and end of the business day to limit report data to that period of time.
- **Select Group Name**: Specify the group that contains the devices that you want to view in the report.
- **Metric Family**: Select the metric family that contains the metric for the report. The metric family determines the values in the Metric field.
- **Metric**: Select the metric for the report.
- **Threshold**: Specify the threshold for the report. For example, to report on when memory usage will meet or exceed 80%, specify 80.
- **Days to Threshold Filter/ Days to Threshold Value**: Specify the scope of the data in the report by adding a filter and value:

- Filter: Specify one of the following operators: <, =, >
- Value: Specify the number of day

For example, if you specify < 30, CA Advanced Authentication shows all inventory that will reach the threshold in less than 30 days.

- **Include Historical Situations**: When you select this option, CA Advanced Authentication displays items that have already exceeded the threshold. The number of days since the item exceeded the metric is displayed in parentheses. For example, (9) indicates that the item exceeded the threshold 9 days ago.
- **Show with Trend**: Displays an arrow in the Slope Trend column of the report. The arrow indicates whether the metric value is increasing or decreasing.

Figure 69: Sits_to_watch_report



CA Performance Center - Situations to Watch Report
Top 20 elements with Bits Per Second close to 20 ordered by days to threshold ascending
Group: All Items
Timeframe: Period: 30d; Granularity: RATE, Days: All Days, Hours: All Hours
Filters:

Device Name	Name	Average for period	Slope Trend	Days to (from) Threshold	Predicted value in			
					Now	7 days	14 days	21 days
Cisco-Cat3500XL-10.224.10.42	Gi0/5	18.96	↗	10	19.35	19.79	20.23	20.67
Cisco-AS5300-Rtr-10.224.10.58	Se0:4	21.29	↘	11	20.84	20.34	19.84	19.34
Cisco-AS5300-Rtr-10.224.10.61	Se0:8	21.33	↘	12	20.88	20.38	19.89	19.39
Cisco-AS5300-Rtr-10.224.10.63	Se0:5	21.33	↘	15	20.95	20.53	20.11	19.68
Cisco-AS5300-Rtr-10.224.10.56	Se0:20	21.46	↘	16	21.07	20.63	20.18	19.74
Cisco-AS5300-Rtr-10.224.10.52	Se0:23	21.21	↘	17	20.89	20.53	20.17	19.81
Cisco-AS5300-Rtr-10.224.10.56	Se0:22	21.28	↘	17	20.95	20.57	20.19	19.81
Cisco-AS5300-Rtr-10.224.10.62	Se0:26	21.33	↘	18	20.99	20.62	20.25	19.88
Cisco-AS5300-Rtr-10.224.10.56	Se0:6	21.27	↘	19	20.95	20.6	20.25	19.9
Cisco-AS5300-Rtr-10.224.10.58	Se0:19	21.21	↘	19	20.91	20.58	20.25	19.92
Cisco-AS5300-Rtr-10.224.10.58	Se0:8	21.29	↘	19	20.97	20.63	20.28	19.93
Cisco-AS5300-Rtr-10.224.10.58	Se0:9	21.36	↘	21	21.05	20.7	20.36	20.01

- **Top All**: Includes all data.
- **Delta in Days:** Specify an increment in days that DX NetOps Performance Management uses to predict 3 future metric values. If you specify 30, DX NetOps Performance Management provides predicted metric values for 30, 60, and 90 days in the future.

SystemEDGE System Response Path Test Metrics

DX NetOps Performance Management discovers ESX hosts and virtual machines in the following ways:

- Through ICMP
- Through SNMP, if the servers have an SNMP agent deployed
- Through discovery of a server running SystemEDGE with the VCAIM

For more information, see Discovery and Polling in VMware Environments.

Response path tests help you see the overall status for devices that support service level agreements (SLA) using various protocols. This information lets you gain a broad sense of overall SLA performance for the enterprise or a specific device.

For example, as a network infrastructure manager, you can identify which protocols are experiencing the highest latency and slowest response times from the dashboard. You can then look at the device-level views to determine which devices are experiencing the problems. This information can help determine if increased activity, a device that is down, or something else caused an issue.

Response path test metrics are available specifically for SystemEDGE.

The SystemEDGE Response Path vendor certification includes the Service Availability Response Path metric family, which provides the following metrics:

- Attempts
- Avg. DNS Lookup Time
- Avg. Response Time
- Avg. TCP Connect Time
- Avg. Transaction Time
- Bytes Received Total
- Descriptions
- Failed Transactions
- Indexes
- Max. DNS Lookup Time
- Maximum Response
- Max. TCP Connect Time
- Min. Transaction Time
- Names
- Percent Failed Attempts
- Percent Successful Attempts
- Response/Limit
- Response Throughput
- Successful Attempts
- Successful Transactions
- Total Errors

Troubleshooting

Use the troubleshooting section to diagnose and resolve issues with DX NetOps Performance Management. Each troubleshooting page in this section contains one or more symptoms, and at least one solution to resolve the issue.

Insufficient Permissions During Installation

Symptom:

After you specify an installation path the following error appears:

```
Error: Insufficient permissions for installation path: Folder_PathPath must have
 executable permission for 'other'.Exiting the installer...
```

Solution:

Run the following command to ensure all child and parent directories have the necessary executable permission:

```
chmod 755 "Folder_Path"
```

Installation Stopped Abruptly

In CA Virtual Network Assurance 3.6.6, when the installation stops abruptly for any reason, and you start the installation again, the installer does not prompt you to install the product on a custom directory. Perform the following steps to resolve the issue and install the product on a custom directory.

Follow these steps:

1. On the server where the installation stopped, navigate to the `/etc` directory.
2. Delete the VNA.cfg file using the `rm -rf VNA.cfg` command.
3. Follow the steps in the Installing section and complete the installation.

Enable NetOps Portal Logging

If you encounter an issue with a specific view in the NetOps Portal user interface, you can easily enable logging on that view. You can use this method to provide the necessary details to CA Support.

Follow these steps:

1. Log in as a user who has the Generate URLs from Views role right.
2. Open the dashboard that contains the problematic view.
3. Click the **Edit** (gear) icon on the view, and select **Generate URL**.
   The Generate URL dialog opens.
4. Enable **Detailed View Logging**.
5. Click **Preview.**
6. Take a screenshot of the preview for CA Support.
7. Run CARE on the NetOps Portal host and the Data Aggregator host.
8. Provide the archive files from CARE and the screenshot to CA Support.

Enable Data Collector Logging

If you encounter missing data, detailed poll logging is available from the Data Collector debug tool. When enabled for an IP address, the tool generates logs on the Data Collector. Before you download the logs, you should wait at least two poll cycles (ten minutes) after enabling logging to collect sufficient data.

The logs record the following details of every poll for the specified IP address:

- The requests that are sent to the device
- The responses that are received from the device
- The delta processing the Data Collector performs on the responses
- The expression evaluation the Data Collector performs on the responses

You cannot enable detailed poll logging for more than one IP address at a time. If a second IP address is enabled, detailed poll logging for the previous IP address is disabled. The logs for the previous IP address are discarded.

Detailed poll logging stores a maximum of 60 MB of logged messages by default. If the limit is reached, the oldest entries are discarded to make room for the newer messages. You can configure this limit with the cache setting.

Logging can be filtered by Poll Group ID, or by Component Item ID. Filtering restricts the logging to a specific Vendor Certification or a specific component.

> **WARNING**
>
> Detailed poll logging is stored in the runtime memory of the Data Collector. If the Data Collector is shut down, the log is lost. If the Data Collector restarts, the log resets.

Follow these steps:

1. Go to the following location:
   DA_host:*port*/dcdebug/enabledebug.html
2. Specify the IP address of the polled device.
3. Select an IP domain.
4. (Optional) Specify Filtering and Cache settings.
5. Select **Enable Debug Logging**.
6. Click **Enable Debug Logging**.

Download Data Collector Logs

If you encounter missing data, you can use the Data Collector debug tool to download the relevant logs. Before you download the logs, you should wait at least two poll cycles (ten minutes) after enabling logging to collect sufficient data. You can use this method to provide the necessary details to CA Support. After you download the logs, you should disable detailed poll logging.

Follow these steps:

1. Go to the following location:
   DA_host:*port*/dcdebug/searchdebug.html
2. Specify the IP address of the polled device.
3. Select an IP domain.
4. Select **Download all logs for IP**.
5. Click **Download As Zip**.

Disable Data Collector Logging

After you download the Data Collector logs, you should disable detailed poll logging.

Follow these steps:

1. Go to the following location:
   DA_host:*port*/dcdebug/enabledebug.html
2. Specify the IP address of the polled device.
3. Select an IP domain.
4. Select **Disable Debug Logging**.
5. Click **Disable Debug Logging**.

SNMP Querying Tools

SNMP querying tools are included with DX NetOps Performance Management. The `sapwalk2` utility and the `sappoll` utility are available in the *DC_Install_Directory*`/scripts/` directory. The `sapwalk2` command line tool gathers an SNMP snapshot of network devices. Internal CA engineers can use this snapshot to reproduce issues and verify SNMP values on devices. The `sappoll` utility retrieves SNMP data for a timeframe.

CA Support finds the following `sapwalk2` command helpful:

```
sapwalk2 -i ip_address -v snmp_version -s starting_oid -c community_name -
xv bridge_table_oid -o output_file.walk
```

Example:

```
sapwalk2 -i 10.253.190.15 -v v2c -s 1.3.6.1 -c public -xv 1.3.6.1.2.1.17 -o
 10.253.190.15.walk
```

CA Support finds the following `sappoll` command helpful:

```
sappoll - i ip_address -v snmp_version -c community_name -p snmp_port -r retries -d
 timeout -f file_containing_oids -n number_of_polls -t poll_interval
```

Example:

```
sappoll -i 138.42.110.45 -v v2c -c xxxxx -p 161 -r 3 -d 3000 -f /tmp/OidList -n 5 -t 2
```

The file containing OIDs (for example, `/tmp/OidList`), should include the OIDs to poll in the following format:

```
I instance_oid
N node_oid
```

Example:

```
I 1.3.6.1.2.1.1.1
```

Enable the ActiveMQ Admin Console for the Data Aggregator or Data Collector

Generally, the ActiveMQ admin console should not be available on the network. However, if you absolutely need the console to address a problem, you can enable it for the Data Aggregator or Data Collector.

Follow these steps:

1. Go to one of the following files:
   - **Data Aggregator**
     DA_Install_Directory/broker/apache-activemq-*version*/conf/activemq.xml
   - **Data Collector**
     DC_Install_Directory/broker/apache-activemq-*version*/conf/activemq.xml
2. Uncomment
   `<import resource="jetty.xml"/>`
   .
3. (Optional) To update user access, edit the
   `jetty-realm.properties`
   .
4. (Optional) To encrypt the user passwords, run one of the following commands:
   - **Data Aggregator**
     DA_Install_Directory/broker/apache-activemq-*version*/lib/web run java -cp jetty-all-9.2.22.v20170606.jar org.eclipse.jetty.util.security.Password *passwordpassword*
   - **Data Collector**
     DC_Install_Directory/broker/apache-activemq-*version*/lib/web run java -cp jetty-all-9.2.22.v20170606.jar org.eclipse.jetty.util.security.Password *passwordpassword*
5. Shut down the ActiveMQ broker on each Data Collector:
   `service activemq stop`
6. Shut down the ActiveMQ broker on the Data Aggregator:
   `service activemq stop`
7. Start the ActiveMQ broker on the Data Aggregator:
   `service activemq start`
   If you do not, the Data Aggregator starts the broker automatically.
8. The Data Collectors automatically restart the ActiveMQ brokers. Use the following command to restart the brokers manually:

```
service activemq start
```

Access Denied to MySQL Utilities

Symptom:

Access is denied when you try to access a MySQL utility from the command line.

Example:

If you try to run `mysqldump` without a password, the following error returns:

```
mysqldump: Got error: 1045: Access denied for user 'root'@'localhost' (using password:
 NO) when trying to connect
```

Solution:

As a step toward enhanced security, the MySQL utilities require a MySQL password. In previous releases, if you were running the utilities as the root user, the password had a default. We strongly recommend you upgrade or at least change the default password with the `mysqladmin` utility.

Example:

To change the default 'root' password from the default ('*`default`*' in the following example) to your custom password ('*`newpassword`*' in the following example), use the following command:

```
mysqladmin -uroot -pdefault password newpassword
```

Automatic Rediscovery Does Not Run After Updating Vendor Group Priority

Symptom:

DX NetOps Performance Management did not run rediscovery after I updated the vendor certification priorities and updated the order of vendor certifications.

Solution:

Updating the order of vendor certifications and the PriorityGroup tag in the same REST operation can disrupt automatic rediscovery. Manually trigger rediscovery for the affected metric family.

Follow these steps:

1. Mouseover **Administration**, and click the Data Aggregator data source.
2. Click **Monitoring Configuration, Metric Families**.
3. Select the affected metric family.
4. Click **Update Metric Family**.
   Discovery runs for all devices in the metric family.

Browser Shows Error when Logging In

Symptom:

When I enter my password on the Login page, I am redirected to an error page in the web browser.

Solution:

This symptom does not indicate that you entered incorrect SAML credentials. Instead, the browser error (such as 401 or 500) indicates that Single Sign-On redirected the browser to the login URL, but the Identity Provider (IdP) server is down.

Follow these steps to correct the issue:

- Verify that the IdP server is running.
- Test the network connection between the CA NetOps Portal server and the IdP server.

Cannot Create a Vendor Certification

Symptom:

I cannot create a vendor certification and I receive an error message.

Solution:

Open the karaf log files in the Data Aggregator installation directory, and follow these steps:

1. Look for the MIB name string or the name of the metric family that you selected.
2. Review the stack trace of the exception to find the CertManagerException and the reason for the error. The reason for the error follows the exception.
   Example: The expression parser did not expect the token after ++, as shown:

```
Caused by: com.ca.im.dm.certmgr.interfaces.CertManagerException: Tech Cert: {http://
im.ca.com/
normalizer}NormalizedCPUInfo, Unable to compile expression: [Error: expected end of statement

[Near : {... stemID ++ extremeSystemBoardID ....}]
```

3. Fix the error based on the reason that is provided. Verify that the following requirements are met:
   - The expression group does not contain a mix of scalar and table entries.
   - Expressions contain valid syntax.
   - At least one expression is defined for a metric family variable.
   - At least two metric family variables are defined. Names and Indexes are required (except for scalar-only metrics).
   - The vendor certification variable used in the expression is from the chosen MIB table (valid in the user interface).

Cannot Remove a Custom Vendor Certification

Symptom:

I want to stop using a custom vendor certification. I cannot find a method to remove or deactivate the certification.

Solution:

Removing a custom vendor certification is not supported. Move the vendor certification to the bottom of the vendor certification priority list.

Cannot Find the Data Aggregator RIB Document

Symptom:

I cannot locate the Data Aggregator RIB document.

Solution:

Follow these steps:

1. Verify that Data Aggregator has been successfully added to CA NetOps Portal. For more information, see Manage Data Sources.
2. Verify that the RIB web service for Data Aggregator is running.
3. Verify that the RIB web service for Data Aggregator is publishing the RIB document:

a. Use the Data Source's RIB web service to request the list of available RIB documents.
Example:

```
http://da_host:8581/rib/doclist
```

4. If you know the document ID, check the document:

```
http://da_host:8581/rib/doc/docId
```

Example:

```
http://da_host:8581/rib/doc/CA.IM.DA.NormalizedPortInfo
```

'Cannot Find Valid Certification Path' Exception After Enabling SSL

Symptom:

After you enable SSL on CA NetOps Portal, CA Business Intelligence reports fail to open or start. Errors that resemble the following appear in the long stack trace:

```
2017-11-20 14:38:55,410 ERROR AsyncJasperPrintAccessor,pool-6-thread-1:321 - Error
 during report execution
net.sf.jasperreports.engine.JRException: Can't get response from capm
at com.ca.capm.jasper.datasource.util.CAPMConnector.getCAPMResponse(CAPMConnector.java:46)
...

Caused by: sun.security.validator.ValidatorException: PKIX path building failed:
 sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid
 certification path to requested target
```

Solution:

For reporting to work in an SSL-Enabled NetOps Portal instance, you must add the appropriate certificates to the CA Business Intelligence trust store. The appropriate certificates are the root or intermediate certificates comprising the chain of trust for the NetOps Portal public certificate.

The NetOps Portal public certificate (capc_cert.cer), and the root and intermediate certificates must exist in the cacerts keystore.

For more information, see the section Add a Certificate for SSL-Enabled NetOps Portal in Install CA Business Intelligence Reports and Dashboards.

Cannot Remove a Metric Family

Symptom:

I want to stop using a custom metric family. I cannot find a method to remove the metric family from my instance of DX NetOps Performance Management.

Solution:

Removing a custom metric family is not supported. To stop using the custom metric family, remove it from all monitoring profiles. For more information, see Configure Monitoring Profiles.

Cannot View More than 5000 Device Components in Inventory List

Symptom:

I have more than 5000 devices registered in NetOps Portal. However, the list of device components in the inventory or search results shows only 5000 items.

Resolution:

By default, NetOps Portal limits the number of devices in the Inventory to 5000. You can manually change this default value.

Follow these steps:

1. Go to the following URL:

   ```
   http://PC_ADDRESS:8181/pc/center/admin/debug
   ```

   PC_Address The hostname of your NetOps Portal installation.
2. Log in as an administrative user.
3. Select **Global Attributes**.
4. Specify a value for the UniversalList.Limit attribute.
5. Click **Update**.
   The maximum number of registered devices that are visible in the inventory is equal to the value of UniversalList.Limit. You do not need to restart any services for the changes to take effect.

   WARNING

   Increasing the default device limit of 5000 can cause performance issues in NetOps Portal.

Data Aggregator Disk Space is Decreasing

Symptom: The available disk space on the Data Aggregator host is decreasing unexpectedly.

Solution: Sometimes, an error causes orphaned Rollup messages to build up in ActiveMQ. These message files are large. Verify the presence of the messages and purge them if necessary.

Follow these steps:

1. Navigate to the following directory on the Data Aggregator host:
   installation_directory/broker/apache-activemq-*vers*/data/kahhadb

2. Search for "not removing data file ":
   ```
   grep "not removing data file"  db-*.log | more
   ```

   Compare the return to the following example:
   ```
   2015-10-13 09:30:31,411 [eckpoint Worker] TRACE MessageDatabase                - not
   removing data file:
   ```

   This return indicates that the ActiveMQ broker has orphaned messages that are blocking the queue.

If the return in your system is similar, continue to the next step.

3. Navigate to the following URL to access the ActiveMQ broker:
   `http://DA_host:8161/admin/queues.jsp`
   ActiveMQ might require login with the Admin account.
   Default Username: `admin`
   Default Password: `admin`
4. Locate ActiveMQ.DLQ. If the pending message count is greater than 0, click **purge**.
   ActiveMQ clears the orphaned messages.

Data Aggregator Fails to Synchronize

Symptom:

When I try to synchronize Data Aggregator with NetOps Portal, I see a 'Synchronization failure' message in the Status column.

Solution:

Data Aggregator could not handle the data that is sent to it during synchronization. Review the Device Manager application log file, called DMService.log. This file appears in the CA/PerformanceCenter/DM/logs directory. The log entry shows a general SOAP exception if Data Aggregator is unable to handle data that was received from CA NetOps Portal during synchronization.

Look for an exception and stack trace within the following phases of synchronization:

- Pull
- Global Sync
- Bind (only executes when initially synchronizing with a data source)
- Push

Contact CA Technical Support with this information.

Data Aggregator or Data Collector Does Not Initialize

Symptom:

On certain systems, the Data Aggregator or Data Collector processes start successfully, but the application is never initialized. The status shows that the process is running. This problem typically occurs on systems with underpowered CPUs. This problem only occurs when the Data Aggregator or Data Collector are upgraded, or restarted after clearing out the karaf data directory.

Solution

To verify that this issue occurred, complete the following verification steps:

1. Look at the following log file: `apache-karaf-<vers>/data/log/karaf.log`
   This issue produces an error message in the log. The following message is an example of this error:

```
2016-01-12 02:32:46,303 | WARN  | Event Dispatcher | AetherBasedResolver
  | mvn.internal.AetherBasedResolver  583 | 3 - org.ops4j.pax.logging.pax-
logging-api - 1.8.3 |  | Error resolving artifactcom.ca.im:data-
mgmt.provision:xml:features:2.7.0-RELEASE-137:Could not find artifact com.ca.im:data-
mgmt.provision:xml:features:2.7.0-RELEASE-137 in central (http://repo1.maven.org/
maven2/)
```

```
shaded.org.eclipse.aether.resolution.ArtifactResolutionException: Could not find
 artifact com.ca.im:data-mgmt.provision:xml:features:2.7.0-RELEASE-137 in central
 (http://repo1.maven.org/maven2/)
    at
 shaded.org.eclipse.aether.internal.impl.DefaultArtifactResolver.resolve(DefaultArtifactResol
    at
 shaded.org.eclipse.aether.internal.impl.DefaultArtifactResolver.resolveArtifacts(DefaultArti
…
…
2016-01-12 02:32:46,327 | WARN  | Event Dispatcher | FeaturesServiceImpl
 | res.internal.FeaturesServiceImpl 1367 | 7 - org.apache.karaf.features.core - 2.4.3
 |  | Unable to add features repository mvn:com.ca.im/data-mgmt.provision/2.7.0-
RELEASE-137/xml/features at startup
java.io.IOException: Error resolving artifact com.ca.im:data-
mgmt.provision:xml:features:2.7.0-RELEASE-137: Could not find artifact
 com.ca.im:data-mgmt.provision:xml:features:2.7.0-RELEASE-137 in central (http://
repo1.maven.org/maven2/)
    at
 org.ops4j.pax.url.mvn.internal.AetherBasedResolver.resolve(AetherBasedResolver.java:584)
    at
 org.ops4j.pax.url.mvn.internal.AetherBasedResolver.resolve(AetherBasedResolver.java:528)
    at
 org.ops4j.pax.url.mvn.internal.AetherBasedResolver.resolve(AetherBasedResolver.java:506)
    at
 org.ops4j.pax.url.mvn.internal.AetherBasedResolver.resolve(AetherBasedResolver.java:481)
    at org.ops4j.pax.url.mvn.internal.Connection.getInputStream(Connection.java:123)
```

2. Confirm that the Data Aggregator or Data Collector features are not resolved in karaf
   a. Log in to the karaf console on the relevant host:
      - On the Data Aggregator:

        ```
        ssh -p8501 karaf@localhost
        ```

      - On the Data Collector:

        ```
        ssh -p8601 karaf@localhost
        ```

   b. List the features:

      ```
      features:list | grep data
      ```

      When this issue occurs, this list does not include the following features:
      - `data-aggregator`
      - `data-collection-manager`

If this issue occurs, apply the fix for the relevant service:

Data Aggregator

1. Identify the features repository URL to be added from the following file: `INSTALL_ROOT/apache-karaf-2.4.3/etc/org.apache.karaf.features.cfg`

   ```
   [root@sapv1 ~]# grep --color provision /opt/IMDataAggregator/apache-karaf-2.4.3/etc/
   org.apache.karaf.features.cfg
   featuresRepositories = mvn:org.apache.karaf.assemblies.features/standard/2.4.3/
   xml/features,mvn:org.apache.karaf.assemblies.features/spring/2.4.3/xml/
   features,mvn:org.apache.karaf.assemblies.features/enterprise/2.4.3/xml/
   features,mvn:com.ca.im/data-mgmt.provision/2.7.0-RELEASE-133/xml/features
   #featuresRepositories=mvn:com.ca.im/data-mgmt.provision/2.7.0-RELEASE-133/xml/
   features
   ```

   Select the content from this file according to the error message in the karaf.log file. The number after **RELEASE-** can be different.
2. Connect through karaf:

   ```
   ssh -p8501 karaf@localhost
   ```

   The password is **karaf**.
3. Manually add the features:
   –
   ```
   features:addurl mvn:com.ca.im/data-mgmt.provision/2.7.0-RELEASE-133/xml/features
   ```
   –
   ```
   features:install data-aggregator
   ```
4. Verify the following log does not include a repeat of the error message: `apache-karaf-2.4.3/data/log/karaf.log`
   During normal operation, the log shows the various bundles in the application initializing.
5. Verify that the Data Aggregator has initialized. For example, verify that the Data Aggregator REST page loads from the following URL:
   `da_host:8581/rest/`
6. Restart each data collector. Do not apply the Data Collector fix unless the error message also appears in the karaf log on the Data Collector.

Data Collector

1. Identify the features repository URL to be added from the following file: `INSTALL_ROOT/apache-karaf-2.4.3/etc/org.apache.karaf.features.cfg`

   ```
   [root@sapv1 ~]# grep --color provision /opt/IMDataCollector/apache-karaf-2.4.3/etc/
   org.apache.karaf.features.cfg
   featuresRepositories = mvn:org.apache.karaf.assemblies.features/standard/2.4.3/
   xml/features,mvn:org.apache.karaf.assemblies.features/spring/2.4.3/xml/
   features,mvn:org.apache.karaf.assemblies.features/enterprise/2.4.3/xml/
   features,mvn:com.ca.im/data-mgmt.provision/2.7.0-RELEASE-133/xml/features
   ```

```
#featuresRepositories=mvn:com.ca.im/data-mgmt.provision/2.7.0-RELEASE-133/xml/
features
```

Select the content from this file according to the error message in the karaf.log file. The number after **RELEASE-** can be different.

2. Connect through karaf:

```
ssh -p8601 karaf@localhost
```

The password is **karaf**.

3. Manually add the features:
   –
   ```
   features:addurl mvn:com.ca.im/data-mgmt.provision/2.7.0-RELEASE-133/xml/features
   ```
   –
   ```
   features:install data-collection-manager
   ```
4. Verify the following log does not include a repeat of the error message: `apache-karaf-2.4.3/data/log/karaf.log`
   During normal operation, the log shows the various bundles in the application initializing.
5. Verify that the Data Collector has initialized. For example, in NetOps Portal, go to **Administration**, **Data Collector List**, and verify that the status is **Collecting Data**.

Data Collector Dropped Polling Event Message

Symptom:

A "Data Collector Dropped Polling" event appeared in my events list.

Solution:

Clock drift can sometimes cause polling to stop on some components and devices. When this occurs, restart the Data Collector.

For more information, see Restart the Data Collector.

Data Collector Installs But Does Not Appear in the Data Collector List Menu

Symptom:

I installed Data Collector successfully, but Data Collector does not appear in the Data Collector List menu.

Solution:

Do the following steps:

1. Review the *DC_Install_Directory*`/apache-karaf-`*version*`/shutdown.log` file to ensure that the Data Collector was not shut down automatically. Data Collector is shut down automatically when you specify the Data Aggregator host, tenant, or IP domain incorrectly when you install Data Collector. The shutdown.log file provides error information as to why Data Collector was shut down. The Data Collector shuts down for one of the following reasons:
   – The Data Aggregator host information, tenant, or IP domain that was specified during the Data Collector installation were incorrect:

- If you specified the Data Aggregator host information incorrectly, uninstall and reinstall the Data Collector.
- If you specified the tenant incorrectly, uninstall and reinstall Data Collector.
- If you specified the IP domain incorrectly, uninstall and reinstall Data Collector.

– Contact with Data Aggregator could not be established.

2. Type the following command to ensure that an established connection to Data Aggregator exists:

```
netstat  - a | grep 61616
```

3. If a connection to Data Aggregator does not exist, do the following steps:
   a. View the *DC_Install_Directory*/broker/apache-karaf-*version*/conf/activemq.xml file on the Data Collector host. This file contains the hostname or IP address of the Data Aggregator host that you specified when you installed Data Collector.
   b. Search for the “networkConnector” section of the activemq.xml file. This section should contain a line as follows:

```
<networkConnector name="manager"

       uri="static:(tcp://test:61616)"

       duplex="true"

          suppressDuplicateTopicSubscriptions="false"/>
```

   Ensure that the Data Aggregator hostname that is specified in the "networkConnector" section is correct and resolves through DNS or /etc/hosts entries. Data Collector cannot communicate with Data Aggregator if you entered the Data Aggregator hostname incorrectly during the Data Collector installation.
   c. Type the following command help ensure that the connection opens successfully when you open a telnet connection to the Data Aggregator host on port 61616:

```
telnet dahostname 61616
```

   This command confirms that Data Aggregator is listening in on that port.
   d. If the telnet connection does not open successfully, the reasons could be as follows:
   - Data Aggregator is not running. Ensure that Data Aggregator is running. Open a console and type the following command:

```
service dadaemon status
```

NOTE

For RHEL 7.x and OL, service invokes systemctl . You can use systemctl instead.

   e. If Data Aggregator is not running, start Data Aggregator. Log on to the Data Aggregator host computer as the root user or a sudo user with access to a limited set of commands. Do one of the following steps:
   - Start the Data Aggregator services:

```
service dadaemon start
```

   - (Fault tolerant environment) Run one the following commands to enable the fault tolerant Data Aggregator so that it can start when necessary:
     - **RHEL 6.x:**

```
service dadaemon activate
```

- **RHEL 7.x, SLES, or OL:**

```
DA_Install_Directory/scripts/dadaemon activate
```

The request to initiate the connection is not making it from Data Collector to Data Aggregator successfully. Ensure that the port that is specified in the “networkConnector” section of the broxer.xml file is open for incoming connections on Data Aggregator. Be sure that there are no firewall rules preventing this connection.

Data Is Missing from Views

Symptom:

Some of the table columns in the interface views are empty. For example, interface and device names, interface speeds, and utilization data are missing from views.

Solution:

Some data sources do not support authentication passwords or privacy passwords that are below the minimum length.

SNMP profiles that use the SNMPv3 format let you enable authentication and privacy options. When you create a valid SNMPv3 profile, specify an authentication password that is eight characters or more in length. These profiles may not be successful in communicating with devices. In this case, SNMP data are missing for the affected interfaces.

Similarly, blank passwords are not supported for SNMP v3 profiles with MD5 or SHA as the Authentication Protocol.

Data Source Registration Fails

Symptom:

I attempt to add a new data source, but the registration fails. The following message appears: 'Create Data Source Failed: Data source communication failure.'

Solution 1:

This message indicates that the data source is unreachable. Do the following:

- Verify that the data source is running.
- Verify that the DNS hostname or IP address of the server where the data source database is installed is correct. You can edit the data source to view this information.
- Check intervening firewalls. Make sure they are configured to let CA NetOps Portal communications reach the data sources. For more information about the ports to open, see the *Installation Guide*.

Solution 2:

If the failure occurred with a CA Infrastructure Management Data Aggregator data source, verify that it is running. Access the following URL:

```
http://<host>:<port_number>/rest
```

where 'host' is the IP address of the server where the Data Aggregator is installed, and 'port_number' is the port used to access the RESTful web service, usually 8181.

The web service status indicates whether the Data Aggregator is running.

Solution 3:

Check the Device Manager application.log file. The file is written to the following directory:

```
CA\PerformanceCenter\PC\logs
```

The log entry references the URI used by CA NetOps Portal to communicate with the data source, along with a stack trace.

Data Source Synchronization Fails

Symptom:

When I try to perform a data source synchronization, the 'Synchronization failure' message appears.

Solution 1:

A synchronization failure might indicate that the data source is unreachable. Do the following:

- Verify that the data source is running.
- Verify that the DNS hostname or IP address of the server where the data source database is installed is correct on the Add Data Source page.

Solution 2:

A synchronization failure can indicate that the data source could not handle the data sent to it during synchronization.

First, check the Data Source Log for the data source. For more information, see View the Data Source Log.

If you still cannot determine the source of the problem, check the Device Manager application.log file. It is written to the following directory:

```
CA\PerformanceCenter\PC\logs
```

If the data source was unable to handle data received from CA NetOps Portal during synchronization, the log entry shows a general SOAP exception.

Solution 3:

CA NetOps Portal encountered an issue during the attempted synchronization.

Check the log files, as instructed above. Look for an exception and stack trace within the following phases of synchronization:

- Pull
- Global Sync
- Bind (only executes when initially synchronizing with a data source)
- Push

The log contains detailed information about the steps that are performed during each phase. This information can help pinpoint the cause for the synchronization failure.

Solution 4:

System times are not synchronized. Check the NTP server or the system time on each server (including data sources and the CA NetOps Portal server).

Data Source Test Fails

Symptom:

I test a data source during the registration process, but the test fails.

Solution 1:

Do the following:

- Verify the DNS hostname or IP address of the server where the database for the data source is installed.
- Try running the data source registration. The data source registration might succeed even if the test failed.
- Check the logs for registration failure information. For more information, see Data Source Registration Fails.

Solution 2:

If the failure occurred with a Data Aggregator data source, verify that it is running. Access the following URL:

```
http://<host>:<portnumber>/rest
```

> **NOTE**
> This URL does not open the correct page in Mozilla Firefox. Use another supported browser.

The web service status indicates whether the Data Aggregator is running.

Solution 3:

If the failure occurred with a data source other than a Data Aggregator, check the application log file (PC/logs/application.log) for a corresponding event. The log entry includes the URL that CA NetOps Portal uses to communicate with the data source, as well as a stack trace.

Discovery Does Not Start

Symptom:

I select discovery profiles and click Run to run a discovery, but discovery fails to start, or the Run button is disabled.

Solution:

Possible reasons for a discovery failure or for a disabled Run button include the following:

- The IP domain previously specified in the discovery profile has been deleted. Assign the discovery profile to an IP domain.
- No Data Collector has been installed for the IP domain that is specified in the selected discovery profile.

  > **NOTE**
  > For information about installing Data Collector hosts, see Installing.
- One or more Data Collector hosts are installed for the IP domain that is specified in the selected discovery profile. However, all of the Data Collector hosts that are installed for the IP domain are stopped. Start the Data Collector hosts.
- The tenant is deactivated. Activate the tenant.

ETL Failures

Symptom:

After you upgrade the Data Aggregator, ETL jobs start failing and new items are missing from your dashboards.

Solution:

Run the `etlHealth.sh` script. For releases 20.2.3 and higher, the script is included with the Data Repository installation. For earlier releases, contact Support. We recommend you run this script well in advance of the upgrade and again directly before the upgrade.

Follow these steps:

1. Log in to one of the Data Repository nodes as the root user.
2. Run the validation script:

   ```
   ./etlHealth.sh dauser dapassword
   ```

3. If the health check fails, follow the instructions in the prompt to collect the irep and the data collected by the `etlHealth.sh` script, and submit these details in a Support ticket.

Gaps Appear in Reports or Views

Symptom:

A view or report shows a gap in the value for a metric.

Solution:

By default, DX NetOps Performance Management shows gaps in the data for counter values for counter wraps, bad counter values, and missing data. This behavior protects the integrity of report data. For more information, see Configure Counter Behavior and Hide Gaps in Trend Views.

Gaps in Data Appear during Throttling

Symptom:

Certain devices are missing data in DX NetOps Performance Management. You notice a gap in polling or missed polls even though the device is running.

Solution:

The throttling of outgoing SNMP requests can cause gaps in data. Throttling occurs when some polls in a poll group have responded with REQUEST_TIMED_OUT in a single poll cycle. The device is assumed to be too busy to respond to polls for that poll group. Polling is attempted again in the next poll cycle.

To address this issue, verify that polling is being throttled. To prevent gaps in data, reduce the poll rate or configure a monitoring profile poll filter. For more information, see Poll Sensitive and Critical Devices Without a Performance Impact and Configure Monitoring Profiles.

To verify that polling is being throttled, use the following method that is most convenient:

- Select the **Details** tab for a device. Look for any events related to throttling. "Polling Stopped" appears in the event list when a throttling event occurs on the device that is missing data. For more information about these events, see Polling Stopped Event Message.
- Hover over **System Health**, and click **Data Aggregator Polling**:
  - Review the Device Polling Statistic chart, which shows a systemwide total of stopped polls. Investigate any spikes above zero.
  - Review the Stopped Polls by Device chart, which shows the devices with the most stopped polls. Investigate the top devices struggling to keep up with the polling load.
- Query the `NormalizedDevicePollingStatistics` metric family. Determine the number of polls that were stopped during the poll cycle. Also, determine the number of poll groups that were stopped.
- Review the Poll Summary log for `badPollRequestTimedOutCount` and `notSentPollRequestCount` .
- If you have Detailed Poll Logging from dcdebug, review the dcdebug Detailed Poll Log for the following string. If the value is above zero, then polls have been throttled.

```
DCMResponseVariable [name={http://im.ca.com/
normalizer}NormalizedDevicePollingStatistics.NumPollsStoppedDueToPriorTimeouts,
value=
```

Group Membership Is Not Updated During Synchronization

Symptom:

After synchronization, your group memberships are not updated completely.

Solution:

Your system may have many rule groups to evaluate each synchronization cycle. Each rule group is all the rules for a single group. DX NetOps Performance Management processes as many rule groups as possible within the configured time. By default, the processing time is 60 seconds. If processing takes longer than the configured time, DX NetOps Performance Management finishes processing the current rule group. During the next synchronization cycle, the oldest unprocessed rule groups are processed first.

To evaluate more rules groups during each cycle, increase the processing time.

Follow these steps:

1. Log in to the NetOps Portal host.
2. Set a new processing time:

```
mysql netqosportal -unetqos -ppassword -e "replace into general
 values('Rules.MaxTimeSeconds', 'time');"
```

Time defines the rule processing time in seconds.
Default: 60

TIP

Set the processing time to 120 seconds and evaluate the system performance.

DX NetOps Performance Management allocates the specified time to rules processing during synchronization.

Insecure Connection Message in Firefox

Symptom:

When logging in to NetOps Portal, the following warning message appears when you click inside the **Username** or **Password** field:

"This connection is not secure. Logins entered here could be compromised."

Solution:

Some versions of Firefox show this warning for any login screen that is not using HTTPS. NetOps Portal uses HTTP by default because it is hosted as an internal application behind corporate firewalls. For more security, Configure Performance Center to Use HTTPS. For more information about this feature, see the Firefox help.

Inventory is Empty After a Data Source is Registered

Symptom:

I installed a data source and registered it, but I do not see any managed items in the Inventory.

Solution 1:

Check to make sure that the data source is registered and has an active status. Do the following:

1. Log in as a user with administrative privileges.
2. Select **Administration**, **Data Sources**, and click **Data Sources**.
   The Manage Data Sources page opens.

Solution 2:

One of the following may have occurred:

- Data source registration failed. For more information, see Data Source Registration Fails.
- Data source synchronization failed. For more information, see Data Source Synchronization Fails.

Solution 3:

Check the permissions for the user account that you used to log in. If the user account has no assigned permission groups, you see no managed items.

Make sure that you have not logged in as a user associated with the Default Tenant, who sees no managed items.

Low Data Aggregator Disk Space

Symptom:

The Data Aggregator ran out of disk space.

Solution:

Move the Data Aggregator to a new location on the same host with more disk space.

Follow these steps:

1. Do one of the following steps:
   – Stop the ActiveMQ and Data Aggregator services:

     ```
     service activemq stop
     ```

     ```
     service dadaemon stop
     ```

     NOTE

     For RHEL 7.x or OL, `service` invokes `systemctl` . You can use `systemctl` instead.
   – (Fault tolerant environment) Run one the following commands to enable the fault tolerant Data Aggregator so that it can start when necessary:
     - **RHEL 6.x:**

       ```
       service dadaemon activate
       ```

     - **RHEL 7.x, SLES, or OL:**

       ```
       DA_Install_Directory/scripts/dadaemon activate
       ```

2. Uninstall the ActiveMQ service and the Data Aggregator service from the following Data Aggregator locations:

   ```
   Current_DA_Install_Dir/scripts/activemq uninstall
   ```

   ```
   Current_DA_Install_Dir/scripts/dadaemon uninstall
   ```

   – ***Current_DA_Install_Dir*** The installation directory for the DA**Default:** `/opt/IMDataAggregator`

3. Move the contents of the current directory to the new directory:

   ```
   cp -
   rfp Current_DA_Install_Dir
   New_DA_Install_Dir
   ```

4. Delete the current directory:

   ```
   rm -rf Current_DA_Install_Dir
   ```

5. Clean up the `apache-karaf-version/data` directory:

   ```
   rm -rf New_DA_Install_Dir/apache-karaf-version/data
   ```

6. Update the following files to point to the new directory:

   ```
   /etc/DA.cfg file
   /var/.com.zerog.registry.xml

   New_DA_Install_Dir/scripts/activemq

   New_DA_Install_Dir/scripts/dadaemon

   New_DA_Install_Dir/apache-karaf-version/bin/setenv

   New_DA_Install_Dir/apache-karaf-version/bin/restart
   ```

7. Install the ActiveMQ service and Data Aggregator service from the following locations:

   ```
   New_DA_Install_Dir/scripts/activemq install

   New_DA_Install_Dir/scripts/dadaemon install
   ```

8. Do one of the following steps:
   – Start the ActiveMQ and Data Aggregator services:

     ```
     service activemq start


     service dadaemon start
     ```

   – (Fault tolerant environment) Run one the following commands to enable the fault tolerant Data Aggregator so that it can start when necessary:
     • **RHEL 6.x:**

```
service dadaemon activate
```

- **RHEL 7.x, SLES, or OL:**

```
DA_Install_Directory/scripts/dadaemon activate
```

Metric Family is Incomplete

Symptom:

I successfully imported a custom metric family, but later found a defective metric definition. For example, the <Name> property has a maximum length of 32 characters. If this limit is exceeded, it can cause synchronization problems.

Solution:

Delete the custom metric family.

WARNING

This procedure is unsupported and is an emergency procedure only.

Follow these steps:

1. Locate the following directory:

```
/opt/IMDataAggregator/apache-karaf-<vers>/certifications/custom/deploy
```

2. Delete the XML files that were created and deployed for the metric family. They are named as follows:
   – im.ca.com-normalizer-<technology>.xml
   – im.ca.com-inventory-<technology>.xml
   If applicable, also delete the file that was created for the vendor certification:
   – im.ca.com-certifications-snmp-<vendor>.xml
3. Do one of the following steps:
   – Start the Data Aggregator service:

```
service dadaemon start
```

NOTE

For RHEL 7.x or OL, `service` invokes `systemctl` . You can use `systemctl` instead.

   – (Fault tolerant environment) Run one the following commands to enable the fault tolerant Data Aggregator so that it can start when necessary:
   - **RHEL 6.x:**

```
service dadaemon activate
```

   - **RHEL 7.x, SLES, or OL:**

```
DA_Install_Directory/scripts/dadaemon activate
```

After Data Aggregator restarts, verify that the previously imported metric family or vendor certification does not appear in NetOps Portal. All previously discovered components for this custom certification are also deleted.

4. Click **Administration**, **Data Sources**.
5. Select **Data Aggregator** and click the **Resync** button.
   The components for remaining metric families synchronize between Data Aggregator and NetOps Portal.
6. Edit and correct your custom metric family XML file.
7. Import your corrected metric family XML file.

Metric Family is Not Supported

Symptom:

I created a monitoring profile to poll metric families on a collection of devices. However, in the Polled Metric Families table, one of those metric families has a status of 'Unsupported.'

Solution:

To correct the problem, follow these steps:

1. Verify that the polled device responds to SNMP queries.
2. Navigate to the unsupported metric family.
3. Verify that a vendor certification supports the metric family. If no vendor certification is defined, create a custom vendor certification.
4. Verify that all key vendor certification attributes are supported on the device. If all key vendor certificate attributes are supported:
   a. Navigate back to the device
   b. Select the metric family for which you added a custom vendor certification
   c. Click Update Metric Family.
      Your device configuration is updated.

Metric Values Do Not Appear in Table in OpenAPI

Symptom:

I ran a query to generate a table with metric values. However, when I open the table, no metric values appear in the results.

Solution:

The OpenAPI does not currently support the aggregation of multiple data samples into a single value for a particular time range.

To preview the results of your query, select the appropriate output format, such as:

- HTML table with extra metrics
- JSON
- XML

MIB Fails to Compile

Symptom:

When I review the list of MIBs in the Select MIB page of the Create Vendor Certification wizard, I receive an error message that a MIB did not compile.

Solution:

If a MIB failed to compile, follow these steps:

1. Check the error message in the Select MIB page.
2. Perform one of these actions, depending on the error type:
   – Syntax error -- Using details in the error message, correct the syntax error in your MIB and import the corrected MIB.
   – Dependency error -- Upload the required MIB to resolve the dependency issue.

   When a new MIB is imported, any existing MIB that failed to compile is recompiled in addition to the new or modified MIB.

Multiple SNMP Devices Trigger Intrusions Alarms

Symptom:

I have many SNMP devices behind a more restricted firewall configuration (such as DMZ networks). For security reasons, the SNMP devices have different community strings. I defined an SNMP profile for each different community string, but now I am getting intrusion alarms and have been logged out of CA NetOps Portal.

Solution:

To find the correct SNMP profile for a device, CA NetOps Portal tries all of the SNMP profiles. This behavior can trigger intrusion alarms and can log you out of CA NetOps Portal.

To resolve this issue, follow this process:

1. Create a separate discovery profile for a critical SNMP device.
2. Assign the SNMP profile with the correct community string to the discovery profile.
3. Repeat steps one and two for each critical SNMP device.

When discovery is run, only the assigned SNMP profile is used.

No Charts or Images are Visible in IE with HTTPS

Symptom:

Some charts or images do not appear in NetOps Portal when I use Internet Explorer with HTTPS. A red X appears to indicate that the chart or image is broken.

Solution:

By default, the TLS setting needed for HTTPS is set to TLS 1.0 in IE. To view the broken charts or missing images, turn on TLS 1.1.

Follow these steps:

1. Click Tools at the top of the Internet Explorer browser, or click the gear icon at the top right.
2. Click Internet Options.
3. Click the Advanced tab, and select the TLS 1.1 checkbox.
4. Click Apply.
   The TLS settings are saved. Charts and images appear in CAPC.

'No Data to Display' Message in Views

Symptom:

Several views on the dashboard are empty. A message states, "No Data to Display".

Solution:

A graph or table view on a dashboard can show "No Data to Display" for the following reasons:

- The data source for the view has not been installed or has not been registered.
  Each view receives data from a single data source. Some view containers appear on dashboards even if the corresponding data source is not registered. They are always empty until the data source is registered.
  You can change display settings so that such views are never displayed in dashboards.
- The data source is registered, but it has been temporarily disabled.
  A disabled data source is not polled for data. An administrator can edit the data source to enable it. For more information, see Configure a Data Source.
- The view type requires editing before data can be displayed.
  Some types of view do not have default settings. For example, multiviews and multitrend views require customization before they display any data.
- No data is available for the selected time range. To test this theory, select a different time range.
- Not enough time has transpired since polling started on the devices that are selected for reporting.
  If the polling interval is fairly long, the first data point can take a little longer to appear. Polling rates are set in the data sources.
  A service is not running.
  If the device manager service is not running, the "no data" message may appear.
- The current group does not contain items of the required type for this view.
  The group of items whose data is reported on the dashboard is shown above the Time Period selector. Check the view: is this Server report trying to show data from a group of routers?
- The group is new or has recently been changed.
  Check group membership. A group rule might be misconfigured.
  If your user account has the required role right, edit the view to select another group context. Or click the Group Filter link above the Time Period selector and select another group context for the dashboard.
- The user account of the logged-in user does not have permission to view monitored items that have reported data. For more information, see How to Create a User Account.
- The data source has not properly synchronized with CA NetOps Portal. For more information, see Data Source Synchronization Fails.
- Components or managed items were not discovered.
  This problem is data source-specific, so consult the online Help for the data source. For a Data Aggregator data source, you can check inventory discovery history. On the Discovery Profiles List page, select the Discovery profile that you created for the initial discovery and click the History button.
- Metric families were not configured or enabled.
  A Data Aggregator data source automatically applies predefined (factory) monitoring profiles to the predefined collections, such as the All Routers collection. However, custom groups and custom collections are subject to misconfigured custom monitoring profiles.
- The database query timed out. Network connectivity issues between the NetOps Portal server and the data source can cause this problem.

No Output is Generated After Running the Device Pack Generator

Symptom:

I ran the Device Pack Generator but no output files were generated.

Solution:

To verify that validation of the devicePackConfig file is successful, execute the following command:

```
createIMDevicePack   <path_of_devicePackConfig.xml>   -v
```

If the validation is unsuccessful, review the error message in the console. To fix the problem, make the recommended changes in the devicePackConfig file and rerun the script using the -v option.

You can also verify that the directory information provided in <Path> is correct and that it contains the CSV files. Also verify that the header information provided matches the CSV files.

If the problem persists, contact CA Technologies Support.

No Performance Data for a Device Pack

Symptom:

I generated and deployed my device pack in NetOps Portal. I do not see any performance data for my device pack.

Solution:

Data becomes available after two performance polls.

Old Router Getting Created When Refresh Occurs

> **NOTE**
> This scenario is applicable for all CA Network Flow Analysis (9.3.3, 9.5.0, and 9.5.1) that function as expected.

CA Network Flow Analysis and Data Aggregator monitor the same router and a router refresh has occurred in the past (when CA Network Flow Analysis is upgraded to 10.0 and DX NetOps Performance Management to 3.7)

- You can see all of the pre-refresh CA Network Flow Analysis performance data on the old router in NetOps Portal. Additionally, you can see all DA-collected performance and all of the post-refresh CA Network Flow Analysis data on the new router in NetOps Portal.

Only NFA monitors the router and a router refresh has occurred in the past (when NFA is upgraded to 10.0 and PM to 3.7)

- You can see two routers with all of the pre-refresh CA Network Flow Analysis performance data on the OLD Router in NetOps Portal and all of the post-refresh CA Network Flow Analysis data on the new Router in NetOps Portal.

Old Router Not Getting Created When Refresh Occurs

This scenario is applicable for all CA Network Flow Analysis versions (9.3.6 and 9.3.8) that do not function as expected.

CA Network Flow Analysis and Data Aggregator monitor the same router and a router refresh has occurred in the past (when CA Network Flow Analysis is upgraded to 10.0 and DX NetOps Performance Management to 3.7)

- You can see only one router with one set of interfaces that have been incorrectly reconciled by NetOps Portal. The reason is that CA Network Flow Analysis has reported two interfaces with the same interface index (IfIndex) on the same device. In this scenario, the CA Network Flow Analysis devices are reconciled in a non-deterministic manner.

OpenAPI Query Results in Empty Table

Symptom:

After you run a query, the resulting Query URL shows an empty table.

Solution:

When selecting to show metric columns in a table format, select at least one configuration column, such as item name. Another solution is to select a format other than 'HTML table with export'.

Performance Center Cannot Contact Data Aggregator

Symptom:

I installed Data Aggregator successfully, but its status in the Manage Data Sources dialog is 'Unable to Contact'.

Solution:

Do the following steps:

1. Log on to the Data Aggregator host computer. Open a console and type the following command to verify that Data Aggregator is running:

   ```
   service dadaemon status
   ```

   NOTE

   For RHEL 7.x or OL, `service` invokes `systemctl` . You can use `systemctl` instead.
2. If Data Aggregator is running, a network issue is most likely preventing CA NetOps Portal from contacting Data Aggregator. Resolve all network problems.
3. If Data Aggregator is not running, start Data Aggregator. Log on to the Data Aggregator host computer as the root or sudo user with access to a limited set of commands. Do one of the following steps:
   – Start the Data Aggregator services:

     ```
     service dadaemon start
     ```

   – (Fault tolerant environment) Run one the following commands to enable the fault tolerant Data Aggregator so that it can start when necessary:
     - **RHEL 6.x:**

       ```
       service dadaemon activate
       ```

     - **RHEL 7.x, SLES, or OL:**

       ```
       DA_Install_Directory/scripts/dadaemon activate
       ```

Polling Does Not Complete for My Sensitive Device

Symptom:

Polling on my critical device cannot complete in a single polling cycle. The large amount of network traffic sometimes completely stops my device. How can I reliably poll this device to help ensure quality performance?

Solution:

Polling is vital for monitoring a device. However, excessive polling can cause a large amount of network traffic, which can degrade your ability to monitor a device successfully. If too much network traffic is overwhelming your sensitive device, you can try the following adjustments to reduce overall traffic to the device:

- Adjust your monitoring profile to remove unnecessary metric families from polling.
- Apply a filter in your monitoring profile to reduce the number of polled interfaces.
- Adjust your monitoring profile to poll less often (for example, change the SNMP Poll Rate to 15 minutes, instead of the default 5 minutes).
- Adjust the SNMP traffic threshold to lower the number of SNMP requests that are sent to the device at a time.
- Adjust the SNMP timeouts threshold to control how many polling timeouts cause polling to suspend for the current polling cycle.

Polling Has Stopped on Discovered Metric Family

Symptom:

I select a device from the Monitored Devices page and see that polling has stopped for a metric family that the device supports. I did not intend for polling to stop for that metric family.

Solution:

Follow this process to determine why polling has stopped and perform the appropriate steps to address the cause:

1. Verify that a monitoring profile is defined and is set to poll the desired metric family.
   If this requirement is not already met, create or edit a monitoring profile with the desired metric family defined in it.
2. Verify that the device is associated with the device collection.
   If the device is not associated with the device collection, add the device to the device collection.

   NOTE
   For information about adding a device to a device collection, see the *CA NetOps Portal Administrator Guide*.
3. Verify that the monitoring profile is associated with the device collection and the device.
   If the monitoring profile is not associated, create the relationship between the monitoring profile and the device collection.

After you complete one of these actions to restart polling, select the device on the Monitored Devices page to verify:

- The status of the metric family on the Polled Metric Families tab has changed.
- The status in the Interface Components table has changed to Active.

Polling resumes automatically on existing devices.

New devices can be discovered using one of the following methods:

- Select the polled metric family on the Monitored Devices page, and click Update Metric Family.
- Set the Change Detection rate in the monitoring profile for that metric family, with Automatic Discovery set to True.

Polling Safety Valve Event Message

Symptom:

A "Polling Safety Valve" event appeared in my events list.

Solution:

Clock drift can sometimes cause polling to stop on some components and devices. When this occurs, restart the Data Collector.

For more information, see Restart the Data Collector.

Polling Stopped Event Message

Symptom:

A "polling stopped" event appeared in my events list.

Solution:

By default, Data Aggregator controls SNMP polling, helping to ensure that too many poll requests do not overwhelm a device. One method for controlling poll traffic is the SNMP timeouts threshold. The default threshold value is 15. When 15 or more SNMP requests timeout for a polling group, polling is suspended for the remainder of the current polling cycle for that polling group. Other polling groups on the device continue to poll. An event is generated, informing you of the situation.

> **NOTE**
> Polling resumes at the beginning of each poll cycle. When no timeouts occur in a complete 5-minute poll cycle, a "clear" event is generated.

For more information, see Poll Sensitive and Critical Devices Without a Performance Impact and Gaps in Data Appear during Throttling.

PrimaryIPAddress ATTRIBUTE_VALUE_NOT_ALLOWED Error in Karaf Log

Symptom

The following error message appears in the karaf.log file:

```
ERROR | 0c98d87-thread-1 | date time | AbstractReconciler |
 iliation.impl.AbstractReconciler  495 | .im.aggregator.discovery |       |
 Got unexpected error for detect change only write for attr {http://im.ca.com/
inventory}Device.PrimaryIPAddress on item Item_ID#: ATTRIBUTE_VALUE_NOT_ALLOWED
```

Solution

This error is benign and occurs when the system detects a duplicate IP address. One possible cause is that the vCenter has a decommissioned VM and the IP address has been reused. Both VMs are created in the DX NetOps Performance Management environment.

QueryBuilder Certificate Warning

Symptom:

In an environment with a Data Aggregator that is configured to use HTTPS, a certificate warning appears when you try to run QueryBuilder.

Solution:

QueryBuilder requires Single Sign-On authentication. The QueryBuilder URL redirects to NetOps Portal Single Sign-On. The Single Sign-On URL uses the NetOps Portal host IP address instead of its host name. This scenario leads to a certificate warning because SSL certificates are generated for host names not IP addresses. Add the NetOps Portal IP address as a Subject Alternative Name (SAN) in the v3 certificate extension. For more information, see Set Up SSL Certificates for Performance Center.

Report on All Pages Times Out

Symptom:

When you run a report on All Pages, it times out. When you click the link for a timed out report, the PDF contains a "Query failed" error message.

Solution:

If you experience memory issues, ensure all your servers, especially the Data Repository, meet the minimum requirements and sizing guidelines. For information about the sizing requirements, see the DX NetOps Performance Management Sizing Tool. If a report on All Pages times out and the minimum requirements and sizing guidelines are met, reach out to CA Support.

Unable to Back Up Data Repository

Symptom:

When I run the vbr.py script to back up the Data Repository, the following message appears: "Another vbr instance is already running".

Solution:

A previous backup attempt failed (for example, password-less SSH was not set up correctly).

To back up the Data Repository, remove the /tmp/.initiator.mutex file from the computer where the Data Repository is installed. The next scheduled backup occurs normally.

Unable to Resolve Issue

If you cannot resolve an issue, use the CA Remote Engineer (CARE) tool. The CARE tool gathers data that a CA Support Engineer can use to help you troubleshoot a problem. CARE contains configuration files in its scripts directory for every product that CARE supports.

CARE is installed when you install DX NetOps Performance Management components:

Component	Installation Directory
CA NetOps Portal	`/opt/CA/PerformanceCenter/RemoteEngineer`
Data Aggregator	`/opt/IMDataAggregator/RemoteEngineer`
Data Collector	`/opt/IMDataCollector/RemoteEngineer`
Data Repository	The Data Repository installer extracts CARE to the following directory: `/opt/CA/IMDataRepository_vertica9/RemoteEngineer`

When prompted by a CA Support Engineer, take the following steps to run CARE to collect troubleshooting data.

Follow these steps:

1. At a command prompt, navigate to the relevant installation directory:

   ```
   cd Component_Install_Directory
   ```

   – ***Component_Install_Directory***
   Specify the directory where CARE is installed.
2. Enter the following command:

   ```
   ./re.sh
   ```

   The first time you run the command, a license agreement appears.

Unexpected Data Aggregator Shutdown

Symptom:

Data Aggregator shuts down unexpectedly.

Solution:

Data Aggregator shuts down if it loses contact with Data Repository. If contact with Data Repository is lost, an audit message is logged in the *DA_installation_directory*/apache-karaf-<*vers*>/shutdown.log file.

> **NOTE**
> The *DA_installation_directory*/apache-karaf-<*vers*>/shutdown_details.log logs heartbeat messages between Data Aggregator and Data Repository, as well as any Data Aggregator shutdowns for debugging purposes.

To resolve any connectivity or other Data Repository issues, perform the following steps:

1. Verify that the Data Repository process is running. Do the following actions:
   a. Log in to the database server you use for Data Repository as the database administrator user, not as the root user.
   b. Type the following command:

      ```
      /opt/vertica/bin/adminTools
      ```

      The Administration Tools dialog opens.
   c. Select (1) View Database Cluster State.
      The returning window should state: “ALL” for Host and “UP” for State.
2. If Data Repository is not running, attempt to start it by performing the following steps:
   a. Log in to the database server you use for Data Repository.
   b. Type the following commands:

      ```
      /opt/vertica/bin/adminTools
      ```

      The Administration Tools dialog opens.
   c. Select (3) Start Database.
   d. Press the Space bar next to the database name, select **OK**, and press Enter.
      You are prompted for the database password.
   e. Enter the database password and press Enter.
      The Data Repository database starts.

      > **NOTE**
      > If you see an error message stating that you cannot connect because of a username or password error, it is possible that a database password change is why Data Aggregator has disconnected from Data Repository.
   f. Select (E) Exit and press Enter.
      If Data Repository does not start, contact CA Technical Support.
3. If Data Repository is running, you have a network connection problem, such as a network latency issue. Address your network connectivity problem.
4. Once Data Aggregator is running again, set up an automatic recovery of the Data Aggregator process.

Vendor Certification Expression is Erroneous

Symptom:

The MVEL compiler may not give an evaluation exception (error) for bad expressions. This situation can happen for some syntax errors, including missing or open parentheses, and multiple asterisks.

The incorrect expression is compiled without an error condition until an expression evaluation is performed with the appropriate variables. Database columns that are the target for the intended expression are not populated.

Solution:

Turn on debug logging for the ExpressionEvaluator using the following steps:

1. Locate the IMDataAggregator/apache-karaf-<*vers*>/etc directory.
2. Open the org.ops4j.pax.logging.cfg file and create the following entry:

```
log4j.logger.com.ca.im.core.expressionevaluator=DEBUG
```

3. Do one of the following steps:
   – Start the Data Aggregator service:

```
service dadaemon start
```

NOTE

For RHEL 7.x or OL, `service` invokes `systemctl` . You can use `systemctl` instead.

   – (Fault tolerant environment) Run one the following commands to enable the fault tolerant Data Aggregator so that it can start when necessary:
   - **RHEL 6.x:**

```
service dadaemon activate
```

   - **RHEL 7.x, SLES, or OL:**

```
DA_Install_Directory/scripts/dadaemon activate
```

4. Search for evaluation exceptions in the karaf.log file in the IMDataAggregator/apache-karaf-<*vers*>/data/log directory.

Vertica Fails to Install in a Cluster Environment

Symptom:

Vertica fails to install in my cluster environment.

Solution:

Verify that passwordless SSH is set up for the Vertica Linux database administrator user between the nodes in the cluster. For more information, see Prepare to Install the Data Repository.

Vertica Fails to Start

Symptom:

The Data Repository does not restart after you stopped the database through Vertica admintools.

Solution:

A leap second occurred on 30 June 2015. The extra second causes Vertica to fail to start on certain versions of the Linux operating system.

To resolve this issue, restart each node before the first time you start the database.

This issue is fixed in the following Linux Kernel versions:

- RHEL 6.1 EUS kernel-2.6.32-131.30.2 or later
- RHEL 6.2 EUS kernel-2.6.32-220.25.1.el6 or later
- RHEL 6.3 kernel-2.6.32-279.5.2 or later

RHEL 6.4 and later already contain a fix for this issue.

Vertica Fails to Install due to 'iptables' Error

Symptom:

When you run the dr_install.sh script, Vertica fails to install due to the following 'iptables' error:

```
Prerequisites not fully met during local (OS) configuration for

verify-ipAddress.xml:

   WARN (N0010): https://my.vertica.com/docs/version/HTML/index.htm#cshid=N0010

        Linux iptables (firewall) has some non-trivial rules in tables: filter
```

Solution:

If you encounter this Vertica known issue, reach out to CA Support for a workaround.

View Shows Invalid RIB Query Syntax Error

Symptom

A view does not load, and the view shows the following error message:

Invalid RIB query syntax. See logs for details.

Solution

Percentile or Projection attributes are disabled. Verify that the attributes for Percentiles and Metric Projections are configured correctly in the metric family XML. Enable the calculation, or remove the reporting field from the view.

For more information, see Metric Family XML Structure.

APIs

DX NetOps Performance Management offers APIs that let you automate provisioning and configuration tasks or extract data from the DX NetOps Performance Management database. The most frequently repeated or time-consuming tasks are exposed to you by web services.

Some APIs consist of RESTful web services. The REST model lets you access a set of resources by a fixed set of operations. This model takes advantage of widely deployed HTTP features that are supported by common hardware, such as gateway devices.

The global administrator can use the RESTful web services to perform many administrative tasks.

The following categories of APIs are available:

- **NetOps Portal REST Web Services**
  Use these web services to automate tasks that are manually performed in NetOps Portal.
  For more information, see Performance Center REST Web Services.
- **Data Aggregator REST Web Services**
  Use these web services to manage administrative operations, such as retrieving data or managing relationships between profiles and tenants or groups.
  For more information, see Data Aggregator REST Web Services.
- **OpenAPI**The Open API is a flexible tool that lets users easily extract data from the DX NetOps Performance Management database. The OpenAPI enables integration between DX NetOps Performance Management data and external applications.
  For more information, see OpenAPI.

Set Up a REST Client

You must set up a REST client to interact with the available web services. We expect that users working with our web services are familiar with RESTful APIs and how to set up a REST client.

> **NOTE**
> Due to known limitations, we recommend setting up a REST client using a browser other than Firefox.

Performance Center REST Web Services

The CA NetOps Portal RESTful web services can programmatically perform the following tasks:

- Create and update user accounts
- Create containers for MSP customer sites ("tenants")
- Create and manage groups
- Import and export dashboards
- Manage data sources
- Create, edit, and delete SNMP profiles
- Create IP domain definitions, and associate them with tenants
- Provide lists of all configuration items, such as custom user accounts, roles, or groups, that are already in the system

Access NetOps Portal Web Services

API components are installed automatically with DX NetOps Performance Management. Access the web services from a web browser. The launch page includes a list of the available web services, endpoint addresses, and WADL and WSDL URLs. The full list of all available web services for NetOps Portal is available at the following URL:

```
http://PC_host:8181/pc/center
```

From this page, you can also access the WADL of each web service. If you use a testing utility to run web service calls, you receive feedback that is useful for debugging purposes. Using a testing utility is also a timesaver. You can supply username and password parameters as service endpoints for automatic authentication of all service calls.

Such utilities require a WSDL file that describes the service being tested. A WSDL file is an XML file that conforms to the Web Services Description Language. In the REST format, the simpler Web Application Description Language (WADL) is used instead. The CA NetOps Portal API launch page gives you access to a WADL file for each web service that you can use for testing. A link to the WSDL is provided for the SOAP web services.

Most web services provide their own documentation, including lists and descriptions of the available parameters and operations. The documentation is accessible in HTML format from the API launch page:

```
http://PC_host:8181/pc/center/rest
```

Connect a REST Client to NetOps Portal

You can connect any REST client to NetOps Portal.

Follow these steps:

1. Launch the REST client.
2. Type a URL for the CA NetOps Portal RESTful web services API in the URL field. Use the following format:

   ```
   http://PC_host:8181/pc/center/webservice/Web_Service_Name
   ```

   For example, to invoke the tenants web service, supply the following URL:

   ```
   http://PC_host:8181/pc/center/webservice/tenants/
   ```

3. Select **GET** for the **HTTP Method**.
4. Select **Basic Authentication** from the **Authentication** menu.
   The Basic Authorization dialog opens.
5. Type a valid username and password for a user account that has global administrator access, and click **OK**.
6. Select **Custom Header** from the **Headers** menu.
   The **Request Header** dialog opens.
7. Type `Content-Type` as the value for the **Name** parameter.
8. Type `application/xml` in the **Value** field, and click **OK**.
   The Headers section now shows the following updated values:
   Authorization: Basic YWRtaW46YWR...
   Content-Type: application/xml

Endpoints and the XML Schema

All endpoints act on a common set of data. The data can be represented in different data formats (for example, MIME types). The format depends on the endpoint that consumes or produces the data. An XML schema describes the data and other supported data formats, such as JSON.

This documentation describes the basic terms and parameters of the XML schema to create scripts for the NetOps Portal RESTful web services. Data can be grouped by namespace. A schema describes the types and elements of each namespace. *Types* define the structure of the data, while *elements* are instances of a type. For example, elements are produced or consumed by a REST endpoint, and the structure of each element is described by its type.

Use Performance Center Web Services

The following examples illustrate some standard procedures for interacting with the DX NetOps Performance Management RESTful web services. The procedures in these examples require Administrator access.

Basic REST Web Service Examples

The following is an example of a simple PUT operation that updates the description parameter of a tenant:

```
http://PC_host:8181/pc/center/webservice/tenants/tenantName/tenantName/
tenantDescription/NewDescription
```

Substitute the desired values for the italicized terms. Some parameters are required:

- **tenantName**
  The name of the tenant that you want to edit.
- **tenantDescription**
  The new description to identify this tenant.

The following is an example of a simple GET operation that returns a list of tenant IDs and names using the tenants web service:

```
http://PC_host:8181/pc /center/webservice/tenants/idNames
```

The following XML is returned:

```
<?xml version="1.0" encoding="UTF-8"?>
<idNames>
  <idName value="tenantAccountId" />
  <idName value="tenantItemId" />
  <idName value="tenantName" />
</idNames>
```

HTTP requests always return a response and a status code, even when successful. The response text is either the expected result or an error message to indicate a problem. The status code is 200 for a successful response, or a numeric error indicator. The following HTTP response code ranges are used:

- 200 - Command status is 'OK'.
- 400 - A user error has occurred. Errors in this range indicate a problem with the input text (400) or the user credentials (403) and can usually be easily corrected.
- 500 - A system error occurred. Errors in this range typically indicate a system fault. Such errors can require assistance from CA Technical Support to resolve them.

For more information about HTTP status codes, see the IETF website.

Create User Account Roles Using the Client

Best Practice: Create user account roles before creating user accounts.

The role is a parameter of the user account that grants the user access to certain administrative features and to perform selected tasks.

You can use the REST client that you have configured to invoke the NetOps Portal roles web service.

Follow these steps:

1. In the REST client interface, type the following URL in the URL field:
   ```
   http://PC_host:8181/pc/center/webservice/roles/
   ```
2. Select **Post** for the method.
3. Type the following text into the Body text field:
   ```
   <role>
     <name>{Name for the role}</name>
     <description>{Text to describe the role in the UI}</description>
     <enabled>{Whether the role is activated or disabled}</enabled>
   ```

```
</role>
```

4. Replace any values with the values that you want to use for the new role.
   For example, supply the following parameters:

```
<role>
  <name>Data Center Manager</name>
  <description>Manages data center operations and performs troubleshooting</description>
  <enabled>True</enabled>
</role>
```

5. Click **Send** to run the method.
6. Run the following GET method to get the role ID that has been automatically assigned to the new role:
   `http://PC_host:8181/pc/center/webservice/roles/idNames`
7. Click the Response Body (Highlight) tab to see the response.
   The values of the `roleId` and `roleName` properties are returned.

   NOTE
   You can substitute either the `roleID` value or the `roleName` value for the `{roleIdName}` property where it appears in the web services documentation interface.
8. (Optional) Verify that the new role has been created by taking the following steps:
   a. Log in to NetOps Portal as a global administrator.
   b. Click **Admin**, **Roles**.
   The Manage Roles page shows a list of roles. The new role appears in the list.

Create User Accounts Using the Client

Create user accounts using the REST client that you configured. Supply the role ID for the new role that you created in the previous procedure.

Follow these steps:

1. Enter a URL for the NetOps Portal RESTful web services API in the REST client. Use the following syntax to invoke the users web service:
   `http://PC_host:8181/pc/center/webservice/users/role/{roleIdName}/{roleIdValue}/`
   For `{roleIdName}` and *`{roleIdValue}`*, use the values that were returned when you ran the roles/idNames method in an earlier procedure.

   NOTE
   You can substitute either the `roleID` value or the `roleName` value for the `{roleIdName}` property where it appears in the web services documentation interface.

   If you are deploying multi-tenancy, NetOps Portal automatically associated the role that you created with the tenant with which your user account is associated.
   a. Select **Post** for the **HTTP Method**.
   b. Leave `'application/xml'` configured as the `'Body Content-type'`.
2. Type the following parameters in the Body text field:

```
<user>
   <name>{UserName}</name>
   <description>{UserDescription}</description>
   <enabled>{UserEnabled}</enabled>
   <removable>{UserRemovable}</removable>
   <timezone>{UserTimeZone}</timezone>
   <culture>{UserCulture}</culture>
</user>
```

   Replace any values with the values that you want to use for the new user account.

For example, supply the following parameters:

```
<user>
   <name>Jane Doe</name>
   <description>User associated with the John Doe Corporation tenant.</description>
   <enabled>true</enabled>
   <removable>true</removable>
   <timezone>CST6CDT</timezone>
   <culture>en-US</culture>
</user>
```

For more information about the basic user account parameters, see Use Web Services to Create Tenants Programmatically.

3. Run the method.
4. Repeat the preceding steps until you have created as many users as you require.

Dashboards Web Service

As an administrator, you can use the NetOps Portal API web services to perform dashboard management tasks.

Use the dashboards RESTful web service to create and manage dashboard pages to display data views:

- Create new dashboards
- Import and export dashboards
- Build the dashboards once and deploy them in an additional tenant. The import feature lets you deploy extra dashboards without manually recreating them.

To import a dashboard for a specific tenant, log in as the administrator for that tenant.

> **WARNING**
>
> Exporting a dashboard in XML format via the RESTful web service can result in a broken dashboard after you post the XML. To avoid this issue, create a dashboard by copying an existing one using the UI. For more information, see Manage Dashboards.

Issue the following call to see the parameters for the dashboards web service:

```
http://PC_host:8181/pc/center/rest/dashboards/documentation
```

Export a Dashboard

Perform a GET operation to export a dashboard to an XML file that is suitable for importing into another instance of CAPC.

To export a dashboard, the internally assigned page ID is required. Use the product user interface to find the ID:

1. Navigate to the dashboard to export.
2. In the browser window, find the page ID in the URL.
   Example:
   ```
   http://PC_host:8181/pc/desktop/page?pg=2000040
   ```
   The page ID is 2000040.
3. In a REST client, set the URL to the following:
   ```
   http://PC_host:8181/pc/center/webservice/dashboards/pageId
   ```
4. Perform a GET operation.

An error or success message appears in the response.

Import a Dashboard

You can import a dashboard from an XML file.

1. Browse to select the XML file that represents a dashboard that you have exported, or paste the exported text of the XML file into the Body field.
2. Update the dashboardMenu, menuItem, and dashboardTitle attributes in the XML body for the dashboard that you import:
   ```
   <dashboardMenu>MyCustomDashboard_DashboardMenuName</dashboardMenu>
   <menuItem>MyCustomDashboard_MenuItemName</menuItem>
   <dashboardTitle>MyCustomDashboard_DashboardTitleName</dashboardTitle>
   ```
   - **<dashboardMenu>**The title that appears at the top of the dashboard menu. Specify an existing top-level dashboard menu title for this attribute.**Examples:**Infrastructure Health, Capacity Planning
   - **<menuItem>**The title of the dashboard that appears in the dashboards menu. This property is unique to each tenant.
   - **<dashboardTitle>**The title of the dashboard that appears at the top of the dashboard page. This property is unique to each tenant.
3. (Optional) Test the import of a dashboard before you import it.
   Perform the following POST operation:
   ```
   http://PC_host:8181/pc/center/webservice/dashboards/test/
   ```
   The following message appears to indicate that the test was successful:
   ```
   <?xml version="1.0" encoding="UTF-8"?>
   <msg>Validation successful.</msg>
   ```
   Checking is performed for the unique menuItem and dashboardTitle attrributes.
4. Set the URL to the following:
   ```
   http://PC_host:8181/pc/center/webservice/dashboards/import
   ```
5. Perform a POST operation.
6. An error or success message appears in the response.
   The web service assigns a page ID to the imported dashboard.

Update a Dashboard

You can update a dashboard based on a page ID. Edit the XML to modify the dashboard:

1. Set the URL to the following:
   ```
   http://PC_host:8181/pc/center/webservice/dashboards/pageId
   ```
2. Paste the XML file that represents the dashboard into the Body field, and edit the XML as needed.
3. Perform a PUT operation.
   An error or success message appears in the response.

Data Sources Web Service

The API provides the data sources web service, which lets you perform the following tasks to manage data sources:

- Get item IDs from local IDs.
- Get local IDs from the item IDs.
- Manage the data source log file.
- Modify data source settings, such as the authentication method to use.
- Register new data sources.
- Remove data sources.
- Synchronize data sources.
- View the current registered data sources.
- Get the current NetOps Portal version.

Issue the following call to see the available operations and parameters for the data sources web service:
`http://PC_host:8181/pc/center/rest/datasources/documentation`

Data Source Parameters

The current values for the following data source settings are available from the GET command:

- **id**
  An internally assigned identifier for the data source
- **enabled**
  This value indicates whether the data source is enabled
  Default: true
- **name**
  The hostname of the data source
- **authtype**
  The type of authentication to use for this data source. One of the following values:
  - **NONE**
  - **BASIC**
- **type**
  One of the following data source type values:
  - **REPORTER**: CA Network Flow Analysis
  - **SUPER_AGENT**: CA Application Delivery Analysis
  - **VOIP_MONITOR**: CA Unified Communications Monitor
  - **ANOMALY_DETECTOR**: CA Anomaly Detector
  - **EVENT_MANAGER**: Event Manager
  - **SPECTRUM_IM**: CA Spectrum
  - **APM**: CA Application Performance Management
  - **DATA_AGGREGATOR**: The default DX NetOps Performance Management data source
  - **CATALYST_CONNECTOR**: The CA Catalyst Connector for CA NetOps Portal
  - **SERVICE_OPERATIONS_INSIGHT**: CA Service Operations Insight
  - **RESERVED_CUSTOMER_N**: An enum that has been reserved for the unspecified custom uses.
- **consoleSameAsDataSource**
  This parameter indicates whether the data source web console address is the same as the hostname. Use this parameter in cases where network address translation is deployed.
  Default: true
- **consoleAddress**
  The IP address where DX NetOps Performance Management contacts the data source console.

Data Sources Web Service Examples

The following examples demonstrate some of the operations for the data sources web service.

- **get item ids**
  This operation retrieves the NetOps Portal item ids given the data source local id. If an id is not found, it is omitted from the returned data.
  URL: `http://hostname:8181/pc/center/webservice/datasources/{idName}/{idValue}/itemids`
  Where:
  – **{idName}**
    One of the property name values that the get id names method of this web service returns.
    {idName} can be any of the following values:
    - **dataSourceId**
      Example: `http://PC_address:8181/pc/center/webservice/datasources/dataSourceId/3/itemids`
    - **dataSourceGUID**
      Example: `http://PC_address:8181/pc/center/webservice/datasources/dataSourceGUID/7b0ef6b1e9094d599821b2b07d78d83d/itemids`
    - **dataSourceConsoleName**
      Example: *http://PC_address:8181/pc/center/webservice/datasources/dataSourceConsoleName/Data%20Aggregator%40PC_DA.ca.com/itemids*
  – **{idValue}**
    A value for the property denoted by **idName**
  HTTP method = POST
  XSD for the provided XML: `http://hostname:8181/pc/center/rest/datasources/xsd`
  To translate an array of local ids into item ids, supply XML in the following format:

```
<LocalIDs>
     <LocalID ID="4514"/>
     <LocalID ID="4705"/>
     <LocalID ID="4501"/>
     <LocalID ID="4540"/>
     <LocalID ID="4511"/>
     <LocalID ID="4499"/>
</LocalIDs>
```

  The function returns an array of **ItemIDResult** objects.
  Example:

```
<ItemIDResults>
     <ItemIDResult ItemID="406" LocalID="4514"/>
     <ItemIDResult ItemID="407" LocalID="4705"/>
     <ItemIDResult ItemID="408" LocalID="4501"/>
     <ItemIDResult ItemID="409" LocalID="4540"/>
     <ItemIDResult ItemID="410" LocalID="4511"/>
     <ItemIDResult ItemID="411" LocalID="4499"/>
</ItemIDResults>
```

- **get local ids**
  This operation retrieves the data source local id given the NetOps Portal item ids. If an id is not found, it is omitted from the returned data.
  URL: `http://hostname:8181/pc/center/webservice/datasources/{idName}/{idValue}/localids`
  Where:
  – **{idName}**
    One of the property name values that the get id names method of this web service returns.

{**idName**} can be any of the following values:

- **dataSourceId**
  Example: *http://PC_address:8181/pc/center/webservice/datasources/dataSourceId/3/localids*
- **dataSourceGUID**
  Example: `http://PC_address:8181/pc/center/webservice/datasources/dataSourceGUID/7b0ef6b1e9094d599821b2b07d78d83d/localids`
- **dataSourceConsoleName**
  Example: `http://PC_address:8181/pc/center/webservice/datasources/dataSourceConsoleName/Data%20Aggregator%40PC_DA.ca.com/localids`

– **{idValue}**
A value for the property denoted by **idName**

HTTP method = POST
XSD for the provided XML: `http://hostname:8181/pc/center/rest/datasources/xsd`
To translate an array of the specified NetOps Portal item ids to local ids, supply XML in the following format:

```
<ItemIDs>
     <ItemID ID="412"/>
     <ItemID ID="413"/>
</ItemIDs
```

The function returns an array of **ItemIDResult** objects.
Example:

```
<ItemIDResults>
     <ItemIDResult ItemID="412" LocalID="4590"/>
     <ItemIDResult ItemID="413" LocalID="4760"/>
</ItemIDResults>
```

Devices Web Service

The API provides the devices web service to let you view the devices that belong to a tenant. You can also retrieve a list of interfaces from a device (a router or switch) in the inventory of managed items.

All devices are associated with a tenant. If you are not deploying multi-tenancy, devices are associated with the Default Tenant. Run the getlist method to the root /devices/ to retrieve a list of devices that are associated with your tenant.

Issue the following call to see the parameters for the devices web service:

`http://PC_host:8181/pc/center/webservice/devices`

Issue the following call to see a list of supported operations:

`http://PC_host:8181/pc/center/rest/devices/documentation`

Available GET Methods

- **get list**
  Gets a list of devices that belong to the tenant of the logged-in user.

```
http://PC_host:8181/pc/center/webservice/devices
```

- **get interfaces**
  Gets a list of all interfaces that belong to a specified device.

  ```
  http://PC_host:8181/pc/center/webservice/devices/idName/idValue/interfaces
  ```

The following URL lets you retrieve a list of internally assigned identifiers that can be used in other methods to identify devices:

```
http://PC_host:8181/pc/center/webservice/devices/idNames
```

Or you can retrieve a list that is filtered by tenant and by device subtype as follows:

```
http://PC_host:8181/pc/center/webservice/devices/subtype/tenant/tenantIdName/
tenantIdValue
```

The subtype further describes the device. For example, the server, switch, and router subtypes identify devices, while an interface can have subtype "physical" or "virtual".

NOTE

: A GET to the following URL for the groups web service returns a list of group members that includes their subtype:

```
http://PC_host:8181/pc/center/webservice/groups/idName/idValue/items
```

Set the Alias Name for a Device

You can set an alias name for a single monitored device. The alias name appears in the inventory list of devices and in the inventory list of interfaces.

NOTE

An alias that is set using REST web services takes precedence over the alias that you can set by importing a CSV file.

To set an alias name for a single monitored device, run a PUT to the following URL:

```
http://PC_host:8181/pc/center/webservice/devices/deviceItemId/device_id/
aliasName/alias_name
```

- ***device_id***
  Is the device item identification number for the monitored device for which you are setting an alias name.
- ***alias_name***
  Is the alias name for the monitored device.

Example:

Run a PUT to the following URL to set 'Alias name for 96.24' as the alias name for a device with a device item ID of 119:

```
http://PC_host:8181/pc/center/webservice/devices/deviceItemId/119/aliasName/Alias name
 for 96.24
```

> **NOTE**
> To set the aliases for multiple device items, use the `update_alias_name.sh` script.

Set Interface Name Aliases

As an administrator, you can set aliases for interface component names per device item. An interface alias is a name that the administrator configures and applies to the associated interface component in NetOps Portal. Users see the interface aliases in their dashboards and views depending on the role rights you assign.

> **NOTE**
> An alias that is set using REST web services takes precedence over the alias that you can set by importing a CSV file.

The following URL shows the syntax:

```
http://PC_host:8181/pc/center/webservice/devices
```

Follow these steps:

1. Determine which interfaces you want to set an alias for:
   a. To return a list of all of the interfaces on a device item, enter the following URL in the REST client:

      ```
      http://PC_host:8181/pc/center/webservice/devices/deviceItemId/device_id/interfaces
      ```

      - ***device_id***
        Is the device item identification number for the monitored device that interface components are associated with.

   b. Select **GET** for the **"HTTP" Method**.
   c. Determine which interfaces you want to set an alias for.
2. Enter one of the following URLs in the REST client:

   ```
   http://PC_host:8181/pc/center/webservice/devices/deviceItemId/device_id/
   setInterfaceNameAlias
   ```

   ```
   http://PC_host:8181/pc/center/webservice/devices/domainItemId/domain_id/device_IP/
   setInterfaceNameAlias
   ```

   – ***domain_id***
     Is the domain identification number for the domain that interface components are associated with.
3. Select **PUT** for the **"HTTP" Method**.

4. To set the interface name aliases, enter the following information in the Body tab of the HTTP Request pane:

```
<interfaces>

     <interface>

          <itemId>interface_item_id</itemId>

          <nameAlias>alias_for_interface</nameAlias>

     </interface>

     ...

</interfaces>
```

Example:

To set 'Et0 - alias' and 'Se0 - alias' as the interface name alias for two interfaces, enter the following information in the Body tab of the HTTP Request pane:

```
<interfaces>

     <interface>

          <itemId>164</itemId>

          <nameAlias>Et0 - alias</nameAlias>

     </interface>

     <interface>

          <itemId>165</itemId>

          <nameAlias>Se0 - alias</nameAlias>

     </interface>

</interfaces>
```

NOTE
To set the aliases for interface component names across multiple device items, use the `update_alias_name.sh` script.

Set Component Name Aliases

As an administrator, you can set aliases for component names per device item. A component alias is a name that the administrator configures and applies to the associated component in NetOps Portal. Users see the component aliases in their dashboards and views depending on the role rights you assign.

> **NOTE**
>
> An alias that is set using REST web services takes precedence over the alias that you can set by importing a CSV file.

The following URL shows the syntax:

```
http://PC_host:8181/pc/center/webservice/devices
```

Follow these steps:

1. Determine which components you want to set an alias for:
   a. To return a list of all the components on a device item, use one of the following URLs in the REST client:
      - `http://PC_host:8181/pc/center/webservice/devices/deviceItemId/device_id/components`
        - ***device_id***
          The device item identification number for the parent device of the components
      - `http://PC_host:8181/pc/center/webservice/devices/domainItemId/domain_id/device_IP/components`
        - ***domain_id***
          The domain identification number for the IP domain that the device is in.
   b. Select GET for the **"HTTP" Method**.
      All components on the device are returned. Determine which components you want to set an alias for.
2. Enter one of the following URLs in the REST client:
   - `http://PC_host:8181/pc/center/webservice/devices/deviceItemId/device_id/setComponentNameAlias`
   - `http://PC_host:8181/pc/center/webservice/devices/domainItemId/domain_id/device_IP/setComponentNameAlias`
3. Select PUT for the "HTTP" Method.
4. To set the component name aliases, use the following information in the Body tab of the HTTP Request pane:

```
<deviceComponents>

    <deviceComponent>

        <itemId>component_ item_ id</itemId>

        <nameAlias>alias_for_component</nameAlias>

    </deviceComponent>

    ...

</deviceComponents>
```

NOTE

To set the aliases for for component names across multiple device items, use the `update_alias_name.sh` script.

Update the Device Life Cycle State

Life cycle states define the monitoring behavior for devices. Lifecycle management enables you to define the usage state of SNMP and ICMP devices. Components inherit the life cycle state of the associated device.

NOTE

Lifecycle management does not apply to devices from DX NetOps Mediation Manager.

You can manage the device life cycle state using the NetOps Portal API or the Manage Device Life Cycle user interface in NetOps Portal. To use the Manage Device Life Cycle user interface, see Manage Device Life Cycles.

TIP

In environments where DX NetOps Spectrum manages the life cycle state, use the DX NetOps Spectrum REST API to change the life cycle state for devices instead of this REST API. For more information, see the CA Spectrum documentation.

To define when a device is active, retired, or in maintenance, change the life cycle state.

To update the device life cycle state for a single monitored device, run a PUT to the following URL:

```
http://PC_host:8181/pc/center/webservice/devices/{idName}/{idValue}/
lifeCycleState/{newState}
```

- ***{idName}***
  Specify one of the property name values that the `get id names` method of this web service returns.
- ***{idValue}***
  Specify a value for the property that is denoted by ***idName*** .
- ***{newState}***
  Specify in ALL CAPS one of the following new life cycle states:
  - **ACTIVE**
    Specifies the normal operating status of a device.
  - **RETIRED**
    Specifies that the device is no longer in use and that no monitoring occurs. This state disables polling, threshold monitoring, notifications, and change detection. The system does not update the SNMP Profile or change the hostname.

    NOTE

    Any virtual device (Virtual Machine or ESX), discovered with systemEdge is not polled for its vCenter statistics when the device lifecycle state is Retired.
  - **MAINTENANCE**
    Specifies that the device is temporarily under maintenance. The behavior of this state is configurable.

Domains Web Service

The API provides the domains web service to let you view, create, and modify IP domain definitions.

Managed items are associated with IP domains by the data sources during data collection. Consult the documentation for each registered data source to determine the steps to take to associate items with domains.

Domains Web Service Example Syntax

Issue the following call to see the parameters for the domains web service:

```
http://PC_host:8181/pc/center/webservice/domains
```

Issue the following call to see a list of supported operations:

```
http://PC_host:8181/pc/center/rest/domains/documentation
```

Available GET Methods

- **get list**
  Gets a list of all of the IP domains that belong to the tenant for the logged-in user.
  Run the `getList` method to the root `/domains/` to retrieve a list of all IP domain IDs for the tenant of the currently logged-in administrator:
  – If you are logged in as a global administrator, you see a list of domain IDs for the Default Tenant only.
  – If you are logged in as a tenant administrator, you see a list of domain IDs for your tenant.
- **get**
  Retrieves information about a specified IP domain.

  ```
  http://PC_host:8181/pc/center/webservice/domains/idName/idValue
  ```

  NOTE
  The global administrator sees only the IP domains that fall within the Default Tenant. A tenant administrator sees only the IP domains within that tenant.
- **get domain for group**
  Gets the IP domain associated with a specified group.

  ```
  http://PC_host:8181/pc/center/webservice/domains/group/groupIdName/groupIdValue
  ```

- **get id names**
  Retrieves a list of identifiers that can be used to identify IP domains in other web service operations.

  ```
  http://PC_host:8181/pc/center/webservice/domains/idNames
  ```

- **get list**
  Gets all of the IP domains that belong to the tenant for the logged-in user.

  ```
  http://PC_host:8181/pc/center/webservice/domains
  ```

- **get list by tenant**
  Retrieves the list of all of the IP domains that are associated with the specified tenant ID:

  ```
  http://PC_host:8181/pc/center/webservice/domains/tenantItemId/tenantId
  ```

- **get list with translation**

Gets all of the IP domains that belong to the tenant for the logged-in user. Any localized text is translated to the specified language.

```
http://PC_host:8181/pc/center/webservice/domains/cultureId
```

Available POST Method

- **create**
  Creates an IP domain.

```
http://PC_host:8181/pc/center/webservice/domains
```

Available PUT Method

- **update**
  Updates a specified IP domain.

```
http://PC_host:8181/pc/center/webservice/domains/idName/idValue
```

Available DELETE Method

- **delete**
  Deletes an IP domain definition.

```
http://PC_host:8181/pc/center/webservice/domains/{idName}/{idValue}
```

Basic IP Domain Parameters

The current values for the following domain parameters are available from the GET command:

- **cultureID**
  Specifies a language (locale). Supply a language identifier from the following list:
  - en-US (English, United States)
  - ja-JP (Japanese)
  - zh-CN (Simplified Chinese)
  - fr-FR (French, France)
- **dnsProxyAddress**
  Is the IP address of the DNS proxy server.
- **domainItemID**
  Is an internal (database) identifier for a tenant definition.
- **groupItemID**
  Is an internal (database) identifier for the group definition associated with a domain.
- **itemDesc**
  Describes this domain namespace, such as naming the enterprise that owns it.
- **itemName**
  Identifies the domain.

tenantIDIs an internal (database) identifier for a tenant definition. Identifies the tenant associated with this domain.

- **primaryDNSAddress**
  Is the IP address of the primary name server for this domain.
- **primaryDNSPort**
  Is the port number that the primary name server uses.
- **secondaryDNSAddress**
  (Optional) Is the IP address of the secondary name server for this domain.
- **secondaryDNSPort**
  (Optional) Is the port number that the secondary name server uses.
- **isDNSProxyEnabled**
  Indicates whether the proxy address is enabled for this IP domain.
- **deviceAlias**
  Indicates the alias to use for a managed item. A device alias is a user-configured name that is applied to the associated managed item in CA NetOps Portal.
- **deviceAliasList**
  Identifies a CSV or TXT file of alternate interface descriptions. A comma-separated list of values that include the device IP address, interface name, interface description, and alternate interface description (alias) mappings.
- **interfaceDescriptionOverride**
  Indicates the alternate description to use for an interface. Overrides the interface descriptions that appear in CA NetOps Portal by default.

Groups Web Service

The NetOps Portal API provides web services to let you perform common group management tasks.

Use the groups RESTful web service to create and manage groups of monitored items. You can create new groups and can add items to them manually. You can also write rules to create and populate groups that are based on item attributes.

Looking Up the ID of a Group

For most of the NetOps Portal API calls, you need the numeric groupItemId for a group. You can use any of the following three methods to look up these IDs.

1. **groupItemId Lookup Service**
   This service lets you look up the ID of a single group path using a simple HTTP GET.
   To look up the ID of "All Groups/Inventory", use the *groupItemId* parameter. The syntax is as follows:

   ```
   http://PC_host:8181/pc/center/webservice/groups/groupPath/All%20Groups%2FInventory/groupItemId
   ```

   It is not mandatory to specify the entire path. For example, you can find the "`All Groups/Tenants/Acme/Groups/Sites/Boston`" group by searching for "`groupPath/Acme%2FGroups%2FSites%2FBoston/groupItemId`".

   NOTE

   When specifying the group path, use URL text encoding. Spaces must be replaced with %20 and the groups should be separated by %2F.

   The xml that is returned is as follows:

   ```
   <groupItemId>
       <groupItemId>5</groupItemId>
   </groupItemId>
   ```

NOTE
The search might match more than one group, but returns only one group.

2. **Bulk Group Find Service**
   This service lets you look up the ID of several groups with a single HTTP POST. The syntax is as follows:
   ```
   http://PC_host:8181/pc/center/webservice/groups/find
   ```
   The XML to POST is as follows:
   ```
   <groups>
   <group Path='/All Groups/Tenants/Acme/Groups/Sites/San Francisco'/>
   <group Path='/All Groups/Tenants/Acme/Groups/Sites/Austin'/>
   <group Path='/All Groups/Tenants/Acme/Groups/Sites/Boston'/>
   </groups>
   ```
   You can look up for any number of groups. For each group path that is found, the corresponding group ID and the group Name are returned. If a path does not match a group, no ID is returned.
   The XML that is returned for the above POST is as follows:
   ```
   <?xml version="1.0"
   encoding="UTF-8"?>
   <groups>
        <group ID="10503"Name="San Francisco" Path="/All Groups/Tenants/Acme/Groups/Sites/San Francisco"/>
        <group Path="/All Groups/Tenants/Acme/Groups/Sites/Austin"/>
        <group ID="10505"Name="Boston" Path="/All Groups/Tenants/Acme/Groups/Sites/Boston"/>
           </groups>
   ```

   NOTE

   The 'Austin' group does not exist, so no group ID is returned for this group.
3. **Groups Web Service**
   This service lists all child groups of the specified group. This call takes several minutes to return a large group hierarchy. This method is the slowest as it returns more data.
   To look up the ID of "All Groups/Inventory", the syntax is as follows:
   ```
   http://PC_host:8181/pc/center/webservice/groups/groupPath/All%20Groups%2FInventory
   ```

   NOTE

   When specifying the group path, use URL text encoding. Spaces must be replaced with %20 and the groups should be separated by %2F.

Syntax for Site Group Management

To get a list of all groups under All Groups, use the groupPath parameter or the groupItemId parameter.

Issue the following call to use the groupPath parameter to identify the default group:

```
http://PC_host:8181/pc/center/webservice/groups/groupPath/All%20Groups
```

NOTE
When using some REST clients, the ‘All Groups’ syntax is required rather than ‘All%20Groups’. But in general, blank spaces are not valid in URLs.

Issue the following call to get the identifier (siteId) for a site group:

```
http://PC_host:8181/pc/center/webservice/groups/groupItemId/siteId
```

The following XML shows an example of the return for the site ID:

```
<GroupTree siteId="118" inheritDefault="true" path="Austin, TX">
```

To get a list of all subgroups under a group that you specify, you have two options:

- Use the `groupPath` parameter.
- Use the `groupItemId` parameter.

Issue the following call to use the `groupPath` parameter to identify the group whose subgroups are listed in the XML that is returned:

```
http://PC_host:8181/pc/center/webservice/groups/groupPath/All%20Groups%2FInventory
```

Issue the following call to use the `groupItemId` parameter to identify the group whose subgroups are listed in the XML that is returned:

NOTE
The `groupItemId` of the default 'Inventory' group is 5.

```
http://PC_host:8181/pc/center/webservice/groups/groupItemId/5
```

The XML that is returned includes the syntax of any group rules that are applied to that group. Test with various rules that you create in the user interface, and examining the syntax that is generated.

Site Groups and Rules

Group rules support multiple comparisons, in addition to regular expressions. For example, use the following syntax in the XML to post a group rule that adds devices whose name begins with the word 'Cisco':

```
<Match>
   <Compare readOnly="true" using="MEMBER_OF">
       <Property name="ItemID" type="device"/>
           <Value reference="/All Groups">1</Value>
   </Compare>
   <Compare readOnly="false" using="STARTS_WITH">
      <Property name="AlternateName" type="device"/>
           <Value>Cisco*</Value>
   </Compare>
</Match>
```

For group path syntax, forward-slash characters are appropriate for the XML documents that you post. This example assumes that you already have a group structure of “All Groups\Texas\Austin”:

```
<GroupTree inheritDefault="true" path="/All Groups/Texas/Austin">
    <Group desc="" inherit="true" location="" name="CA Officetype="custom group">
        <Group desc="" inherit="true" location="" name="Austin Lab" type="custom group"/>
    </Group>
    <Group desc="" inherit="true" location="" name="Austin Data Center" type="custom group"/>
</GroupTree>
```

In the URL for the web service request, use a backslash character for a group path. Do not use forward slashes in the URL.

Groups Web Service Example Syntax

Issue the following call to see the parameters for the groups web service:

```
http://PC_host:8181/pc/center/webservice/groups/idNames
```

Issue the following call to see a list of supported operations:

```
http://PC_host:8181/pc/center/rest/groups/documentation
```

To get a list of all groups under the highest-level group in the Groups tree (the default, 'All Groups') you can use the groupPath parameter or the groupItemId parameter.

Issue the following call to use the groupPath parameter to identify the default group:

```
http://PC_host:8181/pc/center/webservice/groups/groupPath/All%20Groups
```

> **NOTE**
> When using some REST clients, the ‘All Groups’ syntax is required rather than ‘All%20Groups’. But in general, blank spaces are not valid in URLs.

Issue the following call to use the groupItemId parameter to identify the default group (whose groupItemId value is 1):

```
http://PC_host:8181/pc/center/webservice/groups/groupItemId/1
```

Subgroup Syntax

To get a list of all subgroups under a group that you specify, you have two options:

- Use the `groupPath` parameter.
- Use the `groupItemId` parameter.

Issue the following call to use the groupPath parameter to identify the group whose subgroups are listed in the XML that is returned:

```
http://PC_host:8181/pc/center/webservice/groups/groupPath/All%20Groups%2FInventory
```

Issue the following call to use the groupItemId parameter to identify the group whose subgroups are listed in the XML that is returned:

> **NOTE**
> The `groupItemId` of the default 'Inventory' group is 5.

```
http://PC_host:8181/pc/center/webservice/groups/groupItemId/5
```

The XML that is returned includes the syntax of any group rules that are applied to that group. Test with various rules that you create in the user interface, and examining the syntax that is generated.

Site Group Syntax

To create a site group, use the following XML syntax in a POST command:

```
<GroupTree path="/All Groups">
```

```
    <Group name="group_name" desc="group_description"
    inherit="true" type="site" location="North America"
    bHourID="99990" timeZone="EST"/>
 </GroupTree>
```

- **inherit**
  Indicates whether the group includes child items of group members. For example, if this attribute is set to true, device interfaces are added to a group if the device is added to the group.
- **type**
  Indicates the type of group. The following values are supported.
  Values:
  – **user group** (default)
    A group that a user has created.
  – **site**
    A user-created group that represents a physical site.
- **bHourID**
  (Optional) The internally assigned identifier of the business-hour definition that you want to associate with this site group.
- **timeZone**
  (Optional) The time zone to associate with this site group. Time zones can only be associated with site groups, not with user groups.

Group Rules

Group rules support multiple comparisons, in addition to regular expressions. For example, you want a group rule that adds devices in the Routers group whose name begins with 'Cisco'. The group ID of the Router group is 31. Use the following syntax in the XML:

```
<Match>
   <Compare readOnly="true" using="MEMBER_OF">
       <Property name="ItemID" type="device"/>
           <Value reference="/All Groups/Inventory/All Items/Routers">31</Value>
   </Compare>
   <Compare readOnly="false" using="STARTS_WITH">
      <Property name="AlternateName" type="device"/>
           <Value>Cisco*</Value>
   </Compare>
</Match>
```

NOTE

Scope group rules as narrowly as possible. Do not scope group rules to All Groups.

AllowDeletes

Group deletion requires the `allowDeletes` parameter to be set to 'true'. When the `allowDeletes` parameter is set to 'true' for a group, groups under the `GroupTree path` are retained, unless they are excluded from the XML body. Groups under the `GroupTree path` that are excluded from the XML body are deleted. When the `allowDeletes` parameter is set to 'false' for a group, groups under the `GroupTree path` are retained regardless of whether they are included in the XML body.

Apply and set this parameter to 'true' for a container group when you want to delete a subgroup. For example, the following XML deletes any unlisted groups under Austin:

```
<GroupTree inheritDefault="true" path="/All Groups/Texas/Austin" allowDeletes=”true”>
```

```
    <Group desc="" inherit="true" location="" name="CA Office" type="user group">
        <Group desc="" inherit="true" location="" name="Austin Lab" type="user group"/>
    </Group>
    <Group desc="" inherit="true" location="" name="Austin Data Center" type="user
group"/>
</GroupTree>
```

The following XML deletes any unlisted groups under Texas. For example, this XML would delete the Austin subgroup from the previous example.

```
<GroupTree path="/All Groups/Texas">
    <Group name="USA" desc="Group to represent the entire
    United States" allowDeletes="true" type="user group"/>
</GroupTree>
```

For group path syntax, forward-slash characters are appropriate for the XML documents that you post.

In the URL for the web service request, use a backslash character for a group path. Do not use forward slashes in the URL.

Roles Web Service

The API provides the roles web service to let you view, create, and modify user account roles.

The *role* is a parameter assigned to a user account that controls user access to product features and dashboard pages. Based on user job functions, the role grants administrative access to product configuration using *role rights*. Roles let users access data and product features that they require to perform their duties and restrict access to features that they do not require. To assign custom or factory roles to user accounts, use the users RESTful web service.

Basic Role Parameters

The current values for the following user account role settings are available from the GET command:

- **accessRights**
  Are the role rights that are allocated to this role. Multiple role rights are available for allocation, and some are data source-specific. To get a list of all available role rights, the **categoryId** is required. The categoryId with value 1 corresponds to both CA NetOps Portal and Data Aggregator.
- **culture**
  Specifies a language (locale). Supply a language identifier from the following list:
  - en-US (English, United States)
  - ja-JP (Japanese)
  - zh-CN (Simplified Chinese)
  - fr-FR (French, France)
- **description**
  (Optional) Describes the role to help you identify it.
- **enabled**
  Determines whether the role is enabled for use (activated). Values are true or false.
- **name**
  Is a name for the role. The name is limited to 50 characters.
- **selections**

Provides sets of access rights that you can selectively grant to this role, organized into categories.

- **userCount**
  Is the number of users who have this role assigned to their user account.
- **userID**
  Is an internally assigned value for the role.

Roles Web Service Example Syntax

Issue the following call to see the available operations and parameters for the roles web service:

```
http://PC_host:8181/pc/center/rest/roles/documentation
```

Available GET Methods

- **get access rights**
  Retrieves a list of the role rights that are assigned to a specified role. Use the following syntax:
  ```
  http://PC_host:8181/pc/center/webservice/roles/idName/idValue/rights/cultureId
  ```

- **get access rights by category**
  Retrieves a list of all available role rights for a specified category. The category is either CA NetOps Portal or a data source. A categoryId of 1 applies to both CA NetOps Portal and Data Aggregator.
  ```
  http://PC_host:8181/pc/center/webservice/roles/rights/categoryid/cultureId
  ```

- **get category**
  Retrieves an XML document that includes the categoryId:
  ```
  http://PC_host:8181/pc/center/webservice/roles/rights/categories/en-US
  ```

- **get by tenant**
  Retrieves a list of all roles that are associated with the tenant for the logged-in user. Use the following syntax:
  ```
  http://PC_host:8181/pc/center/webservice/roles/idName/idValue/rights/tenant/
  tenantIdName/tenantIdValue/cultureId
  ```

- **get categoryId**
  Retrieves XML that shows the available category IDs.
  ```
  http://PC_host:8181/pc/center/webservice/roles/rights/categories/cultureId
  ```

  A value of 1 corresponds to CA NetOps Portal and Data Aggregator. Role rights in that category only apply to a Data Aggregator data source.
- **get id names**
  Retrieves a list of identifiers that can be used to identify roles in other web service methods. Use the following syntax:
  ```
  http://PC_host:8181/pc/center/webservice/roles/idNames
  ```

Available PUT Methods

- **copy**
  Copies a role. This method creates a copy of a specified role and associates it with the tenant for the logged-in user. Use the following syntax:
  ```
  http://PC_host:8181/pc/center/webservice/roles/idName/idValue/copy/roleName/
  description/enabled/cultureId
  ```

- **update**
  Modifies a specified role. The role name and tenant ID are required elements of the role parameters.
  `http://PC_host:8181/pc center/webservice/roles/idName/idValue/cultureId`

Available POST Methods

- **create**
  Creates a role. The new role is associated with the tenant for the logged-in user:
  `http://PC_host:8181/pc/center/webservice/roles/`

- **create for tenant**
  Creates a role and assigns it to the specified tenant.
  `http://PC_host:8181/pc/center/webservice/roles/tenant/tenantIdName/tenantIdValue`

Create a Role

Use the roles web service to create user account roles.

Issue the following call to see the parameters for the roles web service:

`http://PC_host:8181/pc/center/webservice/roles`

Follow these steps:

1. Enter a URL for the CA NetOps Portal RESTful web services API in the REST client. Use the following format:
   `http://PC_host:8181/pc /center/webservice/roles`
2. Select **POST** for **"HTTP" Method**.
3. Provide a valid Username and Password for a user account that has global administrator access to NetOps Portal.
4. Select **'application/xml**' as the **'Body Content-type'** in the Body settings.
5. Paste XML in the Body field that resembles the following example:

```
<role>
  <name>TestRoleName</name>
  <description>Test Role Description</description>
  <enabled>true</enabled>
  <accessRights>
  <accessRight>
  <accessRightName>ViewToS</accessRightName>
  <categoryId>1</categoryId>
  <enabled>true</enabled>
  </accessRight>
  </accessRights>
</role>
```

   - **accessRights**
     Correspond to role rights.
   - **accessRightName**
     The name of the role right. For example, the administerGroups role right lets the user with this role manage a limited section of the Groups tree.
   - **categoryId**

Identifies the category of role rights, such as CA NetOps Portal role rights or data-source-specific role rights.

6. Run the method.

Users Web Service

The API provides a users web service to manage user accounts. You can perform the following tasks:

- Retrieve a current list of user accounts.
- Create user accounts.
- Modify user accounts.
- Delete user accounts.

The following topics provide information about related parameters:

Basic User Account Parameters

The following is a list of the parameters for the users web service:

NOTE
You can get the parameters and available operations for the users web service by issuing the following call:
`http://PC_host:8181/pc/center/rest/users/documentation`

- **userID**
  Internally-assigned value for the user account.
- **name**
  Login name for the user account. The name is limited to 50 characters.
- **description**
  (Optional) Describes the user account to help you identify it.
- **enabled**
  Determines whether the user account is enabled for use (activated).
- **removable**
  Determines whether you can delete the user account (remove it from the database).
  Values: true or false.

  NOTE
  You can delete all user accounts except the **admin** and **user** predefined user accounts.
- **timezone (tz)**
  Corresponds to the time zone in which the user views report data.
  Default: UTC (Coordinated Universal Time).
- **userLevel**
  Identifies the product privilege assigned to the user account.
- **role**
  Specifies the role assigned to the user account.
- **permissionId**
  Specifies the ID for the **My Assigned Groups** for the user account.
- **tenantId**
  The internal (database) identifier for the tenant with which the user account is associated.
- **culture**
  Specifies the language (locale) code for the user.
  Options:

- en-US (English, United States)
- ja-JP (Japanese)
- zh-CN (Simplified Chinese)
- zh-TW (Traditional Chinese)
- fr-FR (French, France)

Example:

Issue the following GET call to see the current values for the `culture` parameter for users that are associated with the same tenant as the account that is used to run this command:
`http://PC_host:8181/pc/center/webservice/users/cultureId`
For *`cultureId`*, supply the language code for the language in which you would like to view the output, such as `en-US`.

Users Web Service Example Syntax

The following HTTP methods are available with the users web service:

Available GET Methods

NOTE
You cannot retrieve password information using the GET method.

- **get groups owned by user**
  Retrieves a list of groups for the specified user. The groups that are returned are groups that are owned by the specified user, meaning that this user can modify or delete these groups:
  `http://PC_host:8181/pc/center/webservice/users/idName/idValue/groupsOwnedByUser`
- **get groups**
  Retrieves a list of groups to which the specified user has view access. The groups that are returned are groups that are within the permission set of the specified user. The user cannot modify or delete these groups:
  `http://PC_host:8181/pc/center/webservice/users/idName/idValue/groups`
- **get administered groups**
  Retrieves a list of all administered groups associated with a specified user account:
  `http://PC_host:8181/pc/center/webservice/users/idName/idValue/administeredGroups`
- **get id names**
  Retrieves a list of identifiers that can be used to identify users in other web service methods:
- **idValue**
  The value for the identifying category. For example, if *`idName`* is `userID`, provide the user ID. If *`idName`* is `userName`, provide the login name for the user acount.
- **get authentication types**
  Returns a list of identifiers that can be used to assign authentication types to users:
  `http://PC_host:8181/pc/center/webservice/users/authenticationTypes`

Available PUT Methods

NOTE
You can update the password of a specified user account using the PUT method. The password is sent in cleartext to avoid publicizing the encryption key for the web service to use. As a result, to protect the password privacy, use this method for changing the password only on NetOps Portal host.

- **update role**
  Updates the role assignment of a specified user account:
  `http://PC_host:8181/pc/center/webservice/users/idName/idValue/role/roleIdName/roleIdValue`
- **update time zone**

Updates the time zone of a specified user account:
`http://PC_host:8181/pc/center/webservice/users/idName/idValue/timeZone/newTimeZone`

- **set groups**
  Updates the permission groups that have been granted to a specified user account:
  `http://PC_host:8181/pc/center/webservice/users/idName/idValue/groups`
  To set groups, supply XML in the following format:

```
<group>
  <group ID="5245"/>
  <group ID="5246"/>
  <group ID="5247"/>
...
</group>
```

- **add groups**
  Adds the specified groups:
  `http://PC_host:8181/pc/center/webservice/users/idName/idValue/addGroups`
  To add groups, supply XML in the following format:

```
<group>
  <group ID="5245"/>
  <group ID="5246"/>
  <group ID="5247"/>
...
</group>
```

- **remove groups**
  Removes the specified groups:
  `http://PC_host:8181/pc/center/webservice/users/idName/idValue/removeGroups`
  To remove groups, supply XML in the following format:

```
<group>
  <group ID="5245"/>
  <group ID="5246"/>
  <group ID="5247"/>
...
</group>
```

- **set administered groups**
  Updates the branches of the groups tree to which the specified user has administrative access:
  `http://PC_host:8181/pc/center/webservice/users/idName/idValue/administeredGroups`
  The following example XML sets administered groups to a user account:

```
<group>
  <group ID="5245"/>
  <group ID="5246"/>
  <group ID="5247"/>
...
 </group>
```

- **set default group**
  Updates the default group for the specified user:
  `http://PC_host:8181/pc/center/webservice/users/idName/idValue/defaultGroup/groupId`
  Example:
  `http://PC.broadcom.net:8181/pc/center/webservice/users/userId/7/defaultGroup/11`
- **add administered groups**
  Adds the specified groups to a user account:
  `http://PC_host:8181/pc/center/webservice/users/idName/idValue/addAdministeredGroups`
  The following example XML adds administered groups to a user account:

```
<group>
  <group ID="5245"/>
  <group ID="5246"/>
  <group ID="5247"/>
...
 </group>
```

- **remove administered groups**
  Removes the specified groups from a user account:
  `http://PC_host:8181/pc/center/webservice/users/idName/idValue/removeAdministeredGroups`
  The following example XML removes administered groups from a user account:

```
<group>
  <group ID="5245"/>
  <group ID="5246"/>
  <group ID="5247"/>
...
 </group>
```

- **update product privilege per datasource**
  Updates the product privilege of a specified user account to enable access to the user interface of a specific data source:
  `http://PC_host:8181/pc/center/webservice/users/idName/idValue/ds/dsId/productPrivilege/newProductPrivilege`
- **dsId**
  The data source identifier. For a list of available data source identifiers, see Basic Datasource Parameters.

Available POST Methods

- **create**
  Creates a user account. This user is associated with the tenant for the logged-in user. The parameters include a role assignment:
  `http://PC_host:8181/pc/center/webservice/users/role/roleIdName/roleIdValue`
- **create in tenant**
  Creates a user account in the specified tenant, with the specified role assignment:
  `http://PC_host:8181/pc/center/webservice/users/tenant/tenantIdName/tenantIdValue/role/roleIdName/roleIdValue`
- **clear administered groups**
  Clears administered groups with a special ID that indicates no groups are selected:
  `http://PC_host:8181/pc/center/webservice/users/idName/idValue/administeredGroups`
  The following example XML clears administered groups:

```
<group>
  <group ID="2"/>
 </group>
```

Create a User Account

Use any REST client to create and configure a user account using the users web service. User accounts are associated with a tenant automatically. If you are deploying multi-tenancy, this user is assigned to the tenant of the authenticated user account that was used to make the REST service call. If you are not deploying multi-tenancy, this association is transparent to you; new user accounts are associated with the default tenant.

Follow these steps:

1. Set up a REST client with a connection to the NetOps Portal host.
2. Enter a URL for the NetOps Portal RESTful web services API in the REST client. Use the following format:

`http://PC_host:8181/pc/center/webservice/users/role/roleIdName/roleIdValue/`

- **roleIdName**
  Use the values that are specified at the following URL:
  `http://PC_host:8181/pc/center/webservice/roles/idNames`
  Examples:
  - `roleName`
  - `roleId`
- **roleIdValue**
  This value depends on the `roleIdName` that you selected. For example, if `roleName` is used, substitute a valid role name for `roleIdValue`.

  NOTE
  This role must be available within the tenant.

3. Select **POST** for HTTP Method.
4. Provide a valid **Username** and **Password** for a user account that has host or tenant administrator access to NetOps Portal.

   NOTE
   The password is automatically set to be the same as the user name.
5. Select **application/xml** as the **Body Content-type** in the Body settings.
6. Add the following parameters within the **Body** text section:

```
<user>
   <name>{UserName}</name>
   <description>{UserDescription}</description>
   <enabled>{UserEnabled}</enabled>
   <removable>{UserRemovable}</removable>
   <timezone>{UserTimeZone}</timezone>
   <culture>{UserCulture}</culture>
    <administeredGroups>
      <group ID="{groupID}"/>
      <group ID="{groupID}"/>
   </administeredGroups>
</user>
```

7. Replace values with the values that you want to use for this user account.
   For example, supply the following parameters:

```
<user>
   <name>Jane Doe</name>
   <description>User associated with the John Doe Corporation tenant.</description>
   <enabled>true</enabled>
   <removable>true</removable>
   <timezone>CST6CDT</timezone>
   <culture>en-US</culture>
   <administeredGroups>
      <group ID="105"/>
      <group ID="367"/>
   </administeredGroups>
</user>
```

   NOTE
   The `administeredGroups` tag is optional. To create a user without administered groups, exclude the tag.
8. Run the method.
9. Repeat the preceding steps until you have created as many users as you require.

User Account Product Privilege Settings

Use NetOps Portal to determine the product privilege of a specified user account. The *product privilege* is a type of permission set associated with a user account. The product privilege grants user access to features in selected data sources and does not apply to NetOps Portal functionality.

User accounts have one of the following product privilege assignments:

- **NONE**
  Indicates that this user does note have access to a specified data source.
- **ADMINISTRATOR**
  Indicates that this user has the administrator product privilege for the indicated data source and can perform administrative tasks.
- **POWER_USER**
  Indicates that this user has the power user product privilege for the indicated data source and can perform some administrative tasks associated with user accounts and dashboards.
- **USER**
  Indicates that this user has the user product privilege for the indicated data source and has no access to administrative features.

Tenants Web Service

The API provides the tenants web service to let you view, create, and modify tenant definitions.

The basic tenant definition contains a few parameters to identify the tenant. All of the infrastructure -- devices, networks, servers -- and all monitoring parameters for a customer's monitored systems must be associated with the tenant definition. Each tenant must contain at least one IP domain, plus as many of the following definitions as required to manage the associated enterprise infrastructure and applications:

- User accounts
- Roles
- Custom and system groups
- Custom reports
- Custom menus

To associate definitions and monitoring parameters with an existing tenant definition, log in as the tenant administrator and use the required web services to create the required definitions. The definitions are then associated with the tenant definition and available to users logged in with this tenant's user accounts.

A use case document is available on the CA NetOps Portal Documentation Bookshelf to help you use the tenants and users web services.

Tenants Web Service Parameters

Issue the following call to see the available parameters and operations for the tenants web service:

```
http://PC_host:8181/pc/center/rest/tenants/documentation
```

Parameters

- **tenantDescription**
  (Optional) Describes the tenant.
- **idName**

Is a name for the tenant.

- **status**
  Is the status of this tenant. Select one of the following values:
  – Activated: Enables tenant user accounts for use.
  – Disabled: Prevents any actions by user accounts that are associated with this tenant.
- **removable**
  States whether the item can be deleted (removed from the database).
  Values: true or false.
- **theme**
  Specifies the format -- the theme that controls the appearance of the page in the browser window -- to use for this tenant. All operators whose user account is associated with this tenant see this same theme. Two themes are available: CA-Blue and CA-Gray.
  Default: CA-Blue.
- **defaultCulture**
  Specifies a language (locale). Supply a language identifier from the following list:
  – en-US (English, United States)
  – ja-JP (Japanese)
  – zh-CN (Simplified Chinese)
  – fr-FR (French, France)
- **accountId**
  Identifies this tenant; usually corresponds to the MSP account number. If a value is supplied as input, it must be unique across all defined tenants.
- **tenantID**
  Is an internal (database) identifier for a tenant definition.

Tenants Web Service Example Syntax

The tenants web service lets you get a current list of tenant definitions, create new tenant definitions, and modify those definitions by changing their parameters.

A use case document is available on the CA NetOps Portal Documentation Bookshelf to help you use the tenants web service.

Operations

The following basic operations are supported by the tenants web service:

- **GET**
  Returns a list of tenant definitions sorted by name. Available on the /tenantID endpoint. Use the following syntax:

  ```
  http://PC_host:8181/pc/center/webservice/tenants/
  ```

- **POST**
  Creates a custom tenant. Use the following syntax:

  ```
  http://PC_host:8181/pc /center/webservice/tenants/
  ```

- **PUT**
  Updates an existing tenant definition. Use the following syntax:

  ```
  http://PC_host:8181/pc/center/webservice/tenants/
  ```

Create a Tenant

Use any REST client to create and configure a tenant using the tenants web service.

Follow these steps:

1. Set up a REST client with a connection to the NetOps Portal server.
2. Enter a URL for the NetOps Portal RESTful web services API in the REST client. Use the following format:

   ```
   http://PC_host:8181/pc/center/webservice/tenants/
   ```

3. Select **POST** for **"HTTP" Method**.
4. Provide a valid Username and Password for a user account that has global administrator access to NetOps Portal.
5. Select **'application/xml'** as the **'Body Content-type'** in the Body settings.
6. Add the following XML within the "Body" text section:

   ```
   <tenant>
     <tenantName>Name of tenant</tenantName>
     <tenantDesc>Description of the tenant</tenantDesc>
     <accountIdentifier>unique string for this tenant</accountIdentifier>
     <status>{activated or disabled}</status>
     <removable>{true or false}</removable>
     <theme>{CA-Blue or CA-Gray}</theme>
     <defaultCulture>culture</defaultCulture>
   </tenant>
   ```

7. Replace any values with the values that you want to use for the new tenant.
   For example, supply the following parameters:

   ```
   <tenant>
     <tenantName>John Doe</tenantName>
     <tenantDesc>John Doe Corporation tenant</tenantDesc>
     <accountIdentifier>JD1234</accountIdentifier>
     <status>Enabled</status>
     <removable>false</removable>
     <theme>CA-Blue</theme>
     <defaultCulture>en-US</defaultCulture>
   </tenant>
   ```

8. Run the method.
9. Repeat the preceding steps until you have created as many tenants as you require.

Supporting Web Services

The API provides the consoleinfo web service and the event web service. These web services supply configuration items and identifying data to the administrative web services.

Consoleinfo Web Service

You can retrieve information about NetOps Portal console configuration using the consoleinfo web service that the API provides. You can pass this information into other web service methods, such as time zone information that is required to create business hour definitions.

Consoleinfo Web Service Example Syntax

Issue the following call to see the parameters for the consoleinfo web service:

`http://PC_host:8181/pc/center/rest/consoleinfo/documentation`

Available GET Methods

- **get all time zones**
  Gets a list of all of the time zones that are available for use in business hour definitions:
  `http://PC_host:8181/pc/center/webservice/consoleinfo/allTimezones/cultureID`

  – **cultureId**
    Specifies a language (locale). Supply a language identifier from the following list:
    - en-US (English, United States)
    - ja-JP (Japanese)
    - zh-CN (Simplified Chinese)
    - fr-FR (French, France)
- **get installed language packs**
  Gets a list of language packs that are installed on the server:
  `http://PC_host:8181/pc/center/webservice/consoleinfo/languagepacks/cultureID`

- **get time zones assigned to sites**
  Retrieves a list of all of the time zones that are assigned to site groups to which the logged-in user has access:
  `http://PC_host:8181/pc/center/webservice/consoleinfo/timezonesAssignedToSites`

Event Web Service

You can retrieve a list of events that have been raised in your environment using the event web service.

Perform a GET to the following URL to see a list of events for the specified managed item:

`http://PC_host:8181/pc/eventId/item/itemId`

- **eventId**
  Event ID of the event to retrieve properties for.
- **itemId**
  Item ID of the item that the event is on.

For more information, see Devices Web Service.

Use Web Services to Create Tenants Programmatically

NetOps Portal offers a set of APIs that let you automate provisioning and configuration tasks. The most frequently repeated or time-consuming tasks are exposed to you with web services. Some of these APIs consist of web services that conform to the Representational State Transfer (REST) model.

This use case illustrates a procedure that an administrator can deploy to create multiple tenant definitions using the CA NetOps Portal RESTful web services. Because each tenant has its own user accounts to provide access to CA NetOps Portal, we also describe user account creation within tenants. In this use case, we describe the steps to take when using a REST client, a generic web services user interface application. The examples in this use case contain URIs that are constructed using the default server port, 8181.

Create Tenants Programmatically

The global administrator can use the CA NetOps Portal RESTful web services to create tenants with specific parameters. The basic tenant definition contains a few parameters to identify the MSP customer and let other operators access managed items and configuration for the customer. An administrator account is a required component of the tenant definition so that the customer can perform some tenant setup.

You can associate monitored devices and product settings with the tenant definition in separate steps. Each tenant must contain at least one IP domain. You and the tenant administrator can then configure other product settings that are required to manage the enterprise infrastructure and applications for that customer.

Create Tenants Using Web Services

Use any REST client to create and configure a tenant using the tenants web service.

Follow these steps:

1. Set up a REST client with a connection to the NetOps Portal server.
2. Enter a URL for the NetOps Portal RESTful web services API in the REST client. Use the following format:
   `http://PC_host:8181/pc/center/webservice/tenants/`
3. Select **POST** for **"HTTP" Method**.
4. Provide a valid Username and Password for a user account that has global administrator access to NetOps Portal.
5. Select **'application/xml'** as the **'Body Content-type'** in the Body settings.
6. Add the following XML within the "Body" text section:

```
<tenant>
  <tenantName>Name of tenant</tenantName>
  <tenantDesc>Description of the tenant</tenantDesc>
  <accountIdentifier>unique string for this tenant</accountIdentifier>
  <status>{activated or disabled}</status>
  <removable>{true or false}</removable>
  <theme>{CA-Blue or CA-Gray}</theme>
  <defaultCulture>culture</defaultCulture>
</tenant>
```

7. Replace any values with the values that you want to use for the new tenant.
   For example, supply the following parameters:

```
<tenant>
  <tenantName>John Doe</tenantName>
  <tenantDesc>John Doe Corporation tenant</tenantDesc>
  <accountIdentifier>JD1234</accountIdentifier>
  <status>Enabled</status>
  <removable>false</removable>
  <theme>CA-Blue</theme>
  <defaultCulture>en-US</defaultCulture>
</tenant>
```

   For more information about tenant parameters, see Tenants Service Example Syntax.
8. Run the method.
9. Repeat the preceding steps until you have created as many tenants as you require.

Create Users Using Web Services

Use any REST client to create and configure a user account using the users web service.

Every user account is automatically associated with a tenant. If you are deploying multi-tenancy, the new user is assigned to the tenant of the authenticated user account that was used to make the REST service call. If you are not deploying multi-tenancy, this association is transparent to you; new user accounts are associated with the Default Tenant.

Follow these steps:

1. Set up a REST client with a connection to the NetOps Portal server.
2. Enter a URL for the NetOps Portal RESTful web services API in the REST client. Use the following format:
   `http://PC_host:8181/pc/center/webservice/users/role/{roleIdName}/roleIdValue/`

   – ***{roleIdName}***
   Use values that are specified in `http://PC_host:8181/pc/center/webservice/roles/idName` s.
   Examples: '`roleName` ' and '`roleId` '.
   – ***roleIdValue***
   This value depends on the *`roleIdName`* that you selected. For example, if `'roleName'` is used, substitute a valid role name for *`roleIdValue`* .
   This role must be available within the tenant.
3. Select **POST** for **"HTTP" Method**.
4. Provide a valid Username and Password for a user account that has host or tenant administrator access to NetOps Portal.
5. Select **'application/xml'** as the **'Body Content-type'** in the Body settings.
6. Add the following parameters within the "Body" text section:

```
<user>
   <name>{UserName}</name>
   <description>{UserDescription}</description>
   <enabled>{UserEnabled}</enabled>
   <removable>{UserRemovable}</removable>
   <timezone>{UserTimeZone}</timezone>
   <culture>{UserCulture}</culture>
    <administeredGroups>
      <group ID="{GroupID}"/>
      <group ID="{GroupID}"/>
   </administeredGroups>
</user>
```

7. Replace any values with the values that you want to use for the new user account.
   For example, supply the following parameters:

```
<user>
   <name>Jane Doe</name>
   <description>User associated with the John Doe Corporation tenant.</description>
   <enabled>true</enabled>
   <removable>true</removable>
   <timezone>CST6CDT</timezone>
   <culture>en-US</culture>
    <administeredGroups>
      <group ID="105"/>
      <group ID="367"/>
   </administeredGroups>
</user>
```

   For more information about user parameters, see Users Web Service.

   NOTE

   The `administeredGroups` tag is optional. To create a user without administered groups, exclude the tag.

8. Run the method.
9. Repeat the preceding steps until you have created as many users as you require.

Basic User Account Parameters

The current values for the following user account settings are available from a GET operation:

`http://PC_host:8181/pc /center/webservice/users/cultureId.`

> **NOTE**
> This URL returns information about users associated with the same tenant as the account that is used to execute this command.

For *cultureId*, supply the language code for the language in which you would like to view the output, such as 'en-US'.

- **userID**
  Is an internally assigned value for the user account.
- **name**
  Is a login name for the user account. The name is limited to 50 characters.
- **description**
  (Optional) Describes the user account to help you identify it.
- **enabled**
  Determines whether the user account is enabled for use (activated).
- **removable**
  States whether the item can be deleted (removed from the database).
  Values: true or false.

  > **NOTE**
  > You cannot delete the two predefined user accounts (**admin** and **user**).

- **timezone (tz)**
  Corresponds to the time zone in which the user will view report data.
  Default: UTC (Coordinated Universal Time).
- userLevel
  Identifies the product privilege assigned to this user account.
- **role**
  Specifies the role assigned to the user account.
- **tenantId**
  Is an internal (database) identifier for the tenant with which the user account is associated.
- **culture**
  Specifies a language (locale). Supply a language identifier from the following list:
  - en-US (English, United States)
  - ja-JP (Japanese)
  - zh-CN (Simplified Chinese)
  - fr-FR (French, France)

  > **NOTE**
  >
  > : The GET method does not return password information. When you create a new user account, the password is automatically set to be the same as the user name.
  >
  > A separate PUT method lets you update the password of a specified user account. The password is sent in cleartext to avoid publicizing the encryption key for the web service to use. As a result, the method for changing the password must only be used on the server where CA NetOps Portal is installed to protect the password privacy.

User Account Product Privilege Settings

Use the CA NetOps Portal user interface to determine the product privilege of a specified user account. The *product privilege* is a type of permission set associated with a user account. The product privilege grants user access to features in selected data sources and does not apply to CA NetOps Portal functionality.

A user account has one of the following product privilege assignments:

- **NONE**
  Indicates that this user has no access to a specified data source.
- **ADMINISTRATOR**
  Indicates that this user has the administrator product privilege for the indicated data source and can perform administrative tasks.
- **POWER_USER**
  Indicates that this user has the power user product privilege for the indicated data source and can perform some administrative tasks associated with user accounts and dashboards.
- **USER**
  Indicates that this user has the user product privilege for the indicated data source and has no access to administrative features.

Use Web Services to Manage Groups

NetOps Portal offers a set of APIs that let you automate provisioning and configuration tasks. The most frequently repeated or time-consuming tasks are exposed to you with web services. Some of these APIs consist of web services that conform to the Representational State Transfer (REST) model.

This use case illustrates procedures that an administrator can deploy to create and manage groups or collections using the NetOps Portal REST web services. You can add and delete multiple groups or subgroups. You can add items to groups and create group rules to populate groups. You can also export and import group definitions.

In this use case, we describe the steps to take when using a REST client, a generic web services user interface application. The examples in this use case contain URIs that are constructed using the default server port, 8181.

Create Groups Programmatically

The global administrator can use the CA NetOps Portal REST web services to create and populate groups. Groups are always associated with a tenant definition. If you are not deploying multi-tenancy, groups are associated with the Default Tenant. Otherwise, group management tasks only apply to the current tenant.

The web service offers two methods of identifying groups:

- Using the full *group path*, which identifies the group as a subgroup of the default first-level or root group ('All Groups'). The group itself is a node on the *Groups tree*, which extends hierarchically from the root group.
- Using the *group ID*, an internally assigned numeric value that identifies a node. No identifier is assigned to the *path* to a group.
  Use this method to identify groups if they are nested within multiple containers to avoid typing the full group path.

You can create rules to add monitored devices to groups automatically. You can also add subgroups to your groups.

Get Identifying Information for a Group

You can use the group path or the group ID for a group to identify that group when you want to update it. The group path shows the position of a group within the Groups tree. The group ID is typically a better way to retrieve group information and to manage a group than using the group path, which can be lengthy.

Follow these steps:

1. Set up a REST client with a connection to the CA NetOps Portal server.
2. Use the following format for the URL in the REST client:
   `http://PC_host:8181/pc/center/webservice/groups/idName/idValue`

   – **IdName**
   A supported value for specifying a group. Possible values can be retrieved by performing a GET operation on the following URL:
   `http://PC_host:8181/pc/center/webservice/groups/idNames`

   Specify one of the following parameters for the **idName**:
   - **groupItemId** - The internally assigned ID of the group.
   - **groupPath** - The path of the group, with each group delimited by encoded front slashes (%2F).

   – **idValue**
   The value that is used to specify a group, based on the nature of the ID name. Dependent on the value of the **idName** field, as follows:
   groupItemId - The ID of the group is expected.
   groupPath - The path of the group, with each group delimited by encoded front slashes (%2F) is expected.

   Example: To view the details and rule definitions of a given group, issue a GET request to the parent of that group. The following call retrieves the details and rule definitions for the groups under a specific group that is named **ParentGroup** under **All Groups**:
   `http://PC_host:8181/pc/center/webservice/groups/groupPath/All%20Groups%2fParentGroup`

3. Select **GET** for the **HTTP method**.
4. Provide a valid Username and Password in the request header for a user account that has global administrator access to NetOps Portal.
5. Run the method.
   The XML that is returned contains the group item identifier.
   You can also use this method to get the full group path.

In the following example, the XML that is returned includes information about the system group named Inventory:

```
<?xml version="1.0" encoding="UTF-8"?>
<GroupTree id="5" inheritDefault="true" path="Inventory">
  <Group desc="This group contains groups for the various item types
  that are associated with all data sources." id="12" inherit="false"
  location="" name="All Items" type="automatic group">
    <Group desc="Includes networks from all Application Delivery
     Analysis data sources." id="40" inherit="true" location=""
     name="Application Delivery Analysis Networks" type="automatic
     group" />
    <Group desc="This group contains all the applications reported by
     each data source." id="41" inherit="true" location="" name=
     "Applications" type="automatic group" />
    <Group desc="This group contains all the device components reported
     by each data source." id="113" inherit="true" name="Device
     Components" type="automatic group" />
    <Group desc="This group contains all the VMware ESX hosts reported
     by each data source." id="35" inherit="true" location="" name="ESX
     Hosts" type="automatic group" />
    <Group desc="Includes interfaces from all data sources." id="50"
```

```
     inherit="true" location="" name="Interfaces" type="automatic group" />
    <Group desc="This group contains all the pingable devices reported
     by each data source." id="114" inherit="true" name="Pingable
     Devices" type="automatic group" />
    <Group desc="Includes routers from all data sources." id="31"
     inherit="true" location="" name="Routers" type="automatic group" />
    <Group desc="Includes servers from all data sources." id="32"
     inherit="true" location="" name="Servers" type="automatic group" />
    <Group desc="Includes switches from all data sources." id="33"
     inherit="true" location="" name="Switches" type="automatic group" />
    <Group desc="This group contains all the virtual machines reported
     by each data source." id="34" inherit="true" location=""
     name="Virtual Machines" type="automatic group" />
    <Group desc="This group contains all the voice interfaces reported
     by each data source." id="51" inherit="true" location="" name=
     "Voice Interfaces" type="automatic group" />
    <Group desc="This group contains all the voip locations reported
     by each data source." id="52" inherit="true" location="" name=
     "VoIP Locations" type="automatic group" />
  </Group>
  <Group desc="Includes every data source that has reported
  configuration information to the performance center." id="4"
  inherit="false" location="" name="Data Sources" type="automatic
  group">
    <Group desc="Contains configuration information reported to the
     performance center by Data Aggregator" id="115" inherit="true"
     name="da" type="system group" />
    <Group desc="Contains configuration information reported to the
     performance center by Event Manager" id="100" inherit="true"
     name="EventManager@servername.domain.com" type="system group">
      <Group desc="Devices created by event manager" id="101" inherit=
      "true" name="Devices" type="system group" />
    </Group>
  </Group>
```

The above results show that the group ID for the Inventory group is 5:

```
<GroupTree id="5" inheritDefault="true" path="Inventory">
```

You can use this value to get the complete group path. Enter the following URL:

```
http://PC_host:8181/pc/center/webservice/groups/groupItemId/5
```

From this result, you can see that the complete path to the Inventory group is "/All Groups/Inventory". All subgroups of the Inventory system group are also returned.

You can then use the values that are returned for the group name parameter to construct a group path for a subgroup, as in the following example:

```
/All%20Groups/Inventory/Device%20Components
```

The URL to get the group ID from the group path would resemble the following example:

```
http://PC_host:8181/pc/center/webservice/groups/groupPath/All%20Groups%2FInventory
%2FDevice%20Components
```

NOTE

When specifying a group path in the URL, use the percent encoded value for front slashes (%2F, as shown above) to delimit each group in the path. This requirement comes from the fact that non-encoded front slashes are reserved as the delimiting character for path segments in the URL syntax.

Get a List of Group Members

Use the groups web service to get a list of subgroups and items that are direct members of a specified group. The XML that is returned displays information about the specified group as the parent element, and separates groups and items into their own separate elements.

The list that is returned does not include items that were added to the group as children of a managed item that was directly added to the group. For example, if a router is a direct member of a group, the list does not include the interfaces that belong to that router.

To get a list of members for a group whose name or ID you know, enter the following base URL:

`http://PC_host:8181/pc/center/webservice/groups/idName/idValue/items`

- **idName**
  A supported ID name for specifying a group. Possible values can be retrieved by performing a GET operation on the following URL:
  `http://PC_host:8181/pc/center/webservice/groups/idNames.`
  Use one of the following parameters for the **idName**:
  - **groupItemId** - The internally assigned ID of the group.
  - **groupPath** - The path of the group, with each group delimited by encoded front slashes (%2F).
  - **idValue**
    The value that is used to specify a group, based on the nature of the ID name. Dependent on the value of the **idName** field, as follows:
    - **groupItemId** - The ID of the group is expected.
    - **groupPath** - The path of the group, with each group delimited by encoded front slashes (%2F) is expected.

Follow these steps:

1. Set up a REST client with a connection to the NetOps Portal server.
2. Use the following format for the URL in the REST client:
   `http://PC_host:8181/pc/center/webservice/groups/idName/idValue/items`
3. Select **GET** for the **HTTP** method.
4. Provide a valid Username and Password in the request header for a user account that has global administrator access to NetOps Portal.
5. Run the method.
   The XML that is returned lists subgroups and managed items that are members of the group. It resembles the following:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<group id="787" name="Test Group" description="" type="group" subType="user">
  <groups>
    <group id="788" name="Test Child Group 1" description="" type="group" subType="user"/>
    <group id="789" name="Test Child Group 2" description="" type="group" subType="user"/>
  </groups>
  <itemTypes>
```

```
    <itemType type="Devices">
      <items>
        <item id="121" name="Austin Switch" description="Cisco IOS Software, Switch 192.168.1.1"
type="device" subType="switches" addedBy="BY_USER"/>
        <item id="124" name="Austin Router" description="Cisco IOS Software, Router 192.168.0.1"
type="device" subType="router" addedBy="BY_USER"/>
        <item id="127" name="Austin Server" description="Linux" type="device" subType="server"
addedBy="BY_USER"/>
      </items>
    </itemType>
    <itemType type="Interfaces">
      <items>
        <item id="417" name="eth0/1/0:7" description="Ethernet0/1/0:7" type="interface" subType="physical"
addedBy="BY_USER"/>
        <item id="418" name="eth0/1/0:8" description="Ethernet0/1/0:8" type="interface" subType="physical"
addedBy="BY_USER"/>
        <item id="420" name="eth0/1/0:9" description="Ethernet0/1/0:9" type="interface" subType="physical"
addedBy="BY_USER"/>
      </items>
    </itemType>
  </itemTypes>
</group>
```

Get a List of Groups that You Own

You can only modify or delete groups that you own. If you created a group, you are the owner of that group. The global administrator is the owner of all groups in the Groups tree. In addition, the global administrator or tenant administrator can edit your user account to give you ownership of a "branch" of the Groups tree.

Use the Users web service to get a list of the groups of which you are the owner. Consult the following descriptions to use the Users web service:

Issue the following call to see the parameters and available operations for the users web service:

`http://PC_host:8181/pc/center/rest/users/documentation`

Available GET Methods

- **get groups owned by user**
  Retrieves a list of groups for the specified user. The groups that are returned are groups that are owned by the specified user, meaning that this user can modify or delete these groups:
  `http://PC_host:8181/pc/center/webservice/users/idName/idValue/groupsOwnedByUser`

- **get groups**
  Retrieves a list of groups to which the specified user has view access. The groups that are returned are groups that are within the permission set of the specified user. The user cannot modify or delete these groups:
  `http://PC_host:8181/pc/center/webservice/users/idName/idValue/groups`

- **get id names**
  Retrieves a list of identifiers that can be used to identify users in other web service methods.
  `http://PC_host:8181/pc/center/webservice/users/idNames`

  - **idValue**
    Is the value for the identifying category. For example, if ***idName*** is **userID**, provide the user ID. If ***idName*** is **userName**, provide the user name.
- **get authentication types**
  Returns a list of identifiers that can be used to assign authentication types to users.
  `http://PC_host:8181/pc/center/webservice/users/authenticationTypes`

Available PUT Methods

- **update role**
  Updates the role assignment of a specified user account.
  `http://PC_host:8181/pc/center/webservice/users/idName/idValue/role/roleIdName/roleIdValue`

- **update time zone**
  Updates the time zone of a specified user account.
  `http://PC_host:8181/pc/center/webservice/users/idName/idValue/timeZone/newTimeZone`

- **set groups**
  Updates the permission groups that have been granted to a specified user account.
  `http://PC_host:8181/pc/center/webservice/users/idName/idValue/groups`

- **set administered groups**
  Updates the groups of a specified user account.
  `http://PC_host:8181/pc/center/webservice/users/idName/idValue/administeredGroup`

  To add administered groups to a user account, supply XML in the following format (for example):

```
<groups>
  <group ID="5245"/>
  <group ID="5246"/>
  <group ID="5247"/>
...
 </groups>
```

- **update product privilege per datasource**
  Updates the product privilege of a specified user account to enable access to the user interface of a specific datasource.
  `http://PC_host:8181/pc/center/webservice/users/idName/idValue/ds/dsId/productPrivilege/newProductPrivilege`

  - **dsId**
    The data source identifier. For a list of available data source identifiers, see Basic Datasource Parameters. For more information about product privileges, see User Account Product Privilege Settings.

Available POST Methods

- **create**
  Creates a new user account. The new user is associated with the tenant for the logged-in user. The parameters include a role assignment.
  `http://PC_host:8181/pc/center/webservice/users/role/roleIdName/roleIdValue`

- **create in tenant**
  Creates a new user account in the specified tenant, with the specified role assignment.
  `http://PC_host:8181/pc/center/webservice/users/tenant/tenantIdName/tenantIdValue/role/roleIdName/roleIdValue`

Create Groups Using Web Services

Use any REST client to create and configure a group associated with the Default Tenant using the groups web service. The steps to create groups within a custom tenant are slightly different. You can supply the group ID or the group path as parameters.

The basic URL to create a group is as follows:

`http://PC_host:8181/pc/center/webservice/groups/useIds/allowDeletes/items`

- **useIds**
  Indicates that the **groupItemId** parameter is used to identify the group. Indicates whether or not the supplied id attribute of a group should be used to identify it. A value of 'false' ensures that the groups web service does not try to use the IDs of the groups to create any new groups. Because they are internally assigned, the IDs are a less reliable way to identify groups that have been exported and reimported.
- **allowDeletes**
  Enables deletion of the group that you are creating. Lets the groups web service update any rules that are defined in any existing groups that are overwritten by this XML document.

Example:

`http://PC_host:8181/pc/center/webservice/groups/groupPath/All%20Groups%2FAustin`

Follow these steps:

1. Set up a REST client with a connection to the NetOps Portal server.
2. Use the following format for the URL in the REST client:
   `http://PC_host:8181/pc/center/webservice/groups/false/false`
3. Select **POST** for **"HTTP" Method**.
4. Provide a valid Username and Password in the request header for a user account that has global administrator access to NetOps Portal.
5. Select **'application/xml'** as the **'Body Content-type'**.
6. Add the following XML within the "Body" text section, replacing the values with the values that you want to use for the new group:

```
<GroupTree path="/All Groups">
    <Group name="Group Name" desc="Description of the group"
    inherit="true" type="user group"/>
</GroupTree>
```

   - **inherit**
     Indicates whether the group includes child items of group members. For example, if the "inherit" attribute is set to true, device interfaces are group members if the device has been added to the group.
   - **type**

Indicates the type of group. The following values are supported.
Values:

- **user group** (default)
  A group that a user has created.
- **site**
  A user-created group that represents a physical site.

For example, supply the following XML:

```
<GroupTree path="/All Groups">
    <Group name="USA" desc="Group to represent the entire
    United States" inherit="true" type="user group"/>
</GroupTree>
```

7. Run the method.
   A new group, 'USA,' is created under the default 'All Groups' group in the Groups tree.
8. Repeat the preceding steps to create a site group. Supply the following XML for a site group:

```
<Group name="USA-Site" desc="Site group to represent the
entire United States" inherit="true" type="site group"
location="North America"/>
```

9. Repeat these steps until you have created as many groups and site groups as you require.
10. Use a similar procedure to create subgroups. Add the following XML within the "Body" text section:

```
<GroupTree path="/All Groups/USA">
    <Group name="Raleigh" desc="This is the group for
    managed items in Raleigh, NC" inherit="true"
    type="user group"/>
</GroupTree>
```

11. Run the method.
    In this example, a new group, "Raleigh," is added as a subgroup of the "All Groups\USA" group.

Create Groups in a Custom Tenant

The procedure for creating groups within a custom tenant definition is slightly different from the procedure to create groups in the Default Tenant.

Supply the group ID or the group path that you retrieved in a previous procedure as parameters. But create the group in the "`/All Groups/Tenants/` " path.

Here is an example of how to add groups to a non default tenant. In the user interface, the result looks like the following image:

Follow these steps:

1. Set up a REST client with a connection to the NetOps Portal server.
2. Use the following format for the URL in the REST client:

   `http://PC_host:8181/pc/center/webservice/groups/false/false`

3. Select **POST** for **"HTTP" Method**.
4. Provide a valid Username and Password in the request header for a user account that has global administrator access to NetOps Portal.
5. Select **'application/xml'** as the **'Body Content-type'**.
6. Add the following XML within the "Body" text section, replacing the values with the values that you want to use for the new group:

```
<GroupTree path="/All Groups/Tenants/MyCustomTenant/Groups">
    <Group name="North Carolina" desc="NC in the Southeast Region" inherit="true" type="user group"/>
```

```
</GroupTree>
```

7. Run the method.
   A new group, 'North Carolina,' is created under the custom tenant, MyCustomTenant. Only the global administrator and the administrator for the custom tenant can manage this group, and only MyCustomTenant users can see this group.

Add Items to Groups

Add individual managed items to groups using the groups web service. The item ID for each item is required.

Start by using the devices web service to get a list of managed items in the database and their IDs. The get id names method returns a list of identifiers that can be used in other methods to identify devices. A submethod, get interfaces, returns a list of device interfaces. You can filter the results by item subtype.

Checking is performed to avoid item duplication and make sure the group is valid. If any of these steps fail, then the service will exit out with an error:

- The group exists.
- The user who authenticated with the web service has access permissions to the group.
- The group can have items added to it. You can add items of the following subtypes to groups: user, site, service provider-defined items.

Each item that is specified in the list is also validated according to the following criteria:

- The list of items to add does not contain duplicate IDs. Nor does the list contain the IDs of any items that are already members of the target group.
- An item that corresponds to the specified ID exists.
- The item is NOT a group. If a user wants to add a group to an existing group, there are other services for that.
- The user has access to (permission to view) that item.

The XML that is returned shows the results of the validation. They include a report of the items that were added and the items that were not added.

Follow these steps:

1. Set up a REST client with a connection to the NetOps Portal server.
2. Use the following format for the URL in the REST client:
   `http://PC_host:8181/pc/center/webservice/devices/idNames`
3. In the XML that is returned, locate the device IDs for the devices to add to the target group.
4. Now type the following URL into the REST client:
   `http://PC_host:8181/pc/center/webservice/groups/idName/idValue/items`

   Example:
   `http://PC_host:8181/pc/center/webservice/groups/groupPath/All%20Groups%2FmyGroup/items`

   – **idName**
   A supported ID name for specifying a group. Possible values can be retrieved by performing a GET operation on the following URL:
   `http://PC_host:8181/pc/center/webservice/groups/idNames`
   Use one of the following parameters for the **idName**:

- **groupItemId** - The internally assigned ID of the group.
- **groupPath** - The path of the group, with each group delimited by encoded front slashes (%2F).

– **idValue**
The value that is used to specify a group, based on the nature of the ID name. Dependent on the value of the **idName** field, as follows:
groupItemId - The ID of the group is expected.
groupPath - The path of the group, with each group delimited by encoded front slashes (%2F) is expected.

5. Select **POST** for the **HTTP method**.
6. Provide a valid Username and Password in the request header for a user account that has global administrator access to NetOps Portal.
7. Select **'application/xml'** as the **'Body Content-type'**.
8. Add the following XML within the "Body" text section:

```
<items>
  <item id="value"/>
</items>
```

For example, enter something like the following to add a single router with managed item ID 684 to the group named "Austin":

```
<items>
  <item id="684"/>
</items>
```

You can also submit a list of managed items to add to a single group.

9. For example, add XML like the following to add five items:

```
<items>
  <item id="123"/>
  <item id="234"/>
  <item id="345"/>
  <item id="456"/>
  <item id="567"/>
</items>
```

The items that you specified in your list are added to the group.

10. Use the product user interface to verify group membership.

Add Group Rules

Add rules to groups that determine the managed items to include. As items are discovered, they are automatically added to the groups whose rules apply to those items.

The same URL that is used to create groups is used to add group rules. The body of the XML in the POST is modified to include rules. The web service adds XML to the group definition to create the rules.

Follow these steps:

1. Set up a REST client with a connection to the NetOps Portal server.
2. Use the following format for the URL in the REST client:

```
http://PC_host:8181/pc/center/webservice/groups/idName/idValue/groupItemId
```

3. Select **POST** for the **HTTP method**.
4. Provide a valid Username and Password in the request header for a user account that has global administrator access to NetOps Portal.
5. Select **'application/xml'** as the **'Body Content-type'**.
6. Add the following XML within the "Body" text section, replacing the values with the values that you want to use for the group rule:

```
<GroupTree path="/All Groups/USA">
   <Group name="Austin" desc="The group for items in Austin, TX">
      <Rules allowDeletes="true" saveRules="true">
         <Rule add="device" name="Add Devices">
            <Match>
               <Compare readOnly="true" using="MEMBER_OF">
                  <Property name="ItemID" type="device"/>
                     <Value reference="/All Groups">1</Value>
               </Compare>
               <Compare readOnly="false" using="EQUALS">
                  <Property name="AlternateName" type="device"/>
                     <Value>Cisco-3345</Value>
               </Compare>
            </Match>
         </Rule>
      </Rules>
   </Group>
</GroupTree>
```

This example creates a rule that adds a device named 'Cisco-3345' to the group named "Austin".

7. Run the method.
8. Repeat the preceding steps until you have created as many rules as you require.

We recommend using the Manage Groups page to create a few rules before you use the groups web service to create them. You can view the syntax for a rule that you have applied by taking the steps in Get Identifying Information for a Group. The XML that is returned includes group rules.

Remove Items from Groups

You can remove individual items from a group using the rules feature of the groups web service. You must first use the Manage Groups page in the CA NetOps Portal user interface to view current group membership. Then apply a new rule to the group that specifies the **allowDeletes** property. When you post the rule, it deletes items that are already in the group but that do not meet the criteria that the rule specifies.

Follow these steps:

1. Set up a REST client with a connection to the NetOps Portal server.
2. Use the following format for the URL in the REST client:

   `http://PC_host:8181/pc/center/webservice/groups/false/true`

   NOTE
   Specifying 'true' for the final property lets you delete items from the group.
3. Select **POST** for the **HTTP method**.
4. Provide a valid Username and Password in the request header for a user account that has global administrator access to NetOps Portal.
5. Select **'application/xml'** as the **'Body Content-type'**.
6. Add the following XML within the "Body" text section, replacing the values with the values that you want to use for the group rule:

```
<GroupTree path="/All Groups/USA">
   <Group name="Austin" desc="The group for items in Austin, TX">
      <Rules allowDeletes="true" saveRules="true">
         <Rule add="device" name="Add Devices">
            <Match>
```

```
                <Compare readOnly="true" using="MEMBER_OF">
                   <Property name="ItemID" type="device"/>
                      <Value reference="/All Groups">1</Value>
                </Compare>
                <Compare readOnly="false" using="EQUALS">
                   <Property name="subType" type="device"/>
                      <Value>server</Value>
                </Compare>
                <Compare readOnly="false" using="NOT_EQUALS">
                   <Property name="subType" type="device"/>
                      <Value>VM</Value>
                </Compare>
             </Match>
          </Rule>
       </Rules>
    </Group>
 </GroupTree>
```

This example creates a rule that adds devices with subtype 'server' but removes devices with subtype 'VM' (virtual machine) from the group named "Austin".

7. Run the method.
8. Use the Manage Groups page in the CA NetOps Portal user interface to verify group membership.

Delete Groups

The groups web service does not offer a DELETE method to delete a user group or a custom collection based on the group ID. Deletion of groups is possible using the update method. You perform another POST of the entire XML for a group with the group elements that need to be deleted removed from the XML. Their absence, and the **allowDeletes** parameter, removes them.

Set **allowDeletes** on a group that is one level higher in the tree to ‘true’ to ensure that a subgroup is deleted if no entry for that group is included in the XML. This attribute is inherited by all subgroups. It is not applied to the parent group itself.

You can only delete user groups. System groups and out-of-the-box collections cannot be deleted. To delete groups that you defined in My User Groups, use the user interface.

WARNING
The user groups that you delete can only be restored by recreating them.

You can supply the group ID or the group path that you retrieved in a previous procedure as parameters.

The basic URL to create a group that allows deletion is as follows:

```
http://PC_host:8181/pc/center/webservice/groups/useIds/allowDeletes
```

Example:

In Create Groups Using Web Services, the example XML created a group named 'Raleigh' that was a subgroup of a group named 'USA'. The full XML for this task would be as follows:

```
<GroupTree path="/All Groups">
    <Group name="USA" desc="Group to represent the entire United
    States" inherit="true" type="user group">
       <Group name="Raleigh" desc="This is the group for managed items
       in Raleigh, NC" inherit="true" type="user group"/>
    </Group>
</GroupTree>
```

The XML had to be posted to the following URL to enable future deletion of these groups:

```
http://PC_host:8181/pc/center/webservice/groups/false/true/
```

Now delete the Raleigh subgroup.

Follow these steps:

1. Set up a REST client with a connection to the NetOps Portal server.
2. Enter a URL for the NetOps Portal REST web services API in the REST client. Use the following format:
   `http://PC_host:8181/pc/center/webservice/groups/false/true/`
3. Select **POST** for **"HTTP" Method**.
4. Provide a valid Username and Password in the request header for a user account that has global administrator access to NetOps Portal.
5. Select **'application/xml'** as the **'Body Content-type'**.
6. Add the following XML within the "Body" text section. Replace the values with values to identify the groups that contain the groups to be deleted:
   ```
   <GroupTree path="/All Groups">
       <Group name="USA" desc="Group to represent the entire United
       States" allowDeletes="true" type="user group"/>
   </GroupTree>
   ```
7. Run the method.
   The web service updates the XML for the specified group (All Groups\USA) to remove the Raleigh subgroup. The Raleigh group is thus deleted from the Groups tree.

Import Group Definitions

Use any REST client to import a list of group definitions using the groups web service. To import group definitions into a new database, you must first retrieve, and then export, all of your existing definitions.

You can supply the group ID or the group path that you retrieved in a previous procedure as parameters.

The basic URL to retrieve group definitions is as follows:

```
http://PC_host:8181/pc/center/webservice/groups/groupPath/All%20Groups
```

That method uses the group path name to identify the root group. You can also retrieve the "All Groups" structure by group item ID:

```
http://PC_host:8181/pc/center/webservice/groups/groupItemId/1
```

Prepare to export all of the groups that you have defined in NetOps Portal by performing a retrieval of the "All Groups" group. Retrieving this root group calls up the entire tree in the XML that is returned.

Follow these steps:

1. Set up a REST client with a connection to the NetOps Portal server.
2. Use the following format for the URL in the REST client:
   `http://PC_host:8181/pc/center/webservice/groups/groupItemId/1`
3. Select **GET** for **"HTTP" Method**.
4. Provide a valid Username and Password in the request header for a user account that has global administrator access to NetOps Portal.

5. Select **'application/xml'** as the **'Body Content-type'**.
6. Run the method.
   The XML document that is returned defines your entire group structure.
   You are now ready to import the group definitions into another NetOps Portal database.
7. Open a connection to the NetOps Portal server from a REST client.
8. Select **POST** for **"HTTP" Method**.
9. Use the following format for the URL:
   `http://PC_host:8181//pc/center/webservice/groups/false/true`

   The 'false' and 'true' values refer to the following parameters:
   - **useIds**
     Indicates whether the supplied *id* attribute of a group is used to identify it. Ensures that the groups web service does not try to use the IDs of the groups to create any new groups. Because they are internally assigned, the IDs are a less reliable way to identify groups that have been exported and reimported.
   - **allowDeletes**
     Enables deletion of the group that you are creating. Lets the groups web service update any rules that are defined in any existing groups that are overwritten by this XML document.
10. Run the method.
    The groups that you exported are added as subgroups of the "All Groups" root group.
11. Use the user interface to verify the structure of the imported groups (Admin, Group Settings, Groups).

Use Web Services to Manage Business Hours

This use case illustrates procedures that an administrator can deploy to create and manage business hours definitions to filter views using the NetOps Portal RESTful web services. NetOps Portal administrators can enhance reporting by creating sets of business hours definitions. Business hours help product operators prioritize their troubleshooting workload by highlighting events that occur during business-critical time periods.

Use the groups web service to create site groups, and use the business hours web service to manage business hours definitions. If you add business hours first, you can then associate them with site groups during site-group creation.

You can add and delete multiple business hours definitions and assign them to site groups.

In this use case, we describe the steps to take when using a REST client, a generic web services user interface application. The examples in this use case contain URIs that are constructed using the default server port, 8181.

Create Site Groups Using Web Services

Use any REST client to create and configure a site group that is associated with the Default Tenant using the groups web service. With the groups web service, you can create rules and apply them to site groups so that items are added automatically. The steps to create groups within a custom tenant are slightly different. You can supply the group ID or the group path as parameters.

The basic URL to create a group is as follows:

`http://PC_host:8181/pc/center/webservice/groups/useIds/allowDeletes`

`useIdsIndicates` that the `groupItemId` parameter is used to identify the group. Indicates whether the supplied id attribute of a group should be used to identify it. A value of `false` ensures that the groups web service does not try to use the IDs of the groups to create any new groups. Because they are internally assigned, the IDs are a less reliable way to identify groups that have been exported and reimported. In this example, the XML does not contain a group ID, so the value is 'false'.

- **allowDeletes**

Enables deletion of the group that you are creating. The groups web service can update the rules that are defined in the existing groups that are overwritten by this XML document.

Example:

```
http://PC_host:8181/pc/center/webservice/groups/false/true
```

Follow these steps:

1. Set up a REST client with a connection to the NetOps Portal server.
2. Use the following format for the URL in the REST client:
   ```
   http://PC_host:8181/pc/center/webservice/groups/false/true
   ```
3. Select **POST** for **"HTTP" Method**.
4. Provide a valid Username and Password in the request header for a user account that has global administrator access to NetOps Portal.
5. Select **'application/xml'** as the **'Body Content-type'**.
6. Add the following XML within the "Body" text section, replacing the values with the values that you want to use for the new site group:
   ```
   <GroupTree path="/All Groups">
      <Group name="East Coast USA" desc="This is a site group"
      inherit="true" type="site group" location="North America"
      bHourID="99990" timeZone="EST"/>
   </GroupTree>
   ```
   - **inherit**
     Indicates whether the group includes child items of group members. For example, if the "inherit" attribute is set to true, device interfaces are group members if the device has been added to the group.
   - **type**
     Indicates the type of group. Accepts the following values:
     - **custom group**: A group that a user has created.
     - **site group**: A user-created group that represents a physical site.
     - **system group**: A predefined group that cannot be modified or deleted.
     - **automatic group**: A predefined group of items from a data source other than Data Aggregator that cannot be modified or deleted.
   - **bHourID**
     The internally assigned identifier of the business-hour definition that you created previously.
   - **timeZone**
     The time zone to associate with this site group.

   For example, supply the following XML:
   ```
   <Group name="East Coast" desc="Site group to represent the
       entire United States" inherit="true" type="site group" location="North
       America" bHourID="99990" timeZone="EST"/>
   ```
7. Run the method.
   The USA site group is created under the default All Groups group in the Groups tree.
8. Repeat the preceding steps until you have created as many site groups as you require.
9. Use a similar procedure to create subgroups. Add the following XML within the "Body" text section:
   ```
   <GroupTree path="/All Groups/USA">
       <Group name="Raleigh" desc="This is the group for
       managed items in Raleigh, NC" inherit="true"
       type="custom group"/>
   </GroupTree>
   ```
10. Run the method.
    In this example, a new group, "Raleigh," is added as a subgroup of the "All Groups\USA" group.

Manage Time Zone Associations

You can manage associations of time zones with site groups using web services.

Assign a time zone to a site group using the following URL in a PUT operation:

```
http://PC_host:8181/pc/center/webservice/businesshours/assign/timezone/site/siteGroupId
```

Use the following syntax in a PUT operation to remove the assignment:

```
unassign/timezone/site/siteGroupId
```

Use the following syntax to get a list of time zones that are assigned to site groups:

```
http://PC_host:8181/pc/center/webservice/businesshours/timezonesAssignedToSites
```

Remove a time zone association by running the same method with the `unassign/timezone` syntax.

Follow these steps:

1. Using the REST client with a connection to the NetOps Portal server, enter a URL for the NetOps Portal RESTful web services API in the REST client. Use the following format:
   ```
   http://PC_host:8181/pc/center/webservice/businesshours/assign/timezone/site/siteGroupId
   ```
2. Select **PUT** for **"HTTP" Method**.
3. Provide a valid Username and Password for a user account that has administrator access to NetOps Portal.
4. Select **'application/xml'** as the **'Body Content-type'** in the Body settings.
5. Add the following XML within the "Body" text section:
   ```
   <GroupTree path="/All Groups">
       <Group name="East Coast USA" desc="This is a site group"
      inherit="true" type="site group" location="North America"
      bHourID="99990" timeZone="EST"/>
   </GroupTree>
   ```
   - **bHourID**
     The internally assigned identifier of the business-hour definition that you created previously.
   - **timeZone**
     The time zone to associate with this site group.
6. Run the method.
   If it does not already exist, the USA site group is created under the default All Groups group in the Groups tree. The Eastern Standard timezone (EST) is assigned to this site group.
7. Repeat the preceding steps until you have associated time zones with all site groups to which you plan to apply business hours.

Create a Business Hours Definition

You can create business hours definitions using the business hours web service and any REST client. For this procedure, you can log in as a global administrator or as a tenant administrator. The global administrator creates the business hours definitions within the Default Tenant, while the tenant administrator creates the definition within that tenant.

The steps to create business hours within a custom tenant are slightly different. You can supply the tenant ID as a parameter.

Business hours definitions comprise a starting hour and an ending hour. The `startHour` and `endHour` parameters in the XML must be the same for all days to avoid an error in any POST or PUT operation.

For systems with multiple tenants, specify the tenant in the URL. To get a list of tenant IDs, perform a GET to the following URL:

```
http://PC_host:8181/pc/center/webservice/tenants/idNames
```

NOTE
You can also modify business hours within a custom tenant by performing a PUT operation to the following URL:
`/businesshours/tenantId/tenant Id/id/id`
where `id` is the ID of the business-hour definition.

Follow these steps:

1. Using the REST client with a connection to the NetOps Portal server, enter a URL for the NetOps Portal RESTful web services API in the REST client. Use the following format:
   - **Default Tenant:**
     `http://PC_host:8181/pc/center/webservice/businesshours/`
   - **Specific Tenant**
     `http://PC_host:8181/pc/center/webservice/businesshours/tenantId/tenant_ID`
2. Select **POST** for **"HTTP" Method**.
3. Provide a valid Username and Password for a user account that has global administrator or tenant administrator access to NetOps Portal.
4. Select **'application/xml'** as the **'Body Content-type'** in the Body settings.
5. Add the following XML within the "Body" text section:

```
<BusinessHour>
       <Name>Bakery</Name>
       <Description>HEB Bakery</Description>
       <Monday>
<HourRange startHour="5" endHour="12"/>
</Monday>
       <Tuesday>
<HourRange startHour="5" endHour="12"/>
</Tuesday>
       <Wednesday>
<HourRange startHour="5" endHour="12"/>
</Wednesday>
       <Thursday>
<HourRange startHour="5" endHour="12"/>
</Thursday>
       <Friday>
<HourRange startHour="5" endHour="12"/>
</Friday>
       <Saturday/>
       </Sunday>
</BusinessHour>
```

   In this example, a business-hour definition named HEB Bakery is created. The business hours start at 5 a.m. and end at noon. Saturday and Sunday are excluded.
6. Run the method.
7. Repeat the preceding steps until you have created as many business hours definitions as you require.

Manage Business Hours Associations

To apply a business hours filter, associate a business hours definition with a site group.

Use the following URL in a PUT operation to assign business hours to a site group:

`http://PC_host:8181/pc/center/webservice/businesshours/assign/businesshour/businessHourId/site/siteGroupId`

NOTE
NetOps Portal validates the site group for an associated time zone by performing a verification. If the site group does not have a time zone assignment, an error message appears.

Use the following syntax in a PUT operation to remove the assignment:

```
unassign/businesshour/site/siteGroupId
```

Follow these steps:

1. Using the REST client with a connection to the NetOps Portal server, enter a URL for the NetOps Portal RESTful web services API in the REST client. Use the following format:
   ```
   http://PC_host:8181/pc/center/webservice/businesshours/assign/businesshour/businessHourId/site/siteGroupId
   ```
   – **businessHourId**
   The internally assigned identifier of the business-hour definition that you created previously.
2. Select **PUT** for **"HTTP" Method**.
3. Provide a valid Username and Password for a user account that has administrator access to NetOps Portal.
4. Select 'application/xml' as the 'Body Content-type' in the Body settings.
5. Add the following XML within the "Body" text section:
   ```
   <GroupTree path="/All Groups">
       <Group name="East Coast USA" desc="This is a site group"
      inherit="true" type="site group" location="North America"
      bHourID="99990" timeZone="EST"/>
   </GroupTree>
   ```
6. Run the method.
   If it does not already exist, the USA site group is created under the default All Groups group in the Groups tree. The business-hour definition with ID 99990 is assigned to this site group.
7. Repeat the preceding steps until you have associated time zones with all site groups to which you plan to apply business hours.

You can remove a business hour association by running the same method with the `unassignbusinesshour` syntax.

Query the RIB to Return a View with Business-Hour Filtering

You can return data for a specific metric by entering queries into a web browser by querying the Report Information Base (RIB). This example presents a NetOps Portal RIB query that returns Top Discards data from a data aggregator data source.

Precede the NetOps Portal RIB queries with the following URL:

```
http://PC_host:8481/dm/rib/query/
```

You can append URL parameters to specify property values:

```
http://PC_host:8481/dm/rib/query/ribquery/?property1=value1&property2=value2
```

The following RIB query returns Top Discards data from a data aggregator data source:

```
http://<server IP address>:port/dm/rib/query/
SELECT .PollItem.ID, .PollItem.DevDisplayName, .Item.DisplayName, .Discards.Sum, .DiscardsIn.Sum, .DiscardsOut.Sum
 FROM CA.IM.DA.MF.NormalizedPortInfo.IFSTATS WHERE .Group.GroupID = 1039 AND .EndTime(300) > 1366208760
 AND .EndTime(300) <= 1366212360 GROUPBY .PollItem.ID, .Item.DisplayName, .PollItem.DevDisplayName
 ORDERBY .Discards.Sum DESC LIMIT 10
```

TIP
If necessary, you can escape the RIB query and parameters in your web browser, for example:

```
http://PC_host:port/dm/rib/query/SELECT%20.PollItem.ID,
%20.PollItem.DevDisplayName,%20.Item.DisplayName,
%20.Discards.Sum,%20.DiscardsIn.Sum,%20.DiscardsOut.Sum%20FROM
%20CA.IM.DA.MF.NormalizedPortInfo.IFSTATS%20WHERE%20.Group.GroupID%20=
%201039%20AND%20.EndTime(300)%20%3E%201366208760%20AND%20.EndTime(300)%20%3C=
%201366212360%20GROUPBY%20.PollItem.ID,%20.Item.DisplayName,
%20.PollItem.DevDisplayName%20ORDERBY%20.Discards.Sum%20DESC%20LIMIT%2010
```

Add the following URL parameters to return Top Discards data for a set of business hours in a specific time zone. NetOps Portal administrators configure sets of business hours.

NOTE
Some queries support only data filtering by time zone and business hours.

- **RIB.TimeZone**
  Is the string identifier of the time zone used to filter data results.
- **RIB.BusinessHours**
  Is the NetOps Portal ID of the business hour definition used to filter data results. Include this parameter in the `propertiesToTranslate` value to ensure that the ID is translated. IDs that are not translated are submitted unchanged to each applicable data source.
- **propertiesToTranslate**
  Is a list of parameter names whose values contain a NetOps Portal ID to translate to a local data source ID.

Example 1

To return data filtered by time zone, add the time zone parameter (shown in bold text) to the URL. In the following example, the data is filtered to include only data for items in sites configured for the America/New_York time zone:

```
http://pchost:8481/dm/rib/query/
SELECT .PollItem.ID, .PollItem.DevDisplayName, .Item.DisplayName, .Discards.Sum, .DiscardsIn.Sum, .DiscardsOut.Sum
 FROM CA.IM.DA.MF.NormalizedPortInfo.IFSTATS WHERE .Group.GroupID = 1039 AND .EndTime(300) > 1366208760
 AND .EndTime(300) <= 1366212360 GROUPBY .PollItem.ID, .Item.DisplayName, .PollItem.DevDisplayName
 ORDERBY .Discards.Sum DESC LIMIT 10?RIB.TimeZone=America/New_York
```

Example 2

To return data filtered by time zone and business hours, add the time zone and business hours parameters (shown in bold text) to the URL. In the following example, the data is filtered to include only data for items in sites configured for the America/New_York time zone and business hours matching the NetOps Portal definition for ID 6434:

```
http://pchost:8481/dm/rib/query/
SELECT .PollItem.ID, .PollItem.DevDisplayName, .Item.DisplayName, .Discards.Sum, .DiscardsIn.Sum, .DiscardsOut.Sum
 FROM CA.IM.DA.MF.NormalizedPortInfo.IFSTATS WHERE .Group.GroupID = 1039 AND .EndTime(300) > 1366208760
 AND .EndTime(300) <= 1366212360 GROUPBY .PollItem.ID, .Item.DisplayName, .PollItem.DevDisplayName
 ORDERBY .Discards.Sum DESC LIMIT 10?RIB.TimeZone=America/
New_York&RIB.BusinessHours=6434&propertiesToTranslate=RIB.BusinessHours
```

Troubleshooting

Errors are returned if valid syntax invokes a definition that is not itself valid. For example, you attempt to create a site group. The syntax includes the businessHourId for an invalid business-hour definition. In such a case, the HTTP Response XML includes an error message similar to the following text:

```
<Group bHourID="99990" desc="This is a site group" inherit="true" location="North America" name="East Coast
 USA" result="Error with validating business hour ID: Business hour definition with an ID of '99990' not
 found!" timeZone="EST" type="site group"/>
```

Business hours definitions comprise a starting hour and an ending hour. The `startHour` and `endHour` parameters in the XML must be the same for all of the days that are included in the definition. Otherwise, you see an error in POST or PUT operations.

Use Web Services to Manage Maintenance Indicators

An administrator can use web services to create maintenance indicators. Maintenance indicators represent times when maintenance is occurring. After you schedule maintenance indicators, views indicate maintenance with shading.

Maintenance indicators apply to all the devices and components in a site group. When the associated site group is selected in the context, maintenance indicators appear in applicable views as you navigate between dashboards. The subtitle of each view indicates whether maintenance indicators apply to the view.

Subgroups do not directly inherit maintenance indicators from the site groups. Associate the maintenance indicators with each relevant subgroup. However, when rendering views, these filters apply to all items based on the selected site group. The filters of the selected site group apply to all items in that group and in any subgroups. When you change the selected site group to a subgroup, the filters of the parent group are no longer applicable.

Reference groups inherit associated maintenance indicators from the original site group.

Maintenance Indicators Web Service Operations

Issue the following call to see the available operations for the maintenance indicators web service:

```
http://PC_host:8181/pc/center/rest/maintenanceindicators/documentation
```

Operations

- ***PC_host*:8181**
  Specify the NetOps Portal IP address. 8181 is the required port.
- **get list**
  Get a list of all maintenance indicators definitions for the authenticated user.
  URL: `http://PC_host:8181/pc/center/webservice/maintenanceindicators/`
  HTTP method: `GET`
  XSD for the returned XML: `http://PC_host:8181/pc/center/rest/maintenanceindicators/xsd`
- **add**
  Create a maintenance indicators definition from the provided XML.
  URL: `http://PC_host:8181/pc/center/webservice/maintenanceindicators/tenantId/{tenantId}`
  – ***{tenantId}***
  Specify the ID for the desired tenant.
  HTTP method: `POST`
  XSD for the provided XML: `http://PC_host:8181/pc/center/rest/maintenanceindicators/xsd`
  XSD for the returned XML: `http://PC_host:8181/pc/center/rest/maintenanceindicators/xsd`
- **delete**
  Delete the maintenance indicators definition using the specified ID.
  URL: `http://PC_host:8181/pc/center/webservice/maintenanceindicators/id/{maintenanceIndicatorId}`
  – ***{maintenanceIndicatorId}***

Specify the ID for the desired maintenance indicator.

HTTP method: `DELETE`
Return type is a string.

- **get**
  Get a specific maintenance indicators definition.
  URL: `http://PC_host:8181/pc/center/webservice/maintenanceindicators/id/{maintenanceIndicatorId}`
  HTTP method: `GET`
  XSD for the returned XML: *http://PC_host:8181/pc/center/rest/maintenanceindicators/xsd*
- **get id names**
  Return a list of identifiers that can be used in other methods to identify certain objects. For this web service, an empty list is returned.
  URL: *http://PC_host:8181/pc/center/webservice/maintenanceindicators/idNames*
  HTTP method: `GET`
- **get list for tenant**
  Get a list of all maintenance indicators definitions for the specified tenant.
  URL: `http://PC_host:8181/pc/center/webservice/maintenanceindicators/tenantId/{tenantId}`
  – ***{tenantId}***
  Specify the ID for the desired tenant.
  HTTP method: `GET`
  XSD for the returned XML: `http://PC_host:8181/pc/center/rest/maintenanceindicators/xsd`
- **update**
  Update the specified maintenance indicators definition from the provided XML.
  URL: `http://PC_host:8181/pc/center/webservice/maintenanceindicators/tenantId/{tenantId}/id/{maintenanceIndicatorId}`
  HTTP method: `PUT`
  XSD for the provided XML: `http://PC_host:8181/pc/center/rest/maintenanceindicators/xsd`
  XSD for the returned XML: `http://PC_host:8181/pc/center/rest/maintenanceindicators/xsd`

Create Maintenance Indicators Using Web Services

Use any REST client to create and configure maintenance indicators using the maintenance indicators web service.

Follow these steps:

1. Set up a REST client with a connection to the NetOps Portal server.
2. Use the following format for the URL in the REST client:
   http://PC_host :8181/pc/center/webservice/maintenanceindicators/tenantId/8

   NOTE

   8 is the ID for the Default Tenant.
3. Select
   `POST`
   for the **'HTTP' Method**.
4. Provide a valid Username and Password for a user account that has global administrator access to NetOps Portal.
5. Select **'application/xml'** as the **'Body Content-type'**.
6. Add the following XML within the "Body" text section:

```
<MaintenanceIndicator>
   <Name>Maintenance Indicator Name</Name>
   <Description> Maintenance Indicator Description</Description>
   <MaintenanceYear>Year</MaintenanceYear>
```

```
    <MaintenanceMonth>Month</MaintenanceMonth>
    <MaintenanceDay>Day</MaintenanceDay>
    <StartHour>StartHour</StartHour>
    <EndHour>EndHour</EndHour>
    <SelectedSites>
       <SelectedSite>SiteID</SelectedSite>
       <SelectedSite>SiteID</SelectedSite>
    </SelectedSites>
 </MaintenanceIndicator>
```

- **Name**
  Specify a name for this maintenance indicators definition.
- **Description**
  Specify a description for this maintenance indicators definition.
- **MaintenanceYea**r
  Specify a four-digit year for when the related maintenance takes place.
- **MaintenanceMonth**
  Specify a month (1 - 12) for when the related maintenance takes place.
- **MaintenanceDay**
  Specify a day (1 - 31) for when the related maintenance takes place.
- **StartHour**
  Specify an hour (0 - 23) for when the related maintenance takes place.
- **EndHour**
  Specify an hour (1 - 24) for when the related maintenance takes place.
- **SelectedSites**
  (Optional) Specify one or more tags containing the site IDs to assign this maintenance indicators definition to. To create a definition without sites that are assigned, omit this parameter

7. Replace any values with the values that you want to use for the new maintenance indicators.
   For example, supply the following parameters, where 3569 is the site ID for the **Framingham** site:

```
 <MaintenanceIndicator>
    <Name>Framingham router maintenance</Name>
    <Description>Upgrade of the network for framingham</Description>
    <MaintenanceYear>2016</MaintenanceYear>
    <MaintenanceMonth>8</MaintenanceMonth>
    <MaintenanceDay>5</MaintenanceDay>
    <StartHour>19</StartHour>
    <EndHour>22</EndHour>

    <SelectedSites>
       <SelectedSite>510</SelectedSite>
       <SelectedSite>511</SelectedSite>
    </SelectedSites>

 </MaintenanceIndicator>
```

8. Run the method.
9. Repeat the preceding steps until you have created as many maintenance indicators as you require.

Use Web Services to Manage Alarm Attributes

An administrator can use web services to create alarm attributes. You can create them for a specific tenant or for all tenants (global).

> **NOTE**
>
> You can create the same attribute at the tenant level and the global level. The tenant level attribute applies to the tenant user and the global level attribute applies only to all other tenant users.
>
> **WARNING**
>
> Do not use these web services to manage external attributes, which directly correspond to specific MIB variables in DX NetOps Spectrum.

For more information, see Alarms View.

Alarm Attributes Web Service Operations

Issue the following call to see the available operations for the alarm attributes web service:

```
http://PC_host:8181/pc/center/rest/alarmattributes/documentation
```

Operations

- ***PC_host*:8181**
- Specifies the NetOps Portal host name. 8181 is the required port.
- **delete**
  Delete the attribute using the specified ID.
  URL: `http://PC_host:8181/pc/center/webservice/alarmattributes/tenantId/{tenantId}/attributeId/{attributeId}`
  – ***{tenantId}***
    The id for the desired tenant
  – ***{attributeId}***
    The hexadecimal id of the alarm attribute
  HTTP method: `DELETE`
  Return type is a string.
- **delete global**
  Delete the global attribute using the specified ID.
  URL: `http://PC_host:8181/pc/center/webservice/alarmattributes/global/attributeId/{attributeId}`
  – ***{attributeId}***
    The hexadecimal id of the alarm attribute
  HTTP method: `DELETE`
  Return type is a string.
- **get id names**
  Return an empty list. In other web services, it can potentially return a list of identifiers that can be used in other methods to identify certain objects.
  URL: `http://PC_host:8181/pc/center/webservice/alarmattributes/idNames`

HTTP method: `GET`

- **get list for tenant**
  Get a list of all attribute definitions for the specified tenant.
  URL: `http://PC_host:8181/pc/center/webservice/alarmattributes/tenantId/{tenantId}`
  – ***{tenantId}***
  The id for the desired tenant
  HTTP method: `GET`
  XSD for the returned XML: `http://PC_host:8181/pc/center/rest/alarmattributes/xsd`
- **add or update**
  Add or update an attribute for the specified tenant.
  URL: `http://PC_host:8181/pc/center/webservice/alarmattributes/tenantId/{tenantId}`
  – ***{tenantId}***
  The id for the desired tenant
  HTTP method = `POST`
  XSD for the provided XML: `http://PC_host:8181/pc/center/rest/alarmattributes/xsd`
  XSD for the returned XML: `http://PC_host:8181/pc/center/rest/alarmattributes/xsd`
- **get global list**
  Get a list of all global attribute definitions.
  URL: `http://PC_host:8181/pc/center/webservice/alarmattributes/global`
  HTTP method: `GET`
  XSD for the returned XML: `http://PC_host:8181/pc/center/rest/alarmattributes/xsd`
- **add or update global**
  Add or update a global attribute.
  URL: `http://PC_host:8181/pc/center/webservice/alarmattributes/global`
  HTTP method = `POST`
  XSD for the provided XML: `http://PC_host:8181/pc/center/rest/alarmattributes/xsd`
  XSD for the returned XML: `http://PC_host:8181/pc/center/rest/alarmattributes/xsd`

Create Alarm Attributes Using Web Services

Use any REST client to create and configure alarm attributes using the alarm attributes web service.

Follow these steps:

1. Set up a REST client with a connection to the NetOps Portal server.
2. Use the following format for the URL in the REST client:
   `http://PC_host :8181/pc/center/webservice/alarmattributes/tenantId/8`

   NOTE

   8 is the ID for the Default Tenant.
3. Select
   `POST`
   for the **'HTTP' Method**.
4. Provide a valid Username and Password for a user account that has global administrator access to NetOps Portal.
5. Select **'application/xml'** as the **'Body Content-type'**.
6. Add the following XML within the "Body" text section:

```
<AlarmAttribute>
    <AttributeID>HEX_ID</AttributeID>
    <Name>Attribute_Name</Name>
    <Description>Attribute_Description</Description>
    <Type>Value_Type</Type>
```

```
    <AddAsFilter>true</AddAsFilter>
   <AddAsColumn>true</AddAsColumn>
</AlarmAttribute>
```

- **AttributeID**
  Specify the hexadecimal ID of the alarm attribute.
- **Name**
  Specify a name for this alarm attribute.
- **Description**
  Specify a description for this alarm attribute.
- **Type**
  Specify one of the following allowable types:
  - **STRING**
    A String field, for example, the Device Name
  - **BOOLEAN**
    A Boolean value, for example, Acknowledged
  - **ADDRESS_RANGE**
    An IP address field, for example, the IP Address
  - **INTEGER**
    A numeric value, for example, Number of Occurrences
  - **HEX**
    A hexadecimal value, for example, Cause Code
  - **OCTET_STRING**
    A string value that is treated as a numeric value
- **AddAsFilter**
  Specify whether to add the attribute to the filter list.
  Default: true
- **AddAsColumn**
  Specify whether to add the attribute as a column in an Alarm view. The column is hidden by default.
  Default: true

7. Replace any values with the values that you want to use for the new alarm attribute.
   Example:

```
<AlarmAttribute>
    <AttributeID>0x129e7</AttributeID>
    <Name>Topology Model Name</Name>
    <Description>The Topology Model Name String</Description>
    <Type>STRING</Type>
    <AddAsFilter>true</AddAsFilter>
   <AddAsColumn>true</AddAsColumn>
</AlarmAttribute>
```

8. Run the method.
9. Repeat the preceding steps until you have created as many alarm attributes as you require.

Data Aggregator REST WebServices

Data Aggregator REST web services manage administrative operations, such as retrieving data or managing relationships between profiles and tenants or groups. Use any REST client tool or use an HTTP tool that can send requests and can receive responses. Some REST web services cannot be scoped to tenant domains.

These REST services include two types:

- **Data-Driven**
  Read and modify Data Aggregator configuration information, such as Monitoring Profiles and Groups.
- **Generic**
  Manage metric families and limited SNMP vendor certification support. Generic REST web services are self-filtering and do not use an argument within the URL to manage relationships.

Basic Data Aggregator REST Operations

Each endpoints maps to types of items, such as groups, monitoring profiles, tenants, and device certifications. Use specific endpoints to return a list of results, or create, update, or delete an item.

> **NOTE**
>
> Individual metric family items and SNMP vendor certification items are specified using name instead of ID.
>
> **WARNING**
> Set the Context-type files to **application/xml** when you perform operations using Data Aggregator REST web services.

Basic operations:

- **GET http://.../*endpoint***
  Returns a list of all items of the specified type. The getlist.xsd schema defines the format for the return data.
- **GET http://.../*endpoint*/[*id* | *name*]**
  Returns the details for a single item with the specified ID or certification name. The XSD schema defines the format for the return data.
- **POST http://.../*endpoint***
  Creates an object of the specified type with specified facets. The XSD schema defines the format for the return data.

  > **NOTE**
  > With POST operations, you can create objects with the same name but different IDs. This convention is allowed because it is often valid when the objects are scoped to different tenants.
- **PUT http://.../*endpoint*/[*id* | *name*]**
  Updates the attributes of the specified item. The update.xsd schema defines the format and expected fields.

  > **NOTE**
  > With PUT operations, you can create objects with the same name but different IDs. This convention is allowed because it is often valid when the objects are scoped to different tenants.
- **DELETE http://.../*endpoint*/[*id* | *name*]**
  Deletes the item that is specified using the ID or certification name.

Generic REST Web Services

Use generic REST web services to manage metric families and for limited SNMP vendor certification support. Generic REST web services are self-filtering and do not use an argument within the URL to manage relationships.

The following URL displays details about every user-facing, generic web service in the system:
`http://DA_host:8581/genericWS`

This URL includes detailed information about generic web services, including XSDs, URIs, supported HTTP methods, attributes, and relationships.

To view detailed documentation about a specific endpoint, access the following URL: `http://DA_host:8581/genericWS/endpoint/documentation`

Manage Relationships with Generic REST Web Services

Generic REST web services do not use an argument within the URL to manage relationships. Instead, generic REST web services rely on the basic operations alone to manage relationships. The endpoints filter on themselves to expose the information. These methods are used for managing relationships between the metric families and SNMP vendor certifications.

Relationships for generic REST web services are viewed, created, and deleted using the following methods:

- GET
- PUT
- DELETE

All methods use the following URL:
`http://DA_host:8581/genericWS/endpoint/name/endpoint`

Example: List Metric Families Related to an SNMP Vendor Certification

To return a list of metric families for a specified SNMP vendor certification, use the GET operation at the following URL:
`http://DA_host:8581/genericWS/certifications/snmp/name/metricfamilies`

Data-Driven REST Web Services

As an administrator, use data-driven REST web services for most Data Aggregator web services, such as monitoring profiles and groups.

The following URL display details about every user-facing, data-driven web service in the system:
`http://DA_host:8581/rest`

This URL includes detailed information about endpoints, and data-driven web services, including XSDs, URIs, supported HTTP methods, attributes, and relationships.

To view detailed documentation about a specific endpoint, access the following URL:
`http://DA_host:8581/rest/endpoint/documentation`

View XSD Schemas for Data-Driven REST Web Services

Do the following steps before performing an HTTP request:

- Verify the XML schema definition (XSD) for the endpoint.
- Review the format of the return or upload XML that the service provides.

Each item of content that is placed in an XML document must adhere to the description of the endpoint.

The XSD files for each operation contain tags that describe the attributes and the purpose of the metric families. To obtain the XSD for an endpoint, use the following paths with the data-driven web services URL:
`http://DA_host:8581/rest/endpoint/XSD/operation.xsd`

- ***operation***
  Specifies the type of operation to execute.
  Values:
  – **get**

The XSD for a single item get
- **getlist**
  The XSD for a list of the endpoint items
- **filterselect**
  The XSD for advanced filter criteria and return XML format to be specified using GET Tunneling
- **create**
  The XSD that any input XML must match when trying to create
- **update**
  The XSD that any input XML must match when trying to update

> **NOTE**
> Not all operations are supported for each endpoint. If an operation is not supported, the web service fails and returns a '403 Forbidden' message.

Filter on Attributes in the XSD Schema for Data-Driven REST Web Services

You can filter on attributes such as the item name, description, and other such attributes. For example, filter monitoring profiles by the metric families they contain. You can use this information to determine whether to add or remove metric families to or from a monitoring profile.

Follow these steps:

1. Enter the following URL in a web browser:
   `http://DA_host:8581/rest/`
   A list shows the available data-driven web services.
2. Click a web service.
   The documentation page for that web service opens.
3. Click the URL under the "filtered get list" method.
   The XSD schema opens.
4. Look for the elements that include the following property:
   `substitutionGroup="AttributeFilterTypeSubstitution`
   Use this information to determine which attribute you want to filter on.
5. Open a REST client editor or HTTP tool that sends requests and gets responses, and set the Content-type to **application/xml**.
6. Enter the following filter criteria:
   - URL: `http://DA_host:port/rest/endpoint/filtered/`
   - HTTP method = POST
     This method must define the filter criteria.
   - Basic filter select criteria on the Body tab in the HTTP Request pane:

```
<FilterSelect xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="filter.xsd">

       <Filter>

                  <elementName type="CONTAINS">filter-criteria</elementName>

       </Filter>

</FilterSelect>
```

- ***filter-criteria***

Specifies the actual value of the attribute.

- ***elementName***
  Specifies the element name (attribute) to filter on.

  NOTE
  You can specify selection criteria also, such as poll rates only. This method is also known as Get Tunneling. For more information, see the following example.

Results are returned in the Body tab of the HTTP Response pane.

Example: Return a List of Monitoring Profiles that Contain a Metric Family Using Filter and Selection Criteria (Get Tunneling).

To return the monitoring profiles that contain a metric family using poll rate as the selection criteria, use the following method:

- **Method:** POST
- **URL**: `http://DA_host:8581/rest/monitoringprofiles/filtered/`
- **Body:**

```
<FilterSelect xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="filter.xsd">

               <Filter>

                     <MonitoringProfile.FacetTypes type="CONTAINS">{http://
im.ca.com/normalizer}NormalizedPortInfo</MonitoringProfile.FacetTypes>

               </Filter>

               <Select use="exclude"  isa="exclude">

                  <MonitoringProfile use="exclude">

                      <PollRate use="include"/>

                  </MonitoringProfile>

               </Select>

</FilterSelect>
```

Example: Return a List of Monitoring Profiles that Contain an Item **Using Filter and Selection Criteria**.

To return the monitoring profiles that contain an item using the name as the selection criteria, use the following method:

- **Method:** POST
- **URL**: `http://DA_host:8581/rest/monitoringprofiles/filtered/`
- **Body:**

```
<FilterSelect xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="filter.xsd">
```

```
                <Filter>

                      <MonitoringProfile.FacetTypes type="CONTAINS">{http://
im.ca.com/normalizer}NormalizedPortInfo</MonitoringProfile.FacetTypes>

                </Filter>

                <Select use="exclude"  isa="exclude">

                   <Item use="exclude">

                      <Name use="include"/>

                   </Item>

                </Select>

</FilterSelect>
```

Manage Basic Relationships with Data-Driven REST Web Services

To create or delete relationships, use the PUT or DELETE operation at the following URL:
`http://DA_host:8581/rest/endpoint/id/relatesto/endpoint/id`

To view nested relationships, use the following URL:
`http://DA_host:8581/rest/endpoint/id/relatesto/endpoint/endpoint`

Example

To see all associated groups and devices for a specific monitoring profile ID 781, use a GET call at the following URL:
`http://DA_host:8581/rest/monitoringprofile/781/relatesto/groups/devices`

The response URL lists all the related groups and devices for the specified monitoring profile ID with the attributes in the current getlist.xsd file.

Limit Data-Driven Scope to Tenant Domains

You can limit the data-driven scope of your operations to a specific tenant domain for some of the Data Aggregator features, rather than accessing information in the entire global repository.

Use the following basic method to access information for an endpoint with a specific tenant domain:

```
GET http://DA_host:port/rest/tenant/id/endpoint
```

You can use some, but not all, web services with tenant domains.

Change When Same Day, Same Hour Baseline Averages Are Calculated

After a limited amount of data has been collected, the baseline average is calculated for the same hour for every preceding day of the week.

When more data is available, a switchover in the calculation method occurs automatically and Data Aggregator establishes "normal" by averaging hourly samples across available preceding same days of the week.

By default, this automatic switchover occurs when at least 3 same day of the week, same hour data samples are available for the past 12 weeks. You can change when this automatic switchover occurs.

Consider the following information about changing when same day of the week, same hour baseline averages are calculated:

- You have the option of changing both attributes, or just one of the attributes.
- Both attribute values must be a numeric value that is greater than or equal to 1.
- No upper limit is enforced. However, the retention policy of the hourly roll ups defines the upper limit. By default, the hourly retention rate is 90 days (which is approximately 12 weeks of data). If you increase the maximum number of preceding weeks, increase the hourly roll-up retention rate also.
- The MinimumNumberOfRequiredDataPoints attribute value must be less than or equal to the MaximumNumberOfWeeks value.

Follow these steps:

1. Enter the following information in a web browser:
   `http://DA_host:port/rest/sdshbaselineconfig`
   – ***DA_host*:*port***
   Specifies the Data Aggregator host name and the port number.
   Default port: 8581
   The sdshbaselineconfig webservice endpoint URL opens.
2. Review the current values for the minimum required number of data points and the maximum number of preceding weeks.
3. Open a REST client editor or HTTP tool that sends requests and gets responses and set the Content-type to application/xml.
4. Enter the following criteria:
   – HTTP method = PUT
   – Enter the minimum required number of data points within the maximum number of preceding weeks (to trigger the switchover in the baseline calculation method) that you want to change on the Body tab in the HTTP Request pane. For example:

```
<SdshBaselineConfiguration version="1.0.0">

 <SDSHSettings>

  <MinimumNumberOfRequiredDataPoints>5</MinimumNumberOfRequiredDataPoints>

  <MaximumNumberOfWeeks>10</MaximumNumberOfWeeks>

  </SDSHSettings>

</SdshBaselineConfiguration>
```

   In this example, the minimum number of data points to be available for baseline average calculation has been changed to 5. The number of preceding weeks to look for these data points has been changed to 10.

Manage Polling Behavior for Components

You can use this Data Aggregator REST web service to disable or enable polling on other components. This feature allows for more granular polling control than monitoring profile filters alone.

You can also disable or enable polling on specific interfaces from the UI. For more information, see Configure Monitoring Profiles.

By default, polling is enabled for all new components. You can use a Data Aggregator REST web service to disable polling for all new components associated with specific metric families. For more information, see Manage Default Polling Behavior.

Follow these steps:

1. Set up a REST client with a connection to the Data Aggregator server.
2. Query the following REST URL to find the `pollable` item ID:
   `http://DA_host:8581/rest/pollable`
3. Use the following format for the URL in the REST client:
   `http://DA_host:8581/rest/pollable/itemID`
   - ***itemID***
     Specify the `pollable` item ID.
4. Select
   `PUT`
   for the **'HTTP' Method**.
5. Provide a valid Username and Password for a user account that has global administrator access to the Data Aggregator.
6. Select **'application/xml'** as the **'Body Content-type'**.
7. Add the following XML within the "Body" text section:

```
<Pollable version="1.1.1">

   <IsPollEnabled>false</IsPollEnabled>

</Pollable>
```

   - **<IsPollEnabled>**
     To disable polling, specify false. To enable polling, specify true.
8. Run the method.

Manage Default Polling Behavior

By default, polling is enabled for all new components. You can use this Data Aggregator REST web service to disable polling for all new components of specific metric families.

You can disable or enable polling on specific interfaces through the UI. For more information, see Manage Interface Polling Behavior.

You can also use a Data Aggregator REST web service to disable or enable polling on other components. For more information, see Manage Polling Behavior for Components.

This feature allows for more granular polling control than monitoring profile filters alone. For more information, see Configure Monitoring Profiles.

Disable Polling for New Components

You can disable polling for all new components of specific metric families.

Follow these steps:

1. Set up a REST client with a connection to the Data Aggregator server.
2. Query the following REST URL to find the `DiscoveryDefaultConfig` item ID:
   `http://DA_host:8581/rest/discoverydefaultconfig`
3. Use the following format for the URL in the REST client:
   `http://DA_host:8581/rest/discoverydefaultconfig/itemID`
   – ***itemID***
   Specify the `DiscoveryDefaultConfig` item ID.
4. Select
   `PUT`
   for the **'HTTP' Method**.
5. Provide a valid Username and Password for a user account that has global administrator access to the Data Aggregator.
6. Select **'application/xml'** as the **'Body Content-type'**.
7. Add the following XML within the "Body" text section:

```
<DiscoveryDefaultConfig version="1.0.0">
  <DisablePollingOfNewComponentsList>
     <DisablePollingOfNewComponents>{http://im.ca.com/
normalizer}Metric_Family_Name</DisablePollingOfNewComponents>
  </DisablePollingOfNewComponentsList>
</DiscoveryDefaultConfig>
```

   – ***Metric_Family_Name***
   Specify the metric family that is associated with the components for which you want to disable polling.
8. Edit the XML to include the metric families that are associated with the components for which you want to disable polling.
   Example:

```
<DiscoveryDefaultConfig  version="1.0.0">
   <DisablePollingOfNewComponentsList>
     <DisablePollingOfNewComponents>{http://im.ca.com/
normalizer}NormalizedPortInfo</DisablePollingOfNewComponents>
     <DisablePollingOfNewComponents>{http://im.ca.com/normalizer}NormalizedCPUInfo</
DisablePollingOfNewComponents>
   </DisablePollingOfNewComponentsList>
</DiscoveryDefaultConfig>
```

9. Run the method.

Re-enable Polling for New Components

If you want to re-enable polling for new components of a specific metric family, simply reput the XML excluding the metric family.

Follow these steps:

1. Set up a REST client with a connection to the Data Aggregator server.
2. Query the following REST URL to find the `DiscoveryDefaultConfig` item ID:

http://DA_host:8581/rest/discoverydefaultconfig

3. Use the following format for the URL in the REST client:
   http://DA_host:8581/rest/discoverydefaultconfig/itemID
   – ***itemID***
     Specify the `DiscoveryDefaultConfig` item ID.
4. Select
   `PUT`
   for the **'HTTP' Method**.
5. Provide a valid Username and Password for a user account that has global administrator access to the Data Aggregator.
6. Select **'application/xml'** as the **'Body Content-type'**.
7. Add the following XML within the "Body" text section:

```
<DiscoveryDefaultConfig version="1.0.0">
  <DisablePollingOfNewComponentsList>
     <DisablePollingOfNewComponents>{http://im.ca.com/
normalizer}Metric_Family_Name</DisablePollingOfNewComponents>
  </DisablePollingOfNewComponentsList>
</DiscoveryDefaultConfig>
```

   – ***Metric_Family_Name***
     Specify the metric family that is associated with the components for which you want to disable polling.
8. Edit the XML to exclude the metric families that are associated with the components for which you to re-enable polling.
   Example: You want to re-enable polling for new components of the `NormalizedCPUInfo` metric family.

```
<DiscoveryDefaultConfig  version="1.0.0">
   <DisablePollingOfNewComponentsList>
     <DisablePollingOfNewComponents>{http://im.ca.com/
normalizer}NormalizedPortInfo</DisablePollingOfNewComponents>
   </DisablePollingOfNewComponentsList>
</DiscoveryDefaultConfig>
```

9. Run the method.

Manage Filtered Components

You can control whether filtered components are created with the `DoNotCreateFilteredItems` attribute.

Follow these steps:

1. Set up a REST client with a connection to the Data Aggregator server.
2. Query the following REST URL to find the `DiscoveryDefaultConfig` item ID:
   http://DA_host:8581/rest/discoverydefaultconfig
3. Use the following format for the URL in the REST client:
   http://DA_host:8581/rest/discoverydefaultconfig/itemID
   – ***itemID***
     Specify the `DiscoveryDefaultConfig` item ID.
4. Select
   `PUT`
   for the **'HTTP' Method**.
5. Provide a valid Username and Password for a user account that has global administrator access to the Data Aggregator.

6. Select **'application/xml'** as the **'Body Content-type'**.
7. Add the following XML within the "Body" text section:

```
<DiscoveryDefaultConfig version="1.0.0">
  <DoNotCreateFilteredItems>true</DoNotCreateFilteredItems>
</DiscoveryDefaultConfig>
```

   – ***DoNotCreateFilteredItems***
   Specify whether true or false. True does not create filtered items. False creates filtered items.
8. Run the method.

Poll Sensitive and Critical Devices Without a Performance Impact

Critical devices are sensitive to too many polls, which can lead to performance problems. To throttle the SNMP poll requests and avoid overwhelming your sensitive devices, configure the SNMP polling controls.

By default, SNMP polling is controlled in three ways:

- **SNMP traffic threshold**
  No more than 15 SNMP requests can be sent to a device at a time. Poll and discovery SNMP requests over 15 are queued and sent to a device when possible during the polling cycle. Up to 600 requests can be queued.
- **SNMP timeouts threshold**
  When 15 or more SNMP requests timeout, polling is suspended for the remainder of the current polling cycle. An event is generated, informing you of the situation.
  Polling resumes at the beginning of each poll cycle. When the timeouts do not exceed the 15 threshold, a "clear" event is generated.
- **SNMP PDU Segmentation**
  SNMP packets can be limited to contain a maximum number of varbinds by splitting a larger SNMP request into multiple smaller requests and reassembling the responses.

These thresholds are designed to prevent overwhelming a device with too many poll requests or PDUs that are too large. You can override these SNMP polling thresholds defaults by modifying the following parameters:

- **TimeoutFailSafeThrottleDefault**Specifies the default maximum number of timeouts that trigger the fail-safe throttle**.**
- **MaxOutstandingRequestsDefault**Specifies the default maximum number of outstanding requests.
- **MaxRequestSizeDefault**Specifies the default maximum number of varbind that a single SNMP request may contain.

These thresholds can also be modified per device so that a specific device can be assigned individual threshold values.

For example, your older router is exceptionally sensitive to polling. But, this router is critical and must be polled as frequently as possible. You already adjusted your monitoring profile to remove unnecessary metric families from polling. You also applied a filter in your monitoring profile to reduce the number of polled interfaces. However, polling still causes this router to crash. Therefore, your only option is to adjust the default SNMP polling parameters for your sensitive router.

You can add any of the following parameters to the policy for individual IPs or IP ranges in an IPRange section within the IPRangeList:

- **TimeoutFailSafeThrottle**The maximum number of timeouts that are applied on the devices within this IP range
- **MaxOutstandingRequests**
  The maximum number of outstanding requests that are sent to the devices within the indicated IP range
- **MaxRequestSize**
  Limits the number of varbinds in an outgoing SNMP request
  If the number of varbinds in the SNMP request exceeds the value of `MaxRequestSize` , the outgoing request is split into two or more smaller requests.
  Some IP ranges are not covered in the IPRange sections. For global settings, use the `MaxRequestSizeDefault` parameter to set the varbind limit.

NOTE

If `MaxRequestSize` is 0, the original request is sent regardless of its size.

For related troubleshooting information, see Gaps in Data Appear during Throttling and Polling Stopped Event Message.

Follow these steps:

1. Find the ID for your IP Domain (that contains your sensitive router) by opening:
   `http://DA_host:port/rest/ipdomains`

   – ***DA_host*:*port***
   Specifies the Data Aggregator hostname and the port number where you are accessing the REST web services from.
2. Locate your IP Domain ID in the following SNMP throttle policy list, and note the corresponding policy ID:
   `http://DA_host:port/rest/snmpthrottlepolicies`

3. Determine the number of varbinds that you want to include in a single outgoing SNMP request. Some devices ignore requests that are too large without sending an error. As a result, the SNMP poller cannot reach the device. Use the `MaxRequestSize` value to allow the Data Collector to monitor these devices.
   Example
   If the interface SNMP request has 27 varbinds and *MaxRequestSizeDefault* is set to 15, the outgoing request is split into two smaller requests. One request contains 14 varbinds, and the other contains 13 varbinds.
   Example: The following example from an SNMP throttle policy shows that the policy ID is "601" for IP Domain "2" with no limit on the number of varbinds:

```
<SnmpThrottlePolicy version="1.0.0">
    <ID>601</ID>
    <MaxOutstandingRequestsDefault>15</MaxOutstandingRequestsDefault>
    <QueueLength>600</QueueLength>
    <TimeoutFailSafeThrottleDefault>15</TimeoutFailSafeThrottleDefault>
    <MaxRequestSizeDefault>0</MaxRequestSizeDefault>
     <IPDomainID>2</IPDomainID>
    </SnmpThrottlePolicy>
```

4. Open a REST client editor or HTTP tool that sends requests and gets responses, and set the Content-type to application/xml.
5. Open and edit the SNMP throttle policy for your IP Domain by entering the following criteria:
   – URL: `http://DA_host:port/rest/snmpthrottlepolicies/policyID`
   • ***policyID***
   Specifies a unique identification number that is assigned to the SNMP throttle policy for the IP Domain that contains your sensitive device.
   Example: `http://DA_host:port/rest/snmpthrottlepolicies/601`
   – HTTP method = PUT
   – Adjust the following values for your IP Range on the Body tab in the HTTP Request pane:
   • **<MaxOutstandingRequests>**
   SNMP traffic threshold
   • **<TimeoutFailSafeThrottle>**
   SNMP timeouts threshold

NOTE
Both values are required for every IP Range entry. You can disable either parameter by setting the value to "0."

– Remove the following lines:
  - `<ID>`
  - `<IPDomainID>`

Results are returned in the Body tab of the HTTP Response pane.
Example: In this example, the thresholds are lowered to "10" for device 10.231.41.7 only. For this device, the number of varbinds is limited to 50. The default thresholds and other IP Range thresholds continue using the default value of "15." For devices 10.231.41.1-10.231.41.255, SNMP requests are limited to 30 varbinds.

```
<SnmpThrottlePolicy version="1.0.0">
    <IPRangeList>
      <IPRange>
            <IPRangeText>10.231.41.7</IPRangeText>
            <MaxOutstandingRequests>10</MaxOutstandingRequests>
            <TimeoutFailSafeThrottle>10</TimeoutFailSafeThrottle>
            <MaxRequestSize>50</MaxRequestSize>
      </IPRange>
      <IPRange>
            <IPRangeText>10.231.41.1-10.231.41.255</IPRangeText>
            <MaxOutstandingRequests>15</MaxOutstandingRequests>
            <TimeoutFailSafeThrottle>15</TimeoutFailSafeThrottle>
            <MaxRequestSize>30</MaxRequestSize>
      </IPRange>
    </IPRangeList>
        <MaxRequestSizeDefault>0</MaxRequestSizeDefault>
        <MaxOutstandingRequestsDefault>15</MaxOutstandingRequestsDefault>
        <QueueLength>600</QueueLength>
        <TimeoutFailSafeThrottleDefault>15</TimeoutFailSafeThrottleDefault>
</SnmpThrottlePolicy>
```

NOTE
You can adjust the thresholds for a single device or a range of devices. The IP Range definition and the IP Range order determine which threshold applies. The IP Ranges are listed in priority order. That is, the first IP Range that applies to a device determines the threshold value to apply.

6. Always include the `MaxOutstandingRequestsDefault`, `MaxRequestSizeDefault`, `TimeoutFailSafeThrottleDefault`, and `QueueLength` parameters in the update/POST XML at the root level. Include the parameters even if the values do not differ from the default.
   Example:
   This PUT command generates the policy that follows.

```
Update XML: PUT on URL DA-HOST:8581/rest/snmpthrottlepolicies/21
   <SnmpThrottlePolicy version="1.0.0">
   <IPRangeList>
     <IPRange>
       <IPRangeText>130.119.103.8</IPRangeText>
       <MaxOutstandingRequests>10</MaxOutstandingRequests>
       <TimeoutFailSafeThrottle>10</TimeoutFailSafeThrottle>
```

```
        <MaxRequestSize>20</MaxRequestSize>
      </IPRange>
    </IPRangeList>
    <MaxRequestSizeDefault>0</MaxRequestSizeDefault>
    <MaxOutstandingRequestsDefault>15</MaxOutstandingRequestsDefault>
    <TimeoutFailSafeThrottleDefault>15</TimeoutFailSafeThrottleDefault>
    <QueueLength>600</QueueLength>
    </SnmpThrottlePolicy>
```

This command generates the following policy:

```
<SnmpThrottlePolicy version="1.0.0">
<ID>21</ID>
<QueueLength>600</QueueLength>
<TimeoutFailSafeThrottleDefault>15</TimeoutFailSafeThrottleDefault>
<IPDomainID>2</IPDomainID>
<IPRangeList>
<IPRange>
<IPRangeText>130.119.103.8</IPRangeText>
<MaxOutstandingRequests>10</MaxOutstandingRequests>
<TimeoutFailSafeThrottle>10</TimeoutFailSafeThrottle>
<MaxRequestSize>20</MaxRequestSize>
</IPRange>
</IPRangeList>
<MaxRequestSize>0</MaxRequestSize>
<MaxOutstandingRequestsDefault>15</MaxOutstandingRequestsDefault>
</SnmpThrottlePolicy>
```

Schedule Data Purges

You schedule how often Data Repository purges all data that is older than the specified retention periods. You can modify the start hour, start minute, and you can modify the start second. By default, the Data Aggregator purges data every day at 2:00:00 AM.

Follow these steps:

1. Enter the following information in a web browser:

   `http://DA_host:port/rest/globalretentionscheduledefinition`

   – ***DA_host*:*port***
   Specifies the Data Aggregator hostname and the port number where you are accessing the REST web services from.

   The `globalretentionscheduledefinition` webservice endpoint URL.
2. Take note of the ID that is assigned to the `globalretentionscheduledefinition` .
3. Look for the elements that have `StartMinute, StartHour,` and `StartSecond` . Use this information to determine whether you want to modify the start hour, start minute, or start second when old data is purged.

4. Open a REST client editor or HTTP tool that sends requests and gets responses and set the Content-type to application/xml.
5. Enter the following criteria:
   – URL: `http://DA_host:port/rest/globalretentionscheduledefinition/ID`
     - ***ID***
       Is a unique identification number that is assigned to the `globalretentionscheduledefinition`.
   – HTTP method = PUT
   – Enter the time values that you want to change in the Body tab of the HTTP Request pane.
     For example:

```
<GlobalRetentionScheduleDefinition version="1.0.0">

  <StartMinute>28</StartMinute>

  <StartHour>17</StartHour>

  <Enabled>true</Enabled>

  <Status>Scheduled to run everyday at 17:28:00</Status>

</GlobalRetentionScheduleDefinition>
```

WARNING
Be sure that there is no white space at the beginning of each of these lines, otherwise the PUT operation fails.

In this example, the start hour has been changed to 17 and the start minute has been changed to 28.

NOTE
To disable the purge job, set <Enabled> to false. To reenable the purge job, set <Enabled> to true.

Results are returned in the Body tab of the HTTP Response pane.
For example:

```
<GlobalRetentionScheduleDefinitionList>

<GlobalRetentionScheduleDefinition version="1.0.0">

  <ID>9</ID>

  <StartMinute>28</StartMinute>

  <StartHour>17</StartHour>

  <Enabled>true</Enabled>

  <JobStatus>Has never run</JobStatus>

  <Status>Scheduled to run everyday at 17:28:00</Status>

  <StartSecond>0</StartSecond>
```

```
<Item version="1.0.0">

  <CreateTime>Thu Dec 15 15:52:20 EST 2011</CreateTime>

  <Name>Global Retention Schedule Definition</Name>

  </Item>

  </GlobalRetentionScheduleDefinition>

  </GlobalRetentionScheduleDefinitionList>
```

In this example, data that is older than the specified retention periods will be purged every day at 17:28:00.

Schedule Rollup Processing and Baseline Calculations

Administrators can schedule rollup processing and baseline calculations. Changing the time when these operations run lets administrators schedule these intensive operations to occur during off-hours. When these operations occur during off-hours, users who generate reports will not be impacted during business hours.

By default, rollup processing and baseline calculations are run at the bottom of the hour, every hour of the day.

Follow these steps:

1. Enter the following URL in a web browser:
   `http://DA_host:port/rest/rollups/config`
   – ***DA_host*:*port***
   Specifies the Data Aggregator host name and the port number.
   Default port: 8581
2. Take note of the ID that is assigned to the configuration item.
3. Open a REST client editor or HTTP tool that sends requests and gets responses. Enter the following criteria:
   – URL: `http://DA_host:port/rest/rollups/config/ID`
   • ***DA_host*:*port***
   Specifies the Data Aggregator host name and the port number.
   Default port: 8581
   • **ID**
   Is a unique identification number that is assigned to the configuration item. You noted this number in the previous step.
   – HTTP method: PUT
   – Enter the hour of the day when you want rollup processing to begin and end in the Body tab of the HTTP Request pane.

   By default, the following results are returned:

```
<RollupsConfigurationList>

     <RollupsConfiguration version="1.0.0">

          <ID>8</ID>

          <StartHour>0</StartHour>
```

```
            <EndHour>23</EndHour>

        </RollupsConfiguration>

    </RollupsConfigurationList>
```

- **StartHour**
  Defines the hour of the day (in your local timezone) in 24-hour time format when rollup processing will begin.
- **EndHour**
  Defines the hour of the day (in your local timezone) in 24-hour time format when rollup processing should end. No new rollups will be kicked off after the end-hour, but any rollups that are in-progress will be allowed to complete.

  NOTE
  For more details on these attributes, see http://*DA_host*:*port*/rest/rollups/config/documentation.

Example: In this example, you change the schedule so that rollup processing and baseline calculations run only from 20:00 to 7:00.

```
<RollupsConfigurationList>

    <RollupsConfiguration version="1.0.0">

      <ID>8</ID>

      <StartHour>20</StartHour>

      <EndHour>6</EndHour>

    </RollupsConfiguration>

</RollupsConfigurationList>
```

The <EndHour> is inclusive. In this example, that means if you specify 6 as the EndHour, rollup processing and baseline calculations will be initated in at the bottom of the hour during the 06:00 hour, but they will not be initiated at the 07:00 hour. Any calculations that are in progress will be allowed to complete.

WARNING
Any modifications to the default schedule can result in a larger delay in data showing up in reports pertaining to the corresponding resolution. For example, if hourly rollups are delayed, then a reporting showing hourly resolution data will not be current until the hourly rollup has been performed.

OpenAPI

The OpenAPI is a flexible tool that lets users easily extract data from the DX NetOps Performance Management database. The OpenAPI enables integration between DX NetOps Performance Management data and external applications.

The OpenAPI is a public API that uses the QueryBuilder GUI. The QueryBuilder is a guided URL builder that lets you create custom Query URLs to extract and explore performance data. The URLs return customized data in the specified format. You can view the data in a browser or process the data in a custom web application.

The OpenAPI uses the OData 2.0 industry standard. The documentation for this standard is found at the OData 2.0 web page.

When new metrics families are added to the system, the metrics are automatically added to the OpenAPI schema.

Access the OpenAPI

All users who have NetOps Portal credentials can use QueryBuilder and the OpenAPI. To log in to the OpenAPI QueryBuilder, use your NetOps Portal credentials. The OpenAPI uses Single Sign-On (SSO) for credential authentication.

> **NOTE**
>
> If the PC hostname is overridden, the Web Service Host must be updated using the Single Sign-On Configuration Tool. Otherwise, QueryBuilder cannot access SSO to log in users. If you override the PC hostname, update the Web Service Host property in the Single Sign-On Configuration Tool. For more information, see Update Performance Center Website Settings.

To access the QueryBuilder, go to the following URL: `http://DA_host:8581/odataquery`

In a fault tolerant environment, specify the proxy server as the *DA_host*.

> **NOTE**
>
> You **cannot** run OpenAPI queries outside the QueryBuilder as a user with Security Assertion Markup Language (SAML) 2.0 authentication. SAML authentication works in the QueryBuilder UI. LDAP authentication and NetOps Portal authentication support direct OpenAPI queries.

The schema XML description provides detailed information about the items and relationships in your system. To review the schema XML description and metadata, go to the following URL:
`http://DA_host:8581/odata/api/$metadata`

Use the OpenAPI QueryBuilder

The OpenAPI QueryBuilder creates query URLs that export configuration and polled data.

To access the QueryBuilder, go to the following URL: `http://DA_host:8581/odataquery`

In a fault tolerant environment, specify the proxy server as the *DA_host*.

Create an OpenAPI Query

To extract customized data sets from the data source, use the QueryBuilder to generate OpenAPI queries.

> **WARNING**
> Queries that return large sets of results can negatively affect your system. Refine your queries to return only the results that are relevant to your needs.

> **NOTE**
>
> The address bar of the Web browser updates to show the selected tokens in the Query Expression field. To continue editing the query, copy and save this URL

Follow these steps:

1. Click the Query Expression field to start a query.
2. Select the **for** option that represents what you want in the data set.
3. Add the **select**, **expand metrics**, and **expand** tokens to define the output data.
4. Add more tokens to refine the results.
   QueryBuilder creates the OData URL.

- To run the query in the QueryBuilder browser window, click **Run**.
- To run the query from a Web browser, REST tool, or custom application, copy the OData URL.

OpenAPI Controls

The OpenAPI QueryBuilder uses tokens that represent logical elements in the OpenAPI query syntax. Tokens appear when you click the Query Expression field. A token lets you select the type of attribute, filter, metric family, metric, or time range for the query. Selecting a token updates the Query URL.

Use tokens to create and export Query URLs:

- **for**
  This token determines the type of item that the query retrieves data about. This token appears only once in each query. Most options for this token include selections that set up an automatic filter. For example, when you select **interface** in the token, you can select **within Groups that...** This option adds a filter token with the group filter selected as the first criterion.

  NOTE

  If a newly added metric family does not appear in the QueryBuilder, refresh the Web browser.
- **select**
  This token determines the item properties to include in the data set.
  Property order is not supported in the results.
- **expand metrics**
  This token determines the performance metrics to include in the data set.
  Property order is not supported in the results.
- **filter**
  This token adds custom filters that are based on logical functions using the AND and OR operations. Select whether the filter is case-sensitive for all logical functions.
  When OData evaluates a filter expression, the 'Any' operator is used to determine whether the Boolean expression is True or False for a collection of items. The following examples return True:
  - `odata/api/devices?$select=ID,Name&$filter=((startswith(Name, 'cisco') eq true))`
  - `odata/api/cpus?$top=10&$filter=cpumfs/im_Utilization gt 80 where cpumfs/im_Utilization is the collection [60,65,81,68,61]`

  NOTE

  In large systems, some queries might time out if the filters are applied inefficiently. OpenAPI evaluates filter expressions in the order they appear in the query. Use the filter that limits the data more first. For example, if you want all groups where interface utilization is great than 50 percent in North America, place the North America filter first.
- **filter metrics**
  This token adds custom filters that are based on the selected metrics.
- **time range**
  This token limits the time range of the results for performance metrics and sets the granularity of the data.

  NOTE

  If the time granularity is set to Day and the Time Zone is set to any value other than the default, the Start Time and End Time fields are disabled.
- **expand**
  This token adds data that is related to the base item to the returned data set. For example, if you selected data for interface, use expend to get data about the related devices.
  In the expand token, select which columns to add to the data set. For example, for device, you might select name, primary IP address, and location.

Each expanded attribute appears in the HTML table in the same format as a metric column. For example, if to display interface information for a specific device, you can use the following query:

```
select=device/Name&$expand=device
```

- **group/aggregate**
  This token defines grouping and aggregation for the returned data set. For example, aggregate CPU utilization to the device level, and average the utilization for the last hour.

  NOTE

  The OpenAPI QueryBuilder currently supports only the '`groupby` ' and '`aggregate` ' OData transformations. If the starting entity is metric family, then the `groupby` transformation can be applied to `ID` , `DeviceItemID` , and `Timestamp` . If the starting entity if a config entity and aggregation is for metric family data, the `groupby` transformation can be applied to `ID` and `DeviceItemID` . The `groupby` transformation can now be done on the group level using group ID. The `aggregate` transformation can now include up to 5 expressions.
- **sort**
  This token controls the sorting of the query output.
  The OpenAPI does not support sorting on properties of a related entity. For example, the following query is supported:

  ```
  /cpus?$orderby=Name
  ```

  The following query is **not** supported:

  ```
  /cpus?$orderby=device/Name
  ```

  The OpenAPI supports sorting only on the properties of the target entity of your query.

  ```
  http://hostname:8581/odata/api/TargetEntity
  ```

  For example, sorting of metric columns works best when the target entity of your query is the metric family.
- **limit (top)**
  This token specifies the maximum number of rows or expanded rows in the query output.
  - **Maximum number of rows**
    The number of rows in the Results table
  - **Maximum number of expanded rows**
    The number of rows in the Expand window
    This number only applies to the number of rows when the Expand token is used.

    NOTE

    To modify the default values or the maximum number of rows, see Configure OpenAPI Defaults and Limits.
  - **Skip**
    The number of leading rows to omit from the query output
- **format**
  This token determines the format of the returned data set. The OpenAPI supports the following formats:
  - **HTML Table**
    If the query does not include this token, HTML Table is the default format.

    NOTE

    The HTML table format is supported only in the OpenAPI QueryBuilder. Direct OpenAPI queries support JSON, XML, Atom, and CSV.
  - **JSON**

NOTE

The JSON format removes the metadata from the resulting data set. Doing so reduces the network transfer time, the payload size up to 50-75 percent, and the client processing time. To add metadata to JSON output manually, use the following HTTP header:

```
Accept: application/json;odata=verbose
```

You can also add metadata to JSON output. To do so, add the $format query option as the following text to the OData URL:

```
$format=application/json;odata=verbose
```

- **XML**
- **Atom**
- **CSV**
  To specify CSV as a format manually, append the following text to the end of the OData URL:
  ```
  $format=text/csv
  ```

- **custom parameter**
  This token adds custom parameters in OData syntax to the OpenAPI query.

Find Metric Family Names for QueryBuilder

The metric family names in the OpenAPI are shortened versions of the internal metric family names. The OpenAPI metric family names remove the prefix 'Normalized' and the suffix 'Info'. The capitalization is removed and the suffix 'mf' is added to the string. For example, the internal name of the Interface metric family is 'NormalizedPortInfo'. The OpenAPI metric family name is 'portmf'.

To see the internal metric family name, look up the name in NetOps Portal.

Follow these steps:

1. Select **Administration**, and click the Data Aggregator data source.
2. Expand **Monitoring Configuration**, and click **Metric Families**.
3. Hover over one of the column headings, and click the down arrow.
4. Expand the Columns menu, and select **Internal Name**.
   The **Internal Name** column appears on the page.
5. Search for the metric family, and record the internal name.

Best Practices for OpenAPI Performance

The OpenAPI enables quick and flexible extraction of data from the database. For queries that produce large results, use the following best practices to improve performance:

- When you apply multiple filters to a query, ensure that all filters rules against the same object are adjacent. Splitting these filters reduces the efficiency of the query.
- Avoid filtering by strings. For example, to filter on a group, look up the group ID instead of filtering on the group name.
- Combine group expressions. Using an OR operator between a group expression and another expression is unsupported. For example, the following expression returns an error:

```
odata/api/interfaces?&$select=ID,Description&$filter=(groups/Name eq 'Manageable
Devices')  or (substringof('Gigabit',Description) eq true)
```

- Aggregation functions against large data sets take a long time.
- OpenAPI is not a bulk data export tool. Use OpenAPI to extract only the data that you need.
- Use the top and skip values to apply paging for large data sets.
- Use the largest granularity that is relevant for your data set. For example, if you want data that is aggregated by sum, the hourly or daily values provide the same information as rate data.

OpenAPI QueryBuilder Examples

The following examples highlight the flexibility of the OpenAPI. Use these examples as a model to create your own OpenAPI queries:

Export Inventory and Configuration Details

The following examples show how to extract lists of items from inventory with specific configuration details.

Get a List of All Devices with Cisco in the Name

You want to get a list of Cisco devices and IP addresses of those devices. For these devices, you know that the device name includes "Cisco".

To get this list, use the following tokens to build the query:

- **for: device**
- **select: ID, Name, PrimaryIPAddress**
- **filter: name contains cisco**

OpenAPI URL: `http://da_host:8581/odata/api/devices?$select=ID,Name,PrimaryIPAddress&$filter=((groups/Name eq 'cisco'))`

Get a List of Devices and Location Information

You want a list of Cisco devices and the location information (location, latitude, longitude, location description, and elevation) for those devices.

> **TIP**
>
> The **Location** attribute is tied to the MIB **sysLocation** value. The **Latitude**, **Longitude**, **LocationDesc**, and **Elevation** attributes are user-defined values. An administrator must set these values on the Data Aggregator. If these values are not set, they return a null result.

To get this list, use the following tokens to build the query:

- **for: device**
- **select: ID, Name, PrimaryIPAddress, Location, Latitude, Longitude, LocationDesc, Elevation**
- **filter: name contains cisco**

OpenAPI URL: `http://da_host:8581/odata/api/devices?$select=ID,Name,PrimaryIPAddress,Location,Latitude,Longitude,LocationDesc,Elevation&$filter=((groups/Name eq 'cisco'))`

Get a List of Devices and Components If the Device IP Address Begins with 10.251

You want a list of all devices where the IP address begins with 10.251. You also want a list of the components for those devices.

TIP

The **expand** token lets you get items that are related to the main entity of the query.

To get this list, use the following tokens to build the query:

- **for: device**
- **select: ID, Name, PrimaryIPAddress**
- **filter: PrimaryIPAddress starts with 10.251**
- **expand: components, DisplayName**

OpenAPI URL: `http://da_host :8581/odata/api/devices?$expand=components&$select=ID,Name,PrimaryIPAddress,components/DisplayName&$filter=((startswith(PrimaryIPAddress, '10.251') eq true))`

Get a List of Devices and Components in the Boston Group

You want a list of all devices in the Boston group. You also want a list of the components for those devices.

TIP

When you selected a group filter, OpenAPI always includes subgroups of the selected group.

To get this list, use the following tokens to build the query:

- **for: device**
- **select: ID, Name, PrimaryIPAddress**
- **filter: groups/Name equal Boston**
- **expand: components, DisplayName**

OpenAPI URL: `http://da_host:8581/odata/api/devices?$expand=components&$select=ID,Name,PrimaryIPAddress,components/DisplayName&$filter=((groups/Name eq 'Boston'))`

Get a List of Metric Families and Vendor Certifications for a Specific Device

You want to know which metric families and vendor certifications are used to monitor specific devices. From this list, you can derive the available metrics for a device. The name of the device is MyDevice-123.

To get this list, use the following tokens to build the query:

- **for: device**
- **filter: DisplayName equal MyDevice-123**
- **select: ID, Name**
- **expand: metricfamilyhistory, MetricFamilyID, VendorCertDisplayName**

OpenAPI URL: `http://da_host:8581/odata/api/devices?$expand=metricfamilyhistories&$select=ID,Name,metricfamilyhistories/MetricFamilyID,metricfamilyhistories/VendorCertDisplayName&$filter=((DisplayName eq 'MyDevice-123'))`

Get a List of Supported Metric Families

You want to know which metrics are available for reporting. From a list of metric families, you can determine which metric families are supported by which monitored devices.

To get this list, use the following tokens to build the query:

- **for: metricfamilyhistory**
- **select: Status**
- expand: device/ID, device/Name, metricfamilydef/ID, metricfamilydef/DisplayName, metricfamilydef/DisplayDescription, **metricdefs/ID, metricdefs/DisplayName, metricdefs/DisplayDecsription,**
- **filter: Status/Status equal SUPPORTED**

OpenAPI URL:

```
http://da_host:8581/odata/api/metricfamilyhistories?$expand=metricfamilydef,metricdefs,device&
$select=device/ID,device/Name,Status,metricfamilydef/ID,metricfamilydef/DisplayName,metricfamilydef/
DisplayDescription,metricdefs/ID,metricdefs/DisplayName,metricdefs/DisplayDescription&$filter=((Status eq
'SUPPORTED'))
```

Export Inventory and Monitoring Data

The following examples show how to extract monitoring data for items with specific configuration details or monitoring status.

List the Interfaces in the North America Group That Are Not Being Polled

You want to know which interfaces in the North America group are not currently being monitored.

To get this list, use the following tokens to build the query:

- **for: interface**
- **select: ID, Name**
- **filter: groups/Name equal North America and IsPolled False**

OpenAPI URL: `http://da_host:8581/odata/api/interfaces?$select=ID,Name&$filter=((groups/Name eq 'North America') and (IsPolled eq false))`

Get the Polling Statistics for Devices in a Group for the Last 24 Hours

You want to know the success and failure rate of devices in the Boston group for the last 24 hours. The database does not store the number of unsuccessful polls. To derive this value, use the **group aggregate** token to subtract the number of successful polls from the number of polls sent.

> **TIP**
>
> Time selections are absolute. When you select the time for the query, that time is fixed. To create a query with a flexible time, use a script to replace the time and data in the query.

To get this data, use the following tokens to build the query:

- **for: device**
- **select: Name**
- **expand metrics: devicepollingstatisticsmfs: im_NumSuccessfulPolls**
- **group aggregate:**
  - **Group By: devicepollingstatisticsmfs, ID**
  - Add an Aggregation Function: **Sum Of** `im_NumPollsSent sub im_NumSuccessfulPolls`
- **filter: groups/Name equal Boston**
- **time range: Last24Hours**

OpenAPI URL:

```
http://da_host :8581
```

```
/odata/api/devices?$apply=groupby(devicepollingstatisticsmfs/ID,
 aggregate(devicepollingstatisticsmfs(im_NumPollsSent sub im_NumSuccessfulPolls
 with sum as Value)))&period=1d&resolution=RATE&$expand=devicepollingstatisticsmfs&
$select=Name,devicepollingstatisticsmfs/im_NumSuccessfulPolls&$filter=((groups/Name eq 'Boston'))
```

Get a List of Devices with More Than 25 Percent Polling Failures in the Last 30 Days

You want to identify devices that respond inconsistently to poll requests. For this query, use the aggregation function to calculate the poll failure rate, then set a limit against that value.

> **TIP**
>
> The **group aggregate** token enables you to derive values from metrics in the system. In the results, the aggregated data is reported as "Value". You can use Value as a filter criteria for the filter metrics token.

To get this data, use the following tokens to build the query:

- **for: device**
- **select: Name**
- **group aggregate:**
  - **Group By: devicepollingstatisticsmfs, ID**
  - Add an Aggregation Function: **Sum Of** `((im_NumPollsSent sub im_NumSuccessfulPolls) div im_NumPollsSent) mul 100`
- **filter metrics: devicepollingstatisticsmfs/Value greater 25**
- **time range: Last30Days**

OpenAPI URL:

```
 http://da_host :8581
/odata/api/devices?$apply=groupby(devicepollingstatisticsmfs/ID,
 aggregate(devicepollingstatisticsmfs(((im_NumPollsSent sub im_NumSuccessfulPolls) div im_NumPollsSent) mul
 100 with sum as Value)))&period=1m&resolution=RATE&$select=Name&$filter=((devicepollingstatisticsmfs/Value gt
 25))
```

Export Device Status

The following examples show how to extract details regarding device statuses.

Get a List of All Devices from the Last 24 Hours with Management Issues

You want to identify recent devices with management issues.

To get this data, use the following tokens to build the query:

- **for: device**
- **select: ID, Name**
- **group aggregate:**
  - **Group By: availabilitymfs, DeviceItemID**
  - Add an Aggregation Function: **Count Distinct Of** `(Timestamp)`
- **filter metrics: availabilitymfs/Value less 1**
- **time range: Last24Hours/As polled**
- **limit(top): top=2000, skip=0, expandtop=0**

OpenAPI URL: *http://da_host* :8581/odata/api/devices?$apply=groupby(availabilitymfs/DeviceItemID, aggregate(availabilitymfs(Timestamp with countdistinct as

```
Value)))&resolution=RATE&period=1d&$top=2000&$skip=0&top=0&$select=ID,Name&
$filter=((availabilitymfs/Value lt 1))
```

Get a List of All Devices from the Last 24 Hours with Reachability Issues

You want to identify recent devices with reachability issues.

To get this data, use the following tokens to build the query:

- **for: device**
- **select: ID, Name**
- **group aggregate:**
  - **Group By: reachabilitymfs, DeviceItemID**
  - Add an Aggregation Function: **Count Distinct Of** `(Timestamp)`
- **filter metrics: reachabilitymfs/Value less 1**
- **time range: Last24Hours/As polled**
- **limit(top): top=2000, skip=0, expandtop=0**

OpenAPI URL: http://*da_host*
:
```
8581/odata/api/devices?$apply=groupby(reachabilitymfs/DeviceItemID,
aggregate(reachabilitymfs(Timestamp with countdistinct as
Value)))&resolution=RATE&period=1d&$top=2000&$skip=0&top=0&$select=ID,Name&
$filter=((reachabilitymfs/Value lt 1))
```

Get a List of All Devices from the Last 24 Hours that are Up without a Single CPU Polled

You want to identify devices that need CPU metric polling restarted.

To get this data, use the following tokens to build the query:

- **for: device**
- **select: ID, Name**
- **group aggregate:**
  - **Group By: cpumfs, DeviceItemID**
  - Add an Aggregation Function: **Count Distinct Of** `(Timestamp)`
- **filter metrics: cpumfs/Value less 1**
- **time range: Last24Hours/As polled**
- **limit(top): top=2000, skip=0, expandtop=0**

OpenAPI URL: http://*da_host*
:
```
8581//odata/api/devices?$apply=groupby(cpumfs/DeviceItemID, aggregate(cpumfs(Timestamp
with countdistinct as Value)))&resolution=RATE&period=1d&$top=2000&$skip=0&top=0&
$select=ID,Name&$filter=((cpumfs/Value lt 1))
```

Get a List of All Devices from the Last 24 Hours that are Up without a Single Interface Polled

You want to identify that the devices that need Interface metric polling restarted.

To get this data, use the following tokens to build the query:

- **for: device**
- **select: ID, Name**
- **group aggregate:**

  - **Group By: portmfs, DeviceItemID**
  - Add an Aggregation Function: **Count Distinct Of** `(Timestamp)`
- **filter metrics: portmfs/Value less 1**
- **time range: Last24Hours/As polled**
- **limit(top): top=2000, skip=0, expandtop=0**

OpenAPI URL: http://*da_host*
:
```
8581/odata/api/devices?$apply=groupby(portmfs/DeviceItemID, aggregate(portmfs(Timestamp
with countdistinct as Value)))&resolution=RATE&period=1d&$top=2000&$skip=0&top=0&
$select=ID,Name&$filter=((portmfs/Value lt 1))
```

Extract Inventory and Time-Series Data

The following examples show how to extract trend and time-series data for items with specific configuration details.

Export As-Polled Bits-In and Bits-Out for Interfaces in a Group

You want to extract key interface monitoring data from the Boston group to use in a new visualization. You want to extract one week of as polled data.

> **TIP**
>
> To get the as polled data for a week, set the maximum number of expanded rows to the correct value. For this example, the poll interval is 5 minutes, so 2016 data point is one week of data.

To get this data, use the following tokens to build the query:

- **for: interface**
- **select: name**
- **expand metrics: portmfs: im_BitsIn, im_BitsOut**
- **filter: groups/Name equal Boston**
- **time range: Last7Days/As polled**
- **limit: Maximum number of expanded rows: 2016**

OpenAPI URL:
```
   http://da_host :8581
/odata/api/interfaces?period=1w&resolution=RATE&$top=50&$skip=0&top=2016&$expand=portmfs&$select=Name,portmfs/
im_BitsIn,portmfs/im_BitsOut&$filter=((groups/Name eq 'Boston'))
```

Get Total Hourly Bits-In and Bits-Out for Interfaces Associated to Devices by IP Address Substring

You want to extract key throughput information for interfaces where the parent device IP address begins with 10.251. You want the hourly throughput totals for the month.

> **TIP**
>
> For 30 days of hourly data, set the maximum number of expanded rows to 720

To get this data, use the following tokens to build the query:

- **for: interface**
- **select: name**
- **expand metrics: portmfs: im_BitsIn, im_BitsOut**
- **filter: device/PrimaryIPAddress starts with 10.251**
- **time range: Last30Days/Hour**
- **limit: Maximum number of expanded rows: 720**

OpenAPI URL:

```
   http://da_host :8581
/odata/api/interfaces?period=1m&resolution=HOUR&$top=50&$skip=0&top=720&$expand=portmfs&$select=Name,portmfs/
im_BitsIn,portmfs/im_BitsOut&$filter=((startswith(PrimaryIPAddress, '10.251') eq true))
```

Get Total Daily Bits-In & Bits-Out for Interfaces That Have Less Than 1-GB Daily Throughput for the Last 30 Days

You want to identify which interfaces are underutilized.

> **TIP**
>
> Because bits in and bits out aggregate by sum, daily resolution provides the same value and the query runs faster.

To get this data, use the following tokens to build the query:

- **for: interface**
- **select: name**
- **expand metrics: portmfs: im_BitsIn, im_BitsOut**
- **filter metrics: portmfs/im_Bits less 1000000000**
- **time rang: Last30Days/Day**

OpenAPI URL:

```
   http://da_host :8581
/odata/api/interfaces?period=1m&resolution=DAY&$expand=portmfs&$select=Name,portmfs/im_BitsIn,portmfs/
im_BitsOut&$filter=((portmfs/im_Bits lt 1000000000))
```

Extract Inventory and Aggregated Values

The following examples show how to extract aggregated data for items with specific configuration details.

> **TIP**
>
> The **group aggregate** token applies customizable run-time analytics to the data set.
>
> **NOTE**
>
> You cannot mix percentile aggregations with other aggregation types.

Show Me the 95th Percentile Utilization Value for Each Interface in a Group for the Last Seven Days

You want the 95th percentile of utilization for each interface in the Boston group.

To get this data, use the following tokens to build the query:

- **for: interface**
- **select: name**
- **expand metrics: portmfs: im_Utilization**
- **group aggregate:**

  - **Group By: portmfs, ID**
  - Add an Aggregation Function: **Percentile95** `im_Utilization`
- **filter: groups/Name equal Boston**
- **time range: Last7Days/As polled**

OpenAPI URL:

```
http://da_host :8581/odata/api/interfaces?$apply=groupby(portmfs/ID, aggregate(portmfs(im_Utilization with
percentile95 as Value)))&resolution=RATE&period=1w&$expand=portmfs&$select=Name,portmfs/ID&$filter=((groups/
Name eq 'Boston'))
```

Show Me the Average CPU Utilization for Each Device in a Group for the Last 30 Days

You want to the average CPU utilization for routers in the Boston group

To get this data, use the following tokens to build the query:

- **for: device**
- **select: name**
- **group aggregate:**
  - **Group By: cpumfs, ID**
  - Add an Aggregation Function: **AverageOf** `im_Utilization`
- **filter: groups/Name equal Boston**
- **time range: Last30Days/As polled**

OpenAPI URL:

```
http://da_host :8581
/odata/api/devices?$apply=groupby(cpumfs/ID, aggregate(cpumfs(im_Utilization with average as
Value)))&period=1m&resolution=RATE&$select=Name&$filter=((groups/Name eq 'Boston'))
```

Show Me the Interfaces in a Group That Have Averaged Less Than 40 Percent Hourly Utilization

You want to see which interfaces in your network were underutilized during the last quarter.

To get this data, use the following tokens to build the query:

- **for: interface**
- **select: name**
- **expand metrics: portmfs: im_utilization**
- **group/aggregate**
  - **Group By: portmfs, ID**
  - Add an Aggregation Function: **AverageOf** `im_utilization`
- **filter metrics: portmfs/Value less 40**
- **time range: Last3Months/Hour**

OpenAPI URL:

```
http://da_host :8581
/odata/api/interfaces?$apply=groupby(portmfs/ID, aggregate(portmfs(im_Utilization with average as
Value)))&period=3m&resolution=HOUR&$expand=portmfs&$select=Name,portmfs/im_Utilization&$filter=((portmfs/
Value lt 40))
```

Show Me the Devices with Average CPU and Memory Utilization More Than 75 Percent Based on IP Address Substring

You want to see devices in the subnet starting with 10.251 that have high CPU and memory utilization.

To get this data, use the following tokens to build the query:

- **for: device**
- **select: Name**
- **expand metrics: cpumfs: im_MemoryUtilization, im_Utilization**
- **filter metrics: cpumfs/im_MemoryUtilization greater 75 OR cpumfs/im_Utilization greater 75**
- **filter: PrimaryIPAddress starts with 10.251**

OpenAPI URL:

```
   http://da_host :8581
/odata/api/devices?$expand=cpumfs&$select=Name,cpumfs/im_MemoryUtilization,cpumfs/im_Utilization&
$filter=((cpumfs/im_MemoryUtilization gt 75) or (cpumfs/im_Utilization gt 75)) and
 ((startswith(PrimaryIPAddress, '10.251') eq true))
```

Extract Metrics Aggregated at the Group Level

The following examples show how to extract metrics aggregated at the group level. Use these queries to understand capacity that is related to the logical groupings of resources. For example, by customers, geography, services, or other.

> **NOTE**
>
> In large systems, some queries might time out if the filters are applied inefficiently. OpenAPI evaluates filter expressions in the order they appear in the query. Use the filter that limits the data more first. For example, if you want all groups where interface utilization is greater than 50 percent in North America, place the North America filter first.

Show Me the Top Ten Interfaces in Group ID 10 with the Highest Bandwidth Utilization

You want to see the top ten interfaces in Group ID 10 with the highest bandwidth utilization.

To get this data, use the following tokens to build the query:

- **for: portmf**
- **select: ID, im_utilization**
- **expand: interface/Name**
- **filter: groups/ID equal 10**
- **sort: im_Utilization (DESC)**
- **limit(top): top=10, skip=0, expandtop=10**

OpenAPI URL: `http://da_host:8581 /odata/api/portmfs?$orderby=im_Utilization desc&$top=10&$skip=0&top=10&$expand=interface&$select=ID,im_Utilization,interface/Name&$filter=((groups/ID eq 10))`

Show Me the Average Interface Utilization of Each Interface in Group ID 10

You want to know the average interface utilization of each interface in Group ID 10.

> **NOTE**
>
> The following query works best for a small group because paging on aggregated values is unsupported.

To get this data, use the following tokens to build the query:

- **for: group**
- **group/aggregate: groupby(portmfs/ID, aggregate(portmfs(im_utilization with average as Value)))**
- **expand: interfaces/ID, interfaces/Name**
- **select: ID, Name, Description**

NOTE
Note: Value is selected by default on the server side.

- **filter: ID equal 10**

OpenAPI URL: `http://da_host:8581 /odata/api/groups?$apply=groupby(portmfs/ID, aggregate(portmfs(im_Utilization with average as Value)))&$expand=interfaces&$select=ID,Name,Description,interfaces/ID,interfaces/Name&$filter=((ID eq 10))`

Show Me the Average Interface Utilization for All Interfaces Within Group ID 10 and Group ID 11

You want to know the average interface utilization for all interfaces within Group ID 10 and Group ID 11.

NOTE

The following query works best for small groups because paging on aggregated values is unsupported.

To get this data, use the following tokens to build the query:

- **for: group**
- **group/aggregate: groupby(portmfs/ID, aggregate(portms(im_Utilization with average as Value)))**
- **select: ID, Name**

NOTE
is selected by default on the server side.

- **filter: (ID equal 10) or (ID equal 11)**

OpenAPI URL: `http://da_host:8581 /odata/api/groups?$apply=groupby(portmfs/ID, aggregate(portmfs(im_Utilization with average as Value)))&$select=ID,Name&$filter=((ID eq 10) or (ID eq 11))`

Show Me the Maximum Interface Utilization Value for Each Child Group Within the Boston Group

You want to know the maximum interface utilization value for each child group within the Boston group.

To get this data, use the following tokens to build the query:

- **for: group**
- **group/aggregate: groupby(portmfs/ID, aggregate(portms(im_Utilization with max as Value)))**
- **select: ID, Name**

NOTE
is selected by default on the server side.

- **filter (GroupPathLocation contains Boston)**

OpenAPI URL: `http://da_host:8581 /odata/api/groups?$apply=groupby(portmfs/ID, aggregate(portmfs(im_Utilization with max as Value)))&$select=ID,Name&$filter=((substringof('Boston:', GroupPathLocation) eq true))`

Show Me the 95th Percentile of Interface Utilization for Each Group That Has the Word "Boston" in the Description

You want to know the 95th percentile of interface utilization for each group with the word "Boston" in the Description.

To get this data, use the following tokens to build the query:

- **for: group**
- **group/aggregate: groupby(portmfs/ID, aggregate(portms(im_Utilization with percentile95 as Value)))**
- **select: ID, Name**

NOTE
is selected by default on the server side.

- **filter (Name contains Boston)**

OpenAPI URL: `http://da_host:8581 /odata/api/groups?$apply=groupby(ID, aggregate(portmfs(im_Utilization with percentile95 as Value)))&$select=ID,Name& $filter=((substringof('Boston',Name) eq true))`

Show Me the Top Five Group IDs with the Highest Group Interface Utilization for All Child Groups in the Boston Group

You want to see the top five groups with the highest group interface utilization for all child groups in the Boston group.

To get this data, use the following tokens to build the query:

- **for: portmf**
- **group/aggregate: groupby(groups/ID,aggregate(im_Utilization with average as Value))**
- **filter: (groups/GroupPathLocation contains Boston) and (Value greater than or equal 0)**
- **select: ID, Value**
- **sort: Ordering, Value (DESC)**

OpenAPI URL:

```
http://da_host:8581
/odata/api/portmfs?$apply=groupby(groups/ID, aggregate(im_Utilization with average as Value))&$orderby=Value
desc&$select=ID,Value&$filter=((substringof('Boston:',groups/GroupPathLocation) eq true) and (Value ge 0))
```

Extract Metrics within Specified Business Hours

The following examples show how to extract metrics within specified business hours. For more information see, Advanced OpenAPI Query Examples.

Show Me Data for Monday through Friday 9:00 AM to 5:00 PM Eastern Standard Time (EST)

You want to see data within your business hours, which are Monday through Friday 9:00 AM to 5:00 PM EST.

- **for: group**
- **group/aggregate: groupby(portmfs/ID, aggregate(portms(im_Utilization with average as Value)))**
- **time range: Last 7 Days/As polled/-05/Mon,Tue,Wed,Thu,Fri 9:00-17:00**
- **select: ID, Name**

OpenAPI URL: `http://da_host:8581/odata/api/groups?$apply=groupby(portmfs/ID, aggregate(portmfs(im_Utilization with average as Value)))&resolution=RATE&period=1w&bh=Mon-Fri 9:00-17:00&tz=-05&$select=ID,Name`

Show Me Data for Monday, Tuesday, and Friday 9:00 AM to 12:00 PM and 1:00 PM to 6:00 PM

You want to see data within your business hours, which are Monday, Tuesday, and Friday 9:00 AM to 12:00 PM and 1:00 PM to 6:00 PM.

- **for: group**
- **group/aggregate: groupby(portmfs/ID, aggregate(portms(im_Utilization with average as Value)))**
- **time range: Last 7 Days/As polled/Mon,Tue,Fri 9:00-12:00 13:00-18:00**
- **select: ID, Name**

OpenAPI URL:

http://da_host:8581/odata/api/groups?$apply=groupby(portmfs/ID, aggregate(portmfs(im_Utilization with average as Value)))&resolution=RATE&period=1w&bh=Mon-Tue 9:00-12:00 13:00-18:00,Fri 9:00-12:00 13:00-18:00&$select=ID,Name

Show Me Data for Monday through Friday 8:00 PM to 4:00 AM

You want to see data within your business hours, which are Monday through Friday 8:00 PM to 4:00 AM.

- **for: group**
- **group/aggregate: groupby(portmfs/ID, aggregate(portms(im_Utilization with average as Value)))**
- **time range: Last 7 Days/As polled/Mon,Tue,Wed,Thur,Fri 20:00-4:00**
- **select: ID, Name**

OpenAPI URL: *http://da_host:8581/*odata/api/groups?$apply=groupby(portmfs/ID, aggregate(portmfs(im_Utilization with average as Value)))&resolution=RATE&period=1w&bh=Mon-Fri 20:00-4:00&$select=ID,Name

Show Me Data for Monday through Friday 9:00 AM to 5:00 PM in January

You want to see data for January within your business hours, which are Monday through Friday 9:00 AM to 5:00 PM.

- **for: group**
- **group/aggregate: groupby(portmfs/ID, aggregate(portms(im_Utilization with average as Value)))**
- **time range: 2017-01-02T09:00~2017-01-31T17:00/As polled/Mon,Tue,Wed,Thu,Fri 9:00-17:00**
- **select: ID, Name**

OpenAPI URL: *http://da_host:8581/odata/api/groups?$apply=groupby(portmfs/ID, aggregate(portmfs(im_Utilization with average as Value)))&resolution=RATE&starttime=1483365600&endtime=1485900000&bh=Mon-Fri 9:00-17:00&$select=ID,Name*

Extract SD-WAN Data

The following examples show how to extract SD-WAN data.

Show Me All Tunnels and Attributes in a Group

You want to see data for your SD-WAN tunnels.

- **for: sdntunnel**
- **filter: groups/Name contains Site 1**

OpenAPI URL: http://*da_host*:8581/odata/api/sdntunnels?$filter=((substringof('Site 1', groups/Name) eq true))

Show Me the Latency, Loss, and Jitter Values for All Tunnels in a Group

You want to see the latency, loss, and jitter values for your SD-WAN tunnels.

- **for: sdntunnel**
- **filter: groups/Name contains Site 1**
- **expand metrics: sdntunnelmfs(im_Jitter,im_Latency,im_PacketLossPercentage)**

OpenAPI URL: *http://da_host:8581/odata/api/sdntunnels?$expand=sdntunnelmfs&$select=sdntunnelmfs/std_im_Jitter,sdntunnelmfs/std_im_Latency,sdntunnelmfs/std_im_PacketLossPercentage&$filter=((substringof('Site 1', groups/Name) eq true))*

Show Me the System Metrics for the Source and Destination Devices for All Tunnels in a Group

You want to see the system metrics for your source and destination devices.

- **for: sdntunnel**
- **filter: groups/Name contains Site 1**
- **expand metrics: sdntunnelmfs(std_im_Jitter,std_im_Latency,std_im_PacketLossPercentage)**

OpenAPI URL: *http://da_host:8581/odata/api/sdntunnels?$expand=sdntunnelmfs&$select=sdntunnelmfs/std_im_Jitter,sdntunnelmfs/std_im_Latency,sdntunnelmfs/std_im_PacketLossPercentage&$filter=((substringof('Site 1', groups/Name) eq true))*

Show Me the Virtual Interface Metrics for the Source and Destination Interfaces in a Group

You want to see the virtual interface metrics for your source and destination interfaces.

- **for: sdnvirtualinterface**
- **filter: groups/Name contains Site 1**
- **expand metrics: stdvitualinterface(std_im_BitsIn,std_im_BitsOut)**

OpenAPI URL: *http://da_host:8581/*odata/api/adnvirtualinterfaces?$expand=virtualinterfacemfs&$select=virtualinterfacemfs/std_im_BitsIn,virtualinterfacemfs/std_im_BitsOut&$filter=((substringof('Site 1', groups/Name) eq true))

Advanced OpenAPI Query Examples

To take further advantage of the flexibility of the OpenAPI, use advanced OpenAPI queries. You can use advanced OpenAPI queries to build your own OpenAPI applications.

The OpenAPI service declares its structure in the Metadata Document, so that you can view the following details:

- The requests that can be executed
- The structure of the results
- How the service can be navigated

You can invoke the Metadata Document using the following URI:

http://*da_host*:8581/odata/api/$metadata

Special Aggregation Functions

Special aggregation functions can be used to:

- Calculate projection values
- Determine the projected time to a threshold
- Get the properties of the linear data model such as the slope and intercept that are used for these calculations

The following special aggregation functions are available for these calculations:

- **intercept**
  Where the projection line crosses the Y-axis
- **slope**
  The steepness of the projection line
- **projection**
  A line predicting the future performance of a metric
- **datasetcount**

The number of data points that are used to calculate the linear data model

- **timetothreshold**
  The number of seconds required for the projection line to cross a threshold
  This value is calculated from the end of the time interval that is defined by the `starttime` and `endtime` parameters

  NOTE

  Note: If `starttime` and `endtime` are unspecified, the default time interval is used.

Threshold Projection Parameters

To extract projections and threshold projections, include the following parameters in advanced OpenAPI queries:

- **prjoffset**
  The amount of time in seconds to calculate out the projection
- **threshold**
  The threshold value for the query

Extract Projections and Threshold Projections

Projections predict future performance. Threshold projections indicate when a metric is predicted to cross a threshold. Use this functionality to manage the capacity of your infrastructure and reprovision or acquire hardware as needed.

The following examples show how to extract projections and threshold projections.

Show Me the Projection for a Single Metric

You want to see the projected CPU utilization for the next 24 hours starting from Wednesday, Aug 10, 2016 at 14:05:00 GMT.

To get this data, use the following OpenAPI URL:

```
http://da_host
:8581
/odata/api/cpumfs?apply=groupby(ID,aggregate(im_Utilization with projection as Value))&
$select=ID,Value&starttime=1470837900&endtime=1470927900&resolution=DAY&prjoffset=86400
```

Show Me the Threshold Projection for a Single Metric

You want to see when CPU utilization is predicted to cross 40 percent. The default resolution for calculating the data is as polled. The threshold projection is returned in seconds.

To get this data, use the following OpenAPI URL:

```
http://da_host
:8581
/odata/api/cpumfs?$apply=groupby(ID,aggregate(im_Utilization with timetothreshold as Value))&
$select=ID,Value&threshold=40
```

Show Me the Threshold Projection for a Specified Resolution

You want to see when the CPU utilization is predicted to cross 40 percent. Your desired resolution for calculating the data is daily. The threshold projection is returned in seconds.

To get this data, use the following OpenAPI URL:

```
http://da_host
:8581
/odata/api/cpumfs?$apply=groupby(ID,aggregate(im_Utilization with timetothreshold as Value))&
$select=ID,Value&resolution=DAY&threshold=40
```

Show Me Multiple Projections from a Single Query

You want to see the projected throughout (bits per second) and discard rate for both inbound and outbound traffic in 60 days. You want to use hourly data from the past three months.

To get this data, use the following OpenAPI URL:

```
http://da_host:8581/odata/api/portmfs?apply=groupby(ID,aggregate((im_BitsPerSecondIn
with projection as im_BitsPerSecondIn),(im_BitsPerSecondOut with
projection as im_BitsPerSecondOut),(im_DiscardsIn with projection as
im_DiscardsIn),(im_DiscardsOut with projection as im_DiscardsOut)))&
$select=ID,Value&starttime=1463320821&endtime=1471269621&resolution=HOUR&prjoffset=5184000
```

Business Hours Parameters

To extract data based on business hour time ranges, include the following parameters in advanced OpenAPI queries:

- **duration**
  Machine-based time interval in time units (**s** second, **h** hour, **d** day, **w** week)
  Machine-based time models a quantity or amount of time in terms of seconds. Machine-based time can be accessed using other duration-based units, such as hours, days, and weeks. In addition, the days unit is treated as exactly equal to 24 hours, thus ignoring daylight savings effects. Use this parameter when you want to ignore daylight savings effects. You can use with `starttime` and `endtime` parameters to specify the start point or endpoint of the duration.
  Example: `duration=1w`
- **period**
  Human-based time interval in time units (**s** second, **h** hour, **d** day, **w** week, **m** month, **y** year)
  Human-based time models a quantity or amount of time in terms of years, months, and days. Human-based time includes daylight savings and other effects. Use this parameter when you want to account for daylight savings effects. You can use with `starttime` and `endtime` parameters to specify the start point or endpoint of the period.
  Example: `period=1w`
- **tz**
  Time zone offset from Greenwich Mean Time (GMT) applied to the start and end times of the query and any specified business hours. If no `tz` parameter is used, the default time zone of the server is used.
  Example: `period=20d&tz=-4:00`

  NOTE

  If you use a daily or weekly resolution in your query, the `tz` parameter is ignored.
- **bh**
  Business hours
  Examples:
  ```
  bh=Mon-Fri
  bh=Mon,Tue,Fri
  bh=Mon 8:30am-5:30pm
  bh=Mon 8:30-17:30
  bh=Mon 8:30-12:30 13:30-17:30
  ```

You can use `starttime` and `endtime` parameters with the business hours parameters as shown in the following examples:

- `starttime=1493145475&endtime=1493145775&bh=Mon-Fri 8:30am-5:30pm`
- `starttime=2017-03-15T18:00:00&period=20d&bh=Mon-Fri 8:30am-5:30pm&tz=-8:00`
- `endtime=2017-04-15T18:00:00&period=20d&bh=Mon-Fri 8:30am-5:30pm&tz=-8:00`
- `startime=2017-03-15T18:00:00&period=20d&bh=Mon-Fri 8:30am-5:30pm&tz=-8:00`
- `starttime=2017-03-15T18:00:00&endtime=2017:04:15T18:00:00&bh=Mon-Fri 8:30am-5:30pm`

Custom Functions

The following custom functions are available:

- **getSchemaVersion**
  See the schema version information.
  Example: *http://da_host:8581/odata/api/getSchemaVersion*
- **getDataCollectors**
  See a list of your Data Collectors and their details including hostname, IP address, and status.**Example:** *http://da_host:8581/odata/api/getDataCollectors*
- **getGroupMetricFamilies**
  See a list of metric families by group.**Example:** *http://da_host:8581/odata/api/getGroupMetricFamilies?&ID=GroupID$format=json&timeout=30*
  - **GroupID**
    Specify the Group ID from the Data Aggregator.
- **getDataAggregators-**
  See a list of Data Aggregators and their details including hostname, IP address, and status.**Example:** *http://da_host:8581/odata/api/getDataAggregators*
- **getResultLimiters**
  See a list of result limiters.**Example:** *http://da_host:8581/odata/api/getResultLimiters*

Configure OpenAPI Defaults and Limits

To perform operations that heavily affect the system during off-hours, override the QueryBuilder properties. You can override several parameters in the QueryBuilder. The default parameter values that are defined in the OData limiters configuration file protect the system from OpenAPI queries that negatively affect the overall system performance. These parameters either limit the returned set of results, or define the timeout threshold for potentially large operations. You can customize and override the default parameter values, which vary according to the scale and capability of the system.

Override an OpenAPI Parameter

To override a parameter, append the following code to the OpenAPI URL:

```
&<Override_parameter>=<override_value>
```

For example, to override the number of rows that are returned for a device query, use the following URL:

```
http://da_host:8581/odata/api/devices?$expand=cpumfs&$select=Name,PrimaryIPAddress,cpumfs/
im_MemoryUtilization,cpumfs/im_Utilization
&$top=200
```

The following parameters can be overridden:

Parameter	Default Value	Override	Description
`defaultTopLimit`	50	$top	Number of rows to return
`defaultExpandTopLimit`	100	top	Custom parameter for number of expanded rows to return
`defaultQueryTimeoutSecs`	30 (sec)	timeout	Custom parameter for overall time query can execute before timeout exception

Configure Web Service Parameter

The OpenAPI web service parameters limit concurrent use and limit the processing time for requests beyond that limit.

To override a web service parameter, edit the value in the following configuration file:
/opt/IMDataAggregator/apache-karaf-<vers>/etc/com.ca.im.odata.filters.OpenAPIRequestLimiterFilter.cfg

You can override the following web service parameters.

- **maxRequests**Number of simultaneous OpenAPI requests. Other requests are suspended until one of the current queries finishes.**Default:** 4
- **suspendMs**The length of time, in milliseconds, that each additional request is suspended when the OpenAPI reaches the maximum number of simultaneous requests. For example, specifying 20000 suspends the request for 20 seconds, after which time the request is rejected. A rejected request generates the 503 Unavailable error, and the OpenAPI does not run the query again automatically.**Default:** -1 (Use the value of defaultQueryTimeoutSecs)
- **waitMS**The length of time, in milliseconds, to wait before trying to accept a new request. This parameter is used when the **maxRequests** limit is reached. We recommend setting this parameter to the same value as `defaultQueryTimeoutSecs` in the following file:/opt/IMDataAggregator/apache-karaf-<*vers*>/etc/com.ca.im.odata.beans.ODataLimiters.cfg

Configure Maximum Results

To modify the defaults or increase the limit, configure the values for the number of rows that OpenAPI queries return.

Follow these steps:

1. Log on to the Data Aggregator host.
2. Locate and edit the following file:
   `/opt/IMDataAggregator/apache-karaf-<vers>/etc/com.ca.im.odata.beans.ODataLimiters.cfg`
3. Modify the file to set the limits and defaults. The following example shows the default values for this file. The bold attributes control the limits and defaults:

```
defaultTopLimit=50


defaultExpandTopLimit=100


maxTopLimit=20000
```

```
maxSubQueryLimit=2000000


defaultRateTimeIntervalSecs=3600

defaultHourlyTimeIntervalHours=168

defaultDailyTimeIntervalDays=30

defaultWeeklyTimeIntervalWeeks=52

defaultQueryTimeoutSecs=30

maxQueryTimeoutSecs=120
```

- **defaultTopLimit**
  Defines the default value for the maximum number of rows in the output.
  Default: 50
- **defaultExpandTopLimit**Defines the default value for the maximum number of expanded rows in the output.
  Default: 100
- **maxTopLimit**Defines the limit for the value in the maximum number of rows in the output.
  Default: 20000
- **maxSubQueryLimit**
  Defines the maximum number of expanded rows that an OpenAPI query can return. In QueryBuilder, the limit for the maximum number of expanded rows is calculated by dividing this value by the specified value for the maximum number of rows.
  Default: 2000000
- **defaultRateTimeIntervalSecs**
  Defines the default value for the query time interval when the resolution equals rate. If the time interval is unspecified, the query time is for 3,600 seconds (one day).
  Default: 3600
- **defaultHourlyTimeIntervalHours**
  Defines the default value for the query time interval when the resolution equals hour. If the time interval is unspecified, the query time is for 168 hours (7 days).
  Default: 168
- **defaultDailyTimeIntervalDays**
  Defines the default value for the query time interval when the resolution equals day. If the time interval is unspecified, the query time is for 30 days.
  Default: 30
- **defaultWeeklyTimeIntervalWeeks**
  Defines the default value for the query time interval when the resolution equals week. If the time interval is unspecified, the query time is for 52 weeks.
  Default: 52
- **defaultQueryTimeoutSecs**
  Defines the standard timeout for all queries.

Default: 30

- **maxQueryTimeoutSecs**
  Defines the maximum timeout, which can be specified in the URL with the `timeout` parameter.
  Default: 120

4. Save the changes.
   The new values apply when you load or reload QueryBuilder in the browser.

OpenAPI Apps

OpenAPI apps use the flexibility of OpenAPI queries to deliver and present data in a highly customizable way. OpenAPI apps allow custom content to be served to OpenAPI app views on dashboards and context pages. You can build your own apps or you can select from various available apps. You can then deploy and display these apps in NetOps Portal.

The following video demonstrates how to download, deploy, and add OpenAPI apps to NetOps Portal:

OpenAPI App Development

OpenAPI apps are made up of HTML, JavaScript, CSS, and metadata contained in a zip file. Generally, apps use NetOps Portal web services and OpenAPI.

The following process guides you through the OpenAPI app development and testing process:

1. If one exists, find a similar OpenAPI app to use as the basis for your new app.
2. Set up a local folder to house the files for the app. Use a single directory that contains all the required directories and files for the app.
3. Create the appConfig.properties file. This properties file is required for deployment in NetOps Portal and for the App View. See the following documentation for information about the contents of this file.

   NOTE

   The appConfig.properties file must reside in the top-level single directory referenced in the previous step.
4. Use OpenAPI QueryBuilder to create sample odata queries for your app. Save the output from these queries as to a static file. Start developing your code with the static data.
   After development, static data is useful for debugging to ensure that your parsing code works correctly.
5. Build a demo page. Ensure that the page can load the static data. Also, ensure it can save the data to a variable attached to the window object. To enable browser debugging tools, add the debugger statement in the JavaScript code.
   The demo page lets you use a browser debugging tool. Use the tool to look at the data that came over, and verify that part of the app worked.
6. Start parsing parameters.
   Include the URL you used to get the original data, and use a parameter to switch into debug mode. Use parameters from the URL to the build OpenAPI query. Use the console.log function to log any URL to the console, such as console.log("url: " + url).
7. Pick the JavaScript libraries that you want to use for your app.
   Look at existing apps for examples and copy the libraries into your app folder. Remove unnecessary files.
8. Start writing the app.
   - If the data includes multiple items, start with one item first.
   - At this stage, do not focus on the app appearance. However, *do* include CSS classes in the resulting HTML to make it easier.
   - Put all your labels in a single place in your code, in case you localize the app.
9. When you have something working, deploy the app to your NetOps Portal test system. Verify that it works on a remote server. If you refactor your code to get it to work remotely, there is less code to refactor at this stage.

10. Zip the app folder, including the properties file and deploy the app through the UI.
11. To test the app in the UI, create a dashboard, and add two App Views to the dashboard.
    Edit one of the App Views, and select your app from the drop-down list. If the app does not appear in the drop-down list, something in the appConfig.properties file is incorrect.
    Once the app appears in the first view, edit the other App View and add the same app. This test ensures that the app can support multiple instances on a single page.
12. Continue to work locally, and periodically redeploy the app to the test system.
    If you update the URL in the metadata, edit the App View. Reselect the app to get the updated URL.
13. Use a more restricted user account on Performance Create to verify that the app still works correctly.
    If your app is context or time range aware, change the associated controls and verify that your app responds correctly.
14. If appropriate, use the redirector to add context page links to your app. The redirector makes the app feel like a native part of the system.
15. When the app is complete, consider contributing your app to the GitHub repository for others.
    Before you add the app to the repository, remove any personal information.

appConfig.properties

Use the following format for the appConfig.properties file:

```
   appName=App Name
description=Optional app description
url=Required_URL_parameters
height=height in pixels  supportedContext=context_code
```

- **appName**A unique name for the app
- (Optional) **description**
  A description that appears in the App View when you select the app
- **url**
  The app URL and URL parameters
  For information about URL parameters in NetOps Portal, see Browser Views.
- (Optional) **height** The height of the app view in pixels **Default:** 250
- (Optional) **supportedContext**
  The context where the app appears in the app view. You can use this parameter to restrict the available apps to specific contexts. For example, if you specify only value **i**, the app is available only in app views on interface context pages and the app is unavailable at the dashboard level.
  Values:
  - **d**
    Device
  - **i**
    Interface
  - **s**
    Server
  - **r**
    Router
  - **g**
    Group
  - **nc**
    None, appears in all contexts

  Default: nc
  To designate multiple contexts, separate the values with a comma.

Example:

```
  appName=Percentile Trend App
description=This is a Percentile Trend App
url=index.html?
id={ItemIdDA}&startTime={TimeStartUTC}&endTime={TimeEndUTC}&metric=im_UtilizationIn
height=750 supportedContext=i,d
```

App Development Best Practices

Use the following best practices for developing OpenAPI apps:

- Do not use externally sourced JavaScript libraries. If you need libraries for your app, include the library in the app folder. Also include any files that describe the license terms of the library.
- Do no use any NetOps Portal JavaScript, CSS, or images. When these entities are updated, that app could break.
- Test the app to verify that it works at scale and with different configurations. Many apps depend on system resources, such as web services. Verify that the app is not generating unnecessary load.
- Build test modes into the app to help debug problems. Add an optional parameter that switches the app from using a web service API to a canned set of data included in the app.
- If you distribute the app to other users, verify that no private information is included. Sanitize the static data files that are included for debugging.
- Use relative paths and never include the full URL. If the system is behind a firewall, or changes DNS names, using full URLs breaks the app. Using relative paths helps the transition between working locally and then deploying the app.
- Verify that the app works in an iframe. Both the Browser View and App View use iframes to isolate the apps from the rest of the page.
- The OpenAPI uses Data Aggregator item IDs. NetOps Portal IDs are not recognized. Use the Data Sources web services to convert between the Data Aggregator item ID and the NetOps Portal ID. For more information, see Data Sources Web Service.
- To access OpenAPI data, the request must come from an app on the NetOps Portal host. The app deployment places the app folder on the NetOps Portal host.
- For direct oData queries in the apps, use the relative path for the NetOps Portal system to use the OpenAPI proxy: `/pc/odata/api/...`

Download an App

Various apps are available on GitHub. These applications are supported through GitHub open-source collaboration:

```
Copyright (c) 2018 CA Technologies
The MIT License
Permission is hereby granted, free of charge, to any person obtaining a copy of this
software and associated documentation files (the "Software"), to deal in the Software
without restriction, including without limitation the rights to use, copy, modify,
merge, publish, distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to the following
conditions:
The above copyright notice and this permission notice shall be included in all copies
or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
```

```
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
```

Follow the steps:

1. Go to https://github.com/CA-PM.
   A list of available apps with descriptions appear.
2. Click the app that you want.
   The contents of the app, a sample visualization, and readme instructions appear.
3. Click **Clone or download** and **Download ZIP**.
4. Save the files to your preferred directory.

Deploy an App

To deploy OpenAPI apps, load the app through NetOps Portal. You can deploy apps while the system is running. App deployment does not require a restart. For successful deployment, the app must meet the following requirements:

- The app files are contained in a single folder.
- The folder includes the appConfig.properties file.
- The app folder is zipped in a ZIP file.
- The maximum ZIP filesize is 100 MB.

This procedure requires the Administrator role.

Follow the steps:

1. Hover over **Administration**, and click **Configuration Settings: App Deployment**.
2. Click **Browse**, and select a zipped app.
3. Click **Add**.
   The app is copied to the user app directory. Use the App Viewer to show the app in NetOps Portal.

Display an App in NetOps Portal

Use the app view in NetOps Portal to display an app.

> **NOTE**
>
> PDF and CSV printing and emailing options are unsupported for app views.

Follow the steps:

1. Add or edit a dashboard or page for the intended item context.
2. Add an app view for the app.
3. Select the app for the view.
4. Update the parameters and height as desired.
5. Click **Save**.

Audit OpenAPI Usage

To track the performance of the OpenAPI QueryBuilder, see the usage statistics. To view the usage statistics of the OpenAPI QueryBuilder, review the OpenAPI log file or create an OpenAPI query. Auditing OpenAPI usage helps to diagnose system performance issues.

OpenAPI Usage Log File

The log file provides detailed information about each query that is executed, and whether the query succeeded or failed. Each entry for a single query in the log file documents the start and end of the request. The log file is located in the following directory:
`/opt/IMDataAggregator/apache-karaf-<vers>/data/log/odata-services.impl.log`

The following query details are documented in the log file:

- Host that made the API call
- Tenant
- User
- URL
- Length of time of the request

Example

The following entry is an example of a successful query request:

```
INFO  | qtp1695827227-67 | 2015-06-01 12:17:55,207 | ODataRequestProcessor |
 odata.impl.ODataRequestProcessor  910 || Start request (Thread id 67): host:
 '10.42.200.100' tenant: 'Default Tenant' user: 'admin' url: 'http://da.ca.com:8581/
odata/api/components?$expand=device&$select=ID,Name,device/ID,device/Name&$format=json&
$top=20000' - Run budget 30000 ms (30 seconds).

INFO  | qtp1695827227-67 | 2015-06-01 12:17:55,482 | ODataRequestProcessor |
 odata.impl.ODataRequestProcessor  948 || Done request (Thread id 67): host:
 '10.42.200.100' tenant: 'Default Tenant' user: 'admin' url: 'http://da.ca.com:8581/
odata/api/components?$expand=device&$select=ID,Name,device/ID,device/Name&$format=json&
$top=20000' - Execution time 273 ms.
```

The following entry is an example of a failed query request:

```
INFO  | p1695827227-1474 | 2015-06-01 12:33:30,892 | ODataRequestProcessor |
 odata.impl.ODataRequestProcessor  910 || Start request (Thread id 1474): host:
 '10.42.200.100' tenant: 'Default Tenant' user: 'admin' url: 'http://da.ca.com:8581/
odata/api/components?$expand=device&$select=ID,Name,device/ID,device/Name&$format=json&
$top=500000' - Run budget 30000 ms (30 seconds).

ERROR | p1695827227-1474 | 2015-06-01 12:33:30,894 | ODataRequestProcessor |
 odata.impl.ODataRequestProcessor 1047 || Failed to complete request (Thread id
 1474): host: '10.42.200.100' tenant: 'Default Tenant' user: 'admin' url: 'http://
da.ca.com:8581/odata/api/components?$expand=device&$select=ID,Name,device/ID,device/
Name&$format=json&$top=500000' - Execution time 0 ms. Error: The top value must be in
 the range (0 .. 20000).
```

Track OpenAPI Usage

Run an OpenAPI query to track the overall OpenAPI usage. This query can show you the number of queries during a specific time period or the average processing time. The following statistics are available:

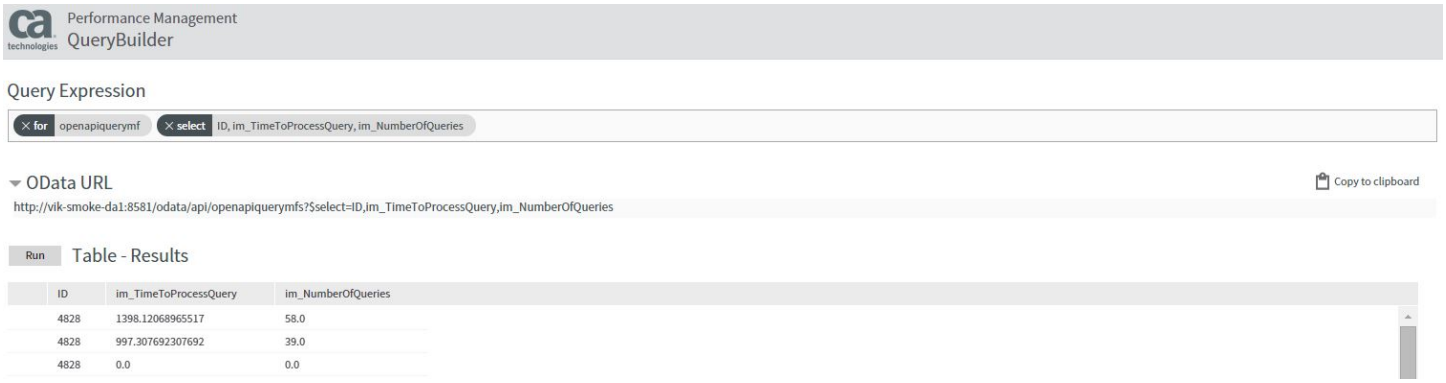
- **Number of Queries**
  The number of queries that occurred since the last read
- **Number of Successful Queries**
  The number of successful queries that occurred since the last read
- **Number of Failed Queries**
  The number of failed queries that occurred since the last read
- **Time to Process Queries**
  The average time to process queries

  NOTE

  This data is collected every 5 minutes.

Follow these steps:

1. Select **Click to start query** in the Query Expression field.
2. Select **metric family, openapiquerymf** from the Query Expression field.
3. Select **select** from the Query expression field.
4. Select the statistics that you want to view. For example, select the following metrics:
   – ID
   – im_TimeToProcessQuery
   – im_NumberOfQueries
5. Click **Run**.
   The OpenAPI usage statistics appear, as shown in the following example:



Product Accessibility Features

CA Technologies is committed to making sure that all customers, regardless of ability, can successfully use its products and supporting documentation to accomplish vital business tasks. This section outlines the accessibility features of DX NetOps Performance Management.

NOTE
If you use a screen reader, we recommend using a browser other than Internet Explorer to access our product. For more information about known screen reader limitations, see Known Limitations.

Product Enhancements

DX NetOps Performance Management offers accessibility enhancements in the following areas:

- Display
- Sound
- Keyboard
- Mouse

> **NOTE**
>
> The following information applies to Windows-based and Macintosh-based applications. Java applications run on many host operating systems, some of which already have assistive technologies available to them. For these existing assistive technologies to provide access to programs written in JPL, they need a bridge between themselves in their native environments and the Java Accessibility support that is available from within the Java virtual machine (or Java VM). This bridge has one end in the Java VM and the other on the native platform, so it will be slightly different for each platform it bridges to. Sun is currently developing both the JPL and the Win32 sides of this bridge.

Display

To increase visibility on your computer display, you can adjust the following options:

- **Font style, color, and size of items**

Defines font color, size, and other visual combinations.

- **Screen resolution**

Defines the pixel count to enlarge objects on the screen.

- **Cursor width and blink rate**

Defines the cursor width or blink rate, which makes the cursor easier to find or minimize its blinking.

- **Icon size**

Defines the size of icons. You can make icons larger for visibility or smaller for increased screen space.

- **High contrast schemes**

Defines color combinations. You can select colors that are easier to see.

Sound

Use sound as a visual alternative or to make computer sounds easier to hear or distinguish by adjusting the following options:

- **Volume**

Sets the computer sound up or down.

- **Text-to-Speech**

Sets the computer's hear command options and text read aloud.

- **Warnings**

Defines visual warnings.

- **Notices**

Defines the aural or visual cues when accessibility features are turned on or off.

- **Schemes**

Associates computer sounds with specific system events.

- **Captions**

Displays captions for speech and sounds.

Keyboard

You can make the following keyboard adjustments:

- **Repeat Rate**

Defines how quickly a character repeats when a key is struck.

- **Tones**

Defines tones when pressing certain keys.

- **Sticky Keys**

Defines the modifier key, such as Shift, Ctrl, Alt, or the Windows Logo key, for shortcut key combinations. Sticky keys remain active until another key is pressed.

Mouse

You can use the following options to make your mouse faster and easier to use:

- **Click Speed**

Defines how fast to click the mouse button to make a selection.

- **Click Lock**

Sets the mouse to highlight or drag without holding down the mouse button.

- **Reverse Action**

Sets the reverse function controlled by the left and right mouse keys.

- **Blink Rate**

Defines how fast the cursor blinks or if it blinks at all.

- **Pointer Options**

Let you do the following:

- – Hide the pointer while typing
  – Show the location of the pointer
  – Set the speed that the pointer moves on the screen
  – Choose the pointer's size and color for increased visibility
  – Move the pointer to a default location in a dialog box

Keyboard Shortcuts

The following table lists the keyboard shortcuts that *DX NetOps Performance Management* supports:

Keyboard	Description
Ctrl+X	Cut
Ctrl+C	Copy

Ctrl+K	Find Next
Ctrl+F	Find and Replace
Ctrl+V	Paste
Ctrl+S	Save
Ctrl+Shift+S	Save All
Ctrl+D	Delete Line
Ctrl+Right	Next Word
Ctrl+Down	Scroll Line Down
End	Line End

Keyboard Navigation

Keyboard	Description
Tab	• Advance the focus to the next field or control in the page.
Shift+Tab	• Move the focus to the previous field or control in the page.
Enter	• Activate the control that has focus (for example, click a button).
Shift+Enter	• Make a selection in a menu or tree.
Space	• Change selection of a checkbox that has focus.
Down Arrow	• Move the focus into a menu or into a tree. • Move the focus to the next item in a menu, tree, radio-button group, or chart.
Up Arrow	• Move the focus to the previous item in a menu, tree, or radio-button group, or chart.
Left Arrow	• Open a closed item in a menu or tree. • Move to the next element of a radio button group and select it.
Right Arrow	• Close an open item in a menu or tree. • Move to the previous element of a radio button group and select it.
Esc	• Close an open menu or dialog box.
Ctrl+Left Arrow	• When the focus is on a grid column header, make the column narrower.
Ctrl+Right Arrow	• When the focus is on a grid column header, make the column wider.
Ctrl+Up Arrow	• When the focus is on a grid column header, move the column to the left.
Ctrl+Down Arrow	• When the focus is on a grid column header, move the column to the right.

Product References and Abbreviations

List only CA product names with a shortened version (such as "CA Service Operations Insight (CA SOI)") and abbreviations/acronyms (such as "virtual machine (VM)") that you use in your wiki space. Sort the list alphabetically. Place this page at the bottom of your TOC so that it appears just above the Announcements & News link.

This documentation references the following products and abbreviations:

- CA Application Delivery Analysis (CA ADA)
- CA eHealth®
- CA Mediation Manager (CAMM)
- CA Network Flow Analysis
- DX NetOps Performance Management (CA PM)
- CA Single Sign-On
- CA Spectrum®
- CA Unified Communications Monitor (CA UCM)
- CA Virtual Network Assurance (CA VNA)

Documentation Legal Notice

Modern Network Monitoring

DX NetOps Virtual Network Assurance enables existing infrastructure management solutions to monitor software-defined networking (SDN) and network functions virtualization (NFV).

This section contains everything you need for modern network monitoring from getting started to troubleshooting information.

Getting Started

DX NetOps Virtual Network Assurance enables existing infrastructure management solutions to monitor software-defined networking (SDN) and network functions virtualization (NFV).

Key Features

DX NetOps Virtual Network Assurance reduces the challenge and risk of SDN/NFV deployments by providing extended visibility and sustained reliability for self-service, automated networks. DX NetOps Virtual Network Assurance provides the following functionality as part of the CA performance monitoring solution:

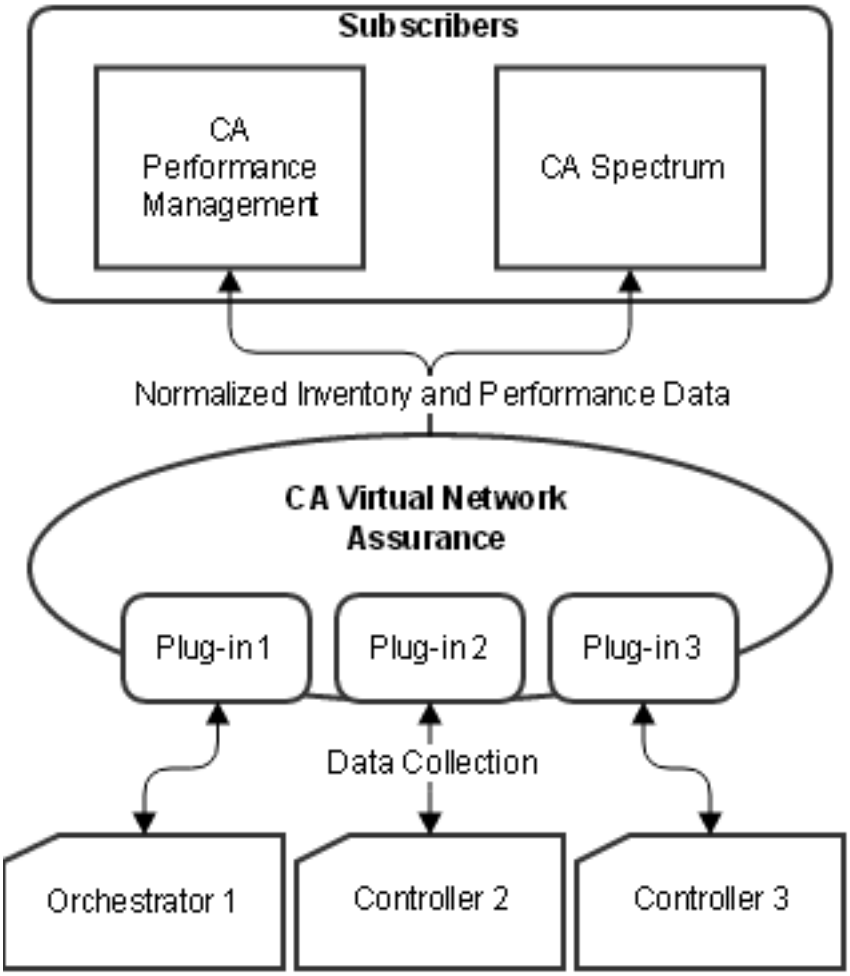
- **SDN/NFV Stack Correlation**
  Shows and monitors all the physical and virtual layers of the stack that supports virtual network functions (VNFs).
- **Service Chain Visibility**
  Provides insight into the building blocks that form service chains.
- **Multi-vendor Protocol and Platform Coverage**
  Supports for SDN controllers, orchestrators, OpenStack, and vendor agents including on-demand certification.
- **Integrated Metrics**
  Enables SDN/NFV monitoring and analytics with traditional SNMP monitoring and analytics.

Architecture

DX NetOps Virtual Network Assurance collects data from SDN/NFV controllers and orchestrators and provides that data to subscribers. Each orchestrator or controller requires a plug-in to configure the connection. DX NetOps Virtual Network Assurance normalizes the collected data and sends updates to subscribers that consume the data.

The following diagram shows this relationship:

Figure 70: Data Collection with CA Virtual Network Assurance



SDN/NFV Assurance in the Application Economy

The following video explains the necessity of SDN/NFV in the application economy:

Release Notes

This release includes the following new features and enhancements:

For the complete CA Performance Management and CA Virtual Network Assurance fixed issues list, see the CA Performance Management documentation.

New Features in the Current Release

New Plug-ins

The following plug-ins are newly supported:

- 128T SD-WAN
- Broadcom BroadView
- Broadcom Mirror on Drop (20.2.3 and Higher Only)
- Silver Peak

Poll Time Configuration

The default timeout value for a poll in the engine of any plug-in is 30 minutes. If a poll requires more than the default value, you can configure the `sdn_poll_timeout_minutes` parameter in the `/etc/VNA.cfg` file. The unit for the parameter value is minutes. After you update this parameter, you must restartDX NetOps Virtual Network Assurance.

Enhancements in Current Release

Cisco Meraki Plug-In (20.2.2 and Higher Only)

You can now configure whether to enable REST API or SYSLOG events along with a notification poll rate and delta time.

The following performance metrics are now available for Access Points:

- Authentication Failures
- Association Failures
- Successful Authentications
- Successful Associations
- Successful DHCP
- Successful DNS
- Successful Connections

For more information, see Cisco Meraki.

Secure Authentication for the VNA Console (20.2.4 or higher only)

The VNA console is now secured with authentication and requires credentials to log in. You can set credentials during a fresh DX NetOps Virtual Network Assurance installation or upgrade. The default user name is "admin".

VMware vSphere Plug-In (20.2.2 and Higher Only)

You can now configure whether to enable performance metrics for DVS ports.

For more information, see VMware vSphere.

Cisco ACI Plug-In

You can now configure either a whitelist or blacklist filter, but not both at the same time. Use the new `filterType` configuration option to configure blacklist or whitelist filters. If this option is unspecified, the filter defaults to a whitelist filter.

You can now configure whether fault or event subscription is enabled.

The Cisco ACI plug-in now includes a new configuration parameter for historical events. You can also configure faults and filters.

You can now configure an `INVENTORY_ENABLE_ENDPOINTS` parameter. If the property is set to No or false, DX NetOps Virtual Network Assurance does not poll for endpoint inventory. If the property is set to Yes or true, DX NetOps Virtual Network Assurance polls for endpoints inventory.

During the upgrade from DX NetOps Virtual Network Assurance 3.7.x to 20.2, the installer prompts the new configuration parameter with the default value (Yes).

For more information, see Cisco ACI.

Cisco Meraki Plug-In

You can now specify whether to enable SSID or policy polling. For more information, see Cisco Meraki.

Nuage Plug-In

You can now specify a notification poll rate, notification delta time, and a time zone. For more information, see Nuage.

Versa SD-WAN Plug-in

You can now configure the number of threads required to run the Inventory poll, the Performance and Availability poll, and the Notification poll.

Tunnels between branches and between branches and controllers are now supported.

You can now specify the out of band management IP address for appliances in multiple ways.

If the default out of band management IP address does not work for you, and if the appliance IP addresses are also the out of band management IP address for the appliances in your environment, then setting the `SET_APPLIANCE_IP_AS_OOBM_IP` flag to "true" forces the Versa plug-in to set the appliance IP address as the out of band management IP address for the discovered appliances.

Otherwise, use the `OOBM_INTERFACE_NAME` parameter to specify a name for the interface that is used as the management interface for your appliances (for example, "eth-0/0").

You can now configure whether to use the `ip-address` field value of the Versa device as the out of band management IP address of the device. For more information, see Versa SD-WAN.

Viptela Plug-in

The Viptela plug-in now collects the following metrics for tunnels:

- Time in Up State
- Time in Down State
- Time in Unknown State
- Pct Time in Up State
- Pct Time in Down State
- Pct Time in Unknown State
- Bytes In (Gauge)
- Bytes Out (Gauge
- Packets In (Gauge)
- Packets Out (Gauge)

This release supports Viptela 19.x.

For more information, see Viptela.

VMware vSphere

You can now specify the number of threads required to run the performance poll. For more information, see VMware vSphere.

DX NetOps Virtual Network Assurance Aggregation Server

Full synchronization fails when partial updates are high in volume. As a result, alarms are missing in DX NetOps Spectrum. This release introduces enhancements to the DX NetOps Virtual Network Assurance Aggregation server. You can now use a new configuration option to specify the queue size. The new configuration parameter is `sdn_client_queue_size` in `/etc/VNA.cfg`.

Example:

```
ca.root=/opt/CA
vna_db_name=vna_lvnqa004994
container_install=0
sdn_client_queue_size=1000
mysql_host=localhost
mysql_port=3306
```

Third-Party Software Acknowledgements

All third-party software has been used in accordance with the terms and conditions for use, reproduction, and distribution as defined by the applicable license agreements. The following file contains the license agreements: TPSAs

The file contains the following license agreements:

- aws-java-sdk 1.11.429
- bc-fips 1.0.1
- Camel 2.16.1
- cdi-api.2.0
- cglib.3.2.12
- cglib-nodep 3.2.12
- commons-beanutils 1.9.3
- Commons Cli 1.4
- Commons Collections 4.4.0
- Commons Logging 1.2
- Commons net 3.6
- commons-compress 1.18
- Commons IO 2.6
- Commons Lang 2.6
- Commons Lang 3.3.9
- dom4j 1.6.1
- Ganymed SSH-2 Build 262
- gpars 1.2.1
- Groovy 2.5.7
- Guava 28.0-jre
- Hibernate Validator 6.0.17.Final
- hibernate_core 5.3.10.Final
- HttpComponent 4.5.9
- jackson-databind 2.9.9.1
- javaee-pi.8.0.1
- Jaxen 1.1 B8
- Jaxrs-api 3.0.12
- JBoss RESTEasy 3.8.0 Final
- jgrapht 1.3.1
- Joda-time 2.10.3
- json-lib 2.4
- json-path 2.4.0
- kryo 4.0.2
- kryo-seriallizers 0.45
- liquibase 3.8.0
- mvel 2.4.0 Final
- MySQL 8.0.12
- ovsdb 1.2.1
- Saxon-B 9.1.0.8
- ServingXML 1.1.2
- slf4j 1.7.26
- swagger-core 1.5.22
- Syslog4j 0.9.46
- trimou 2.5.0 Final
- TrueZIP 6.6
- Tyrus 1.15
- VMware VI (vSphere) Java API 5.5
- WildFly 17.0.1.
- xercesImpl 2.6.2
- xml-apis 1.0.b.2
- xnio-nio 3.6.5 Final
- yamlbeans 1.13

Plug-in Compatibility

DX NetOps Virtual Network Assurance collects performance data for virtual networks. To enable data collection, configure the plug-in for each technology in your environment

The following table contains the supported plug-ins:

Table 2: Supported Plug-ins

Technology	Versions
128T SD-WAN	• 128T SD-WAN 4.1.1 SNMP support is not available in 4.1.1.
Amazon Web Services (AWS)	• Cloud
Broadcom BroadView	• SONiC BUZZNIK
Cisco ACI	• Cisco ACI 4.2.4 • Cisco ACI 4.0 • Cisco ACI 3.2 • Cisco ACI 2.0
Cisco Meraki	• Cloud
Nuage	• Nuage 4.0.0 R3
OpenContrail	• 2.20 • 2.21
OpenDaylight	• Nitrogen
OpenStack	• Juno • Kilo • Liberty • Mitaka • Newton • Ocata • Pike
Open vSwitch	• 2.8.2
Silver Peak	• Unity Orchestrator 8.6.1.40151
Versa SD_WAN	• Versa SD-WAN 16.1R2 **Note:** Only the Versa SD-WAN 16.1R2 version supports out-of-band IP management for the branches that this plugin discovers. Without this IP management, DX NetOps Spectrum cannot automatically poll the branches using SNMP. • Versa SD-WAN 16.1R1
Viptela	• Viptela 20.1.1 • Viptela 19.x. • Viptela 18.4 • Viptela 18.3 • Viptela 17.2.0 • Viptela 17.1.0
VMware vSphere	• VMware vSphere 6.0. • VMware vSphere 6.5.

Installing

Ensure someone with the appropriate level of familiarity with your network deploys and configures DX NetOps Virtual Network Assurance.

Contents:

DX NetOps Virtual Network Assurance Compatibility

DX NetOps Virtual Network Assurance is compatible with the following product versions:

- DX NetOps Performance Management 20.2
  For DX NetOps Performance Management installation information, see the DX NetOps Performance Management documentation.
- DX NetOps Spectrum 10.4.2
  For DX NetOps Spectrum installation information, see the DX NetOps Spectrum documentation.
- Network Flow Analysis 10.0.4
  For Network Flow Analysis installation information, see Network Flow Analysis documentation.

For more information about supported releases, see the DX NetOps Interoperability.

Verify the Requirements of Your Technologies

Verify the requirements of each technology in your virtual network environment:

> **TIP**
>
> We recommend, but do not require, a single DX NetOps Virtual Network Assurance (CA VNA) per data center or technology grouping in your virtual network environment. For example, you can have OpenStack, Open vSwitch, and Cisco ACI on the same instance of VNA.
>
> You can have multiple plug-ins of the same technology type on a single instance of CA VNA.

- 128T SD-WAN
- Amazon Web Services (AWS)
- Broadcom BroadView
- Cisco ACI
- Cisco Meraki
- Nuage
- OpenContrail
- OpenDaylight
- OpenStack
- Open vSwitch
- Poll Rate Configuration
- Silver Peak
- Versa SD-WAN
- Viptela
- VMware vSphere

Ensure that you install the product as the root user.

Prepare to Install DX NetOps Virtual Network Assurance

Before you install DX NetOps Virtual Network Assurance, verify that your system meets the requirements.

Follow these steps:

1. Verify that the DX NetOps Virtual Network Assurance host meets the system requirements:
   - **Memory:** 12 GB
   - **CPU:** 4vCPU
   - **Disk:** 20 GB
   - **Operating System:** RHEL/CentOS 6.5 through 7.x
   - **Packages:** `libaio`
2. **(Scaled-Environment Only)** Verify that the DX NetOps Virtual Network Assurance host meets the heap size requirements:
   - **Heap Memory:** 16 GB
   - **CPU:** 8vCPU
   - **Disk:** SSD recommended 120 GB
3. Install Oracle or any OpenJDK 64-bit Java 8 as the default JDK.

   NOTE

   DX NetOps Virtual Network Assurance does not run on IBM SDK Java and any JRE. Only JDKs are supported.

   IMPORTANT
   Use the `alternatives` tools to select the default JDK version of Java to use.
   For more information, see the Red Hat documentation.
   Alternatively, create a link in `/usr/bin/java` to the Java 8 JDK Java executable.
4. Verify that the system hostname resolves to an IP address in `/etc/hosts` .
5. Verify that the following ports are open on the DX NetOps Virtual Network Assurance host:
   - Verify any ports that are required for communication to the orchestrators and controllers.
   - Port 8080 is required for communication between DX NetOps Virtual Network Assurance and other CA products.
   - Port 9990 is the Wildfly management port.
6. Wildfly is installed as part of DX NetOps Virtual Network Assurance. Verify that no other versions of Wildfly are installed on the host system. Do not install DX NetOps Virtual Network Assurance on a system with another application that uses Wildfly.

Install DX NetOps Virtual Network Assurance

To set up your system, install DX NetOps Spectrum. The script creates the wildfly user and the mysql user.

Follow these steps:

1. Copy the installation files to the host server.
2. Open a command line and run the following command:
   `./CA_VNA-`*`version`*`-Linux.txe`

   NOTE

   To avoid prompts and use the default values, add the following parameter to the install command: `--no-prompt`
3. Accept the EULA.
4. Follow the instructions in the console.
   The prompts have the following default values:
   - Installation directory = /opt/CA
     You can install CA VNA on a custom directory.
   - Default user domain = Default Domain

For more information about user domains, see Manage Domain Groups.
- Wildfly admin password = admin
- Wildfly JMS password = app
- Configure APM = N
- MySQL server port = 3306
- MySQL root user password = admin

5. (Optional) To configure a CA Application Performance Management (CA APM) agent during installation, provide the CA APM host and port.
   To configure the agent after installation, see Monitor CA Virtual Network Assurance Broker Performance.
   The script installs DX NetOps Virtual Network Assurance.

Install Logs

After successful installation, you can find the installation log `CA_Virtual_Network_Assurance_install_dd_mm_yyyy_hh_mm_ss.log` in the `$CA_VNA_ROOT/VNA/logs` directory.

In case the installation fails or you stop the installation abruptly , the logs are saved to the directory where you have the CA VNA installer. You can find the log with name

```
CA_Virtual_Network_Assurance_install_dd_mm_yyyy_hh_mm_ss.log
```

Configure the Limit on the Number of Open Files

Configure the limit on the number of open files. Verify that the `wildfly` user has a limit of at least 65536 on the number of open files. Set this value permanently.

Follow these steps:

1. Log in as the `wildfly` user on the VNA host.
2. Change the `ulimit` for the open files limit to at least 65536:
   ```
   ulimit -n ulimit_number
   ```
   Example:
   ```
   ulimit -n 65536
   ```
3. Open the following file:
   `/etc/security/limits.conf`
4. Add the following lines:
   ```
   # Added by VNA
   * soft nofile 65536
   # Added by VNA
   * hard nofile 65536
   ```

   NOTE

   Restart VNA for these changes to take effect. If you are upgrading, the upgrade process automatically restarts VNA.
5. Verify that the number of open files is set properly:
   ```
   ulimit -n
   ```
   The command returns the limit that you specified earlier.

Configure the Plug-ins

CA Virtual Network Assurance collects performance data for virtual networks. To enable data collection, configure the plug-in for each technology in your environment. For more information, see Building.

Uninstall

To remove DX NetOps Virtual Network Assurance from your system, run the uninstaller.

Follow these steps:

1. Log in to the DX NetOps Virtual Network Assurance host as 'root', or use the 'sudo' account that you configured for the installation.
2. Uninstall DX NetOps Virtual Network Assurance using the following command:

```
/opt/CA/VNA/tools/bin/uninstall_vna.sh
```

3. Confirm the uninstall.

Upgrading

If you have DX NetOps Performance Management, to upgrade DX NetOps Virtual Network Assurance, use the following upgrade path. If you have DX NetOps Spectrum, see the documentation.

> **NOTE**
> If you have the Cisco ACI plug-in configured, when you upgrade to 3.7.8 and higher, all the previously set values for fault-codes, fault-types, and fault-severities are cleared. All the faults are whitelisted and saved in the database. To prevent this, stop the Wildfly service within 5 minutes of the upgrade and configure the filters as needed.

Verify the Prerequisites

Verify the following prerequisites before you upgrade DX NetOps Virtual Network Assurance:

- Verify whether the numactl package is installed on the DX NetOps Virtual Network Assurance host:

  ```
  rpm -qa | grep ^numactl
  ```

  If the required package is missing, install the package:

  > **NOTE**
  >
  > If you are not the root user, use the sudo prefix.

  ```
  yum -y install numactl
  ```

- Before you upgrade, back up the database. For more information, see Back Up and Restore CA Virtual Network Assurance.

Review Plug-in Configurations

During the DX NetOps Virtual Network Assurance upgrade, your plug-in inventory and configurations are maintained. You can query the following REST URL to review your plug-in configurations. For more information, see Building.

Follow these steps:

1. Query the following REST URL to find the engine ID:
   `http://VNA_host:8080/vna/rest/v1/admin/engines`
2. Query the following REST URL for your plug-in configuration:
   `http://VNA_host:8080/vna/rest/v1/admin/engines/Engine_ID/config`
3. Review the response.
   Example:

   ```
   {
   ```

```
    "AVAILABILITY_DELTA_TIME": "300",
    "AVAILABILITY_POLL_RATE": "0 */5 *",
    "DOMAIN_ID": "1",
    "INVENTORY_DELTA_TIME": "600",
    "INVENTORY_POLL_RATE": "0 */10 *",
    "PERFORMANCE_DELTA_TIME": "300",
    "PERFORMANCE_POLL_RATE": "0 */5 *",
    "PROTOCOL": "https",
    "VCENTER_IP": "10.241.16.37",
    "VCENTER_PASSWORD": "****",
    "VCENTER_USER_NAME": "vnaUser@vsphere.local"
}
```

NOTE

Passwords in the response is always encrypted.

Upgrade DX NetOps Virtual Network Assurance to a Custom Directory

From CA VNA 3.6.6, you can move existing CA VNA instance to a custom directory.

NOTE

Do not delete CA VNA server from Spectrum OneClick admin page or CA PC.

Migrate CA VNA and MySQL Data

When you upgrade to current version of CA VNA, you back up the CA VNA and MySQL data. You can import them post upgrade to the new installation location.

Follow these steps:

1. On the CA VNA server, copy the latest CA VNA 3.6.6 and higher version.
2. Stop WildFly using one of the following commands:

   ```
   service wildfly stop or systemctl stop wildfly
   ```

   Back up the Current CA VNA
3. To back up your current CA VNA instance, execute the following commands:

   ```
   mkdir -p /<backup_directory>/db
   mkdir -p /<backup_directory>/collector
   mkdir -p /<backup_directory>/data

   /opt/CA/VNA/tools/bin/db_backup.sh /<backup_directory>/db/vna_backup
   cp -r /opt/CA/VNA/data/* /<backup_directory>/data/
   cp -r /opt/CA/VNA/collector/* /<backup_directory>/collector/
   ```
4. Uninstall CA VNA using the `/opt/CA/VNA/tools/bin/uninstall_vna.sh` command.

Install CA VNA in new Location and Restore the Data

This section describes the steps to install the CA VNA on a custom directory and restore the old data from the `/opt/CA/VNA` .

Follow these steps:

1. Install CA VNA 3.6.6 or higher from the copied installer.

2. Enter a custom installation path.
   Follow the installation wizard and complete the installation.
3. After the installation, note the owner of `<new_vna_directory>/data` and `<new_vna_directory>/collector` directories.
   By default `wildfly` is the owner.
4. Note the owner & group of `<new_vna_directory>/data/ID_CACHE>dat` file.
   By default `wildfly:wildfly` is the owner and group name.
5. (Optional) For an Aggregator VNA, note the owner and group of `<new_vna_directory>/REMOTE_ID_CACHE.dat` file.
   By default `wildfly:wildfly` is the owner and group name.
6. Restore the database using the `/<new_install_directory>/VNA/tools/bin/db_restore.sh /<backup_directory>/db/vna_backup` command.
7. Stop the existing CA VNA instance using the, `service wildfly stop` or
   ```
   systemctl stop wildfly
   ```
   command.
8. Restore the files using the following commands:
   ```
   cp -r /<backup_directory>/data/* /root/CA/VNA/data/
   cp -r /<backup_directory>/collector/* /root/CA/VNA/collector/
   ```
9. Use
   ```
   chown
   ```
   command to restore the owner and group of restored data files.
10. Start CA VNA using the `service wildfly start` command.
11. Log in to CA VNA Swagger UI: http://*VNA_host*:8080/vna
12. Update the existing plug-in.

Upgrade DX NetOps Virtual Network Assurance on Existing Directory

Follow these steps:

1. In Performance Center, hover over **Administration**, and click **Monitored Items Management: VNA Gateways**.
2. Select the active VNA Gateway, and click **Edit**.
3. Set **Administrative Status** to **Down**, and click **Save**.
4. Upgrade DX NetOps Performance Management.
5. Install the new DX NetOps Virtual Network Assurance version. For more information, see Installing.
   The plug-in inventory and configurations are maintained.
6. Complete prompts for all new plug-in parameters.
7. Wait for a single polling cycle to complete.
8. In Performance Center, for the VNA Gateway, set **Administrative Status** to **Up**.

> **NOTE**
>
> You can find the installation log `CA_Virtual_Network_Assurance_install_dd_mm_yyyy_hh_mm_ss.log` in the `/opt/CA/VNA/logs` directory.

Building

DX NetOps Virtual Network Assurance collects performance data for virtual networks. To enable data collection, configure the plug-in for each technology in your environment.

The following video shows an example of plug-in configuration with the Cisco ACI plug-in:

Configure a Plug-in

For each technology in your environment, configure an instance of the plug-in. Ensure that the technology is set up correctly to enable data collection. For more information, see the relevant page for the specific plug-in.

Follow these steps:

1. Navigate to the DX NetOps Virtual Network Assurance API:
   `http://gateway_host:8080/vna/`
2. Expand **Admin**.
3. Select the following GET call, specify the **pluginName**, and click **Try it out!**
   `/v1/admin/plugins/{pluginName}/template`

   TIP

   To list all installed plug-ins, use the following GET call:

   `/v1/admin/plugins`
4. Copy the content of the template and complete the configuration details. For examples of the configuration JSON template, see the documentation for each plug-in.

   WARNING

   In the configuration JSON template, use the IP address to identify the technology. *Do not* use the hostname.
5. POST the configuration JSON template to the following endpoint:
   `/v1/admin/plugins/{pluginName}`

   NOTE

   To update the configuration an existing plug-in, PUT the updates to the following endpoint:

   `/v1/admin/plugins/{pluginName}`
6. Specify the **pluginName**, and write a meaningful description in **configDesc**. After you post the configuration JSON template, you cannot view the configuration details. The description is required to identify the configuration.
7. Update the required details in the JSON template, and click **Try it out!**
   DX NetOps Virtual Network Assurance uses the configuration details to connect to the technology and begin collecting performance and inventory data.

Verify Plug-in Status

To verify the status of a plug-in, change the engine status.

Follow these steps:

1. Get the **configId** value of the plug-in. Select the following GET call, specify the **pluginName**, and click **Try it out!**
   `/v1/admin/plugins/{pluginName}/configs`
2. From the response body, copy the **configId** value for the plug-in. To identify the plug-in, use the **configDesc**.
3. Select the following GET call, specify the **configId** value as the **engineid**, and click **Try it out!**
   `/v1/admin/engines/{engineid}`

   If the plug-in is operating correctly, the response body is **RUNNING**.

Upload a New Plug-in

For plug-ins that are delivered outside the release, upload the plug-in to the gateway before you configure the plug-in. The procedure requires CURL.

Follow these steps:

1. Download the new plug-in from CA Technologies and copy the plug-in to a system that has access to the gateway.
2. Upload the plug-in to the gateway:

   ```
   curl -X POST http://gateway_host:8080/vna/rest/v1/admin/plugins -H "Context-Type:
    application/octet-stream" -T new-plugin.jar
   ```

   – **new_plugin** is the name of the plug-in JAR file.

   The plug-in is uploaded to the gateway. To collect data, configure the plug-in.

   NOTE

   To revert a plug-in to an earlier version, add the following parameter to the URL:

   ```
   ?forceDowngrade=true
   ```

128T SD-WAN

Inventory and performance metrics from 128T SD-WAN support SD-WAN monitoring.

The plug-in collects inventory for the following items:

- Sites

  NOTE
  Sites from 128T SD-WAN map to the Site Groups in Performance Center. If desired, you can manage your site groups in Performance Center to update a site name.

- Routers

  NOTE
  Router inventory items are supported only for DX NetOps Performance Management
- Interfaces (Device Interfaces/Network Interfaces associated to Router)
- Tunnels (Connectivity between two Network Interfaces associated to Routers)
- Application/SLA Paths (SLA Class/Profile associated with Tunnels)
- Alarms (Related to Router)

  NOTE
  DX NetOps Spectrum consumes these alarms.

The plug-in collects the following performance metrics:

Table 3:

Item Types	VNA Metric Families and Metrics
Devices	• CPU – Utilization • Memory – Memory Utilization
Interfaces	• Interface – Incoming Packets – Outgoing Packets
Tunnels	• Network Interface – Jitter – Latency – Packet loss

Item Types	VNA Metric Families and Metrics
Application/SLA Paths	• Network Interface – Percentage of Jitter SLA Threshold – Percentage of Packet Loss SLA Threshold – Percentage of Latency SLA Threshold

Supported Releases

DX NetOps Virtual Network Assurance supports the following releases of Versa SD-WAN:

- 128T SD-WAN 4.1.1

  NOTE
  SNMP support is not available in 4.1.1.

Requirements

- The Conductor port must be open to the DX NetOps Virtual Network Assurance host.

Plug-in Configuration Example

The following JSON example shows the Versa SD-WAN plug-in configuration:

```
{
   "PLUGIN_CONFIG": {
        "CONDUCTOR_HOST": "54.177.55.140",
        "CONDUCTOR_HOST_PORT": 443,
        "CONDUCTOR_HOST_USER_NAME": "admin",
        "CONDUCTOR_HOST_USER_PASSWORD": "128Tadmin",
        "PROTOCOL": "https",
        "INVENTORY_POLL_RATE": "0 */10 *",
        "INVENTORY_DELTA_TIME": 600,
        "PERFORMANCE_POLL_RATE": "0 */5 *",
        "PERFORMANCE_DELTA_TIME": 300,
        "NOTIFICATION_POLL_RATE": "0 */1 *",
        "NOTIFICATION_DELTA_TIME": 60,
        "DOMAIN_ID": 0
        }
   }
```

- **CONDUCTOR_HOST**
  The IP address of the 128T Conductor
- **CONDUCTOR_HOST_PORT**
  The 128T Conductor port Default: 443
- **CONDUCTOR_HOST_USER_NAME**
  The 128T Conductor user name
- **CONDUCTOR_HOST_USER_PASSWORD**
  The password for the 128T Conductor user
- **PROTOCOL**
  The HTTP security scheme for 128T SD-WAN Default: https
- **INVENTORY_POLL_RATE**
  How often the product collects inventory data
- **INVENTORY_DELTA_TIME**

The time difference between inventory polls (in seconds)
- **PERFORMANCE_POLL_RATE**
  How often the product collects performance data
- **PERFORMANCE_DELTA_TIME**
  Difference between polls (in seconds) for performance data requests
- **NOTIFICATION _POLL_RATE**
  How often the product collects notification data
- **NOTIFICATION _DELTA_TIME**
  Difference between polls (in seconds) for notification data requests
- **DOMAIN_ID**
  CA Virtual Network Assurance assigns inventory from this plug-in to the specified domain.

Amazon Web Services (AWS)

Amazon Web Services (AWS) is a cloud computing platform. The AWS provides a combination of Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and packaged Software as a Service (SaaS) offerings.

The plug-in collects inventory for the following items:

- Regions
- Availability Zones
- Amazon EC2 (VM)
- Virtual Private Cloud (Virtual Network)
- Subnets
- NAT Gateways
- Customer Gateways
- Internet Gateways
- VPN Gateways
- Tunnels

AWS Performance Metrics

The plug-in collects the following performance metrics:

Item Types	VNA Metric Families and Metrics	CA PM Metric Family	Metric Name/ Description
EC2	CPU Utilization	Virtual Machine Aggregate Metrics	CPU Utilization
	Aggregated Network Incoming Bytes	Virtual Machine Aggregate Metrics	Aggregated Bits In Of Interfaces
	Aggregated Network Outgoing Bytes	Virtual Machine Aggregate Metrics	Aggregated Bits Out Of Interfaces
	Aggregated Network Bytes	Virtual Machine Aggregate Metrics	Aggregated Bits Of Interfaces
	Aggregated Network Bytes Per Second	Virtual Machine Aggregate Metrics	Aggregated Bits Per Second Of Interfaces
	Aggregated Network Incoming Bytes Per Second	Virtual Machine Aggregate Metrics	Aggregated Bits In Per Second Of Interfaces

	Aggregated Network Outgoing Bytes Per Second	Virtual Machine Aggregate Metrics	Aggregated Bits Out Per Second Of Interfaces
	Aggregated Network Incoming Packets	Virtual Machine Aggregate Metrics	Aggregated Packets In Of Interfaces
	Aggregated Network Outgoing Packets	Virtual Machine Aggregate Metrics	Aggregated Packets Out Of Interfaces
	Aggregated Network Packets	Virtual Machine Aggregate Metrics	Aggregated Packets Of Interfaces
	Aggregated Network Incoming Packets Per Second	Virtual Machine Aggregate Metrics	Aggregated Packets In Per Second Of Interfaces
	Aggregated Network Outgoing Packets Per Second	Virtual Machine Aggregate Metrics	Aggregated Packets Out Per Second Of Interfaces
	Aggregated Network Packets	Virtual Machine Aggregate Metrics	Aggregated Packets Per Second Of Interfaces
	Disk Capacity	Virtual Disk	Capacity
	Disk IOPS	Virtual Disk	I/O per second
	Disk Read Bytes	Virtual Disk	Bytes Read Per Sec
	Disk Write Bytes	Virtual Disk	Bytes Written Per Sec
Tunnels	Tunnel State	SDN Tunnel	Availability
	Tunnel Data In	SDN Tunnel	Bytes In
	Tunnel Data Out	SDN Tunnel	Bytes Out

The plug-in collects the following threshold notifications:

VPN Connections	• Tunnel State • Tunnel Data In • Tunnel Data Out

The plug-in collects the flow data from VPC logs from AWS Console with the following header:

```
<version> <account-id> <interface-id> <srcaddr> <dstaddr> <srcport> <dstport> <protocol>
 <packets> <bytes> <start> <end> <action> <log-status>
```

The plug-in uses `aws-java-sdk` to pull data, which uses the HTTP protocol.

The following table contains the HTTP counts for each poll type:

NOTE

The following counts may vary depending on the environment. You can get more details for HTTP count from the `server.log` by turning the root logger of WildFly to debug.

Poll Type	HTTP Count
Inventory	35
Performance	39
Flow	17
Notification	17

NOTE

More Information:

For more information about flow log header, refer https://docs.aws.amazon.com/vpc/latest/userguide/flow-logs.html

Plug-in Configuration Example

The following JSON example shows the AWS plug-in configuration:

```
{
"PLUGIN_CONFIG": {
"AWS_ACCESS_KEY": "xxxxxxxxxxx",
"AWS_SECRET_KEY": "JWUSJJSJSUIWKSOW",
"PROTOCOL": "https",
"INVENTORY_POLL_RATE": "0 */10 *",
"INVENTORY_DELTA_TIME": 600,
"NOTIFICATION_POLL_RATE": "0 */5 *",
"NOTIFICATION_DELTA_TIME": 300,
"PERFORMANCE_POLL_RATE": "0 */5 *",
"PERFORMANCE_DELTA_TIME": 300,
"FLOW_POLL_RATE": "0 */10 *",
"FLOW_DELTA_TIME": 600,
"DOMAIN_ID": 0
}
}
```

AWS_ACCESS_KEY
Access keys to sign programmatic requests to AWS
AWS_SECRET_KEY
Secret keys to sign programmatic requests to AWS (Access key and Secret access key that is used together to authenticate requests to AWS)
PROTOCOL
The communication protocol with the AWS
INVENTORY_POLL_RATE
How often the product collects inventory data
INVENTORY_DELTA_TIME
Difference between polls (in seconds)
NOTIFICATION_POLL_RATE
How often the product collects notifications from the AWS
NOTIFICATION_DELTA_TIME
Difference between notification polls (in seconds)
PERFORMANCE_POLL_RATE
How often the product collects performance data
PERFORMANCE_DELTA_TIME
The time difference between performance polls (in seconds)
FLOW_POLL_RATE
How often the product collects flow data
FLOW_DELTA_TIME
The time difference between flow polls (in seconds)
DOMAIN_ID
CA Virtual Network Assurance assigns inventory from this plug-in to the specified domain.

Broadcom BroadView

The BroadView™ Instrumentation Agent unlocks the potential of Broadcom® switch silicon by augmenting CPU functionality to deliver advanced network analytics. The BroadView™ Instrumentation Agent communicates with the underlying Broadcom switch silicon. It collects various telemetry and visibility information, runs algorithms on the data, packages the information appropriately, and provides it to registered clients. Similarly, the agent configures the silicon based on the configuration requests from the client.

With the proliferation of Software-Defined Networking (SDN), multi-tenant networks, and server virtualization—aided by cloud deployments for applications and storage—network complexity is growing exponentially. Troubleshooting such networks has become an increasingly daunting task. Network operators need increased visibility into the network and deeper telemetry data in order to remain in control of the network and to ensure optimal network resource utilization. BroadView™ Instrumentation software provides this critical visibility and telemetry information.

The Buffer Statistics and Tracking (BST) feature of the the BroadView™ Agent, lets you monitor buffers Utilization and detect MicroBurst (Congestion) of Broadcom devices.

For more information, see the following BroadView™ documentation:

- BroadView™ Instrumentation Agent Guide
- SONiC Command Line Interface Guide

The BView BST plug-in collects inventory for the following items:

- Devices (Switches)
- Interface Buffers
- Device Buffers
- Alarms (Trigger Reports) raised on the Agent

The plug-in collects the following performance metrics:

> **NOTE**
> All metrics represent the corresponding buffer utilization. SONiC BUZZNIK supports only a subset of the following metrics.

Table 4:

Item Types	VNA Metric Families and Metrics
Devices (Switches)	• Device – data • Ingress service pool – um-share-buffer-count • Egress service poll. – um-share-buffer-count – mc-share-buffer-count – mc-share-queue-entries • Egress Unicast Queue Group – uc-buffer-count • Egress CPU Queue – cpu-buffer-count
Interfaces	• Ingress port priority group – um-share-buffer-count – um-headroom-buffer-count • Ingress port service pool – um-share-buffer-count • Egress Unicast Queue – uc-buffer-count • Egress Multicast Queue – mc-buffer-count – mc-queue-entries • Egress port service pool – uc-share-buffer-count – mc-share-queue-entries – mc-share-buffer-count – um-share-buffer-count

SONIiC BUZZNIK supports only the following metrics:

NOTE
All metrics represent the corresponding buffer utilization.

Table 5:

Item Types	VNA Metric Families and Metrics
Interfaces	• Ingress port priority group – um-share-buffer-count • Egress Unicast Queue • uc-buffer-count • Egress Multicast Queue • mc-buffer-count
Devices (Switches)	• Egress CPU queue – cpu-buffer-count

Notifications

The BView Agent allows you to configure a threshold for any of the monitored buffers.

For more information, see the `configure-bst-thresholds` API in the BroadView™ Instrumentation Agent Guide.

When any of the configured thresholds is triggered because the buffer usage crosses the configured threshold, the BView BST sends an asynchronous notification (BView Trigger Report) to DX NetOps Virtual Network Assurance.

The BView Agent keeps sending Trigger Reports to DX NetOps Virtual Network Assurance every `trigger-rate-limit-interval` seconds while the metric value remains above the threshold.

For more information, see the `configure-bst-feature` API in the BroadView™ Instrumentation Agent Guide.

Alarms Clearance

The plug-in raises an DX NetOps Virtual Network Assurance alarm when the first Trigger Report for a specific BST entity is received.

The plug-in checks every `ALARM_CLEAR_INTERVAL_SEC` seconds whether the time between *now* and the *last Trigger Report recieved timestamp* for the same BST entity is more than the `ALARM_CLEAR_INTERVAL_SEC` . If the time more than the `ALARM_CLEAR_INTERVAL_SEC` , the DX NetOps Virtual Network Assurance alarm lears.

> **NOTE**
> The BView Agent `trigger-rate-limit-interval` setting should be less than VNA plugin setting ALARM_CLEAR_INTERVAL_SEC. By default this is true

Supported Releases

DX NetOps Virtual Network Assurance supports the following releases of SONiC:

- SONiC BUZZNIK

Requirements

- The network device must support the BView BST feature. Configured the device to a working state.
- The network device must have an SNMP agent installed.
- For consistency, we recommend that each SONiC BView container have the same *trigger-rate-limit-interval* value set by the *configure-bst-feature* API.

Configure the SONiC BView BST Container

Use the BView configuration API to configure the container

Follow these steps:

1. Configure the BView SONiC container to send Periodic and Trigger Reports to the DX NetOps Virtual Network Assurance host (IP and Port).
   a. Log into the managed item with SSH.
   b. Run the following command:
      ```
      sudo su
      ```
   c. Run the following command to set the BST Data Collector IP Adress and Port:
      ```
      config broadview collector IP_Address  Port
      ```
      Example:
      ```
      config broadview collector 10.74.112.112 9070
      ```
2. Configure the following BView BST parameters with the `configure-bst-feature` REST API call:
   a. Set up a REST client with a connection to the managed item.

b. Specify the following URL:
`http://IP_Address:Port/broadview/bst/configure-bst-feature`

c. POST the JSON for defining the BView BST parameters.
Example:

```
{
    "jsonrpc":"2.0",
    "method":"get-bst-feature",
    "asic-id":"0",
    "params": {
    "bst-enable": 1,
    "send-async-reports": 1,
    "collective-interval": 60,
    "stat-units-in-cells": 0,
    "trigger-rate-limit": 1,
    "send-snapshot-on-trigger": 0,
    "trigger-rate-limit-interval": 30,
    "async-full-reports": 1,
    "stats-in-percentage": 1,
    }
     "id": 1,
}
```

3. Configure the following BST thesholds for the required realms or ports with the `configure-bst-thresholds` REST API call:

NOTE
To get a list of ports, priority groups, or queues, run the following command:
`get-bst-threshold`

a. Set up a REST client with a connection to the managed item.

b. Specify the following URL:
`http://DeviceIP:Port/broadview/bst/configure-bst-thresholds`

c. POST the JSON for defining the BView BST parameters.
Example:

```
{
    "jsonrpc":"2.0",
    "method":"get-bst-thresholds",
    "asic-id":"0",
    "params": {
    "include-ingress-port-priority-group": 1,
    "include-ingress-port-service-pool": 0,
    "include-ingress-service-pool": 0,
    "include-egress-port-service-pool": 0,
    "include-egress-service-pool": 0,
    "include-egress-us-queue": 1,
    "include-egress-us-queue-group": 0,
    "include-egress-mc-queue": 1,
    "include-egress-cpu-queue": 1,
    "include-egress-rqe-queue": 0,
    "include-device": 0,
    }
     "id": 1,
}
```

NOTE
We recommend that you use sniffer to make sure the BView Heartbeat messages and Periodic Reports are from the managed item to the BView Collector.

Plug-in Configuration Example

The following JSON example shows the BView BST plug-in configuration:

```
{
   "PLUGIN_CONFIG": {
        "PROTOCOL": "http",
        "SOCKET_PORT": 9070,
        "SEVERITY_INFO": "[]",
        "SEVERITY_WARN": "[device]",
        "SEVERITY_MINOR": "[ingress-port-service-pool]",
        "SEVERITY_MAJOR": "[ingress-port-priority-group, egress-cpu-queue]",
        "SEVERITY_CRITICAL": "[egress-service-pool, ingress-service-pool, egress-service-pool]",
        "WORKER_THREAD_COUNT": 32,
        "BVIEW_CONFIG_REQUEST_INTERVAL_MIN": 30,
        "ALARM_CLEAR_INTERVAL_SEC": 60,
        "SWITCH_INITIALIZATION_TIMEOUT_SEC": 300,
        "DOMAIN_ID": 0
        }
   }
```

- **PROTOCOL**
  The communication protocol with the Broadview Agent.
  Values: "http" only
- **SOCKET_PORT**
  The plug-in listens to this port.
- **SEVERITY_INFO**
  Comma separated list of realms. Alarms for the listed realms have the INFO severity.
- **SEVERITY_WARN**
  Comma separated list of realms. Alarms for the listed realms have the WARN severity.
- **SEVERITY_MINOR**
  Comma separated list of realms. Alarms for the listed realms have the MINOR severity.
- **SEVERITY_MAJOR**
  Comma separated list of realms. Alarms for the listed realms have the MAJOR severity.
- **SEVERITY_CRITICAL**
  Comma separated list of realms. Alarms for the listed realms have the CRITICAL severity.
- **WORKER_THREAD_COUNT**
  Number of workers (threads) that Netty uses to process incoming messages. This parameter is usually the number of cores * 2.
- **BVIEW_CONFIG_REQUEST_INTERVAL_MIN**
  How often the plug-in updates the threshold cache and updates the BView BST configuration.
- **ALARM_CLEAR_INTERVAL_SEC**
  How often the plug-in checks alarms cache and clears expired alarms.
- **SWITCH_INITIALIZATION_TIMEOUT_SEC**
  Time required to process and add to the database inventory data before consuming Trigger and Periodic reports.
- **DOMAIN_ID**

CA Virtual Network Assurance assigns inventory from this plug-in to the specified domain.

Troubleshoot Plug-in Issues

Issues with the BView BST plug-in are logged in the following locations:

- `/opt/CA/VNA/wildfly/standalone/log/gateway.log`
- `/opt/CA/VNA/wildfly/standalone/log/oc.log .`

You can adjust the log levels. For more information, see Troubleshooting.

Follow these steps:

1. Ensure the MoD Container on the switch is functioning correctly:
   a. Log in to the switch and run the following command:
      ```
      docker ps|grep tam
      ```
      The switch should appear up.
      Example:
      ```
      6389a19196e9    docker-tam:latest    "/usr/bin/supervisord"   3 days ago   Up 2 days   tam
      ```
   b. BView Periodic Reports
   c. If configured, BView Trigger Reports (depends on traffic load)
2. Check the log files.
3. Reproduce packets drops with any Drop Reason supported by MoD.
4. Use the DX NetOps Virtual Network Assurance client to confirm that the MoD DX NetOps Virtual Network Assurance notifications are available at the DX NetOps Virtual Network Assurance Gateway.
5. Check the DX NetOps Spectrum and DX NetOps Performance Management logs.

Broadcom Mirror on Drop

The BroadView Packet Drop Monitor (PDM) feature helps network administrators understand the packet drop pattern on Broadcom silicon, such as the BCM56870 (Trident 3). The PDM feature allows users to look for drops within specific patterns of flows. The PDM feature also reports the dropped packets to a specified collector. The Broadcom Mirror on Drop plug-in serves as Mirror on Drop (MoD) alarms collector. You can view MoD alarms in the NetOps Portal alarms view and DX NetOps Spectrum.

The Broadcom Mirror on Drop plug-in provides the following drop reasons:

- **L3_DEST_MISS**
  Missing L3 route/host entry
- **UNKNOWN_VLAN**
  Vlan of packet unknown
- **MARTIAN_ADDR**
  Packet with source IP=0.0.0.0
- **DOS_ATTACK**
  Packet with Source IP=Destination IP
- **L3_MTU_FAIL**
  MTU > configured MTU
- **L3_ADDR_BIND_FAIL**
  L3 Packet with incorrect SMAC
- **INVALID_TPID**
  Packet with an invalid TPID
  (Tag Protocol Identifier)
- **L3_HEADER_ERROR**

Error in L3 header
- **TTL1**
  Packet with TTL=1
- **TTL**
  Packet with TTL=0
- **PARITY_ERROR**
  Parity error detected

Supported Releases

DX NetOps Virtual Network Assurance supports the following releases of SONiC:

- SONiC BUZZNIK

Requirements

- Configure the MoD feature on the switch. In general, this includes specifying flows (five-tuple) of interest and related details. For exmample, specify the interface for drop detection, and so on.
  For more information, see the Broadview Packet Drop Monitor section of the official StrataXGS-MOD-AN100 documentation.

  NOTE
  Since MoD/PDM is a premium feature, the documentation is unavailable for some customers. To get it, contact Support and request access to a document with the "StrataXGS-MOD-AN100" DocSafe ID.

Plug-in Configuration Example

The following JSON example shows the Broadcom Mirror on Drop plug-in configuration:

```
{
   "PLUGIN_CONFIG": {
        "BVIEW_AGENT_PROTOCOL": "http",
        "BVIEW_AGENT_PORT": 8361,
        "MOD_COLLECTOR_PORT": 7777,
        "SONIC_REST_PROTOCOL": "https",
        "SONIC_REST_PORT": 443,
        "SONIC_REST_USER": "restapi",
        "SONIC_REST_PASSWORD": "restapi",
        "WORKER_THREAD_COUNT": 32,
        "INVENTORY_UPDATE_INTERVAL_MIN": 1440,
        "ALARM_CLEAR_INTERVAL_MIN": 10080,
        "DISABLE_CLEAR_EVENTS_PROCESSING": 0,
        "SWITCH_INITIALIZATION_TIMEOUT_SEC": 300,
        "DOMAIN_ID": 0
        }
   }
```

- **BVIEW_AGENT_PROTOCOL**
  The communication protocol with the Broadview Agent.
  Values: "http" only
- **BVIEW_AGENT_PORT**
  The communication port with the BroadView Agent.
- **MOD_COLLECTOR_PORT**
  The UDP port to which MoD alarms are sent from the switch.
- **SONIC_REST_PROTOCOL**

The communication protocol with the SONiC REST API.
Values: "http" or "https"

- **SONIC_REST_USER**
  Username for SONiC REST API authentication.
- **SONIC_REST_PASSWORD**
  Password for SONiC REST API authentication.
- **SONIC_REST_PORT**
  The communication port with the SONiC REST API.
- **WORKER_THREAD_COUNT**
  Number of workers (threads) that the plug-in uses to process incoming messages and data. This parameter is usually the number of cores * 2.
- **INVENTORY_UPDATE_INTERVAL_MIN**
  How often the plug-in updates the inventory cache with interface and flow information.
- **ALARM_CLEAR_INTERVAL_SEC**
  How often the plug-in checks alarms cache and clears expired alarms.
- **DISABLE_CLEAR_EVENTS_PROCESSING**
  If set to 1, disables `DROP_STOP` MoD packet processing for the plug-in. As a result, MoD alarms remain Active until the scheduled cleanup task in DX NetOps Virtual Network Assurance, or you manually change them in DX NetOps Spectrum.
- **SWITCH_INITIALIZATION_TIMEOUT_SEC**
  Time required to process and add to the database inventory data before consuming Trigger and Periodic reports.
- **DOMAIN_ID**
  CA Virtual Network Assurance assigns inventory from this plug-in to the specified domain.

Mod Alarms

The Broadcom Mirror on Drop plug-in regularly checks whether the time between now and the timestampt of the last MoD Event received for the same MoD Flow entity is more than the configured `ALARM_CLEAR_INTERVAL_MIN` value. If the time is more, then the VNA alarm is cleared. The check interval is calculated as `ALARM_CLEAR_INTERVAL_MIN / 10`, but cannot be less than 1 minute and more than 60 minutes.

> **IMPORTANT**
> If `DISABLE_CLEAR_EVENTS_PROCESSING=1` , the plug-in does not process `DROP_STOP` events from the switch. As a result, a lot of MoD alarms for short-living flows might occur. In this case, you can clear MoD alarms only with alarm clear tasks in DX NetOps Virtual Network Assurance, or manually in DX NetOps Spectrum.

If a you condifure the MoD feature and specify an interface name during the MoD Session creating, MoD alarms are raised on the appropriate interface in DX NetOps. Otherwise, MoD alarms are raised on a device item.

Troubleshoot Plug-in Issues

Issues with the Broadcom Mirror on Drop plug-in are logged in the following locations:

- `/opt/CA/VNA/wildfly/standalone/log/gateway.log`
- `/opt/CA/VNA/wildfly/standalone/log/oc.log`.

Follow these steps:

1. To verify that the BView Container on your the switch is working, use a sniffer and verify that the following messages are being sent:
   - BView Heartbeat messages
   - BView Periodic Reports
   - If configured, BView Trigger Reports (depends on traffic load)

2. Check the log files.
3. Use the DX NetOps Virtual Network Assurance client to confirm that the BView BST DX NetOps Virtual Network Assurance notifications are available at the DX NetOps Virtual Network Assurance Gateway.
4. Check the DX NetOps Spectrum and DX NetOps Performance Management logs.

Cisco ACI

Cisco Application Centric Infrastructure (ACI) is an orchestrator that applies the SDN policy model across networks, servers, storage, security, and services.

The plug-in collects inventory for the following items:

- ACI interfaces
- ACI tenants
- ACI virtual routing and forwarding (VRF)
- APIC controller
- APIC interfaces
- Applications profiles
- Bridge domains
- Contracts
- End points
- End point groups (EPG)
- L2/L3 EPGs
- Subnets
- Switches (leafs and spines)

Cisco ACI collects the following performance metrics:

Table 6:

Item Type	VNA Metric Families and Metrics
Fabric	• Health Score – Health Score – Critical Count – Major Count – Minor Count – Warning Count
Tenant	• Health Score – Health Score – Critical Count – Major Count – Minor Count – Warning Count
Pod	• Health Score – Health Score – Critical Count – Major Count – Minor Count – Warning Count

Item Type	VNA Metric Families and Metrics
Interface	• Interface – Incoming Bytes – Outgoing Bytes – Incoming Packets – Outgoing Packets • Health Score – Health Score – Critical Count – Major Count – Minor Count – Warning Count
Controller	• Disk – Disk Usage – Disk Available – Disk Capacity
Switch Policy	• Cam – Policy Capacity – Policy Usage
Application Profile	• Health Score – Health Score – Critical Count – Major Count – Minor Count – Warning Count
Bridge Domain	• Health Score – Health Score – Critical Count – Major Count – Minor Count – Warning Count
Contracts	• Health Score – Health Score – Critical Count – Major Count – Minor Count – Warning Count
End Point & End Point Group	• Health Score – Health Score – Critical Count – Major Count – Minor Count – Warning Count
VRF	• Health Score – Health Score – Critical Count – Major Count – Minor Count – Warning Count

Supported Releases

DX NetOps Virtual Network Assurance (VNA) supports the following releases of Cisco ACI:

- Cisco ACI 4.2.4
- Cisco ACI 4.0
- Cisco ACI 3.2
- Cisco ACI 2.0

Requirements

- Add the DX NetOps Virtual Network Assurance host IP address to the list of IPs in the Client Group Policies.
- For the DX NetOps Performance Management Data Collector that polls switches through SNMP, add the host IP address to the list of IPs in the Client Group Policies.
- The service account requires read-only access. We recommend, but do not require, the service account have access to all tenants.

  NOTE

  If you grant access to only a single tenant, then you must deploy multiple plug-ins for visibility into the other tenants (one per tenant).

  You can have multiple plug-ins of the same technology type on a single instance of VNA.
- We only require the service account have access to a single APIC. Other APIC controllers are identified through an inventory request.

  NOTE

  The plugin does not use the controllers that are discovered through the inventory request.

Plug-in Configuration Example

The following JSON example shows the Cisco ACI plug-in configuration:

```
{
"PLUGIN_CONFIG": {
     "APIC_HOST_IP": "10.241.17.185",
     "APIC_HOST_aaaUser_NAME": "api",
     "APIC_HOST_aaaUser_PASSWORD": "apiReadOn1y",
     "APIC_MAX_PAGE_SIZE": 15,
     "PROTOCOL": "https",
     "INVENTORY_ENABLE_ENDPOINTS": "Yes",
     "INVENTORY_POLL_RATE": "0 */10 *",
     "INVENTORY_DELTA_TIME": 600,
     "PERFORMANCE_POLL_RATE": "0 */5 *",
     "PERFORMANCE_DELTA_TIME": 300,
     "AVAILABILITY_POLL_RATE": "0 */5 *",
     "AVAILABILITY_DELTA_TIME": 300,
     "FAULT_SUBSCRIPTION_ENABLED": true,
     "EVENT_SUBSCRIPTION_ENABLED": true,
     "HISTORICAL_EVENT_DAYS": 1,
     "DOMAIN_ID": 0
     }
}
```

- **APIC_HOST_IP**

The IP address of the APIC controller host
- **APIC_HOST_aaaUser_NAME**
  The APIC controller user name
- **APIC_HOST_aaaUser_PASSWORD**
  The APIC controller password
- **APIC_MAX_PAGE_SIZE**
  The number of response objects per page
- **PROTOCOL**
  The communication protocol with the APIC controller
  Values: http or https (case-sensitive)
- **INVENTORY_ENABLE_ENDPOINTS**
  Whether to poll the endpoint inventory.
  Values: true, false, Yes, No
  Default: Yes
- **INVENTORY_POLL_RATE**
  How often the product collects inventory data
- **INVENTORY_DELTA_TIME**
  Difference between polls (in seconds)
- **PERFORMANCE_POLL_RATE**
  How often the product collects performance data
- **PERFORMANCE_DELTA_TIME**
  Difference between polls (in seconds) for performance data requests
- **AVAILABILITY_POLL_RATE**
  How often the product polls the availability of the controller
- **AVAILABILITY_DELTA_TIME**
  Difference between polls (in seconds) for availability data requests
- **FAULT_SUBSCRIPTION_ENABLED**
  Whether fault subscription is enabled.
- **EVENT_SUBSCRIPTION_ENABLED**
  Whether event subscription is enabled.
- **HISTORICAL_EVENT_DAYS**
  The number of days to retrieve historical events.
  Default: 1
  Maximum: 30
- **DOMAIN_ID**
  DX NetOps Virtual Network Assurance assigns inventory from this plug-in to the specified domain.

Configure Faults

The ACI plug-in collects faults based on certain patterns. Configure faults to add, delete, or update fault patterns. You can add new patterns in any order. For assistance, contact Support.

Follow these steps:

1. Stop the wildfly service:
   ```
   service wildfly stop
   ```
2. Edit the fault configuration file:
   VNA_install_directory/plugins/ACI Plugin/config/aci-fault-config.xml
   Example:
   ```
   <fault-patterns> <regex>.*fv-\[(.+?)].*</regex> <regex>.*ra-\[(.+?)].*</regex>
   <regex>.*client-\[(.+?)].*</regex> <regex>.*rtd-\[(uni\/.+?)\/</regex> <regex>.*jobs-
   ```

```
\[(uni\/.+?)\/</regex> <regex>(.+?)\/sys</regex> <regex>(.+?)\/local</regex>
<regex>(uni\/.+?)\/</regex> </fault-patterns> >
```

3. Start the wildfly service:
   ```
   service wildfly start
   ```

Configure Filters

You can configure blacklisting filters based on fault codes and severities. You can configure whitelisting filters as well. You can configure multiple fault code separated by a comma. DX NetOps Virtual Network Assurance filters new alarms generated after you configure the filter. The filter does not apply to pre-existing alarms. If the customer wants to apply the filter on existing alarms in the spectrum, the existing engine needs to be deleted from VNA and deploy a new engine after configuring the filter.

Follow these steps:

1. Stop the wildfly service:
   ```
   service wildfly stop
   ```
2. Edit the fault configuration file: *VNA_install_directory*/plugins/ACI Plugin/config/aci-fault-config.xml

   Examples:

   NOTE
   When you upgrade to 3.7.8 and higher, all the previously set values for fault-codes, fault-types, and fault-severities are cleared. All the faults are whitelisted and saved in the database. To prevent this, stop the Wildfly service within 5 minutes of the upgrade and configure the filters as needed.

   ```
   <filter enabled="true" filterType="whitelist"
        <fault-types>configuration,operational</fault-types>
        <fault-codes>F2603,F2631,F2632,F2633,F2634</fault-codes>
        <fault-severities>minor,warning</fault-severities>
   </filter>
   ```

   - **faultTypes**
     Values:
     - **whitelist**
       Shows only the alarms that match all the configured filter values.
     - **blacklist**
       Excludes only the alarms that match all the configured filter values.
   - **<fault-types>**
     Values: config, generic, equipment, connectivity, environmental, management, network, operational
   - **<fault-severities>**
     Values: critical, major, minor, warning, info, cleared
3. Start the wildfly service:
   ```
   service wildfly start
   ```

Cisco Meraki

Cisco Meraki includes cloud-controlled WiFi and SD-WAN solutions centrally managed from the web. This plug-in allows you to monitor MR WiFi Access Point devices and MX Security and SD-WAN appliances.The plug-in collects inventory for the following components:

- Organization
- Network
- Access Point
- SSID
- L3FirewallRules for SSIDs and Routers
- L7FirewallRules for Routers
- CellularFirewallRules for Routers
- VPNFirewallRules for Routers
- Meraki Controller
- Routers
- Uplink Interfaces

NOTE
The relationship between an access point and an SSID is discovered only when traffic has flowed from the access point on the SSID at least once in the past 7 days. If no traffic flowed from an access point on any SSID in the past 7 days, CA Virtual Network Assurance does not record the relationship.

The plug-in collects the following performance metrics:

Item Type	VNA Metric Families and Metrics
Access Point	Normalized Mobile WLAN AP Info • Bytes • Bytes In • Bytes Out • Client • Authentication Failures (20.2.2 and higher only) • Association Failures (20.2.2 and higher only) • Successful Authentications (20.2.2 and higher only) • Successful Associations (20.2.2 and higher only) • Successful DHCP (20.2.2 and higher only) • Successful DNS (20.2.2 and higher only) • Successful Connections (20.2.2 and higher only)
Meraki Controller	Normalized Wireless Controller Info • Active Access Points • Clients Associated
Router	• Device Utilization
Tunnel	• Latency • Packet Loss Percentage

Enable Tunnel Metric Collections

To enable tunnel metric collections, add the tunnels to the Uplink statistics. For more information, see the Cisco Meraki documentation.

Follow these steps:

1. Go to the Cisco Meraki Dashboard, Security & SD-WAN, Configure, SD-WAN & traffic shaping, Uplink configuration, and Uplink statistics.
2. Add the IPv4 public addresses of the WAN Interfaces for the other routers that have tunnels with this router.

Requirements

- Access to the Cisco Meraki Dashboard API must be enabled.
- An API key must be generated on your profile.

For more information, see the Cisco Meraki documentation.

Plug-in Configuration Example

The following JSON example shows the Cisco Meraki plug-in configuration:

```
{
"PLUGIN_CONFIG": {
    "MERAKI_DASHBOARD_HOST": "api.meraki.com",
    "MERAKI_API_KEY": "xxx3xxxxxe9xx",
    "PROTOCOL": "https",
    "DELAY_BETWEEN_REST_CALLS_IN_MS": 400,
    "MAX_RETRIES": 3,
    "RETRY_DELAY_IN_MS": 500,
    "SYSLOG_PORT": 1542,
    "INVENTORY_POLL_RATE": "0 */10 *",
    "INVENTORY_DELTA_TIME": 600,
    "PERFORMANCE_POLL_RATE": "0 */10 *",
    "PERFORMANCE_DELTA_TIME": 600,
    "BLACKLISTED_EVENT_TYPES":"[]",
    "POLL_SSIDS":"true",
    "POLL_POLICIES":"true",
    "POLL_CLIENTS":"true",
    "NETWORK_ID_LIST":"[]",
    "DOMAIN_ID": 0
    }
}
```

20.2.2 and Higher Only

```
{
"PLUGIN_CONFIG": {
    "MERAKI_DASHBOARD_HOST": "api.meraki.com",
    "MERAKI_API_KEY": "xxx3xxxxxe9xx",
    "PROTOCOL": "https",
    "DELAY_BETWEEN_REST_CALLS_IN_MS": 400,
    "MAX_RETRIES": 3,
    "RETRY_DELAY_IN_MS": 500,
    "SYSLOG_PORT": 1542,
    "INVENTORY_POLL_RATE": "0 */10 *",
    "INVENTORY_DELTA_TIME": 600,
    "PERFORMANCE_POLL_RATE": "0 */10 *",
    "PERFORMANCE_DELTA_TIME": 600,
    "NOTIFICATION_POLL_RATE": "0 */1 *",
    "NOTIFICATION_DELTA_TIME": 60
    "ENABLE_REST_API_EVENTS":"false",
    "ENABLE_SYSLOG_EVENTS":"false",
    "BLACKLISTED_EVENT_TYPES":"[]",
    "POLL_SSIDS":"true",
    "POLL_POLICIES":"true",
```

```
    "POLL_CLIENTS":"true",
    "NETWORK_ID_LIST":"[]",
    "DOMAIN_ID": 0
    }
}
```

- **MERAKI_DASHBOARD_HOST**
  The address host of the Meraki dashboard
- **MERAKI_USER_NAME**
  The Meraki dashboard user name
- **MERAKI_API_KEY**
  The API key of the Meraki dashboard
- **PROTOCOL**
  The communication protocol with the Meraki dashboard
  Values: http or https (case-sensitive)
- **DELAY_BETWEEN_REST_CALLS_IN_MS**
  Shows the delay between consecutive rest calls in milliseconds. The plug-in can make the maximum five REST calls to the Meraki Controller in a second.
- **MAX_RETRIES**
  The maximum retry attempts for a REST call by the plug-in in a poll when the Meraki Dashboard API returns the HTTP response code 429: 'Too Many Requests'.
- **RETRY_DELAY_IN_MS**
  The time delay in milliseconds that the plug-in waits before reattempting a REST call when the Meraki Dashboard API returns the HTTP response code 429: 'Too Many Requests'.
- **SYSLOG_PORT**
  The port to which the CA VNA syslog listens to
- **INVENTORY_POLL_RATE**
  How often the product collects inventory data
- **INVENTORY_DELTA_TIME**
  Difference between polls (in seconds)
- **PERFORMANCE_POLL_RATE**
  How often the product collects performance data
- **PERFORMANCE_DELTA_TIME**
  Difference between polls (in seconds) for performance data requests
- **NOTIFICATION_POLL_RATE** (20.2.2 and Higher Only)
  How often the product collects event notifications data through the REST API
- **NOTIFICATION_DELTA_TIME** (20.2.2 and Higher Only)
  Difference between polls (in seconds) for event notification data requests through the REST API
- **ENABLE_REST_API_EVENT** (20.2.2 and Higher Only)
  Whether to enable event notifications polling through the REST API
  Values: true or false

  NOTE
  You cannot enable this parameter at the same time as the `ENABLE_SYSLOG_EVENTS` parameter.
- **ENABLE_SYSLOG_EVENTS** (20.2.2 and Higher Only)
  Whether to enable event notifications polling through SYSLOG
  Values: true or false

  NOTE
  You cannot enable this parameter at the same time as the `ENABLE_REST_API_EVENT` parameter.
- **BLACKLISTED_EVENT_TYPES**
  Comma separated list of event types that are filtered out

Example: `"BLACKLISTED_EVENT_TYPES":"[vpn_connectivity_change,ids_alerts]",`
The following Access Point event types are supported:
- `association`
- `disassociation`
- `wpa_auth`
- `wpa_deauth`
- `8021x_eap_failure`
- `8021x_deauth`
- `8021x_eap_success`
- `splash_auth`
- `device_packet_flood`
- `device_packet_flood`
- `rogue_ssid_detected`
- `ssid_spoofing_detected`

For more information, see the Meraki MR Access Point.
The following Router event types are supported:
- `vpn_connectivity_change`
- `uplink_connectivity_change`
- `client_dhcp_lease`
- `ids_alerts`
- `ids_alerted`
- `security_filtering_file_scanned`
- `security_filtering_disposition_change`

For more information, see the Meraki MX Security Appliance.

- **POLL_SSIDS**
  Whether to enable SSID polling
  Values: true or false
- **POLL_POLICIES**
  Whether to enable policy polling
  Values: true or false
  NOTE
  If SSID polling is disabled, we recommend that you specify false.
- **POLL_CLIENTS**
  Whether to enable performance polling of access points and Meraki Controller metrics
  Values: true or false
- **NETWORK_ID_LIST**
  Network IDs to poll
  NOTE
  If empty, all available networks are polled.
- **DOMAIN_ID**
  DX NetOps Virtual Network Assurance assigns inventory from this plug-in to the specified domain

Configure Dashboard

To enable CA VNA to listen to syslog messages from the Access Points and Routers, add the following configuration for each of the networks. See the steps to configure dashboard in the Cisco documentation on Syslog Server Overview and Configuration.

- Enter the VNA IP as the Server IP.
- Enter the Syslog Port from the plug-in configuration as the Port.
- For Access Points, add the `Wireless event log` and the `Air Marshal events` as the Roles.
- For Routers, add the `Security events` and the `Appliance event log` as Roles.

Nuage

Nuage is an SDN solution optimized for data center networks. DX NetOps Virtual Network Assurance collects inventory and performance data from Nuage.

NOTE

This plug-in currently supports only the VSP (virtual services platform) SD-WAN solution with NSG (network service gateway) configuration of Nuage.

The plug-in collects inventory for the following items:

- Enterprises
- L3 domains
- Sites (3.7.9 and higher only)
- Network services gateway (NSG)
- NSG ports
- Tunnels (3.7.9 and higher only)
- Policy Groups (3.7.9 and higher only)
- Policies/SLA Paths (3.7.9 and higher only)
- Application/SLA Paths (3.7.9 and higher only)
- Alarms and Events

  NOTE
  DX NetOps Spectrum consumes these alarms
- Subnets
- Virtual ports
- Virtual router switch (VRS)
- Virtual services controller (VSC)
- VRS disk
- VSC disk
- Zones

The plug-in collects the following metrics for all QoS queues for network uplink ports:

- Bytes out
- Packets out
- Packets dropped
- Overlimits
- Requeues
- Lended tokens
- Borrowed tokens

The plug-in collects the following metrics for vPorts:

- Packets In/Out
- Bytes In/Out
- Errors In/Out
- Dropped Packets In/Out

The plug-in collects the following metrics for VRSs:

- Average CPU utilization
- Average memory utilization
- System uptime availability

The plug-in collects the following metrics for VSCs:

- Average CPU utilization
- Average memory utilization

The plug-in collects the following metrics for VSC disks:

- Available bytes
- Capacity
- Used bytes

The plug-in collects the following metrics for tunnels (3.7.9 and higher only):

- Jitter
- Latency
- Packet loss

The plug-in collects the following metrics for application/SLA paths (3.7.9 and higher only)

- Percentage of jitter SLA threshold
- Percentage of packet loss SLA threshold
- Percentage of latency SLA threshold

Supported Releases

DX NetOps Virtual Network Assurance supports the following release:

- Nuage 4.0.0 R3

Requirements

- Stats collection must be enabled on the Nuage items.

Plug-in Configuration Example

The following JSON example shows the Nuage plug-in configuration:

```
{
 "PLUGIN_CONFIG": {
     "VSD_HOST_IP": "10.241.17.16",
     "VSD_PORT": 8443,
     "PROTOCOL": "https",
     "STATS_HOST_IP": "10.241.17.26",
     "STATS_PORT": 9200,
     "STATS_PROTOCOL": "http",
     "VSD_USERNAME": "csproot",
     "VSD_PASSWORD": "csproot",
     "NUAGE_ORGANIZATION": "csp",
     "INVENTORY_POLL_RATE": "0 */10 *",
     "INVENTORY_DELTA_TIME": 600,
```

```
      "AVAILABILITY_POLL_RATE": "0 */5 *",
      "AVAILABILITY_DELTA_TIME": 300,
      "PERFORMANCE_POLL_RATE": "0 */5 *",
      "PERFORMANCE_DELTA_TIME": 300,
      "DOMAIN_ID": 0
      }
 }
```

3.7.9 and Higher Only:

```
 {
  "PLUGIN_CONFIG": {
      "VSD_HOST_IP": "10.241.17.16",
      "VSD_PORT": 8443,
      "PROTOCOL": "https",
      "STATS_HOST_IP": "10.241.17.26",
      "STATS_PORT": 9200,
      "STATS_PROTOCOL": "http",
      "VSD_USERNAME": "csproot",
      "VSD_PASSWORD": "csproot",
      "NUAGE_ORGANIZATION": "csp",
      "INVENTORY_POLL_RATE": "0 */10 *",
      "INVENTORY_DELTA_TIME": 600,
      "AVAILABILITY_POLL_RATE": "0 */5 *",
      "AVAILABILITY_DELTA_TIME": 300,
      "PERFORMANCE_POLL_RATE": "0 */5 *",
      "PERFORMANCE_DELTA_TIME": 300,
      "NOTIFICATION_POLL_RATE": "0 */5 *",
      "NOTIFICAION_DELTA_TIME": 300,
      "TIMEZONE": 0,
      "DOMAIN_ID": 0
      }
 }
```

- **VSD_HOST_IP**
  The Virtualized Services Directory (VSD) host
- **VSD_PORT**
  The port that the VSD UI/API server listens on
  Default: 8443
- **PROTOCOL**
  The communication protocol with the VSD
  Values: http or https
- **STATS_HOST_IP**
  The IP address of the stats server
- **STATS_PORT**
  The port the stats server is listening on for REST requests
  Default: 9200
- **STATS_PROTOCOL**

The communication protocol with the stats server
Values: http or https

- **VSD_USERNAME**
  The VSD username
- **VSD_PASSWORD**
  The VSD password
- **NUAGE_ORGANIZATION**
  The name of the Nuage enterprise to manage.
- **INVENTORY_POLL_RATE**
  How often the product collects inventory data
- **INVENTORY_DELTA_TIME**
  Difference between polls (in seconds)
- **AVAILABILITY_POLL_RATE**
  How often the product collects availability data
- **AVAILABILITY_DELTA_TIME**
  Difference between polls (in seconds)
- **PERFORMANCE_POLL_RATE**
  How often the product collects performance data
- **PERFORMANCE_DELTA_TIME**
  Difference between polls (in seconds)

  NOTE

  `PERFORMANCE_POLL_RATE` and `PERFORMANCE_DELTA_TIME` must be even multiples of the poll rate configured for the Nuage environment.
- **NOTIFICATION_POLL_RATE (3.7.9 and higher only)**
  How often the product collects notification data
- **NOTIFICATION_DELTA_TIME (3.7.9 and higher only)**
  Difference between polls (in seconds)
- **TIMEZONE (3.7.9 and higher only)**
  The time zone of the system, which must match the VSD time zone

  WARNING
  If this parameter is set incorrectly, the plug-in might not collect metrics for alarms and events.
- **DOMAIN_ID**
  DX NetOps Virtual Network Assurance assigns inventory from this plug-in to the specified domain.

OpenContrail

OpenContrail is an open source network controller. DX NetOps Virtual Network Assurance collects host, performance, and service chain data from OpenContrail.

The plug-in collects inventory for the following items:

- Controller
- Tenant
- vRouter

NOTE
In DX NetOps Performance Management, the vRouter is reported as a vSwitch.

- Service Function Chain (SFC)
- Virtual Network Function (VNF)
- VM
- Interface

The plug-in collects the following metrics for interfaces on the vRouter host:

- Incoming Bytes
- Outgoing Bytes
- Incoming Packets
- Outgoing Packets
- Interface speed

The plug-in collects the following metrics for the vRouter host:

- Total memory
- Current memory usage
- CPU count

The plug-in collects the following metrics for the vRouter process:

- Resident memory
- Virtual memory
- Peak virtual memory

The plug-in calculates the following metrics:

- VNFs per Hypervisor
- VNFs per Tenant
- SFCs per Tenant

Supported Releases

DX NetOps Virtual Network Assurance supports the following releases of OpenContrail:

- 2.20
- 2.21

Requirements

- The configuration and analytics endpoint ports have connectivity to the DX NetOps Virtual Network Assurance host.
- The Orchestrator server and the Analytics server are running.

Plug-in Configuration Example

The following JSON example shows the OpenContrail plug-in configuration:

```
{
"PLUGIN_CONFIG": {
     "OPENSTACK_KEYSTONE_IP": "10.241.18.159",
     "OPENSTACK_KEYSTONE_PORT": 5000,
     "OPENSTACK_KEYSTONE_USER_TENANT": "admin",
```

```
      "OPENSTACK_KEYSTONE_USER_NAME": "admin",
      "OPENSTACK_KEYSTONE_USER_PASSWORD": "admin",
      "CONTRAIL_ORCHESTRATOR_IP": "10.241.18.156",
      "CONTRAIL_ORCHESTRATOR_PORT": 8082,
      "CONTRAIL_ANALYTICS_IP": "10.241.18.156",
      "CONTRAIL_ANALYTICS_PORT": 8081,
      "PROTOCOL": "http",
      "INVENTORY_POLL_RATE": "0 */5 *",
      "INVENTORY_DELTA_TIME": 300,
      "AVAILABILITY_POLL_RATE": "0 */5 *",
      "AVAILABILITY_DELTA_TIME": 300,
      "PERFORMANCE_POLL_RATE": "0 */15 *",
      "PERFORMANCE_DELTA_TIME": 300,
      "DOMAIN_ID": 0
      }
}
```

- **OPENSTACK_KEYSTONE_IP**
  The hostname or IP address of OpenStack Identity Service (Keystone)
- **OPENSTACK_KEYSTONE_PORT**
  This parameter refers to the port number of OpenStack Identity Service (Keystone). This port can be a public or an administrative endpoint port.
- **OPENSTACK_KEYSTONE_USER_TENANT**
  The tenant name of the user that is used to access OpenStack Orchestrator REST APIs
- **OPENSTACK_KEYSTONE_USER_NAME**
  The username for connecting and executing OpenStack Orchestrator REST API
- **OPENSTACK_KEYSTONE_USER_PASSWORD**
  Password for connecting and executing OpenStack Orchestrator REST API
- **CONTRAIL_ORCHESTRATOR_IP**
  The hostname or IP address of the OpenContrail Orchestrator REST API Server
- **CONTRAIL_ORCHESTRATOR_PORT**
  The port number of the OpenContrail Orchestrator REST API Server
- **CONTRAIL_ANALYTICS_IP**
  The hostname or IP address of the OpenContrail Analytics REST API Server
- **CONTRAIL_ANALYTICS_PORT**
  The port number of the OpenContrail Analytics REST API Server
- **PROTOCOL**The communication protocol with the OpenContrail orchestrator
  Values: http or https (case-sensitive)
- **INVENTORY_POLL_RATE**
  How often the product collects inventory data
- **INVENTORY_DELTA_TIME**
  The time difference between inventory polls (in seconds)
- **AVAILABILITY_POLL_RATE**
  How often the product polls the availability of the controller
- **AVAILABILITY_DELTA_TIME**
  The time difference between availability polls (in seconds)
- **PERFORMANCE_POLL_RATE**

How often the product collects performance data
- **PERFORMANCE_DELTA_TIME**
  The time difference between performance polls (in seconds)
- **DOMAIN_ID** DX NetOps Virtual Network Assurance assigns inventory from this plug-in to the specified domain.

OpenDaylight

OpenDaylight is an open source network controller. DX NetOps Virtual Network Assurance collects service chain data from the topology endpoint.

This plug-in collects inventory for the following items:

- Controller
- Service function chain (SFC)
- Virtual network function (VNF)

The plug-in calculates the following metrics:

- VNFs per Hypervisor
- VNFs per Tenant
- SFCs per Tenant

Supported Releases

DX NetOps Virtual Network Assurance supports the following releases of OpenDaylight:

- Nitrogen

Requirements

- The ODL CONTROLLER PORT must be open to the DX NetOps Virtual Network Assurance host.
- To support service chain reporting, manually configure the topology to create the links. For more information, see Configure The OpenDaylight Topology to Support Service Chain Monitoring.

Plug-in Configuration Example

The following JSON example shows the OpenDaylight plug-in configuration:

```
{
 "PLUGIN_CONFIG": {
     "ODL_CONTROLLER_IP": "10.198.254.162",
     "ODL_CONTROLLER_PORT": 8181,
     "PROTOCOL": "http",
     "ODL_CONTROLLER_USER_NAME": "admin",
     "ODL_CONTROLLER_PASSWORD": "admin",
     "INVENTORY_POLL_RATE": "0 */5 *",
     "INVENTORY_DELTA_TIME": 300,
     "AVAILABILITY_POLL_RATE": "0 */5 *",
     "AVAILABILITY_DELTA_TIME": 300,
     "DOMAIN_ID": 0
     }
}
```

- **ODL_CONTROLLER_IP**
  The hostname or IP address of OpenDaylight Controller
- **ODL_CONTROLLER_PORT**
  The port number of OpenDaylight REST API
- **PROTOCOL**
  The communication protocol with the OpenDaylight REST API
  Values: http or https
- **ODL_CONTROLLER_USER_NAME**
  The username for executing OpenDaylight REST API
- **ODL_CONTROLLER_PASSWORD**
  The password for executing OpenDaylight REST API
- **INVENTORY_POLL_RATE**
  How often the product collects inventory data
- **INVENTORY_DELTA_TIME**
  The time difference between inventory polls (in seconds)
- **AVAILABILITY_POLL_RATE**
  How often the product polls the availability of the controller
- **AVAILABILITY_DELTA_TIME**
  The time difference between availability polls (in seconds)
- **DOMAIN_ID** DX NetOps Virtual Network Assurance assigns inventory from this plug-in to the specified domain.

OpenStack

OpenStack is an open source cloud orchestrator. DX NetOps Virtual Network Assurance collects inventory and performance data from OpenStack.

The plug-in collects inventory for the following items:

- Orchestrator
- Tenant
- Hypervisor
- VM
- Interface
- Network
- Subnet

The plug-in collects the following metrics for VMs:

- Incoming bytes on virtual interfaces
- Outgoing bytes on virtual interfaces
- Incoming packets on virtual interfaces
- Outgoing packets on virtual interfaces
- Disk capacity
- Disk allocation
- Disk usage
- Disk latency
- Disk IOPS
- Disk read request rate
- Disk write request rate
- Disk read bytes rate
- Disk write bytes rate
- CPU time
- CPU utilization
- vCPU count
- Memory
- Memory utilization

The plug-in calculates the following metrics:

- VMs per Hypervisor
- VMs per Tenant
- Networks per Tenant

Supported Releases

DX NetOps Virtual Network Assurance supports the following releases of OpenStack:

- Juno
- Kilo
- Liberty
- Mitaka
- Newton
- Ocata
- Pike

Requirements

- The ports for the following OpenStack services must be open to the DX NetOps Virtual Network Assurance host:
  - Keystone service
    Default: 35357
  - Nova service
    Default: 8774
  - Neutron service
    Default: 9696
  - Ceilometer service
    Default: 8777
  - Gnocchi service

Default: 8041

- To collect performance data for the Ocata release and previous releases, the Ceilometer monitoring component must be installed and running.
- To collect performance data for the Pike release and higher, the Gnocchi monitoring component must be installed and running.
- To collect inventory data, the Keystone user who is specified in the plug-in configuration XML requires administrator access.
  - For Keystone V3 API (Mitaka and later releases), the user must be part of the admin domain and the admin role must be added to the admin domain.

Plug-in Configuration Example

The following JSON example shows the OpenStack plug-in configuration:

```
{
"PLUGIN_CONFIG": {
     "KEYSTONE_IP": "10.241.17.11",
     "KEYSTONE_PORT": 35357,
     "KEYSTONE_USER_DOMAIN": "default",
     "KEYSTONE_USER_TENANT": "admin",
     "KEYSTONE_USER_NAME": "admin",
     "KEYSTONE_USER_PASSWORD": "admin",
     "PROTOCOL": "http",
     "INVENTORY_POLL_RATE": "0 */10 *",
     "INVENTORY_DELTA_TIME": 600,
     "AVAILABILITY_POLL_RATE": "0 */10 *",
     "AVAILABILITY_DELTA_TIME": 600,
     "PERFORMANCE_POLL_RATE": "0 */10 *",
     "PERFORMANCE_DELTA_TIME": 600,
     "SERVICE_ENDPOINT_TYPE": "admin",
     "DOMAIN_ID": 0,

     "GNOCCHI_COMPONENT": "true"
     }
}
```

- **KEYSTONE_IP**
  The hostname or IP address of OpenStack Identity Service (Keystone)
- **KEYSTONE_PORT**
  The administrative port number of OpenStack Identity Service (Keystone)
- **KEYSTONE_USER_DOMAIN**The domain name of the Keystone user that is used to access the OpenStack orchestrator REST API

  NOTE

  The **KEYSTONE_USER_DOMAIN** is case-sensitive for OpenStack versions before Mitaka.
- **KEYSTONE**_USER_TENANT
  The tenant name of the user that is used to access the OpenStack orchestrator REST APIs
- **KEYSTONE**_USER_NAME

The username for connecting and executing the OpenStack orchestrator REST API
- **KEYSTONE**_USER_PASSWORD
  Password for connecting and executing the OpenStack orchestrator REST API
- **PROTOCOL**
  The communication protocol with the OpenStack orchestrator REST API
  Values: http or https
- **INVENTORY_POLL_RATE**
  How often the product collects inventory data
- **INVENTORY_DELTA_TIME**
  The time difference between inventory polls (in seconds)
- **AVAILABILITY_POLL_RATE**
  How often the product polls the availability of the controller
- **AVAILABILITY_DELTA_TIME**
  The time difference between availability polls (in seconds)
- **PERFORMANCE_POLL_RATE**
  How often the product collects performance data
- **PERFORMANCE_DELTA_TIME**
  The time difference between performance polls (in seconds)
- **SERVICE_ENDPOINT_TYPE**
  Whether the urls for all openstack services except the identity service are admin, public, or internal.
  Values: admin or public
  Default: admin
- **GNOCCHI_COMPONENT**
  If the value is true then performance metrics are collected from the Gnocchi service. If the value is false, then performance metrics are collected from the Ceilometer service.
  Values: true or false
  Default: true
- **DOMAIN_ID**
  DX NetOps Virtual Network Assurance assigns inventory from this plug-in to the specified domain.

Open vSwitch

Open vSwitch (OVS) is an open source controller that provisions virtual switches. DX NetOps Virtual Network Assurance collects interface data from Open vSwitch.

> **NOTE**
>
> This documentation applies to 1.0.2 version of the OVS plug-in.

This plug-in collects inventory for the following items:

- VSwitch
- Interface

This plug-in collects the following metrics for virtual interfaces:

- Incoming Bytes
- Outgoing Bytes
- Incoming Packets
- Outgoing Packets
- Collisions
- Incoming CRC Errors
- Incoming Frame Errors
- Incoming RX Overruns
- Incoming Dropped Packets
- Outgoing Dropped Packets
- Incoming Errors
- Outgoing Errors
- Interface Speed

The plug-in collects the following metrics for the vSwitch entity:

- Virtual memory that the OVS processes use
- Resident memory that the OVS processes use
- Total memory of the system where the OVS is running
- Total CPU time that the OVS processes use

Supported Release

DX NetOps Virtual Network Assurance supports the following releases of Open vSwitch:

- 2.8.2

Requirements

- Add a remote to enable external connections on the OVS host. Run the following command on each compute node that DX NetOps Virtual Network Assurance monitors:

```
ovs-appctl -t ovsdb-server ovsdb-server/add-remote ptcp:6640
```

- Configure the OVS host to provide more performance information. Run the following command on the OVS host:

```
ovsdb-client transact '["Open_vSwitch",{"op": "update","row": {"other_config":
 ["map",[["enable-statistics","true"]] ]},"where": [],"table": "Open_vSwitch"}]'
```

Plug-in Configuration Example

The following JSON example shows the Open vSwitch plug-in configuration:

```
{
 "PLUGIN_CONFIG": {
     "OVS_HOST_IPS": [
     "192.168.1.1"
     ],
     "OVS_PORT": 6640,
     "PROTOCOL": "http",
```

```
      "INVENTORY_POLL_RATE": "0 */5 *",
      "INVENTORY_DELTA_TIME": 300,
      "AVAILABILITY_POLL_RATE": "0 */5 *",
      "AVAILABILITY_DELTA_TIME": 300,
      "DOMAIN_ID": 0
      }
}
```

- **OVS_HOST_IPS**The IP addresses of the OVSDB servers
- **OVS_PORT**
  The remote port that the OVSDB servers listen on
- **PROTOCOL**The communication protocol with the OVSDB server
  Values: http or https (case-sensitive)
- **INVENTROY_POLL_RATE**How often the product collects inventory data in cron syntax
- **INVENTORY_DELTA_TIME**The number of seconds in the inventory poll interval
  This value is used for metric calculations.
- **AVAILABILITY_POLL_RATE**
  How often the product polls the availability of the controller/orechestrator
- **AVAILABILITY_DELTA_TIME**The number of seconds in the availability poll interval
  This value is used for metric calculations.
- **DOMAIN_ID** DX NetOps Virtual Network Assurance assigns inventory from this plug-in to the specified domain.

Poll Rate Configuration

Silver Peak

Silver Peak is an SDN solution optimized for data center networks. DX NetOps Virtual Network Assurance collects inventory and performance data from Silver Peak. This plug-in currently supports only the SD-WAN solution with the Orchestrator (Unity Orchestrator Management) configuration of Silver Peak.

The plug-in collects inventory for the following items:

- Sites
- Orchestrator
- EdgeConnects
- EdgeConnects Interfaces
- Tunnels
- Overlay Tunnels
- Policy Groups
- Policies/SLA Classes (Business Intent Overlays)
- Application/SLA Paths
- Alarms and Events

  NOTE
  DX NetOps Spectrum consumes these alarms

The plug-in collects the following performance metrics:

NOTE
For 20.2.2, the APIs used to collect Tunnel data changed to avoid Orchestrator performance issues.

Item Type	VNA Metric Families and Metrics	SNMP Metric Families
Devices		• Availability • Reachability • SNMP Statistic • System • SIP Connection • IPv4 • Generic System • TCP Statistics • UDP Statistics • Interface
Interfaces	• Interface – Bits Per Second – Bits Per Second In – Bits Per Second Out – Packets – Packets In – Packets Out – Discarded Packets Per Second – Discarded Packets Per Second In – Discarded Packets Per Second Out – Maximum Bandwidth – Maximum Bandwidth In – Maximum Bandwidth Out – Percent Discards – Percent Discards In – Percent Discards Out – Speed In – Speed Out – Utilization – Utilization In – Utilization Out	

Item Type	VNA Metric Families and Metrics	SNMP Metric Families
Tunnels • Overlay Tunnel • Underlay Tunnel	• Tunnel – Jitter – Latency – Packet Loss Percentage – Bandwidth Utilization – Provisioned Speed In – Provisioned Speed Out – Availability – Time in Up State – Time in Down State – Time in Unknown State – Percent Time in Up State – Percent Time in Down State – Percent Time in Unknown State – Max Bandwidth Incoming – Max Bandwidth Outgoing – Bytes In **(20.2.2 and Higher Only)** – Bytes Out **(20.2.2 and Higher Only)**	
Application/SLA Paths	• SLA Path – Percentage of Jitter SLA Threshold – Percentage of Packet Loss SLA Threshold – Percentage of Latency SLA Threshold	

Supported Releases

DX NetOps Virtual Network Assurance supports the following release of Silver Peak:

- Unity Orchestrator 8.6.1.40151

Plug-in Configuration Example

The following JSON example shows the Silver Peak plug-in configuration:

```
{
"PLUGIN_CONFIG":{
    "ORCHESTRATOR_HOST": "192.168.3.209",
    "ORCHESTRATOR_HOST_PORT": 932,
    "ORCHESTRATOR_HOST_USER_NAME": "admin",
    "ORCHESTRATOR_HOST_PASSWORD": "admin",
    "PROTOCOL": "https",
    "LOGIN_TYPE": 0,
    "INVENTORY_POLL_RATE": "0 */5*",
    "PERFORMANCE_POLL_RATE": "0 */5*",
    "NOTIFICATION_POLL_RATE": "0 */5*",
    "DOMAIN_ID": "0 */1*",
    }
}
```

20.2.2 and Higher Only:

```
{
"PLUGIN_CONFIG":{
    "ORCHESTRATOR_HOST": "192.168.3.209",
    "ORCHESTRATOR_HOST_PORT": 932,
    "SYSLOG_PORT": 1542,
    "ORCHESTRATOR_HOST_USER_NAME": "admin",
    "ORCHESTRATOR_HOST_PASSWORD": "admin",
    "PROTOCOL": "https",
    "LOGIN_TYPE": 0,
    "ENABLE_REST_API_EVENTS":"true",
    "ENABLE_SYSLOG_EVENTS":"false",
    "INVENTORY_POLL_RATE": "0 */15*",
    "PERFORMANCE_POLL_RATE": "0 */15*",
    "PERFORMANCE_DELTA_TIME": 900,
    "NOTIFICATION_POLL_RATE": "0 */1*",
    "DOMAIN_ID": "0 */1*",
    }
}
```

- **ORCHESTRATOR_HOST_IP**
  The Unity Orchestrator Management host
- **ORCHESTRATOR_HOST_PORT**
  The port that the Orchestrator UI/API server listens on
  Default: 932
- **SYSLOG_PORT** (20.2.2 and Higher Only)
  The port that the DX NetOps Virtual Network Assurance SYSLOG listens to
  Default: 1542
- **ORCHESTRATOR_HOST_USER_NAME**
  The Orchestrator user name
- **ORCHESTRATOR_HOST_PASSWORD**
  The Orchestrator password
- **PROTOCOL**
  The communication protocol with the Orchestrator
  Values: http or https
- **LOGIN_TYPE**
  Values for using the API
  Values: 0 = local, 1 = RADIUS, 2 = TACACS
- **ENABLE_REST_API_EVENT** (20.2.2 and Higher Only)
  Whether to enable event notifications polling through the REST API
  Values: true or false

  NOTE
  You cannot enable this parameter at the same time as the `ENABLE_SYSLOG_EVENTS` parameter.
- **ENABLE_SYSLOG_EVENTS** (20.2.2 and Higher Only)
  Whether to enable event notifications polling through SYSLOG
  Values: true or false

  NOTE
  You cannot enable this parameter at the same time as the `ENABLE_REST_API_EVENT` parameter.
- **INVENTORY_POLL_RATE**

How often the product calls inventory data
- **PERFORMANCE_POLL_RATE**
  How often the product collects performance data
- **PERFORMANCE_DELTA_TIME** (20.2.2 and Higher Only)
  Difference between polls (in seconds) for performance data requests
- **NOTIFICATION_POLL_RATE**
  How often the product collects notification data
- **DOMAIN_ID**
  DX NetOps Virtual Network Assurance assigns inventory from this plug-in to the specified domain

Versa SD-WAN

DX NetOps Virtual Network Assurance collects inventory and performance metrics from Versa SD-WAN to support DX NetOps Performance Management SD-WAN monitoring.

The plug-in collects inventory for the following items:

- Sites

  NOTE

  Sites from Versa SD-WAN map to the Site Groups in Performance Center. If desired, you can manage your site groups in Performance Center to update a site name.
- Router (FLexVNFs Hub and Branch Appliances)
- SLA Class/Profile (Policies corresponding to FLexVNFs Hub and Branch Appliances)
- Interfaces (Virtual Network Interfaces/Sub Interfaces associated to FLexVNFs Hub and Branch Appliances)
- Tunnels (Connectivity between two Virtual Network Interfaces/Sub Interfaces associated to FLexVNFs Hub and Branch Appliances)

  NOTE
  Tunnels between branches and between branches and controllers are now supported.
- Application/SLA Paths (SLA Class/Profile associated with Tunnels)
- Versa Director and Versa Controller

  NOTE

  The Versa Director and Versa Controller inventory items are supported only for DX NetOps Spectrum.

The plug-in collects notifications of following types:

- Alarms (related to FLexVNFs Hub and Branch Appliances)
- Events (related to Alarms)

The plug-in collects the following performance metrics:

Item Types	VNA Metric Families and Metrics
Devices	• CPU – Utilization • Memory – Memory Utilization

Interfaces	• Interface – Incoming Bytes – Outgoing Bytes – Incoming Packets – Outgoing Packets – Incoming Dropped Packets – Outgoing Dropped Packets – Incoming Errors – Outgoing Errors – Interface Speed
Tunnels	• Network Interface – Jitter – Latency – Packet loss (PDU Loss Ratio)
SLA Paths	• Network Interface – Percentage of Jitter SLA Threshold – Percentage of Packet Loss SLA Threshold – Percentage of Latency SLA Threshold

Supported Releases

CA Virtual Network Assurance supports the following releases of Versa SD-WAN:

- Versa SD-WAN 16.1R2

  NOTE

  Only the Versa SD-WAN 16.1R2 version supports out-of-band IP management for the branches that this plugin discovers. Without this IP management, DX NetOps Spectrum cannot automatically poll the branches using SNMP.
- Versa SD-WAN 16.1R1

Requirements

- The Versa Director port must be open to the DX NetOps Virtual Network Assurance host.
- The Versa Analytics port must be open to the DX NetOps Virtual Network Assurance host.

Plug-in Configuration Example

The following JSON example shows the Versa SD-WAN plug-in configuration:

NOTE

You can specify the out of band management IP address for appliances in multiple ways.

If the default out of band management IP address does not work for you, and if the appliance IP addresses are also the out of band management IP address for the appliances in your environment, then setting the `SET_APPLIANCE_IP_AS_OOBM_IP` flag to "true" forces the Versa plug-in to set the appliance IP address as the out of band management IP address for the discovered appliances.

Otherwise, use the `OOBM_INTERFACE_NAME` parameter to specify a name for the interface that is used as the management interface for your appliances (for example, "eth-0/0").

```
{
     "PLUGIN_CONFIG": {
     "PROTOCOL": "https",
```

```
      "DIRECTOR_IP": "13.57.128.2",
      "DIRECTOR_PORT": 9182,
      "DIRECTOR_USER_NAME": "defaultUser",
      "DIRECTOR_PASSWORD": "defaultPasswrd",
      "ANALYTICS_PROTOCOL": "http",
      "ANALYTICS_IP": "13.57.128.2",
      "ANALYTICS_PORT": 8080,
      "ANALYTICS_USER_NAME": "defaultUser",
      "ANALYTICS_PASSWORD": "defaultPasswrd",
      "INVENTORY_POLL_RATE": "0 */10 *",
      "INVENTORY_DELTA_TIME": 600,
      "SET_APPLIANCE_IP_AS_OOBM_IP": "false",
      "OOBM_INTERFACE_NAME": "",
      "PERFORMANCE_POLL_RATE": "0 */15 *",
      "PERFORMANCE_DELTA_TIME": 900,
      "PERFORMANCE_REQUEST_COUNT": 1000,
      "AVAILABILITY_POLL_RATE": "0 */5 *",
      "AVAILABILITY_DELTA_TIME": 300,
      "NOTIFICATION_POLL_RATE": "0 */1 *",
      "NOTIFICATION_DELTA_TIME": 60,
      "MAX_NOTIFICATION_COUNT": 10000,
      "MAX_INV_THREADS": 1,
      "MAX_PERF_THREADS": 1,
      "MAX_NOTIFICATION_THREADS": 1,
      "TIMEZONE": "GMT",
      "DEVICE_FILTER_FILE_PATH": "",
      "DOMAIN_ID": 0
      }
}
```

- **DIRECTOR_IP**
  The IP address of the Versa Director server
- **DIRECTOR_PORT**
  The Versa Director port
  Default: 9182
- **DIRECTOR_USER_NAME**
  The Versa Director user name
- **DIRECTOR_PASSWORD**
  The password for the Versa Director user
- **ANALYTICS_IP**
  The IP address for the Versa Analytics server
- **ANALYTICS_PORT**
  The Versa Analytics port
- **ANALYTICS_USER_NAME**
  The Versa Analytics user name
- **ANALYTICS_PASSWORD**
  The password for the Versa Analytics user
- **PROTOCOL**
  The HTTP security scheme for Versa SD-WAN
  Default: https
- **INVENTORY_POLL_RATE**

How often the product collects inventory data

- **INVENTORY_DELTA_TIME**
  The time difference between inventory polls (in seconds)
- **SET_APPLIANCE_IP_AS_OOBM_IP**
  Whether to use the `ip-address` field value of the Versa device as the out of band management IP address of the device
- **OOBM_INTERFACE_NAME**
  The name of the out of band management interface of the devices and controllers
  Default:""
- **PERFORMANCE_POLL_RATE**
  How often the product collects performance data
- **PERFORMANCE_DELTA_TIME**
  Difference between polls (in seconds) for performance data requests
- **PERFORMANCE_REQUEST_COUNT**
  The response entry count for performance data requests
- **AVAILABILITY_POLL_RATE**
  How often the product polls the availability of the Versa Director
- **AVAILABILITY_DELTA_TIME**
  Difference between polls (in seconds) for availability data requests
- **NOTIFICATION_POLL_RATE**
  How often the product collects notifications
- **NOTIFICATION_DELTA_TIME**
  Difference between notification polls
- **MAX_NOTIFICATION_COUNT**
  The maximum number of historical active alarms to pull during the first run
- **MAX_INV_THREADS**
  The number of threads required to run the Inventory poll
  Maximum: 8
  Minimum:1
- **MAX_PERF_THREADS**
  The number of threads required to run the Performance and Availability poll
  Maximum: 8
  Minimum:1
- **MAX_NOTIFICATION_THREADS**
  The number of threads required to run the Notification poll
  Maximum: 8
  Minimum:1
- **TIMEZONE**
  The time zone of the system, which must match the Versa Director and Versa Analytics time zone

  WARNING

  If this parameter is set incorrectly, the following issues might occur:

  - Incorrect timestamps might appear on performance data.
  - Performance data might be missing.
- **DOMAIN_ID**
  DX NetOps Virtual Network Assurance assigns inventory from this plug-in to the specified domain.

Viptela

Viptela is a software-defined wide area networking (SD-WAN) solution. DX NetOps Virtual Network Assurance collects inventory and performance metrics from Viptela to support DX NetOps Performance Management and DX NetOps Spectrum SD-WAN monitoring.

The Viptela plug-in collects inventory for the following items:

- Sites

  NOTE
  Viptela sites map to site groups in Performance Center. If desired, you can manage your site groups in Performance Center to update a site name.
- vEdge router
- cEdge router (16.x and 17.x)
- vEdge interfaces
- cEdge interfaces (16.x and 17.x)

  NOTE
  cEdge Interface is available for NetOps 20.2.4 or higher versions.
  Performance stats are not supported for cEdge.
- Tunnels
- Application/SLA Paths
- Alarms and events raised on vEdge routers

  NOTE
  DX NetOps Spectrum consumes these alarms.

  For more information, see the Cisco SD-WAN documentation.
- vManage
- vBond
- vSmart

Viptela collects the following performance metrics:

Item Types	VNA Metric Families and Metrics	SNMP Metric Families	NFA Metric Families and Metrics

Devices	• CPU
	  — Utilization
	• Storage Disk Capacity
	  — Disk Utilization
	• Memory
	  — Memory Utilization
		• Availability
		• Device Unique Identifier
		• Environmental Sensor - Fan
		• Environment Sensor - Power Supply
		• Environmental Sensor - Temperature
		• ISDN
		• Reachability
		• SNMP Statistics
		• System	

Interfaces	• Interface – Incoming Bytes – Outgoing Bytes – Incoming Packets – Outgoing Packets – Incoming Dropped Packets – Outgoing Dropped Packets – Incoming Errors – Outgoing Errors – Interface Speed	• Interface	• NetFlow Statistics – Top Enterprise Hosts by Volume – Top Enterprise Protocols by Volume – Top IP Interface Utilization (Flow) – Top Flows by Volume – Interfaces Over Threshold – Routers with the Most Flow Traffic – Top Conversations – Top Hosts – Top Protocols – ToS Summary
Tunnels	• Network Interface – Jitter – Latency – Packet loss – Time in Up State – Time in Down State – Time in Unknown State – Pct Time in Up State – Pct Time in Down State – Pct Time in Unknown State – Bytes In (Gauge) – Bytes Out (Gauge – Packets In (Gauge) – Packets Out (Gauge)	• VPN Interface • IPSec Tunnel Aggregate	
Application/SLA Paths	• Network Interface – Percentage of Jitter SLA Threshold – Percentage of Packet Loss SLA Threshold – Percentage of Latency SLA Threshold	• Application Route SLA Class • Application Route Statistics	

Supported Releases

DX NetOps Virtual Network Assurance supports the following Viptela releases:

- Viptela 20.x (Only 20.2.4 and higher)
- Viptela 19.x.
- Viptela 18.4
- Viptela 18.3
- Viptela 17.2.0
- Viptela 17.1.0

Requirements

- The vManage port must be open to the DX NetOps Virtual Network Assurance host.
- For application/SLA path data, when you provision tunnels in a Viptela network, you must create policies that define the SLA classes for the different types of traffic. If no sla-class policies are defined, DX NetOps Virtual Network Assurance can discover tunnels, but cannot create application/SLA paths.
  For more information, see the Cisco SD-WAN documentation.

Plug-in Configuration Example

Configure the Viptela plug-in to access Viptela vManage with a user other than the 'admin' user. The Viptela user that is used in the plug-in configuration must be granted the 'operator' role for the Viptela plug-in to function properly.

> **NOTE**
> Viptela locks out a user after excessive unsuccessful login attempts. If the password of the plug-in user is changed in Viptela, prevent the plug-in from locking out the user from vManage by updating the Viptela plug-in configuration with the new password. If Viptela data is missing, check whether the user can log in to vManage. If the user is locked out, unlock the user account.
> For more information about how to unlock a user, see the "request aaa unlock-user" command in the Command Reference section of the Cisco SD-WAN documentation.

The following JSON example shows the Viptela plug-in configuration:

> **NOTE**
> In CA VNA 3.7.1, the `OUT_OF_BAND_MGMT_VPN_ID` property has been removed and the `DEVICE_MODEL_PATTERN_VS_OOBM_VPN_ID_PAIRS` property has been added.

```
{
"PLUGIN_CONFIG": {
      "VMANAGE_IP": "10.241.1.5",
      "VMANAGE_PORT": 8443,
      "VMANAGE_USER_NAME": "capm",
      "VMANAGE_PASSWORD": "capm",
      "PROTOCOL": "https",
       "VMANAGE_PROXY_IP": "0.0.0.0",
      "VMANAGE_PROXY_PORT": "0",
      "VMANAGE_PROXY_PROTOCOL": "https",
      "VMANAGE_PROXY_USER_NAME": "",
      "VMANAGE_PROXY_PASSWORD": "",
      "USE_PROXY": "FALSE",
       "DEVICE_MODEL_PATTERN_VS_OOBM_VPN_ID_PAIRS": "{ .*=512 }",
       "INVENTORY_POLL_RATE": "0 */10 *",
       "INVENTORY_DELTA_TIME": 600,
       "INVENTORY_REQUEST_COUNT": 5000,
      "PERFORMANCE_POLL_RATE": "0 */30 *",
      "PERFORMANCE_DELTA_TIME": 1800,
      "PERFORMANCE_REQUEST_COUNT": 1000,
       "VEDGE_PERFORMANCE_SAMPLE_INTERVAL": 300,
      "INTERFACE_PERFORMANCE_SAMPLE_INTERVAL": 300,
      "TUNNEL_PERFORMANCE_SAMPLE_INTERVAL": 300,
       "TIMEZONE": "GMT",
      "AVAILABILITY_POLL_RATE": "0 */5 *",
```

```
      "AVAILABILITY_DELTA_TIME": 300,
      "NOTIFICATION_POLL_RATE": "0 */1 *",
       "NOTIFICATION_DELTA_TIME": 60,
       "SLA_CLASS_POLL_RATE": "0 0 0",
       "SLA_CLASS_DELTA_TIME": 86400,
       "MAX_NOTIFICATION_COUNT": 10000,
       "DOMAIN_ID": 0
      }
}
```

NOTE
The default values for the Viptela plug-in configuration balance performance with efficiency. Gaps might appear in dashboards that display data collected at intervals longer than the dashboard resolution. You can change the dashboard resolution or increase the polling frequency. Increasing the polling frequency increases the load on the vManage host.

- **VMANAGE_IP**
  The IP address of the vManage host.
- **VMANAGE_PORT**
  The vManage port.
  Default: 8443
- **VMANAGE_USER_NAME**
  A vManage user with the netadmin role.
- **VMANAGE_PASSWORD**
  The password for the vManage user.
- **PROTOCOL**
  The HTTP security scheme for vManage.
  Default: https
- **VMANAGE_PROXY_IP**
  The IP address of the vManage proxy.
- **VMANAGE_PROXY_PORT**
  The port of the vManage proxy.
- **VMANAGE_PROXY_PROTOCOL**
  The HTTP security scheme for the vManage proxy.
- **VMANAGE_PROXY_USER_NAME**
  The user name of the vManage proxy.
- **VMANAGE_PROXY_PASSWORD**
  The password of the vManage proxy.
- **USE_PROXY**Whether to use the vManage proxy
  Allowed values: TRUE or FALSE
- **DEVICE_MODEL_PATTERN_VS_OOBM_VPN_ID_PAIRS**
  Represents a device model pattern and the value is the corresponding VPN id of the management interface.
  Default: ".*

  NOTE
  For example, to configure VPN Id 512 only for device models vedge-1000 and 1 for the rests, use the following property value: `"DEVICE_MODEL_PATTERN_VS_OOBM_VPN_ID_PAIRS": "{ vedge-1000=512, .*=1 }"`
- **INVENTORY_POLL_RATE**

How often the product collects inventory data.
- **INVENTORY_DELTA_TIME**
  The time difference between inventory polls (in seconds).
- **INVENTORY_REQUEST_COUNT**
  The default paging size for the Viptela inventory.
- **PERFORMANCE_POLL_RATE**
  How often the product collects performance data.
- **PERFORMANCE_DELTA_TIME**
  Difference between polls (in seconds) for performance data requests.
- **PERFORMANCE_REQUEST_COUNT**
  The response entry count for performance data requests.
- **VEDGE_PERFORMANCE_SAMPLE_INTERVAL**
  How often Viptela records vEdge device performance.
- **INTERFACE_PERFORMANCE_SAMPLE_INTERVAL**
  How often Viptela records interface performance data.
- **TUNNEL_PERFORMANCE_SAMPLE_INTERVAL**
  How often Viptela records tunnel performance data.
- **TIMEZONE**
  The time zone of the system, which must match the vManage time zone.

  WARNING
  Set this parameter correctly to prevent the following issues:

  - Incorrect timestamps might appear on performance data.
  - Performance data might be missing.
  - Alarms and events might not be collected.
- **AVAILABILITY_POLL_RATE**
  How often the product polls the availability of the controller.
- **AVAILABILITY_DELTA_TIME**
  Difference between polls (in seconds) for availability data requests.
- **NOTIFICATION_POLL_RATE**
  How often the product collects alarm and event data.
- **NOTIFICATION_DELTA_TIME**
  Difference between polls (in seconds) for alarm and event data requests.
- **SLA_CLASS_POLL_RATE**
  How often the product collects SLA class data.
- **SLA_CLASS_DELTA_TIME**
  Difference between polls (in seconds) for SLA class data requests.
- **MAX_NOTIFICATION_COUNT**
  The maximum number of historical active alarms to pull during the first run.
- **DOMAIN_ID**
  DX NetOps Virtual Network Assurance assigns inventory from this plug-in to the specified domain.

VMware vSphere

VMware vSphere is a cloud computing virtualization platform. DX NetOps Virtual Network Assurance collects inventory and performance metrics from VMware vSphere.

The plug-in collects inventory for the following items from the vCenter server:

- Data centers
- Clusters
- Hosts
- Virtual machines
- Host interfaces
- Virtual machine interfaces
- vSphere distributed virtual switches (Created by the Cisco ACI Virtual Machine Manager Domain integration with VMware vSphere)
- vSphere distributed virtual switch ports

NOTE

To see performance metrics for ESX hosts, you must enable SNMP on each host and discover the hosts in Performance Center. DX NetOps Virtual Network Assurance collects performance metrics only for Virtual Machines.

The plug-in supports the following events from the vCenter server:

- **DrsVmMigratedEvent**
  A virtual machine migration that the Distributed Resource Scheduler (DRS) recommended
- **VmMigratedEvent**
  A virtual machine migration
- **VmFailedMigrateEvent**
  A failure to migrate a virtual machine
- **VmBeingHotMigratedEvent**
  A virtual machine is hot-migrating
- **VmBeingMigratedEvent**
  A virtual machine is migrating

NOTE

The plug-in queries vCenter events at a configurable rate. The default rate is every 60 seconds.

VMware vSphere collects the following performance metrics:

Item Type	VNA Metric Families and Metrics
Hosts	• CPU and Memory – CPU – CPU Utlization – Memory Usage – Memory – Aggregated Network Incoming Bytes – Aggregated Network Outgoing Bytes – Aggregated Network Incoming Packets – Aggregated Network Outgoing Packets • Disk – Disk Read Requests Rate – Disk Write Requests Rate – Disk Read Bytes Rate – Disk Write Bytes Rates – Disk Capacity – Disk Latency – Disk Allocation – Disk IOPS – Disk Usage
Virtual Machines	• CPU and Memory – CPU – CPU Utlization – Memory Usage – Memory – Aggregated Network Incoming Bytes – Aggregated Network Outgoing Bytes – Aggregated Network Incoming Packets – Aggregated Network Outgoing Packets • Disk – Disk Read Requests Rate – Disk Write Requests Rate – Disk Read Bytes Rate – Disk Write Bytes Rates – Disk Capacity – Disk Latency – Disk Allocation – Disk IOPS – Disk Usage
DVS Ports	• Interface – Incoming Bytes – Outgoing Bytes – Incoming Packets – Outgoing Packets – Incoming Dropped Packets – Outgoing Dropped Packets – Incoming Errors – Outgoing Errors

Virtual Machine Metrics and Host Metrics

The plug-in collects the following metrics for virtual machines and hosts:

NOTE

The default statistics collection level is level 1. From the vSphere Web Client, manage your settings to increase the statistics collection level to level 3 for all the necessary metrics. Disable weekly, monthly, and yearly performance data saving to reduce the required disk space. For more information, see the VMware vSphere documentation.

Statistics collection level	Supported Metrics
Level 1	CPU Time for Virtual Machine and Host
Level 1	Memory for Virtual Machine and Host
Level 1	Disk Usage for Virtual Machine and Host
Level 2	CPU Time for Virtual Machine and Host
Level 2	Memory for Virtual Machine and Host
Level 2	Disk Usage for Virtual Machine and Host
Level 2	Disk Read for Virtual Machine and Host
Level 2	Disk Write for Virtual Machine and Host
Level 3	Disk capacity for Virtual Machine and Host
Level 3	Disk allocation for Virtual Machine and Host
Level 3	Disk usage for Virtual Machine and Host
Level 3	Disk latency for Virtual Machine and Host
Level 3	Disk IOPS for Virtual Machine and Host
Level 3	Disk read request rate for Virtual Machine and Host
Level 3	Disk write request rate for Virtual Machine and Host
Level 3	Disk read bytes rate for Virtual Machine and Host
Level 3	Disk write bytes rate for Virtual Machine and Host
Level 3	CPU time for Virtual Machine and Host
Level 3	CPU utilization for Virtual Machine and Host
Level 3	Memory for Virtual Machine and Host
Level 3	Memory utilization for Virtual Machine and Host

The plug-in calculates the following metrics:

- Virtual machines per hypervisor

Distributed Virtual Switch Port Metrics

The plug-in collects the following metrics for distributed virtual switch ports:

Statistics collection level	Supported Metrics
Level 2	Incoming bytes on virtual interfaces, Hosts, and Distributed Virtual Switch Port
Level 2	Outgoing bytes on virtual interfaces, Hosts, and Distributed Virtual Switch Port

Level 2	Incoming packets on virtual interfaces, Hosts, and Distributed Virtual Switch Port
Level 2	Outgoing packets on virtual interfaces, Hosts, and Distributed Virtual Switch Port
Level 2	Incoming Dropped Packets for Distributed Virtual Switch Port
Level 2	Outgoing Dropped Packets for Distributed Virtual Switch Port
Level 2	Incoming Errors for Distributed Virtual Switch Port
Level 2	Outgoing Errors for Distributed Virtual Switch Port
Level 3	Incoming bytes on virtual interfaces, Hosts, and Distributed Virtual Switch Port
Level 3	Outgoing bytes on virtual interfaces, Hosts, and Distributed Virtual Switch Port
Level 3	Incoming packets on virtual interfaces, Hosts, and Distributed Virtual Switch Port
Level 3	Outgoing packets on virtual interfaces, Hosts, and Distributed Virtual Switch Port
Level 3	Incoming Dropped Packets for Distributed Virtual Switch Port
Level 3	Outgoing Dropped Packets for Distributed Virtual Switch Port
Level 3	Incoming Errors for Distributed Virtual Switch Port
Level 3	Outgoing Errors for Distributed Virtual Switch Port

Supported Releases

CA Virtual Network Assurance (VNA) supports the following releases of VMWare vSphere:

- VMware vSphere 6.0.
- VMware vSphere 6.5.

Requirements

- vSphere plug-in and ACI plug-in must be configured using the same `DOMAIN_ID`.
- vSphere plug-in requires read-only access to vCenter Server running VMware Infrastructure SDK Web Service.
- The URL used by the plug-in for accessing the VMware Infrastructure SDK Web Service is: `https://<vCenter_IP_Address>/sdk/vimService.wsdl`
- If you have multiple vSphere vCenter environments, configure each vCenter to a different domain in DX NetOps Virtual Network Assurance.

Plug-in Configuration Example

The following JSON example shows the VMWare vSphere plug-in configuration:

```
{
"PLUGIN_CONFIG": {
     "VCENTER_IP": "10.241.16.37",
     "VCENTER_USER_NAME": "vnaUser@vsphere.local",
     "VCENTER_PASSWORD": "Password@1"
     "PROTOCOL": "https",
     "INVENTORY_POLL_RATE": "0 */10 *",
     "INVENTORY_DELTA_TIME": 600,
```

```
      "AVAILABILITY_POLL_RATE": "O */5 *",
      "AVAILABILITY_DELTA_TIME": 300,
      "PERFORMANCE_POLL_RATE": "0 */10 *",
      "PERFORMANCE_DELTA_TIME": 600,
      "MAX_PERF_THREADS": 3,
      "NOTIFICATION_POLL_RATE": "0 */1 *",
      "DOMAIN_ID": 0
      }
}
```

20.2.2 and Higher Only:

```
{
"PLUGIN_CONFIG": {
      "VCENTER_IP": "10.241.16.37",
      "VCENTER_USER_NAME": "vnaUser@vsphere.local",
      "VCENTER_PASSWORD": "Password@1"
      "PROTOCOL": "https",
      "INVENTORY_POLL_RATE": "0 */10 *",
      "INVENTORY_DELTA_TIME": 600,
      "AVAILABILITY_POLL_RATE": "O */5 *",
      "AVAILABILITY_DELTA_TIME": 300,
      "PERFORMANCE_POLL_RATE": "0 */10 *",
      "PERFORMANCE_DELTA_TIME": 600,
      "MAX_PERF_THREADS": 3,
      "ENABLE_DVS_PERF    ": "true,",
      "NOTIFICATION_POLL_RATE": "0 */1 *",
      "DOMAIN_ID": 0
      }
}
```

- **VCENTER_IP**
  The IP address of the vCenter server
- **VCENTER_USER_NAME**
  The vCenter Server user name
- **VCENTER_USER_NAME**
  The vCenter server password
- **PROTOCOL**
  The communication protocol with the vCenter server
  Values: http or https (case-sensitive)
- **INVENTORY_POLL_RATE**
  How often the product collects inventory data
- **INVENTORY_DELTA_TIME**
  Difference between polls (in seconds)
- **AVAILABILITY_POLL_RATE**
  How often the product collects availability data
- **AVAILABILITY_DELTA_TIME**

The time difference between availability polls (in seconds)
- **PERFORMANCE_POLL_RATE**
  How often the product collects performance data
- **PERFORMANCE_DELTA_TIME**
  The time difference between performance polls (in seconds)
- **MAX_PERF_THREADS**
  Denotes the number of threads required to run the performance poll
  Max Value: 3
  Minimum Value: 1
- **ENABLE_DVS_PERF** (20.2 and Higher Only)
  Whether to enable performance metrics for DVS ports
  Values: true or false
- **NOTIFICATION_POLL_RATE**
  How often the product collects alarm and event data
- **DOMAIN_ID**
  CA Virtual Network Assurance assigns inventory from this plug-in to the specified domain

Using

DX NetOps Virtual Network Assurance collects SDN and NFV data and provides that information to other products for analysis.

DX NetOps Performance Management

DX NetOps Performance Management provides inventory and performance data for VMs, VNFs, service chains, supporting physical hardware, and vSwitches. For more information, see the documentation.

DX NetOps Spectrum

DX NetOps Spectrum associates the overlay topology from the virtual network with the physical infrastructure of the underlay. For more information, see the documentation.

Manage Domain Groups

To organize inventory that is collected from various plug-ins, create domain groups. Each plug-in assigns inventory to a single domain. Downstream consumers of DX NetOps Virtual Network Assurance data use these domains to organize inventory into meaningful groups for reporting.

To manage domains, go to user interface for the DX NetOps Virtual Network Assurance API at the following address:
`http://gateway_host:8080/vna/`

View the List of Domains

You can view the list of domains and domain IDs.

Follow these steps:

1. Go to the following URL:
   `http://gateway_host:8080/vna/`
2. Show the **Inventory**.
3. Select the following GET endpoint:
   `/v1/inventory/userdomains`

4. Click **Try it out!**
   View the response body.

Create a Domain

Follow these steps:

1. Go to the following URL:
   `http://gateway_host:8080/vna/`
2. Show the **Inventory**.
3. Select the following POST endpoint:
   `/v1/inventory/userdomains/{domainName}`

4. Specify the **domainName**, and click **Try it out!**
   DX NetOps Virtual Network Assurance creates the domain. View the response body to find the domain ID, which is required for plug-in configuration.

Update a Domain

Follow these steps:

1. Go to the following URL:
   `http://gateway_host:8080/vna/`
2. Show the **Inventory**.
3. Select the following PUT endpoint:
   `/v1/inventory/userdomains/{id}/{newDomainName}`

4. Specify the **id** and **newDomainName**, and click **Try it out!**
   DX NetOps Virtual Network Assurance updates the display name for the domain.

Administrating

DX NetOps Virtual Network Assurance does not require user direct user administration. Administer users in the products that consume and display DX NetOps Virtual Network Assurance data, such as DX NetOps Performance Management and DX NetOps Spectrum.

This section includes information about other administrative functions, such as system monitoring and environment configuration.

Update VNA Console Login Credentials

You can update the VNA console login credentials.

Follow these steps:

1. In the REST client interface, type the following URL in the URL field:
   `http://VNA_IP:8080/vna/rest/v1/auth/user`
2. Select **Put** for the method.
3. Select the following headers:
   `"Accept:text/plain", "Content-Type,application/json"`
4. Customize the following parameters in the Body text field:

```
{
 "EXISTING_USER_NAME": "admin",
```

```
 "EXISTING_PASSWORD": "admin",
 "NEW_USER_NAME": "admin1",
 "NEW_PASSWORD": "admin1",
 "CONFIRM_NEW_PASSWORD": "admin1"
}
```

5. Click **Send** to run the method

Update a Forgotten VNA Console Password

If you forget the VNA console password, you can update it.

Follow these steps:

1. Generate a cryptographic SHA-256 hash of the new password text.
2. Convert the hash to HEX encode and update it in the VNA database with the following SQL statement:

```
update vnaconfig set configValue="XXXXXXXXXXXXXXXX" where configKey="SWAGGER_UI_PASSWORD";
```

Back Up and Restore

Back up the DX NetOps Virtual Network Assurance database before an upgrade. To avoid losing valuable data, back up the DX NetOps Virtual Network Assurance database regularly. You can restore an existing backup of the DX NetOps Virtual Network Assurance database.

Back Up the DX NetOps Virtual Network Assurance Database

Create a backup archive of the current database anytime that you plan to reinstall or upgrade the software. We recommend that you create a backup on a regular basis.

Follow these steps:

1. Log in to the DX NetOps Virtual Network Assurance host as 'root', or use the 'sudo' account that you configured for the installation.
2. Back up the database archive to a specified directory using the following command:

```
/opt/CA/VNA/tools/bin/db_backup.sh -
p MySQL_root_password
backupDir/backup_filename
```

   Example:

```
/opt/CA/VNA/tools/bin/db_backup.sh -p admin /tmp/vna_db.sql
```

   Use any secure location with sufficient space for the backup directory.
3. Copy the following files to the backup directory:

```
cp /opt/CA/VNA/data/ID_CACHE.dat backupDir/ID_CACHE.dat
cp /opt/CA/VNA/data/REMOTE_VNA_CACHE.dat backupDir/REMOTE_VNA_CACHE.dat
```

Restore the DX NetOps Virtual Network Assurance Database

You can restore an existing backup of the DX NetOps Virtual Network Assurance database.

Follow these steps:

1. Log in to the DX NetOps Virtual Network Assurance host as 'root', or use the 'sudo' account that you configured for the installation.
2. Stop the application server:

   ```
   service wildfly stop
   ```

3. Restore the database archive from a specified directory using the following command:

   ```
   /opt/CA/VNA/tools/bin/db_restore.sh -
   p MySQL_root_password
   backupDir/backup_filename
   ```

   Example:

   ```
   /opt/CA/VNA/tools/bin/db_restore.sh -p admin /tmp/vna_db.sql
   ```

4. Restore the following files:

   ```
   cp backupDir/ID_CACHE.dat /opt/CA/VNA/data
   cp backupDir/REMOTE_VNA_CACHE.dat /opt/CA/VNA/data
   ```

5. Start the application server:

   ```
   service wildfly start
   ```

Configure Multi-Tenancy for CA Spectrum

If you want to aggregate data from multiple DX NetOps Virtual Network Assurance (VNA) hosts for DX NetOps Spectrum 10.2.3 and higher, you can configure and deploy a multi-tenant environment. In this environment, a VNA Aggregator aggregates data from multiple remote VNA hosts and sends the data to DX NetOps Spectrum with unique entity IDs.

Requirements

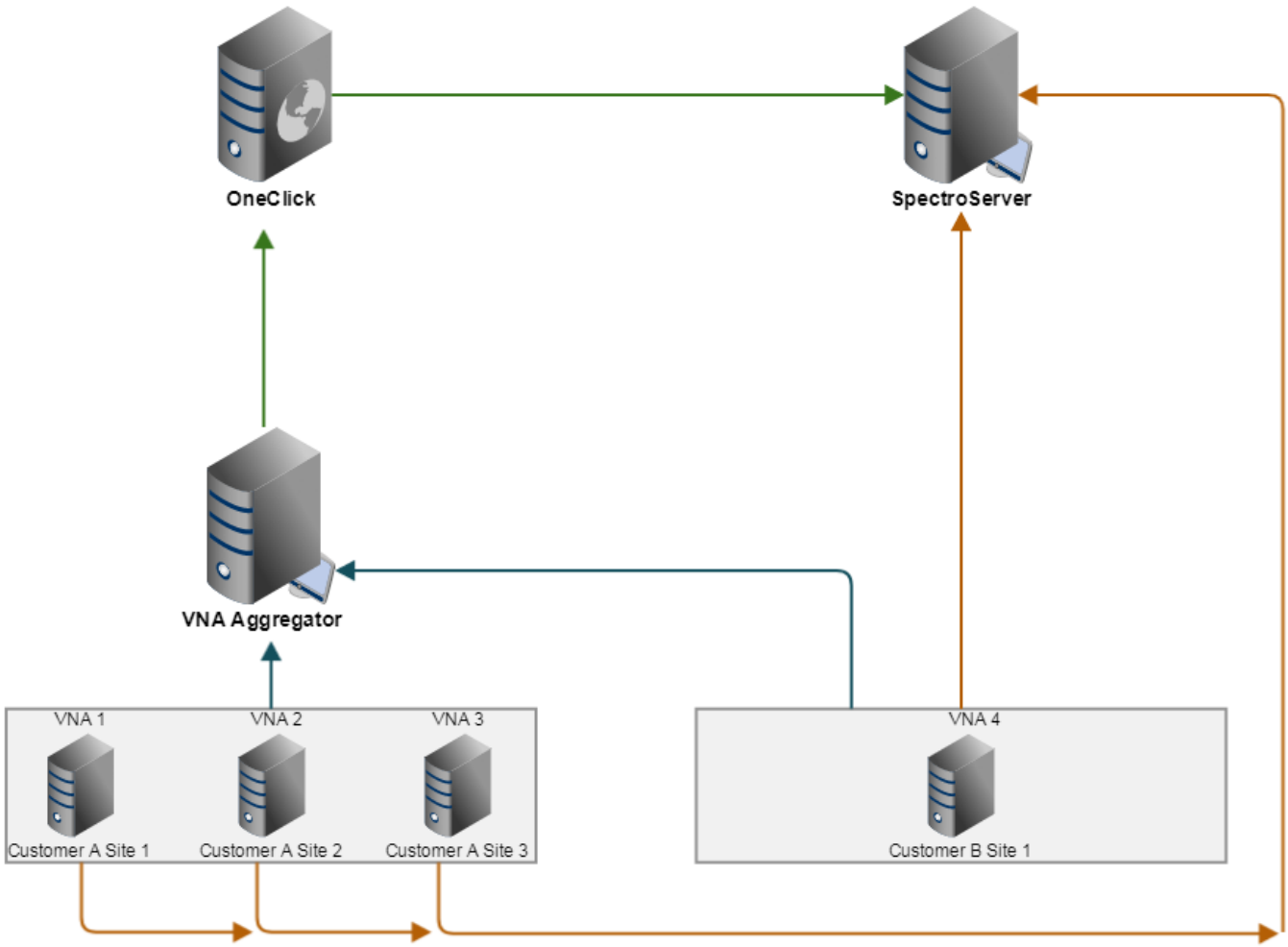
The following requirements apply to a multi-tenant environment:

- The domain name for each VNA host in your multi-tenant environment must be unique. Follow a naming convention (for example, `customer_name_site)` .
- We strongly recommend that the VNA Aggregator be fully dedicated to aggregation. Do not install other plug-ins on this host.
- The VNA Aggregator and OneClick should reside on the same server.
- Do not use or reuse an existing standalone VNA instance as the VNA Aggregator. Always use a freshly installed VNA instance as the VNA Aggregator.

Multi-tenant Architecture

The following diagram shows the system architecture of a multi-tenant environment:

Figure 71: Multi-tenant Architecture



Deploy multi-tenancy

For DX NetOps Spectrum to consume data from multiple VNA hosts, deploy multi-tenancy. Then DX NetOps Spectrum can segregate data on a per site basis for each customer and customers can have multiple sites.

Follow these steps:

1. Navigate to the DX NetOps Virtual Network Assurance API:
   `http://gateway_host:8080/vna/`
2. Expand **Aggregate**.
3. Specify the following GET call:
   `/v1/aggregate/vna`

4. Use a REST client to POST the following JSON parameters for each VNA host:

```
[
      {
            "VNA_HOST": "string",
            "VNA_PORT": 0
      }
]
```

Example:

```
[
      {
            "VNA_HOST": "1.2.3.4",
            "VNA_PORT": 8080
      },
      {
            "VNA_HOST": "5.6.7.8",
            "VNA_PORT": 8080
}
]
```

The connection status changes to "INITIALIZED"

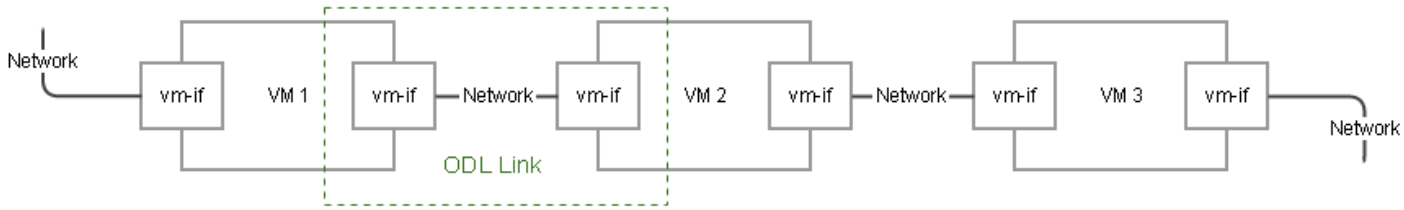
5. Integrate your VNA Aggregator with DX NetOps Spectrum. For more information, see the CA Spectrum documentation.
   When DX NetOps Spectrum connects to receive updates, the VNA Aggregator starts the client for each remote VNA host.
   The status of each remote VNA host changes to "CONNECTED"

Configure The OpenDaylight Topology to Support Service Chain Monitoring

To collect service chain information from the topology endpoint in OpenDaylight (ODL), DX NetOps Virtual Network Assurance requires the ODL topology to be manually configured. The service chain represents VMs connected through the network by virtual interfaces. In ODL, the topology defines the connections between VMs as links. Each node in the topology is a VM and each node termination point is an interface on the VM.

The following diagram shows an example of a flattened topology:

Figure 72: Required ODL Topology



TIP

The ODL topology always has one more link than you have VMs in the service chain.

The following ODL topology creates the previous example:

```
{
```

```
   "network-topology": {
     "topology": [
       {
         "topology-id": "topology-
name",
         "link": [
           {
             "link-id": "c75e6807-d203-4a80-90ed-641292abc2ad",
             "destination": {
               "dest-node": "87565ada-2f75-4d8f-a47a-655c20a01967",
               "dest-tp": "3ba3ae06-e84f-4e01-b67b-b43402a19d86"
             },
             "source": {
               "source-node": "960ac31e-e075-48b1-b35c-7e51a8996a98",
               "source-tp": "6243261b-cdb7-4126-8bc2-562e14719d0f"
             }
           },
           {
             "link-id": "165e5740-31d2-4eaf-ac8b-6e5cf8f3dd45",
             "destination": {
               "dest-node": "RIGHT",
               "dest-tp":
 "RIGHT"
             },
             "source": {
               "source-node": "87565ada-2f75-4d8f-a47a-655c20a01967",
               "source-tp": "2597e994-7266-4e66-97fc-b15a578f5edf"
             }
           },
           {
             "link-id": "bfb7bc3a-d756-40f5-8ae1-b595f77f6b32",
             "destination": {
               "dest-node": "87565ada-2f75-4d8f-a47a-655c20a01967",
               "dest-tp": "78eefad4-5e60-4b67-95ed-cd15bca0edbc"
             },
             "source": {
               "source-node": "LEFT",
               "source-tp":
 "LEFT"
             }
           },
           {
             "link-id": "7040f440-eb17-4a51-a0c3-e6182b9302dc",
             "destination": {
               "dest-node": "960ac31e-e075-48b1-b35c-7e51a8996a98",
               "dest-tp": "c0fff608-14b7-4fdd-9237-53a202f7079a"
```

```
            },
            "source": {
              "source-node": "87565ada-2f75-4d8f-a47a-655c20a01967",
              "source-tp": "edc05766-9480-44b5-8e85-ed28fd14cb3e"
            }
          }
        ],

  "node":
          {
            "node-id": "960ac31e-e075-48b1-b35c-7e51a8996a98",
            "termination-point": [
              {
                "tp-id": "6243261b-cdb7-4126-8bc2-562e14719d0f"
              },
              {
                "tp-id": "c0fff608-14b7-4fdd-9237-53a202f7079a"
              }
            ]
          },
          {
            "node-id": "87565ada-2f75-4d8f-a47a-655c20a01967",
            "termination-point": [
              {
                "tp-id": "3ba3ae06-e84f-4e01-b67b-b43402a19d86"
              },
              {
                "tp-id": "2597e994-7266-4e66-97fc-b15a578f5edf"
              }
            ]
          }
        ]
      }
    ]
  }
}
```

- **"topology-id": "*topology-name*"**The unique name of the service chain in the deployment
- **"link-id":**
  The ID of the OpenStack network that ties two VMs together
- **"dest-node":**
  The OpenStack ID of the VM on the left side of the link
- **"dest-tp":**
  The OpenStack interface ID on the VM that is connected to the network
- **"source-node":**
  The OpenStack ID of the VM on the right side of the link
- **"source-tp":**

The OpenStack interface ID on the VM that is connected to the network

- **"dest-node": "RIGHT"**
  The last network in the chain must have a dest-node and dest-tp set to one of the following values:
  – "RIGHT"
  – "" (empty string)
- **"source-node": "LEFT"**
  The first network in the chain must have a dest-node and dest-tp set to one of the following values:
  – "LEFT"
  – "" (empty string)
- **"node":**
  A list of the nodes IDs and terminations points that are used in the topology

Disaster Recovery

If a large-scale disaster occurs, the disaster recovery plan for DX NetOps Virtual Network Assurance (VNA) enables a switchover to a recovery VNA. The disaster recovery plan involves provisioning a secondary VNA as a recovery VNA and regularly transferring data from the primary VNA.

Configure Disaster Recovery

To configure disaster recovery, install a recovery VNA. To help prevent and minimize data loss in the event of a disaster, back up the primary VNA regularly.

Follow these steps:

1. Install a recovery VNA.
2. Back up the primary VNA regularly.

   NOTE

   Back up the primary VNA to a location that the recovery VNA can access, or copy the backup to the recovery VNA.

Recover from a Disaster

To recover from a disaster, restore the most recent backup files to the recovery VNA and update the Gateway connection information in Performance Center.

Follow these steps:

1. Restore the latest backup files to the recovery VNA.
2. In Performance Center, hover over **Administration**, and click **Monitored Items Management: VNA Gateways**.
3. Select the VNA Gateway and click **Edit**.
4. Specify the recovery VNA host and port, and select the Data Collector that receives VNA data.
   Default Port: 8080
5. Set **Administrative Status** to **Up**, and click **Save**.
   If the Administrative Status is Up, the Data Collector registers VNA to receive inventory and performance data.

Monitor CA Virtual Network Assurance Broker Performance

The DX NetOps Virtual Network Assurance broker is supported for instrumentation with CA Application Performance Management (CA APM). CA APM receives performance metrics about the DX NetOps Virtual Network Assurance service and host.

To verify the supported releases of CA APM, look at the product support matrix for DX NetOps Virtual Network Assurance.

Follow these steps:

1. Edit the following file:
   `/opt/CA/VNA/apm/wily/core/config/IntroscopeAgent.VNA.profile`
2. Add the following lines to the end of the file:
   `introscope.agent.enterprisemanager.transport.tcp.host.DEFAULT=`***`hostname`***
   `introscope.agent.enterprisemanager.transport.tcp.port.DEFAULT=`***`port`***
   - ***hostname*** is the hostname of the CA APM server.
   - ***port*** is the CA APM server port.
     Default: 5001
3. Edit the following file: `/etc/default/wildfly.conf`
4. Add the WILY_HOME environment variable to point to the APM agent installation directory on the DX NetOps Virtual Network Assurance host to the end of the file.**Example:**
   ```
   ...
   export WILY_HOME=/opt/CA/VNA/apm/wily
   ```
5. Restart the wildfly service:
   `sudo service wildfly restart`

Troubleshooting

Use the troubleshooting section to diagnose and resolve issues with DX NetOps Virtual Network Assurance.

Installation Stopped Abruptly

In DX NetOps Virtual Network Assurance 3.6.6, when the installation stops abruptly for any reason, and you start the installation again, the installer does not prompt you to install the product on a custom directory. Perform the following steps to resolve the issue and install the product on a custom directory.

Follow these steps:

1. On the server where the installation stopped, navigate to the `/etc` directory.
2. Delete the VNA.cfg file using the `rm -rf VNA.cfg` command.
3. Follow the steps in the Installing section and complete the installation.

Insufficient Permissions for Install

Symptom:

The following error message appears:

```
Error: Insufficient permissions for installation path: /u01/opt/CA
Path must have executable permission for 'other'.
Exiting the installer...
```

Solution:

Run the following command to ensure that all child and parent directories have the executable permission for 'other'.

```
chmod 755 "Folder_Name"
```

Java Process Suddenly Exits

Symptom:

- The DX NetOps Virtual Network Assurance Java process suddenly exits.
- Root cause unidentified in the VNA log files.
- A file with name hs_err_pidXXXX.log (where XXXX is the VNA Java process PID) is created in the VNA root folder with following text as part of the content:

```
# There is insufficient memory for the Java Runtime Environment to continue.
# Native memory allocation (mmap) failed to map YYYY bytes for committing reserved memory.
```

Solution:

- Increase physical memory or swap space. Check whether the swap backing store is full.
- Decrease Java heap size (-Xmx/-Xms).

Configure DX NetOps Virtual Network Assurance Logging

By default, VNA logging is enabled and set to INFO on the Data Aggregator and Data Collector.

Follow these steps:

1. Go to the following location:
   VNA_host:9990
2. Enter the administrator credentials.
3. Go to **Configuration**, **Subsystems**, **Logging**, and **LOG CATEGORIES**.
4. Edit the logging configurations as desired.

Useful Log Categories

You might find the following log categories useful for diagnosing and resolving issues with DX NetOps Virtual Network Assurance:

- **INVENTORY_SERVICE**Logs inventory updates that are sent from DX NetOps Virtual Network Assurance to DX NetOps Spectrum or DX NetOps Performance Management
  Recommended Log Level: TRACE
  Location: /opt/CA/VNA/wildfly/standalone/log/gateway.log
- **CLIENT_UPDATES**
  Logs client updates that are sent from DX NetOps Virtual Network Assurance to DX NetOps Spectrum or DX NetOps Performance Management
  Recommended Log Level: TRACE
  Location: /opt/CA/VNA/wildfly/standalone/log/gateway.log
- **REST_API = TRACE**
  Logs REST API calls made to DX NetOps Virtual Network Assurance and includes responses that are sent to the caller
  Recommended Log Level: TRACE
  Location: /opt/CA/VNA/wildfly/standalone/log/gateway.log
- **AGGREGATION_SERVICE**Logs messages from the VNA Aggregator**Recommended Log Level:** DEBUG (for a summary of data coming from remote VNAs) or TRACE (for the entire response log coming from remote VNAs)**Location:** /opt/CA/VNA/wildfly/standalone/log/gateway.log
- **OC_ACQUISITION**
  Logs the dump of raw data that is received from the target SDN devices
  Recommended Log Level: DEBUG
  Location: /opt/CA/VNA/collector/*Engine_UUID*/tmp/poll_*
- **SOUTHBOUND_UPDATES**
  Logs the dump of polled data that is sent from the Open Collector to the Broker **Recommended Log Level:** TRACE
  Location: /opt/CA/VNA/debug

View DX NetOps Virtual Network Assurance Logging

VNA logs are available on the Data Aggregator and Data Collector in the following locations:

- *DA_Install_Directory*/apache-karaf-*version*/data/log/sdn.log
- *DC_Install_Directory*/apache-karaf-*version*/data/log/sdn.log

Too Many Open Files Exception

Symptom:

DX NetOps Virtual Network Assurance is polling a larger number of Open vSwitch hosts with the Open vSwitch plug-in.

Exceptions indicating too many open files appear in the following log:

`/opt/CA/VNA/wildfly/standalone/log/gateway.log`

Solution:

Configure the limit on the number of open files. Verify that the `wildfly` user has a limit of at least 65536 on the number of open files. Set this value permanently.

Follow these steps:

1. Log in as the `wildfly` user on the VNA host.
2. Change the `ulimit` for the open files limit to at least 65536:

   ```
   ulimit -n ulimit_number
   ```

   Example:

   ```
   ulimit -n 65536
   ```

3. Open the following file:
   `/etc/security/limits.conf`
4. Add the following lines:

   ```
   # Added by VNA

   * soft nofile 65536

   # Added by VNA

   * hard nofile 65536
   ```

   NOTE

   Restart VNA for these changes to take effect. If you are upgrading, the upgrade process automatically restarts VNA.
5. Verify that the number of open files is set properly:

   ```
   ulimit -n
   ```

   The command returns the limit that you specified earlier.

Product References and Abbreviations

List only CA product names with a shortened version (such as "CA Service Operations Insight (CA SOI)") and abbreviations/acronyms (such as "virtual machine (VM)") that you use in your wiki space. Sort the list alphabetically. Place this page at the bottom of your TOC so that it appears just above the Announcements & News link.

This documentation references the following products and abbreviations:

- CA Application Performance Management
- DX NetOps Performance Management (CA PM)
- DX NetOps Spectrum®
- DX NetOps Virtual Network Assurance (CA VNA)
- Network Functions Virtualization (NFV)
- Software-Defined Network (SDN)
- Virtual Network Functions (VNF)

Product Accessibility Features

CA Technologies is committed to addressing user accessibility in the development of its products and documentation to help all customers, regardless of ability, accomplish vital business tasks.

Documentation Legal Notice

Fault Monitoring

DX NetOps Spectrum is a services and infrastructure management system that monitors the state of managed elements including the following:

- Devices
- Applications
- Host systems
- Connections

Status information such as fault and performance data from these elements is collected and stored. DX NetOps Spectrum constantly analyzes this information to track conditions within the computing infrastructure. If an abnormal condition is detected, the product isolates it, alerts you, and presents the possible causes and solutions.

This section contains everything you need for fault monitoring from getting started to troubleshooting information.

Release Information

DX NetOps Spectrum is one of the most comprehensible extensible portfolios in the industry to manage the underlying IT infrastructure. DX NetOps Spectrum understands the flow of data traffic and use of it to the particular network domain by dramatically simplifying IT management by linking applications, services, and transactions to the underlying infrastructure. For its infrastructure management, DX NetOps Spectrum relies on the area of discovery or infrastructure discovery that is combined with proactive performance monitoring and root cause to yield detailed traffic analysis, flow, and performance.

> **NOTE**
>
> For more information on key features for each DX NetOps Spectrum release, see the Release Comparison table.

Release Notes and upgrade articles prepare you for the new and enhanced features of this release. Understand how the product is compatible with servers, clients, and other software. Learn how to upgrade to the current release and implement the new features:

Upgrading

Features and Enhancements 10.4.2.2

The DX NetOps Spectrum 10.4.2.2 version is the same as DX NetOps 20.2.5 release.

This release includes the following features and enhancements:

OneClick WebApp Enhancements

This release includes the following OneClick WebApp-related enhancements:

- OneClick WebApp Audio Alarm
- OneClick WebApp Security

OneClick WebApp Audio Alarm

From the current release, an audio message announces the new alarm in WebApp.

For more information, see Alarms Tab Preferences.

OneClick WebApp Security

The OneClick WebApp administrator can use the **Admin** console to view all the open WebApp sessions from multiple machines. When you try to launch an open session, you can see a warning message `Session is mirrored by OneClick WebApp Administrator.`

For more information, see OneClick WebApp.

NCM Enhancements

This release includes the following NCM-related enhancements:

- Task Work Queue Size Increased
- Syslog Monitoring Enhancement

Task Work Queue Size Increased

The Task Work Queue Size has been increased to parallelly process a maximum of 200 devices on each DX NetOps Spectrum host.

For more information, see Network Configuration Manager Configurations.

Syslog Monitoring Enhancement

From the current release, when the 'preventDNSlookup' attribute is enabled, instead of resolving the hostname to IP address, DX NetOps Spectrum fetches the model handles of the models from DX NetOps Spectrum Database that has the model name same as hostname received in the Syslog message and asserting Syslog trap on those models. If there are multiple models whose model name matches the hostname from the Syslog message, the trap is asserted on all the matched models.

For more information, see Configuring CA Spectrum to Process Syslog File Matches.

DX NetOps Spectrum Integration Enhancements

This section lists the enhancements to the integration of DX NetOps Spectrum with various other products.

- Integration with DX NetOps Virtual Network Assurance (DX NetOps VNA)

Integration with DX NetOps Virtual Network Assurance (DX NetOps VNA)

This release includes the following enhancement to DX NetOps Spectrum- DX NetOps VNA integration:

- Disable SNMP Modeling
- Support for cEdge Devices
- Scalability and Performance Improvements
- Dump OCS DX VNA Inventory Cache

Disable SNMP Modeling

From the current release, you can disable SNMP model devices from SDN Manager. By default, **Disable SNMP Modeling** option is set to **NO**. If you Enable VNA integration with **Disable SNMP Modeling** option as NO, all the devices which are SNMP reachable get modeled as SNMP model. When DX VNA integration is disabled all DX VNA models except SNMP are deleted.

For more information, see Monitoring SDN Devices.

Support for cEdge Devices

From the current release, cEdge devices are supported.

For more information, see Monitoring SD-WAN for Viptela.

Scalability and Performance Improvements

Prior to the current release, SLA Path used to have the "Topology tab" which displayed the Branch routers connected through Transports. From the current release, the same connections are shown under the SDN Overlay tab. SLA Path models are not displayed under the SDN Manager hierarchy. You can use the locater search to find SLA Path.

For more information, see Monitoring SD-WAN for Versa and Monitoring SD-WAN for Viptela.

Dump OCS DX VNA Inventory Cache

From the current release, inventory cache which maps DX VNA entities with DX NetOps Spectrum model handles, will be dumped in the connected OneClick server. A CSV file is generated under tomcat logs.

For more information, see Monitoring SDN Devices.

Integration Compatibility Matrix

For more information about the integration of DX NetOps Spectrum with other products, see Integration Compatibility.

Platform Support

There are no enhancements to platforms supported in the previous release.

Deprecated Support

From the current release, the following components are not supported:

- CA eHealth
- CA eHealth-CAC

Features and Enhancements 10.4.2.1

This release includes the following features and enhancements:

Processd Enhancement

From the current release, Tomcat and Webtomcat are stopped when processed is stopped.

For more information, see **Process Daemon (processd)** section on Setting Up a Distributed SpectroSERVER Environment page.

NCM Enhancements

This release includes the following NCM-related enhancements:

Permission to Limit NCM to Update One Device at a Time

From the current release, when the privileged role of the user Allow Load Firmware on Multiple Devices is disabled and load firmware task is attempted on more than one device, an error popup should be thrown with the message Restricted Load firmware for a single device, Select the single device to load firmware. when the privileged role of the user **Allow Load Firmware on Multiple Devices** is disabled and load firmware task is attempted on GC with more than one device,

an error popup should be thrown with the message Restricted Load firmware for a single device, maintain single device per GC to load firmware.

For more information, see Network Configuration Manager.

Task Work Queue Size Increased

The Task Work Queue Size has been increased to parallelly process a maximum of 100 devices on each DX NetOps Spectrum host.

For more information, see Network Configuration Manager Configurations.

Event Handling in FT Scenario

From the current release, when DX NetOps Spectrum is set up in FT mode, and the primary host fails, DX NetOps Spectrum runs on the secondary host. In such a situation only events are pulled from DX Performance Management to DX NetOps Spectrum. However, inventory sync is not supported.

For more information, see How to Configure Events for Integration with DX Performance Management.

Encoded/ Plain User Password Using REST API

In the current release, you can encode the user password using the REST API or pass the password as plain text.

For more information, see Alarm Fields, REST Examples, and Attribute Mapping.

Scale SNMPv3 Profile Performance

Active SNMPv3 profiles are available in remote SNMPv3 profiles cache. Invalid and incorrect SNMPv3 profiles are removed from remote SNMPv3 profiles cache to improve the performance of SpectroSERVER.

DX NetOps Spectrum Process Initialization Changed from init to systemctl

From the current release, DX NetOps Spectrum processes are registered with `systemctl` instead of `/etc/init.d`. The use of systemctl helps to reduce the system and application boot time by parallelly booting the processes.

Support to Calculate Threshold Limits for Time-based Results

Specifies the criteria to calculate the threshold. You can specify the minimum, average, or maximum as a threshold limit for Cisco devices and average for non-Cisco devices.

For more information, see Specify Alarm Thresholds for a Test.

OneClick WebApp URLs Included in Alarm Notifier Mail

From the current release, the alarm notification email includes the OneClick WebApp URL.

For more information, see Main Toolbar.

Purge Tomcat and Webtomcat Log Files

From the current release, when the Alarm logs directory exceeds the set threshold, an alarm is displayed with the threshold value. Irrespective of the file size when the threshold value set for `Purge log files older than` is reached, the old log files are deleted.

For more information, see **Purge Tomcat and Webtomcat Log Files** on the OneClick Administration Page.

DX NetOps Spectrum Integration Enhancements

This section lists the enhancements to the integration of DX NetOps Spectrum with various other products.

- Integration with DX NetOps Virtual Network Assurance (DX NetOps VNA)
- Integration with DX Operational Intelligence (DX OI)
- Integration with DX APM

Integration with DX NetOps Virtual Network Assurance (DX NetOps VNA)

This release includes the following enhancement to DX NetOps Spectrum- DX NetOps VNA integration:

You can search the DX VNA models from the **Locator** search.
For more information, see Integrating with DX NetOps Virtual Network Assurance.

Integration with DX Operational Intelligence (DX OI)

This release includes the following enhancement:

- Launch OneClick Console or WebApp from DX OI Alarms
  When `<WebappLaunchUrl></WebappLaunchUrl>` is empty OneClick console is launched and when you provide the WebApp URL the WebApp is launched.
  For more information, see Integrate With DX Operation Intelligence.

Integration with DX APM

From the current release, added support for HTTP in the APM SaaS integration.

For more information, see **Discover Introscope Agents** on the Support for DX NetOps SpectrumIntegration with APM SaaS and DXI page.

Secure Domain Connector (SDC) Enhancement

This release includes the following SDC-related enhancement:

Telnet/SSH Support for SDC Modeled Devices

From the current release, you can communicate from OneCLick console to devices managed through SDC using the telnet and SSH. Select View, Preferences, Topology Tab, Telnet/SSH connection Type to access the option. You can connect through the OC client, using the following preferences:

- Connect to the device through OneClick web server and SpectroServer
- Connect to the device through SpectroServer
- Connect to the device directly

For more information, see Discover Devices Using an SDConnector Host.

Integration Compatibility Matrix

For more information about the integration of DX NetOps Spectrum with other products, see Integration Compatibility.

Platform Support

There are no enhancements to platforms supported in the previous release.

Features and Enhancements 10.4.2

This release includes the following features and enhancements:

NCM Enhancements

This release includes the following NCM-related enhancements:

Task Work Queue Size Increased

The Task Work Queue Size has been increased to parallelly process a maximum of 40 devices on each DX NetOps Spectrum host.

For more information, see Network Configuration Manager Configurations.

Creating NCM Policy Independent of Device Family

In this release, the NCM policy functionality has been enhanced to help you perform various tasks without any issue. Now, you do not need to provide the device family information at the time of creating a policy. You can create an NCM policy independent of the device family. This means that the policy is not tightly coupled with the device family at the time of its creation. This ability now gives you more flexibility to create a policy and then apply it based on your business requirements; for example:

- You can apply a single policy to one or more device families. The policy is then applied to all the devices included in those device families.
- You can also apply multiple policies to a single device family.
- You can apply a single policy to one or more global collections. This, in turn, lets you apply the policy to different device families that are included in the associated global collections.
- You can also apply multiple policies to a single global collection.
- You can apply a policy to one or more individual devices.
- You can view the policy violators at the device family, device, and global collection levels.
- You can also view the devices, device families, and global collections that are associated with a specific policy.

For more information about creating a policy, see the "Create a Policy" section in Network Configuration Manager Policies. For more information about applying policies, see the related sections in Manage Policies. For more information about viewing policy violators and policy associations, see the related sections in View Policy Information.

Comparing with Specified Content Based on a Script

From this release, when defining a multi-line block policy, you can explicitly set a script to validate content for each block of the current configuration. You define custom scripts to validate the content. DX NetOps Spectrum also provides a default script for reference. By default, the bash prompt is used to execute the script.

For more information, see the "Compare with Specified Content Based on Script" section in Network Configuration Manager Policies.

Using CLI Mode for NCM Operations on Juniper JUNOS

This release provides the following two new configuration settings in OneClick for the Juniper JUNOS device family:

- Use CLI mode for capturing configuration
- Use CLI mode for uploading configuration

When you enable these options, Network Configuration Manager (NCM) starts using the CLI mode to perform the NCM operations (capture and upload) on the Juniper JUNOS devices. Previously, only XML RPC was available for executing

the NCM operations on the JUNOS device family. Now, users can decide whether they want to use XML RPC or CLI mode.

For more information, see the "Using CLI Mode for NCM Operations on Juniper JUNOS Device Family" section in Network Configuration Manager Introduction.

NCM Script Mode Using ncmservice

In this release, DX NetOps Spectrum uses the ncmservice (NCM Service) instead of the SRAdmin process to perform the NCM script operations. This implementation helps scale up the NCM script operations.

Support for Cisco FEX Module

In this release, the Cisco FEX modules are discovered and displayed under Chassis Manager. All the FEX ports are listed under the FEX modules.

For more information, see Support for Chassis Devices.

OneClick WebApp Improvements

The following improvements are done in the current release:

- Windows undocking is supported. Click the UP arrow in the title bar of the browser to toggle the window docking.
- DX NetOps Spectrum OneClick WebApp is 508 compliant.
- The memory footprint of Web Tomcat is reduced by around 60%, hence the OneClick WebApp is that much lighter.
- Launch the OneClick WebApp in-context to open the alarm, explorer, and topology. For more information, see Launch OneClick Clients with Context.

DX NetOps Spectrum Integration Enhancements

This section lists the enhancements to the integration of DX NetOps Spectrum with various other products.

- Integration with Unified Infrastructure Management
- Integration with DX NetOps Virtual Network Assurance (DX NetOps VNA)
- Integration with DX Operational Intelligence (DX OI)

Integration with Unified Infrastructure Management

This release includes the following enhancement to DX NetOps Spectrum- UIM integration:

- Handling of unreported Inventory
  Using the new deleteUnreportedEntities property, DX NetOps Spectrum checks the presence of the unreported inventory models in UIM and deletes the models that are no longer present in UIM.
  For more information, see UIM Integration Architecture.

Integration with DX NetOps Virtual Network Assurance (DX NetOps VNA)

This release includes the following enhancements to DX NetOps Spectrum- DX NetOps VNA integration:

- Reduced Logging
  Prior to the current release, by default, the notification messages from the DX NetOps VNA application were logged to stdout.log (Windows)/catalina.out (Linux) tomcat log file causing quick increase in log file size. Now the logging happens only if debug mode is enabled. By default, the notification messages are not logged.
- Upgrading IP devices to SNMP devices
  You can now upgrade the existing IP pingable devices to SNMP devices. Enable the **ApplySNMPCapabilitiestoSDNVMs** attribute to upgrade the IP devices to SNMP devices.

For more information, see the "Upgrade IP Devices to SNMP Devices" section in Monitoring SDN Devices.

- Introducing the Dump_inventory attribute to log the DX NetOps VNA inventory data in CSV format
  For more information, see the "Logging VNA Inventory Data" section in Monitoring SDN Devices.
- Handling issues related to DX NetOps Spectrum-DX NetOps VNA on-demand sync of specific sites
  When a response is not received from DX NetOps VNA, two new states waiting_for_response and sync_failed are added.
  For more information, see the "Handling DX NetOps Spectrum-DX NetOps VNA On-Demand Sync of Particular Sites Issues" section in DX NetOps VNA Standalone and Aggregator Integration.
- Supporting Silver Peak network devices
  DX NetOps Spectrum now supports the monitoring of Silver Peak network devices through DX NetOps VNA integration. This functionality allows you to use the SD-WAN solution that is provided by Silver Peak. SD-WAN stands for Software-Defined Wide Area Networking. It is a combination of Software Defined Networking (SDN) and Wide Area Networking (WAN).
  For more information, see Monitoring SD-WAN for Silver Peak.

Integration with DX Operational Intelligence (DX OI)

This release includes the following enhancement:

- Merging inventory push to a single connector
  This release supports sending the topology inventory and its relations from DX NetOps Spectrum to Topology Analytics Service (TAS) using DX NetOps Spectrum Data Publisher (Spub).
  For more information, see Integrate with DX Operational Intelligence.

Secure Domain Connector (SDC) Enhancements

This release includes the following SDC-related enhancements:

Telnet/FTP Support for SDC-Modeled Devices

This release now supports communication through Telnet/FTP for the SDC-modeled devices. This ability lets you perform the following NCM tasks on SDC-modeled devices using Telnet/FTP as a communication mode:

- Capture
- Upload
- Sync
- Save to Startup

For more information, see the "Telnet/FTP Support for SDC-Modeled Devices" section in NCM Enablement in Secure Domain.

Running NCM Scripts on SDC-Managed Devices

DX NetOps Spectrum now lets you use Perl scripts for any operation that NCM executes on devices managed by SDC. All the operations (such as capturing or writing a startup configuration, capturing or uploading a running configuration, uploading device firmware, reloading a device, and canceling the reload operation on a device) are supported. You can configure scripts within NCM for each of these operations and execute them on the SDC-managed devices. These scripts now also accept model attributes as part of the script parameters.

For more information about supporting NCM operations on SDC-managed devices, see NCM Enablement in Secure Domain. For more information about using customized scripts to perform operations, see Network Configuration Manager Extension Utility.

Added Filtering Mechanism to Filter Traps in SDC

A new action tunnelfwd has been added in this release. This action lets you forward traps through the SDC-SS tunnel. For more information about this action, see the "Configure Filters" section in SDC TrapX Support.

SDC TrapX Enhancements

This release includes the following enhancements:

Spectrum TrapInsight Dashboard View

From this release, DX NetOps Spectrum TrapInsight provides a real-time trap trend analysis dashboard for the distributed DX NetOps Spectrum environment. The administrator can run this tool to get the trap analysis trend; it is disabled by default. This feature is part of SDC with TrapX installation. When a new trap is received, TrapX forwards the trap to Logstash. Logstash processes and sends the trap to the Influx database using the logstash-influx-output plugin.

For more information, see Spectrum TrapInsight Dashboard View.

Converting SDC Setup into SDC-TrapX Setup

In this release, you can convert an SDC setup into an SDC-TrapX setup by simply adding the trapX.config file. You can then add actions in it based on your requirements. You do not need to uninstall the existing environment. Note that the underlying approach has not been changed. For example, users can use their setup either as SDC or SDC-TrapX. They cannot use a combination of both.

For more information, see the "Considerations" section in SDC TrapX Support.

DX NetOps Spectrum API Improvements

This release includes the following enhancement:

Restrict OneClick RESTful Access to Users

As an administrator, you can allow or restrict the access to the OneClick RESTful APIs only to the DX NetOps Spectrum users. You can set API type-level access for GET, POST, PUT, and DELETE.

For more information, see the user access-related scenario in Troubleshooting User Administration.

LDAP User Group Authentication

You can now log in to DX NetOps Spectrum when it is integrated with LDAP even when the user is not present in DX NetOps Spectrum. The user is automatically created in DX NetOps Spectrum during the first login. However, only users part of configured LDAP user groups in DX NetOps Spectrum can log in automatically.

For more information, see the "LDAP User Group Authentication" section in OneClick Administration Pages.

Monitoring IPv6 BGP Peer Session

From this release, DX NetOps Spectrum uses the Cisco BGP MIB to monitor IPv6 and IPv4 BGP Peer sessions. If the device does not have Cisco BGP MIB, then DX NetOps Spectrum uses the BGP standard MIB.

For more information, see BGP Peer Session Monitoring.

Spectrum Performance View (Beta) Improvements

From the current release, DX NetOps Spectrum installer configures the Influx server and automatically creates the user *spectrum*. A new page named InsideView Configuration(beta) is added under the OneClick Administration page to configure InsideView, which has options to save, start, and stop InsideView.

For more information and to configure the influxd using https, see InsideView Configuration (beta) in OneClick Administration Pages and Spectrum Performance View (Beta).

SHA2 Support for SSH/Telnet Sessions from OneClick Clients Using MindTerm

From the current release, DX NetOps Spectrum uses the SHA2 encryption through the MindTerm utility to make Telnet and SSH connectivity.

Supporting SHA-256 and SHA-512 for SNMPv3

DX NetOps Spectrum now supports SHA-256 and SHA-512 hashing algorithms for SNMPv3 communication. This ability helps you communicate with the devices by using a more secure SNMP communication. You can use the appropriate option (SHA256 or SHA512) while creating the SNMPv3 profile. You can then select that created profile to discover (Module Discovery) the related device or to fetch (MIB Tools) the device information.

For more information about how to use these options, see the "Configuring the SNMPv3 Profile" section in Edit SNMPv3 Profiles Dialog.

Adding TraceRoute Option in OneClick

In this release, the TraceRoute option is now available in the OneClick client. Traceroute is a network diagnostic tool that helps you find the *hop* (route) information between a source and destination. The TraceRoute option is enabled for both SpectroSERVER- and SDC-modeled devices. You can access the TraceRoute option by using the shortcut Ctrl+R, right-clicking a modeled device/alarm, or using the toolbar.

For more information, see Main Toolbar.

Enhancement to Export Functionality of IP SLA Test Results

In this release, the functionality to export the IP SLA test results using Spectrum Report Manager (SRM) has been enhanced. An additional option has been introduced on the SPM Data Export page under OneClick Administration. This option lets you specify whether you want to export the minimum, maximum, and/or average value of the IP SLA tests. Based on the selected option, the corresponding data is exported. You can view the same result in the SPMResult file, which is created in the specified directory. These values are picked up from the SPM result events that are created for these tests. By default, the average value is exported.

For more information, see the "SPM Data Export Page" section in OneClick Administration Pages.

Enable/Disable Modeling of a Connection Between Two Ports

This release provides a new attribute (LockPort) on the port model type. This attribute helps you decide whether you want to create connections between two ports. By default, the value of the attribute is set to *No*. This implies that there is no change in the existing behavior while performing the discovery or establishing a connection between two ports. However, if you set the value to *Yes* and again perform the discovery, then no connection is established on the port for which LockPort is set to Yes. The LockPort value acts only when there is no connection between ports. If a connection already exists and you change the port's LockPort value to Yes and again perform the discovery, then no change happens to the existing connection.

For more information, see Enable or Disable Modeling of a Connection Between Two Ports.

Enabling Performance Tuning Options (By Default)

Prior releases of DX NetOps Spectrum introduced the following performance tuning options; however, they were disabled by default. From this release, these options are now enabled by default:

- Improved Fanout Performance

For more information, see the "Improved Fanout Performance" section in Fault Isolation.

- Discover Connections Only toward Access Points
  For more information, see the "Discover Connections Only toward Access Points" section in WLC Manager.
- Self-Health Monitoring
  For more information, see the Self-Health Monitoring article.

IP SLA Test Discovery Enabled for ASR 9900 Series Routers

In previous releases, users were not able to discover IP SLA tests for devices that did not support rttMonCtrlAdminOwner OID. From this release, IP SLA test discovery will be successful for devices that support rttMonCtrlAdminOwner OID or rttMonCtrlAdminTag OID.

Integration Compatibility Matrix

For more information about the integration of DX NetOps Spectrum with other products, see Integration Compatibility.

Platform Support

Support for the following platforms is added in the current release:

- Red Hat Enterprise Linux (RHEL) 8.0 and 8.1
- Microsoft Windows Server 2019

Device Certifications

NOTE

Information! Click **here** to access the Device Certification Database. The Device Certification Database lets you search all DX NetOps Spectrum-certified devices. For more information, see Access the Device Certification Database Online.

Device Certifications in 10.4.2.2 Release

The certification for the following devices are enhanced in this release:

Device Name	Sys OID
Checkpoint Gaia R80.30 VSX	1.3.6.1.4.1.2620.1.6.123.1.49

Device Certifications in 10.4.2.1 Release

The certification for the following devices are enhanced in this release:

Device Name	Sys OID
Teldat Atlas 360	1.3.6.1.4.1.2007.1.1.146
Teldat-V	1.3.6.1.4.1.2007.1.1.191
Teldat C1i+	1.3.6.1.4.1.2007.1.1.123
Cisco VG310	1.3.6.1.4.1.9.1.1769
Cisco VG350	1.3.6.1.4.1.9.1.1557
Cisco Catalyst 9400	1.3.6.1.4.1.9.1.2664
Cisco SNS3495K9	1.3.6.1.4.1.9.1.2139
Cisco Catalyst 6824-X-LE-40G	1.3.6.1.4.1.9.1.2275

Juniper PSA-7000	1.3.6.1.4.1.12532.256.5.1
Juniper PSA-5000	1.3.6.1.4.1.12532.256.3.1
Cisco ASR 1002-HX	1.3.6.1.4.1.9.1.2252
Cisco Catalyst 3850-48U-E	1.3.6.1.4.1.9.1.1768
Catalyst 2960L 8PSLL	1.3.6.1.4.1.9.1.2361
Catalyst 2960CX-8PC-L Switch	1.3.6.1.4.1.9.1.2191
Catalyst 9200	1.3.6.1.4.1.9.1.2694
Cisco SF302-08	1.3.6.1.4.1.9.6.1.82.8.1
Cisco SG300-28P	1.3.6.1.4.1.9.6.1.83.28.2
Cisco 819G-4G-VZ-K9	1.3.6.1.4.1.9.1.2060
Cisco ENCS 5406	1.3.6.1.4.1.9.1.2375
Cisco ENCS 5412	1.3.6.1.4.1.9.1.2377
Cisco SG300-20	1.3.6.1.4.1.9.6.1.83.20.1
Cisco Catalyst 3560CX-8PC-S Switch	1.3.6.1.4.1.9.1.2136
Palo Alto M-100	1.3.6.1.4.1.25461.2.3.30
IC4500	1.3.6.1.4.1.12532.253.2.1
Cisco ASR 920U 16SZ-IM	1.3.6.1.4.1.9.1.2504
Catalyst 9500	1.3.6.1.4.1.9.1.2593
FortiGate 3240C	1.3.6.1.4.1.12356.101.1.32401
Palo Alto M-600	1.3.6.1.4.1.25461.2.3.39
CAS-S400-A4	1.3.6.1.4.1.3417.1.4.1
CAS-S500-A1	1.3.6.1.4.1.3417.1.4.2
SG VM WARE	1.3.6.1.4.1.3417.1.1.30
SG9000	1.3.6.1.4.1.3417.1.1.29
SG-S500	1.3.6.1.4.1.3417.1.1.36
BigIP Virtual Edition	1.3.6.1.4.1.3375.2.1.3.4.43
Fortigate VM64GCP	1.3.6.1.4.1.12356.101.1.65
SG-KVM	1.3.6.1.4.1.3417.1.1.44
VeloCloud Edge 5X0	1.3.6.1.4.1.8072.3.2.10
VeloCloud Edge 5X0	1.3.6.1.4.1.8072.3.2.10
Vormetric Data Security	1.3.6.1.4.1.21513
Palo Alto M-600	1.3.6.1.4.1.25461.2.3.39
Palo Alto 3260	1.3.6.1.4.1.25461.2.3.44
Palo Alto 3220	1.3.6.1.4.1.25461.2.3.43
FortiGate 2500E	1.3.6.1.4.1.12356.101.1.25000
FortiGate 3200D	1.3.6.1.4.1.12356.101.1.32000
FortiGate 3240C	1.3.6.1.4.1.12356.101.1.32401
FortiAnalyzer 2000B	1.3.6.1.4.1.12356.103.3.20002
FortiGate 3240C	1.3.6.1.4.1.12356.101.1.32401
FortiGate VM64	1.3.6.1.4.1.12356.101.1.30
FortiGate 501E	1.3.6.1.4.1.12356.101.1.5006

FortiGate 80E	1.3.6.1.4.1.12356.101.1.842
FortiGate 80D	1.3.6.1.4.1.12356.101.1.803
FortiGate 30E	1.3.6.1.4.1.12356.101.1.306

Device Certifications in 10.4.2 Release

The following is the list of devices certified in this release:

Device Name	Sys OID	Support Level
Silver Peak ECV	1.3.6.1.4.1.23867.1.2.46	Enhanced
SilkWorm 6510	1.3.6.1.4.1.1588.2.1.1.109	Enhanced
Crossbeam C2	1.3.6.1.4.1.6848.1.3.1.4	Enhanced
Crossbeam C25	1.3.6.1.4.1.6848.1.3.1.7	Enhanced
Crossbeam C6	1.3.6.1.4.1.6848.1.3.1.5	Enhanced
LE-427	1.3.6.1.4.1.6141.1.37	Enhanced
LE-311v	1.3.6.1.4.1.6141.1.70	Enhanced
Quidway S6906R	1.3.6.1.4.1.2011.2.23.219	Enhanced
Alcatel SAS-M 24F 2XFP 7210	1.3.6.1.4.1.6527.6.2.1.2.2.2	Enhanced
Catalyst 9500-12Q	1.3.6.1.4.1.9.1.2418	Enhanced
Catalyst 9500-24Q	1.3.6.1.4.1.9.1.2419	Enhanced
Catalyst 9500-40X	1.3.6.1.4.1.9.1.2420	Enhanced
Catalyst 9500-32C	1.3.6.1.4.1.9.1.2566	Enhanced
Catalyst 9500-32QC	1.3.6.1.4.1.9.1.2567	Enhanced
Catalyst 9500-48Y4C	1.3.6.1.4.1.9.1.2568	Enhanced
Catalyst 9500-24Y4C	1.3.6.1.4.1.9.1.2573	Enhanced
Catalyst 9500-16X	1.3.6.1.4.1.9.1.2592	Enhanced
Catalyst 9500	1.3.6.1.4.1.9.1.2688	Enhanced
Cisco ENCS 5408	1.3.6.1.4.1.9.1.2376	Enhanced
Cisco TSCodecG3	1.3.6.1.4.1.9.1.2161	Enhanced
Aruba 3810M-40G-8SR	1.3.6.1.4.1.11.2.3.7.8.5.4	Enhanced
HP A5120-16G SI	1.3.6.1.4.1.25506.11.1.11	Enhanced
Juniper MAG-SM360	1.3.6.1.4.1.12532.254.4.1	Enhanced
Juniper MAG-2600	1.3.6.1.4.1.12532.254.1.1	Enhanced
Cisco SG350-28P	1.3.6.1.4.1.9.6.1.95.28.5	Enhanced
Teldat ATLAS 150	1.3.6.1.4.1.2007.1.1.104	Enhanced

Release Comparison

This release comparison table compares and provides details of the key features available in specific DX NetOps Spectrum release versions:

New Features and Enhancement2	10.4.2.2	10.4.2.1	10.4.2	10.4.1	10.4	10.3.2	10.3.1	10.3.0	10.2.3
Integration with DX NetOps Virtual Network Assurance (DX NetOps VNA): • Support for cEdge Devices • Scalability and Performance Improvements – Monitoring SD-WAN for Versa – Monitoring SD-WAN for Viptela • Disable SNMP Modeling • Dump OCS DX VNA Inventory Cache	Yes	No	No	No	No	No	No	No	No
OneClick WebApp Enhancements • OneClick WebApp Audio Alarm • OneClick WebApp Security	Yes	No	No	No	No	No	No	No	No
NCM Enhancements: • Task Work Queue Size Increased to 200 • Syslog Monitoring Enhancement	Yes	No	No	No	No	No	No	No	No
Deprecated Support: • CA eHealth • CA eHealth-CAC	Yes	Yes	Yes	No	No	No	No	No	No
Task Work Queue Size Increased to 100	Yes	Yes	No	No	No	No	No	No	No
Network Configuration Manager Configurations	Yes	Yes	No	No	No	No	No	No	No
Event Handling in FT Scenario	Yes	Yes	No	No	No	No	No	No	No
Encoded/ Plain User Password Using REST API	Yes	Yes	No	No	No	No	No	No	No
Permission to Limit NCM to Update One Device at a Time	Yes	Yes	No	No	No	No	No	No	No
Scale SNMPv3 Profile Performance	Yes	Yes	No	No	No	No	No	N	
DX NetOps Spectrum Process Initialization Change from init to systemctl	Yes	Yes	No	No	No	No	No	No	No
Support for Different Telnet/ SSH Connection from OneClick Console to Devices	Yes	Yes	No	No	No	No	No	No	No

Support to Calculate Threshold Limits for Time-based Results	Yes	Yes	No	No	No	No	No	No	No
OneClick WebApp URLs Included in Alarm Notifier Mail	Yes	Yes	No	No	No	No	No	No	No
Purge Tomcat and Webtomcat Log Files	Yes	Yes	No	No	No	No	No	No	No
Search DX VNA models from the Locator	Yes	Yes	No	No	No	No	No	No	No
Launch OneClick Console or WebApp from DX OI Alarms	Yes	Yes	No	No	No	No	No	No	No
Added Support for HTTP in the APM SaaS Integration	Yes	Yes	No	No	No	No	No	No	No
Support for Hub-Spoke Topology to monitor SD-WAN for Versa	Yes	Yes	Yes	No	No	No	No	No	No
Support for Silver Peak Network Devices	Yes	Yes	Yes	No	No	No	No	No	No
Spectrum Performance View (Beta) Improvements • InsideView Configuration(beta)	Yes	Yes	Yes	No	No	No	No	No	No
Launch OneClick Clients with Context	Yes	Yes	Yes	No	No	No	No	No	No
Spectrum TrapInsight Dashboard View	Yes	Yes	Yes	No	No	No	No	No	No
Monitoring IPv6 BGP Peer Session	Yes	Yes	Yes	No	No	No	No	No	No
LDAP User Group Authentication	Yes	Yes	Yes	No	No	No	No	No	No
Support for Cisco FEX Module	Yes	Yes	Yes	No	No	No	No	No	No
Compare with Specified Content Based on Script	Yes	Yes	Yes	No	No	No	No	No	No
Integration with Unified Infrastructure Management • Handling of Unreported Inventory For more information, see UIM Integration Architecture.	Yes	Yes	Yes	No	No	No	No	No	No
Integration with DX NetOps Virtual Network Assurance Enhancements • Upgrade IP Devices to SNMP Devices. • Handling DX NetOps Spectrum-VNA On-Demand sync of particular sites Issues • Logging VNA Inventory Data.	Yes	Yes	Yes	No	No	No	No	No	No

Restrict OneClick Restful Access to Users	Yes	Yes	Yes	No	No	No	No	No	No
Integrate with DX OI Connector Using DX NetOps Spectrum Merge Inventory Push to Single Connector	Yes	Yes	Yes	No	No	No	No	No	No
Supporting SHA-256 and SHA-512 for SNMPv3	Yes	Yes	Yes	No	No	No	No	No	No
Using CLI Mode for NCM Operations on Juniper JUNOS	Yes	Yes	Yes	No	No	No	No	No	No
Creating NCM Policy Independent of Device Family	Yes	Yes	Yes	No	No	No	No	No	No
Running NCM Scripts on SDC-Managed Devices	Yes	Yes	Yes	No	No	No	No	No	No
Adding TraceRoute in OneClick	Yes	Yes	Yes	No	No	No	No	No	No
Enhancement to Export Functionality of IP SLA Test Results	Yes	Yes	Yes	No	No	No	No	No	No
Telnet/FTP Support for SDC-Modeled Devices	Yes	Yes	Yes	No	No	No	No	No	No
Enable/Disable Modeling of a Connection Between Two Ports	Yes	Yes	Yes	No	No	No	No	No	No
Spectrum Performance View (Beta)	Yes	Yes	Yes	Yes	No	No	No	No	No
View the Client Details	Yes	Yes	Yes	Yes	No	No	No	No	No
Dockerized DX NetOps Spectrum Deployment using Kubernetes	Yes	Yes	Yes	Yes	No	No	No	No	No
Supporting Single Sign-On Using SAML 2 Authentication	Yes	Yes	Yes	Yes	No	No	No	No	No
Self Certify the Unregistered Devices	Yes	Yes	Yes	Yes	No	No	No	No	No
Supporting Swagger for REST API Documentation	Yes	Yes	Yes	Yes	No	No	No	No	No
Map CA UIM Origins DX NetOps Spectrum Landscapes	Yes	Yes	Yes	Yes	No	No	No	No	No
Export Topology View in PNG and XML Format.	Yes	Yes	Yes	Yes	No	No	No	No	No

DX NetOps Spectrum Report Manager Improvements • Enhanced SRM Alarm Handler • Partition Handler updated • Set Report Manager Preferences • SRM purge process improvement • Reporting Interval is now customizable • Enhanced RpmgrInitializeLandscape.bat/sh script	Yes	Yes	Yes	Yes	No	No	No	No	No
OneClick WebApp (Beta) Improvements • SSO Support for WebApp • Manual start of WebApp is not required • Logging out of WebApp redirects to administration page • Handling WebApp to launch using any available port • Configure OneClick WebApp User Session Timeout	Yes	Yes	Yes	Yes	No	No	No	No	No
Integration with Virtual Network Assurance Enhancements • Addition of Meraki Dashboard API • Support for New fault Codes for Cisco ACI • Retaining the SNMP models on disabling VNA integration • Roll-up Condition being propagated up to SDN Hierarchy	Yes	Yes	Yes	Yes	No	No	No	No	No
Map DX NetOps Spectrum Attributes to CA NIM Custom Attributes	Yes	Yes	Yes	Yes	No	No	No	No	No
Secure Domain Connector (SDC) Fault Isolation	Yes	Yes	Yes	Yes	No	No	No	No	No
SNMPv3 Community String Access Control	Yes	Yes	Yes	Yes	No	No	No	No	No
Support for SDC TrapX Varbind Filtering	Yes	Yes	Yes	Yes	No	No	No	No	No
Support for Diffie-Hellman (DH) Profile on SNMPv3	Yes	Yes	Yes	Yes	No	No	No	No	No
Network Configuration Manager on Secure Domain Connector	Yes	Yes	Yes	Yes	No	No	No	No	No

OneClick WebApp (beta) Improvement	Yes	Yes	Yes	Yes	Yes	No	No	No	No
Support for AdoptOpenJDK Java	Yes	Yes	Yes	Yes	Yes	No	No	No	No
SSL support for DX NetOps Spectrum Overview Dashboard	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No
SDC TrapX Support	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No
Support for Cisco Meraki	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No
Support for AWS Network Monitoring	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No
NCM Enhancement - Configuration Search	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No
OneClick WebApp (Beta) Improvements	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No
Quality Enhancements	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No
Platform Updates	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No
Introduced the OneClick WebApp - Beta	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No
Introduced Network Configuration Manager Capture Using SSH Commands	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No
Monitoring SD-WAN for Viptela and Versa	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No
Support for DX NetOps Spectrum Integration with APM SaaS and DXI	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No
CA Remote Engineer Tool to Collect Troubleshooting Data for DX NetOps Spectrum	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No
WLC Manager-Discover Connections only towards Access Points	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No
Support for Dynamic VPN (DMVPN)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No
Distribution of Virtual Center (VC) Across Landscapes - DX NetOps Spectrum-CA UIM Integration Enhancements	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No
Dockerization with Autoinstall DX NetOps Spectrum DSS	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No
Quality Enhancements	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No
Multitenant SRM with Jasper Support	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
DX NetOps Spectrum-CA Digital Operational Intelligence (DOI) Integration	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
DDMDB Performance Improvements	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No

Installing and Configuring the SCOM Manager	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
HPIRF Support	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
Modeling Gateway Enhancements	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
CXF Upgrade	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
Filtering Interfaces	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
OneClick View for 'Communication Link Down Alarms	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
DX NetOps Spectrum Dockerization	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
DX NetOps Spectrum and CA APM Integration Enhancements	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
Co-Existence of CA VNA and CA UIM Integration in DX NetOps Spectrum	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
Device Certification Updates	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Platform Updates	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
Multitenancy Support with DX NetOps Spectrum-VNA Integration and DX NetOps Spectrum- UIM Integration Single Point of Integration.	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
VMware vSphere Plugin Support	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
vMotion Support	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
VNA Cisco ACI Faults	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
WebClient Enhancements	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes
DX NetOps Spectrum-CA SOI Connector Enhancements	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
SRM Jasper Enhancements	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Device Certifications	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Integration Compatibility

Following is the compatibility matrix for the third-party and CA products that are integrated with DX NetOps Spectrum:

NOTE
For more information about supported DX NetOps releases, see the NetOps Interoperability.

Integration Product	CA Spectrum 10.4	DX NetOps Spectrum 10.4.1	DX NetOps Spectrum 10.4.2	DX NetOps Spectrum 10.4.2.1	DX NetOps Spectrum 10.4.2.2
DX NetOps Performance Management	3.6 and 3.7	19.4.1 and 3.7	See DX NetOps Interoperability	See DX NetOps Interoperability	See DX NetOps Interoperability
DX NetOps Virtual Network Assurance (DX NetOps VNA)	3.7 and 3.6.5	19.4.1 and 3.7	See DX NetOps Interoperability	See DX NetOps Interoperability	See DX NetOps Interoperability

DX Operational Intelligence (DXOI)	1.3	DX OI SaaS	1.3.3, DX OI SaaS, and 20.06	1.3.3, DX OI SaaS, and 20.06	1.3.3, DX OI SaaS, and 20.2
DX Application Performance Management	10.7 SP3	10.7 SP3, and DX APM SaaS	10.7 SP3, and DX APM SaaS 20.1	10.7 SP3, and DX APM SaaS 20.1	10.7 SP3, and DX APM SaaS 20.1
CA Virtual Assurance for Infrastructure Managers (VAIM)	12.9	12.9	12.9	12.9	12.9
SystemEdge/VAIM	12.9 (VC-AIM 5.9)	12.9 (VC-AIM 5.9)	12.9 (VC-AIM 5.9)	12.9 (VC-AIM 5.9)	12.9 (VC-AIM 5.9)
Unified Infrastructure Managemen (UIM) (Nimsoft) Deploy the hotfixes that are available for the probes.	UIM v8.5.1, v8.5.1 SP1, v9.0.2, and v9.2.0 • EMS probe - v10.2 • nas probe - v9.06 • maintenance_mode probe - 9.02 • spectrumgtw probe - v8.67 • DX NetOps Spectrum UIM Services - v8.67 • AWS Monitoring probe - v5.26 • Microsoft Azure Monitoring probe - v3.11	UIM v20.1, UIM v9.0.2, v9.1.0, and v9.2.0 • Spectrumgtw probe - v8.67, v8.68 • Spectrum UIM Services - v8.67	UIM 20.1, and UIM 9.2.0 • Spectrumgtw probe - v8.68 • Spectrum UIM Services - v8.68	UIM 20.1, and UIM 9.2.0 • Spectrumgtw probe - v8.68 • Spectrum UIM Services - v8.68	UIM 20.3.1, UIM 20.3, UIM 20.1, and UIM 9.2.0 • Spectrumgtw probe - v8.68 • Spectrum UIM Services - v8.68
Service Operations Insight (SOI)	4.2	4.2	4.2 MUK SO09444	4.2 MUK SO09444	4.2 MUK SO09444
DX NetOps Spectrum SOI Connector	2.0.0.259	2.0.0.259	2.0.0.259	2.0.0.259	2.0.0.259
CABI BOXI or CABI JasperReports® Server	CABI JasperReports Server 6.3, 6.4.2 and 6.4.3 CABI BOXI is not supported	CABI JasperReports Server 6.4.2, 6.4.3, 7.1.1 (Linux Only for 7.1.1) CABI BOXI is not supported	See DX NetOps Interoperability	See DX NetOps Interoperability	See DX NetOps Interoperability
Layer7 SiteMinder	12.8	12.8 and 12.8.03	12.8.03	12.8.03	12.8.03
CA Embedded Entitlements Manager	12.6.0.5	12.6.0.5, 12.6.1	12.6.1	12.6.1	12.6.1
LDAP	2008 and 2012	2012 and 2016	2012 and 2016	2012 and 2016	2012 and 2016
SAML	N.A.	N.A.	2.0	2.0	2.0
ServiceNow	Kingston, Jakarta, Madrid, and London	New York, Madrid, and London	New York, and Madrid	New York, Madrid, and Orlando	New York, Madrid, and Orlando
BMC Remedy ITSM (On-Premise ONLY)	9.1	9.1	9.1	9.1	9.1,
HP Service Manager	9.41 and 9.32	9.41	9.41	9.41	9.4.1

ServiceAIDE	North Star	Sapphire	Jasmine	Jasmine	Jasmine
Microsoft SCOM	SCOM 2012R2 and SCOM 2016	SCOM 2012 R2 and SCOM 2016	SCOM 2016	SCOM 2016	SCOM 2016
CA Service Desk Manager/CMDB	17.1	17.2	17.2	17.2	17.2
REX Connector	15.0 and 15.1.17	15.0 and 15.1.17	15.0 and 15.1.17	15.0 and 15.1.17	15.0 and 15.1.17

NOTE

To know the specific CA SystemEDGE agent version supported by VAIM 12.9 (VC-AIM 5.9), see VAIM 12.9 documentation.

Product Accessibility Features

CA Technologies, a Broadcom Company, is committed to ensuring that all customers, regardless of ability, can successfully use its products and supporting documentation to accomplish vital business tasks. This section outlines the accessibility features that are part of DX NetOps Spectrum.

Product Enhancements

DX NetOps Spectrum offers accessibility enhancements in the following areas:

- Display
- Sound
- Keyboard
- Mouse
- *[other enhancements if available]*

NOTE

The following information applies to Windows-based and Macintosh-based applications. Java applications run on many host operating systems, some of which already have assistive technologies available to them. For these existing assistive technologies to provide access to programs written in JPL, they need a bridge between themselves in their native environments and the Java Accessibility support that is available from within the Java virtual machine (or Java VM). This bridge has one end in the Java VM and the other on the native platform, so it will be slightly different for each platform it bridges to.

Display

To increase visibility on your computer display, you can adjust the following options:

- **Font style, color, and size of items** Lets you choose the font color, size, and other visual combinations.
- **Screen resolution** Lets you change the pixel count to enlarge objects on the screen.
- **Cursor width and blink rate** Lets you make the cursor easier to find or minimize its blinking.
- **Icon size** Lets you make icons larger for visibility or smaller for increased screen space.
- **High contrast schemes** Lets you select color combinations that are easier to see.

Sound

Use sound as a visual alternative or to make computer sounds easier to hear or distinguish by adjusting the following options:

- **Volume** Lets you turn the computer sound up or down.
- **Text-to-Speech** Lets you hear command options and text read aloud.
- **Warnings** Lets you display visual warnings.
- **Notices** Gives you aural or visual cues when accessibility features are turned on or off.
- **Schemes** Lets you associate computer sounds with specific system events.
- **Captions** Lets you display captions for speech and sounds.

Keyboard

You can make the following keyboard adjustments:

- **Repeat Rate** Lets you set how quickly a character repeats when a key is struck.
- **Tones** Lets you hear tones when pressing certain keys.
- **Sticky Keys** Lets those who type with one hand or finger choose alternative keyboard layouts.

Mouse

You can use the following options to make your mouse faster and easier to use:

- **Click Speed** Lets you choose how fast to click the mouse button to make a selection.
- **Click Lock** Lets you highlight or drag without holding down the mouse button.
- **Reverse Action** Lets you reverse the functions controlled by the left and right mouse keys.
- **Blink Rate** Lets you choose how fast the cursor blinks or if it blinks at all.
- **Pointer Options** Let you do the following:
  - Hide the pointer while typing
  - Show the location of the pointer
  - Set the speed that the pointer moves on the screen
  - Choose the pointer's size and color for increased visibility
  - Move the pointer to a default location in a dialog box

Keyboard Shortcuts

The following table lists the keyboard shortcuts that *[product name]* supports:

Keyboard	Description
Ctrl+X	Cut
Ctrl+C	Copy
Ctrl+K	Find Next
Ctrl+F	Find and Replace
Ctrl+V	Paste
Ctrl+S	Save
Ctrl+Shift+S	Save All
Ctrl+D	Delete Line
Ctrl+Right	Next Word
Ctrl+Down	Scroll Line Down
End	Line End
[other product-specific keyboard commands]	

Topology View using new Accessibility Icon theme

To provide a better user experience and in adherence to VPAT/508 standards, we have enabled an Accessibility theme for the OneClick topology view.

This theme provides new indicators denoting model severity/icon conditions on top of our existing topology view. This will enable model severity view, which is currently color coded, for users who are unable to distinguish between colors. The accessibility icons will have additional indicators so that users can easily identify the difference in the icon model's severity.

The OneClick Console has two icon themes, Accessibility and OneClick. By default, the OneClick icon theme is selected.

To enable the Accessibility theme, follow these steps:

1. Click **Start Console** from the OneClick Administration page.
   The OneClick Console user interface is launched.
2. Navigate to the menu options, **View** > **Icons Theme**, and then select **Accessibility**.
   The OneClick Console automatically refreshes and the Accessibility indicators are displayed in the topology view.

The Icons theme selection is persistent across sessions. Once you have selected an icon theme for your topology, you will view the same theme in the OneClick console every time you logon.

Mapping the Accessibility Icons/ indicators with model severity status:

To revert to the (default) OneClick icon theme, follow these steps:

1. Navigate to the **OneClick Console** menu options, **View** > **Icons Theme**, and then select **OneClick.** The OneClick Console automatically refreshes and the default theme is displayed in the topology view.

Internationalization and Localization

This section describes issues that are related to translation, and also describes other limitations that are related to non-English versions of DX NetOps Spectrum software.

Localization Overview

DX NetOps Spectrum is internationalized to support different languages and is localized to support Japanese. DX NetOps Spectrum displays a non-English language as installed by the user and displays English when the installed language is not supported.

In 10.3.1, the localized product technical documentation is not yet available. Translated documentation is forthcoming, however. The Product Management team can provide a timeline for its availability.

> **WARNING**
> Verify that your system is language-compatible and that your operating system can fully deploy the desired language before installing DX NetOps Spectrum in a non-English locale.

Language and Installation

When you install DX NetOps Spectrum, you pick the language that you want to install. If you select a non-English language, then that language *and* English is installed. If you use the GUI installers, then the languages that you can select from are derived from the System Locale. If you use the distinct installer, you specify the language on the command line. We recommend that you set the system Locale to the desired language before you install DX NetOps Spectrum.

NOTE
Installation into a localized directory is not supported.

English is always installed to provide backup to the language you select. No more than two languages can be installed - English and one non-English language. In this scenario, the non-English language is the primary language and English is the backup. A non-English language cannot be the backup.

EvFormat and PCause files are installed in *$SPECROOT*/SG-Support. English-only installations should see no directory structural difference from previous releases of DX NetOps Spectrum. When a non-English language is installed, the EvFormat and PCause files are installed in *$SPECROOT*/SG-Support/zh_TW (for example) and the English files are installed in *$SPECROOT*/SG-Support. If the non-English version of an EvFormat or PCause file cannot be found, the English version is used instead.

Customizations in *$SPECROOT*/custom follow the same structural configuration.

The suffixes for all EvFormat and PCause file names consist of ISO country and language codes. The following table identifies the supported languages, their suffixes, and sample directory files.

Language	Suffix	Sample Directory and File Name
English, United States	en_US	SG-Support/Event000aa005_en_US
Japanese, Japan	ja_JP	SG-Support/ja_JP/Event000aa005_ja_JP

Rule of Localization Homogeneity

The *Rule of Localization Homogeneity* states that all components in a distributed DX NetOps Spectrum installation must run on servers that use the same operating system Locale. Think of DX NetOps Spectrum as one application running with one language, rather than as a set of distributed services potentially running different languages.

By following the *Rule of Localization Homogeneity*, you ensure that all access and modification of data through different communication paths use one consistent language. Otherwise, myriad languages can be stored in the DX NetOps Spectrum database. The multiple languages cause problems with such data issues as display, fonts, searching, and sorting.

We recommend that you set the Locale on the servers that run DX NetOps Spectrum processes before you install DX NetOps Spectrum. Such servers include the Location server, Processd, SpectroSERVERs, OneClick servers, clients, and the Secure Domain Manager.

Rule of Localization Limits

The *Rule of Localization Limits* states that not everything in DX NetOps Spectrum is localized to a non-English language. Localization is limited because DX NetOps Spectrum manages diverse network devices that generally support only English/ASCII. In addition, the DX NetOps Spectrum core database (ModelType Catalog) is not localized. Therefore, non-English users see English in various places in DX NetOps Spectrum.

Operating System Locale

DX NetOps Spectrum supports the following locales:

- English (ISO code en_US)
- Japanese (ISO code ja_JP)

The "language packs" that are provided for each of these languages let DX NetOps Spectrum display completely in those languages. DX NetOps Spectrum uses the language for the default locale to present values for package.properties, CsEvFormat, CsPCause, and EventTables.

DX NetOps Spectrum also supports any UTF-8 characters, including character encodings such as French, German, Hebrew, Cyrillic, and more. In sum, DX NetOps Spectrum has been internationalized so that any UTF-8 characters can

be used. You can enter data in these encodings, and DX NetOps Spectrum can display that data. However, languages for which language packs are not provided lack localized files for package.property, CsEvFormat, CsPCause and EventTable support. The affected user-interface elements are displayed in English.

Setting the system locale to the proper ISO code lets DX NetOps Spectrum recognize the language support files for the language. English is used when the system locale is set to an unsupported value, such as de_DE (German). German can still be used in DX NetOps Spectrum. However, because no German language pack is available for DX NetOps Spectrum, only the German entered by the user is displayed as German.

You can validate the locale setting that DX NetOps Spectrum uses with one of the following methods. Both methods return the same value to ensure the *Rule of Localization Homogeneity*:

- Send action 0x10405 to the VNM Model to determine which locale the SpectroSERVER is using.
- Open URL http://<*host*><:*port*>/spectrum/restful/oneclick/locale to determine which Locale the OneClick server is using.

Localization Fallback

The following rules apply to the lookup of the language-specific resources: package.properties, EvFormat, PCause, and EventTables:

- If an unsupported Locale is used, everything in DX NetOps Spectrum is presented in English (en_US).
- Package.properties values for OneClick and the OneClick Admin web page:
  - If a package.properties value for OneClick cannot be found in the non-English installation, OneClick attempts to display the English version instead.
  - If a package.properties value for OneClick cannot be found in the English installation, OneClick displays the package.properties key.
- EvFormat, PCause, and EventTables:
  - If an EvFormat or PCause file cannot be found for the non-English version, DX NetOps Spectrum displays a localized error message indicating the specified file is missing. DX NetOps Spectrum then displays the English version of the specified file.
  - If an EventTable file cannot be found for the non-English version, DX NetOps Spectrum displays the EventTable index instead of the value. Consistent with the English version, no error message is generated.
- Customization continues to follow the same rules, as appropriate for the specified Locale.

Limitations and Considerations

Upgrade Considerations

The following upgrades are supported only for English to English. For upgrades from version 9.2 English to Release 9.4 non-English, only the database migration installation is supported, and English values are not converted to non-English.

- 9.2.2 (H09) to 9.4
- 9.2.3 to 9.4
- 9.2.3 (H11) to 9.4
- 9.2.3 (H12) to 9.4
- 9.3.0 to 9.4
- 9.3.0 (H01) to 9.4

For upgrades from version 9.4 (English) to 10.2 (English) only the database migration installation is supported and English values are not converted to non-English. For upgrades from 10.2 English to 10.3 Non-English, all English values are converted to non-English.

- 9.4.x to 10.2
- 10.2 to 10.3

Translation Scope

The following categories of items are not translated in this release of DX NetOps Spectrum.

Core Functionality

- **Internet standards**, such as IP addresses, MAC addresses, and MIB names and contents.
- **Common acronyms**, such as LAN, WAN, ISDN, and IP.
- **Database catalog items**, such as Model Type, Attribute IDs and Names, Relation Names, Enumerations, Flags, and import XML.
- **SNMP and device data**, such as community strings, trap contents, device configurations, GET, and SET MIB data of Type String.
  - An SNMP read of non-ASCII device data is not blocked, but is considered unsupported.
  - An SNMP SET containing non-ASCII data is rejected and an appropriate message is displayed.
  - NCM does not support non-ASCII data at any level.
  - Non-ASCII data is not supported in traps, nor is the subsequent Event processing of varbinds from the trap data, including SeverityMaps, SBG, Event Procedures, and other functional units that process trap-originating varbinds. However, our Support team has performed preliminary field testing with UTF-8 encoded characters in traps and has found no issues. Contact our Support team if you have questions about using CA eHealth SystemEDGE agents in a non-English environment.
- **CORBA (components and APIs)**, such as server names, hostnames, domain names, and DNS names. All servers in a DX NetOps Spectrum DSS environment must use ASCII hostnames and domains.
- **System-level and external utilities**, such as TFTP, FTP, SSH hostnames, and directories. This restriction is necessary in part due to limitations of the external tools. For example, the Cygwin bash shell does not support UTF-8.
- **Non-English usernames and passwords** may not work in all browsers.
- **Debug and output logs**, such as installation logs, VNM.OUT, Tomcat output logs, and Processd_log. In general, user messages are localized. System messages are not localized.
- **Configuration files**, such as .vnmrc, sdm.config, *.idb, and web.xml.
- **Developer tools**, such as dbtools.
- **SBG CsVendor/ParseMaps**. These files support consolidated syslog file matching and trap generation.

OneClick Explorer Tab

- **Products, brands, and logos**, such as DX NetOps Spectrum, IBM, Cisco, and Microsoft.
- **DX NetOps Spectrum feature names**, such as Cluster Manager, QoS Manager, Configuration Manager, and most *Manager names.
- **Model names for default models**, where these names are stored in the database or derived from the model type name. Examples of default models are Universe, World, TopOrg, and LostFound.
- **Hostname translation** depends on usage. For example, hostnames that are needed for DNS and network resolution are not localized.

Miscellaneous

- **Copyrights**, patents, and other legal information.
- **DX NetOps Spectrum version number.**
- **XML tags**. Values may be localized, but not tags.
- **DX NetOps Spectrum installation directories**. DX NetOps Spectrum can be installed in a localized directory, but the directories under *$SPECROOT* remain unlocalized.
- **Brand and product names**, such as DX NetOps Spectrum, Policy Manager, VPN Manager, and CA eHealth.
- **Text** that is embedded in images.
- **DX NetOps Spectrum Billing Application**
- **Command-line Interface**. The CLI is a script language and a debug tool, and as such is not localized. Because the CLI has been internationalized, it can work with localized data in the database.
- **Command-line elements**, including commands, error messages, and user-supplied SpectroSERVER database data.
- **Programmatic values**, such as indexes.

Spectrum Report Manager Limitations

Spectrum Report Manager database migration from the 9.1 J version of DX NetOps Spectrum is not supported.

CABI 3.x does not support Traditional Chinese.

As a workaround, install the English version of CABI on servers that are set to the Traditional Chinese locale, or on any unsupported language operating system.

If you use the English version, change the locale of the CABI server to Traditional Chinese (zh_TW) to enable the display of InfoView pages in Traditional Chinese. Set the following InfoView preferences:

- Product Locale: Traditional Chinese
- Preferred Viewing Locale: Chinese (Taiwan).

DX NetOps Spectrum and NetOps Portal Integration

9.4 integrates with CA Performance Center 2.3.00, which provides internationalization support for the same languages.

OneClick Status Bar Changes

To accommodate internationalization, the status bar in the OneClick user interface has undergone minor changes. In some views, only the Status of a selected item is shown. In such cases, tooltips display more details of item status (for example, "Running - 2 of 2 devices").

Font Considerations

Not all operating systems support every font for every language. If the console displays strange or unreadable characters, consider adjusting the default logical font to display text in Java-based user-interface applications. Examples of default logical fonts include Serif, SansSerif, Monospaced, Dialog, and DialogInput. Examples of Java-based user interface applications include OneClick, MTE, MIB Tools, and Event Configuration.

Regular Expression Considerations

DX NetOps Spectrum uses regular expressions for processing. Many expressions are visible, such as in the Locater search editor. In general, consider the data source that a regular expression evaluates:

- Localized regular expressions are supported for user data that is stored in the DX NetOps Spectrum database. However, not all DX NetOps Spectrum data is localized, which can affect the functionality of the regular expression.
- Although Event Procedures use regular expressions to parse data from traps, DX NetOps Spectrum supports only SNMP traps that contain ASCII data. DX NetOps Spectrum does not support SNMP, traps, and Event Procedures that

use non-ASCII data; however, preliminary field testing found no issues with UTF-8 character encoding in traps, as, for example, from CA eHealth SystemEDGE monitoring.
- DX NetOps Spectrum does not support localized device configuration data. Therefore, it does not support non-ASCII regular expressions for NCM block policies.

Resolved Issues

Resolved Issues in 10.4.2.2

This release contains the following fixed defects:

- **Symptom:** Tunnel interfaces disappear from the OneClick Interfaces list.
  Resolution: Code changes are made to identify the number of visits to avoid duplicates in later visits. (DE461252, 31915552, 10.4.2.2)
- **Symptom:** When an LDAP server gets changed behind a Load-Balancer, it causes an LDAP authentication issue and users cannot log into OneClick Client.
  Resolution: Code changes are made to avoid authentication issues When an LDAP server gets changed behind a Load-Balancer. (DE450066, 31720987, 10.4.2.2)
- **Symptom:** Spectrum InsideView does not monitor the DSS environment having more than six landscapes.
  Resolution: Code changes are made to monitor the DSS environment having more than six landscapes. (DE479367, 32197707, 10.4.2.2)
- **Symptom:** DX NetOps Spectrum is unable to display the Current Firmware version for the Cisco WLCs
  Resolution: DX NetOps Spectrum is reading attribute `bsnAPSoftwareVersion 0x00213a56` to display the firmware version. Now we change the code to read agentInventoryProductVersion 0x00215152 as Firmware version. (DE479855, 32248330,10.4.2.2)
- **Symptom:** Multiple patches that were provided for prior releases are missing.
  Resolution: It' was a porting defect. Changes are made to port various bug fixes which were provided as part of different PTF/debug patches. (DE469356, 32065353, 10.4.2.2)
- **Symptom:** SpectroSERVER crashes while expanding the checkpoint VSX view table.
  Resolution: Code changes are made to stop SpectroSERVER from crashing while expanding the checkpoint VSX view table. (DE467946, 32065353, 10.4.2.2)

Resolved Issues in 10.4.2.1

This release contains the following fixed defects:

- **Symptom:** Duplicate maintenance schedules getting created in DX NetOps Spectrum and DX UIM.
  Resolution: Recreating schedules if there is schedule time change received in maintenance schedule update from UIM. Duplicate maintenance schedules are not created in DX NetOps Spectrum and DX UIM. (DE461033, 10.4.2.1)
- **Symptom:** OneClick WebApp hangs when retrieving alarm details.
  Resolution: Updated the WebSwing to v20.1.5 to resolve the issue. (DE472756, 32137291, 10.4.2.1)
- **Symptom:** When SPUB integrated with the OneClick server, TopologyNameString and collectionModelNameString are not getting updated for GCs and Containers.
  Resolution: Code changes are made to update the GCs and Containers when SPUB integrated with OneClick server. (DE469977, 32170529, 10.4.2.1)
- **Symptom:** The following issues are identified:
  - False 'MODULE REMOVAL DETECTED' Alarms on Juniper Devices in regards to PowerModules.
  - This results in false alarms for event id 0x10f6d.
  - This mismatch is only seen on devices of type MX960 which have a specific PDM (power distribution module) Module integrated.

Resolution: For the PDM module, not relying on jnxOperatingState value, even if it says down, we cannot confirm on the status. Hence, overriding it and avoiding false module down alarms. (DE412869, 10.4.2.1)

- **Symptom:** The bsnAPDisassociated alarm raised on AP is not cleared now.
  Resolution:
  bsnAPAssociated is deprecated and will not be used for clearing of bsnAPDisassociated alarm raised on AP.
  ciscoLwappApAssociated is used for clearing the bsnAPDisassociated alarm.
  (DE470878, 32100629, 10.4.2.1)
- **Symptom:** When I filter alarms using the REST API query, I do not see the total alarms as shown from the OneClick UI report
  Resolution: Code changes are made to filter and display the exact number of alarms. (DE470364, DE416400, 10.4.2.1)
- **Symptom:** Aruba Access Point models are switching between Primary and Secondary controllers very frequently.
  Resolution: Only actively connected Aruba Access Points will be modeled and associated with the controller. Switching is avoided until failover happens. (DE475205, 32176341, 10.4.2.1)
- **Symptom:** Rare, unpredictable SpectroSERVER, and SDC crashes with varying stacks.
  Resolution: Resolved rare multi-threaded memory corruption in core Spectrum code. (DE472576, 10.4.2.1)
- **Symptom:** NRM_LineCard was displayed only for non-empty serial number cards.
  Resolution: Provided requested functionality by doing code changes and also by adding .vnmrc attribute. (DE471204, 32073889, 10.4.2.1)
- **Symptom:** Occasionally whenever we discover the device interface names are the models are not getting populated.
  Resolution: In one of the corner cases, interface names are not getting updated correctly. Added code to fix it. (DE470582,32108738 , 10.4.2.1)
- **Symptom:** Importing User Models using ModellingGateway adds additional Security Community String at the user level and grants all privileges to the user.
  Resolution: Importing User Models using ModellingGateway will import only expected Security Communities as similar to the older environment. No additional Security Communities at the user level will be imported/ added. (DE470548,32099863 , 10.4.2.1)
- **Symptom:** Alarm Activity by users: group report is not working when we enable the security in Report Manager preferences.
  Resolution: Alarm Activity by users: group report works when security is enabled. (DE469636, 32097585 , 10.4.2.1)
- **Symptom:** Empty varbinds was getting dropped by SDC-TrapX when processing the traps.
  Resolution: Allowing empty varbinds in traps when SDC-TrapX is processing the traps. (DE469286, 10.4.2.1)
- **Symptom:** Alarms not synced to OC when network glitch/firewall enable happens
  Resolution: Alarms getting synced with OC after network/firewall restored. (DE468540,32077565 , 10.4.2.1)
- **Symptom:** Rare, unpredictable SpectroSERVER crashes with varying stacks.
  Resolution: Resolved rare multi-threaded memory corruption in core Spectrum code. (DE466467,32040852 , 10.4.2.1)
- **Symptom:** Tomcat log prints a lot many error messages like "Error while updating item_model_mapping table for MH:"
  Resolution: Tomcat log will be free of error messages like "Error while updating item_model_mapping table for MH:" (DE465226, 31934777, 10.4.2.1)
- **Symptom:** WLC trap processing and Meraki interface check causing the memory leak.
  Resolution: WLC trap processing and Meraki interface check do not cause memory growth. (DE464269,31955069 , 10.4.2.1)
- **Symptom:** When exporting a report to excel, clear/set and start/end times stamp showing as 'a' instead of 'AM'
  Resolution: Report data (set/clear or start/end time) in Excel will display the time as 'AM' or 'PM'. (DE462631,31925449 , 10.4.2.1)
- **Symptom:** Multiple security issues fixed.

  - OC server memory increase when TAS integration is enabled
  - Landscapes were missing from Explorer Tab.
  - GlobalCollection creation fails and UI hangs.
  - Subscription request for MTypeSubscription fails with NullPointerException.
  - Andvance Options button in Discover Console fails with NullPointerException.

  Resolution: Fixed Multiple issues as follows (DE461016, 10.4.2.1):

  - OC server memory increase is fixed when TAS integration is enabled with Proxy and sending device interfaces.
  - No Landscapes missing in Explorer Tab.
  - GlobalCollection creation is successful without any exceptions or hang.
  - The subscription request for MTypeSubscription succeeds without any exceptions.
  - Andvance Options button in Discover Console works fine as expected.
- **Symptom:** Multitenant spectrumgtw is creating a model on a different SpectroServer other than the configured on.
  - Multitenant spectrumgtw is creating a model on a different SS other than the configured one
  - VMsync not happening
  - Threads block issues with vmsync and Vmotions.

  Resolution: Fixed issues as follows (DE460275,31905329 , 10.4.2.1):

  - Multitenant spectrumgtw is creating a model on a different SS other than the configured one
    VMsync is failing
    Spectrum_10.03.02.D175: Backported known fixes from 10.4.1 to XSD changes done for vmsync to allow if one the vCenter is not displayed. That is empty relations and vertices, allowing it to process and move them to unreported inventory.
  - Threads block issues with vmsync and Vmotions.
    Spectrum_10.03.02.D175: Added separate thread pools for vmsync, vmotions, and host sync to run independently without waiting for threads. Vmotions tasks will be cleared once vmsync starts.
- **Symptom:** With Save LDAP Passwords to Spectrum DB option set to YES (on LDAP Configuration webpage) and Allow user to log in with spectrum password, if local Authentication fails then go for LDAP Authentication set to NO for the NON-LDAP user, this NON-LDAP user was still able to log in with a local password.
  Resolution: With Save LDAP Passwords to Spectrum DB option set to YES(on LDAP Configuration webpage) and Allow user to log in with spectrum password, if local Authentication fails then go for LDAP Authentication set to NO for the NON-LDAP user, this NON-LDAP user should never be allowed to log in with local password because his account does not exist in LDAP server. (DE458903,31883885 , 10.4.2.1)
- **Symptom:** Customization of Tomcat log path is not supported and hence Tomcat logs continuously prints "WARNING: Tomcat not started yet" every 5 seconds even after Tomcat has been started.
  Resolution: Customization of Tomcat log path is now supported and hence Tomcat logs do not print "WARNING: Tomcat not started yet" after Tomcat starts up. (DE458239, 31866224, 10.4.2.1)
- **Symptom:** SRM not starting with BeanNotReady exceptions in the tomcat startup.
  Resolution: Resolved the BeanNotReady and Context null exception in the SRM tomcat startup. (DE457610, 31869148 , 10.4.2.1)
- **Symptom:** Alarms not synced to OC when network glitch/firewall enable happens
  Resolution: Alarms getting synced with OC after network/firewall restored. (DE457082,31862002 , 10.4.2.1)
- **Symptom:** Special characters in Event causes ArchMgr memory to grow with flat-file configuration.
  Resolution: ArchMgr memory does not grow with flat-file configuration. (DE452069, 31800261, 10.4.2.1)
- **Symptom:** The Discovery console shows vEdge devices as "Unknown" when the discovery process uses multiple SNMP community strings.
  Resolution: Discovery console should discover vEdge devices when it consists of multiple SNMP community strings. (DE439988, 20107725, 10.4.2.1)
- **Symptom:** Auto-discovery fails to model the SNMP devices when invalid SNMP string used.

Resolution: Auto-discovery skips the invalid SNMP strings and model the device with the correct SNMP string. (DE427911, 20037581, 10.4.2.1)

- **Symptom:** Interface tab rendering is taking a long time.
  Resolution: The interface tab takes less time to populate the interfaces hierarchy. (DE470917, , 10.4.2.1)
- **Symptom:** Tomcat log prints lot many error messages like "Error while updating item_model_mapping table for MH :"
  Resolution: Tomcat log will be free of error messages like "Error while updating item_model_mapping table for MH :" (DE472398, 32136166, 10.4.2.1)
- **Symptom:** When service managers are configured to monitor services/resources on other landscape's in the DSS environment, these service managers are not showing the correct state when one of the landscape's get restarted.
  Resolution: The service manager generic starts bit early and tries get the information about resources/service managers from other landscape after restart. But as the communication between the two landscapes is not fully established the SM's are not able to successfully establish the contact. Correct that code by adding those SM's to list which will re-establish connection after landscapes are fully activated. (DE468261, 10.4.2.1)
- **Symptom:** SpectroSERVER crashes while reading the external attributes from the device for the SNMP mib mismatched datatypes.
  Resolution: SpectroSERVER not crashes while reading the external attributes from device for the snmp mib mismatched datatypes. (DE477168, 32215136, 10.4.2.1)
- **Symptom:** DX NetOps Spectrum alerts flashing in/out periodically due to `orb_timeout=120` entry in the .vnmrc file.
  Resolution: Improved SpectroSERVER Heartbeat retry to prevent retry when SpectroSERVER startup. So, OneCLick retry SpectroSERVER Heartbeat in case of Heartbeat lost later due to network issues, but not SpectroSERVER startup time (DE477428, 32219325, 10.4.2.1)
- **Symptom:** CLI show mh attribute 0x129fa fetches empty data.
  Resolution: CLI show mh attribute 0x129fa fetches model handle info. (DE475246, 32175986, 10.4.2.1)

Resolved Issues in 10.4.2

This release contains the following fixed defects:

- **Symptom:** REST subscription delete requests were always successful and were returning a 200 OK response even when the subscription ID was invalid.
  Resolution: REST subscription delete requests will be successful and will return a 200 OK response only for valid subscription IDs. For invalid subscription IDs, it provides the response as 400 BAD_REQUEST. (DE449183, 20321677, 10.4.2)
- **Symptom:** Locator search was returning an error when the list contained more than 250 items.
  Resolution: Locator search is working fine now as the search criteria filtering mechanism on the OneClick server has been enhanced. (DE446038, 20165224, 10.4.2)
- **Symptom:** OneClick Webapp was slow when connected through NAT VPN.
  Resolution: OneClick Webapp now launches quickly when connected through NAT VPN. (DE445051, 20260732, 10.4.2)
- **Symptom:** SpectroSERVER stopped responding during the SDM-SDC socket communication when signed integer was being passed.
  Resolution: SpectroSERVER no longer crashes during the SDM-SDC socket communication. (DE441938, 20139157, 10.4.2)
- **Symptom:** When SNMP requests were taking more time to process, SNMP v3 credentials corresponding to those requests were getting displayed in the plain text in the VNM.out file.
  Resolution: Sensitive SNMP v3 credentials are now getting masked out in the VNM.out file. (DE441789, 20089499, 10.4.2)
- **Symptom:** Export preferences from Group-to-Group were not successful.

Resolution: Now, export preferences from Group-to-Group are working without any issue. (DE441102, 20098500, 10.4.2)

- **Symptom:** NCM policies were not generating alarms on violations. The policies remained green and did not show any violations.
  Resolution: NCM policies now generate alarm as soon as any violation is identified on the associated policy devices. (DE435905, 20080461, 10.4.2)
- **Symptom:** SSH connection from the OneClick console to a device was disconnecting after every few minutes even when the session was being actively used.
  Resolution: DX NetOps Spectrum web server sometimes was getting the NULL value for the port in the HTTP request. This was causing the connection closure due to NumberFormatException. To handle such cases, a configuration for the port in web.xml has been added with the default value as 22. (DE428628, 1367885, 10.4.2)
- **Symptom:** SLM debug (slm_debug) fault code was making the SpectroSERVER unresponsive.
  Resolution: Fixed the fault code in this release. (DE409908, 01324240, 10.4.2)
- **Symptom:** NCM needs to pass model attributes.
  Resolution: NCM now allows passing of model attributes through the command line. (DE448840, 20320396, 10.4.2)
- **Symptom:** OneClick Tomcat was running out of memory.
  Resolution: For users with huge inventory, it was taking more time to respond to the subscription request. As a result, multiple subscription requests were being sent to DX NetOps Spectrum. DX NetOps Spectrum was receiving multiple subscription requests for a single need. It had to send notifications to all those subscriptions (instead of sending a single notification). Eventually, this was causing the out-of-memory issue. This issue is now fixed. (DE452329, 31812662, 10.4.2)
- **Symptom:** The Advanced Options button on the discovery console was not opening.
  Resolution: The Advanced Options button on the discovery console now opens successfully. (DE448279, 20304211, 10.4.2)
- **Symptom:** DX NetOps Spectrum incremental sync with CA PC was taking a long time and was failing/timing-out. DX NetOps Spectrum was synchronizing unnecessary models as Tunnels/SLAPaths.
  Resolution: The Advanced Options button on the discovery console now opens successfully. DX NetOps Spectrum incremental sync with CA PC is now quicker and is not failing/timing-out. DX NetOps Spectrum now does not sync unnecessary models to CA PC. (DE425532, 20031392, 10.4.2)
- **Symptom:** Creation and saving of V3 management profiles was making SpectroSERVER unresponsive.
  Resolution: The issue has been resolved. Now, the engineID is not updated for 127.0.0.1. (DE449679, 31697282, 10.4.2)
- **Symptom:** AlarmClear stopped responding while processing the response message received from SpectroSERVER.
  Resolution: This issue has been fixed in this release. The issue was occurring when NCM sync task was coinciding with AlarmClear. (DE445737, 20129167, 10.4.2)
- **Symptom:** When migrating DSS landscapes from 10.2.1 to 10.4 by using the Modeling Gateway, some of the features (Alarm Filters and Global Collection Hierarchy) were not getting imported.
  Resolution: Alarm Filters and Global Collection Hierarchy are now getting imported successfully. Appropriate attributes have been added to the modelinggatewayresource.xml file. (DE444697, 20107478, 10.4.2)
- **Symptom:** During policy creation, watches were showing only for the first landscape.
  Resolution: Appropriate code changes have been made to ensure that watches for all the landscapes are shown. (DE444671, 20102819, 10.4.2)
- **Symptom:** DX NetOps Spectrum stops responding after a policy is applied to a global collection.
  Resolution: Fixed the issue by adding the validation of 8 MB to the Attr data length value. The issue was occurring because the Attr data length value was going above 8 MB; that is, the database field size value. (DE440918, 20111260, 10.4.2)
- **Symptom:** The trap storm alarm was not getting cleared even when the device was in maintenance.

Resolution: Added a check to process the trap storm stop event so that the trap stop alarm is cleared. (DE439806, 20107829, 10.4.2)

- **Symptom:** SpectroSERVER was not responding.
  Resolution: This issue was occurring because of negative nsize getting converted by the htonl() syscall. The issue has been resolved now. (DE429253, 20002093, 10.4.2)
- **Symptom:** Duplicate models were getting created in DX NetOps Spectrum for the CA UIM devices.
  Resolution: Duplicate models are no longer getting created. (DE418965, 01360123, 10.4.2)
- **Symptom:** Users were observing performance issues in the product.
  Resolution: The product now works as expected without any performance issues. (DE401831, 01279345, 10.4.2)
- **Symptom:** OnceClick server was running out of memory with Java heap space error.
  Resolution: The issue was occurring because of the low web server memory. It has been fixed now. (DE451492, 20313084, 10.4.2)
- **Symptom:** Users were facing issues while building Syslog Filter views (Nexus devices (7K, 9K models).
  Resolution: Changes are done to attach CiscSyslogApp model to Nexus devices. XML changes are also done to bring the syslog OneClick view for Nexus devices. (DE445204, 20261490, 10.4.2)
- **Symptom:** The Component Detail view was showing incorrect values for Secunet SINA Box. For example, the information like "Software Version" was appearing as a long string of numbers instead of a valid version number.
  Resolution: The Component Detail view was not showing the correct value for the attributes of Secunet SINA Box devices that had octet string as datatype. This issue has been fixed. (DE444712, 20199577and 31784985, 10.4.2)
- **Symptom:** The SPM tests that were scheduled and discovered from the device were not running on the configured time interval.
  Resolution: With the defect being fixed, the test is being scheduled and run after the configured time interval, as expected. (DE443008, 20076530, 10.4.2)
- **Symptom:** TL1 models were causing segment fault during the secondary SepctroSERVER startup.
  Resolution: The secondary SpectroSERVER now starts up without any issue. (DE457761, 31874041, 10.4.2)
- **Symptom:** Timeout failures were occurring on the DX NetOps Spectrum side for the PM Alarm View reports.
  Resolution: The Alarms API has been updated to perform better. Now, OneClick will fetch all alarms from SpectroSERVER based on the time range and will perform model filtering at the OneClick level instead of passing to SpectroSERVER. (DE448189, 20294595, 10.4.2)
- **Symptom:** Users did not want to create connections to certain devices by using the LockConnection attribute. However, the connections were getting created.
  Resolution: The LockConnection attribute serves a different purpose. To resolve this issue, a new attribute LockPort has been created on the port model type. This new attribute allows users to achieve the desired functionality. (DE441622, 20093962, 10.4.2)
- **Symptom:** Heartbeat was getting lost between a SpectroSERVER on one location and a OneClick server on another location. This was causing issues with the stability of the environment and continued monitoring of the network.
  Resolution: If the heartbeat is not received in 5 minutes, then it forces the domain connection to SpectroSERVER. (DE434135, 20067429, 10.4.2)
- **Symptom:** OneClick was not showing the Firmware and Serial Number information in Asset Information for Fortinet devices, which are certified.
  Resolution: Updated the default value for NRM_RunningFirmwareAttr to read the fgSysVersion for the Fortinet devices. (DE431916, 1333237, 10.4.2)
- **Symptom:** SpectroSERVER stopped responding while creating a map between the Ports and OID value list and during unpacking the OID value for Juniper hub routers/devices.
  Resolution: It was found that the value was a Component_OID (Instance_ID) of an interface, which was NOT a regular OID. To fix this issue, the value of 'is_an_object_id' is set to FALSE. (DE433667, 20066484, 10.4.2)
- **Symptom:** Weekly and monthly schedules were not running correctly.

Resolution: Corrected the behavior of weekly and monthly discovery schedules. (DE449745, 20306898, 10.4.2)

- **Symptom:** During topology update for SLA paths, rendering connection between vEdge devices and SDN_Transports was returning pipe issues. The pipe links were disappearing after appearing for some time.
  Resolution: During topology update for SLA paths, rendering connection between vEdge devices and SDN_Transports now returns link pair objects to render the connection. (DE459699, 31850589, 10.4.2)

Known Issues

This release includes the following known issues:

Cannot Start or Stop SDC on Linux If CAPKIHOME Not Set

Symptom: I tried to stop/start SDC on Linux from the SDC installed location (/opt/CA/SDMConnector/bin). DX NetOps Spectrum throws the following error if CAPKIHOME is not set.

```
n./SdmConnectorService.exe --stop
Please check atleast one of the following conditions are met.
*) Set CAPKIHOME environment variable.
            *) Pass valid second parameter to etpki_lib_init function. Ex: if the second parameter is /a/b/c/
[lib]cryptocme2.[dll][so][sl], it is assumed that /a/b/c has all the required CAPKI shared libraries
```

Resolution: Set the CAPKIHOME environment variable to export CAPKIHOME=/opt/CA/SharedComponents/CAPKI.

OneClick Jasper Integration Not Working in Kubernetes Docker Environment

Symptom: OneClick Jasper integration is not working in Kubernetes Docker environment.

Resolution: Review the following information:

First step: Create NodePort

apiVersion: v1 kind: Service metadata: name: ocstwonodeport spec: type: NodePort ports: - port: 3306 protocol: TCP name: mysql selector: name: ocstwo

Second step: Edit Jasper call

jdbc:mysql://<mastername>:45673/reporting

for example: [root@autok-c74vm1 hari]# kubectl get svc -n spectrum NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE ocsone ClusterIP 10.233.59.221 <none> 9443/TCP,8080/TCP 18h ocsonenodeport NodePort 10.233.53.13 <none> 3306:31905/TCP 20s [root@autok-c74vm1 hari]#

So, basically, "ocsone" is the Tomcat svc and "ocsonenodeport" is the MySQL svc for the port mapping.

```
jdbc:mysql://http://10.17.153.220:31905/reporting" dir="ltr">10.17.153.220:31905/reporting
```

Reconciliation Not Happening When Elastic Has a Large Number of Alarms

Symptom: Reconciliation does not happen when elastic has a large number of alarms.

Resolution: No known resolution for the issue as of now.

Bi-Monthly Patches (BMPs) on 10.4.2

> **NOTE**
>
> There are currently no BMPs released on the current release. Watch this space for any future release.

Bi-Monthly Patch(BMP) is a bundle of PTF patches available on top of a DX NetOps Spectrum GA-release. DX NetOps Spectrum BMPs are released in every two months cadence on a regular schedule. DX NetOps Spectrum Bi-Monthly

Packs are self-extracting install files for each of the supported operating systems. The BMP must be installed on both the SpectroSERVER and OneClick server. For installation and uninstall instructions, refer to the Release Notes of the BMP.

Third-Party Software License Acknowledgements

DX NetOps Spectrum 10.4.2.1

Third-party software was used in the creation of DX NetOps Spectrum. All third-party software has been used in accordance with the terms and conditions for use, reproduction, and distribution as defined by the applicable license agreements. The following list contains the third-party software license agreements for applications that are added/ included as part of the current release of DX NetOps Spectrum:

- Adopt OpenJDK 1.8u265
- Tomcat 9.0.37
- WebSwing 20.1.5

> **NOTE**
> To view the license agreements for 10.4.2.1, click here.

DX NetOps Spectrum 10.4.2

Third-party software was used in the creation of DX NetOps Spectrum. All third-party software has been used in accordance with the terms and conditions for use, reproduction, and distribution as defined by the applicable license agreements. The following list contains the third-party software license agreements for applications that are added/ included as part of the current release of DX NetOps Spectrum:

- Adopt OpenJDK 1.8u242 Hotspot
- Apache CXF 3.3.5
- Apache HTTP Server 2.4.41
- certifi 2019.11.28
- chardet 3.0.4
- Cygwin 3.1.4
- ehcache 3.8.1
- idna 2.8
- Logstash 7.6.2
- MindTerm 4.2.2
- MySQL 5.7.27
- Struts 2.5.22
- Tomcat 9.0.34
- urllib3 1.25.8
- WebSwing 20.1

> **NOTE**
> To view the license agreements 10.4.2 release, click here.

Getting Started

DX NetOps Spectrum provides robust, comprehensive and sophisticated capabilities IT organizations need to effectively monitor and manage their dynamic, complex infrastructure. DX NetOps Spectrum is a single platform that features proactive network change management, fault isolation, and root cause analysis. With DX NetOps Spectrum, you can track, manage, and optimize not only the network infrastructure but the business services running on top of it.

DX NetOps Spectrum offers:

- **Service-aware management**
  DX NetOps Spectrum helps staff discover, model, monitor, and manage the relationships between the infrastructure and the business services it supports.
- **Intelligent fault detection**
  DX NetOps Spectrum delivers patented technologies that automate device discovery and root cause analysis, speeding issue detection and remediation.
- **Proactive change management**
  DX NetOps Spectrum delivers the visibility and control that administrators need to minimize the erroneous changes that lead to performance issues and outages.
- **Comprehensive device and platform coverage**
  With DX NetOps Spectrum, managers can gain comprehensive visibility across physical, virtualized and cloud environments, and across vendors, including SDNs.
- **Customized, role-based views**
  DX NetOps Spectrum offers pre-packaged, easily customizable dashboards and reports that can be tailored to the specific needs of administrators, users, and customers.

Overview

DX NetOps Spectrum is a services and infrastructure management system that monitors the state of managed elements including the following:

- Devices
- Applications
- Host systems
- Connections

Status information such as fault and performance data from these elements is collected and stored. DX NetOps Spectrum constantly analyzes this information to track conditions within the computing infrastructure. If an abnormal condition is detected, the product isolates it, alerts you, and presents the possible causes and solutions.

Client-Server Model

The DX NetOps Spectrum design is based on the client-server model.

- Its primary server, the SpectroSERVER, is responsible for collecting, storing, and processing data.
- The SpectroSERVER uses Inductive Modeling Technology (IMT) to perform these functions. IMT combines an object-oriented database and the intelligence of inference handlers. The object-oriented database contains model types that define the representation of each managed element, and models that represent specific managed elements. The object-oriented database also contains relations that define possible associations between model types.
- The inference handlers provide more functionality to this system by reacting to events that DX NetOps Spectrum or managed elements produce.
- The SpectroSERVER stores data in the knowledge base where model types, models, and relations are defined.
- The SpectroSERVER polls manage elements and receives alert information from the computing infrastructure.
- The SpectroSERVER analyzes and stores this information in the knowledge base, and let the client applications access this information.

Client Applications

DX NetOps Spectrum includes a number of client applications.

- Its main client application is OneClick.
- OneClick provides the graphical user interface to monitor the network and to launch other client applications.
- OneClick Console provides views that contain icons, tables, and graphs to represent the different elements of the network. These graphical components present status information and provide access to management facilities specific to the managed element they represent.
- All client application data is retrieved from the SpectroSERVER.

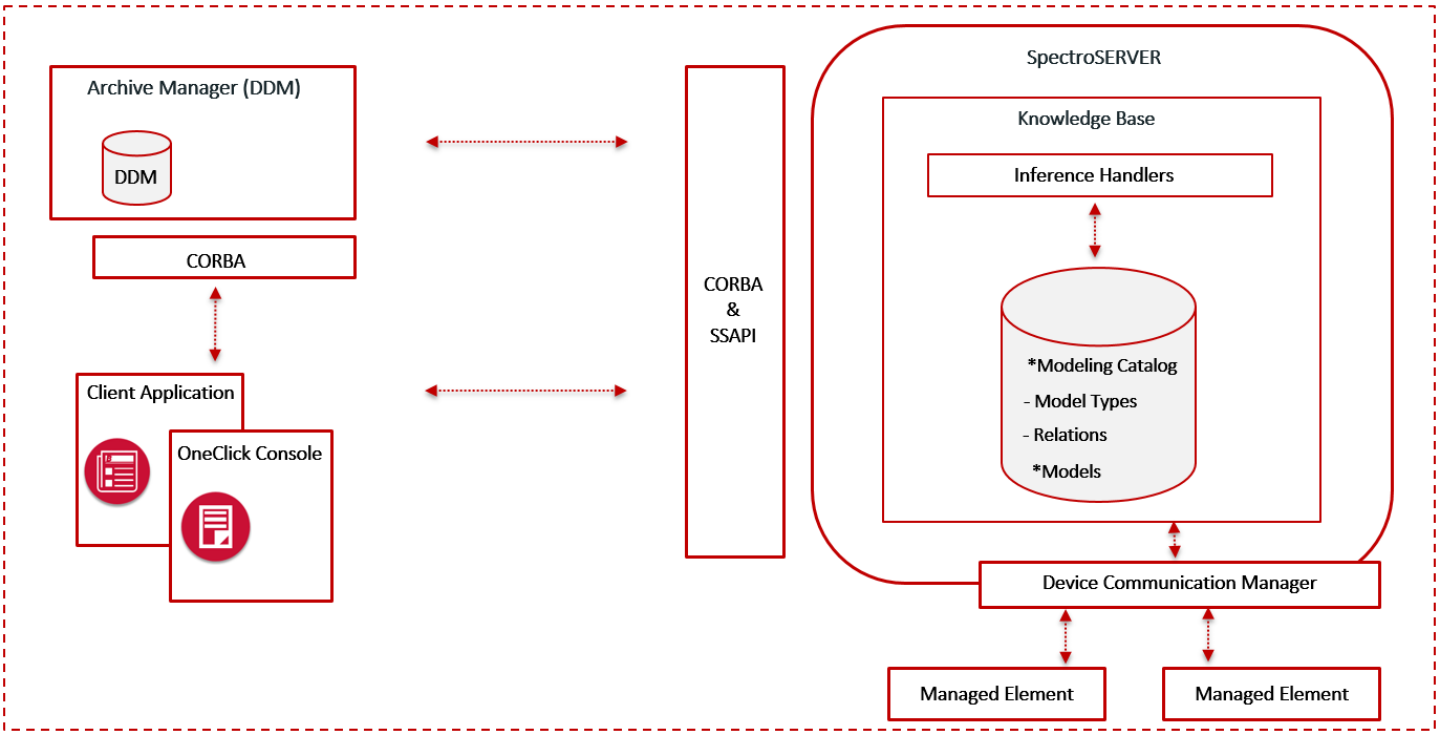
SpectroSERVER and Spectrum Databases Overview

This section explains the functions and components of SpectroSERVER, and also explains functions of the various DX NetOps Spectrum databases available.

SpectroSERVER

The SpectroSERVER is the primary server for DX NetOps Spectrum. The SpectroSERVER functions as a database server, a modeling engine, and a device manager. The SpectroSERVER processes events, generates alarms, and tracks statistics for managed elements. The client applications get this information through the SpectroSERVER application programming interface (SSAPI) and the DX NetOps Spectrum CORBA interface.

The following illustration shows the SpectroSERVER components in a simplified view:



NOTE
The SpectroSERVER is also referred to as the VNM (Virtual Network Machine). Specifically, the term VNM refers to the portion of the SpectroSERVER that is responsible for modeling managed elements.

DX NetOps Spectrum Databases

DX NetOps Spectrum includes the following databases:

- SpectroSERVER database.

For more information, see the Database Management section.
- Distributed Data Manager (DDM) database, which stores DX NetOps Spectrum events and statistical data for use across multiple landscapes.
  For more information, see the Database Management section.
- MIB Tools database, which supports the MIB Tools utility.
  For more information, see the Certification section.
- Reporting database, which support Report Manager and Service Manager.
  For more information, see the Install Report Manager section.

Knowledge Base

The knowledge base is a main component of the SpectroSERVER. The knowledge base comprises both the data and the procedural information necessary to manage computing infrastructure.

The knowledge base includes a component that stores model types, models, relations, and event and statistical information. A sophisticated system of models and relationships between models lets the knowledge base represent and store information about the network elements. This system of models and their relationships, when viewed as a single logical entity, describes the physical and logical topology of the computing infrastructure. DX NetOps Spectrum builds its *root cause analysis* capabilities on this foundation.

All models in the knowledge base are based on templates that are known as *model types*. The model types define the properties that make up an instantiated model. All model types are stored in the knowledge base modeling catalog.

The knowledge base also contains processes that give some intelligence to model types. These processes include inference handlers and actions. Process data is stored in memory while the SpectroSERVER is running and is also part of the knowledge base.

The knowledge base uses the Archive Manager and the Distributed Data Manager (DDM) to store the historical event and statistical information about specific models. This information is accumulated over time, letting DX NetOps Spectrum gain extensive knowledge about the computing infrastructure being managed.

Modeling Catalog

The modeling catalog is the metadata repository of the knowledge base. The modeling catalog objects ship with DX NetOps Spectrum and are relatively static, but you can manipulate some catalog aspects for tuning purposes. You can also customize the modeling catalog to make DX NetOps Spectrum aware of new network technologies or new types of managed network elements. The following sections describe the various specific types of objects that are included in the modeling catalog.

Model Types

The model types correspond primarily to managed element families and are the templates that are used to build models. The model types contain the information, or attributes, required to manage a specific type of managed element. The model types possess the intelligence that tells DX NetOps Spectrum how the managed element represented by the model type behaves. This intelligence also describes how the managed element reacts to events occurring on the managed element or elsewhere in the network.

For example, the DX NetOps Spectrum modeling catalog contains a NokiaFW model type. This model type represents certain types of Nokia Firewalls such as IP330, IP440, IP650, and IP740. DX NetOps Spectrum uses it to create a model that represents a specific Nokia Firewall in a network.

Each model type is uniquely identified in the modeling catalog using a *model type handle* number, typically represented in hexadecimal format.

Model Type Attributes

Each model type has attributes that define the characteristics and properties of the managed element that the model type represents. These attributes can be either internal or external. The internal attributes reflect information that is specific to the DX NetOps Spectrum management of a particular element. The external attributes reflect objects from the MIBs that the managed element supports. All attributes have default values that are associated with the model type.

In many cases, attributes take on new values when a model of a specific model type is instantiated. The attribute values are specific to the managed element that the model represents. Some attributes, however, are shared attributes. All models of the given model type access the same shared attributes and their values. These attributes and values are not duplicated in memory or the database for each model.

Each attribute is uniquely identified in the knowledge base using a number that is known as an *attribute ID*, typically represented in hexadecimal format. Many attributes are used across numerous model types. For example, almost every model type in the modeling catalog uses the attribute Modeltype_name or IPAddress. The attribute IDs for these attributes remain the same across all model types. This normalization of attributes is achieved by using model type inheritance.

Relations

Relations define the potential ways in which model types can be related to each other. Relations are defined in the DX NetOps Spectrum knowledge base. Examples of relations are Contains, Manages, and Connects_to. Each relation has a unique number, typically represented in hexadecimal format, that identifies it. This identifier is known as a *relation handle*.

Meta-Rules

Meta-rules give meaning to a relation by defining the context in which the relation is used. A meta-rule identifies the model types that can participate in a relation. To understand the concept of a meta-rule, think of model types and relations as nouns and verbs, respectively. Noun and verb phrases are combined to form a sentence. For a sentence to be meaningful, it must meet three criteria:

- The sentence must be in the format (subject) noun + verb + (object) noun.
- The sentence must be logical; one cannot use any verb to link any two nouns.
- The sentence must reflect reality.

The DX NetOps Spectrum notion of meta-rules enforces the second criterion. Meta-rules can be defined on a verb to limit the nouns that the verb can link. Define meta-rules carefully, so that the restrictions they impose on verbs are logical. Typically, several meta-rules govern each verb.

Consider a language where the following nouns and verbs are defined as the model types and relations for the objects constituting a network. Further, assume that meta-rules are defined to impose a logic on the way the nouns and verbs can be used together. Meta-rules are defined for a relation and consist of two model types: a left model type and a right model type. This left-right order in meta-rules is the format for building logical sentences. The left model type is the subject, the relation is the verb, and the right model type is the object of the sentence.

- **Nouns**
  building
  room
- **network**
  LAN
  printer
  workstation
- **Verbs**
  contains
  collects
- **Meta-rules**
  contains [building, room], [room, workstation]

collects [LAN, printer], [LAN, workstation], [network, LAN]

To create logical statements in this language, the first two criteria must be met. The following examples meet the first and second requirements and are realistic representations of a computing infrastructure:

- Engineering building contains a testing lab
- The testing lab contains workstation ABC
- Engineering LAN collects workstation ABC
- Engineering LAN collects LaserJet printer

The following sentences are invalid because they do not use the noun/verb/noun format, and therefore do not meet the first criterion:

- Contains building collects
- Room LAN workstation

The following sentences meet the format requirement, but are either illogical or do not follow the defined meta-rules:

- Building contains workstation
- LAN collects room
- Printer collects LAN

Cardinality of Relations

Relations are defined to have a cardinality of either one-to-many or many-to-many. For example, the Contains relation has a one-to-many cardinality. A meta-rule has been defined to let the Contains relation exist between the Room model type and the Workstation model type. As Contains is a one-to-many relation, a single room can contain many workstations, but a single workstation can only be in one room.

The Connects_to relation is an example of a many-to-many relation. A meta-rule has been defined to let the Connects_to relation exist between the switch and router model types. A single switch is connected to many different things, one of them the router. Likewise, a router is connected to many things, one of them being a switch.

The cardinality of relations lets DX NetOps Spectrum models be logically linked, associated, or combined in ways that can truly represent real-world computing infrastructures.

Model Type Hierarchy

The model types are built in a hierarchical fashion. The more general model types are built first, and the more specific model types are derived from the general types. A model type is derived using the principle of inheritance. Multiple inheritances are used to derive a model type from multiple base model types.

A derived model type inherits both the attributes and the intelligence of the model type or model types from which it is derived. The derived model type also participates in the same meta-rules as the base model types. In addition, the derived model type uses the same inference handlers that the base model types use.

Model types that are derived from multiple base model types inherit attributes and inference handlers from the base model types using a specific order. As a result, the derived model type cannot inherit an attribute or an inference handler multiple times. Inheritance also determines the initial value of an attribute. The new attributes, both internal and external, and new inference handlers can be added to the derived model type. The derived model type is a more specific type than its base.

Models in Knowledge Base

Along with storing model types, the knowledge base stores all the models that have been instantiated to represent elements of the computing infrastructure. A model is created by instantiating a specific model type. A copy of the template (the model type) is made, and the copy is then used to represent a real-world element in the computing infrastructure.

When a model is instantiated, the attributes of that model type take on values. The knowledge base also stores the current value for each model attribute. Some of these model attributes are “shared,” being common to all elements of the same type, and describe aspects or behavior of the model type. Each model of a given model type has the same value for these shared attributes. The unshared attributes have values which can differ for each model according to the current working condition in the infrastructure. The values of the attributes describe the unique aspects, characteristics, and behavior of the single model.

Associations of Models

When DX NetOps Spectrum creates a representation of computing infrastructure components using models, these models do not exist as isolated elements. Models relate to one another as the elements relate to one another in the computing infrastructure. When DX NetOps Spectrum instantiates a model, the applicable relations between the model and other models are also instantiated. An instantiated relation is known as an association. The association must follow the meta-rules that define the relation. OneClick and other client applications enforce meta-rules, not the SpectroSERVER.

Inference Handlers

The inference handlers define the behavior and intelligence of a model type. Each inference handler can perform a specific task. A task can be as simple as changing the value of an attribute. Or, a task can also be as complex as discovering all the managed elements on a network segment. An inference handler can also perform a generic task like calculating an average. Or, an inference handler can perform a detailed task specific to a model type, such as creating models of LAN switch ports. Basically, the inference handlers are the many pieces of intelligence that are the heart of DX NetOps Spectrum. DX NetOps Spectrum can offer its many infrastructure management capabilities because of inference handlers.

An inference handler is a C++ code segment that is associated with a model type. An inference handler is typically a dormant piece of code that supports a wide variety of triggers. Once triggered, an inference handler performs a task. The result of the task can be a new piece of data, an altered modeling scheme, or another DX NetOps Spectrum subsystem (such as another inference handler) triggered. Once the processing of inference handler ends, it goes into an idle state and awaits another trigger.

The inference handlers specify the behavior for models according to their model type and how the model type reacts to certain conditions. They can define:

- The behavior of a model when it is created, destroyed, or activated.

  NOTE
  A model is activated when it establishes the necessary communication with the managed element that it is modeling.
- The behavior of a model if the values of its attributes change, or if an event is generated on it.
- The behavior of a model when it forms a new association with another model, or when the model is removed from an existing association.
- How to handle certain actions.

The inference handlers are related to model types within the knowledge base, and they execute for instantiated models of that model type. When the external condition of two models of a model type changes in a similar way, the reaction of both models is similar. However, the values of the attributes of a specific model that reflect the status of that model have an impact on the inference handler. The attribute values of one model can be different from the attribute values of the other model. Therefore, even though the external condition is the same and the inference handlers react similarly for each model, the result of the reaction by the two models of the same model type can be different.

For example, an inference handler that is associated with a model type that represents a router is designed to perform one specific task: To create models that represent the interfaces of the router whenever a new router model

is instantiated. Once this task ends, the job of the inference handler finishes. The inference handler then waits for the creation of the next router model from this model type, so it can perform this job again.

The router model type has attributes that record the number and type of interfaces that exist on the router. Each instantiated router model represents a specific router in the computing infrastructure. And, it is likely that each router model has different values for these attributes. The number and type of interface models that the inference handler creates, are based on these values. Thus, if multiple router models are created in the knowledge base representing different routers of the type in the network, the same inference handler creates a different (but appropriate) number and type of interface models for each new router model.

The previously mentioned inference handler can once again be triggered when the router model receives a notification that the real-world router has been reconfigured. The reconfiguration can cause the number or type of interfaces on the router to change. When this change occurs, the inference handler recreates these interface models using the new information. This dynamic adaptive modeling capability is an example of one of the fundamental uses of inference handlers throughout DX NetOps Spectrum.

Actions

DX NetOps Spectrum defines a set of operations that can be performed on a model, such as reading or writing an attribute. To expand possible operations, DX NetOps Spectrum allows *actions*. An action is an operation that is not part of the basic set of operations that DX NetOps Spectrum defines for use with a model. Sending an action to a model causes the model type to react in some way. For example, it can return requested data to the action sender, or it can cause the model type to perform a specific task.

'reporting' Database

DX NetOps Spectrum Report Manager (SRM) uses a MySQL database named '**reporting**' to store data. This database contains all the tables that are required to store the data that is used by SRM application to generate reports.

At startup, the DX NetOps Spectrum Report Manager retrieves the data from the primary Archive Manager for each SpectroSERVER through OneClick and stores the data in the SRM databases.

Following is the list of tables and views in the '**reporting**' database.

The relations shown in the following diagrams merely indicate table relations. They do not indicate 1:1 or 1: many relations.

Interpret the column icons as indicated below:

- Represents a Primary Key

- Simple NOT NULL column

- Simple column which can be NULL

Tables

alarm_user

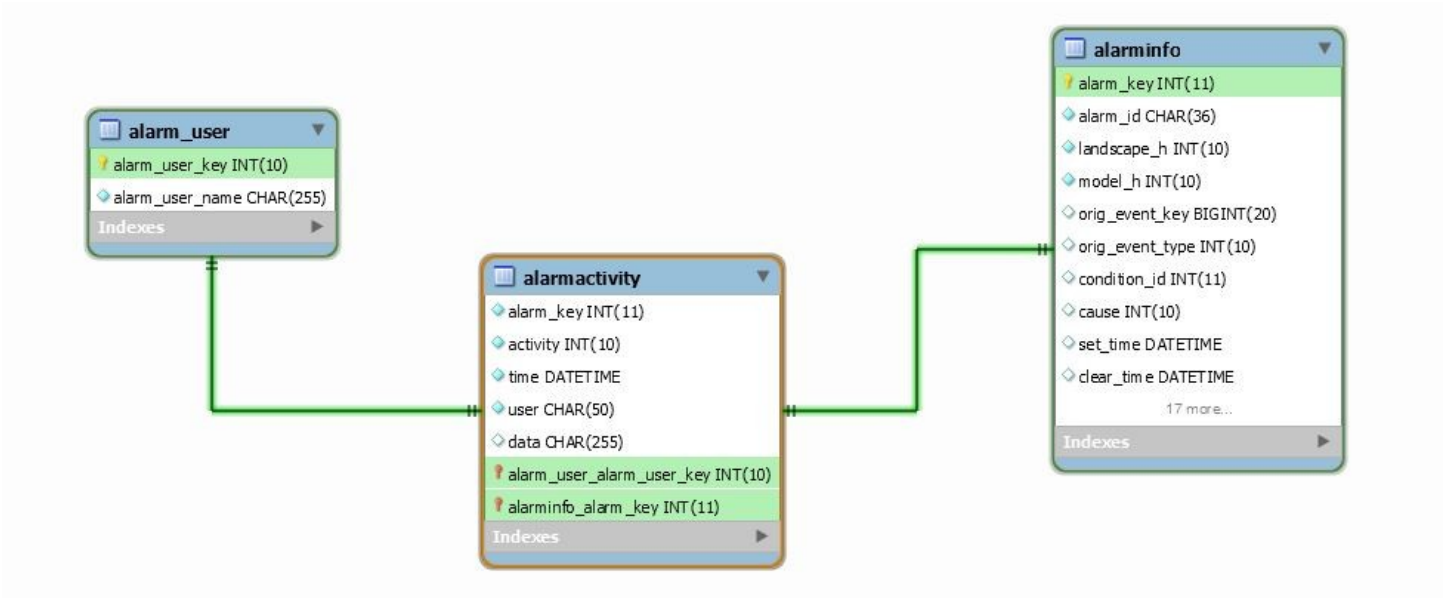
Description

This table lists the users who can manage the alarms in DX NetOps Spectrum.

Columns

Field	Type	Null	Key	Default	Extra	Comment
alarm_user_name	char(255)	NO	UNI			Name of the DX NetOps Spectrum user, who can manage the alarms.
alarm_user_key	int(10) unsigned	NO	PRI		auto_increment	Unique key for each user, who manages the alarm in DX NetOps Spectrum

Relations



alarmactivity

Description

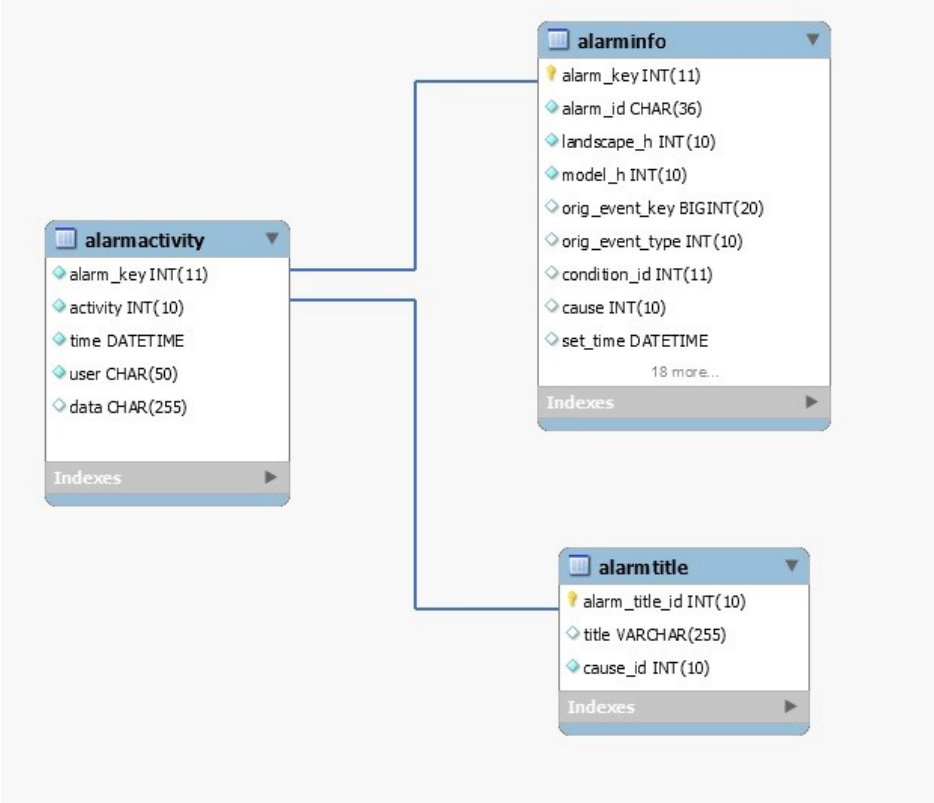
The alarmactivity table stores all the alarm activity that is monitored by SRM. The activity field denotes the type of alarm event generated. This field can be one of the following:

- Set alarm event
- Acknowledge alarm event
- Assign troubleshooter alarm event
- Clear alarm event
- User cleared alarm event
- Assign trouble ticket alarm event

Columns

Field	Type	Null	Key	Default	Comment
user	char(50)	NO	MUL		Name of the user, handling the alarm
time	datetime	NO	MUL	0000-00-00 00:00:00	Time of the activity performed in the alarm.
data	char(255)	YES			
alarm_key	int(11) unsigned	NO	MUL		Key value for specific type of Alarms.
activity	int(10) unsigned	NO	MUL		Activity that is performed by the user on the specific alarm.

Relations



alarmcondition

Description

This table represents the alarm criticality and condition.

Columns

Field	Type	Null	Key	Default	Comment
criticality	tinyint(2)	NO	UNI		Criticality of the alarm like Minor, Major, Critical
condition_name	varchar(11)	NO			Condition name of the alarm, which is mapped to the criticality.
condition_id	int(10) unsigned	NO	PRI		Unique ID for the alarm condition.

Relations

alarminfo

Description

This table contains all the alarm information including the assignee and trouble ticket IDs.

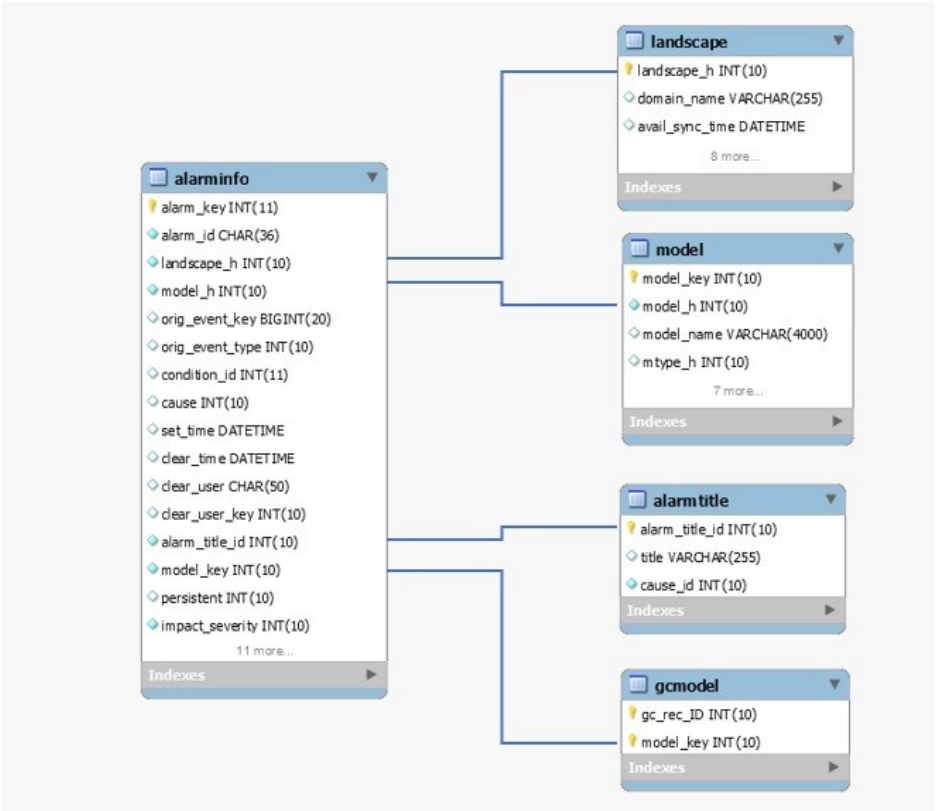
The alarminfo table stores relevant information for an alarm. There is one entry per unique alarm id, as opposed to the alarmactivity table which can have multiple entries for a single Alarm ID. An entry in this table is created when a “set alarm event” is received by SRM. The table is updated through the life of the alarm as each of the other alarm events are received by SRM.

Columns

Field	Type	Null	Key	Default	Comment
set_troubleticket_user_key	int(10) unsigned	YES	MUL		Trouble ticket user key for the specific alarm.
set_troubleticket_time	datetime	YES			Time of the trouble ticket assigned.
set_troubleticket_id	char(255)	YES			Trouble ticket ID of the alarm.
set_time	datetime	YES	MUL		Originated time of the alarm.
persistent	int(10) unsigned	YES	MUL	2	
orig_event_type	int(10) unsigned	YES			Originating event type for the alarm.
orig_event_key	bigint(20) unsigned	YES			Originating event key for the Alarm.
model_key	int(10) unsigned	NO	MUL	0	Unique model key for each alarm.
model_h	int(10) unsigned	NO	MUL		Model handle of the device, which has the alarm on it.

last_assigning_user_key	int(10) unsigned	YES	MUL		
last_assigned_user_key	int(10) unsigned	YES	MUL		
last_assigned_time	datetime	YES			
landscape_h	int(10) unsigned	NO			Landscape handle of the server, where the alarm present.
impact_severity	int(10) unsigned	NO		0	Severity of the alarm.
first_assigning_user_key	int(10) unsigned	YES	MUL		
first_assigned_user_key	int(10) unsigned	YES	MUL		
first_assigned_time	datetime	YES			First assigned time of the alarm to specific user.
condition_id	int(11)	YES			Condition of the alarm, which is mapped to the criticality.
clear_user_key	int(10) unsigned	YES	MUL		
clear_user	char(50)	YES			User who acknowledge the alarm.
clear_time	datetime	YES			When the alarm is cleared from DX NetOps Spectrum.
cause	int(10) unsigned	YES	MUL		Cause/Event ID of the specific alarm.
alarm_title_id	int(10) unsigned	NO	MUL	1	Unique ID for alarm title.
alarm_key	int(11) unsigned	NO	PRI		Unique ID for each alarm present in DX NetOps Spectrum
alarm_id	char(36)	NO	UNI		Alarm ID for specific type of alarms.
ack_user_key	int(10) unsigned	YES	MUL		Specific key for the user who acknowledges the alarm.
ack_time	datetime	YES			When the user acknowledge the alarm.

Relations



alarmtitle

Description

This table represents the title information of DX NetOps Spectrum alarms.

Columns

Field	Type	Null	Key	Default	Extra	Comment
title	varchar(255)	YES				Title of the Alarm.
cause_id	int(10) unsigned	NO	MUL			Cause or Event ID of specific type of alarm in DX NetOps Spectrum
alarm_title_id	int(10) unsigned	NO	PRI		auto_increment	Unique ID for each alarm present in DX NetOps Spectrum.

Relations

backups

Description

This table contains list of database backups.

bo_only_user

Description

This table includes the Business object user information.

boxi_user_sync

Description

This table includes the Business object user sync information.

bucketactivitylog

Description

This table includes the event information from the landscape to process in SRM database.

ca_reportstrings

Description

This table represents the information of localized string for the identifier.

configchangelog

Description

This table contains the information about NCM configuration file and logs.

Columns

Field	Type	Null	Key	Default	Extra	Comment
last_modified	bigint(20)	NO				Last_Modified time of the configuration file.
id	int(10) unsigned	NO	PRI		auto_increment	Unique ID of Configuration file.
filename	varchar(255)	NO				Name of the Configuration file.
content	blob	NO				Content of the configuration file

contentpkg

Description

This table represents the installation content package information.

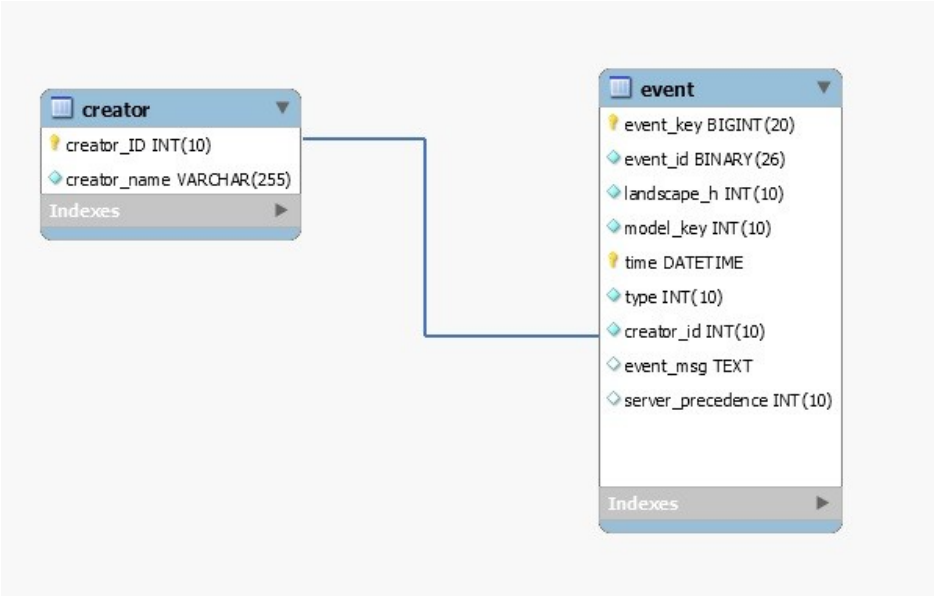
The contentpkg table associates content packages with Crystal Enterprise folder IDs. This table is not meant to be reported against, but instead is actually used by the Report Manager to help identify installation and security issues. A content package might only be installed once and this table helps identify if that is the case.

creator

Description

This table includes the model creator information.

Relations



data_retention_policy_changelog

Description

This table includes the information about the data retention policy.

devicemodel

Description

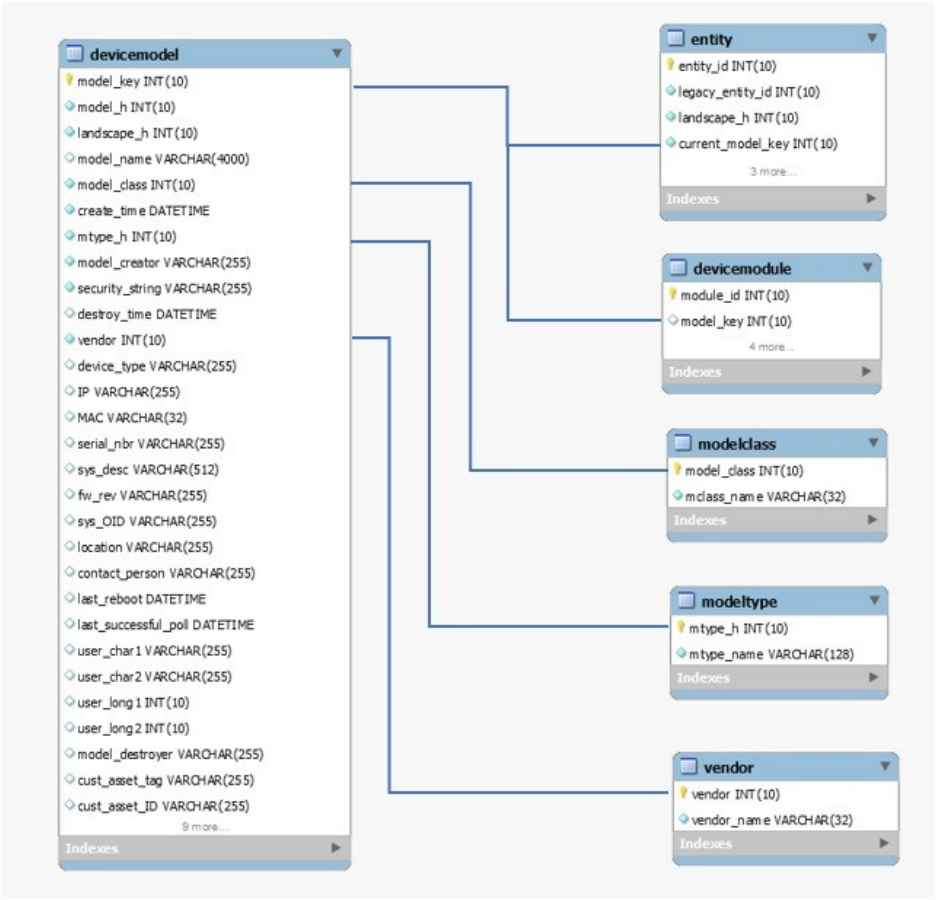
This table represents the complete device model information.

The devicemodel table is filled in initially as the Report Manager extracts model information from the respective SpectroSERVERs. New records are added by the Report Manager as it responds to model creation events for device models.

Certain attributes of this table can change and as such the Report Manager needs a way to keep up with these changes. To keep up with these changes, the Report Manager periodically requests current values for these attributes. These requests are made to the appropriate models through the OneClick architecture. Initially this period for updating device data is set to once every 24 hours.

The user-defined fields are to be left blank, but provide the administrator an opportunity to extend the Report Manager database to include data that is applicable to their assets.

Relations



devicemodel_uda

Description

The devicemodel_uda table contains user-defined polling attribute information.

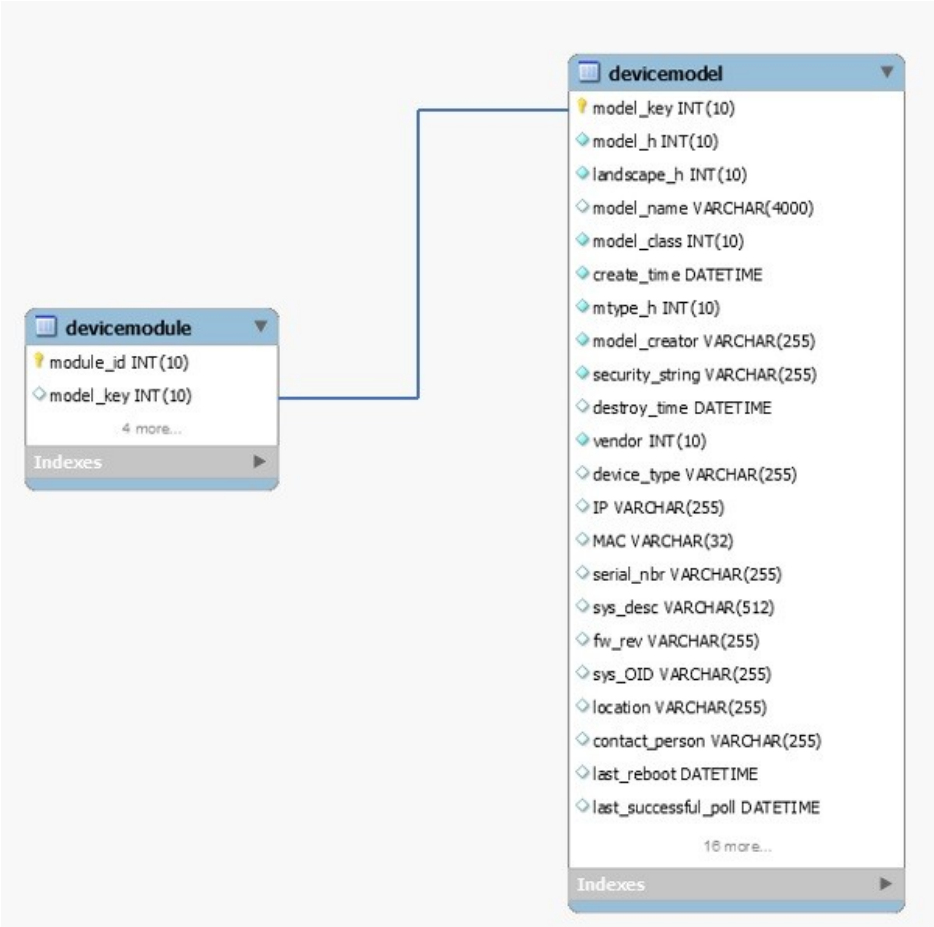
devicemodule

Description

This table includes the chassis information like module name, serial number etc.

This table captures the relationship between Chassis device models and the board modules contained within. This table is dynamically kept up to date.

Relations



Columns (till 10.2.1 release)

Field	Type	Null	Key	Default	Extra
module_id	int(10) unsigned	NO	PRI	NULL	auto_increment
model_key	int(10) unsigned	YES	MUL	NULL	
module_index	int(10)	YES		NULL	
module_name	varchar(255)	YES		NULL	
serial_nbr	varchar(255)	YES		NULL	
software_rev	varchar(255)	YES		NULL	

Columns
(from 10.2.2 release)

Field	Type	Null	Key	Default	Extra
module_id	int(10) unsigned	NO	PRI	NULL	auto_increment
model_key	int(10) unsigned	YES	MUL	NULL	
physical_index	int(10)	YES		NULL	
physical_modelname	varchar(255)	YES		NULL	
physical_name	varchar(255)	YES		NULL	

physical_class	int(10)	YES		NULL	
physical_contained_in	int(10)	YES		NULL	
serial_nbr	varchar(255)	YES		NULL	
software_rev	varchar(255)	YES		NULL	

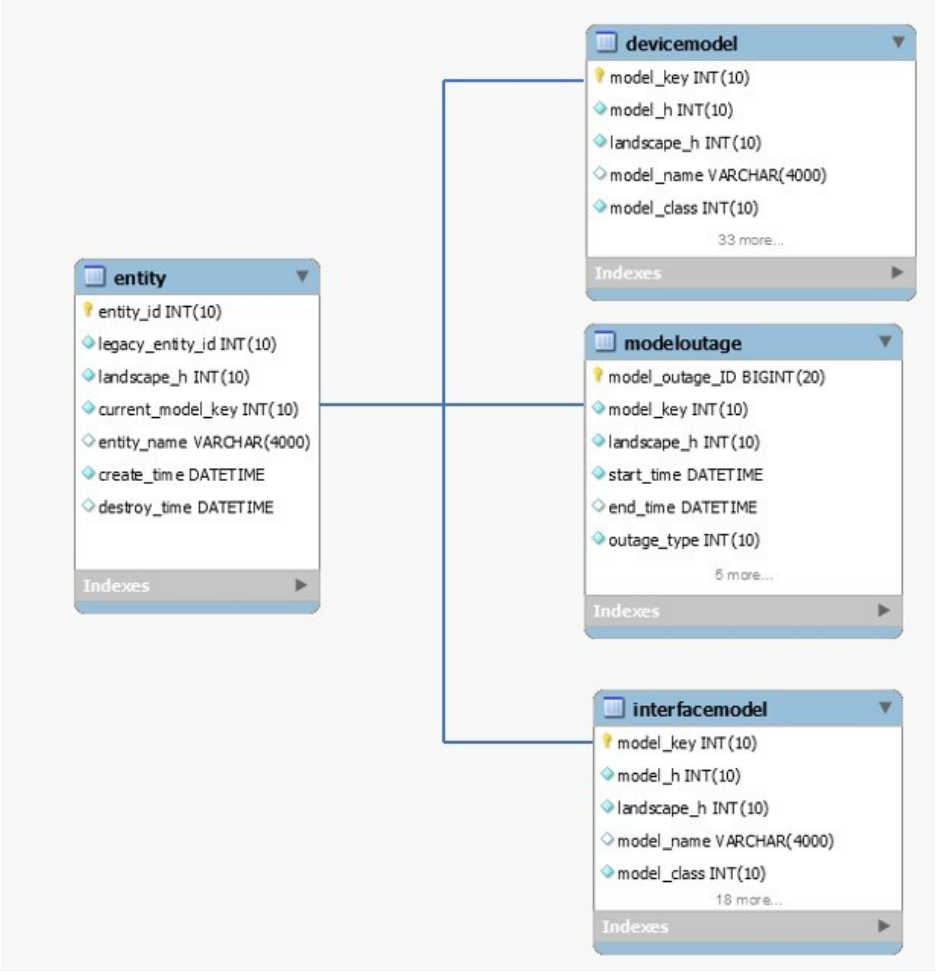
entity

Description

This table maintains distinct devices/interfaces across ALL landscapes.

The entity table is used to identify all entities uniquely that can be reported on. As new unique entities are added to the database, new entity records are created. Entity table record creation is closely tied to devicemodel and interfacemodel table record creation. The current_model, create_time, and destroy_time columns always correspond to the most recently created model.

Relations



entitygroup

Description

This table is about taking the device assets and grouping them into predefined groups that are based on Vendor, Model Class, Landscape.

The EntityGroup table is initially filled in during the startup of the Report Manager application. Queries are made to the individual SpectroSERVERs to learn of the existing model collections (which are actually models themselves).

The EntityGroup table is then kept up to date by having the Report Manager watch for the creation (and destruction) events of the collection models. When a new event occurs indicating the creation of one of these collection models, the name for that collection model is immediately obtained. A search of the EntityGroup table for a record with that name is performed. If no such record exists, one is immediately added. If a record does exist, no further processing is necessary.

entitygroupentity

Description

The EntityGroupEntity table is initially filled in during the startup of the Report Manager application. As EntityGroups are added to the system, queries are made back to each of the servers to determine the membership of those groups. In determining membership, the SpectroSERVER identifies a set of models. Each model can then be referenced in either the devicemodel or interfacemodel table. From there, an entity ID can be obtained and an appropriate entry can be made into this table.

The EntityGroupEntity table can then be kept up to date by monitoring the relationship changes associated with those collection models.

entitygrouptype

Description

The EntityGroupType table is filled in at the time of table creation. EntityGroupTypes are pre-defined before any EntityGroups have been defined. Table records include:

entity_group_type	eg_type_name
101	Vendor group
102	Model Class group
103	Landscape group
1000	User-Defined group
1001	User-Defined group

entitymodel

Description

The entitymodel is used to identify all model handles that an entity has had. This table gets filled in as part of the Entity table updating. When a record gets added to either the devicemodel or interfacemodel table, a process is kicked off to identify if this “new” model is either an existing/known entity, or a new (previously unknown/un-modeled) entity.

A new record gets added to the entitymodel table every time a “new” model record gets added to either the devicemodel or devicemodel table. When a record is added to the entitymodel table, the record is recorded with a timestamp. This timestamp enables the Report Manager to identify the most current model that represents an entity.

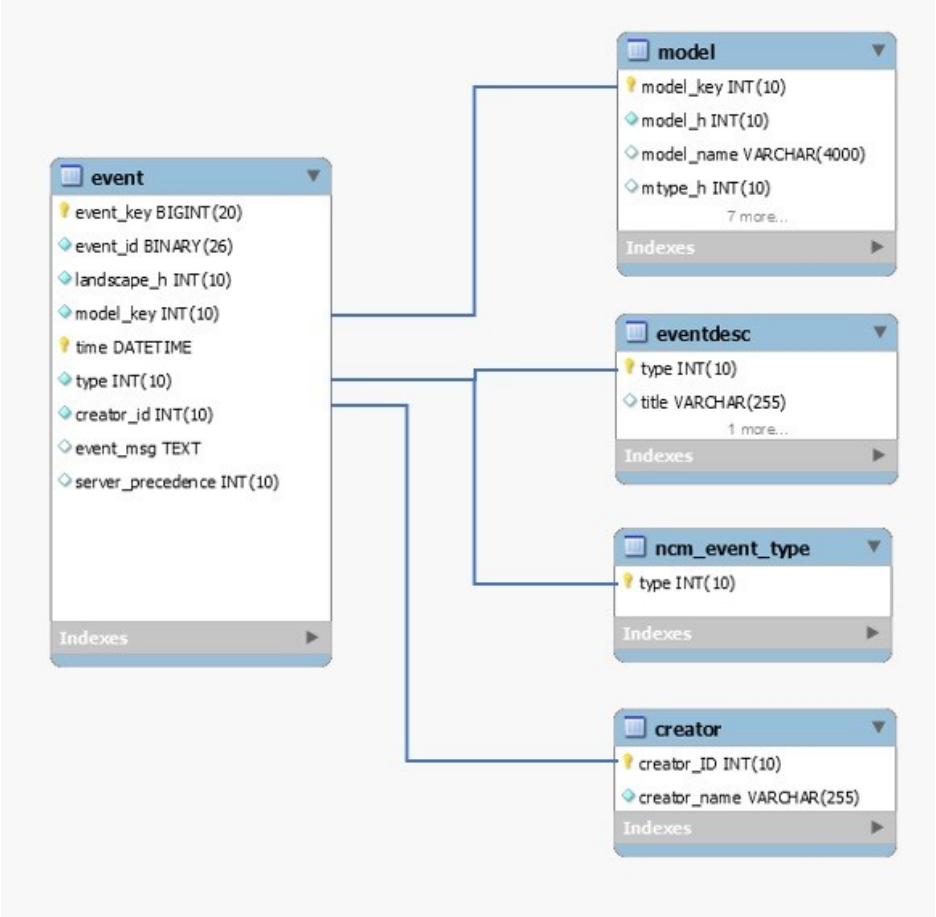
Columns

event

Description

This table described the complete event information like event ID, message, time etc.

Relations



eventactivitylog

Description

Contains number of events processed in each polling cycle for every landscape.

Columns

Field	Type	Null	Key	Default	Extra	Comment
log_id	int(10) unsigned	NO	PRI		auto_increment	Unique log id for each event processing polling cycle
landscape_h	int(10) unsigned	NO	MUL			landscape handle of SS

poll_start_time	datetime	NO				The time when polling cycle started
poll_end_time	datetime	YES				The time when polling cycle end
nbr_of_events	int(10) unsigned	YES				number of events processed in the current polling cycle

eventdesc

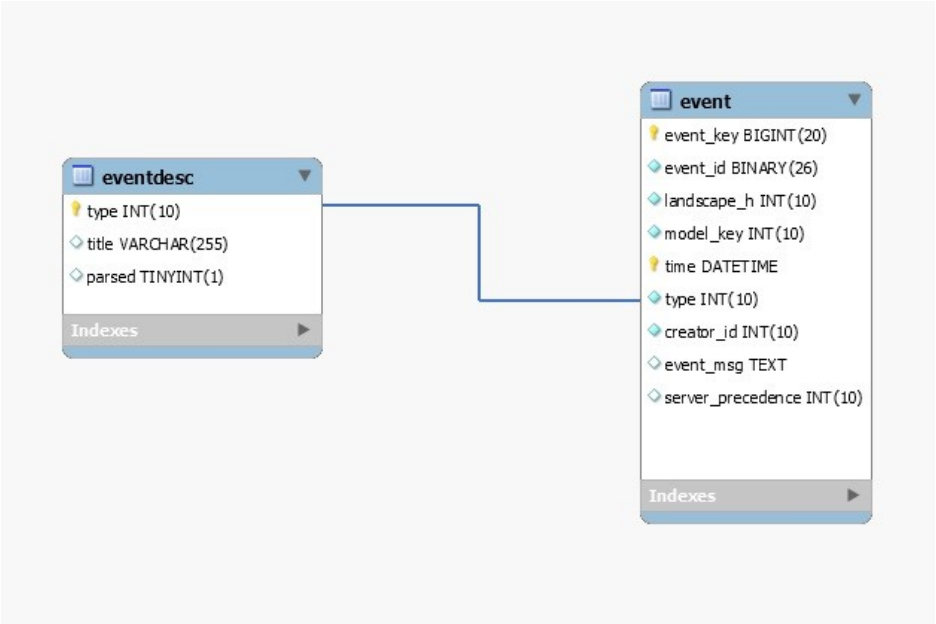
Description

Contains static list of events.

Columns

Field	Type	Null	Key	Default	Comment
type	int(10) unsigned	NO	PRI	0	Event Type
title	varchar(255)	YES			Event title
parsed	tinyint(1)	YES		0	

Relations



folderhierarchy

Description

Contains folder hierarchy present in OneClick explorer view.

Columns

folderidmap

Description

This table maps the SRM entitygroupid of the folder to the CsUniqueID that identifies the folder in One Click and DX NetOps Spectrum. This table is used for custom collection hierarchies.

Columns

Field	Type	Null	Key	Default	Comment
cs_unique_id	char(36)	NO	PRI		
folder_id	int(10) unsigned	NO	MUL		

gcmodel

Description

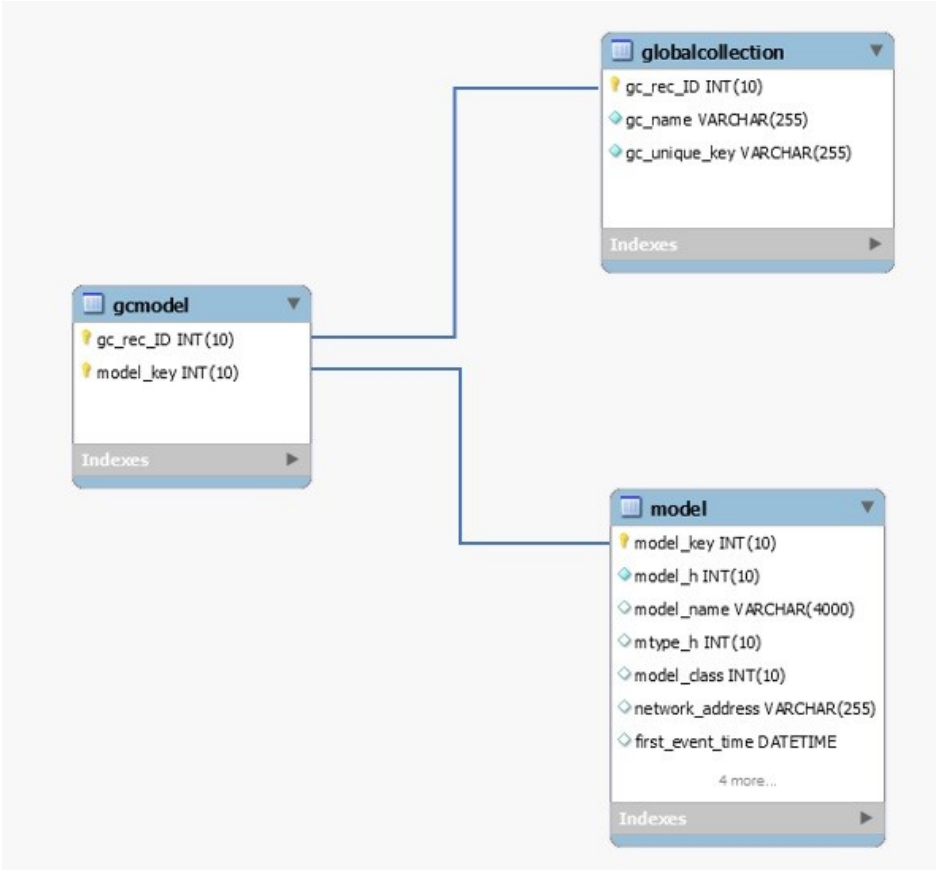
Contains mapping between global collections and models present in it.

Please note that this table does not include a flag like 'destroyed' or 'startdate/enddate' for the relation between the model and the global collection. It only represents the current assignment.

Columns

Field	Type	Null	Key	Default	Comment
gc_rec_ID	int(10) unsigned	NO	PRI		global collection record ID
model_key	int(10) unsigned	NO	PRI		model key of the model present in the global collection

Relations



globalcollection

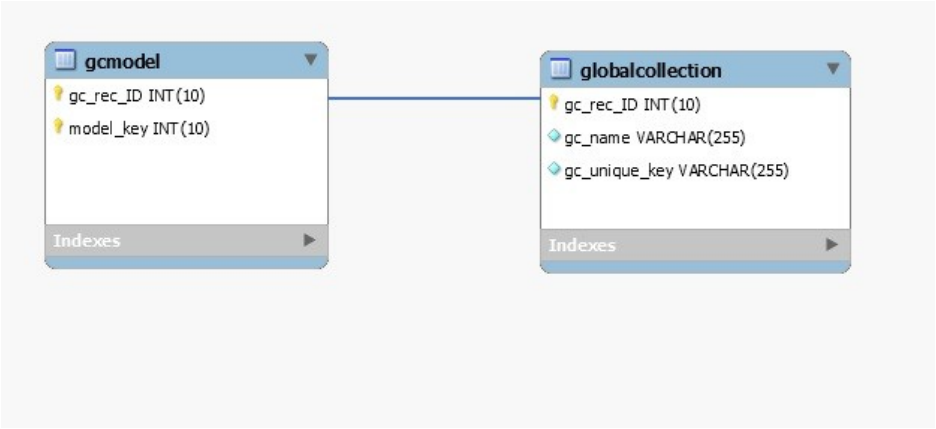
Description

Contains list of global collections.

Columns

Field	Type	Null	Key	Default	Extra	Comment
gc_rec_ID	int(10) unsigned	NO	PRI		auto_increment	unique record ID for each global collection
gc_name	varchar(255)	NO				Name of global collection
gc_unique_key	varchar(255)	NO	UNI			unique key for each global collection

Relations



groupentitygroups

Description

This table stores mapping of group_id to entity _group_id .

Since entity groups of type 'folder' can be made up of multiple entity groups (of type folder, collection or both), this table allows you to find all the entity groups that make up the specified entity group.

> **NOTE**
> When finding all entities of a specified entity group, use this table, not the entitygroup table.

Columns

Field	Type	Null	Key	Default	Comment
group_id	int(10) unsigned	NO	PRI		
entity_group_id	int(10) unsigned	NO	PRI		

handleractivitylog

Description

Containing the log of bucket processing done by all handlers for each landscape.

Columns

Field	Type	Null	Key	Default	Extra	Comment
log_id	int(10) unsigned	NO	PRI		auto_increment	Unique log id for each processing done by the handler
landscape_h	int(10) unsigned	NO	MUL			Landsacpe handle
staging_table	varchar(255)	NO	MUL			Name of the bucket which is getting processed
process_start_time	datetime	NO				Processing start time

process_end_time	datetime	YES				Processing end time
last_event_seq	int(10) unsigned	YES				
event_log_id	int(10) unsigned	NO	MUL			

handlerrollback

Description

This table stores model outage rollback data.

Columns

Field	Type	Null	Key	Default	Comment
landscape_h	int(10) unsigned	NO	PRI		
model_outage_update_ongoing	int(10) unsigned	NO		0	
last_model_outage_id	bigint(20) unsigned	YES			
last_model_outage_event_key	bigint(20) unsigned	YES			
dev_outage_update_ongoing	int(10) unsigned	NO		0	
last_dev_outage_id	int(10) unsigned	YES			
last_dev_event_time	datetime	YES			
int_outage_update_ongoing	int(10) unsigned	NO		0	
last_int_outage_id	int(10) unsigned	YES			
last_int_event_time	datetime	YES			

installedreports

Description

This table contains the file names of the reports that are loaded into the database by SRM. The report ID of the report that is stored in the database with its parent folder ID are also listed. This table is consulted when add or updating reports within SRM.

Columns

Field	Type	Null	Key	Default	Comment
report_ID	int(10) unsigned	NO	PRI		
parent_folder_ID	int(10) unsigned	NO			
file_name	varchar(255)	NO			

interfacemodel

Description

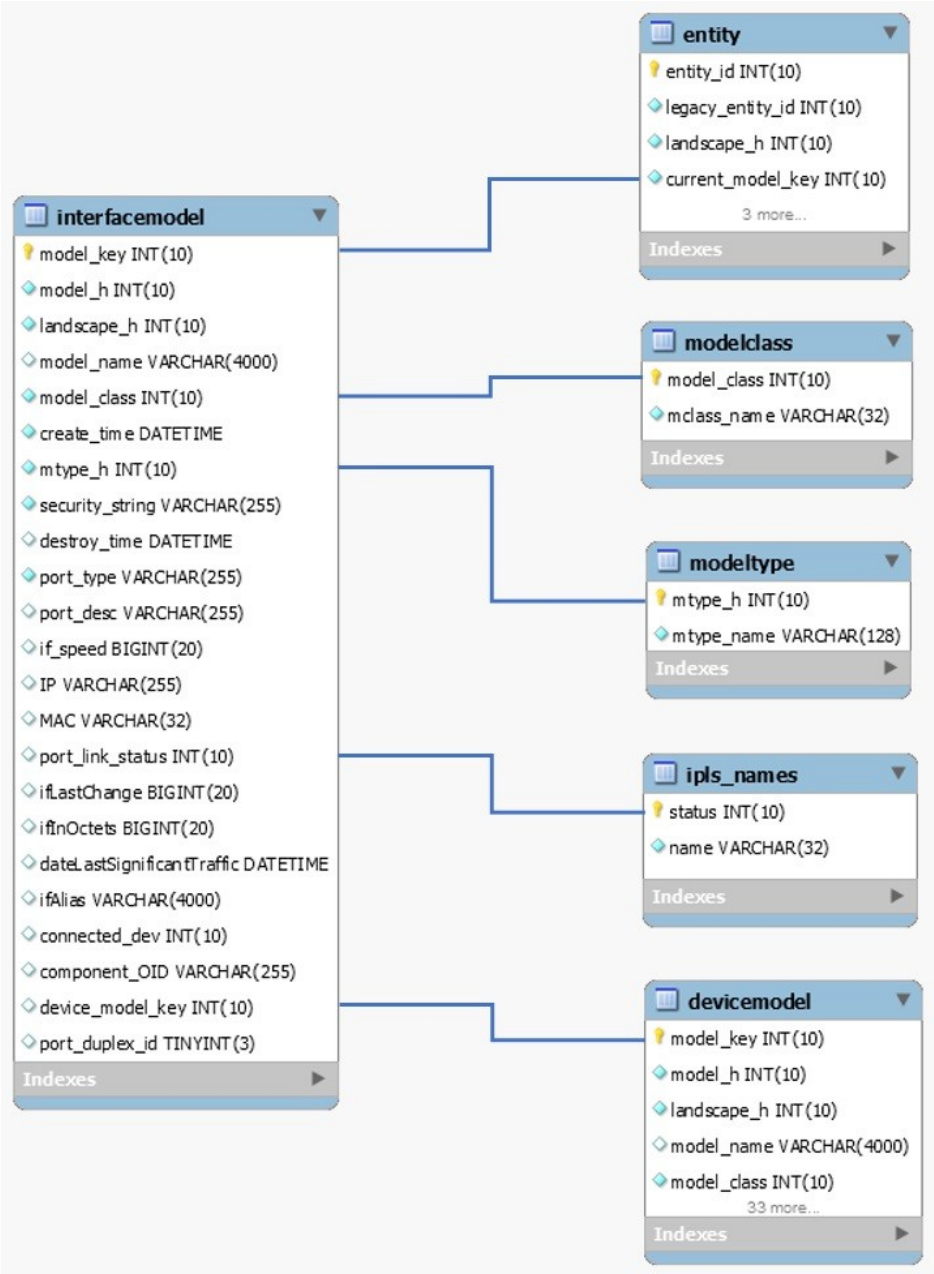
Contains interface information for all device models.

The interfacemodel table is filled in initially as the Report Manager extracts model information from the respective SpectroSERVERs. New records are added by the Report Manager as it responds to model creation events for interface models. The models that are reported on within the interfacemodel table are initially limited to those models that represent physical interfaces. Logical interfaces are not recognized or reported on with this first phase.

Columns

Field	Type	Null	Key	Default	Comment
model_key	int(10) unsigned	NO	PRI	0	model key of the interface
model_h	int(10) unsigned	NO	MUL		model handle of interface
landscape_h	int(10) unsigned	NO	MUL		Landscape handle of SS where the model is present
model_name	varchar(4000)	YES	MUL		Model name of the interface
model_class	int(10) unsigned	NO	MUL		model class of the interface
create_time	datetime	NO			Interface model creation time
mtype_h	int(10) unsigned	NO	MUL		Model Type Handle of the interface
security_string	varchar(255)	NO			security string of interface
destroy_time	datetime	YES			Destroy time of the interface model
port_type	varchar(255)	NO	MUL		Interface / port type
port_desc	varchar(255)	YES			Interface or port description
if_speed	bigint(20) unsigned	YES			Speed of the interface
IP	varchar(255)	YES			IP address of the interface
MAC	varchar(32)	YES			MAC address of the interface
port_link_status	int(10) unsigned	YES	MUL		Port or Interface link status
ifLastChange	bigint(20) unsigned	YES			value of ifLastChange mib attribute
ifInOctets	bigint(20) unsigned	YES			Value of ifInOctets mib attribute
dateLastSignificantTraffic	datetime	YES			The time when the last traffic is seen on the interface or port
ifAlias	varchar(4000)	YES			Contains the value of ifAlias mib attribute
connected_dev	int(10) unsigned	YES			Contains the interface to which this is connected
component_OID	varchar(255)	YES			
device_model_key	int(10) unsigned	YES	MUL		Model key of the device
port_duplex_id	tinyint(3) unsigned	YES			

Relations



ipls_names

Description

This table contains the different values for the port_link_status on an interface model found in the interfacemodel table. This table is filled when it is created with the following values:

Status	Name
0	Good
1	Bad
2	Unknown

3	Disabled
4	Unreachable
5	Init
6	L inked Port Bad
7	Linked Device Bad
8	Dormant
9	Port In Maintenance
10	Bad Suppressed
11	WA Link Bad
12	LL In Maintenance
13	Always Down

Columns

Field	Type	Null	Key	Default	Comment
status	int(10) unsigned	NO	PRI		Status number
name	varchar(32)	NO			Status name

Relations

landscape

Description

Contains the last poll /sync times for each landscape

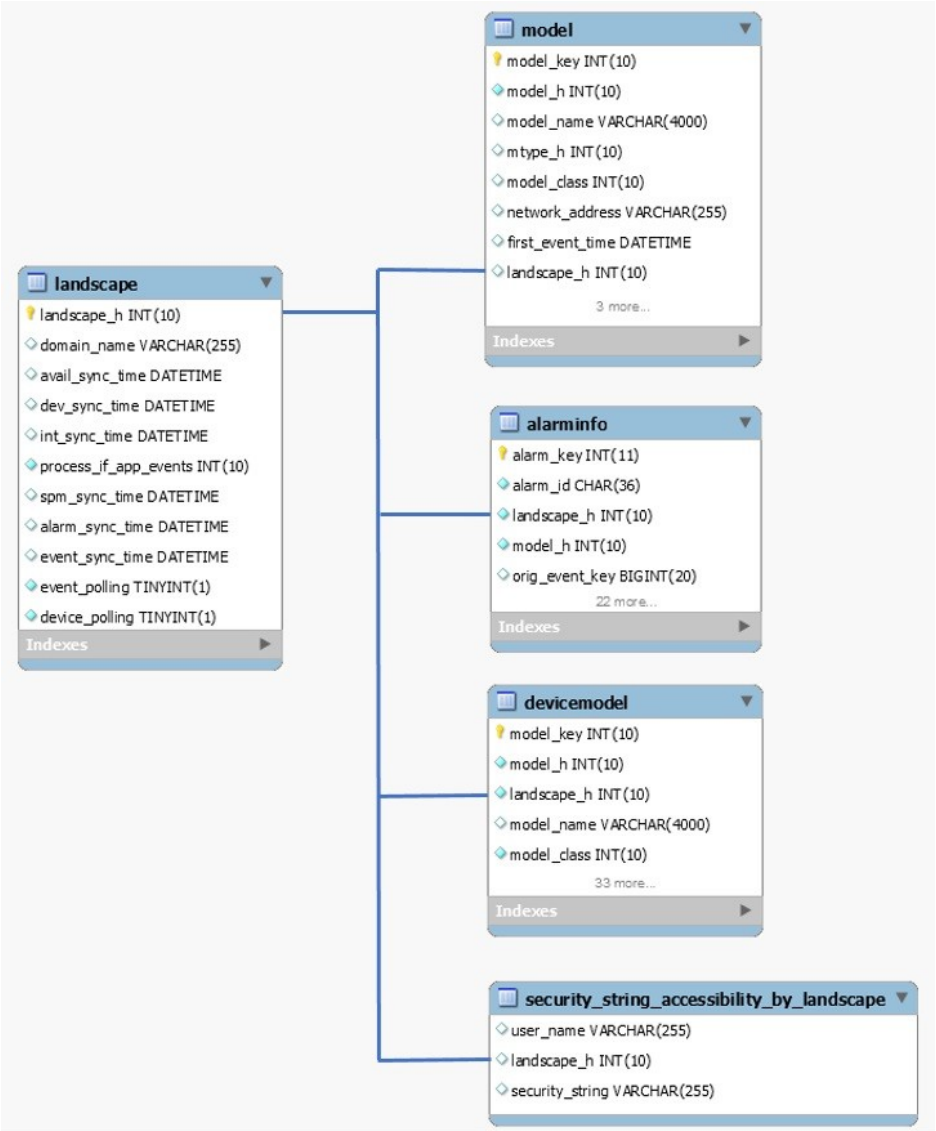
The landscape table lists those landscapes that report manager has seen. The dev_sync_time is the time the last known device event was recorded for the given landscape. The int_sync_time time is the time the last known interface event was recorded.

Columns

Field	Type	Null	Key	Default	Comment
landscape_h	int(10) unsigned	NO	PRI		Landscape handle which is polled by report manager
domain_name	varchar(255)	YES			Domain name of the landscape
avail_sync_time	datetime	YES			The last sync time for availability
dev_sync_time	datetime	YES			The last sync time for device availability
int_sync_time	datetime	YES			The last sync time for interface information

process_if_app_events	int(10)	NO		0	Indicates if it is ok to process application lost /react events for the interfaces
spm_sync_time	datetime	YES			The last sync time for SPM information
alarm_sync_time	datetime	YES			The last sync time for alarms information
event_sync_time	datetime	YES			The last sync time for events information
event_polling	tinyint(1)	NO		1	contains 1 if the event polling is enabled for this landscape else 0
device_polling	tinyint(1)	NO		1	Contains 1 if Asset polling is enabled else 0

Relations



managementoutage

Description

This table stores the management outages for the monitored landscapes.

Columns

Field	Type	Null	Key	Default	Extra	Comment
outage_ID	int(10) unsigned	NO	PRI		auto_increment	
landscape_h	int(10) unsigned	NO	MUL			
start_time	datetime	NO				
end_time	datetime	YES				
outage_type	int(10) unsigned	NO	MUL			

managementoutagetype

Description

This table lists the different types of management outages. This table is filled at table creation time with the following values:

- Expected
- Unexpected
- History

Columns

Field	Type	Null	Key	Default	Comment
outage_type	int(10) unsigned	NO	PRI		
outage_desc	varchar(32)	NO			

model

Description

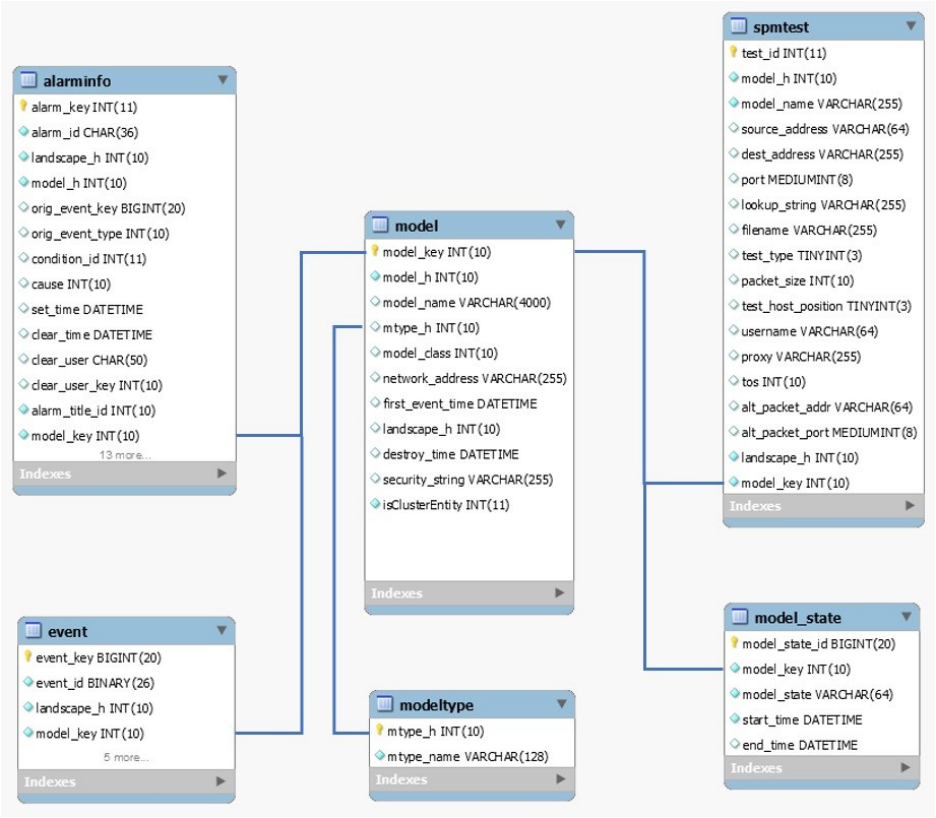
Contains information of all the DX NetOps Spectrum models. This table stores general model information.

Columns

Field	Type	Null	Key	Default	Extra	Comment
model_key	int(10) unsigned	NO	PRI		auto_increment	Unique Model Key for model
model_h	int(10) unsigned	NO	MUL	0		Model Handle of the model
model_name	varchar(4000)	YES	MUL			Name of the Model
mtype_h	int(10) unsigned	YES	MUL			Model Type Handle of the model
model_class	int(10) unsigned	YES	MUL			model class of the model

network_address	varchar(255)	YES				IP or Network address of the model
first_event_time	datetime	YES		2000-01-01 00:00:00		The time when first event got generated on the model
landscape_h	int(10) unsigned	YES	MUL			Landscape handle of SS where the model is present
destroy_time	datetime	YES	MUL			The time when model is destroyed
security_string	varchar(255)	YES	MUL	*UNKNOWN*		security string of the model
isClusterEntity	int(11)	NO		0		Flag that indicated if the model is cluster or not

Relations



model_state

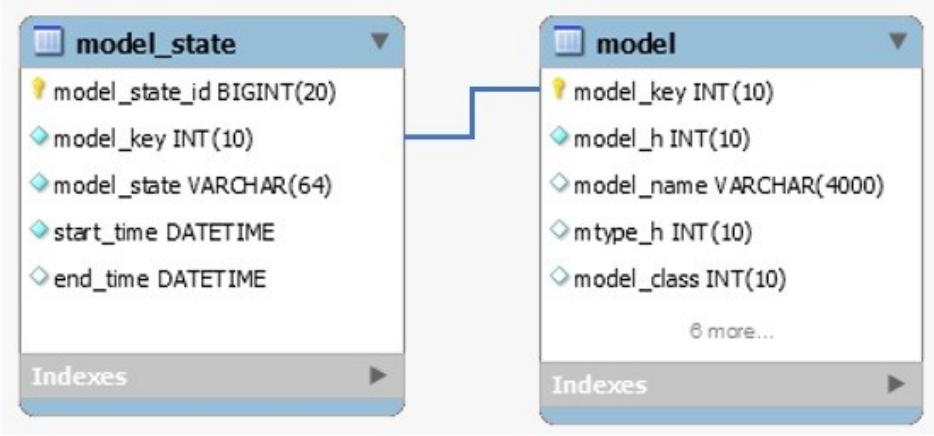
Description

Contains information about model state.

Columns

Field	Type	Null	Key	Default	Extra	Comment
model_state_id	bigint(20) unsigned	NO	PRI		auto_increment	Unique ID for each model state
model_key	int(10) unsigned	NO	MUL			Model key of the model
model_state	varchar(64)	NO	MUL			Model state
start_time	datetime	NO				The time when the model is in this state
end_time	datetime	YES				The time when the model is no longer in this state

Relations



modelclass

Description

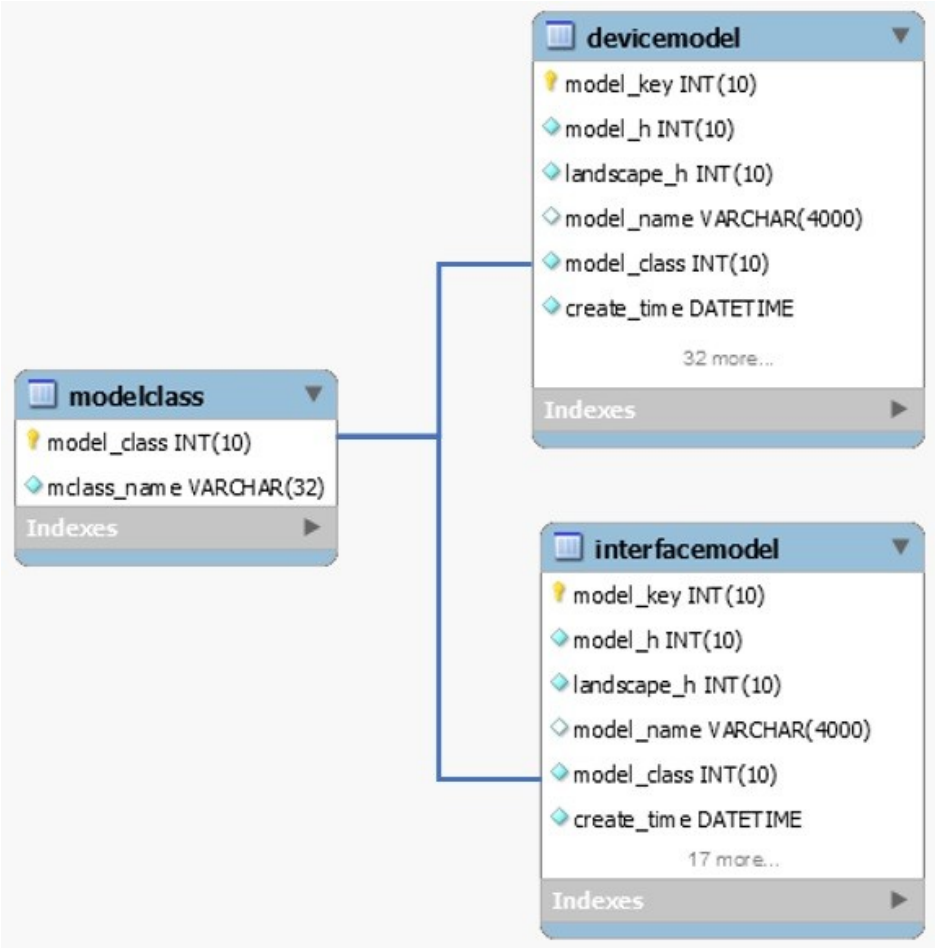
Contains static list containing list of all model class names.

The modelclass table is filled in at the time of table creation.

Columns

Field	Type	Null	Key	Default	Extra	Comment
model_class	int(10) unsigned	NO	PRI			unique number for model class
mclass_name	varchar(32)	NO	MUL			Name of the model class

Relations



modeloutage

Description

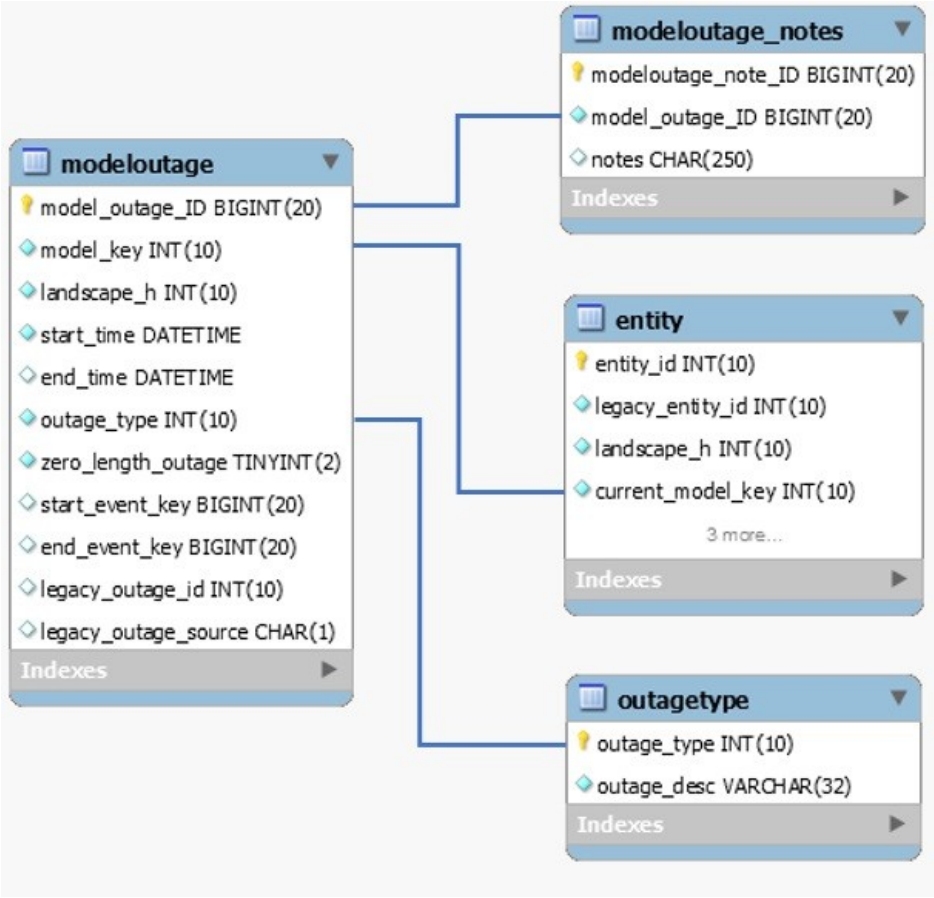
Contains all the outages for the model.

Columns

Field	Type	Null	Key	Default	Extra	Comment
model_outage_ID	bigint(20) unsigned	NO	PRI		auto_increment	Unique ID for each outage of the model
model_key	int(10) unsigned	NO	MUL			Model Key of the model
landscape_h	int(10) unsigned	NO	MUL			landscape handle of SS where the model is present
start_time	datetime	NO	MUL			The time when the outage started for the model
end_time	datetime	YES	MUL			The time when the outage got end for the model
outage_type	int(10) unsigned	NO	MUL			Type of outage on the model
zero_length_outage	tinyint(2)	NO	MUL	0		
start_event_key	bigint(20) unsigned	YES	MUL			

end_event_key	bigint(20) unsigned	YES	MUL			
legacy_outage_id	int(10) unsigned	YES				
legacy_outage_source	char(1)	YES				

Relations



modeloutage_notes

Description

Contains notes for each outage for the model.

Columns

Field	Type	Null	Key	Default	Extra	Comment
modeloutage_note_ID	bigint(20) unsigned	NO	PRI		auto_increment	unique ID for each model outage note
model_outage_ID	bigint(20) unsigned	NO	MUL			Outage ID corresponding to the outage ID in model table
notes	char(250)	YES				model outage notes

Relations

modeltype

Description

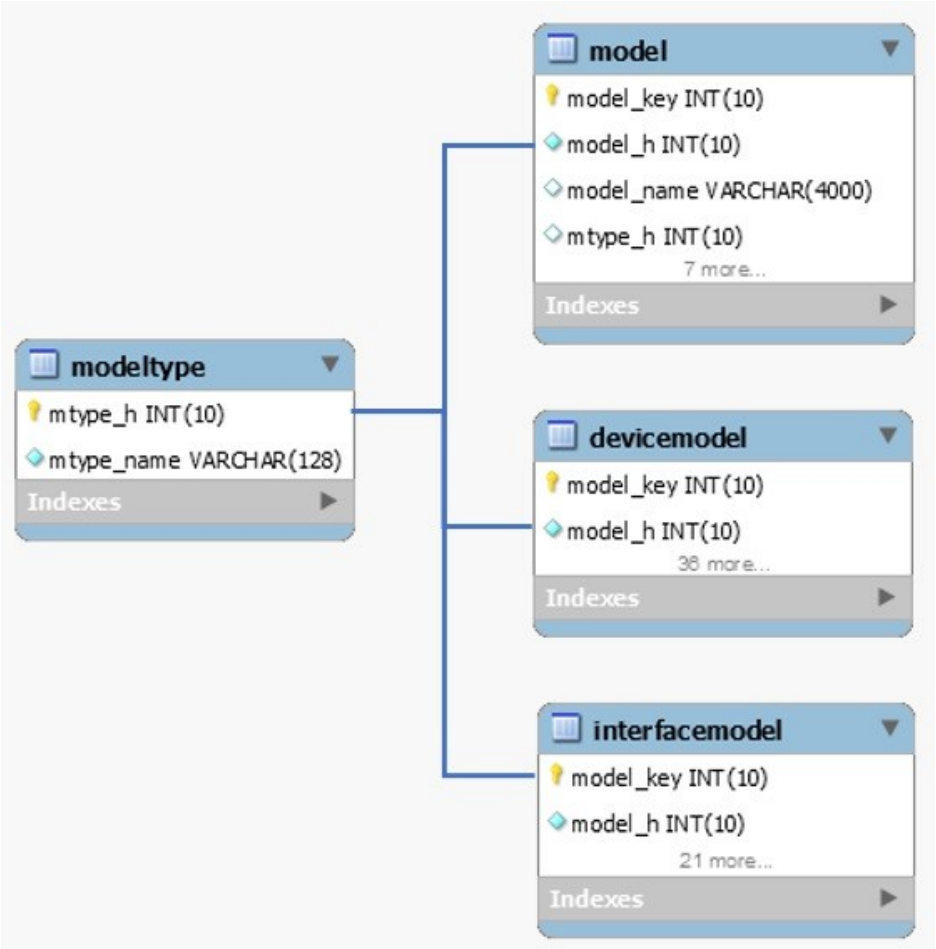
Contains Static list of all model type handles and their names.

The modeltype table is filled in as the Report Manager is starting up. The Report Manager contacts one of the SpectroSERVERs and sends a query requesting the handle and name for all device model types. Once this query is returned, the modeltype table is updated.

Columns

Field	Type	Null	Key	Default	Comment
mtype_h	int(10) unsigned	NO	PRI		Model type handle
mtype_name	varchar(128)	NO			Model Type Name

Relations



ncm_config

Description

Columns

Field	Type	Null	Key	Default	Extra	Comment
config_id	int(10) unsigned	NO	PRI		auto_increment	
config_text_id	int(10) unsigned	NO	MUL			
device_model_key	int(10) unsigned	NO				
change_time	datetime	YES				
trap_from	varchar(255)	NO				
trap_user	varchar(255)	NO				
trap_on	varchar(128)	YES				
lines_changed	int(11)	YES				
rel_lines_changed	int(11)	YES				
violated_policies	varchar(3000)	YES				

compliant_policies	varchar(3000)	YES				
spec_user_name	varchar(255)	YES				
comm_mode	varchar(128)	YES				
landscape	varchar(128)	YES				

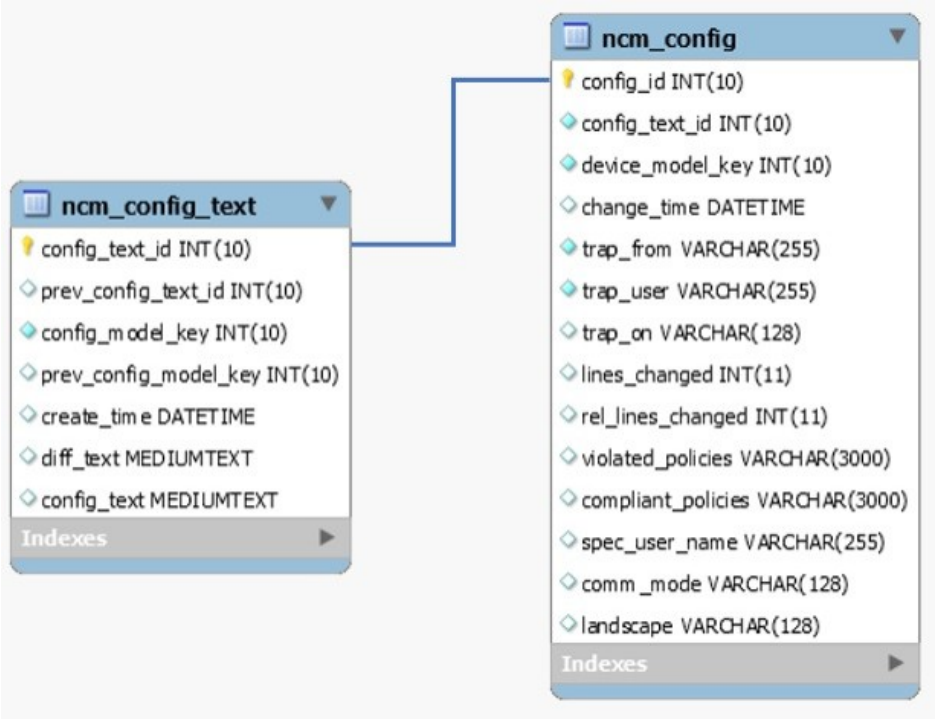
ncm_config_text

Description

Columns

Field	Type	Null	Key	Default	Extra	Comment
config_text_id	int(10) unsigned	NO	PRI		auto_increment	
prev_config_text_id	int(10) unsigned	YES	MUL			
config_model_key	int(10) unsigned	NO	MUL			
prev_config_model_key	int(10) unsigned	YES				
create_time	datetime	YES				
diff_text	mediumtext	YES				
config_text	mediumtext	YES				

Relations



ncm_event_type

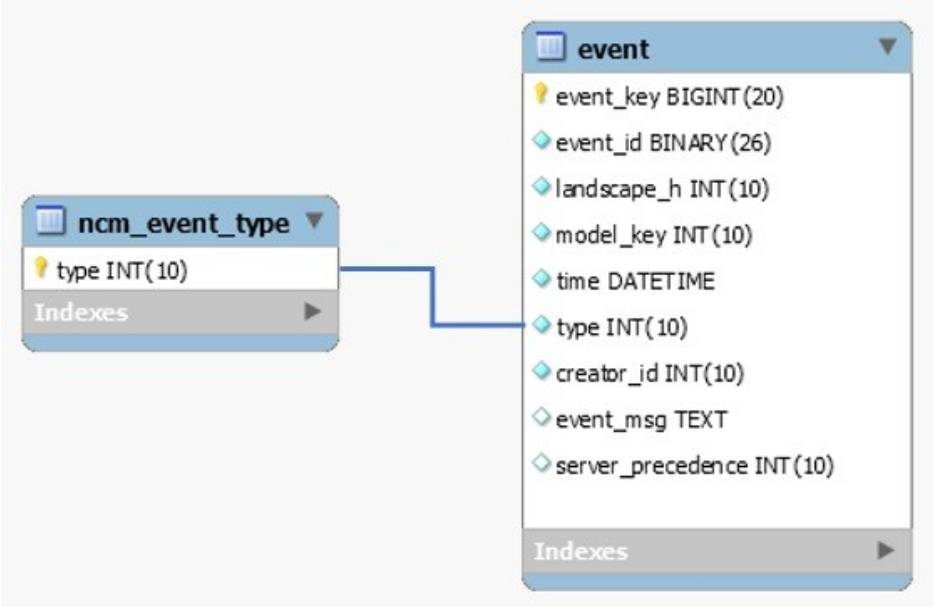
Description

Contains static list of all NCM event types

Columns

Field	Type	Null	Key	Default	Extra	Comment
type	int(10) unsigned	NO	PRI			NCM Event type

Relations



oc_user

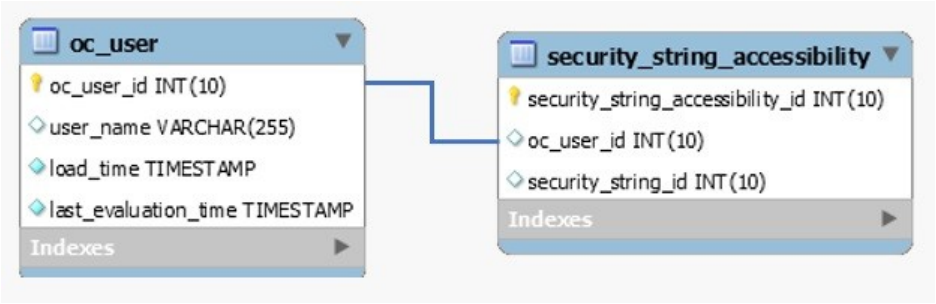
Description

This table contains list of oneclick users

Columns

Field	Type	Null	Key	Default	Extra	Comment
oc_user_id	int(10) unsigned	NO	PRI		auto_increment	
user_name	varchar(255)	YES	UNI			
load_time	timestamp	NO		CURRENT_TIMESTAMP		
last_evaluation_time	timestamp	NO		0000-00-00 00:00:00		

Relations



outagetype

Description

This table contains list outage types and descriptions

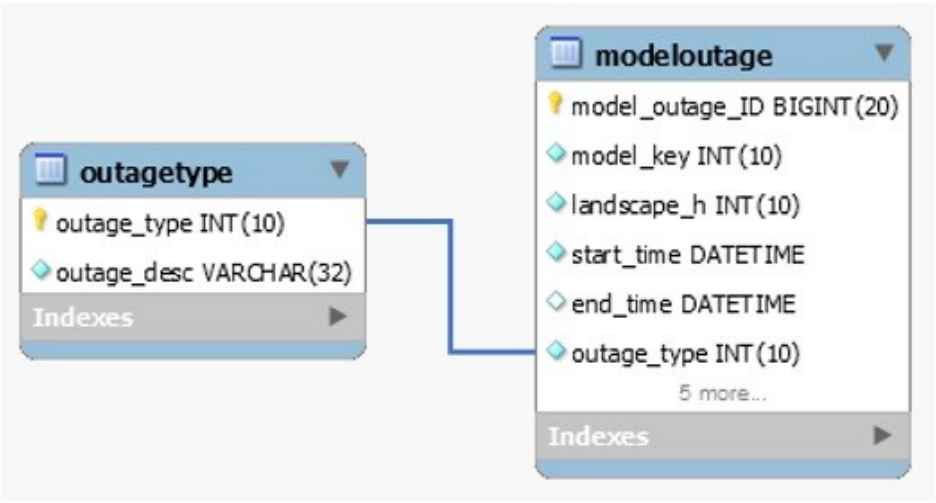
This table is filled in at the time of table creation. Outage types are pre-defined before any outages occur. Table records include:

outage_type	outage_desc
0	Initial
1	Unplanned
2	Planned
3	Exempt

Columns

Field	Type	Null	Key	Default	Extra	Comment
outage_type	int(10) unsigned	NO	PRI			
outage_desc	varchar(32)	NO				

Relations



pcause

Description

This table provides a mapping of cause codes to their titles. It is populated as each new cause is encountered by the alarm handler.

Columns

Field	Type	Null	Key	Default	Extra	Comment
cause_id	int(10) unsigned	NO	PRI			
title	varchar(100)	YES				

performance

Description

This table contains MySQL configuration settings for SRM DB, mostly static data

Columns

Field	Type	Null	Key	Default	Extra	Comment
id	int(10) unsigned	NO	PRI		auto_increment	
runcode	char(255)	YES				
runtime	datetime	YES				
memused	int(10) unsigned	NO				
memfree	int(10) unsigned	NO				
Aborted_clients	int(10) unsigned	NO		0		
Aborted_connects	int(10) unsigned	NO		0		
Binlog_cache_disk_use	int(10) unsigned	NO		0		
Binlog_cache_use	int(10) unsigned	NO		0		
Bytes_received	int(10) unsigned	NO		0		
Bytes_sent	int(10) unsigned	NO		0		
Com_admin_commands	int(10) unsigned	NO		0		
Com_alter_db	int(10) unsigned	NO		0		
Com_alter_table	int(10) unsigned	NO		0		
Com_analyze	int(10) unsigned	NO		0		
Com_backup_table	int(10) unsigned	NO		0		
Com_begin	int(10) unsigned	NO		0		
Com_change_db	int(10) unsigned	NO		0		
Com_change_master	int(10) unsigned	NO		0		
Com_check	int(10) unsigned	NO		0		
Com_checksum	int(10) unsigned	NO		0		
Com_commit	int(10) unsigned	NO		0		
Com_create_db	int(10) unsigned	NO		0		
Com_create_function	int(10) unsigned	NO		0		
Com_create_index	int(10) unsigned	NO		0		

Com_create_table	int(10) unsigned	NO		0		
Com_dealloc_sql	int(10) unsigned	NO		0		
Com_delete	int(10) unsigned	NO		0		
Com_delete_multi	int(10) unsigned	NO		0		
Com_do	int(10) unsigned	NO		0		
Com_drop_db	int(10) unsigned	NO		0		
Com_drop_function	int(10) unsigned	NO		0		
Com_drop_index	int(10) unsigned	NO		0		
Com_drop_table	int(10) unsigned	NO		0		
Com_drop_user	int(10) unsigned	NO		0		
Com_execute_sql	int(10) unsigned	NO		0		
Com_flush	int(10) unsigned	NO		0		
Com_grant	int(10) unsigned	NO		0		
Com_ha_close	int(10) unsigned	NO		0		
Com_ha_open	int(10) unsigned	NO		0		
Com_ha_read	int(10) unsigned	NO		0		
Com_help	int(10) unsigned	NO		0		
Com_insert	int(10) unsigned	NO		0		
Com_insert_select	int(10) unsigned	NO		0		
Com_kill	int(10) unsigned	NO		0		
Com_load	int(10) unsigned	NO		0		
Com_load_master_data	int(10) unsigned	NO		0		
Com_load_master_table	int(10) unsigned	NO		0		
Com_lock_tables	int(10) unsigned	NO		0		
Com_optimize	int(10) unsigned	NO		0		
Com_preload_keys	int(10) unsigned	NO		0		
Com_prepare_sql	int(10) unsigned	NO		0		
Com_purge	int(10) unsigned	NO		0		
Com_purge_before_date	int(10) unsigned	NO		0		
Com_rename_table	int(10) unsigned	NO		0		
Com_repair	int(10) unsigned	NO		0		
Com_replace	int(10) unsigned	NO		0		
Com_replace_select	int(10) unsigned	NO		0		
Com_reset	int(10) unsigned	NO		0		
Com_restore_table	int(10) unsigned	NO		0		
Com_revoke	int(10) unsigned	NO		0		
Com_revoke_all	int(10) unsigned	NO		0		
Com_rollback	int(10) unsigned	NO		0		
Com_savepoint	int(10) unsigned	NO		0		
Com_select	int(10) unsigned	NO		0		
Com_set_option	int(10) unsigned	NO		0		

Com_show_binlog_events	int(10) unsigned	NO		0		
Com_show_binlogs	int(10) unsigned	NO		0		
Com_show_charsets	int(10) unsigned	NO		0		
Com_show_collations	int(10) unsigned	NO		0		
Com_show_column_types	int(10) unsigned	NO		0		
Com_show_create_db	int(10) unsigned	NO		0		
Com_show_create_table	int(10) unsigned	NO		0		
Com_show_databases	int(10) unsigned	NO		0		
Com_show_errors	int(10) unsigned	NO		0		
Com_show_fields	int(10) unsigned	NO		0		
Com_show_grants	int(10) unsigned	NO		0		
Com_show_innodb_status	int(10) unsigned	NO		0		
Com_show_keys	int(10) unsigned	NO		0		
Com_show_logs	int(10) unsigned	NO		0		
Com_show_master_status	int(10) unsigned	NO		0		
Com_show_new_master	int(10) unsigned	NO		0		
Com_show_open_tables	int(10) unsigned	NO		0		
Com_show_privileges	int(10) unsigned	NO		0		
Com_show_processlist	int(10) unsigned	NO		0		
Com_show_slave_hosts	int(10) unsigned	NO		0		
Com_show_slave_status	int(10) unsigned	NO		0		
Com_show_status	int(10) unsigned	NO		0		
Com_show_storage_engines	int(10) unsigned	NO		0		
Com_show_tables	int(10) unsigned	NO		0		
Com_show_variables	int(10) unsigned	NO		0		
Com_show_warnings	int(10) unsigned	NO		0		
Com_slave_start	int(10) unsigned	NO		0		
Com_slave_stop	int(10) unsigned	NO		0		
Com_truncate	int(10) unsigned	NO		0		
Com_unlock_tables	int(10) unsigned	NO		0		
Com_update	int(10) unsigned	NO		0		
Com_update_multi	int(10) unsigned	NO		0		
Connections	int(10) unsigned	NO		0		
Created_tmp_disk_tables	int(10) unsigned	NO		0		
Created_tmp_files	int(10) unsigned	NO		0		
Created_tmp_tables	int(10) unsigned	NO		0		
Delayed_errors	int(10) unsigned	NO		0		
Delayed_insert_threads	int(10) unsigned	NO		0		
Delayed_writes	int(10) unsigned	NO		0		
Flush_commands	int(10) unsigned	NO		0		
Handler_commit	int(10) unsigned	NO		0		

Handler_delete	int(10) unsigned	NO		0		
Handler_discover	int(10) unsigned	NO		0		
Handler_read_first	int(10) unsigned	NO		0		
Handler_read_key	int(10) unsigned	NO		0		
Handler_read_next	int(10) unsigned	NO		0		
Handler_read_prev	int(10) unsigned	NO		0		
Handler_read_rnd	int(10) unsigned	NO		0		
Handler_read_rnd_next	int(10) unsigned	NO		0		
Handler_rollback	int(10) unsigned	NO		0		
Handler_update	int(10) unsigned	NO		0		
Handler_write	int(10) unsigned	NO		0		
Key_blocks_not_flushed	int(10) unsigned	NO		0		
Key_blocks_unused	int(10) unsigned	NO		0		
Key_blocks_used	int(10) unsigned	NO		0		
Key_read_requests	int(10) unsigned	NO		0		
Key_reads	int(10) unsigned	NO		0		
Key_write_requests	int(10) unsigned	NO		0		
Key_writes	int(10) unsigned	NO		0		
Max_used_connections	int(10) unsigned	NO		0		
Not_flushed_delayed_rows	int(10) unsigned	NO		0		
Open_files	int(10) unsigned	NO		0		
Open_streams	int(10) unsigned	NO		0		
Open_tables	int(10) unsigned	NO		0		
Opened_tables	int(10) unsigned	NO		0		
Qcache_free_blocks	int(10) unsigned	NO		0		
Qcache_free_memory	int(10) unsigned	NO		0		
Qcache_hits	int(10) unsigned	NO		0		
Qcache_inserts	int(10) unsigned	NO		0		
Qcache_lowmem_prunes	int(10) unsigned	NO		0		
Qcache_not_cached	int(10) unsigned	NO		0		
Qcache_queries_in_cache	int(10) unsigned	NO		0		
Qcache_total_blocks	int(10) unsigned	NO		0		
Questions	int(10) unsigned	NO		0		
Select_full_join	int(10) unsigned	NO		0		
Select_full_range_join	int(10) unsigned	NO		0		
Select_range	int(10) unsigned	NO		0		
Select_range_check	int(10) unsigned	NO		0		
Select_scan	int(10) unsigned	NO		0		
Slave_open_temp_tables	int(10) unsigned	NO		0		
Slave_retried_transactions	int(10) unsigned	NO		0		
Slow_launch_threads	int(10) unsigned	NO		0		

Slow_queries	int(10) unsigned	NO		0		
Sort_merge_passes	int(10) unsigned	NO		0		
Sort_range	int(10) unsigned	NO		0		
Sort_rows	int(10) unsigned	NO		0		
Sort_scan	int(10) unsigned	NO		0		
Table_locks_immediate	int(10) unsigned	NO		0		
Table_locks_waited	int(10) unsigned	NO		0		
Threads_cached	int(10) unsigned	NO		0		
Threads_connected	int(10) unsigned	NO		0		
Threads_created	int(10) unsigned	NO		0		
Threads_running	int(10) unsigned	NO		0		
Uptime	int(10) unsigned	NO		0		
Com_show_ndb_status	int(10) unsigned	NO		0		
Com_show_triggers	int(10) unsigned	NO		0		
Com_stmt_close	int(10) unsigned	NO		0		
Com_stmt_execute	int(10) unsigned	NO		0		
Com_stmt_fetch	int(10) unsigned	NO		0		
Com_stmt_prepare	int(10) unsigned	NO		0		
Com_stmt_reset	int(10) unsigned	NO		0		
Com_stmt_send_long_data	int(10) unsigned	NO		0		
Com_xa_commit	int(10) unsigned	NO		0		
Com_xa_end	int(10) unsigned	NO		0		
Com_xa_prepare	int(10) unsigned	NO		0		
Com_xa_recover	int(10) unsigned	NO		0		
Com_xa_rollback	int(10) unsigned	NO		0		
Com_xa_start	int(10) unsigned	NO		0		
Handler_prepare	int(10) unsigned	NO		0		
Handler_savepoint	int(10) unsigned	NO		0		
Handler_savepoint_rollback	int(10) unsigned	NO		0		
Innodb_buffer_pool_pages_data	int(10) unsigned	NO		0		
Innodb_buffer_pool_pages_dirty	int(10) unsigned	NO		0		
Innodb_buffer_pool_pages_flushed	int(10) unsigned	NO		0		
Innodb_buffer_pool_pages_free	int(10) unsigned	NO		0		
Innodb_buffer_pool_pages_latched	int(10) unsigned	NO		0		
Innodb_buffer_pool_pages_misc	int(10) unsigned	NO		0		
Innodb_buffer_pool_pages_total	int(10) unsigned	NO		0		
Innodb_buffer_pool_read_ahead_rnd	int(10) unsigned	NO		0		
Innodb_buffer_pool_read_ahead_seq	int(10) unsigned	NO		0		
Innodb_buffer_pool_read_requests	int(10) unsigned	NO		0		
Innodb_buffer_pool_reads	int(10) unsigned	NO		0		
Innodb_buffer_pool_wait_free	int(10) unsigned	NO		0		

Innodb_buffer_pool_write_requests	int(10) unsigned	NO		0		
Innodb_data_fsyncs	int(10) unsigned	NO		0		
Innodb_data_pending_fsyncs	int(10) unsigned	NO		0		
Innodb_data_pending_reads	int(10) unsigned	NO		0		
Innodb_data_pending_writes	int(10) unsigned	NO		0		
Innodb_data_read	int(10) unsigned	NO		0		
Innodb_data_reads	int(10) unsigned	NO		0		
Innodb_data_writes	int(10) unsigned	NO		0		
Innodb_data_written	int(10) unsigned	NO		0		
Innodb_dblwr_pages_written	int(10) unsigned	NO		0		
Innodb_dblwr_writes	int(10) unsigned	NO		0		
Innodb_log_waits	int(10) unsigned	NO		0		
Innodb_log_write_requests	int(10) unsigned	NO		0		
Innodb_log_writes	int(10) unsigned	NO		0		
Innodb_os_log_fsyncs	int(10) unsigned	NO		0		
Innodb_os_log_pending_fsyncs	int(10) unsigned	NO		0		
Innodb_os_log_pending_writes	int(10) unsigned	NO		0		
Innodb_os_log_written	int(10) unsigned	NO		0		
Innodb_page_size	int(10) unsigned	NO		0		
Innodb_pages_created	int(10) unsigned	NO		0		
Innodb_pages_read	int(10) unsigned	NO		0		
Innodb_pages_written	int(10) unsigned	NO		0		
Innodb_row_lock_current_waits	int(10) unsigned	NO		0		
Innodb_row_lock_time	int(10) unsigned	NO		0		
Innodb_row_lock_time_avg	int(10) unsigned	NO		0		
Innodb_row_lock_time_max	int(10) unsigned	NO		0		
Innodb_row_lock_waits	int(10) unsigned	NO		0		
Innodb_rows_deleted	int(10) unsigned	NO		0		
Innodb_rows_inserted	int(10) unsigned	NO		0		
Innodb_rows_read	int(10) unsigned	NO		0		
Innodb_rows_updated	int(10) unsigned	NO		0		
Last_query_cost	int(10) unsigned	NO		0		
Ssl_accept_renegotiates	int(10) unsigned	NO		0		
Ssl_accepts	int(10) unsigned	NO		0		
Ssl_callback_cache_hits	int(10) unsigned	NO		0		
Ssl_client_connects	int(10) unsigned	NO		0		
Ssl_connect_renegotiates	int(10) unsigned	NO		0		
Ssl_ctx_verify_depth	int(10) unsigned	NO		0		
Ssl_ctx_verify_mode	int(10) unsigned	NO		0		
Ssl_default_timeout	int(10) unsigned	NO		0		
Ssl_finished_accepts	int(10) unsigned	NO		0		

Ssl_finished_connects	int(10) unsigned	NO		0		
Ssl_session_cache_hits	int(10) unsigned	NO		0		
Ssl_session_cache_misses	int(10) unsigned	NO		0		
Ssl_session_cache_overflows	int(10) unsigned	NO		0		
Ssl_session_cache_size	int(10) unsigned	NO		0		
Ssl_session_cache_timeouts	int(10) unsigned	NO		0		
Ssl_sessions_reused	int(10) unsigned	NO		0		
Ssl_used_session_cache_entries	int(10) unsigned	NO		0		
Ssl_verify_depth	int(10) unsigned	NO		0		
Ssl_verify_mode	int(10) unsigned	NO		0		
Tc_log_max_pages_used	int(10) unsigned	NO		0		
Tc_log_page_size	int(10) unsigned	NO		0		
Tc_log_page_waits	int(10) unsigned	NO		0		
Com_call_procedure	int(10) unsigned	YES				
Com_create_user	int(10) unsigned	YES				
prepared_stmt_count	int(10) unsigned	YES				
Com_assign_to_keycache	int(10) unsigned	NO		0		
Com_alter_db_upgrade	int(10) unsigned	NO		0		
Com_alter_event	int(10) unsigned	NO		0		
Com_alter_function	int(10) unsigned	NO		0		
Com_alter_procedure	int(10) unsigned	NO		0		
Com_alter_server	int(10) unsigned	NO		0		
Com_alter_tablespace	int(10) unsigned	NO		0		
Com_binlog	int(10) unsigned	NO		0		
Com_create_event	int(10) unsigned	NO		0		
Com_create_procedure	int(10) unsigned	NO		0		
Com_create_server	int(10) unsigned	NO		0		
Com_create_trigger	int(10) unsigned	NO		0		
Com_create_udf	int(10) unsigned	NO		0		
Com_create_view	int(10) unsigned	NO		0		
Com_drop_event	int(10) unsigned	NO		0		
Com_drop_procedure	int(10) unsigned	NO		0		
Com_drop_server	int(10) unsigned	NO		0		
Com_drop_trigger	int(10) unsigned	NO		0		
Com_drop_view	int(10) unsigned	NO		0		
Com_empty_query	int(10) unsigned	NO		0		
Com_install_plugin	int(10) unsigned	NO		0		
Com_release_savepoint	int(10) unsigned	NO		0		
Com_rename_user	int(10) unsigned	NO		0		
Com_rollback_to_savepoint	int(10) unsigned	NO		0		
Com_show_authors	int(10) unsigned	NO		0		

Com_show_contributors	int(10) unsigned	NO		0		
Com_show_create_event	int(10) unsigned	NO		0		
Com_show_create_func	int(10) unsigned	NO		0		
Com_show_create_proc	int(10) unsigned	NO		0		
Com_show_create_trigger	int(10) unsigned	NO		0		
Com_show_engine_logs	int(10) unsigned	NO		0		
Com_show_engine_mutex	int(10) unsigned	NO		0		
Com_show_engine_status	int(10) unsigned	NO		0		
Com_show_events	int(10) unsigned	NO		0		
Com_show_function_status	int(10) unsigned	NO		0		
Com_show_plugins	int(10) unsigned	NO		0		
Com_show_procedure_status	int(10) unsigned	NO		0		
Com_show_profile	int(10) unsigned	NO		0		
Com_show_profiles	int(10) unsigned	NO		0		
Com_show_table_status	int(10) unsigned	NO		0		
Com_stmt_reprepare	int(10) unsigned	NO		0		
Com_uninstall_plugin	int(10) unsigned	NO		0		
Open_table_definitions	int(10) unsigned	NO		0		
Opened_files	int(10) unsigned	NO		0		
Opened_table_definitions	int(10) unsigned	NO		0		
Queries	int(10) unsigned	NO		0		
Uptime_since_flush_status	int(10) unsigned	NO		0		

portduplex

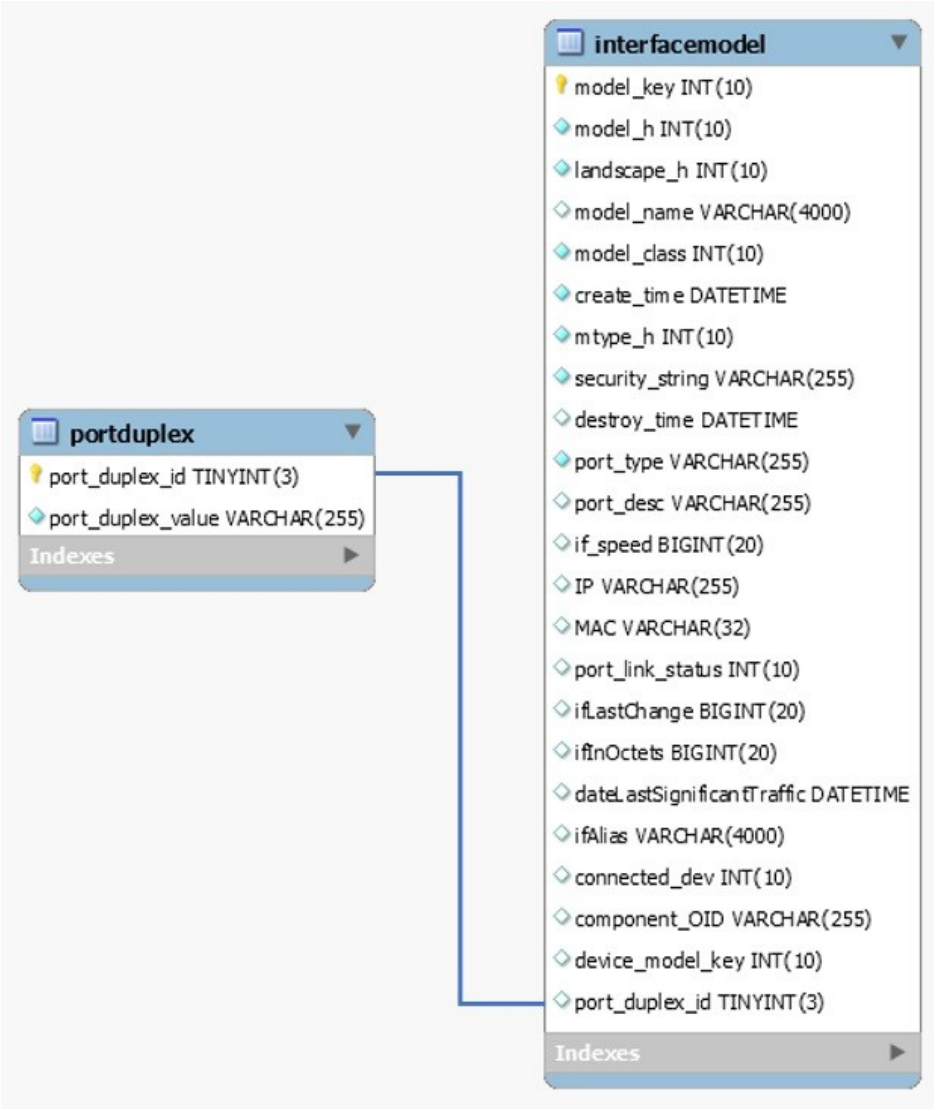
Description

This table contains port duplex IDs and corresponding values

Columns

Field	Type	Null	Key	Default	Extra	Comment
port_duplex_id	tinyint(3) unsigned	NO	PRI			
port_duplex_value	varchar(255)	NO				

Relations



registry

Description

This table provides a storage area for SRM to maintain different properties and attributes of the SRM application. The table stores generic mappings using key/value pairs. Registry entries can have one of the following types:

type id	type name
0	Boolean
1	String
2	Hidden
3	List
4	List Entry

Columns

Field	Type	Null	Key	Default	Extra	Comment
reg_user	varchar(255)	NO	PRI			
OneClickServerEntry	varchar(255)	NO				
SRMPollPeriod	tinyint(4)	NO				
SRMPollStartTime	tinyint(4)	NO				
SRMPollEndTime	tinyint(4)	NO				
IFIdleThreshold	int(11)	NO		0		
ServersList	text	NO				
CrystalInstallRoot	varchar(255)	NO				
CrystalCommonFiles	varchar(255)	NO				
CrystalHome	varchar(255)	NO				
CrystalPassword	varbinary(255)	NO				
ReportSuppressedAlarms	enum('false','true')	NO		false		
BOPassword	varbinary(255)	NO				
BOCommonFiles	varchar(255)	NO				
BOHome	varchar(255)	NO				
BOInstallRoot	varchar(255)	NO				
ConvertedReportsToBOXI	enum('false','true')	NO				
ArchivalRetentionDays	int(11) unsigned	YES		90		
event_archival_retention_days	int(11) unsigned	NO		90		
DataRetentionPolicy	enum('all data','archive','purge')	NO				
customlogopath	text	YES				
isReportingReady	enum('false','true')	NO		false		
handler_batch_size	int(10) unsigned	NO		1000		
event_poller_processing_interval	int(10) unsigned	NO		60		
MonitorSRM	enum('false','true')	YES		true		
SRM_Model	int(11) unsigned	YES				
BOUser	varchar(255)	NO				
BOHost	varchar(255)	NO				
BOPort	mediumint(9)	NO				
BOAuthType	varchar(255)	NO				
BOTomcatPort	mediumint(9)	NO				
CrystalReportsUser	varchar(255)	NO				
CrystalReportsPassword	varbinary(255)	NO				
CrystalReportsHost	varchar(255)	NO				
DBHost	varchar(255)	NO				
BOInfoView	varchar(255)	NO				
BOCmc	varchar(255)	NO				
BOInfoViewCredentials	varchar(255)	NO				

BOInfoViewAuthType	varchar(255)	NO				
UniverseUser	varchar(255)	NO				
UniversePassword	varbinary(255)	NO				
is_security_enabled	enum('false','true')	YES		false		
DefaultBOXIUserPassword	varbinary(255)	NO				
isPerformanceMonitorEnabled	enum('false','true')	NO		false		
install_version	varchar(128)	NO		9.4.2.1.1		
PollTaskMaxTardy	smallint(5) unsigned	YES				
BOSharedSecret	varbinary(255)	YES				

schemaversion

Description

This table stores data related to SRM schema versioning, change log etc. for one-time changes. Table 'schemaversion_recurring' table stores recurring changes.

Internal table, do not edit

Columns

Field	Type	Null	Key	Default	Extra	Comment
change_id	int(10) unsigned	NO	PRI		auto_increment	
session_id	int(10) unsigned	YES				
name	varchar(255)	NO	UNI			
category	varchar(255)	NO				
schema_comments	varchar(255)	NO				
major	tinyint(4)	NO				
minor	tinyint(4)	NO				
service_pack	tinyint(4)	NO				
start_time	datetime	YES				
end_time	datetime	YES				
duration_secs	int(11)	YES				
state	enum('A','N','F')	YES		A		
state_details	text	YES				

schemaversion_recurring

Description

This table stores data related to SRM schema versioning, change log etc. of recurring changes.

Internal table, do not edit

Columns

Field	Type	Null	Key	Default	Extra	Comment
change_id	int(10) unsigned	NO	PRI		auto_increment	

session_id	int(10) unsigned	YES				
name	varchar(255)	NO	MUL			
category	varchar(255)	NO				
schema_comments	varchar(255)	NO				
major	tinyint(4)	NO				
minor	tinyint(4)	NO				
service_pack	tinyint(4)	NO				
start_time	datetime	YES				
end_time	datetime	YES				
duration_secs	int(11)	YES				
State	enum('A','N','F')	YES		A		
state_details	text	YES				

schemaversion_session

Description

Table stores data related to SRM schema versioning session information. Internal table

Columns

Field	Type	Null	Key	Default	Extra	Comment
session_id	int(10) unsigned	NO	PRI		auto_increment	
session_name	varchar(255)	NO				
start_time	datetime	NO				
end_time	datetime	YES				
duration_secs	int(11)	YES				
state	enum('A','F')	YES				

security_string

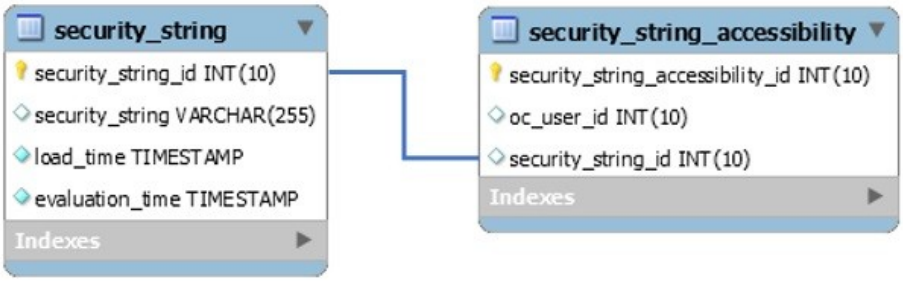
Description

This table stores list of unique security string IDs and security strings for models in reporting DB.

Columns

Field	Type	Null	Key	Default	Extra	Comment
security_string_id	int(10) unsigned	NO	PRI		auto_increment	
security_string	varchar(255)	YES	UNI			
load_time	timestamp	NO		CURRENT_TIMESTAMP		
evaluation_time	timestamp	NO		0000-00-00 00:00:00		

Relations



security_string_accessibility

Description

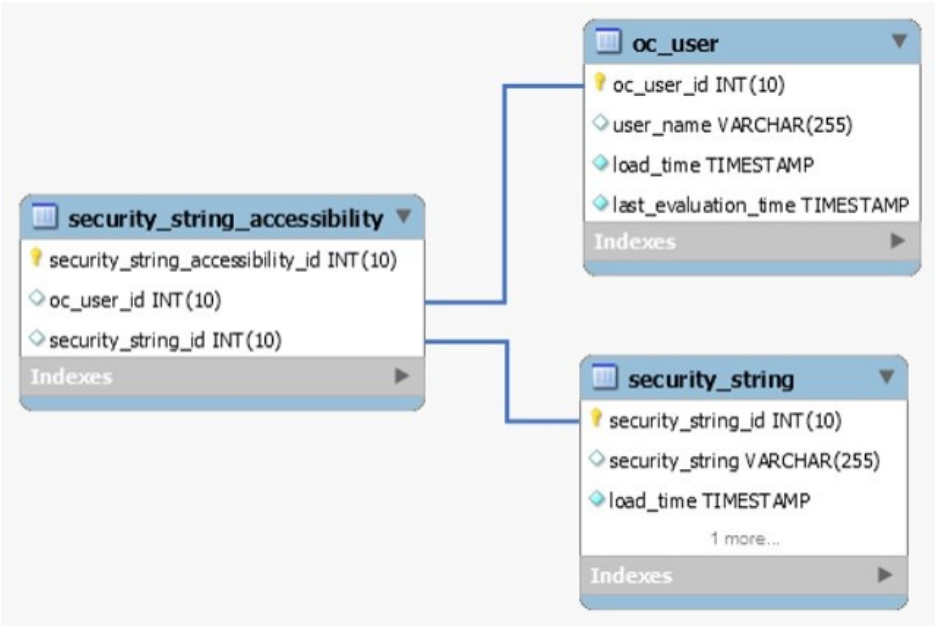
This table stores security string accessibility information. Maps Oneclick user ID to security string ID.

Internal table, do not edit

Columns

Field	Type	Null	Key	Default	Extra	Comment
security_string_accessibility_id	int(10) unsigned	NO	PRI		auto_increment	
oc_user_id	int(10) unsigned	YES	MUL			
security_string_id	int(10) unsigned	YES				

Relations



sm_attributes

Description

Service Manager attributes table - static data

Columns

Column name	Type	Null	Key	Default	Comment
attrID	int(10) unsigned	NO	PRI		Attribute ID of service manager model class
attrName	varchar(255)	NO			Attribute name of service manager model class

Relations

sm_customermhs

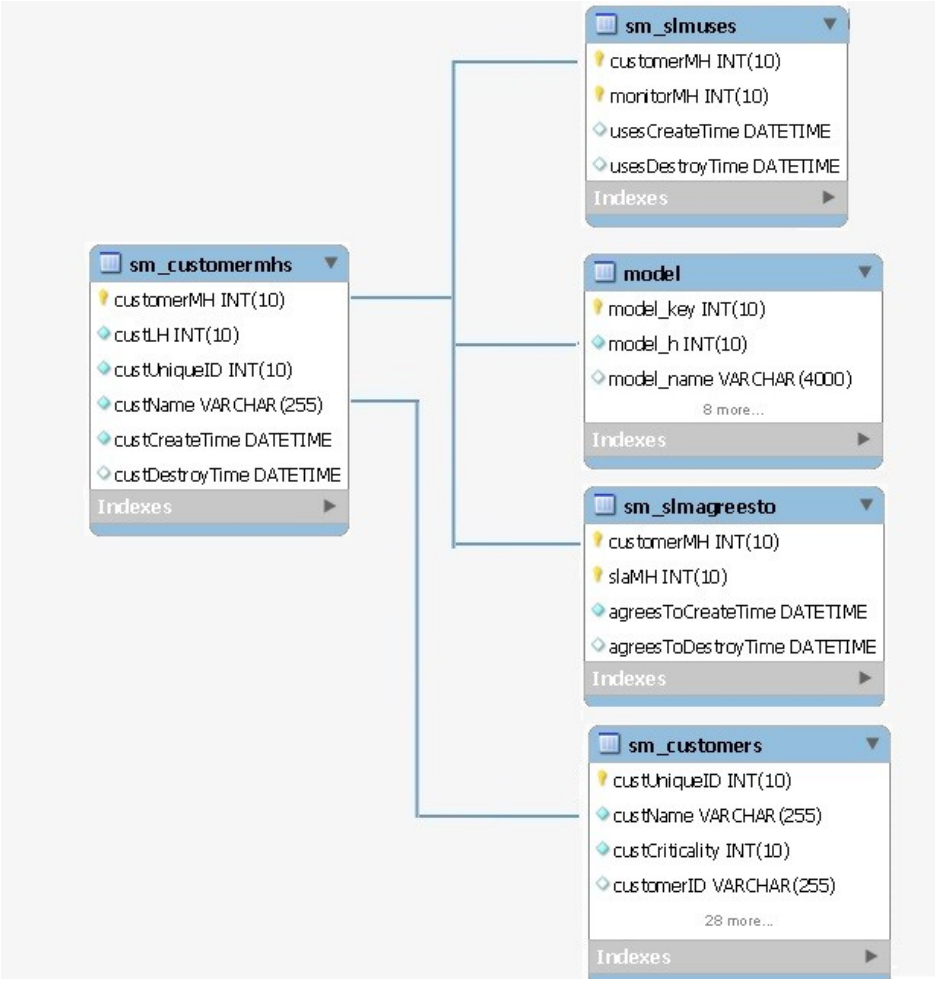
Description

This table stores service manager - customer model handles

Columns

Field	Type	Null	Key	Default	Comment
customerMH	int(10) unsigned	NO	PRI		Customer model handle
custLH	int(10) unsigned	NO			
custUniqueID	int(10) unsigned	NO	MUL		UniqueID of the customer
custName	varchar(255)	NO			customer name
custCreateTime	datetime	NO			customer create time
custDestroyTime	datetime	YES			customer destroy time

Relations



sm_customers

Description

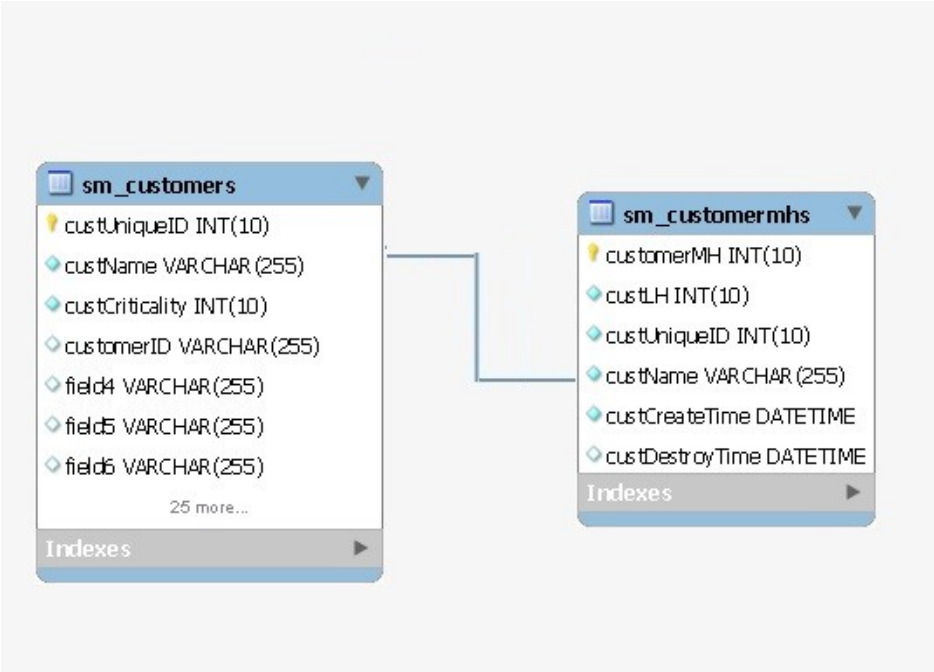
This table stores Service manager – customer details

Columns

Field	Type	Null	Key	Default	Extra	Comment
custUniqueID	int(10) unsigned	NO	PRI		auto_increment	Customer unique ID
custName	varchar(255)	NO	UNI			Customer Name
custCriticality	int(10) unsigned	NO				Customer criticality
customerID	varchar(255)	YES				Customer ID
field4	varchar(255)	YES				
field5	varchar(255)	YES				
field6	varchar(255)	YES				
field7	varchar(255)	YES				

primContName	varchar(255)	YES				Primary contact name
primContTitle	varchar(255)	YES				Primary contact title
primContLocation	varchar(255)	YES				primary contact location
primEmail	varchar(255)	YES				Primary Email
primPhone	varchar(255)	YES				Primary Phone
primMobile	varchar(255)	YES				Primary Mobile
primPager	varchar(255)	YES				Primary Pager
primFax	varchar(255)	YES				Primary FAX
primUserDef1	varchar(255)	YES				
primUserDef2	varchar(255)	YES				
primUserDef3	varchar(255)	YES				
primUserDef4	varchar(255)	YES				
secContName	varchar(255)	YES				Secondary Contact Name
secContTitle	varchar(255)	YES				
secContLocation	varchar(255)	YES				
secEmail	varchar(255)	YES				
secPhone	varchar(255)	YES				
secMobile	varchar(255)	YES				
secPager	varchar(255)	YES				
secFax	varchar(255)	YES				
secUserDef1	varchar(255)	YES				
secUserDef2	varchar(255)	YES				
secUserDef3	varchar(255)	YES				
secUserDef4	varchar(255)	YES				

Relations



sm_guaranteeoutages

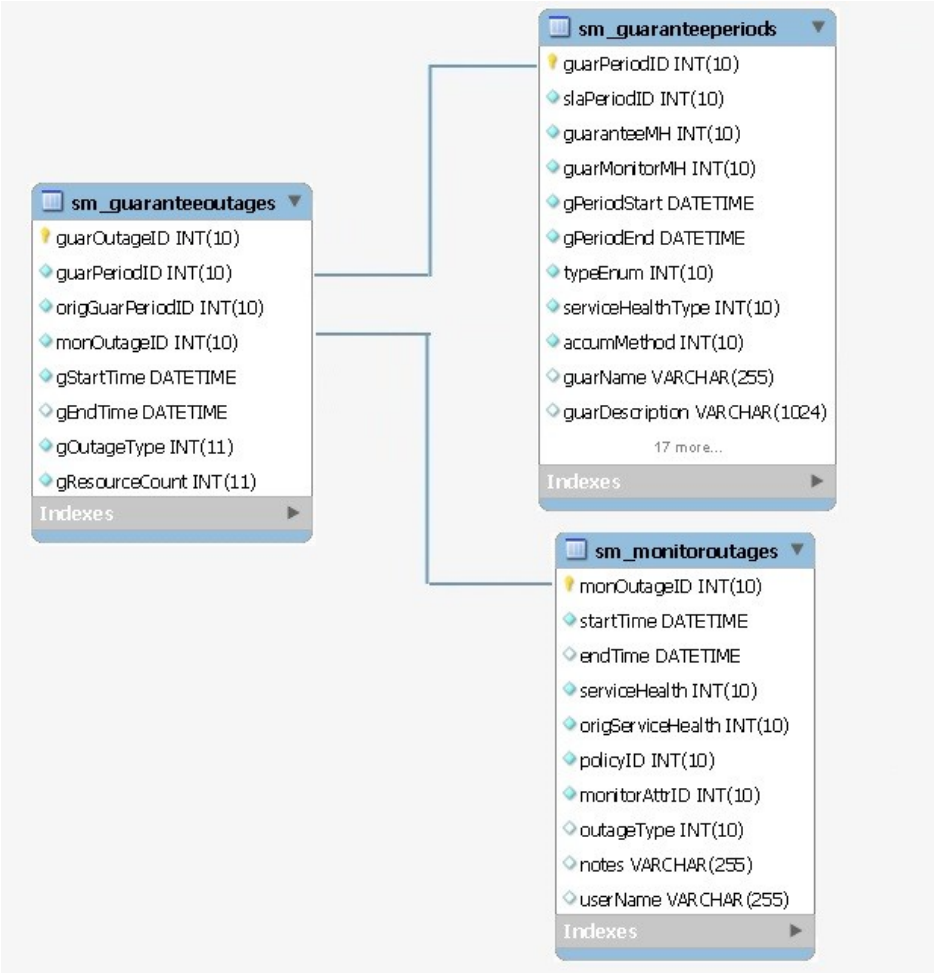
Description

This table stores Service manager - guaranteed outages

Columns

Field	Type	Null	Key	Default	Extra	Comment
guarOutageID	int(10) unsigned	NO	PRI		auto_increment	Guarantee outage ID
guarPeriodID	int(10) unsigned	NO				Guarantee period ID
origGuarPeriodID	int(10) unsigned	NO				Original guarantee period ID
monOutageID	int(10) unsigned	NO				Monitoring Outage ID
gStartTime	datetime	NO				Outage start time
gEndTime	datetime	YES				Outable end time
gOutageType	int(11)	NO				Type of Outage
gResourceCount	int(11)	NO				Resource count affected by outage

Relations



sm_guaranteeperiods

Description

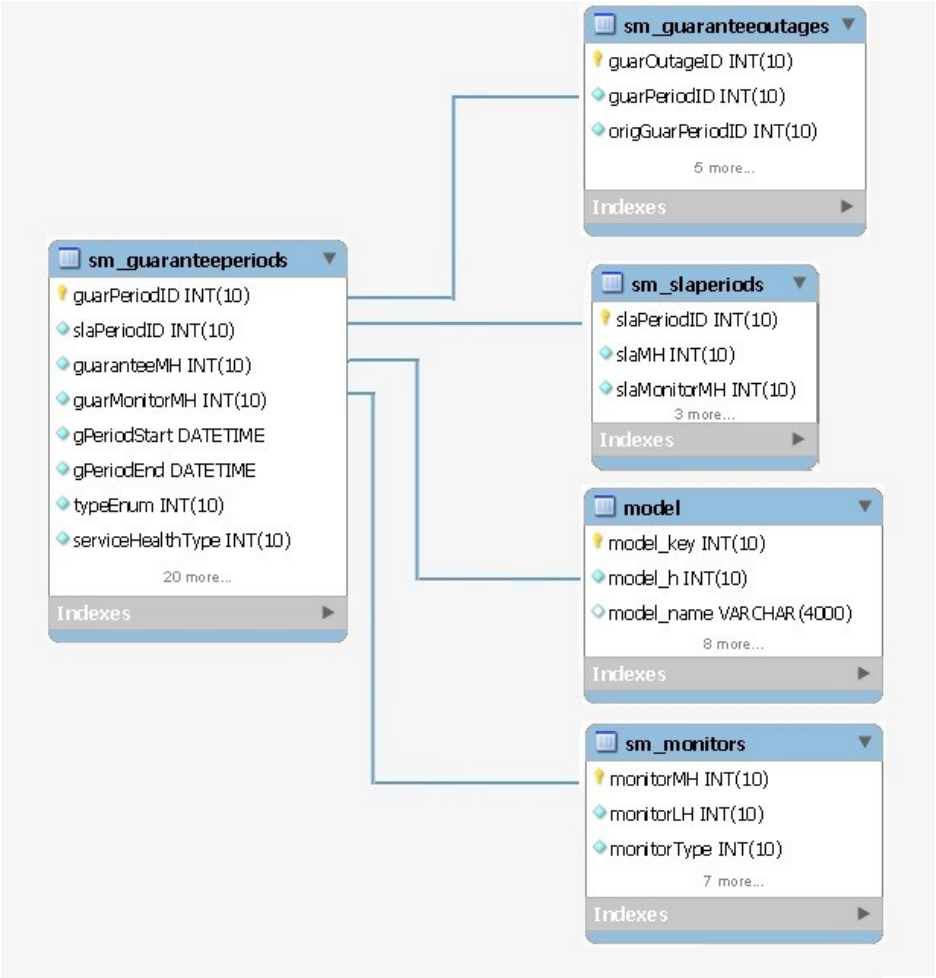
This table stores Service manager guarantee periods

Columns

Field	Type	Null	Key	Default	Extra	Comment
guarPeriodID	int(10) unsigned	NO	PRI		auto_increment	Guarantee period ID
slaPeriodID	int(10) unsigned	NO	MUL			SLA period ID
guaranteeMH	int(10) unsigned	NO				Guarantee model handle
guarMonitorMH	int(10) unsigned	NO				Guarantee monitor model handle
gPeriodStart	datetime	NO				Guarantee period start time
gPeriodEnd	datetime	NO				Guarantee period end time
typeEnum	int(10) unsigned	NO				Type of guarantee period

serviceHealthType	int(10) unsigned	NO				Type of service health
accumMethod	int(10) unsigned	NO				Guarantee period accumulation method
guarName	varchar(255)	YES				Guarantee name
guarDescription	varchar(1024)	YES				
outageTime	int(10) unsigned	NO				Outage time
outageCount	int(10) unsigned	NO				Outage count
activeTime	int(10) unsigned	NO				Time when the outage is active
majorThreshold	int(10) unsigned	NO				
criticalThreshold	int(10) unsigned	NO				
majorThresholdPercent	double	YES				
criticalThresholdPercent	double	YES				
guarStatus	int(10) unsigned	NO				
motValue	int(10) unsigned	NO				
motThreshold	int(10) unsigned	NO				
motStatus	int(10) unsigned	NO				
mttrValue	int(10) unsigned	NO				
mttrThreshold	int(10) unsigned	NO				
mttrStatus	int(10) unsigned	NO				
mtbfValue	int(10) unsigned	NO				
mtbfThreshold	int(10) unsigned	NO				
mtbfStatus	int(10) unsigned	NO				

Relations



sm_monitormaps

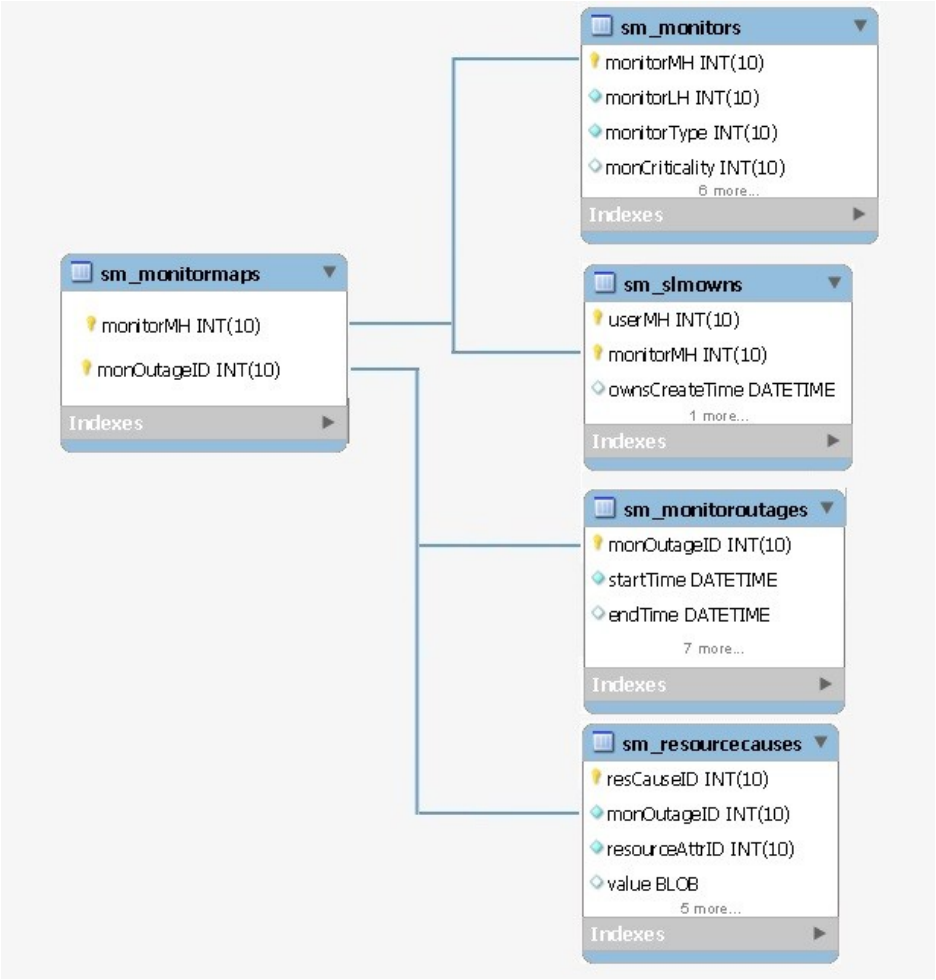
Description

This table stores the mapping of monitor model handle to outage ID

Columns

Field	Type	Null	Key	Default	Extra	Comment
monitorMH	int(10) unsigned	NO	PRI			Model handle of the monitor
monOutageID	int(10) unsigned	NO	PRI			Monitoring outage ID

Relations



sm_monitoroutages

Description

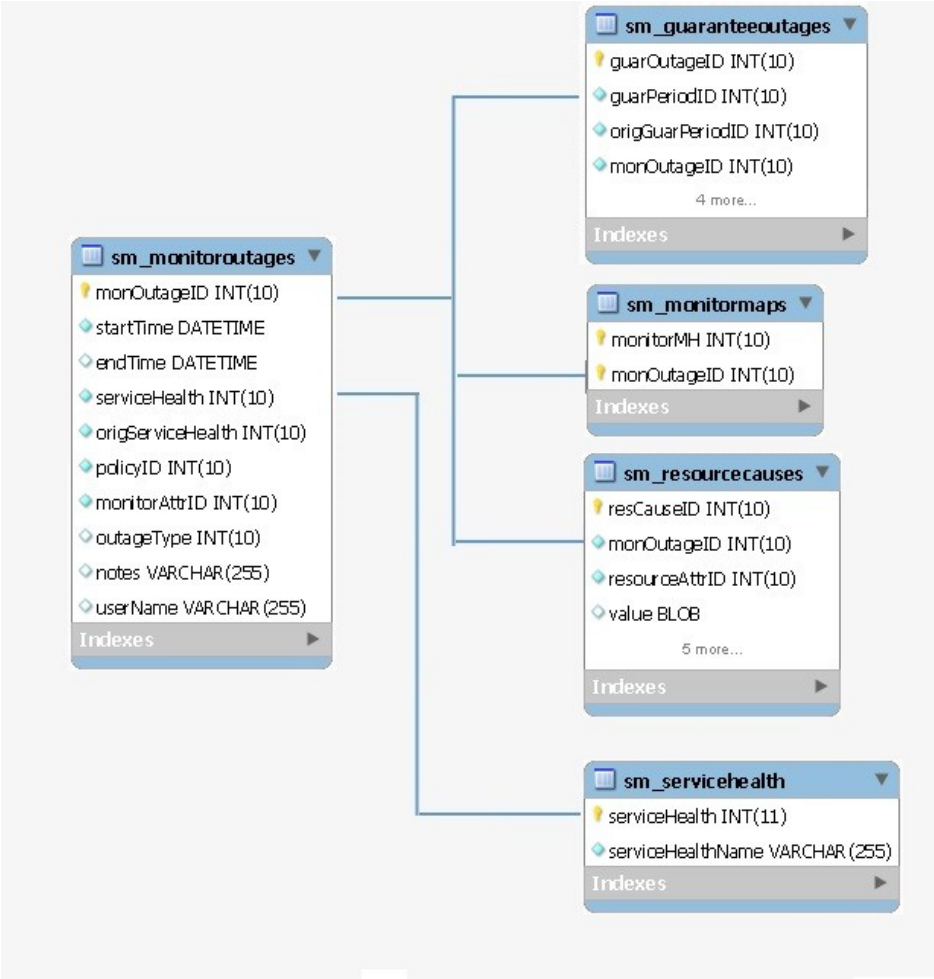
This table stores Service manager - Outages

Columns

Field	Type	Null	Key	Default	Extra	Comment
monOutageID	int(10) unsigned	NO	PRI		auto_increment	Monitoring outage ID
startTime	datetime	NO				Outage starting time
endTime	datetime	YES				Outage ending time
serviceHealth	int(10) unsigned	NO	MUL			Service health
origServiceHealth	int(10) unsigned	NO				Default service health
policyID	int(10) unsigned	NO				policy ID

monitorAttrID	int(10) unsigned	NO				Monitor attribute ID
outageType	int(10) unsigned	YES				Type of outage
notes	varchar(255)	YES				'notes' for the outage
userName	varchar(255)	YES				User name

Relations



sm_monitors

Description

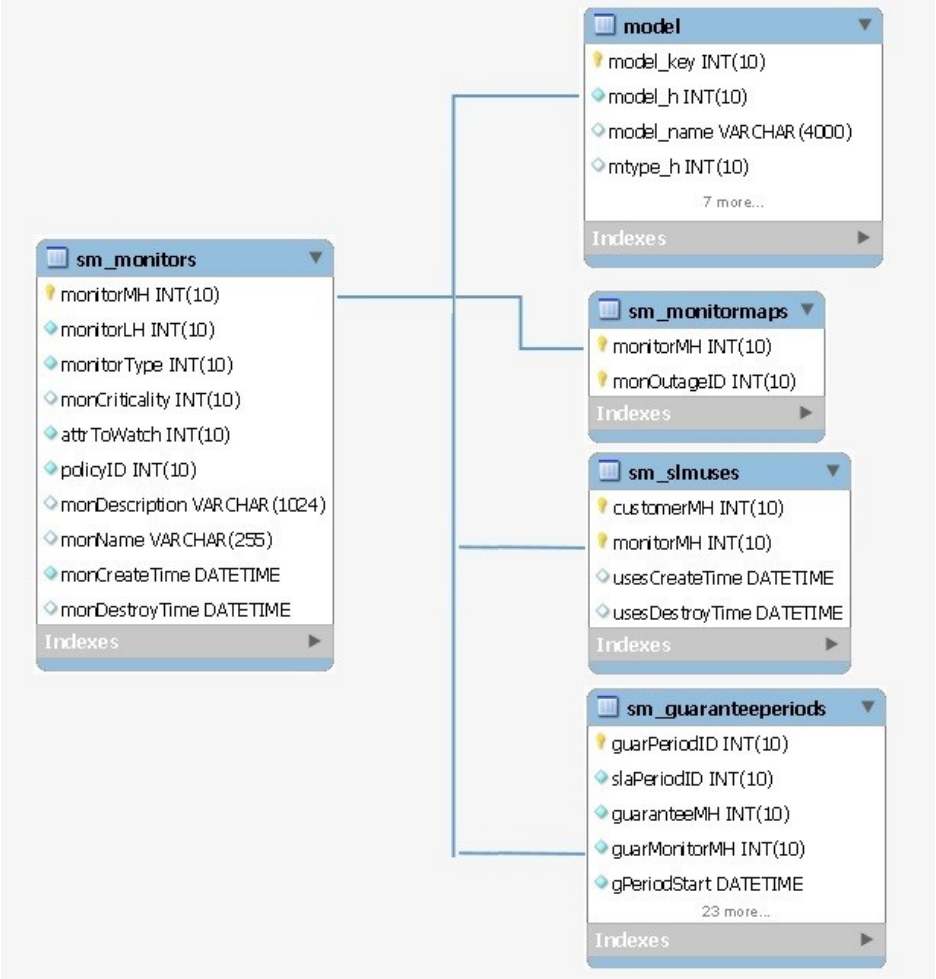
This table stores data of monitor model

Columns

Field	Type	Null	Key	Default	Comment
monitorMH	int(10) unsigned	NO	PRI		Model handle of the monitor

monitorLH	int(10) unsigned	NO			Landscape handle of the monitor
monitorType	int(10) unsigned	NO			type of monitor
monCriticality	int(10) unsigned	YES			Monitor criticality
attrToWatch	int(10) unsigned	NO			Attribute to watch
policyID	int(10) unsigned	NO			Policy ID
monDescription	varchar(1024)	YES			Description of the monitor
monName	varchar(255)	YES			Name of the monitor
monCreateTime	datetime	NO			Monitor creation time
monDestroyTime	datetime	YES			Monitor destroy time

Relations



sm_periods

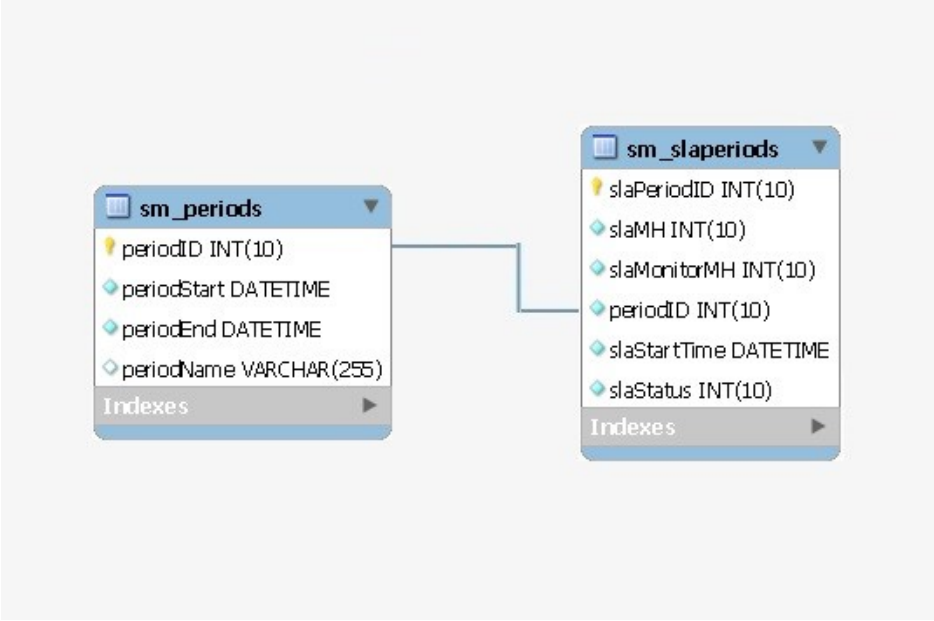
Description

Service Manager Gurantee period Information.

Columns

Field	Type	Null	Key	Default	Extra	Comment
periodID	int(10) unsigned	NO	PRI		auto_increment	Unique ID for each Guarantee period.
periodStart	datetime	NO				Guarantee Start period.
periodEnd	datetime	NO				Guarantee End period.
periodName	varchar(255)	YES				Guarantee period Name.

Relations



sm_policies

Description

Contains information about Service Manager policies.

Column

Field	Type	Null	Key	Default	Comment
policyName	varchar(255)	NO			Name of the Policy
policyID	int(11)	NO	PRI		Unique Policy ID for each service.
policyDescription	text	NO			Description of the Policy

Relations

sm_resourcecauses

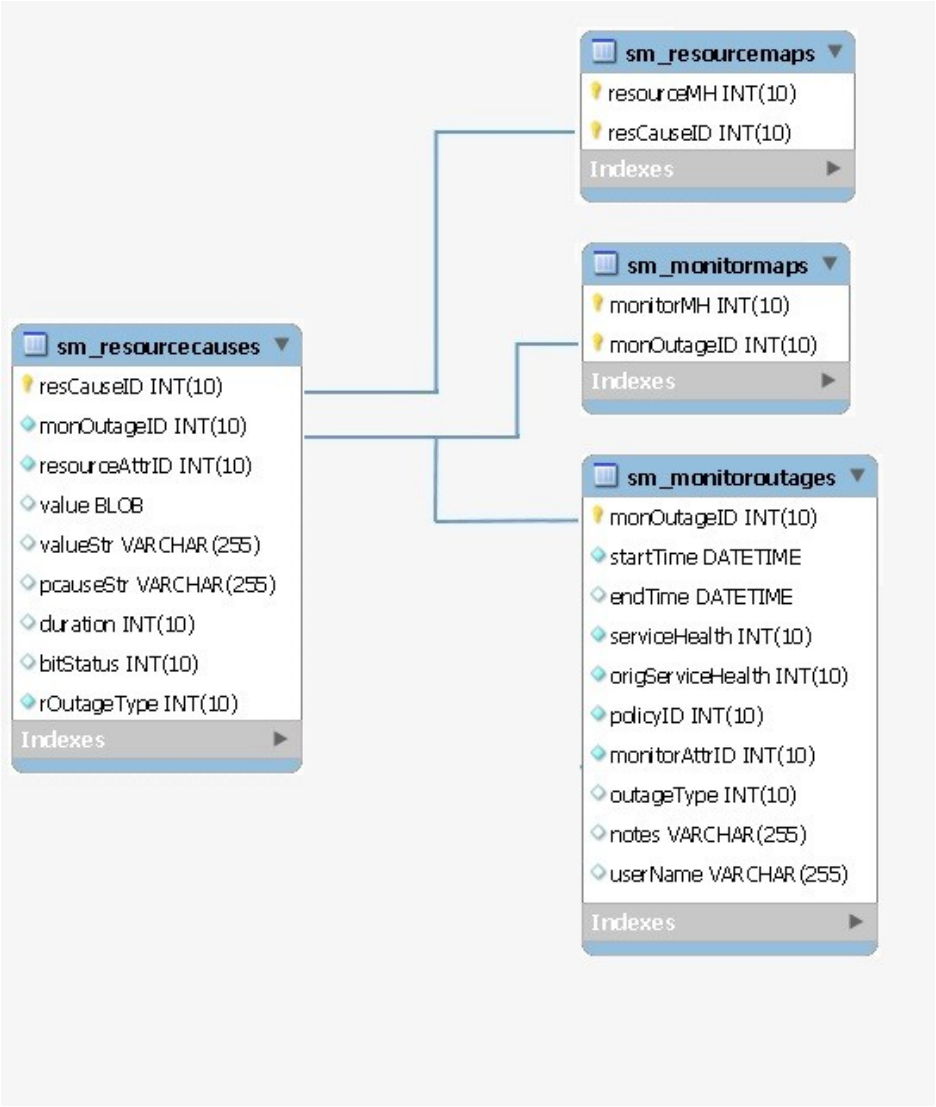
Description

The name of the table used to store the resource outages that make up each monitor outage. Each resource outage contains the model handle of the resource model that the outage was on, and the value that caused the monitor outage.

Column

Field	Type	Null	Key	Default	Extra	Comment
valueStr	varchar(255)	YES				Alarm status of the Resource.
value	blob	YES				Value that caused the monitor outage
rOutageType	int(10) unsigned	NO				Type of the Outage happened to the Resource.
resourceAttrID	int(10) unsigned	NO				Attribute ID of the Resource Monitor
resCauseID	int(10) unsigned	NO	PRI		auto_increment	Unique Cause ID of the Resource.
pcauseStr	varchar(255)	YES				Alarm Title of the Resource.
monOutageID	int(10) unsigned	NO	MUL			Unique Outage ID of the Resource Monitor
duration	int(10) unsigned	YES				Duration of the Event.
bitStatus	int(10) unsigned	YES				bitStatus

Relations



sm_resourcemaps

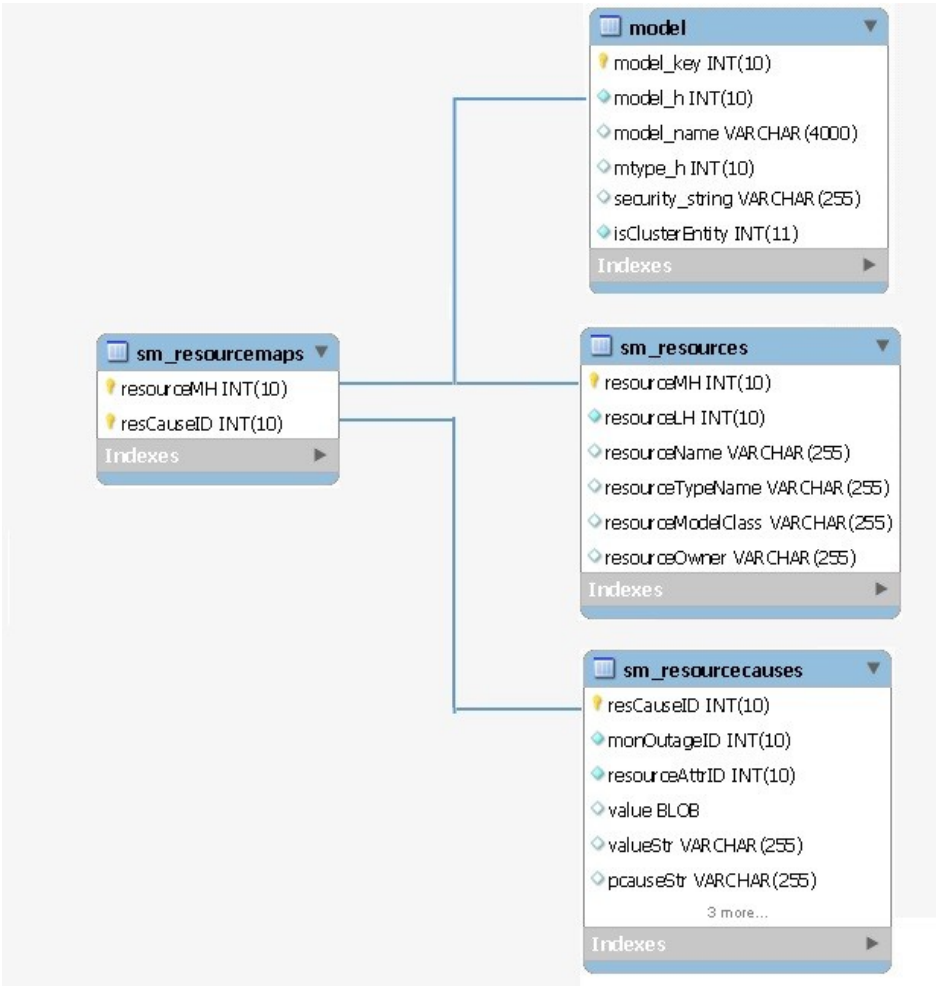
Description

The name of the table used to store the mapping between a resource cause and the model that caused it.

Column

Field	Type	Null	Key	Default	Comment
resourceMH	int(10) unsigned	NO	PRI		Unique Model_Handle of the Resource.
resCauseID	int(10) unsigned	NO	PRI		Unique Cause ID of the Resource.

Relations



sm_resources

Description

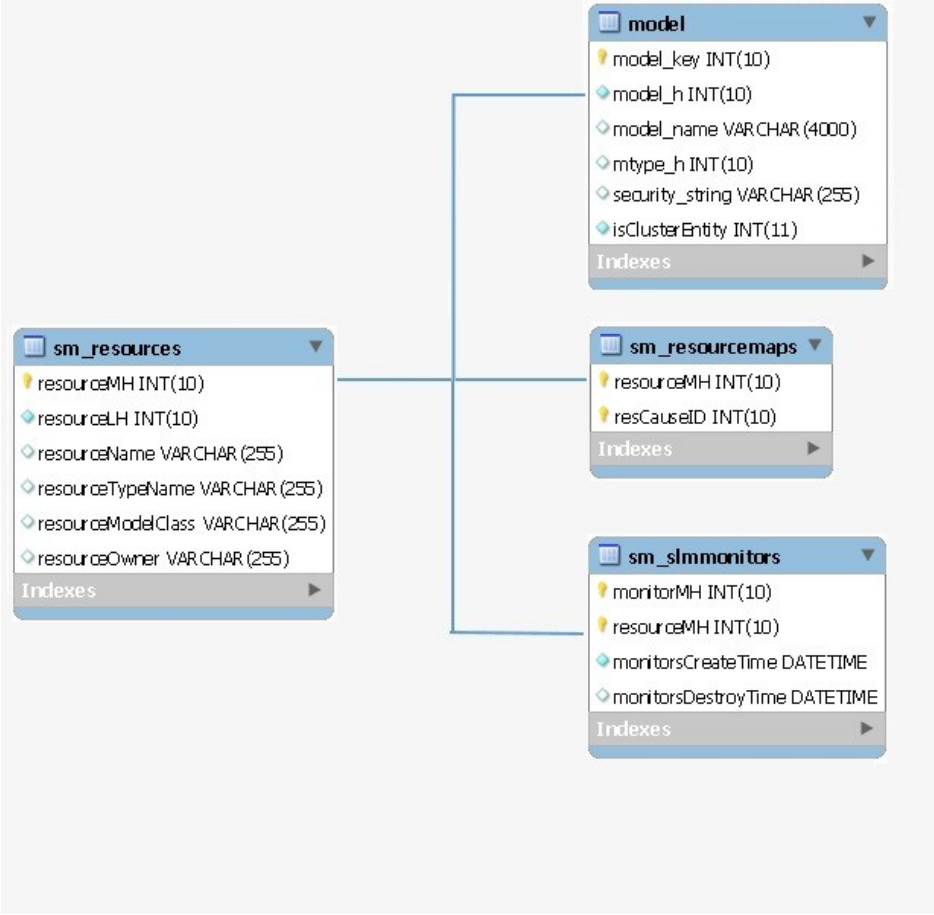
The name of the table that contains all the resources used by a service or attribute monitor.

Column

Field	Type	Null	Key	Default	Comment
resourceTypeName	varchar(255)	YES			Model_Type of the Resource.
resourceOwner	varchar(255)	YES			Owner of the resource
resourceName	varchar(255)	YES			Name of the Resource.
resourceModelClass	varchar(255)	YES			Model_Class of the Resource.
resourceMH	int(10) unsigned	NO	PRI		Unique Model_Handle of the Resource.

resourceLH	int(10) unsigned	NO			Landscape Handle of the Resource.

Relations



sm_schemaversion

Description

The name of the table that contains the current schema version of the SLM DB.

Column

Field	Type	Null	Key	Default	Comment
versionID	varchar(255)	NO	PRI		Unique version ID of the Schema.
comments	varchar(255)	NO			Brief description on the installed version.

Relations

sm_servicehealth

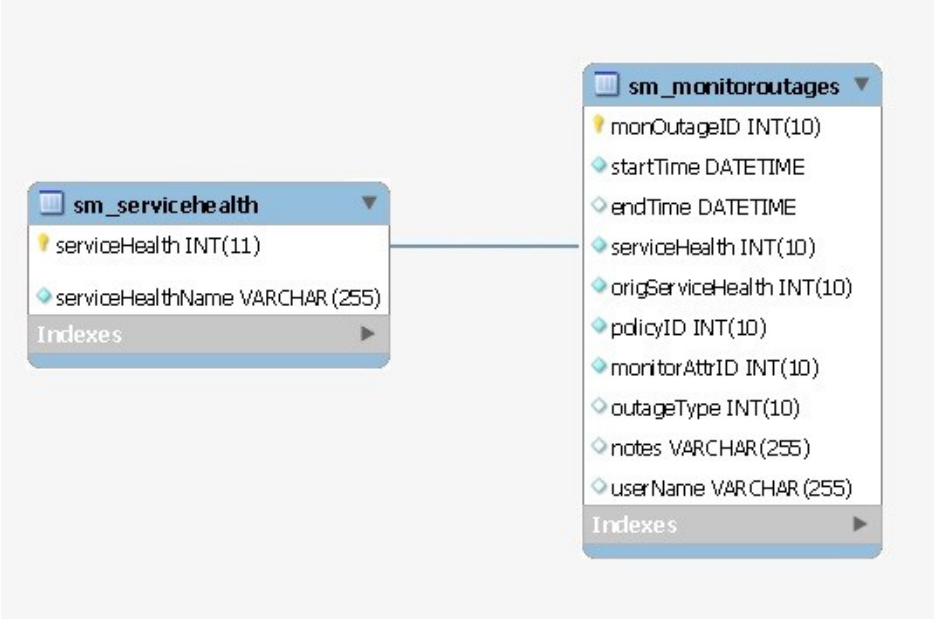
Description

The name of the table used to store attribute type data.

Column

Field	Type	Null	Key	Default	Extra	Comment
serviceHealthName	varchar(255)	NO				Name of the Service.
serviceHealth	int(11)	NO	PRI			Health of the Service.

Relations



sm_slaperiods

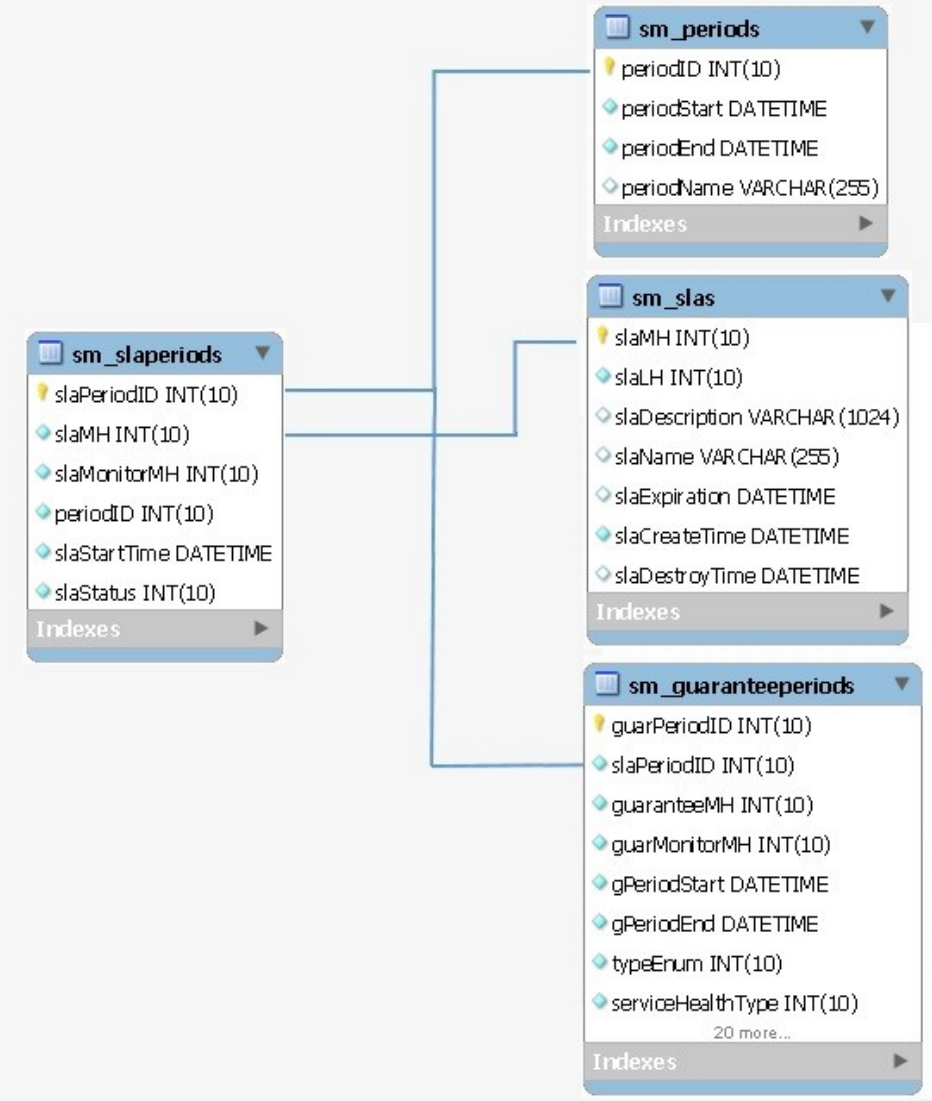
Description

Contains Information of Service Level Agreement.

Columns

Field	Type	Null	Key	Default	Extra	Comment
slaPeriodID	int(10) unsigned	NO	PRI		auto_increment	Unique Period ID for each SLA.
slaMH	int(10) unsigned	NO	MUL			Model_Handle of SLA.
slaMonitorMH	int(10) unsigned	NO				Model_Handle of slaMonitor.
periodID	int(10) unsigned	NO				Period ID of each SLA.
slaStartTime	datetime	NO				Start time of SLA.
slaStatus	int(10) unsigned	NO				Status of SLA

Relations



sm_slas

Description

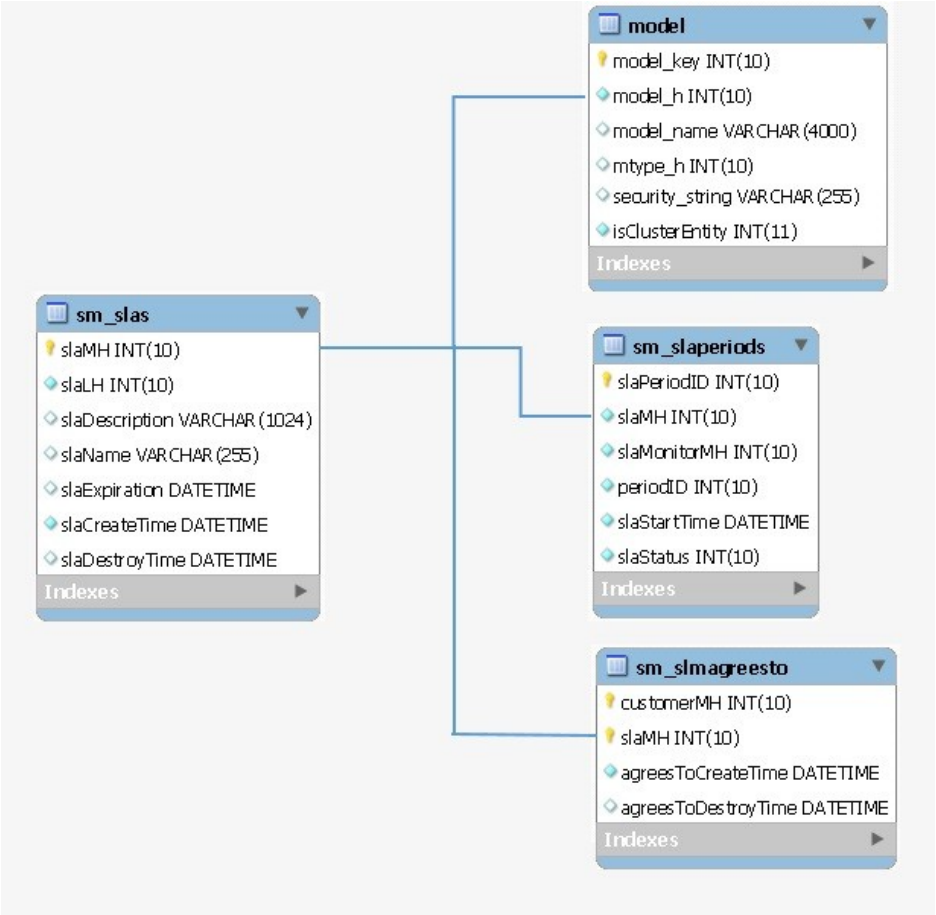
Contains information of all SLA models.

Columns

Column Name	Type	Null	Key	Default	Comment
slaMH	int(10) unsigned	NO	PRI		Model handle of SLA
slaLH	int(10) unsigned	NO			Landscape Handle of SpectroServer where the SLA is present
slaDescription	varchar(1024)	YES			Description of SLA
slaName	varchar(255)	YES			Name of the SLA

slaExpiration	datetime	YES			SLA expiration time
slaCreateTime	datetime	NO			Time when SLA is created
slaDestroyTime	datetime	YES			Time when SLA is destroyed

Relations



sm_slmagreesto

Description

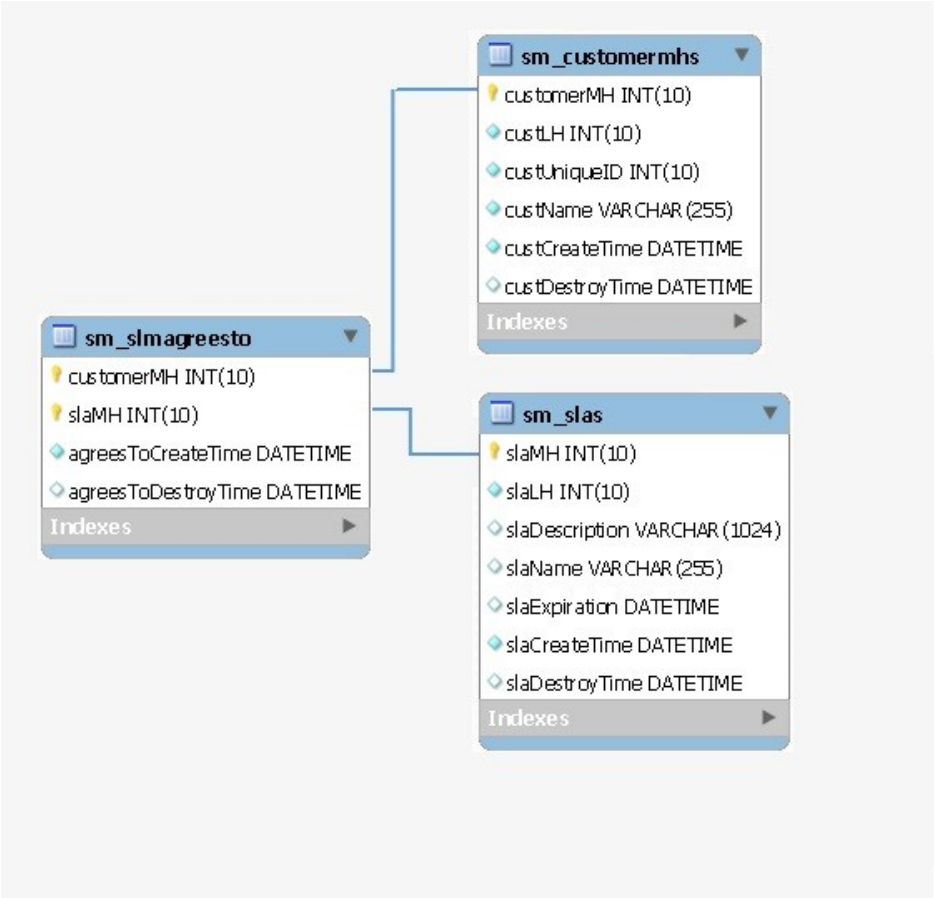
Contains associations between customers and their SLAs.

Columns

Field	Type	Null	Key	Default	Comment
customerMH	int(10) unsigned	NO	PRI		Model handle of customer in decimal
slaMH	int(10) unsigned	NO	PRI		model handle of SLA in decimal

agreesToCreateTime	datetime	NO			The time when customer is associated with SLA
agreesToDestroyTime	datetime	YES			The time when association between customer and SLA is removed

Relations



sm_slmlandscapes

Description

Contains last event retrieved time for each landscape.

Columns

Field	Type	Null	Key	Default	Comment
landscapeHandle	int(10) unsigned	NO	PRI		Landscape handle of SpectroServer
domainName	varchar(255)	NO			Domain Name of SpectroServer

servSyncTime	datetime	YES			Time the last event retrieved from the landscape

Relations

sm_slmmonitors

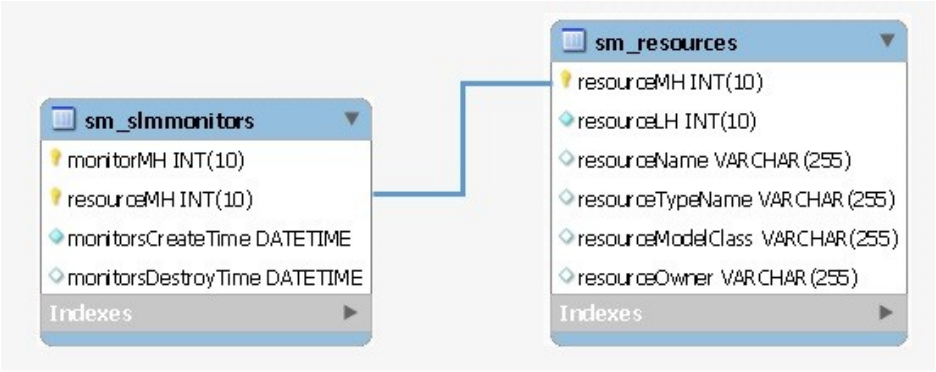
Description

Contains the association between services, resource monitors, and resources.

Columns

Field	Type	Null	Key	Default	Comment
monitorMH	int(10) unsigned	NO	PRI		contains Model handles of services and resource monitors
resourceMH	int(10) unsigned	NO	PRI		Contains model handles of resource monitors and resources (devices)
monitorsCreateTime	datetime	NO			Creation time of monitors (resource monitor or resources/ devices)
monitorsDestroyTime	datetime	YES			Destory time of monitors (resource monitor or resources/ devices)

Relations



sm_slmowns

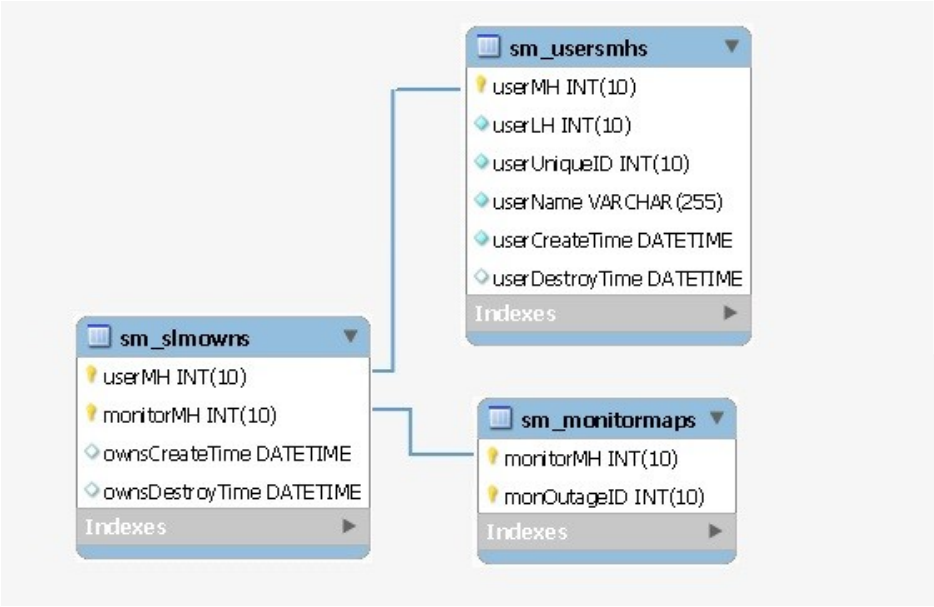
Description

Contains information about the services used by user (owner).

Columns

Field	Type	Null	Key	Default	Comment
userMH	int(10) unsigned	NO	PRI		User model handle
monitorMH	int(10) unsigned	NO	PRI		Model handle of the service that user (owner) uses
ownsCreateTime	datetime	YES			user (owner) creation time
ownsDestroyTime	datetime	YES			user (owner) destroy time

Relations



sm_slmuses

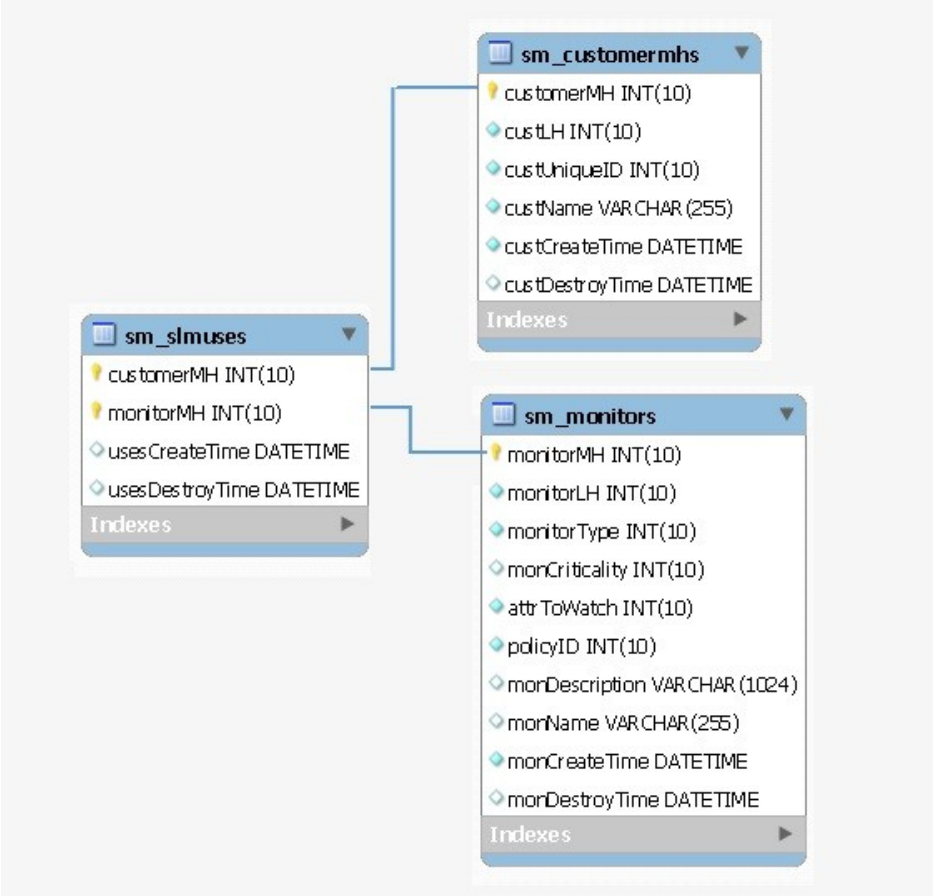
Description

Contains the information about the services used by customer.

Columns

Field	Type	Null	Key	Default	Comment
customerMH	int(10) unsigned	NO	PRI		Customer Model Handle
monitorMH	int(10) unsigned	NO	PRI		Model handle of the service that the customer use
usesCreateTime	datetime	YES			customer creation time
usesDestroyTime	datetime	YES			customer destroy time

Relations



sm_users

Description

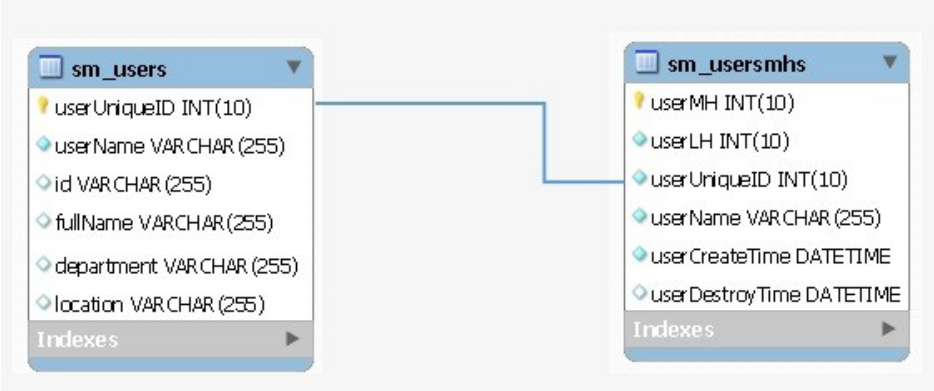
Contains contact information of users.

Columns

Field	Type	Null	Key	Default	Extra	Comment
userUniqueID	int(10) unsigned	NO	PRI		auto_increment	Auto incremented unique user IDs
userName	varchar(255)	NO	UNI			Name of the User
id	varchar(255)	YES				User ID
fullName	varchar(255)	YES				full name of the user
phone	varchar(255)	YES				Phone number of the user
email	varchar(255)	YES				email id of the user
street	varchar(255)	YES				street name of the user

city	varchar(255)	YES				city of the user
state	varchar(255)	YES				state of the user
country	varchar(255)	YES				country of the user
organization	varchar(255)	YES				organization of the user
site	varchar(255)	YES				site of the user loacated
department	varchar(255)	YES				Department of the user
location	varchar(255)	YES				Location of the user

Relations



sm_usersmhs

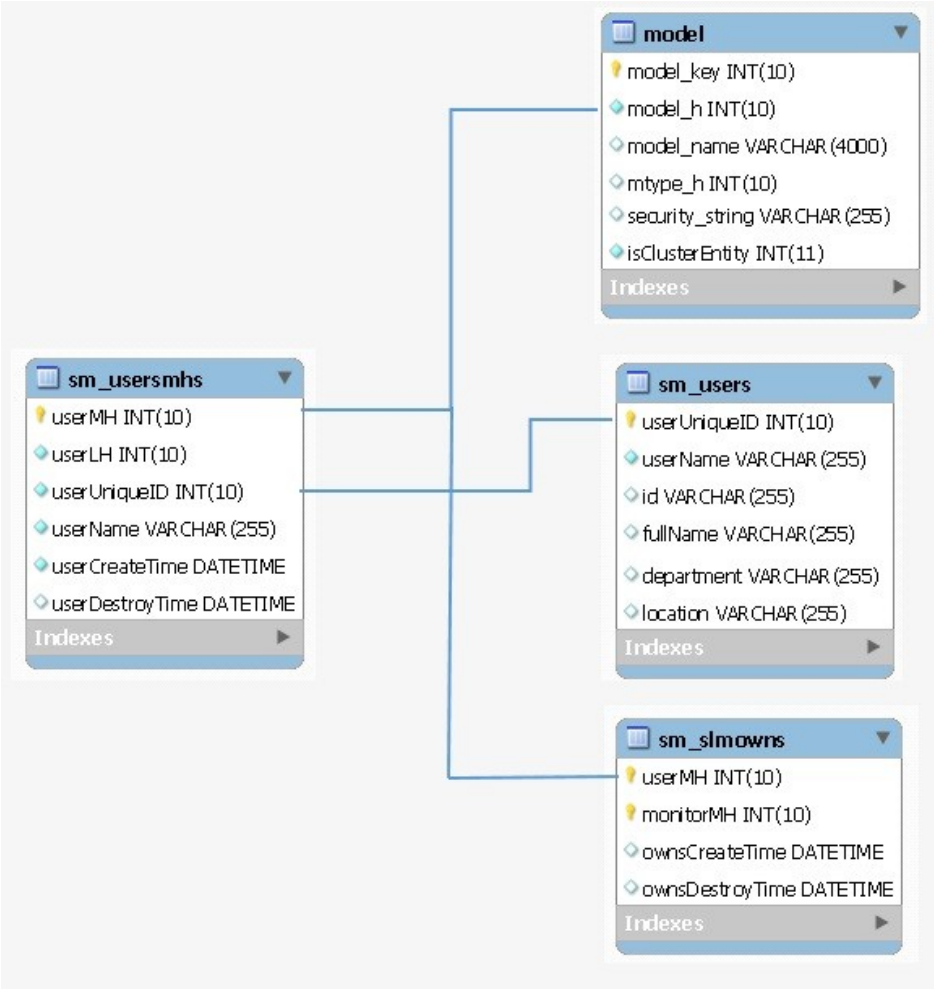
Description

Contain information about service owners or service users.

Columns

Field	Type	Null	Key	Default	Comment
userMH	int(10) unsigned	NO	PRI		service owner model handle
userLH	int(10) unsigned	NO			Landscape handles of the SS where the user is present
userUniqueID	int(10) unsigned	NO	MUL		Auto incremented unique user ids
userName	varchar(255)	NO			Name of service user / owner
userCreateTime	datetime	NO			The time when user created
userDestroyTime	datetime	YES			The time when user Destroyed

Relations



spmbasictestresults

Description

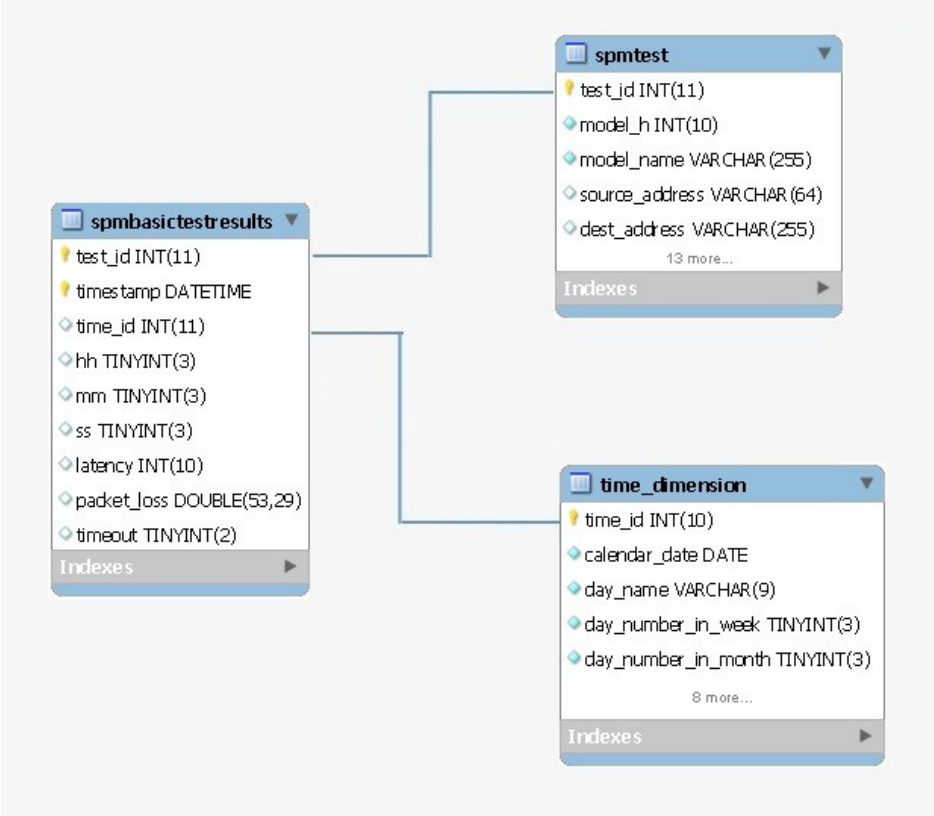
This table stores Service Performance Manager(SPM) basic test results data

Columns

Field	Type	Null	Key	Default	Comment
test_id	int(11) unsigned	NO	PRI	0	Test ID that uniquely identifies a SPM test
timestamp	datetime	NO	PRI	0000-00-00 00:00:00	Timestamp corresponds with time at which result occurred.
time_id	int(11) unsigned	YES	MUL		Time ID from time_dimension table
hh	tinyint(3) unsigned	YES			Hours field of "timestamp" field.

mm	tinyint(3) unsigned	YES			Minutes field of "timestamp" field.
ss	tinyint(3) unsigned	YES			Seconds field of "timestamp" field.
latency	int(10) unsigned	YES			latency for the test
packet_loss	double(53,29)	YES			pocket loss during test
timeout	tinyint(2)	YES			1=timeout occurred, 0=no timeout occurred

Relations



spmhttpfulltestresults

Description

This table stores Service Performance Manager(SPM) http full test results data

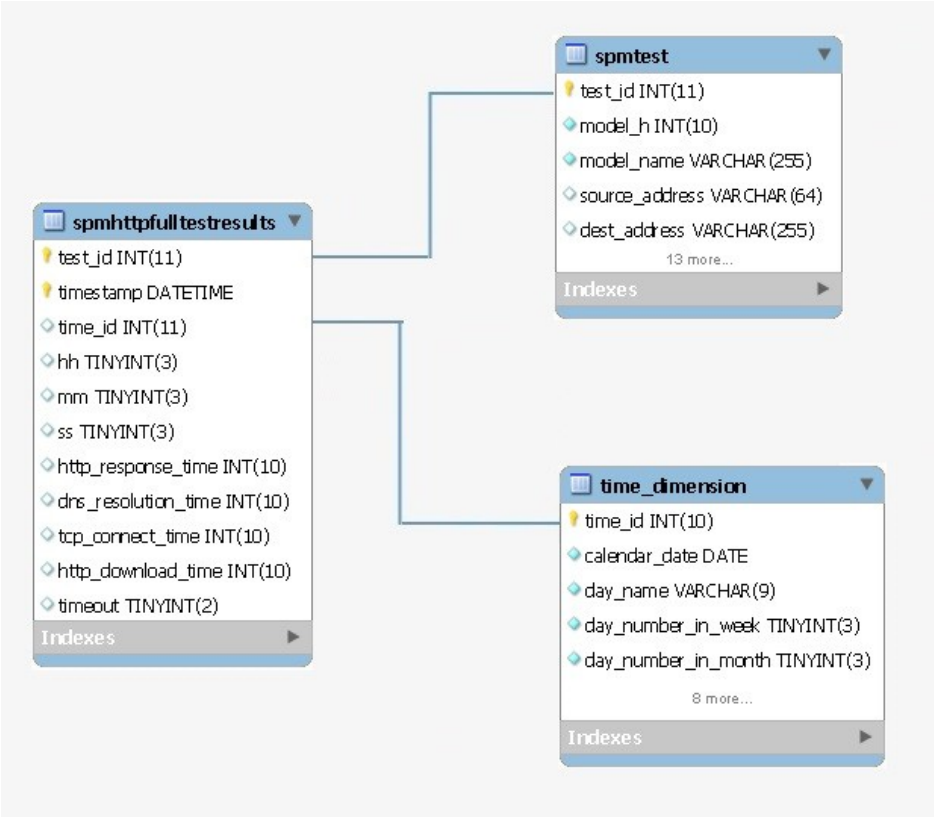
(HTTP tests measure the round-trip time to get a web page.)

Columns

Field	Type	Null	Key	Default	Comment
test_id	int(11) unsigned	NO	PRI	0	Test ID that uniquely identifies a SPM test

timestamp	datetime	NO	PRI	0000-00-00 00:00:00	Timestamp corresponds with time at which result occurred.
time_id	int(11) unsigned	YES	MUL		Time ID from time_dimension table
hh	tinyint(3) unsigned	YES			Hours field of "timestamp" field.
mm	tinyint(3) unsigned	YES			Minutes field of "timestamp" field.
ss	tinyint(3) unsigned	YES			Seconds field of "timestamp" field.
http_response_time	int(10) unsigned	YES			http test response time
dns_resolution_time	int(10) unsigned	YES			time took for dns resolution for this test
tcp_connect_time	int(10) unsigned	YES			tcp connect time
http_download_time	int(10) unsigned	YES			http download time
timeout	tinyint(2)	YES			1=timeout occurred, 0=no timeout occurred

Relations



spmjittertestresults

Description

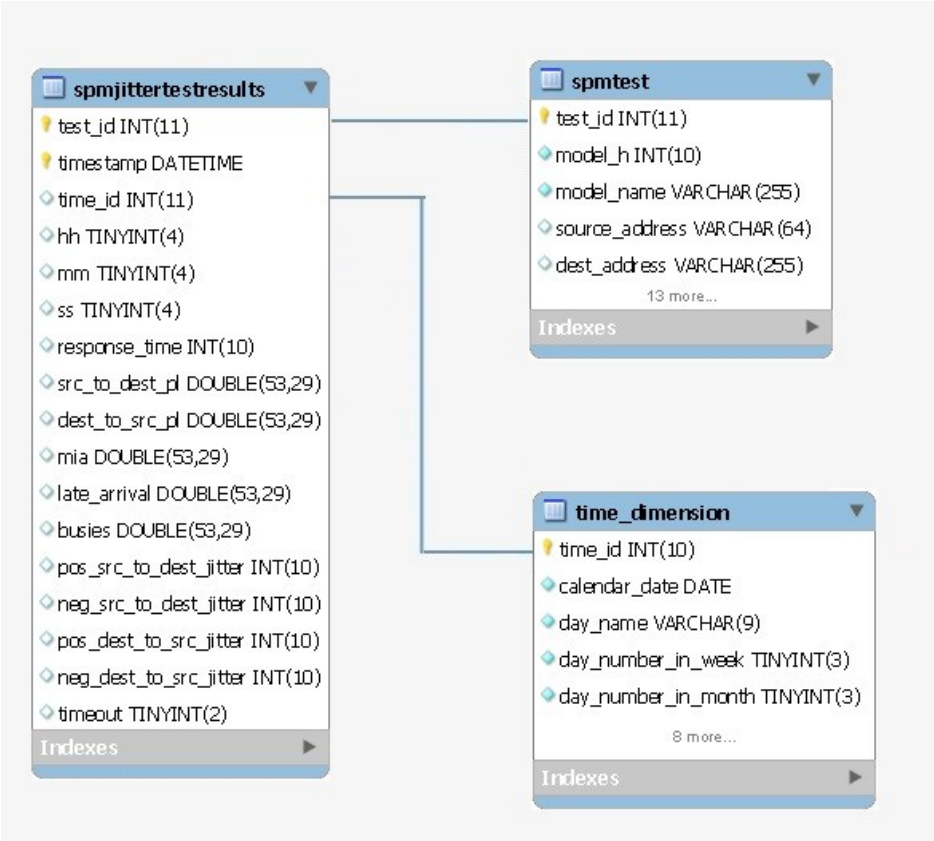
This table stores Service Performance Manager(SPM) jitter test results data

(Jitter tests measure both latency and loss between a test host and a voice-enabled endpoint.)

Columns

Field	Type	Null	Key	Default	Comment
test_id	int(11) unsigned	NO	PRI	0	Test ID that uniquely identifies a SPM test
timestamp	datetime	NO	PRI	0000-00-00 00:00:00	Timestamp corresponds with time at which result occurred.
time_id	int(11) unsigned	YES	MUL		Time ID from time_dimension table
hh	tinyint(4) unsigned	YES			Hours field of "timestamp" field.
mm	tinyint(4) unsigned	YES			Minutes field of "timestamp" field.
ss	tinyint(4) unsigned	YES			Seconds field of "timestamp" field.
response_time	int(10) unsigned	YES			response time for this jitter test
src_to_dest_pl	double(53,29)	YES			Source to Destination Packet Loss
dest_to_src_pl	double(53,29)	YES			Destination to Source Packet Loss
mia	double(53,29)	YES			Missing in Action – Packet Loss with Unknown Direction
late_arrival	double(53,29)	YES			Late Arrival
busies	double(53,29)	YES			Busies
pos_src_to_dest_jitter	int(10) unsigned	YES			Positive source to destination jitter
neg_src_to_dest_jitter	int(10) unsigned	YES			Negative source to destination jitter
pos_dest_to_src_jitter	int(10) unsigned	YES			Positive destination to source jitter
neg_dest_to_src_jitter	int(10) unsigned	YES			Negative destination to source jitter
timeout	tinyint(2)	YES			1=timeout occurred, 0=no timeout occurred

Relations



spmtest

Description

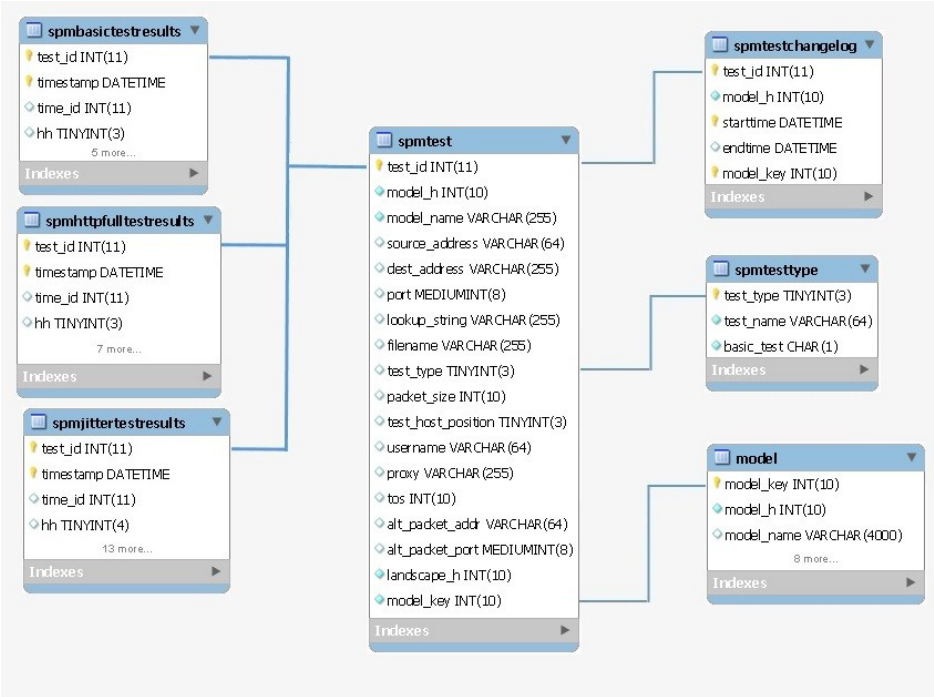
This table stores Service Performance Manager(SPM) test data

Columns

Field	Type	Null	Key	Default	Extra	Comment
test_id	int(11) unsigned	NO	PRI		auto_increment	Test ID that uniquely identifies a SPM test
model_h	int(10) unsigned	NO	MUL	0		Model Handle of the SPM Test Model (decimal form)
model_name	varchar(255)	NO				Model Name of the SPM Test model
source_address	varchar(64)	YES				Source address
dest_address	varchar(255)	YES				Destination address
port	mediumint(8) unsigned	YES				port
lookup_string	varchar(255)	YES				Lookup String

filename	varchar(255)	YES				Filename
test_type	tinyint(3) unsigned	YES	MUL			Type of test
packet_size	int(10)	YES				Packet Size
test_host_position	tinyint(3) unsigned	YES				Test Host Position
username	varchar(64)	YES				user name
proxy	varchar(255)	YES				proxy
tos	int(10) unsigned	YES				Type of service
alt_packet_addr	varchar(64)	YES				Alternate Packet Address
alt_packet_port	mediumint(8) unsigned	YES				Alternate Packet Port
landscape_h	int(10) unsigned	NO		0		Landscape Handle (hexadecimal form)
model_key	int(10) unsigned	NO	MUL	0		Internal Key that uniquely identifies the SPM Test Model

Relations



spmtestchangelog

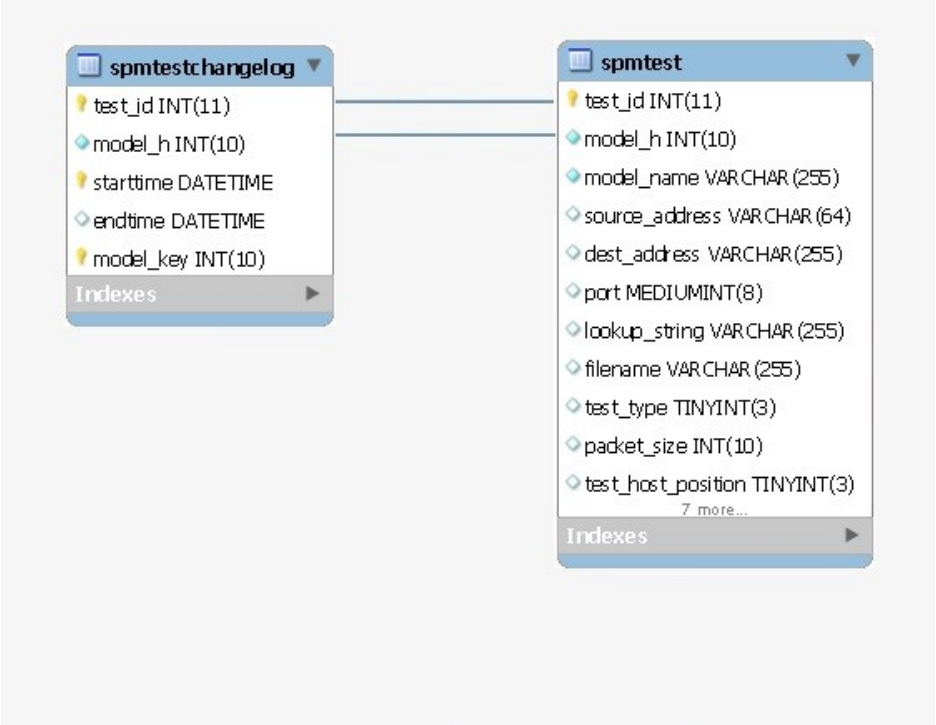
Description

This table stores Service Performance Manager(SPM) test change log data

Columns

Field	Type	Null	Key	Default	Comment
test_id	int(11) unsigned	NO	PRI	0	Test ID that uniquely identifies a SPM test
model_h	int(10) unsigned	NO	MUL	0	Model Handle of the SPM Test Model (decimal form)
starttime	datetime	NO	PRI	0000-00-00 00:00:00	Test start time
endtime	datetime	YES			Test end time
model_key	int(10) unsigned	NO	PRI	0	Internal Key that uniquely identifies the SPM Test Model

Relations



spmtesttype

Description

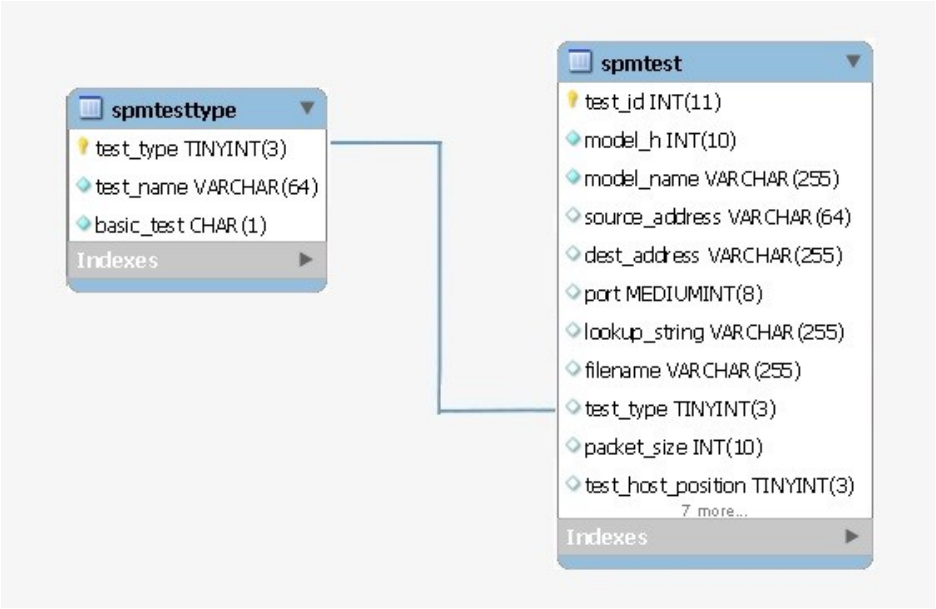
This table stores Service Performance Manager(SPM) test type data like test name and test type etc.

Columns

Field	Type	Null	Key	Default	Comment
test_type	tinyint(3) unsigned	NO	PRI	0	Type of SPM test
test_name	varchar(64)	NO			Name of the SPM test

basic_test	char(1)	NO		Y	Whether this test is basic test or not (Y/N)

Relations



time_dimension

Description

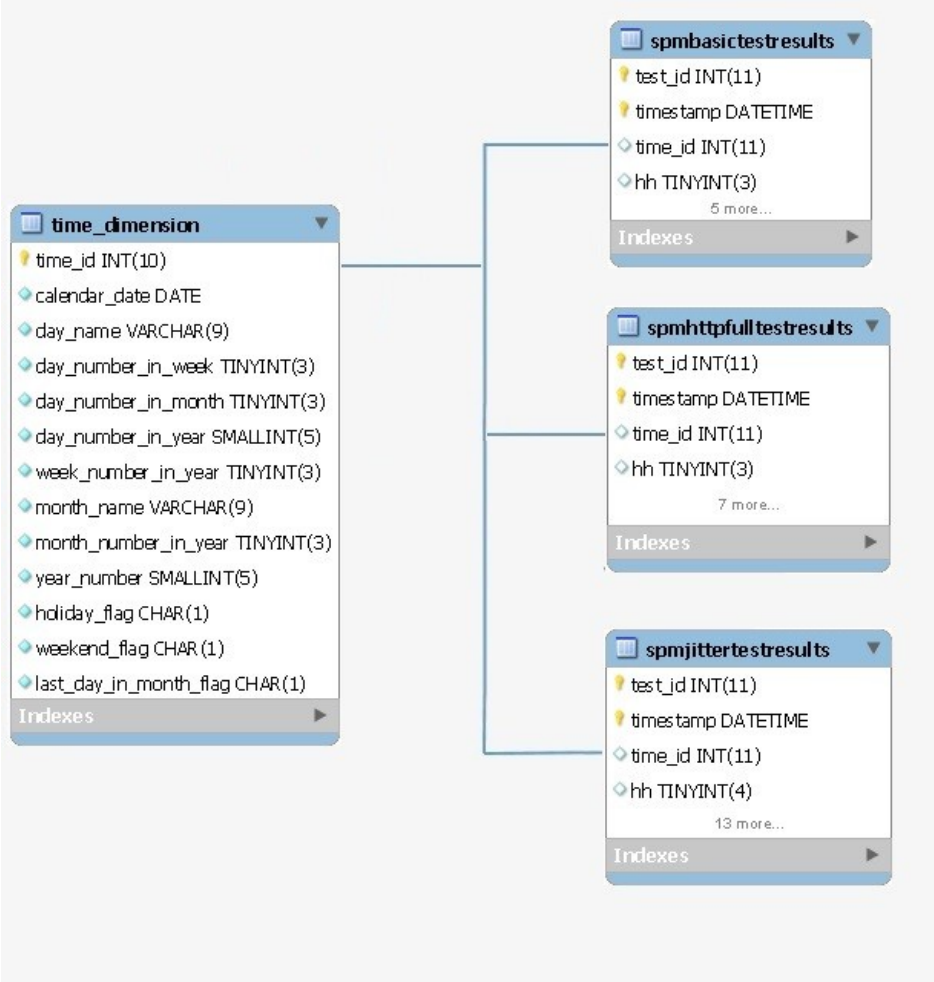
This table stores time dimension data for SPM test result tables

Columns

Field	Type	Null	Key	Default	Extra	Comment
time_id	int(10) unsigned	NO	PRI		auto_increment	Internal ID that uniquely identifies a time in the calendar
calendar_date	date	NO	UNI			Calendar date
day_name	varchar(9)	NO				Day Name (for example, Wednesday)
day_number_in_week	tinyint(3) unsigned	NO				Day Number in Week (Sunday=1, Saturday=7)
day_number_in_month	tinyint(3) unsigned	NO				Day Number in Month
day_number_in_year	smallint(5) unsigned	NO				Day Number in Year
week_number_in_year	tinyint(3) unsigned	NO				Week Number in Year
month_name	varchar(9)	NO				Month Name (for example, January)
month_number_in_year	tinyint(3) unsigned	NO				Month Number in Year (January = 1, December = 12)

year_number	smallint(5) unsigned	NO				Year Number
holiday_flag	char(1)	NO		N		Holiday flag (Y if holiday, N otherwise)
weekend_flag	char(1)	NO		N		Weekend Flag (Y, if Saturday or Sunday)
last_day_in_month_flag	char(1)	NO		N		Last Day in Month Flag (Y, if last day of month)

Relations



translated_string

Description

This table stores string translation data with basic language and target language information

Columns

Field	Type	Null	Key	Default	Extra	Comment
translated_string_id	bigint(20) unsigned	NO	PRI		auto_increment	Translated string ID
base_language	char(5)	YES				Base language of the string
base_language_string	char(255)	YES	MUL			Actual string in the base language
target_language	char(5)	YES				Target language of the string
target_language_string	char(255)	YES				Actual string in the target language

Relations

vendor

Description

This table stores device model vendor data

Columns

Field	Type	Null	Key	Default	Comment
vendor	int(10) unsigned	NO	PRI		Device model vendor ID
vendor_name	varchar(32)	NO			Device model vendor name

Relations

wirelessaps

Description

The table wirelessaps list the following details for event(s) raised on Wireless Controller / Access Points.

Field	Type	Null	Key	Default	Extra	Comment
ap_id	bigint(20) unsigned	NO	PRI	NULL	auto_increment	Unique Identifier
event_key	bigint(20) unsigned	NO		NULL		Event key for the Alarm.
timestamp	datetime	NO		NULL		Timestamp corresponds with time at which event occurred.
controller_mh	int(10) unsigned	NO	MUL	NULL		Model Handle of the WLC Controller.
ap_mh	int(10) unsigned	NO	MUL	NULL		Model Handle of the Access Point.
ap_grpName	varchar(255)	NO	MUL	NULL		Access Point Group Name.
clients_connected	int(10) unsigned	YES		0		Number of Clients connected to Access Point.

data_sent	bigint(20) unsigned	YES		0		The number of bytes sent on the Access Point.
data_received	bigint(20) unsigned	YES		0		The number of bytes received on the Access Point.

wkpeventfilemap

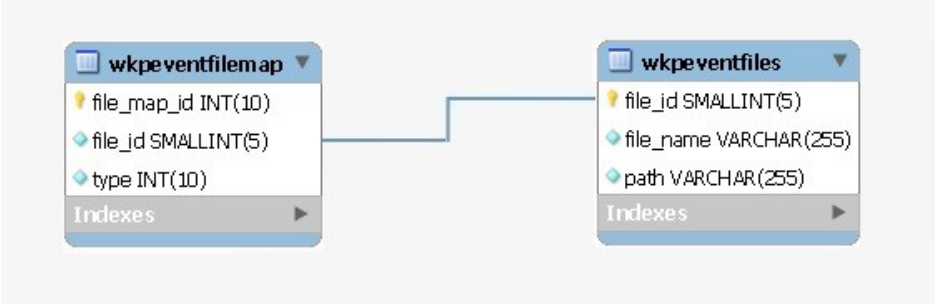
Description

The table wkpeventfilemap maps the event types found in the XML file and maps them to the file from which they were read.

Columns

Field	Type	Null	Key	Default	Extra	Comment
file_map_id	int(10) unsigned	NO	PRI		auto_increment	Event file map ID
file_id	smallint(5) unsigned	NO	MUL			Event file ID
type	int(10) unsigned	NO				Event file type

Relations



wkpeventfiles

Description

This table lists all the event filter XML files that were read from custom directory in OneClick.

Event filters are uniquely named sets of any number of predefined event codes. When users configure an event report, they can elect to specify in the Select the Event Types to include or exclude a field whether to include or exclude data in the report from events in a particular event filter.

An event filter is defined by an XML file that specifies the event codes. Users can create new event filter files, and they can copy and modify the event filter files included with Report Manager. This file resides in custom directory of OneClick.

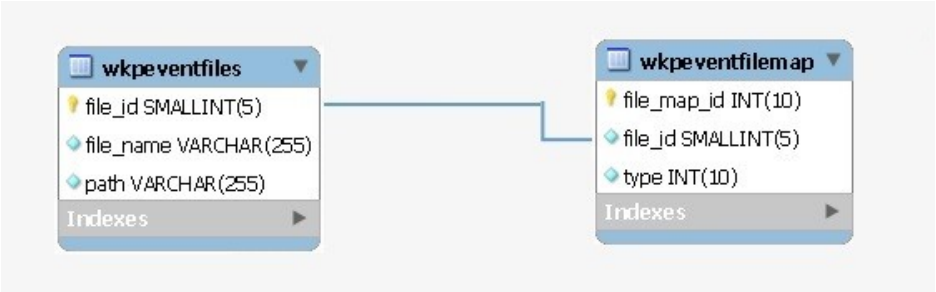
Report manager reads the event filter XML files from custom directory of OneClick. Once read, the data from each XML file is added to two tables: wkpeventfiles and wkpeventfilemap.

Table wkpeventfiles lists all the XML files that were read from the custom directory.

Columns

Field	Type	Null	Key	Default	Extra	Comment
file_id	smallint(5) unsigned	NO	PRI		auto_increment	Event file ID
file_name	varchar(255)	NO				Event file name
path	varchar(255)	NO				Event file path

Relations



z_entity

This is a legacy table, not used in reports

z_entitymodel

This is a legacy table, not used in reports

z_interfaceoutage

This is a legacy table, not used in reports

z_outage

This is a legacy table, not used in reports

Views

v_active_user_model

Description

View for a list of all active oneClick users: All Landscapes

Remarks

View: All Landscapes

v_alarm_activity

Description

List of all alarm activities that are done by oneclick users. Contains information related to assigned user, assigned time, trouble ticket ID etc.

Remarks

View: All Landscapes

v_bi_alarm_activity_by_user

Description

View Definition for report Top N Assets with Most Alarms

Remarks

View: All Landscapes

v_bi_topnalarmtypesmain

Description

View Definition for the report Top N Most Common Alarm Types

Remarks

View: All Landscapes

v_bi_topnassetswithmostalarmsmain

Description

View Definition for the report Alarm Activity by User

Remarks

View: All Landscapes

v_ncm_config_diff

Description

View definition to list NCM configuration changes. Will have data related to configID, change time, number of lines that are changed etc.

Remarks

View: All Landscapes

v_security_string_accessibility_by_landscape

Description

View definition to list the security string access data for users of all landscapes

Remarks

View: All Landscapes

v_user_report_security

Description

View definition as a union of all OC users and BO users

Remarks

View: All Landscapes

The SpectroSERVER and Threads

The SpectroSERVER handles requests from many client applications while simultaneously accessing the disk and the network. For better efficiency, the SpectroSERVER operates using a multithreaded architecture that has less overhead than running separate processes.

The SpectroSERVER creates some threads at startup that terminate only when the SpectroSERVER terminates. The SpectroSERVER creates other threads dynamically and terminates them when they are no longer needed. For example, each time a client connects to the SpectroSERVER or makes a request through an API, a new thread is started.

Typically, it is not necessary to be concerned about this internal threading mechanism. However, this concept can become important on a heavily loaded system when advanced tuning is required to maximize your system throughput.

Managed Elements

The SpectroSERVER uses models to represent managed elements, and these models are based on the model types that are defined in the modeling catalog. Some model types can be instantiated to represent a device, an application, or a host that operates in the computing infrastructure. The SpectroSERVER can communicate directly with these managed elements using SNMP, if appropriate. Some model types are instantiated into models that act as containers and are used to group other models. For example, you can create a LAN model to group certain managed elements on a network segment. Or you can create a Room model to group the managed elements in one room.

A container model can contain either other container models or models that represent managed elements, or both, depending on the container model type. For example, an IPClassB container can contain a model representing a router, and it can also contain several LAN models representing a range of subnets. However, a Building model can only contain container models: a Floor, a Section, or a Room.

The SpectroSERVER uses *management modules* to manage the specific elements of a computing infrastructure. A management module is made up of model types, relations, inference handlers, and support files. A management module uses a series of models to represent each component of the specific type of managed element. The managed element can be represented with a device model. The device functionality is supported with a combination of other types of models, such as application models. Each major functional component of a managed element can be modeled as a separate application or can be incorporated into the device model. An application often corresponds to the functionality of a MIB, or a section of that MIB.

All the models that are used to represent a managed element are based on model types that are defined in the modeling catalog. The associations that the models have with one another are based on the relations and meta-rules that are defined in the modeling catalog. The SpectroSERVER can implement various associations, such as the following examples:

- A container model can contain models representing managed elements or container models
- Connections between device or port models can be established representing physical or logical connectivity
- Application models that support a device model express relations showing what functionality a device provides

> **NOTE**
> Not all model types that are defined in the DX NetOps Spectrum knowledge base can be used to create a model in OneClick. Some function only as base model types from which other model types are derived.

Device Discovery

DX NetOps Spectrum supports an automatic method of discovering and modeling managed elements and connections within the computing infrastructure. *Discovery* is a OneClick feature that automates the process of discovering and modeling the entities in your IT infrastructure. You can create and edit Discovery and modeling configurations to

customize and simplify the process. DIscovery also lets you filter and export the results of Discovery or modeling sessions.

Discovery comprises two components. The Discovery application scans IP address ranges or lists and reads a select group of key MIB objects from each SNMP-enabled managed element encountered. The result is presented to you or is sent to the SpectroSERVER for modeling. The server-side, or back-end, Discovery process uses the key MIB objects of each managed element to determine the best model type to use. Then Discovery creates an instance of that model type to represent that managed element.

After the models are created and activated for the managed elements that Discovery finds, the back-end Discovery process determines their placement and connectivity. Model placement and connectivity are based partially on user-specified options. Based on their IP address, models of bridges and workstations are placed inside a LAN container. These models are then connected to other bridges using spanning tree and source address tables, which are read from the MIBs of the managed elements. Models of routers are placed in network containers or in the Universe. These models are connected to LAN models or to other routers using IP addresses and masks, IP route table information, or proprietary discovery protocol MIBs.

DX NetOps Spectrum also lets you create specific models from the OneClick Console. Two methods can be used. The first method uses the IP address or the DNS name of the managed element. With this information, the SpectroSERVER contacts the managed element and retrieves the name, vendor, description, location, and sysOID of the managed element. DX NetOps Spectrum then creates a model using the model type that best represents the functionality of this device.

You can also create a model by first selecting a model type as a base. In this case, you provide an IP address or a DNS name so the SpectroSERVER can communicate with the managed element. However, the model type that you select is instantiated, regardless of the SpectroSERVER assessment of managed element functionality. The SpectroSERVER creates all of the appropriate supporting model types and matches the functionality that the MIBs for the managed element describe.

Device Communication Manager

The Device Communication Manager (DCM) is the interface between the SpectroSERVER and the managed elements. The DCM includes various protocol interfaces that communicate with managed elements using a specific protocol. One interface exists for each of the two supported protocols, SNMP and ICMP. When the SpectroSERVER communicates with a managed element, the request is sent to the appropriate protocol interface in the DCM. The DCM, in turn, passes the request to the managed element.

Polling

The SpectroSERVER constantly updates its knowledge of network conditions using polling and logging services. The DCM handles communication with the managed element being polled. Model type attributes are defined as either external (to be obtained from the managed element) or internal (stored either in memory or the database). Some external attributes are defined as *polled*, which means that the SpectroSERVER regularly polls the managed elements. The polling frequency depends polling_interval attribute value that is defined for the model. Values of external attributes that do not have polling set are obtained from the managed element upon each client application or inference handler request.

> **NOTE**
> Polling affects the performance of the SpectroSERVER and the network. Shorter polling intervals can limit the responsiveness of the SpectroSERVER and can generate an unacceptable amount of network traffic.

Logging

Attributes can also be defined as being logged, meaning that their values are written through the Archive Manager to the DDM database. The frequency with which values are logged is based on both the polling_interval and the Poll_Log_Ratio defined for the model.

For example, you can set the Poll_Log_Ratio to 10 and polling_interval to 60. This setting logs the attribute value to the statistics file every tenth poll, or every 600 seconds.

NOTE
Logging affects the performance of the SpectroSERVER and the network. Smaller logging ratios can limit the responsiveness of the SpectroSERVER and can generate an unacceptable amount of network traffic.

Alerts, Events, and Alarms

DX NetOps Spectrum is a services and infrastructure management system that notifies you when a fault occurs in a managed element in your network. One way that DX NetOps Spectrum accomplishes this functionality is by receiving alerts (usually SNMP traps) from problem areas in the computing infrastructure. DX NetOps Spectrum then converts those alerts into events and alarms to be displayed in DX NetOps Spectrum applications.

DX NetOps Spectrum uses a series of support files that are named event configuration files to indicate how alerts, events, and alarms are processed. DX NetOps Spectrum receives alerts from problem areas within the managed infrastructure. DX NetOps Spectrum converts alerts into events and alarms, which are displayed in OneClick event and alarm views. Alerts, events, and alarms let DX NetOps Spectrum notify you about significant occurrences in your IT infrastructure.

Alerts

An alert is an unsolicited message that a managed element sends to DX NetOps Spectrum. The primary management protocol that DX NetOps Spectrum uses to communicate with managed elements is SNMP. An SNMP-compliant managed element can send an alert, which is known as a trap. Managed elements with SNMP traps enabled can be configured to direct their traps to the SpectroSERVER. The SpectroSERVER uses the source IP address of the trap to identify the model that is associated with that managed element. Once the model is known, the trap is processed as directed by the AlertMap file that is associated with that model type. An AlertMap file exists for most device model types within DX NetOps Spectrum. The AlertMap is an ASCII file that is used to map SNMP traps into DX NetOps Spectrum events.

Events

An event is an object representing an instantaneous occurrence within DX NetOps Spectrum. Events usually indicate that something significant has occurred regarding the model or another component. Most device model types have an associated EventDisp event configuration file. An EventDisp file is an ASCII file that indicates how an event is processed. After an AlertMap file converts an SNMP trap into an event, the EventDisp file tells DX NetOps Spectrum how to handle this event for this model. The processing of an event can include logging the event and generating an alarm.

Alarms

An alarm is an object which indicates that a user-actionable, abnormal condition exists in the managed environment. Usually an alarm is generated as a result of a received trap -- that is, when an event has occurred and the EventDisp file specifies alarm generation. A configured watch can also generate an alarm, as can DX NetOps Spectrum detecting an abnormal situation that is not based on an event. After the abnormal condition that caused the alarm ends, another event clears the corresponding alarm or you can clear it. Alarm notifications can be sent to applications and inference handlers that need this information. DX NetOps Spectrum can examine myriad network events, analyze them, and produce few important alarms.

Landscapes and the Distributed SpectroSERVER

A landscape is the DX NetOps Spectrum term for a network domain that a single SpectroSERVER manages. A landscape is composed of the models, associations, attribute values, alarms, events, and statistics belonging to a specific SpectroSERVER. Each landscape that is contained in a network is unique, and a unique landscape handle (ID) identifies each landscape. A landscape icon can represent each landscape in the OneClick Console. The landscape icon provides a graphical representation of a SpectroSERVER knowledge base.

The Distributed SpectroSERVER (DSS) is a powerful modeling feature that enables the distribution of management for portions of a large-scale network. The work can be distributed either geographically or across multiple servers in a single physical location. DSS can improve DX NetOps Spectrum performance when managing a computing infrastructure by distributing the network load that management traffic introduces. DSS can also delegate management functions to remote workstations.

Using a DSS, you can create a unified representation of the computing infrastructure, which is composed of multiple landscapes, each with its own local SpectroSERVER. In a DSS environment, a SpectroSERVER client, such as the OneClick Console, can access information from more than one SpectroSERVER simultaneously.

> **NOTE**
> The Distributed SpectroSERVER Administration section contains tips on how to segment your network into landscapes.

10.2 enables you to opt for increased capacity and scale by allowing you to monitor huge number of devices with fewer landscapes. Using the Huge Landscape Handle type (during installation), you can create landscapes which can support huge model count (beyond 1 Million models per SpectroSERVER). Landscape handles are multiples of 64. This option is only available for fresh installs and best suited for new DSS environments and new single-server installations. This capability is not supported for upgrade and migration scenarios. Currently you can scale upto 2 million models per SpectroSERVER.

You need to use the same Landscape Handle type throughout DSS environment, i.e. if you select Huge Landscape Handle type during installation, you need to assign the landscape handles in multiples of 64, and if you select the Legacy Landscape Handle type, you need to assign landscape handles in multiples of 4.

> **WARNING**
>
> For upgrade and migration scenarios, DX NetOps Spectrum automatically uses Legacy Landscape Handle type. All pre-10.2 landscape handles belong to Legacy Landscape Handle type.

When you model multiple landscapes using DSS, the database for each landscape must contain identical modeling catalogs. All model types that exist in the modeling catalog for one landscape must also exist in the modeling catalog of every other landscape. Hence, if you install add-on applications in one landscape, the same applications must be installed on every landscape. For example, if you install VPN Manager in one landscape, install VPN Manager in all landscapes.

Administration of all the modeling catalogs in a distributed environment is made easier with the concept of a master catalog. The master catalog is the SpectroSERVER that you designate to update the other SpectroSERVERs in the landscape map. When a change is necessary, it is made to the master catalog. The entire master catalog is manually copied to all other SpectroSERVERs in the landscape map, propagating all changes and keeping the modeling catalogs consistent.

> **NOTE**
> See the Distributed SpectroSERVER Administration section for more information.

> **NOTE**
>
> From 10.3 onwards, you can load the catalog of the Legacy Landscape on a Huge Landscape and vice versa (using SSdbload -c option). However you will not see the landscape details displayed. For more information on the landscape details, see the Known Anomalies section.

Client Applications Overview

The main DX NetOps Spectrum client application is OneClick. A few other DX NetOps Spectrum client applications also let you interact with the information that is stored and processed on the SpectroSERVER.

You can use the following client applications when customizing DX NetOps Spectrum or integrating with DX NetOps Spectrum:

- **AlarmNotifier:** This application is used to forward alarm data to user-defined scripts or third-party applications. For more information, see About AlarmNotifier.
- **SANM:** This application is used with the AlarmNotifier to specify policies that filter alarm data sent to user-defined scripts or third-party applications. For more information, see Spectrum Alarm Notification Manager (SANM).
- **Report Manager:** This application is a full-featured system that creates reports from data that is extracted from the knowledge base on the SpectroSERVER. Report Manager relies on two more client applications: CA Business Intelligence and InfoView.

For more information, see the AlarmNotifier, Spectrum Alarm Notification Manager (SANM), Report Manager sections.

The following applications are not DX NetOps Spectrum clients, but they are required for some optional DX NetOps Spectrum customization and integration.

- **Process daemon:** The process daemon is a process launching and tracking daemon that lets DX NetOps Spectrum control various processes running on a workstation. The process daemon starts processes when requested by an application, such as the Control Panel. The process daemon can also start processes on system boot when configured to do so. The process daemon automatically restarts critical processes when they stop unexpectedly. The DX NetOps Spectrum Control Panel is the only executable actively launched by you, the DX NetOps Spectrum user. The process daemon launches all other applications following a request by the user or another application. The process daemon operates in the background and is transparent to you. The process daemon automatically starts during DX NetOps Spectrum installation and whenever the system is started.
- **Model Type Editor:** This application is used to derive new model types that support the development of new management modules.

OneClick Console

The OneClick Console displays information from the SpectroSERVER using icons and views. Icons are illustrations of the models that are defined to represent the managed elements of your computing infrastructure. Views are the various ways in which data from the SpectroSERVER is organized for display.

OneClick Console Icons

Icons are graphic representations of instantiated models that are based on model types from the DX NetOps Spectrum modeling catalog. The various icons can represent individual managed elements, groups of managed elements, geographic locations, users, landscapes, connections between models, and so on. Pipes are a special type of icon and are used to represent the connectivity between managed elements.

General information about a model, such as the model name and model type name, is visible on the icon. Detailed information about a model is found within various icon subviews that are accessed by double-clicking the icon. Some icons use color to indicate the condition of the managed elements that they represent.

Hierarchical Views

A view in DX NetOps Spectrum is a way to organize data so it can be displayed or manipulated. Hierarchical views represent ways to structure your network data. When structuring your network data in the XML file, you select from elements that represent each of the hierarchical views. Two types of hierarchical views exist: Topology and Location.

Topology View

The Topology view is really an abstraction of networking components. When working with this view, you represent the physical or logical components of your network and group these components, while factoring in their logical connectivity. You can also represent connections graphically, using pipes that show how devices are connected at the port or device level. In the <oc> Console, this view appears as the Universe topology.

NOTE
For more information about the Universe topology view, see Modeling and Managing Your IT Infrastructure.

Location View

The Location view organizes your network data by physical location. Using this view that you can depict your network in geographical terms. You can start with your global offices and can go right to the wiring closet on each floor of each building in each region where your offices are located. In the <oc> Console, this view appears as the World topology.

NOTE
For more information about the World topology view, see Modeling and Managing Your IT Infrastructure.

Reporting with CA Business Intelligence (CABI)

DX NetOps Spectrum Reporting uses CA Business Intelligence (CABI) to display reports.

CABI is a reporting and analytic software package that DX NetOps Spectrum and other CA products use to present information. DX NetOps Spectrum uses CABI to integrate, analyze, and present information that is required for effective enterprise IT management, through reports.

CABI includes SAP BusinessObjects Enterprise XI, which is a complete suite of information management, reporting, query, and analysis tools.

CABI installs SAP BusinessObjects Enterprise XI as a standalone component. The installation runs independently of DX NetOps Spectrum and other CA products, allowing various CA products to share Business Intelligence services. CABI installation is a distinct and separate activity within the overall CA product installation process.

NOTE
For more information, see the CA Business Intelligence Implementation section and the CA Business Intelligence Release Notes .

InfoView Report Management

BusinessObjects Enterprise InfoView (InfoView) is a web-based interface that lets you manage reports with the following features:

- Browsing and searching capabilities.
- Content access (creating, editing, and viewing).
- Content scheduling and publishing.

The InfoView functions like a Windows application rather than a simple web application. The InfoView toolbar dynamically changes to provide actions through context menus consistent with the function you want to perform. InfoView provides context menus, by simply clicking the right mouse button. You can double-click items in a window to execute default actions. Report structures are consistent and provide powerful security and authorizations.

InfoView provides access to the WebIntelligence (WEBI) designer. This designer lets you create customized reports with a simple drag-and-drop interface. Custom data object selection with effective filtering options, enables powerful reporting capabilities for your environment.

You can access InfoView from the OneClick home page or directly from a Web browser. The typical URL format is as follows:

```
http://<hostname>/InfoViewApp
```

NOTE
For more information, see the Report Manager section.

Attribute and Relation Definitions

This section explains what Attributes and Relations are in DX NetOps Spectrum. Some examples of Attribute and relation are also listed in the following sections:

- Attributes
- Attribute Descriptions
- Relation Descriptions

Attributes

The concept of an attribute and its role regarding models and model types is outlined in Model Type Attributes.

Many attributes in the knowledge base are counterparts of **Management Information Base** (**MIB**) variables. External attributes directly correspond to specific MIB variables. Internal attribute values can be derived from the values of MIB variables. These values have often undergone some mathematical calculations to arrive at their DX NetOps Spectrum value. The naming conventions typically make the relationship between the attribute and the MIB variable evident.

The following tables show the attributes that have been defined to help you integrate other applications with DX NetOps Spectrum. These attributes are used when creating a management module, using the Southbound Gateway Toolkit, or using the Modeling Gateway Toolkit. Each table is a quick reference with attributes grouped by functionality.

NOTE
You can access specific information about attributes and relations from the Model Type Editor. For more information, see the Model Type Editor section.

Application Model Discovery

Attribute	Attribute ID	Found On
default_attr	0230006	Application model types

Device Model Discovery

Attribute	Attribute ID	Found On
DeviceNameList	0x1293E	Device model types
DeviceType	0x23000e	Device model types
Disposable_Precedence	0x114e2	Device model types
Enable_IH_Spec_Dev_Name	0x3d0062	Device model types
Enable_IH_Device_Name	0x3d0008	Device model types
Image_Index	0x3d0001	Device model types
System_OID_Verify	0x110bb	Device model types
System_OID_Verify_List	0x12910	Device model types

General Model Type Information

Attribute	Attribute ID	Found On
CompanyName	0x118b8	Device and application model types
Description	0x230017 and 0x118bc	Device, application, and interface model types
DeviceType	0x23000e	Device model types
Manufacturer	0x10032	Device model types
MMName	0x1196a	Device and application model types
MMPartNumber	0x1196b	Device, application, and interface model types
Model_Class	0x11ee8	Device model types
Model_Name	0x1006e	Device, application, and interface model types
Modeltype_Name	0x10000	Device, application, and interface model types
Vendor_Name	0x11570	Device, application, and interface model types

Network Information

Attribute	Attribute ID	Found On
Network_Address	0x12d7f	Device, application, and interface model types
Network_Mask	0x12dbc	Device, application, and interface model types

Polling Information

Attribute	Attribute ID	Found On
polling_interval	0x10071	Device, application, and interface model types
poll_log_ratio	0x10072	Device, application, and interface model types
pollingstatus	0x1154f	Device, application, and interface model types

Port Identification

Attribute	Attribute ID	Found On
ifAlias	0x11f84	Interface model types
if_Index	0x11348	Interface model types
ifName	0x11f60	Interface model types
if_Phys_Addr	0xd0399	Interface model types
ip_address	0x12dbb	Interface model types

SNMP Information

Attribute	Attribute ID	Found on
Community_Name	0x10024	Device, application, and interface model types
CommunityNameForSNMPSets	0x11a7f	Device, application, and interface model types
isManaged	0x1295d	Device, application, and interface model types
Security_String	0x10009	Device, application, and interface model types

Attribute Descriptions

This section provides more information about the attributes that are outlined in the previous tables.

> **NOTE**
> You can access more information about each attribute by using the Model Type Editor. Details, such as the attribute ID, the data type of the attribute value, or the attribute flags that are set for the attribute, are available. See the Model Type Editor section.

Several attribute flags can be set for each attribute including these main ones:

- **External:** When set to TRUE, this flag indicates that the value for this attribute is maintained outside of the SpectroSERVER. This setting also indicates that an update of the attribute value is done either at a user request or at a polling interval.
- **Readable:** When set to TRUE, this flag informs the SpectroSERVER that a client or other application is allowed to read this attribute value from the SpectroSERVER. If the External flag is set, set this flag according to the MIB definition of the Readable variable for this attribute. If the External flag is not set, set this flag as desired.
- **Writable:** When set to TRUE, this flag informs the SpectroSERVER that a client or other application can write this attribute value to the SpectroSERVER database. If the External flag is set, set this flag according to the MIB definition of the variable for this attribute. If the External flag is not set, set this flag as desired.
- **Shared:** When set to TRUE, this flag declares that one value exists for this attribute and that all models of the current model type share it. The value is not duplicated in memory or in the database for each model.

The following list contains descriptions for the attributes that one or more of the DX NetOps Spectrum integration points uses or references.

- **Community_Name**
  Identifies the SNMP community string that is used when DX NetOps Spectrum attempts to communicate with a managed element using SNMP. This attribute is evaluated when performing SNMP gets. If the attribute CommunityNameForSNMPSets is empty, this attribute is used when performing SNMP sets, too.
- **CommunityNameForSNMPSets**
  Specifies the community name that is used when performing SNMP sets. If left blank, the Community_Name attribute value is used for SNMP sets.
- **CompanyName**

Specifies the company name the device and application model types use. This value is set equal to the name of the company that developed the model type.

- **default_attr**
  Identifies the applications that a particular managed element supports. This attribute is used in the application Discovery process. The value of the default_attr is set equal to the attribute ID of an attribute that represents a MIB object which uniquely identifies the MIB.
- **Description**
  Provides a textual description of the model type.
- **DeviceNameList**
  Contains the device names that correspond to the OIDs in the SysOIDVerifyList attribute. To use this attribute to set the device name, set the Enable_IH_Device_Name attribute and the Enable_IH_Spec_Dev_Name attribute to TRUE.
- **DeviceType**
  Holds the name of the device type when the model type represents only one specific type of device. The value of this attribute is displayed in the field at the bottom of the icon for the model. If multiple device types exist for a given device, use the DeviceNameList attribute.
- **Disposable_Precedence**
  Contains a value that the DX NetOps Spectrum device discovery mechanism uses to resolve conflicts in the device model type selection process. A conflict occurs when multiple model types have a System_OID_Verify value that matches the SystemObjectID of the managed element. To resolve the conflict, Discovery uses the model with the highest Disposable_Precedence value.
- **Enable_IH_Spec_Dev_Name**
  When the Enable_IH_Spec_Dev_Name attribute is TRUE, DX NetOps Spectrum uses the Enable_IH_Device_Name inference handler to determine the vendor name using the enterprise number from the device.
- **Enable_IH_Device_Name**
  When the Enable_IH_Device_Name attribute is TRUE, DX NetOps Spectrum uses the Enable_IH_Spec_Dev_Name inference handler to read the device System Object ID to determine the product name.
- **ifAlias**
  Corresponds to this value from the MIB II Interface table.
- **ifIndex**
  Corresponds to this value from the MIB II Interface table.
- **ifName**
  Corresponds to this value from the MIB II Interface table.
- **if_Phys_Addr**
  Corresponds to this value from the MIB II Interface table.
- **Image_Index**
  Links a GnSNMPDev model with an icon image that represents the model. Possible values and their corresponding images are as follows:
  - **1**
    Generic Device
  - **2**
    Bridge
  - **3**
    Router
  - **4**
    Hub
  - **5**
    PC
  - **6**
    Terminal Server
  - **7**

Workstation
  - **8**
    Switch
- **ip_address**
  Specifies the IP address that is associated with an interface model.
- **isManaged**
  Indicates if a device model is managed. When this attribute is set to TRUE, DX NetOps Spectrum manages this device using SNMP communication.
- **Manufacturer**
  Used with a device model to show the manufacturer responsible for the device.
- **MMName**
  Holds the management module name for device and application models.
- **MMPartNumber**
  Contains the part number that the management module developer has assigned to the management module. Most device, application, and interface models use this attribute.
- **Model_Class**
  The Model_Class defines the type of device that the model represents. The following table lists the model classes that are available in DX NetOps Spectrum and their respective integer identifier. For example, the model class SWITCH uses the identifier 2.
  - **0**
    UNKNOWN
  - **1**
    OTHER
  - **2**
    SWITCH
  - **3**
    ROUTER
  - **4**
    SWITCH_ROUTER
  - **5**
    HUB
  - **7**
    LINK
  - **8**
    NETWORK
  - **9**
    WORKSTATION_SERVER
  - **10**
    CONTAINER
  - **11**
    CHASSIS
  - **12**
    PINGABLE
  - **13**
    MAC
  - **14**
    SNMP
  - **15**
    PORT
  - **16**

USER
- **17**
APPLICATION
- **18**
COMPONENT
- **19**
LANDSCAPE
- **20**
ROUTER_APP
- **21**
SWITCH_APP
- **22**
BRIDGE_APP
- **23**
MIB_APP
- **24**
RMON_APPL
- **25**
UNIX
- **26**
NT
- **27**
FIREWALL
- **28**
IDS
- **29**
SECURITY_SCANNERS
- **30**
ANTI_VIRUS_APPLICATION
- **31**
PKI_SYSTEMS
- **32**
PACKET_SNIFFER
- **33**
SYSLOGS
- **34**
RESPONSE_TIME_TEST
- **35**
RESPONSE_TIME_TEST_HOST
- **36**
TRANSPORT_SERVICE
- **37**
GENERIC_TL1_DEVICE
- **38**
VOIP
- **39**
CMTS
- **40**
WIRELESS
- **41**

CABLE_MODEM_MTA
- **42**
VPN
- **43**
DSL
- **44**
MULTIPLEXOR
- **45**
SAN
- **46**
PBX
- **47**
USER_CUSTOMIZATION
- **48**
PRINTER
- **49**
TRANSPORT_DEVICE
- **50**
SERVICE_MGT_COMPONENT
- **51**
SLA_COMPONENT
- **52**
CUSTOMER
- **53**
PROCESS
- **54**
DIAGNOSTIC_DATA
- **55**
DIAGNOSTIC_COMPONENT
- **56**
HOST_CONFIGURATION
- **57**
DIAGNOSTIC_SCRIPT
- **103**
POWER_SUPPLY
- **104**
AMPLIFIER
- **105**
LINE_MONITOR
- **106**
TEST_POINT
- **107**
FIBER_NODE
- **108**
HEFIBER
- **109**
IP_PHONE
- **110**
TELCO
- **111**

LOAD_BALANCER
- **112**
  LMT
- **113**
  FILE_SERVER
- **114**
  ENVIRONMENTAL
- **115**
  AUTOMATIC_TELLER_MACHINE
- **116**
  CONTENT_FILTER
- **117**
  APPLICATION_SERVER
- **118**
  IPRM
- **119**
  LAYER_3_TOPOLOGY
- **120**
  LAYER_3_PATH
- **121**
  UNMANAGED_DEVICE
- **122**
  SOFTSWITCH
- **123**
  SECURITY_APPLIANCE
- **124**
  EVENT_MODEL
- **125**
  TELECONFERENCE
- **126**
  NETWORK_OPTIMIZER
- **127**
  SERVICE_PROFILE
- **128**
  WIRELESS_LAN_CONTROLLER
- **129**
  DAS_CONTROLLER
- **130**
  VIRTUAL_SWITCH
- **131**
  VIRTUAL_MACHINE
- **132**
  HYPERVISOR
- **133**
  SERVER
- **134**
  CONTROLLER
- **135**
  VIRTUAL_ROUTER
- **136**

CLUSTER
– **137**
VMWARE_VCENTER

- **Model_Name**
Specifies the name that is given to the model. A model name distinguishes the model from others of that type.
- **Modeltype_Name**
Specifies the textual name of the model type.
- **Network_Address**
Contains the network address of the device model. DX NetOps Spectrum uses this value to identify and communicate with the model. This value is usually an IP address.
- **Network_Mask**
Further identifies the logical location for a device model on the network.
- **polling_interval**
Used by most device, application, and interface models. The value of this attribute identifies the number of seconds that must pass between polling requests. You can set this attribute in the Model Information view.
- **poll_log_ratio**
Identifies the polling cycle that logs the polled attributes. For example, a value of 10 means that the polled attributes are logged every tenth polling cycle. This attribute is found on most device, application, and interface models.
- **pollingstatus**
Controls if polling occurs for the model. Most device, application, and interface models use this attribute. A value of TRUE means that polling is enabled for the model. A value of FALSE means that polling is disabled for the model.
- **Security_String**
Defines the SNMP community strings that have access to this model. Most application, device, and interface models use this attribute.
- **System_OID_Verify**
Contains a System Object ID value that Discovery uses to determine the appropriate model type for representing a device. DX NetOps Spectrum assigns the model type when instantiating a model to represent a managed element. DX NetOps Spectrum identifies the appropriate model type by matching the System Object ID of the managed element to the System_OID_Verify value for that model type. If multiple model types are found to have a matching value, the model with the highest Disposable_Precedence value is used. If a model type is used to represent a family of devices, the SysOIDVerifyList attribute is used instead of the System_OID_Verify attribute.
- **SysOIDVerifyList**
Contains a list of System Object ID values that Discovery uses to determine the appropriate model type for representing a device. DX NetOps Spectrum assigns the model type when instantiating a model to represent a managed element. DX NetOps Spectrum identifies the appropriate model type by matching System Object ID of the managed element to a value from the model type SysOIDVerifyList list. If multiple model types are found to have a matching value, the model with the highest Disposable_Precedence value is used.
- **Vendor_Name**
Identifies the vendor of the managed element. Most device, application, and interface models use this attribute.
- **PortDuplex**
The PortDuplex attributes defines the duplex type of an interface. The following are the list of values available for PortDuplex attribute in DX NetOps Spectrum.
1. half duplex
2. full duplex
3. auto
4. unknown
5. disagree

Relation Descriptions

The following relations are among the most important for the DX NetOps Spectrum functionality:

- **Connects_to**
  The Connects_to relation is a many-to-many relation type. Connects_to forms a connection between two models. This relation establishes a connection between a port model and a device or topology model.
- **Contains**
  The Contains relation is a one-to-many relation type. This relation lets a model contain other models. The Location models use the Contains relation to determine the location or device models that can be contained within the location model. DX NetOps Spectrum tests the rules for this relation when you attempt to add or copy models to the Location models.
- **HASPART**
  The HASPART relation is a many-to-many relation type. HASPART establishes an association between a device model and the models that represent the components of the device. Typically, these components are the interface models of the device. However, this relation can also establish an association between a component model and its components.
- **Links_with**
  The Links_with relation is one-to-many relation type. This relation is used to represent a resolved connection between two models. This relation is found typically between two-port models. Both the Link view and Live Pipes rely heavily on this relation.
- **Manages**
  The Manages relation is a one-to-many relation type. This relation forms an association between an application model and the device model that is running it. The Manages relation can also form an association between an element management system model and the element models of this system.
- **Provides**
  The Provides relation is a one-to-many relation type. This relation identifies which applications provide which sub-applications. The Provides relation can also be used to determine which application provided a particular sub-application.

Sizer Tool

The DX NetOps Spectrum Sizer tool is an online web-based sizing tool. This multipurpose tool can provide you with the knowledge of the computer resources that are required to manage a given network that is based on the hardware platform of the choice of the customer. The following section contains detailed instructions for using the DX NetOps Spectrum Sizer tool.

Sizing

Sizing is the capability to estimate computer resource requirements for a given software application. The ability to provide accurate sizing information has become essential to the success of DX NetOps Spectrum. The sizing information is vital to the presales process and is used by customers, partners, and CA teams to assist customers in troubleshooting performance problems.

Working with Sizer

Working with the sizer tool is simple. The user enters in either actual data that is collected from a current network of the customer or virtual data for a network the customer expects to attain in the future. From this data, the sizer runs through a series of calculations. The built-in algorithms of DX NetOps Spectrum Sizer tool comes from performance testing and validation of the software with the supported computer systems. The DX NetOps Spectrum Performance Analysis Team continuously tests and validates the sizer algorithms as new versions of the software are developed.

Knowledge of the supported CA applications and their associated technologies is assumed.

Launching the Sizer

The DX NetOps Spectrum Sizer tool has a 25-minute time-out on the server. If you are idle for more than 25 minutes, you are automatically logged out and your sizing is discarded. Therefore, it is important to have all necessary sizing information with you before logging on to the server. Attempting a sizing without proper information is not recommended, as it most likely results in either an unfinished or inaccurate sizing. Read the Data Collection for DX NetOps Spectrum Sizer section in this article before attempting a sizing.

You can access the DX NetOps Spectrum Sizer tool through a web browser. Click the following link to access the Sizer tool.

http://consumer.interop.ca.com/sizer/

NOTE

Information! Chrome is the recommended browser for Sizer.

System Requirements for DX NetOps Spectrum Sizer

Browser

The DX NetOps Spectrum Sizer tool requires a browser that:

- Accepts cookies (must be turned on).
- Displays tables.
- Performs automatic redirection.

See Browser Configuration for a list of web browsers that are known to work and how to configure them.

Display

- Recommended: 1024 x 768 or higher with 256+ colors
- Minimum: 800 x 600 with 16+ colors
- 640 x 480 is not recommended

Browser Configuration

Microsoft Internet Explorer

Follow these steps to enable cookies:

- Open **Tools** menu, and click **Internet Options**.
- Click the Privacy tab.
- Click the **Advanced** button.
- Select '**Override automatic cookie handling**' checkbox under Cookies section.
- Select **Accept** radio buttons for '**First-party Cookies**' and '**Third-party Cookies**'.
- Select **Always accept cookies** checkbox.

What is a Cookie?

Cookies help a server (our online sizing server) to store information about a computer web session. The cookies that are used by our web server expire 25 minutes after your last access to the sizer. For Sizing, cookies are necessary, as the sizer requires tracking of your input as you go from one sizer page to another.

Sizer Registration

The registration process for DX NetOps Spectrum Sizer tool is a simple, one-time process. Once you are a registered user, you have access to all available options. To become a registered user, Click the '**Register**' link on the Sizer main page. At the register page, fill in at least the required fields to complete the registration. After registering with Sizer, there is a maximum one-business-day turnaround on sizer usage authorization (usually authorization is given within minutes of registering).

The registrations are currently accepted for resellers, outsourcers, and employees of CA Technologies and its partners.

NOTE

The sizer database security work is in progress, we recommend do not use your company password.

The following fields appear on the registration page. A field proceeded by a * is mandatory**. Please register only once.**

- *Enter a username
- *Enter a password
- *Retype password
- *First Name
- *Last Name
- *Email
- *User Type (Direct Sales, Reseller, End User, Support, Outsourcer, Other)
- *Organization (If CA Technologies then put department.)
- Street address
- Address (cont.)
- City
- State/Province
- Zip/Postal code
- Country
- Phone

Once you are a registered user, you can log in and can perform full sizings, create custom accounts, recall any and all previously completed sizings, and E-mail your results in plain text. If you forget your password, send an E-mail to spectrum-sizer.pdl@broadcom.com with your user name and we E-mail you the password.

Data Collection for DX NetOps Spectrum Sizer

Good preparation is the key here. A sizing generated with the sizing tool is only as accurate as the data entered into it. The more accurate the supplied data is, the more accurate the sizer output reflects the required resources to manage the network. The data that is needed for an accurate sizing includes:

- The name of the customer (Company Name, which is used as the Account Name in the sizer)
- Is the sizing for a 'new sale' or for 'support' of an existing customer
- The version of DX NetOps Spectrum to be sized
- The number and types of all devices (for example, switches/routers/hubs) to be managed
- The average number of active interfaces on the devices to be managed
- Number of servers and end systems to be managed
- The desired poll rate for each network device

(***See Appendix A for a fill-in-the-blanks DX NetOps Spectrum sizing worksheet***)

Sizer Login

Log in to use all the available options in Sizer.

Follow these steps:

1. On the main page, click **Start Sizing**.
   The login page appears.
2. Enter your registered sizer username and password.
   You have three attempts to enter your correct username and password before you are sent to the registration page.

After login, the ***Main Sizing Page*** *appears*. You can select either a 'New Sizing' OR a 'Recall Sizing'.

At any time during the sizing, you can always click the **Log Out** link at the top of any page to quit the sizer.

Account Setup

The sizers have been designed to use accounts. An account is used to harbor DX NetOps Spectrum sizings for a particular customer. Upon the first use of the sizer, you are required to enter an account name. Use a meaningful account name such as the name of the customer, this makes it easier for you to manage sizing information of your customer account. A new account name is only saved if sizing is completed for that account (that is, upon sizing for a new customer, complete one sizing to store the account name in our database for future sizing reference).

Recalling a Sizing

The sizer supports the ability to recall completed sizings that are stored in our database. A recalled sizing acts exactly the same as a regular sizing. You can change previously entered information, and then E-mail the recalled sizing.

Modes of a recalled sizing:

- *Just to see its final report output* - when recalling a sizing in this mode, only the end-results are available. You can view the results, e-mail them, or print them to see the end results.
- *As a starting point for a new sizing* - a recalled sizing in this mode uses the recalled sizing as a starting point for a new sizing. All the values of the recalled sizing become the default values for the new sizing. This mode is useful if you want to edit the data in a previous sizing to try different 'what if' scenarios.

  NOTE

  Not completed sizing is ever overwritten. Each completed sizing is assigned a unique id from our server. Therefore, if you recall a sizing to use as a starting point for a new sizing, and you do not change the sizing note or the account name, the only way to tell the sizings apart is by the time/date stamp.

Follow these steps to recall a sizing:

1. Log in to the DX NetOps Spectrum Sizer tool using your registered sizer username and password.
2. At the main sizing page, select **Recall Sizing**.
3. Choose the mode of recall and the account that the sizing belongs to.
4. Click **OK**.The Account Sizing information for the recall account appears.

Performing a Sizing

After you have successfully registered and are logged in, you are ready to size. Sizing is straightforward provided you have all the necessary information with you. When using the DX NetOps Spectrum Sizer tool, be sure to use the **built-in** sizer navigation buttons. If the browser navigation buttons are used, the sizer cannot track your session (**i.e. do not use the browsers 'BACK' button, instead use the 'BACK' button located at the bottom of the sizer page**).

Follow these steps:

1. Select Account Type, Sizing Type, SizingNote, and DX NetOps Spectrum version for sizing
   a. Create an account for the customer or select an existing account that you previously created. Each client/customer must have only one account; however, you can have multiple sizings for each account.
   b. Select the Type of sizing from either a 'New Sale' sizing or a 'Support' sizing.

c. Select the version of DX NetOps Spectrum to be sized.
d. Enter a short Note or keywords about the sizing. The 'Sizing Note' and date of sizing are used to differentiate your sizings.
e. Click NEXT.

2. Depending on the selection for 'Sizing Type' in Step#1, you see different options in Step#2.
   If you select 'New Sale', then you see the following options:
   a. Percentage of devices managed via SNMPv3: Approximate percentage of SNMP v3 devices (% out of the total managed devices).
   b. Percentage of devices managed via SDM: Approximate percentage of devices managed by SDM (% out of the total managed devices).
   c. Future growth: Approximate percentage of expected future device growth (% out of total managed devices).

   If you select 'Support', then you must enter details about the approximate forecast of the server capacity. You see the following options:
   a. Allowed SpectroSERVER capacity (the desired percentage CPU utilization per server); 50 percent is recommended to allow for network growth and transient conditions that require more CPU.
   b. Number of WebServers: Number of OneClick servers that user planned to set up.
   c. Number of Global Collection Elements (that is, 1000): Total number of GC elements that user plans to have across all global collections.
   d. Number of Alarms (that is, 2000): Approximate total number of alarms that are expected from all the managed devices.
   e. Number of Events/sec (that is, 50): Approximate number of events/sec expected from all the managed devices.
   f. Percentage of devices with active port monitoring (that is, pollPortStatus, LivePipes) % of devices that are expected to have active port monitoring.
   g. Percentage of devices managed via SNMPv3: Approximate percentage of SNMP v3 devices (percentage out of the total managed devices).
   h. Percentage of devices managed via SDM: Approximate percentage of devices that are managed by SDM (% out of the total managed devices).
   i. Future growth: Approximate percentage of expected future device growth (% out of total managed devices).
3. Select that the computer platforms/speeds for which the sizer uses when it calculates resource requirements (that is, select the machines which you want to size for). Select at least one computer. Click NEXT.
4. Select devices to be managed (use the completed worksheets from Appendix A for reference). By clicking the 'Save New Entries' button, you stay in step 4, all your devices are entered into the database and you are able to enter more devices (useful if you have more than five different types of devices). Click NEXT. If you have devices with different interfaces, you can divide the quantity of the devices according to interfaces and then enter the devices number in the Quantity field. You can enter interfaces for these devices in Step 5.
5. Enter the 'number of interfaces' for network devices that have a variable number of interfaces.

   NOTE
   If none of the devices you selected in step 4 have variable interfaces, you might not be able to do anything in this step. Click NEXT.
6. This step allows you to go back to any of the five previous steps and change your input. Backtracking is only necessary if the input you entered was incorrect or incomplete.
   Click FINISH to save the sizing and get the output.
   Once the sizing is saved in our database, you can recall the sizing in the future. You also have the option at the top of the screen of E-mailing the completed sizing.

   NOTE

   You can run the perfcollector9 script and can review the <machinename>_SizerInfo.txt file for some of this data (if DX NetOps Spectrum is already installed and running). This task helps you to verify whether they are running over the recommended capacity.

Interpreting the Sizer Results

The finished sizing is a one-to-two page summary of your input data, configuration recommendations of the sizer computer (server), and any other important information you should be aware of while interpreting the results.

DX NetOps Spectrum

The areas of interest for the DX NetOps Spectrum Sizer are the ***SpectroSERVER Sizing*** , and ***Please Note*** sections. The other (top) sections are simply a summary of the data you entered into the sizer.

Following is a partial example of a completed DX NetOps Spectrum sizing (that is, simply the areas of interest are shown).

SpectroSERVER Sizing

# Systems	RAM (GB)	%Utilized	Platform	CPU	Speed (GHz)	OS	Other Info
1	4	10	Linux Intel	Xeon quadcore	3	Red Hat® Enterprise Linux® 6.x	32/64-bit (64-bit only for 10.x)

The *SpectroSERVER* chart is a listing of characteristics about the machines you chose to size for. Each line represents a different type of computer and how many are required to run DX NetOps Spectrum with the given network characteristics.

The following table explains each column of the chart.

Metric	*Description*
#Systems	The number of server machines that are required to meet data input characteristics.
RAM	The minimum amount of memory each server computer must possess. For example, for 10.x, SS starts at around 500MB and reaches 10GB in the peak load of 1 million models.
%Utilized	The amount of server utilization per computer during steady-state polling.
Platform	The server computer type.
CPU	The server computer CPU type.
Speed (MHz)	The server computer CPU speed.
OS	The server computer operating system.
Other Info	Any additional information specific to the server computer.

Please Note:

- This initial sizing is based on the raw SpectroSERVER capacity that is required to manage the network described by the information that is provided to CA Technologies by the client. Actual requirements might vary depending on network architecture and the business requirements of the client.
- **For NT systems** a striped disk configuration consisting of a minimum of two 10,000 RPM SCSI drives is required to achieve this level of performance.

The ***Please Note*** information is a summary of any additional information you are aware of when interpreting and conveying the sizer results.

NOTE

If the tool returns two or more SpectroSERVERs required based on the input of the user, then the actual recommendation would be to have three SpectroSERVERs. One of them is a dedicated MLS with limited modeling while the other two servers would be used to do the actual modeling/management.

Email Last Sizing

A great feature of the DX NetOps Spectrum Sizer tool is the ability to e-mail the results obtained. Once sizing is completed, there is a link at the top of the screen that allows you to e-mail the calculated results to yourself, the customer, or other colleagues for their concurrence. The e-mail option is also available with the recall feature (that is, a recalled sizing can be e-mailed).

> **NOTE**
> Our server sends e-mail in plain text so you can read the results on all E-mail platforms.

Appendix A: DX NetOps Spectrum Sizing Worksheet

Customer Name: _____

Your name: _____

Your e-mail address: _____

The version of SPECTRUM to be sized: _____

Router Information:

Tier	# Routers	# Interfaces per Router	Poll Interval (sec)
1			
2			
3			

Managed Devices:

Qty	Device Type	Poll Interval (sec)

Number of servers and end nodes (workstations) to be managed, i.e. Winwatch:_____

The desired poll rate for each managed end node (default is 600 seconds):_____

Installing and Upgrading

This section provides information about the prerequisites, system requirements, and the relevant procedures that you need to follow to install and upgrade DX NetOps Spectrum and all the related components.

Service Pack and Patch Install

DX NetOps Spectrum engineering is committed to providing high-quality enhancements and features with each service pack to delight its customers and add value to their experience. Each service pack includes all the maintenance updates published prior to this update. By installing the appropriate service pack, you ensure that your systems are up-to-date with the latest information.

NOTE

To install a patch on a service pack or an existing major release, see DX NetOps Spectrum Patch General Information.

Patch Version	Key Features	Certification Packs and Bi-Monthly Packs Changes Included
10.4.2	Currently, no BMPs are available for 10.4.2.	Not Applicable
10.4.1	Currently, no BMPs are available for 10.4.1.	Not Applicable
10.4	The **10.4.0_BMP_10.4.001** pack includes various fixes. See the Bi-Monthly Patches (BMPs) on 10.4 for more information on the defects and fixes in the pack.	10.4.0_BMP_10.4.001
10.3.2	The **10.3.2_BMP_10.3.201** pack includes various fixes. See the Bi-Monthly Patches (BMPs) on 10.3.2 for more information on the defects and fixes in the pack.	10.3.2_BMP_10.3.201
10.3.1	The 10.3.1_BMP_10.3.102 includes the 10.3.1_BMP_10.3.101 fixes and fixes for other defects. See the Bi-Monthly Patches (BMPs) on 10.3.1 for more information on the defects and fixes in the cert pack.	10.3.1_BMP_10.3.102 10.3.1_BMP_10.3.101

10.3.0	The **10.3.0_BMP_10.3.002** includes the **10.3.0_BMP_10.3.001** fixes and other defects. See the Bi-Monthly Patches (BMPs) on 10.3 for more information on the defects and fixes in the cert packs.	10.3.0_BMP_10.3.002 10.3.0_BMP_10.3.001
10.2.3	• Expanded fault management support for Cisco® Application Centric Infrastructure (Cisco ACI™) software-defined data center (SD DC) controller • Support for VMware vSphere® environments, which is a common layer in a Cisco ACI environment • Multi-tenancy for SDx environment for managing different customers or different sites • Expanded support for Juniper Networks® SRX and EX Series devices • Additional new capabilities 1. a. a. Direct integration with CA Digital Operational Intelligence for fault and inventory data analytics b. Enhanced gateway probe for DX NetOps Spectrum – CA UIM integration and improved integration with CA Systems Performance for Infrastructure Managers (formerly CA SystemEDGE) c. 13 new dashlets and 3 new dashboards in conjunction with CA Business Intelligence d. Web client enhancements e. New device certifications	Spectrum_10.02.03.Cert_Pack_001
10.2.2	• Unified Dashboards and Reporting for Infrastructure Management • Multi-Tenant support in DX NetOps Spectrum • Alarm Filter for DX NetOps Spectrum and CA UIM Bidirectional Integration • OneClick Quick Device Search Enhancements • WebClient Enhancements	Spectrum_10.02.02.BMP_10.2.201 Spectrum_10.02.02.Cert_Pack_001

10.2.1	• DX NetOps Spectrum and CA UIM Integration Enhancements • Additional Event mapping from CA UIM Probes to DX NetOps Spectrum • ESX Host Server Maintenance Mode enhancement • SNMP Profile synchronization • SSL communication between Spectrum Gateway (spectrumgtw) probe and DX NetOps Spectrum • Launch CA App Experience Analytics (AXA) Dashboard from WebClient • CA Spectrum10.2.1 Platform Updates • Network Configuration Manager (NCM) Enhancements	Spectrum_10.02.01.BMP_10.02.101 Spectrum_10.02.01.Cert_Pack_002 Spectrum_10.02.01.Cert_Pack_001a

For more information, see the respective link: 10.3 Solutions and Patches and 10.2 Solutions and Patches.

NOTE

It is not recommended to install any patch unless you have the problem the patch is meant to resolve, or you anticipate having the problem.

Prerequisites

NOTE
If you are on 9.2.x, upgrade to 9.3 (or later) to directly upgrade to 10.2 first, and then to the latest service pack release (such as 10.2.3). Installing a base version, such as 10.2.0, is mandatory before installing a service pack.

Install Service Pack

1. Sign in to the support.broadcom.com site.
2. Click Enterprise Software.
3. Click Product Downloads.
4. Enter the product name (for example, Spectrum) in the search bar and click the search icon.
5. Click the link that is displayed after the search is completed
6. Select the release and version to download.
7. After the download is done, follow the instructions given on the install page.

Post Install Tasks

NOTE

After upgrading to 10.2.3, you must install CABI 4.1 SP3 for generating reports. The 10.0 release is not compatible with earlier versions of CABI. CABI 4.1 SP3 is a fresh installation and it does not support upgrades from CABI 4.1 SP3.x. For more information, see Upgrade CABI section.

- 10.2.3 supports JasperReports Server for generating reports. For more information, see CABI JasperReports Server (Windows) and CABI JasperReports Server Integration with DX NetOps Spectrum.
- To see the CABI (BOXI), CABI (JasperReports Server) compatibility with 10.2 and earlier versions of DX NetOps Spectrum, refer Integration Compatibility.

General Patch Information

DX NetOps Spectrum PTF (Program Temporary Fix) and Debug patches should only be applied if customers are experiencing the problem they are meant to resolve or debug. Both PTF and Debug patches are built for specific releases, these are listed as prerequisites for installing. After each Symptom and Resolution, there are two reference numbers listed in parenthesis, the first is the original defect reported to Sustaining Engineering, the second is the original customer support case opened with DX NetOps Spectrum support. CA testing of PTF and Debug patches is limited to the original use case in which the patch was created. As such, PTF and Debug patches are built with consideration of code conflicts for the original customer they were built for.

> **NOTE**
>
> Customers needing PTF and/or Debug patches that were not specifically built for them, should open a new support case for the request.

How to Obtain

The new case must include a copy of the <$SPECROOT>/Install-Tools/.history file and state what operating systems the patches are needed for.

DX NetOps Spectrum Patch Best Practices

Although all PTF and Debug patches can be uninstalled, it is recommended that customers first make a backup copy of their DX NetOps Spectrum install area. This should include a recent save of the DX NetOps Spectrum modeling database.

How to install a Patch on DX NetOps Spectrum

This video describes the environment, best practices, and a step-by-step procedure to install a patch.

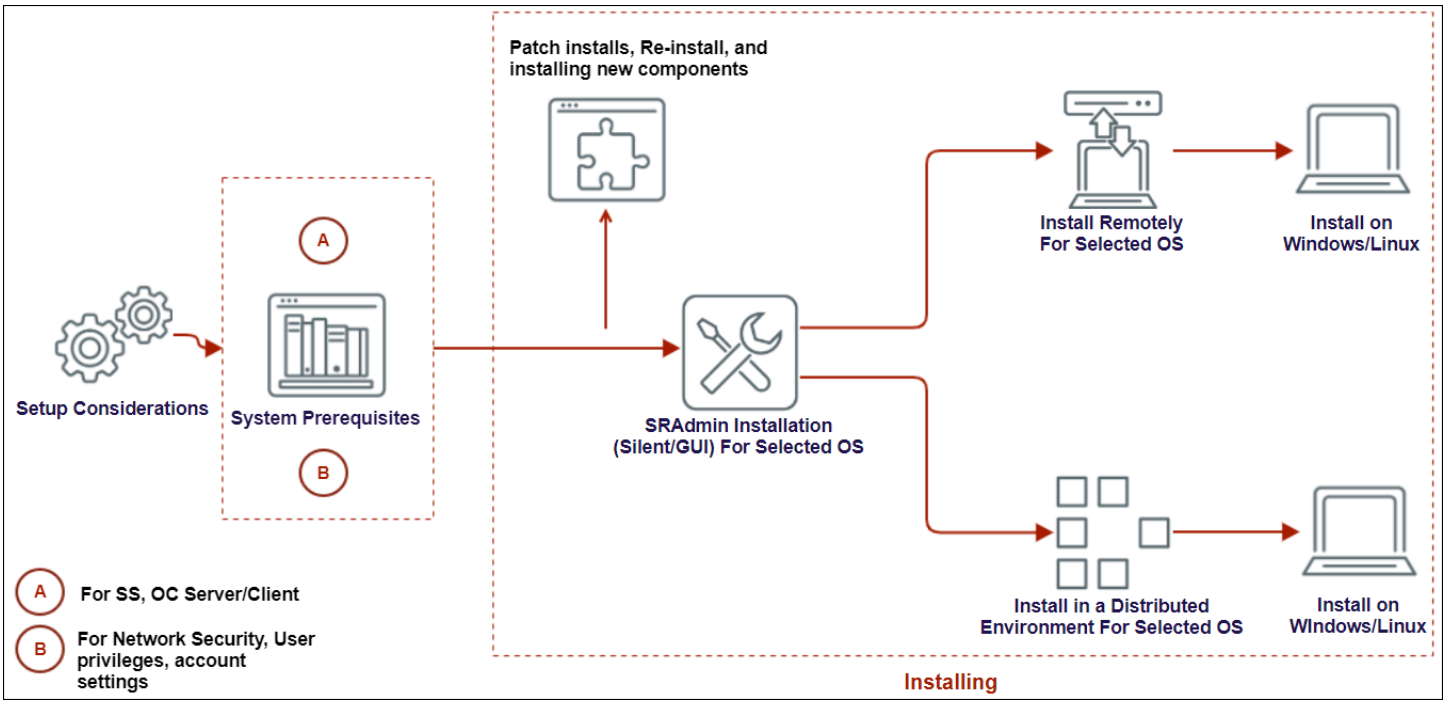
Fresh Install

Installing DX NetOps Spectrum for the first time?

Install the latest version of DX NetOps Spectrum to manage your underlying IT infrastructure effectively. Installing DX NetOps Spectrum with its various components, dramatically simplifies IT management by linking applications and services to the network for detailed traffic analysis, flows, and performance. This section provides you with a detailed step-by-step guide to install DX NetOps Spectrum.

Installation Process

The following diagram shows the overall process:



The tasks outlined in the diagram are as follows:

1. Review the system requirements and set up considerations to learn more about memory, processor speeds, and disk space.
2. Review the SpectroSERVER, OneClick Server and Client system requirements for Windows and Linux.
   Optional: To launch the OneClick Console and OneClick add-on applications, Install JRE for Windows and Linux.
3. Review the prerequisites page for the opted operating system such as Windows and Linux to set up the administrator privileges, user account controls, network, and security settings.
4. Install SRAdmin (through either GUI or silent mode on the selected operating system) to install DX NetOps Spectrum in a distributed environment or remotely before installing DX NetOps Spectrum on the select operating systems.
5. Install DX NetOps Spectrum on Windows and Linux.

Additional Information

- Files created during the installation
- Post-installation configurations
- Starting DX NetOps Spectrum OneClick web server, starting and launching the OneClick client.
- OneClick Web Server Upgrades and New OneClick Privileges

NOTE

Learn more about installing DX NetOps Spectrum Report Manager and Unified Dashboards and Reporting for Infrastructure Management.

System Requirements

WARNING
10.3 (or later) only supports 64-bit operating systems; it does not support 32-bit operating systems.

System Configurations

The configuration information in this section provides guidelines for running DX NetOps Spectrum at peak efficiency. You can achieve optimal system performance when all system resources are robust enough so that a single resource does not limit the other resources. System resources include memory, processor speeds, and disk space.

> **NOTE**
>
> Consult your support or sales representative for more help in determining the best configuration for your network.

You cannot define all configurations and system requirements for all users because of the following complexities and variables:

- Polling frequency
- Device types
- Number of devices in a network

> **WARNING**
> Installing OneClick on a single-CPU SpectroSERVER host system can degrade the performance of both SpectroSERVER and OneClick. To maximize the performance of both, we recommend that you install OneClick on a separate, dedicated computer. If you upgrade SpectroSERVER components, you might also need to upgrade OneClick.

System Support and Setup Considerations

Virtualization Environments

DX NetOps Spectrum supports the following virtualization environments for Windows and Linux

> **NOTE**
> For information about CA VMware guidelines, see Virtualization.

Rule of Localization Homogeneity

The *Rule of Localization Homogeneity* states that all components in a distributed DX NetOps Spectrum installation must run on servers that use the same operating system Locale. Think of DX NetOps Spectrum as one application running with one language, rather than as a set of distributed services potentially running different languages.

By following the *Rule of Localization Homogeneity*, you ensure that all access and modification of data through different communication paths use one consistent language. Otherwise, myriad languages can be stored in DX NetOps Spectrum database. The multiple languages cause problems with such data issues as display, fonts, searching, and sorting.

We recommend that you set the Locale on the servers that run DX NetOps Spectrum processes before you install DX NetOps Spectrum. Such servers include the Location server, Processd, SpectroSERVERs, OneClick servers, clients, and the Secure Domain Manager.

Disk Striping and RAID

For optimum performance, you can run DX NetOps Spectrum on systems with multiple, ultra-wide, 10,000 RPM, SCSI disk drives that use disk striping or RAID (redundant array of independent disks) technologies.

Disk striping is a technique of spreading data over multiple disk drives. RAID is a disk drive system that uses two or more drives in combination for fault tolerance and performance improvement.

Symantec PCAnywhere

Symantec pcAnywhere™ can cause Java to lock and prevent Java applications from launching. Java applications include OneClick Console, DX NetOps Spectrum Control Panel, Model Type Editor, and Performance View. The processes start, but the GUIs do not launch. If you stop PCAnywhere, Java-based applications launch and the GUIs display correctly.

To resolve this issue, install or upgrade to DirectX version 9.0 B, which is available at http://support.microsoft.com.

Alternatively, you can stop the PCAnywhere Host Service before installation. After you install DX NetOps Spectrum and OneClick, you can enable the PCAnywhere Host Service again.

Antivirus Software and Data Backup

DX NetOps Spectrum does not include antivirus software. We recommend installing your preferred antivirus software to protect your networking environment.

> **WARNING**
> To avoid database corruption, exclude DX NetOps Spectrum installation areas and DX NetOps Spectrum files from scans by any local or remote instances of antivirus software. DX NetOps Spectrum installation areas include OneClick and Report Manager installation areas.

Exclude all DX NetOps Spectrum installation areas, including OneClick and Report Manager installation areas, from scans by data backup programs.

Swap Space Recommendations

Windows

> **NOTE**
> DX NetOps Spectrum 10.4.2 has not been validated on Windows Server 2012. However, Broadcom will support any DX NetOps Spectrum product issues, if found. We reserve the right to have you upgrade to Windows Server 2016 (or later) if deemed necessary.

Amount of Installed RAM	Recommended swap space
16- 32 GB	16 GB

Red Hat Enterprise Linux

> **WARNING**
>
> From 10.2.2 and above, support for RHEL 5.x is discontinued. From 10.4.1, support for RHEL 6.x is discontinued.

Amount of installed RAM	Recommended swap space	Recommended swap space if allowing for hibernation
2GB - 8GB	The same amount of RAM	2 times the amount of RAM
8GB - 64GB	At least 4GB	1.5 times the amount of RAM

^ Disk Drives: For optimum performance, run DX NetOps Spectrum on systems with Multi-disk RAID configurations @ 10,000 RPM or greater.

SpectroSERVER and OneClick Requirements

> **WARNING**
> 10.0 and higher versions support only 64-bit operating systems. They do not support 32-bit operating systems.

SpectroSERVER and OneClick Servers

The following table lists the system configurations necessary for SpectroSERVER and OneClick.

NOTE
The following table assumes that each SpectroSERVER manages fewer than 4000 devices with an average of 100 ports on each device.

Component	**Platform**	**SpectroSERVER Only**	**OneClick Server Only**	**SpectroSERVER and OneClick Server**	**OneClick Server Only with Integrations Enabled** (for example, with DX CAPM, DX UIM, DX ServiceDesk, DX VNA and so on)
Memory (RAM) Actual requirements are dependent on the configuration and number of managed devices.	Windows and Linux	8 GB Minimum 32 GB Recommended	8 GB Minimum 16 GB Recommended	16 GB Minimum 32 GB Recommended	32 GB Minimum
Processor	Windows and Linux	Quad-core processor (Minimum Single Core)	2+ GHz dual processor	2.5+ GHz dual processor	2.5+ GHz dual processor
Disk^°	Windows and Linux	Serial Attached SCSI disks @ 10,000 RPM or greater	Serial Attached SCSI disks @ 10,000 RPM or greater	Serial Attached SCSI disks @ 10,000 RPM or greater	Serial Attached SCSI disks @ 10,000 RPM or greater

^ Disk Drives: For optimum performance, run DX NetOps Spectrum on systems with at least Serial Attached SCSI disks @ 10,000 RPM or greater.

WARNING

10.1 and higher versions do not support 32-bit Java clients.

NOTE

64-bit clients are required to take advantage of the increased capacity of 64-bit CA Spectrum 10.1. However, if the additional scale of 64-bit is not needed, and if required, the 32-bit Java client should perform adequately. As a general rule, the maximum heap size of 32-bit clients on Windows systems will range from 1.4 to 1.6 GB of memory.
If this is exceeded the client will **NO LONGER LAUNCH** until a 64-bit client is utilized.

For more information on memory requirements for OneClick web server, see Configure OneClick Web Server Memory Settings.

OneClick Considerations

OneClick consists of a web server-based component and a client-based component, each of which requires different software. The OneClick web server includes the following items:

- Apache Tomcat servlet engine
- MySQL database management system
- Java 2 Software Development Kit (SDK)

NOTE
DX NetOps Spectrum supports only the version of the JDK that ships with OneClick.

The OneClick client includes the following item:

- Java Runtime Environment (JRE) with Java Web Start.

If Microsoft Internet Explorer 10 or later version is installed on the OneClick clients, set the browser security level to medium-low to avoid security-related issues. Or, if the Internet Explorer 10 or later version security level is high, be sure to add the OneClick website to the list of Trusted Sites.

By default, the OneClick website is automatically run in Compatibility Mode if the OneClick web server is installed within an intranet. In Internet Explorer 10 or later versions, intranet sites run in Compatibility Mode by default. Run Internet Explorer in Compatibility Mode to support the DX NetOps Spectrum - CA Service Desk Manager integration and Spectrum Report Manager. For more information, consult the Microsoft website.

OneClick and Report Manager Considerations

If you are installing OneClick with Report Manager, see the Install Report Manager for installation information.

The following table lists the system configuration for OneClick with Report Manager running in the environment.

Component	Platform	OneClick with Report Manager
Memory (RAM)	Windows and Linux	8 GB Minimum 16 GB Recommended
Processor	Windows and Linux	2.5+ GHz dual processor
Disk°†	Windows and Linux	Serial Attached SCSI disks @ 10,000 RPM or greater

* We highly recommend a separate, dedicated system for Report Manager. So, install OneClick with Report Manager and use this OneClick instance only for SRM activities.

[a] Assumes that Report Manager is monitoring the SpectroSERVER (polling for event data and asset change data). Otherwise, 1536 MB is acceptable.

^ Or any comparable Intel x86 1.5 GHz or better processor.

° Disk Drives: For optimum performance, run DX NetOps Spectrum on systems with at least Serial Attached SCSI disks @ 10,000 RPM or greater.

OneClick and Service Manager Considerations

If you are installing OneClick with Service Manager, the following recommendations should be considered:

- Service Manager must be installed on the computer where the SpectroSERVER is installed *and* on the computer where OneClick is installed. The modeling catalog and all modeling intelligence exist within the SpectroSERVER database. The historical database and event handling code exist on the OneClick web server, which is installed with OneClick.
- We recommend that you also install Service Manager when you install Report Manager with OneClick. Installing these components together ensures that the Service and SLA Reporting tables are populated.

  NOTE
  For more information about Service Manager, see Service Manager.

Install JRE

The OneClick Console and OneClick add-on applications require Java Runtime Environment (JRE). JRE includes the Java Web Start client, which is required to run Java Network Launching Protocol (JNLP) applications like OneClick.

NOTE
You must perform the steps recommended in this topic on the DX NetOps Spectrum client side.

After you install JRE, you can start OneClick.

NOTE
10.4.2 supports AdoptOpenJDK JRE (8u242). You can, however, use your corporate license with Oracle to run the Oracle JRE and launch the DX NetOps Spectrum Web Start.

Prerequisites to Install JRE and Java Web Start

To run the OneClick Console on your Windows system, install JRE and Java Web Start. Confirm the following prerequisites before proceeding:

- You have the correct URL for the OneClick web server system.
- You can access the OneClick web server system using HTTP on a web browser.
- Your account allows you to log in to the OneClick web server.
- You have downloaded JRE from AdoptOpenJDK/licensed Oracle JRE, which supports Java Web Start.

Install AdoptOpenJDK JRE with Java Web Start on Windows

To run the OneClick Console on your Windows system, install the AdoptOpenJDK JRE and Java Web Start.
Follow these steps:

1. Log in to your Windows system.
2. Open the OneClick home page in a browser using the URL that your administrator provided. The URL has the following format: *http://hostname:portnumber/*

   NOTE
   Hostname is the name of the OneClick web server. Use portnumber only if the OneClick web server does not use the default port 80. If you cannot access the OneClick web server, notify your administrator.
3. Enter your OneClick login credentials, if prompted. The OneClick home page opens.
4. Navigate to the Install JRE page.
5. Download and launch the installer.
6. Read and accept the license.
7. On the Custom Setup screen, select the features that you want to install.
8. (Optional) Change the default installation directory.
9. Click the directory tree with (x) checkmark to select the following additional features.
   - Set JAVA_HOME variable
   - IcedTea-Web

   NOTE
   These features are installed on the local drive.
10. Click Next.
11. Click Install.
12. When the installation completes, click Finish to close the program.

Associate .jnlp Files with Java Web Start

The file that launches OneClick is a JNLP file. Verify that the .jnlp file extension is mapped to the javaws.exe application.

Follow these steps:

1. Open Windows Explorer.
2. Select Tools, Folder Options. The Folder Options dialog opens.
3. Select the File Types tab. A list of registered file types is displayed.

4. Scroll down and select JNLP. The bottom portion of the dialog displays Details for the ‘JNLP’ extension.
5. Verify that the details for the ‘JNLP’ extension box indicate the following information:
   – **For Windows 2000/2003:** The file opens with javaws.
   – **For Windows XP:** The file opens with Java Web Start Launcher.

   If JNLP files are not set as described, you can manually map the .jnlp extension to the javaws.exe application.
6. Select Change in the Details for ‘JNLP’ extension box. The Open With dialog opens.
7. Scroll down and select ‘javaws’ or ‘JavaTM Web Start Launcher’ and select OK.
8. Select OK in Folder Options.
9. Exit Windows Explorer. The .jnlp file extension is now mapped to the Java Web Start application and can launch the OneClick Console.

Install JRE and Java Web Start on Linux

To run the OneClick Console on your Linux system, install JRE and Java Web Start. On Linux platforms, Oracle no longer provides self-extracting installers. Instead, they provide a tarball that contains the JRE binaries, but does not set any environment variables. To run the OneClick client on Linux, you can download the JRE from the OneClick web page and can associate the .jnlp file type with the Java Web Start application, javaws, using the Mozilla Firefox web browser.

Follow these steps:

1. Log in to your Linux system.
2. Open the OneClick home page in a web browser by using the URL that your administrator provided. The URL has the following format: *http://<hostname>:<portnumber>/*

   NOTE
   <*hostname*> is the name of the OneClick web server. Use <portnumber> only if the OneClick web server does not use the default of port 8080. If you cannot access the OneClick web server, notify your administrator.
3. Enter your OneClick login credentials, if prompted. The OneClick home page opens.
4. Download the JRE (tar.gz) from the OneClick Administration page and save the tar.gz file.
5. Open a terminal session (bash shell or kshell) and execute the following command to extract the binaries:

   ```
   tar -zxvf file_name
   ```
6. After extraction, execute the following commands to set the environment variables:

   ```
   export JAVA_HOME=Path_of_Extracted_Folderexport PATH=$PATH:$JAVA_HOME/bin
   ```

   where “Path_of_Extracted_Folders” corresponds to the location of the binaries after you have extracted them.
7. Associate the .jnlp file type with the Java Web Start application for the OneClick Console to launch:
   a. Select Start Console in Firefox.
   b. Select Open with and select javaws, in the JRE directory (<*JRE*>/bin/javaws).
   c. Select OK.

   The .jnlp file type is now associated with Java Web Start.

Launch 32- or 64-Bit Java Console when Both 32- and 64-Bit Java are Installed on a Client Machine

If you want to force 64-bit or 32-bit OC client, replace <resources> tag with one that specifies an architecture - "amd64" for x64 and "x86" for x32, see the following examples.

You can copy oneclick.jnlp to oneclick32.jnlp or oneclick64.jnlp, and can customize platform and memory, and then new launch points are added automatically to the OC admin page.

NOTE

For arch specifiers to work properly, you must exactly match the JRE version or ensure the "Allow new versions" checkbox is checked in the supported JRE version configuration.

Modify the following values in the oneclick.jnlp file:

- For 32-bit - oneclick32.jnlp:

<resources arch="x86"> for x32

<j2se version="1.8.0_112" href="http://java.sun.com/products/autodl/j2se"
initial-heap-size="96m" max-heap-size="**512m**"/>

- For 64-bit - oneclick64.jnlp:

<resources arch="amd64"> for x64

<j2se version="1.8.0_112" href="http://java.sun.com/products/autodl/j2se"
Initial-heap-size="96m" max-heap-size="**1024m**"/>

Follow these steps:

1. Navigate to: *$SPECROOT/tomcat/webapps/spectrum/oneclick.jnlp*
2. Copy the oneclick.jnlp to oneclick32.jnlp or oneclick64.jnlp in the same location.
3. Navigate to the DX NetOps Spectrum OneClick Web Server home page to launch the OneClick console.
4. Do one of the followings:

- Select x32 to launch a 32-bit OneClick client
- Select x64 to launch a 64-bit OneClick client

System Requirements for Linux

Linux System Requirements

The following table summarizes DX NetOps Spectrum support for Linux operating systems:

System Requirement/ DX NetOps Spectrum Component	OneClick Client	OneCLICK SERVER	SpectroSERVER
Operating System	Red Hat Enterprise Linux 8.0 and 8.1 (64-bit) Red Hat Enterprise Linux 7.x (64-bit) X-based desktop environment (such as KDE or GNOME)	Red Hat Enterprise Linux 8.0 and 8.1 (64-bit) Red Hat Enterprise Linux 7.x (64-bit)	Red Hat Enterprise Linux 8.0 and 8.1 (64-bit) Red Hat Enterprise Linux 7.x (64-bit)
Memory (RAM) [1]	Dependent on the configuration and number of managed devices	Dependent on the configuration and number of managed devices	Dependent on the configuration and number of managed devices
Processor	Dependent on the configuration and number of managed devices	Dependent on the configuration and number of managed devices	Dependent on the configuration and number of managed devices
Disk Space Dependent on the configuration and number of managed devices	Dependent on the configuration and number of managed devices	HDD:100 GB (Minimum 50 GB).	HDD:100 GB (Minimum 50 GB).
Graphical User Interface	X11 system that the JRE supports	Motif	Motif
Java Components	AdoptOpenJDK JRE (8u242)	AdoptOpenJDK JRE (8u242)	AdoptOpenJDK JRE (8u242)
Web Browser	Firefox 10.0 or later	Firefox 10.0 or later	Firefox 10.0 or later

System Requirement/ DX NetOps Spectrum Component	OneClick Client	OneCLICK SERVER	SpectroSERVER
Video System	A video card that supports 32-bit color at 1024x768 pixel resolution and a 20" monitor or larger.	A video card that supports 32-bit color at 1024x768 pixel resolution and a 20" monitor or larger.	A video card that supports 32-bit color at 1024x768 pixel resolution and a 20" monitor or larger.
PDF Document Viewer	Acrobat Reader X or later	None	None
Packages	See "Required Packages" below.	See "Required Packages" below.	See "Required Packages" below.

NOTE

1. See Swap Space Recommendations.
2. From 10.2.2 and above, support for RHEL 5.x is discontinued. From 10.4.1, support for RHEL 6.x is discontinued.
3. 10.3 (or later) Report Manager uses CABI r6.4.x as the report delivery engine.
4. For information about the platforms that CA Business Intelligence JasperReports® Server - 6.4.x supports, see System Requirements.
5. Do not set your foreground font color to white. If this font color is set to white, you cannot read the text on your screen during the installation.
6. Install Perl 5.26.2, if not available.
7. For more information on memory requirements for OneClick web server, see Configure OneClick Web Server Memory Settings.

Required Packages

Install the following RPMs and its dependencies, for the respective RHEL version:

NOTE
For purposes of identification, 32-bit RPMs for Red Hat Enterprise Linux contain "i386" or "i686". The 64-bit RPMs contain "x86_64".

Package	64-bits	32-bits	noarch
abrt	x		
elfutils-libelf	x		
expat	x	x	
fontconfig	x	x	
freetype	x	x	
glibc	x	x	
ksh	x		
libaio	x		
libcanberra-gtk2	x		
libgcc	x	x	
libICE	x	x	
libjpeg / libjpeg-turbo	x	x	
libnsl		x	
libpng	x	x	
libSM	x	x	

Package	64-bits	32-bits	noarch
libstdc++	x	x	
libuuid	x		
libX11	x	x	
libXau	x	x	
libxcb	x	x	
libxcrypt			x
libXdmcp		x	
libXext	x	x	
libXext		x	
libXfont2		x	
libXft	x	x	
libXi	x	x	
libXi		x	
libxkbfile		x	
libXmu	x	x	
libXp	x	x	
libXrender		x	
libXrender	x	x	
libXt	x	x	
libXtst	x	x	
libXtst		x	
motif (RHEL 7.x)	x	x	
ncurses	x		
ncurses-common-libs			x
ncurses-compat-libs	x		
ncurses-libs	x		
nss-softokn-freebl	x	x	
numactl-libs	x		
PackageKit-gtk-module/ PackageKit-gtk3-module	x		
perl-LWP-Protocol-https			x
sssd-client		x	
xorg-x11-fonts-misc	x		
xorg-x11-server-common		x	
xorg-x11-server-Xvfb1	x		
xorg-x11-xkb-utils		x	
xterm	x		
zlib	x	x	

NOTE

1. Required only for OneClick WebApp. See, Steps to Run OneClick WebApp on Linux.

SpectroSERVER Server Minimum Requirements for Virtual Machine

You can allocate dedicated memory (RAM) to your Virtual Machine by enabling the **Reserve all guest memory (All locked)** option. By allocating dedicated memory, you ensure that your VM's memory is not shared with other VMs on the same ESX.

For more information on other VMware configuration, see Virtualization.

System Requirements for Windows

Windows System Requirements

The following table summarizes DX NetOps Spectrum support for Microsoft Windows operating systems.

System Requirement/ DX NetOps Spectrum Component	OneClick Client	OneClick Server	SpectroSERVER
Operating System	Windows 8.1 Windows 10 Pro (64-bit) Windows Server 2016 Windows Server 2019	Windows Server 2016 Windows Server 2019	Windows Server 2016 Windows Server 2019
Memory (RAM) Dependent on the configuration and number of managed devices	8 GB (Minimum 4 GB)	8 GB	16 GB (Minimum 8 GB)
Processor	Dependent on the configuration and number of managed devices.	Dual-core processor	Quad-core processor (Minimum Single Core)
Disk Space Dependent on the configuration and number of managed devices	Dependent on the configuration and number of managed devices	50 GB	100 GB (Minimum 50 GB)
Java Components	AdoptOpenJDK JRE (8u242) Java 8 with x64 JRE is mandatory	AdoptOpenJDK JRE (8u242)	AdoptOpenJDK JRE (8u242)
Web Browser	• Firefox 10.0 or later (verified version - Firefox 39.0) • Internet Explorer 8.0 or later (verified version - 11.0) Internet Explorer 10 and later versions entail some special requirements. For more information, see OneClick Considerations. • In Windows 10, only IE 11 is supported and the default browser EDGE is not supported for OneClick Client[5]	Firefox 10.0 or later	Firefox 10.0 or later

System Requirement/ DX NetOps Spectrum Component	OneClick Client	OneClick Server	SpectroSERVER
Video System	A video card that supports 32-bit color at 1024x768 pixel resolution and a 20" monitor or larger.	Same	Same
PDF Document Viewer	Acrobat Reader X or later	Same	Same

NOTE

1. DX NetOps Spectrum 10.4.2 has not been validated on Windows Server 2012. However, Broadcom will support any DX NetOps Spectrum product issues, if found. We reserve the right to have you upgrade to Windows Server 2016 (or later) if deemed necessary.
2. 10.3 (or later) Report Manager uses CABI r6.4.x as the report delivery engine.
3. For information about the platforms that CA Business Intelligence JasperReports® Server - 6.4.x supports, see System Requirements.
4. For more information on memory requirements for OneClick web server, see Configure OneClick Web Server Memory Settings.
5. Microsoft EDGE is supported from DX NetOps Spectrum 10.4.2.2 release.

SpectroSERVER Server Minimum Requirements for Virtual Machine

You can allocate dedicated memory (RAM) to your Virtual Machine by enabling the **Reserve all guest memory (All locked)** option. By allocating dedicated memory, you ensure that your VM's memory is not shared with other VMs on the same ESX.

For more information on other VMware configuration, see Virtualization.

Prerequisites

Before beginning with the DX NetOps Spectrum the installation process, we recommend that you consider the following:

- Prerequisites for Windows
- Prerequisites for Linux

WARNING
10.4 (or later) supports 64-bit operating systems; it does not support 32-bit operating systems.

IMPORTANT
You must first install 10.4.2 as the base version to install DX NetOps Spectrum 10.4.2.1 or 10.4.2.2 release.

Prerequisites for Windows

NOTE

Ensure that you meet the following prerequisites before you install DX NetOps Spectrum on a Windows system.

Administrator Privileges

To install DX NetOps Spectrum, log in as Administrator or as a user with administrator privileges.

NOTE
If you plan to install DX NetOps Spectrum as a user other than Administrator, turn off User Account Control (UAC).

DX NetOps Spectrum installation software requires administrator privileges to evaluate available resources and to run custom installation scripts. An initial installation generates residual files with administrator ownership. Subsequent upgrade installations also require administrator privileges.

User Account Control (UAC)

To install DX NetOps Spectrum as a user other than Administrator, User Account Control (UAC) must be turned off. UAC is a Windows security component that prompts you for permission when a task requires administrator privileges.

During DX NetOps Spectrum installation, the installation user is elevated to an administrator in DX NetOps Spectrum Remote Administration (SRAdmin). UAC would then need to prompt the user for permission, which is not possible because SRAdmin is a non-interactive service. Disabling UAC allows the installation to run continuously.

> **NOTE**
> Disabling UAC is required for DX NetOps Spectrum installation as a user other than the Administrator only. After installation and during normal DX NetOps Spectrum operation, UAC can be enabled.

How to Disable UAC on Windows Server 2016 (or Later)

To install DX NetOps Spectrum as a user other than Administrator, UAC must be turned off. The following procedure describes how to disable UAC:

> **WARNING**
> DX NetOps Spectrum 10.4.2 has not been validated on Windows Server 2012. However, Broadcom will support anyDX NetOps Spectrum product issues, if found. We reserve the right to have you upgrade to Windows Server 2016 (or later) if deemed necessary.

Follow these steps:

1. From the Start menu, select Control Panel, User Accounts.
2. Click "Change User Account Control settings".
3. Move the slider down to the bottom line, Never notify, and click OK.
4. Disable UAC in Local Security Policy. UAC is now disabled.

How to Disable UAC in Local Security Policy on Windows Server:

In addition to the UAC setting, a Local Security Policy option for UAC must also be disabled, as described in the following procedure.

Follow these steps:

1. In a Run dialog, enter secpol.msc and click OK.
2. In the Local Security Policy window, select the Security Settings, Local Policies, Security Options folder.
3. Right-click the "User Account Control: Run all administrators in Admin Approval Mode" policy, and select Properties.
4. Select Disabled, and click OK.
   The necessary Local Security Policy option for turning off UAC has been disabled.

Fixed IP Address on Windows

Ensure that the system on which you want to install DX NetOps Spectrum has a fixed IP address. You can enable DHCP on the system when the DHCP server issues a static address that never changes.

Emergency Repair Disks

We recommend that you create an emergency repair disk (ERD) before installing DX NetOps Spectrum, because the installation can corrupt files. You can use the ERD to restore Windows configuration files. We recommend that you also create an ERD after successfully installing DX NetOps Spectrum.

NOTE
Windows Help files contain detailed instructions on creating an ERD.

Network and Security Settings

To use email for applications (such as DX NetOps Spectrum Enterprise Alarm Manager), configure the user profile that is logged in and running DX NetOps Spectrum to send email using the supported service provider.

Set security as required for the directory where you install DX NetOps Spectrum. If you set the security before installation, DX NetOps Spectrum preserves the changes to the directory hierarchy security.

NOTE
Restart your system after you make changes.

User Audit

If the user auditing feature is enabled on Windows, every action is audited, resulting in many entries in the Windows Event Log. We recommend that you disable the Windows user auditing feature because it slows DX NetOps Spectrum system performance.

Convert the File System to NTFS

We recommend that you install DX NetOps Spectrum in an NTFS file system partition. If your disk drive is formatted as a FAT partition, convert the Windows file system to NTFS.

Follow these steps:

1. Run the CONVERT utility at the command prompt as per the following syntax:
   ```
   CONVERT C: /FS:NTFS
   ```
   NOTE
   You can run the conversion utility without damaging or deleting existing data. If the hard drive is already converted, a message appears. You can get more help on the convert command by typing '"help convert" in your command-line interface.
2. Restart your system for the reformatting to take effect. The file system is converted.

Prerequisites for Linux

Ensure that you meet the following prerequisites before you install DX NetOps Spectrum on a Linux system.

Root Privileges

DX NetOps Spectrum and DX NetOps Spectrum Remote Administration (SRAdmin) Daemon installations require root privileges to evaluate available resources and run custom installation scripts. Installing under the root ensures root privileges for the setuid executable, which lets the SpectroSERVER connect to SNMP ports. Because an initial installation generates residual files with root ownership, subsequent upgrade installations also require root privileges.

Fixed IP Address

Ensure that the system on which you plan to install DX NetOps Spectrum has a fixed IP address. You can enable DHCP on the system when the DHCP server issues a static address that never changes.

NOTE
DX NetOps Spectrum should only be installed on a static IP address that is from the Network Interface Card and cannot be installed on a sub-interface.

Hosts File

Ensure that the /etc/hosts file has the following format:

```
127.0.0.1 localhost localhost.localdomain
<external IP> <external names>
```

- ***external IP***
  Is the static DNS IP of the host.
- ***external names***
  Are the DNS short names.

Make sure that the first line has the local host after the loop-back address. The loop-back line must have localhost as the official host name.

You can add nicknames after the local host. For example:

```
127.0.0.1 localhost localhost.localdomain
```

The following example is incorrect and would cause host resolution and security problems with DX NetOps Spectrum:

```
127.0.0.1 <external name> localhost localhost.localdomain
```

NFS-Mounted File Systems

For DX NetOps Spectrum install directories that are an NFS-mounted file system, we recommend that you set the NFS mount options to ‘hard’ and ‘nointr’. These settings help ensure database consistency.

However, as there are many possible problems with such a setup, including severe performance impacts, we recommend that you avoid NFS mounts, if possible.

If you *do* use the 'hard' and 'nointr' NFS mount options, take extra care to obtain good database backups.

We do not recommend the 'soft' option at any time.

SE Linux Configuration with RHEL

You can install this release of DX NetOps Spectrum on SE Linux in the Enforcing mode with 6.x. By default, with RHEL 7.x, SE Linux is set up in the Enforcing mode.

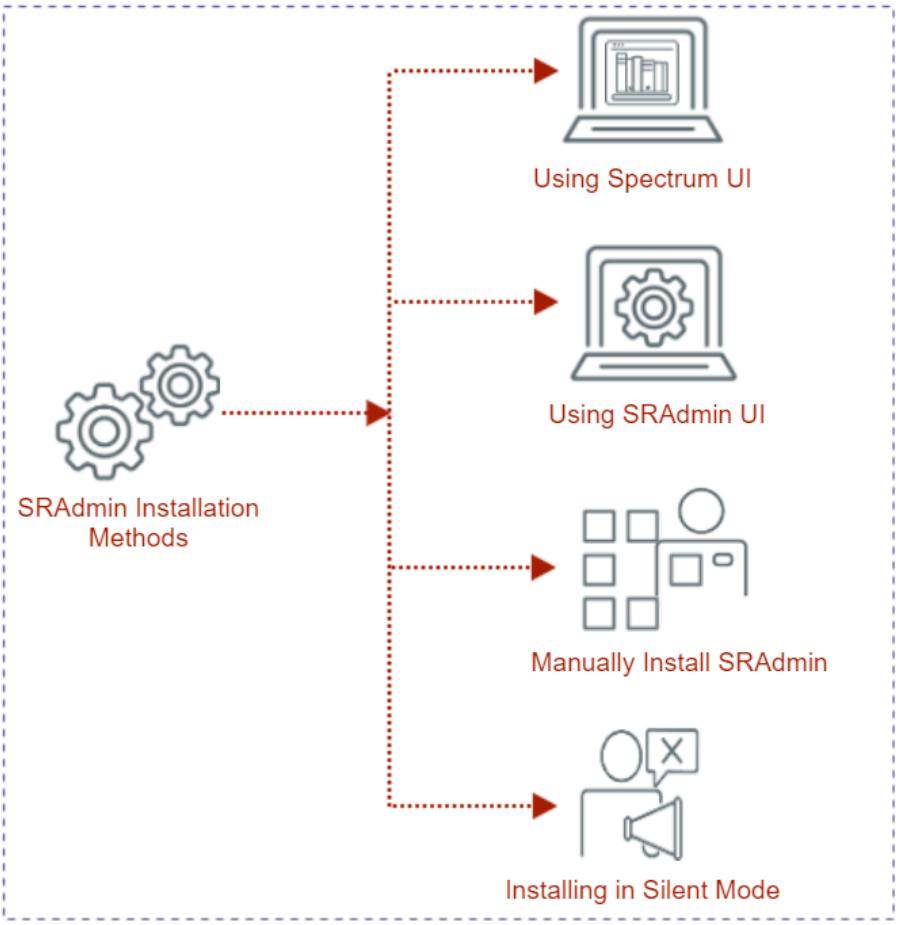
You can change to the permissive mode by editing the following file: /etc/selinux/config

The line that can be adjusted is SELINUX=permissive.

SRAdmin Installation Methods

Installation Methods

The following diagram shows the SRAdmin installation methods at a high level:



As outlined in the diagram, you can install SRAdmin through the following methods:

- DX NetOps Spectrum GUI
  If you install DX NetOps Spectrum on a local computer from the DX NetOps Spectrum GUI, SRAdmin can be installed during the installation process. You cannot install SRAdmin from the DX NetOps Spectrum GUI when you install remotely. If you are using the DX NetOps Spectrum GUI locally to upgrade from a release earlier than 9.0, SRAdmin cannot be automatically upgraded on the local machine. Instead, you have the option to install it.
- SRAdmin GUI
  Install SRAdmin from the SRAdmin GUI when you want to perform a remote installation or a distributed installation of DX NetOps Spectrum. You can also install SRAdmin from the SRAdmin GUI as an alternative to using the DX NetOps Spectrum GUI installation to perform a local DX NetOps Spectrum installation.
- Manually install SRAdmin on Linux and Windows platforms

  NOTE
  As an alternative to the GUI install options, install SRAdmin manually.
- Install SRAdmin in silent mode on Linux and Windows platforms

  NOTE
  As an alternative to the GUI and manual install options, install SRAdmin in silent mode.

Install SRAdmin Daemon

Install SRAdmin from the SRAdmin GUI when you want to perform a remote installation or a local distributed installation of DX NetOps Spectrum. You can also install SRAdmin from the SRAdmin GUI as an alternative to using the DX NetOps Spectrum GUI installation to perform a local DX NetOps Spectrum installation.

> **NOTE**
> If you are upgrading DX NetOps Spectrum from a post 9.0 release, you do not need to install SRAdmin. SRAdmin is automatically upgraded.

Follow these steps:

1. Ensure that you have met the installation prerequisites for Linux, or Windows.
2. Ensure that you are logged in as root when installing on Linux (unless you are using a sudoers file for root permissions). Ensure that you are logged in as a user with Administrator rights if you are installing on Windows.
3. Insert the installation media into the appropriate drive.
   The Install dialog appears.
4. Click Install DX NetOps Spectrum Remote Administration.
   The License Agreement dialog appears.
5. Scroll through and read the license agreement, accept the agreement, and click Next.
   The Destination Location dialog appears with the default directory.
6. Click Next if you want to install in the default location or click Choose and select a different directory and then click Next.

   > **NOTE**
   > The default directory for Windows is C:/Program Files/SRAdmin. The default directory for Linux is /sw/SPECTRUM/SRAdmin.

   A dialog appears with a progress bar.

   > **NOTE**
   > On Linux platforms, the following warning could appear before the installer launches, if you launch the installer from a shell. This warning does not cause any problems with your installation and can be disregarded:

   ```
   awk: cmd. line:6: warning: escape sequence `\.' treated as plain `.'
   ```

   The Installation Complete dialog appears once the installation is complete.
7. Click Done to exit.
   SRAdmin Daemon is installed.

Manually Install SRAdmin Daemon on Windows

You can manually install SRAdmin as an alternative to installing SRAdmin from the DX NetOps Spectrum GUI or the SRAdmin GUI.

> **NOTE**
> Run the visual studio runtime installation before you net start sradmin on Windows. From the command prompt, go to <*spectrum cd directory*>/nt/nttools/VS2012 and run vcredist_x86.exe.

Follow these steps:

1. Ensure that you have met the installation prerequisites.
2. Insert the DX NetOps Spectrum installation media into the appropriate drive.
3. Log in as Administrator or a user with administrator privileges.

   > **NOTE**
   > If you are running the Cygwin32 bash shell, exit it.
4. Open the command prompt and shift to the Program Files directory.

5. Create the SRAdmin directory by entering:
   ```
   mkdir SRAdmin
   ```
6. Run cd SRAdmin.
7. Copy the DX NetOps Spectrum Remote Administration Daemon from the installation media to the SRAdmin directory by entering:
   ```
   copy <installation_media drive>\sdic\windows\sradmin.exe
   ```
8. Install the DX NetOps Spectrum Remote Administration Daemon by entering:
   ```
   sradmin.exe --install
   ```
9. Start the DX NetOps Spectrum Remote Administration Daemon by entering:
   ```
   sradmin.exe --start
   ```
   SRAdmin Daemon is installed.

Manually Install SRAdmin Daemon on Linux

You can manually install SRAdmin as an alternative to installing SRAdmin from the DX NetOps Spectrum GUI or the SRAdmin GUI.

Follow these steps:

1. Ensure that you have met the installation prerequisites.
2. Insert the installation media into the appropriate drive.
3. Log in as root and create the SRAdmin directory path by entering:
   ```
   mkdir -p /sw/SPECTRUM/SRAdmin
   ```
   This directory path is stored in the S99sradmin file in the /etc/rc2.d/ directory.
4. Copy the SRAdmin Daemon to the SRAdmin directory by entering:
   ```
   cp <installation_media drive>/sdic/linux/sradmin.exe /sw/SPECTRUM/SRAdmin
   ```
5. Copy sradmin.rc2 to the initialization directory by entering:
   In 10.4.2, run the follwoing command:
   ```
   cp <installation_media drive>/sdic/linux/sradmin.rc2 /etc/init.d/sradmin
   ```
   In 10.4.2.1, run the follwoing command:
   ```
   cp <installation_media drive>/sdic/linux/sradmin.rc2 /etc/systemd/system/sradmin.service
   ```
6. Change the file permissions by entering:
   In 10.4.2, run the follwoing command:
   ```
   chmod 500 /etc/init.d/sradmin
   ```
   In 10.4.2.1, run the follwoing command:
   ```
   chmod 500 /etc/systemd/system/sradmin.service
   ```
7. Run the following command:
   ```
   /sbin/chkconfig --add sradmin.service
   ```
8. Start the DX NetOps Spectrum Remote Administration Daemon by entering:
   In 10.4.2
   ```
   /etc/init.d/sradmin start
   ```
   In 10.4.2.1
   ```
   systemctl start sradmin
   ```
   SRAdmin Daemon is installed.

Install SRAdmin Daemon in Silent Mode on Windows

As an alternative, you can install SRAdmin Daemon on Windows using silent mode.

NOTE
By default, a silent installation of SRAdmin Daemon is installed into the /sw/SPECTRUM/SRAdmin/directory. To install SRAdmin Daemon into another directory, run the following command before completing the silent installation procedure:

```
srainstall.bin -f <properties file>
```

NOTE
The properties file now contains the following text:

```
INSTALLER_UI=silent
USER_INSTALL_DIR=/sradmin
```

Follow these steps:

1. Ensure that you have met the installation prerequisites.
2. Insert the installation media into the appropriate drive.
3. Log in as Administrator or as a user with administrator privileges.
4. Open the command prompt and go to the appropriate drive.
5. Run the following command:
   ```
   sdic\nt\ srainstall.exe -i silent
   ```
   SRAdmin Daemon is silently installed.

Install SRAdmin Daemon in Silent Mode on Linux

As an alternative, you can install SRAdmin Daemon on Linux using silent mode.

NOTE
By default, a silent installation of SRAdmin Daemon is installed into the /sw/SPECTRUM/SRAdmin/directory. To install SRAdmin Daemon into another directory, run the following command before performing the following procedure:

```
srainstall.bin -f <properties file>
```

NOTE
The properties file now contains the following text:

```
INSTALLER_UI=silent
USER_INSTALL_DIR=/sradmin
```

Follow these steps:

1. Ensure that you have met the installation prerequisites.
2. Insert the installation media into the appropriate drive.
3. Log in as root and navigate to the following directory path:
   ```
   <installation_media drive>/sdic/linux
   ```
4. Run the following command:
   ```
   srainstall.bin -i silent
   ```

   NOTE
   On Linux platforms, the following warning can appear before the installer launches. This warning does not cause problems with your installation and can be disregarded:

   ```
   awk: cmd. line:6: warning: escape sequence `\.' treated as plain `.'
   ```

   SRAdmin Daemon is silently installed.

Install DX NetOps Spectrum in Silent Mode

Create the Host Installation Information File

The Distributed Installer (distinst.exe) uses the information in the Host Installation Information (HII) file to complete the distributed installation.

Follow these steps:

1. Create a text file using a text editor, for host installation information. Alternatively, you can use the hostargs.<*time*> file located in the <*$SPECROOT*>Install-Tools/LOGS/<*version_date*> directory as a starting point.

   NOTE
   The hostargs.<*time*> file does not exist for a new installation.
2. Enter the HII file parameters for each computer on which you plan to install DX NetOps Spectrum.
3. Save the file with a valid filename in a directory, for example, tmp. As long as it is valid, the HII filename is not important.

   NOTE
   You need this file name when you run the distributed installation client.
4. Exit the text editor.
   The HII file is created.

HII File Parameters

The following table describes the parameters in the Host Installation Information File:

Parameter	Description
remote_host=<*remote host to install on*>	The hostname of the target system for the installation. **Note:** Do not enter an IP address OR localhost in place of a hostname.
l_handle=<*landscape handle*>	The landscape handle of the remote system. Required only for SpectroSERVER installations.
install_dir=<*path*>	The directory where DX NetOps Spectrum is installed. For example, /usr/Spectrum or C:/win32app/Spectrum.
install_owner=<*username*>	The owner of the installation.
main_loc_serv=<*location server*>	The hostname of the Main Location Server (required for all-non patch installations). You can only specify a remote hostname if you are installing a SpectroSERVER. **Note:** Do not enter an IP address in place of a hostname.
vcd_path=<*vcd path*>	The path of the installation information. For example, if the installation files are in the local directory, /tmp/SpectrumInstallMedia, enter vcd_path=/tmp/SpectrumInstallMedia
ss_install=yes|no	(Optional) **Default** = Yes for a new installation. Select No if you do not want to install the SpectroSERVER on the remote computer. For first-time installations only.
huge_landscape_handle =yes|no	**Default** = No; "yes" for huge landscape handle type ; "no" for legacy landscape handle type Considered if ss_install=yes; For first-time installations only. For more information, see Landscape Handles.
oc_install=yes|no	(Optional) **Default** = No. Select No if you do not want to install OneClick on the remote computer. For first-time installations only.

xtn_install=yes|no	(Optional) **Default** = Yes for a new installation. Select No if you do not want to install components marked as XTN. For first-time installations only. **Note:** If you set the ss_install parameter to Yes, set the xtn_install parameter to Yes. Do not set xtn_install to Yes when the components marked as XTN are not installed with either OneClick or the SpectroSERVER.
install_type=full|minimal	**Default** = Full. Full indicates standard installation type (all DX NetOps Spectrum components). Select minimal for Remote Operations Server. Required for new installations only.
patch=yes|no	(Optional) **Default** = No. Select Yes for patch installations. **Note:** This parameter is applicable only when upgrading to Service Pack/Hot Fix releases on top of a base release version (for e.g. upgrading to 10.2.1 from 10.2), and does not apply to PTF (Patch Temporary Fix) files or in migration scenarios (for e.g. Migrating from 9.x to 10.x).
same=yes|no	(Optional) **Default** = No. Select Yes to re-install files that have the same version as the presently installed version.
overwrite=yes|no	(Optional) **Default** = No. Select Yes to overwrite all files when selecting same=yes and to avoid the process of comparing installed files with the files to be installed. Files are not preserved.
locale=<*value*>	(Upgrades and Migrations only) Specifies the language to install (evformat/ pcause/eventtables). Do not use this parameter when you are upgrading from 9.3 or 9.3 H01. **Values**: en_US = English; ja_JP = Japanese; zh_CN = Simplified Chinese; zh_TW = Traditional Chinese.
exclude_parts=<*PART-NUMBE*R>; <*PART-NUMBER*>	Excludes components from installation. This list is saved for future upgrades/ patches. For example: exclude_parts=SA-RPT-MGR excludes DX NetOps Spectrum Report Manager from a OneClick distributed installation. exclude_parts=SA-CFMGR;SA-SPM excludes NCM and SPM from a OneClick distributed installation.
ignore_disk_space=yes|no	(Optional) **Default** = No. Select Yes if you want to install, regardless of the disk space warnings.
remove_vnmdb_lock=yes|no	**Default** = No. Removes the vnmdb lock file if one exists. Enter Yes only if SpectroSERVER is not running.
srm_source_host=<*hostname*>	(Optional) Report Manager option (default = no migration). The MySQL hostname needed to obtain the DX NetOps Spectrum Report Manager database.
	(Optional) Report Manager option (default = no migration). The MySQL password is required for the DX NetOps Spectrum Report Manager database.
srm_ss_servers=<*server lists*> (separated by ";")	(Optional) Report Manager option (default = Main Location Server). The SpectroSERVERs from which Report Manager gathers information.
tomcat_port=xxxxxx	The port number for Apache Tomcat. **Default**: 80 for Windows; 8080 for Linux.
tomcat_root=<*tomcat root directory*>	An existing Apache Tomcat directory. The default is the OneClick install directory.

server_username=<*user name*>	(Optional) Used by the Process Daemon (processd) server on Windows only. For a Windows domain, the syntax is <*domain*>\<*username*>. Default = SRAdmin username.
server_password=<*password*>	(Optional) Used by the processd server on Windows only. Default = SRAdmin password.

Creating the Password File

A password file contains accounts and passwords for remote computers. You can create a password file on Linux, and Windows. You add one entry per host to this file. Each line contains:

- host name
- root/administrator account name
- root/administrator account password

You can use a pound (#) or a backslash (\) in the password file. Insert a backslash before the characters of the password or DX NetOps Spectrum interprets them as a comment line.

For example, if your password is test#computer, enter it in the password file as test\#computer. If your password is test \computer, enter it in the password file as test\\computer.

NOTE
You can use a sudoers file to provide users with limited root permissions for remote clients. Root permissions apply only to the commands required to install DX NetOps Spectrum. This option is available for Linux operating systems.

Create a Password File

You can omit a root/administrator password in the password file and only enter a host name and user name. In this case, the DX NetOps Spectrum distributed installation client prompts you to enter a password at the command line.

After you enter a password, the installer asks if you want to use this same password for all entries. If you answer “No," you are prompted for a password each time a host in the password file does not have a password entry.

Follow these steps:

1. Create a password file using a text editor.
2. For each system in which you plan to install DX NetOps Spectrum, add an entry with the host name, account name, and password. Enter this information in the following order:
   – On Linux:
   ```
   <host name> <root account name> <root password>
   ```
   – On Windows:
   ```
   <host name> <administrator account name> <administrator password>
   ```
   NOTE
   The <*root password*> and the <*administrator password*> are optional.
3. Save the file with a valid file name in a directory. If the password file name is valid, it is not important.

   NOTE
   Use this file name for running the distributed installation client.
4. Exit the text editor.
   The password file is created.

Grant Limited Root Permissions (Linux)

Sudo (super user do) is a third-party application. Using this application, a system administrator can let users run certain commands as root or as another user. DX NetOps Spectrum is compatible with the sudoers file (which the Sudo

application uses). Specifically, you can use the sudoers files to grant users root permissions that are needed for running the DX NetOps Spectrum installation on remote systems. This file eliminates the need for the installation program to have full root permissions on all of the remote systems where DX NetOps Spectrum is installed.

> **NOTE**
> DX NetOps Spectrum does not use the actual Sudo application to change user permissions. Instead, it parses the applicable information in the sudoers file to provide installation permissions to the specified user. For information about the Sudo application, see http://www.courtesan.com/sudo/.

SRAdmin Daemon must be installed on all the computers where you plan to install DX NetOps Spectrum. You also need a sudoers file on all the computers where you plan to install DX NetOps Spectrum.

Execute the following steps on each remote computer where you are installing DX NetOps Spectrum.

Follow these steps:

1. Add the following entry to the sudoers file. This entry provides the specified user permission to run the sradmin.exe program as root:
   ```
   <username> <client_host> = <path_to_sraadmin>/sradmin.exe
   ```
   - ***username***
     Specifies the user with root permissions for running the installation. You can set this parameter to ALL to indicate that all users can have root permissions.
   - ***client_host***
     Specifies the name of the local host system (that is, the system where you plan to run the distributed installation). You can set this parameter to ALL to indicate all host computers that exist in the NIS/DNS namespace.
   - ***path_to_sradmin***
     Specifies the path to the sradmin.exe application. The default path is /sw/SPECTRUM/SRAdmin/. You can also use ALL in place of <*path_to_sraadmin*>/sradmin.exe, which indicates that the user has root access to all programs on the specified server.

     > **NOTE**
     > The entry must be on a single line. Do not use line continuation characters.
2. Create a symbolic link file named sudoers in the directory where the sradmin.exe application exists. By default, this directory is /sw/SPECTRUM/SRAdmin. You can use the following command to create the symbolic link file:
   ```
   ln -s <full path to sudoers file from step 1> sudoers
   ```
3. Verify that the following conditions are met:
   - Root(0) owns both of these files.
   - The group is set to root(0).
   - The permissions for the files are 0440.

   Limited root permissions are granted.

Change the Sudoers File Owner (Linux)

By default, root owns the sudoers file. However, to limit the number of users who can access the sudoers file, you can change its owner. Then, modify the sradmin.exe startup parameters so that the sradmin.exe application honors only the configuration found in the sudoers file that the specified user owns. Sudoers files that other users own are ignored.

To change the sudoers file owner, add the --sudoowners parameter to the command line in the S99sradmin file that is used for starting sradmin.exe.

Follow these steps:

1. Open the following file:
   - On Linux: /etc/rc2.d/K09sradmin
2. Locate the following line:
   ```
   $SRADHOME/sradmin.exe --start
   ```

3. Add the following parameter to this line:
   ```
   --sudoowners=<username>
   ```
   – ***username***
   Specifies the user who owns the sudoers file. For example, you can enter:
   ```
   $SRADHOME/sradmin.exe --start --sudoowners=bsmith
   ```
4. Save and close the file.
   The sudoers file owner is changed.

Run the Distributed Installation Client on Windows

The prerequisites for running the distributed installation client on a Windows system are as follows:

- Verify that the time settings for the following systems are synchronized within 2 minutes of each other:
  – The Windows system running the installation
  – The remote hosts receiving the installation.
  If the time setting is not synchronized, the distributed installation fails to authenticate with the SRAdmin Daemon on the remote systems.
- Verify that all DX NetOps Spectrum processes in the distributed environment (including the SpectroSERVER and the OneClick clients) are shut down.
- Verify name resolution.

Follow these steps:

1. Log on to the Windows system.
2. Install SRAdmin Daemon.
3. Create the password file.
4. Create the Host Installation Information file.
5. Locate the distributed installation client (sdicnt.exe):
   – If you are installing from the installation media, the executable is located in <*installation_media drive*>:\sdic directory.
   – If you are installing from a downloaded patch, the executable is located in the <*$SPECROOT*>/Install-Tools/sdic directory.
6. (Optional) Run a verification test before running the DX NetOps Spectrum Distributed Installation Client. This test verifies user names and passwords in the password file, checks SRAdmin versions on remote computers, and validates VCD paths. To run this test, enter the following command:
   ```
   <pathtoexecutable>\sdicnt.exe -h <host file> -p <password file> -test
   ```
   – ***pathtoexecutable***
   Specifies the location of the distributed installation client.
   – ***host file***
   Specifies a file containing the remote host installation information. Include this path when the host file is not located in the same directory as the distributed installation client.
   Example: -h C:/tmp/hostinfo
   – ***password file***
   Specifies the file containing account and password information. Include this path when the password file is not located in the same directory as the distributed installation client.

   NOTE
   Results of the test appear on the screen and in the LOGS_*YYYYMMDD* subdirectory (*YYYY*=year, *MM*=month, *DD*=day). This subdirectory is located in the same directory as the DX NetOps Spectrum Distributed Installation Client.
7. Run the DX NetOps Spectrum Distributed Installation Client as follows:
   ```
   <pathtoexecutable>\sdicnt.exe -h <host file> -p <password file>[-accept]
   ```

- ***pathtoexecutable***
  Specifies the location of the DX NetOps Spectrum Distributed Installation Client (sdicnt.exe).
- ***host file***
  Specifies a file containing the remote host installation information. Include this path when the host file is not located in the same directory as the distributed installation client.
  Example: -h C:/tmp/hostinfo
- ***password file***
  Specifies the file containing account and password information. Include this path when the password file is not located in the same directory as the distributed installation client.
- **-accept**
  (Optional) Acknowledges the license agreement and accepts its terms without the agreement appearing on your screen. This option allows for a silent installation.
  The license agreement is located in the following places:
  - In the installation directory, <*install dir*>/Install-Tools/license/license.txt.
  - On the installation media, <*installation_media*>/<*plat*>/license/license.txt, where *plat* is either nt, sunos, or linux.

    NOTE
    You do not need to perform the following two steps when you use the -accept command.

You can exit the installation at any time by pressing Ctrl + C; however, the remote installations continue.

NOTE
Results of the installation appear on the screen and in the LOGS_*YYYYMMDD* subdirectory. This subdirectory is located in the directory where you ran the distributed installation client.

8. Review the license agreement. When complete, accept the terms of the agreement and continue the installation by entering Y (yes), and pressing Enter.

   NOTE
   Pressing Enter scrolls line-by-line, and pressing the space bar scrolls page-by-page.

   After the installation is complete, the message Installation Complete appears. Running the distributed installation client on Windows is complete.

Run the Distributed Installation Client on Linux

The prerequisites for running the distributed installation client on a Linux system are as follows:

- Verify that the time settings for the following systems are synchronized within 2 minutes of each other:
  - The Linux system running the installation
  - The remote hosts receiving the installation

  If the time setting is not synchronized, the distributed installation fails to authenticate with the SRAdmin Daemon on the remote systems.
- Verify that all DX NetOps Spectrum processes in the distributed environment (including the SpectroSERVER and the OneClick clients) are shut down.
- Verify name resolution.

Follow these steps:

1. Log on to the Linux system.
2. Install SRAdmin Daemon.
3. Create the Host Installation Information file.
4. Create the password file.
5. Locate the distributed installation client (sdiclinux.exe for Linux):

- If you are installing from the installation media, the executable is located in the *<installation_media mount>*/sdic directory.
- If you are installing from a downloaded patch, the executable is located in the *<$SPECROOT>*/Install-Tools/sdic directory.

6. (Optional) Run a verification test before running the distributed installation client. This test verifies user names and passwords in the password file, checks SRAadmin versions on remote computers, and validates VCD paths. To run this test, enter the following command:
   - For Linux:

     ```
     <pathtoexecutable>/sdiclinux.exe -h <host file> -p <password file> -test
     ```

     NOTE
     Results of the test appear on the screen and in the LOGS_*YYYYMMDD* subdirectory (*YYYY*=year, *MM*=month, *DD*=day). This subdirectory is located in the same directory as the distributed installation client.
7. Run the distributed installation client as follows:
   - For Linux:

     ```
     <pathtoexecutable>/sdiclinux.exe -h <host file> -p <password file>
     [-accept]
     ```

   You can exit the installation at any time by pressing Ctrl + C; however, the remote installations continue.

   NOTE
   Results of the installation appear on the screen and in the LOGS_*YYYYMMDD* subdirectory. This subdirectory is located in the same directory as the distributed installation client.
8. Review the license agreement. When complete, accept the terms of the agreement and continue the installation by entering Y (yes), and pressing Enter.

   NOTE
   Pressing Enter scrolls line-by-line, and pressing the space bar scrolls page-by-page.

   After the installation is complete, the message Installation Complete appears. Running the distributed installation client on Linux is complete.

Distributed Installations Without the Root Password

Under certain conditions, you can run a distributed installation in a Linux environment without being prompted for a user name and password.

For this scenario to work properly, run the distributed installation from the local machine as root. The installation program automatically executes without asking for a user name or password.

NOTE
This process does not automate the acceptance of the DX NetOps Spectrum license agreement. You must manually agree to the terms of the license agreement before the installation can proceed.

Log Files

The DX NetOps Spectrum Distributed Installation Client creates a subdirectory named LOGS_*YYYYMMDD* (*YYYY*=year, *MM*=month, and *DD*=day when the installation was started). This subdirectory contains a file for each system where you install DX NetOps Spectrum. To view these files, you need write permissions to the directory where you started the distributed installation client.

These files use the following naming convention:

```
<host_name>.HH.MM
```

host_name

Specifies the remote host name.

- ***HH***
  Specifies the hour when the installation started.
- ***MM***
  Specifies the minute when the installation started.

  NOTE
  Results of a distributed installation appear in the LOGS_*YYYYMMDD* subdirectory.

Installation Duplication

After you complete a DX NetOps Spectrum GUI installation, you can use the hostargs.<*time*> file located in the <*$SPECROOT*>Install-Tools/LOGS/<*version_date*> directory of the new installation as a baseline for more installations. The only potential change that is needed is the remote_host parameter in the hostargs<*time*> file.

Also, add values to the server_user name and server_password parameters for either of these scenarios:

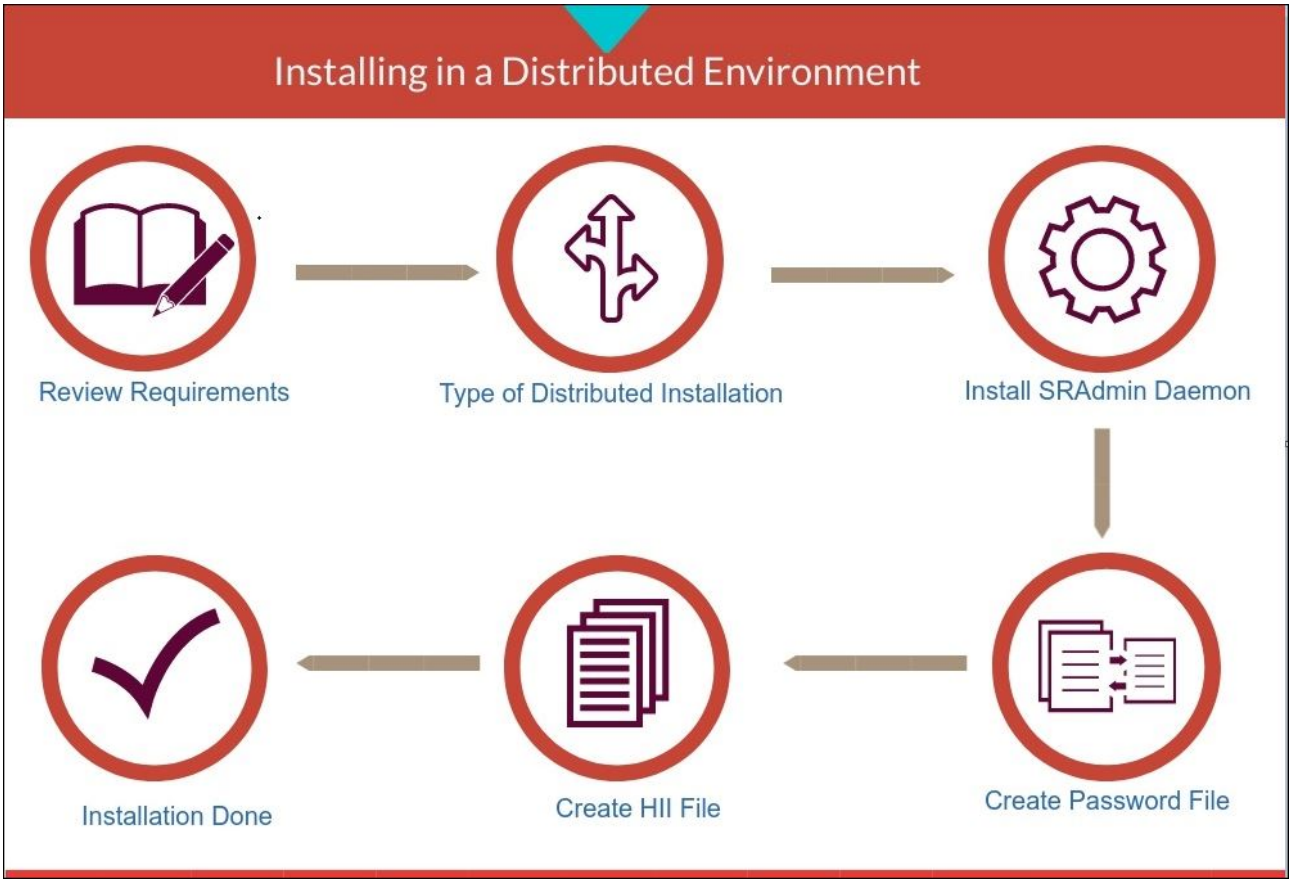
- You are installing on Windows in a domain
- You do not want the existing user name and password used in the <*password file*>.

If you are installing DX NetOps Spectrum on Windows in a domain, create a password file.

Installing DX NetOps Spectrum in a Distributed Environment

Process Overview

The following diagram shows the overall process:



Distributed Installation Requirements

Ensure that you meet the following requirements before you start a distributed installation:

- Check the Release Information about any new parameters that are required for the Host Installation Information file.
- Establish a TCP/IP connection from the target system to remote systems.

  NOTE
  In firewall environments, ensure that port 46517 is opened during a distributed installation.
- Verify that the time setting on the installer host is synchronized to within 2 minutes of the time setting on the remote hosts. This setting is required for secure authentication with the DX NetOps Spectrum Remote Administration Daemon.
- Verify that the remote hosts have sufficient disk space. The DX NetOps Spectrum distributed installation copies temporary files to temporary directories on the target system. Therefore, it requires at least 100 MB of disk space in the temp or tmp directories. For Windows, you can edit the default TEMP location as a user Environment Variable.

Types of Distributed Installations

DX NetOps Spectrum lets you select the type of distributed installation that meets your requirements.

The following table shows the available types of distributed installations and their corresponding procedures found in this section:

Type of Distributed Installation	Procedures
DX NetOps Spectrum installation on Linux	Install SRAdmin Daemon. Create a Root/Administrator Password File and create the Host Installation Information File. Run the Distributed Installation Client on Linux.
Upgrading DX NetOps Spectrum from 9.3 or later on Linux	Install SRAdmin Daemon. Create a Root/Administrator Password File and create the Host Installation Information File. Run the Distributed Installation Client on Linux.
DX NetOps Spectrum installation on Windows	Install SRAdmin Daemon. Create a Root/Administrator Password File and create the Host Installation Information File. Run the Distributed Installation Client on Windows.
Upgrading DX NetOps Spectrum from 9.3 or later on supported Windows platforms	Install SRAdmin Daemon. Create a Root/Administrator Password File and create the Host Installation Information File. Run the Distributed Installation Client on Windows.

How to Perform a Distributed Installation

The DX NetOps Spectrum distributed installation is a command-line interface that lets you install DX NetOps Spectrum and the OneClick web server (locally and remotely). Different types of installations can be performed on each system. For example, you can install the following items in a single distributed installation:

- SpectroSERVER only
- OneClick web server only
- SpectroSERVER and OneClick web server

A DX NetOps Spectrum distributed installation has the following components:

- **DX NetOps Spectrum Remote Administration (SRAdmin) Daemon**
  Allows for secure, remote installations without requiring that you manually set up the NFS mounts or Microsoft Network File shares. A time-critical Triple-DES encryption is used to ensure that the root/Administrator account/password information is safe when it is passed between daemons. Install SRAdmin on each remote system where you install DX NetOps Spectrum.
- **DX NetOps Spectrum Distributed Installation Client (sdicsol.exe for Linux, and sdicnt.exe for Windows)**
  Launches multiple installations across multiple machines and collects the results of these installations. The distributed installation client requires the following files:
  - **Password file**
    Contains accounts and passwords for remote computers.
  - **Host installation file**
    Contains installation information. The topic HII File Parameters contains important information about the required contents of this file.

To perform a distributed installation, complete these steps

1. Install SRAdmin Daemon.
2. Create the password file
3. Create the HII file.

Installing DX NetOps Spectrum Remotely

Installing DX NetOps Spectrum Remotely

You can use Telnet to install DX NetOps Spectrum remotely over your network. You can also use the GUI or distributed installation to install DX NetOps Spectrum remotely. For example, you can use the Windows GUI installer to install DX NetOps Spectrum to a Linux system using the installer for that platform.

You can only perform one system installation at a time. You need the host name of the remote system and the administrator ID and password.

Follow these steps:

1. Install the DX NetOps Spectrum Remote Administration Daemon (SRAdmin) on the computer where you want to install DX NetOps Spectrum remotely.
2. See Install DX NetOps Spectrum to perform the installation.
   DX NetOps Spectrum is installed remotely.

   WARNING

   It is mandatory to reboot the server after you complete a fresh SpectroSERVER installation while installing 10.2.

OneClick Web Server and SpectroSERVER on Separate Systems

To install the OneClick web server and a SpectroSERVER on separate systems, repeat the installation process for each system installation. Be sure to select the appropriate components on the Select Options dialog during installation. You can also use the distributed installation to install different components to separate systems at the same time.

Mount the Installation Media on Linux

If Volume Management is disabled, set up an installation media mount-point directory. Then, run a mount command to access the installation software on the installation media. This procedure varies depending on whether the installation is local (the target system is the host for the drive) or remote (a system other than the target hosts the drive).

If a Linux system has Volume Management enabled, the installation media mounts automatically.

NOTE
Use these steps to mount the CABI r4.1 SP3 installation media.

Follow these steps:

1. Insert the installation media into the appropriate drive.
2. Mount the <*installation_media*> file system by running the following command, where <*installation_media*> is the directory you created:
   – On Linux:

   ```
   mount -t iso9660 /dev/dvd /mnt/<installation_media>
   ```
3. The installation media is mounted.

Start the Installation on Windows

You can start the DX NetOps Spectrum installation on Windows platforms.

WARNING
You cannot install a released version of DX NetOps Spectrum on top of a beta or evaluation version of the product. Instead, uninstall the beta or evaluation version first.

Follow these steps:

1. Stop all non-DX NetOps Spectrum running applications.
2. Stop the following DX NetOps Spectrum applications:
   - Shut down all OneClick clients by logging off all users from OneClick in the Client Details web page in the OneClick home page.

     NOTE
     For more information about shutting down OneClick clients, see OneClick Administration.
   - Stop the SpectroSERVER and the Archive Manager by clicking Stop SpectroSERVER in the DX NetOps Spectrum Control Panel and then close the DX NetOps Spectrum Control Panel. Or you can stop the SpectroSERVER and Archive Manager from the command line by running the "<$SPECROOT>/bin/stopSS.pl" as Spectrum Owner at the command prompt.

     NOTE
     For more information about the DX NetOps Spectrum Control Panel, see OneClick Administration.
   - Stop all VnmSh connections.

     NOTE
     For more information about stopping VnmSh connections, see the Command Line Interface User.
   - Close all Bash shells.
3. Ensure that you have met the Installing and Upgrading.

   WARNING
   Disable your antivirus software's real-time protection before installing DX NetOps Spectrum. Disabling helps avoid potential problems with files that could be in use by the real-time protection software.
4. Log in as a user with administrator rights.
5. Insert the installation media into the appropriate drive. If autorun is disabled, you can double-click setupnt.exe from the Explorer view to start the installation.

The installation starts.

Start the Installation on Linux

You can install DX NetOps Spectrum on Linux platforms.

WARNING
You cannot install a released version of DX NetOps Spectrum on top of a beta or evaluation version of the product. Instead, uninstall the beta or evaluation version first.

Follow these steps:

1. Stop all non-DX NetOps Spectrum running applications.
2. Stop the following DX NetOps Spectrum applications:
   - Shut down all OneClick clients by logging off all users from OneClick in the Client Details web page in the OneClick home page.

     NOTE
     For more information about shutting down OneClick clients, see OneClick Administrator.
   - Stop the SpectroSERVER and the Archive Manager by clicking Stop SpectroSERVER in the DX NetOps Spectrum Control Panel and then close the DX NetOps Spectrum Control Panel. Or you can stop the SpectroSERVER and Archive Manager from the command line by running the "<$SPECROOT>/bin/stopSS.pl" as Spectrum Owner at the command prompt.

     NOTE
     For more information about the DX NetOps Spectrum Control Panel, see the OneClick Administration.
   - Stop all VnmSh connections.

NOTE
For more information about stopping VnmSh connections, see the Command Line Interface User space.

- Close all Bash shells.

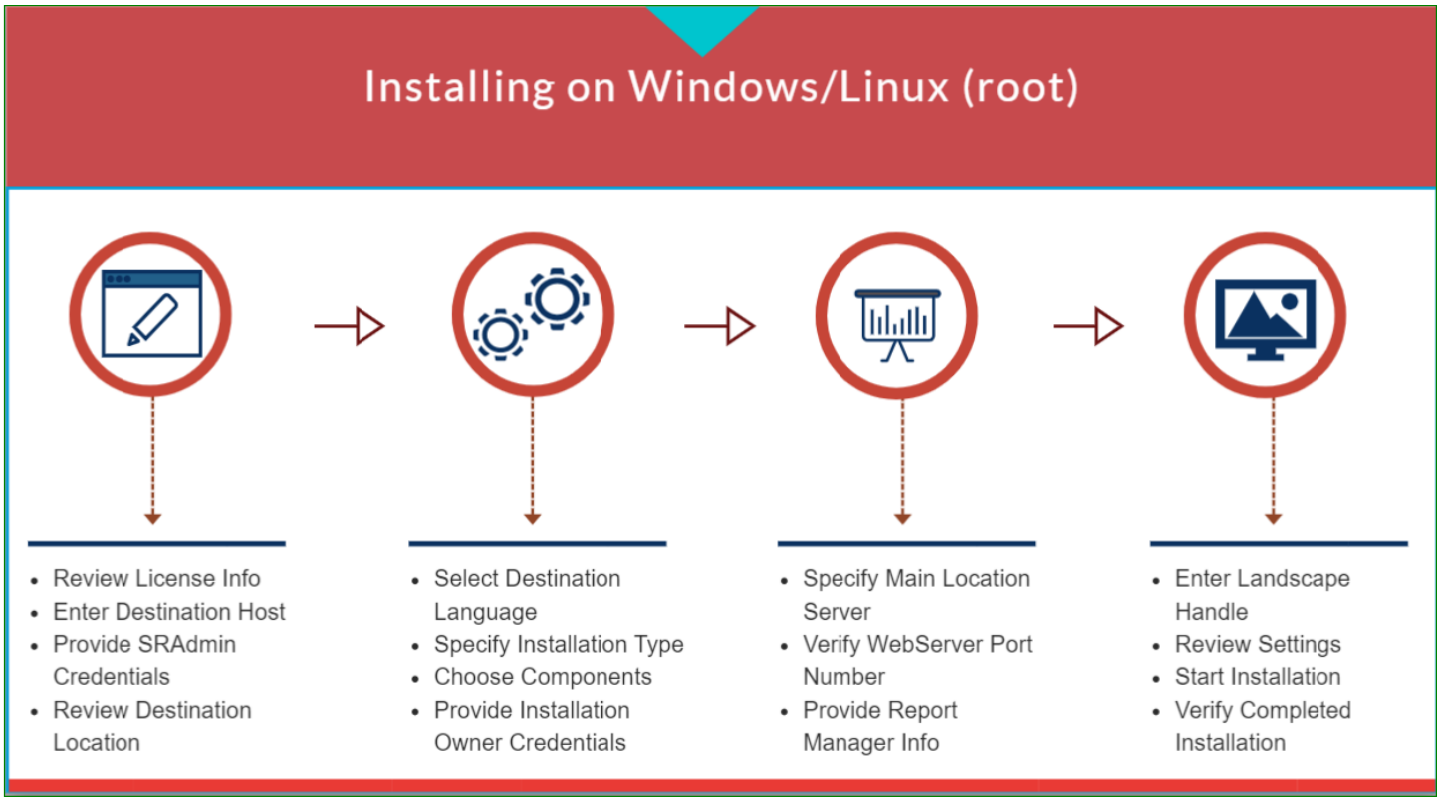
3. Ensure that you have met the system requirements for Linux.
4. Ensure that you have met the prerequisites.
5. Download the DX NetOps Spectrum TAR package for Linux into the appropriate drive.
6. Extract the DX NetOps Spectrum TAR package for Linux by using the "gunzip" command, and then the "tar - xvf" command at the command prompt.
7. If necessary, set your DISPLAY variable to the hostname of the target system:
   - From a C shell, enter:
     ```
     setenv DISPLAY <hostname>:0
     ```
   - From Bourne or Korn shells, enter:
     ```
     DISPLAY=<hostname>:0 ; export DISPLAY
     ```
     NOTE
     Run echo $DISPLAY to confirm this setting is in effect.
8. To display the DX NetOps Spectrum installation GUI on a remote system, run the following command from the target system:
   ```
   /usr/openwin/bin/xhost +<hostname>
   ```
   - ***hostname***
     Is the name of the target system.
9. Perform one of the following steps:
   - For Linux, navigate to the extracted TAR folder, and run the setuplin.exe executable file.
10. Double-click the Installer icon

Installing DX NetOps Spectrum on Windows and Linux (root User)

You can install DX NetOps Spectrum on Windows and Linux (root user).

NOTE
For more information about the prerequisites, see the prerequisites section.

The following diagram shows a summarized view of the installation steps:



WARNING
The C:\Program Files\CA directory on Windows platforms and the /opt/CA directory on Linux platforms are automatically created during a first-time installation of DX NetOps Spectrum. DX NetOps Spectrum components that are also common to other CA products are intentionally installed into this directory. This directory is automatically updated as needed during a DX NetOps Spectrum upgrade. Do not remove files from this directory.

Follow these steps:

1. Start the installation on Windows or Linux.
   The **Install** dialog opens.
2. Select the **Install DX NetOps Spectrum** option.

   NOTE

   On Linux platforms, the following warning could appear before the installer launches if you launch the installer from a shell.

   This warning does not cause any problems with your installation and can be disregarded.

   ```
   awk: cmd. line:6: warning: escape sequence `\.' treated as plain `.'
   ```

   The **Introduction** dialog opens.
3. Click **Next** to proceed.
   The **License Agreement** dialog opens.
4. Scroll through and read the license agreement, accept the agreement, and click **Next**.
   The **Destination Host** dialog opens.
5. Enter the name of the host system where you are installing DX NetOps Spectrum and click **Next**.

NOTE
If you are installing DX NetOps Spectrum and OneClick on remote platforms on your network, follow the steps in Installing DX NetOps Spectrum Remotely.

The **SRAdmin Authentication** dialog opens.

NOTE
If the 'Unable to connect to DX NetOps Spectrum Remote Administration Daemon (SRAdmin)' dialog appears, install SRAdmin before continuing with the installation. To install SRAdmin, click Install on this dialog.

6. Enter a **username** and **password** as follows, and click **Next**:
   - For a **Windows** installation, enter a username that has administrator rights and verify the domain name (if applicable).
   - For **Linux** installation, enter a username with root access. Or, you can use a ***sudoers*** file for root permissions.

     NOTE
     If you have root access when starting this installation, you are not prompted for a user name and password. Ensure that usernames do not exceed 16 characters.

   The **Destination Location** dialog opens.
7. Click **Next** to install DX NetOps Spectrum in the default directory. The default directory for Windows is C:\win32app \SPECTRUM .The default directory for Linux is /usr/SPECTRUM
   - If you are performing an in-place upgrade, the installation program detects the previous installation directory.

     WARNING
     When performing an in-place upgrade, do not change the default destination to a location other than the directory containing the DX NetOps Spectrum database you are upgrading. On Linux, do not use / opt/SPECTRUM as an installation directory. This location and name are reserved for a directory that is created automatically during the installation.
   - To install DX NetOps Spectrum in a location other than the default directory, click **Choose**, select a location, and click **Next**. This option only appears for a local installation (not for a remote installation).

     WARNING

     Ensure there are no spaces in the path name. You cannot install DX NetOps Spectruminto a directory that contains a space anywhere in the path. Spaces within the directory path cause the installation to fail.

   The installer reports, it is extracting installation information. Then, the **Select Destination Language** dialog opens.
8. Select the language in which you want to install, and click **Next**.
   Localized CsEvFormat, CsPCause and **EventTables** is installed for the selected language.
   The **Select Options** dialog opens.
9. Select either one of the Installation Type:
   - **Standard:** Allows the installation of the SpectroSERVER, the OneClick server, and all other DX NetOps Spectrum components.
   - **Remote Operations Server:** Allows the installation of minimal components to run the SpectroSERVER and OneClick server.

   Components are displayed based on the type of installation you entered.

   NOTE
   If you are performing an upgrade, add-on components that exist in your current implementation appear for the Remote Operations Server option.
10. Select the items that you want to install from the **components list** and click **Next**.

    WARNING
    Installing OneClick on a single-CPU SpectroSERVER host system can degrade the performance of both SpectroSERVER and OneClick. We recommend installing OneClick on a separate dedicated system.

The **Host Evaluation** dialog opens.

11. Scroll down to verify that no warnings appear, and click **Next** to proceed.
    The DX NetOps Spectrum **Installation Owner** dialog opens.
12. Enter the **username** and **password** as follows, and click **Next**. This username is used to create the initial DX NetOps Spectrum user (if installing SpectroSERVER) and becomes the installation owner. For a OneClick installation, the username also determines the SpectroSERVERs to which the OneClick web server connects:
    - For a **Linux** installation, enter the username for the host system. The installation owner must be a non-root user.
    - For a **Windows** installation, enter either the domain user username and password or the local user username and password.

      WARNING
      When installing DX NetOps Spectrum on a computer in a domain, the username for the DX NetOps Spectrum installation owner cannot be the same as the computer hostname.

    NOTE

    The username and password are also used to configure the DX NetOps Spectrum Process Daemon service. The username and password are not used or stored in DX NetOps Spectrum

    NOTE
    If the installation owner is a non-administrator, you cannot restart the process service as the installation owner. However, because you typically do not need to restart the service on a normal daily basis, we recommend that the installation owner is a non-administrator. Using a nonadministrator helps increase security and simplify password maintenance.

    NOTE

    For first-time installations, the default DX NetOps Spectrum password for the installation owner is spectrum.

    WARNING

    When installing OneClick, be sure to specify a DX NetOps Spectrum username to which the administrative license is associated. This user needs access to all models in DX NetOps Spectrum (ADMIN access). We recommend that you specify the installation owner that you specified during the SpectroSERVER installations. This user must also exist on the installation host and does not need to be a Windows administrative user.

    The **Main Location Server** dialog opens.
13. When you install DX NetOps Spectrum components, you also automatically install a location server. However, if you install OneClick only, you do not automatically install a location server.

    NOTE
    In a distributed environment, DX NetOps Spectrum uses location servers to maintain the VNM landscape map and provide connection services to client applications. For more information about location servers and the main location server, see the Distributed SpectroSERVER Administration.

    Enter a **hostname** for the main location server and click **Next**.

    NOTE
    DX NetOps Spectrum must be able to resolve the hostname, regardless of whether you provide a fully qualified hostname.

    The **Web ServerPort Number** dialog shows the default value.
14. **(Optional)** Enter a port number other than the default, and click **Next**.

    NOTE
    The default port is 80 for Windows and 8080 for Linux.

If you have previously selected **Report Manager** from the components list, the **Report Manager Servers** dialog opens.

15. If the **Report Manager Servers** dialog opens, select each SpectroSERVER that you want Report Manager to report about and click **Next**.
    The DX NetOps Spectrum **Report Data Migration Panel** dialog opens.
16. If you are performing a Spectrum Report Manager migration, enter the **source hostname** and **root password** for the report database and then click **Next**. Otherwise, leave the fields blank.
    ```
    bc obase=16 <decimal value> * 262144<CTRL>D
    ```
    The bc utility displays a hexadecimal value that you enter in the Landscape Handle box, prefixed by 0x. For example, a decimal value of 24 multiplied by 262144 yields a hexadecimal value of 600000. You would enter 0x600000 in the Landscape Handle field. Unique landscape handles are crucial if you are configuring a distributed SpectroSERVER environment.
    If you are performing an in-place upgrade the **Review Settings** dialog opens, click **Install** to install **DX NetOps Spectrum.**
17. The **Landscape Handle** dialog opens. Enter a value as instructed on the dialog for the landscape handle. **For more information about Landscape Handles, see** Landscape Handles in Setting up a Distributed SpectroSERVER.

    NOTE

    In a Distributed SpectroSERVER environment, you cannot select a mix of Legacy and Huge landscapes. For example, if you have an existing SpectroSERVER running in Legacy and want to add a new SpectroSERVER, you cannot add a Huge Landscape, even if you are installing it from scratch.

    In a Distributed SpectroSERVER environment, all your SpectroSERVERs should be either Legacy or Huge Landscapes.

    This dialog appears only when you are installing a SpectroSERVER. This dialog does not appear during upgrade or migration scenarios.
    What is a landscape handle?
    A landscape is a network domain that a single SpectroSERVER manages. A landscape includes all the models, associations, attribute values, alarms, events, and statistics of a SpectroSERVER. Each landscape in a network is unique, and a unique landscape handle (ID) identifies each.

    NOTE

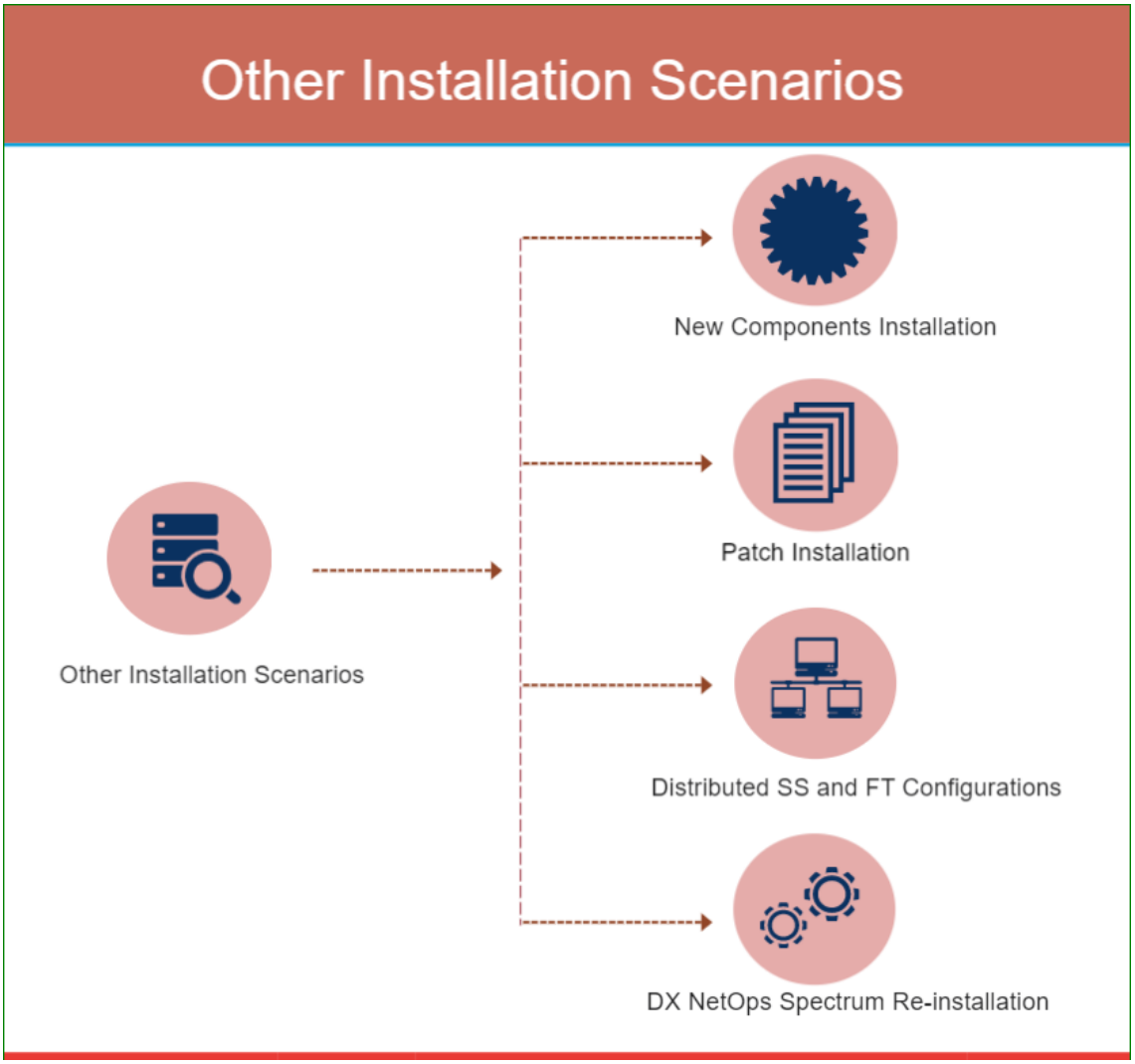
    From 10.3 onwards, you can load the catalog of the Legacy Landscape on a Huge Landscape and vice versa (using SSdbload -c option). However you will not see the landscape details displayed. For more information on the landscape details, see the Known Anomalies section.
18. After you enter a value on the dialog for the landscape handle, Click **Next.**
    The **Review Settings** dialog opens.
19. Scroll down to ensure all the settings are what you had selected and click **Install**.
    The **Installing DX NetOps Spectrum** dialog appears. After DX NetOps Spectrum is installed, the status changes to **Installation Successful** and the **Next** button is enabled. Click **Next**.

    NOTE
    During the installation process, the 'View Logs' button is enabled. Click the button to view the installation logs. The logs are helpful if there are installation failures or errors.
20. The **Installation Complete** dialog opens. Click **Done**.
    The configuration dialog appears for a brief moment and closes. DX NetOps Spectrum is configured for your system.
21. Click **Close** on the initial Install dialog. **Log out,** and **log back in**.
    DX NetOps Spectrum is installed.

    WARNING
    It is mandatory to reboot the server after you complete a fresh SpectroSERVER installation while installing DX NetOps Spectrum.

Other Installation Scenarios

The following diagram shows the additional installation scenarios:



How to Install New Components

If any components were not installed initially, you can add them using the following procedure.

Follow these steps:

1. Stop SpectroSERVER and all DX NetOps Spectrum applications.
2. Run the installation for the version of DX NetOps Spectrum you currently have installed. Note the following guidelines:
   - Retain the default directory on the Destination Location dialog because all components must be installed in the same directory. You cannot install OneClick and SpectroSERVER in different directories.
   - If the component you are adding is listed on the Select Options dialog, select it. Components from the same version that are already installed appear selected and disabled; these components are not reinstalled. If the component you are adding does not appear, verify that the component is installed in the Review Settings dialog. The Review Settings dialog displays all components that are installed.

NOTE
You cannot change the Installation Type when adding components. This option is available on the initial installation only.

The DX NetOps Spectrum installation installs the new components only.

3. After all of the components are installed, reinstall the latest DX NetOps Spectrum maintenance, if any.

Patch Installations

Updates or patches for existing versions of DX NetOps Spectrum are available for downloading at http://broadcom.com/support. Contact a technical support representative for available maintenance patches. Each patch includes a software release notice that provides step-by-step installation instructions.

Distributed SpectroSERVER and Fault-Tolerant Configurations

To install more than one SpectroSERVER to manage different portions of your network, see the Distributed Administrator section before starting the installation.

DX NetOps Spectrum also supports a fault-tolerant configuration so that one or more than one SpectroSERVER can function as standbys for a primary SpectroSERVER. In this scenario, a secondary SpectroSERVER is ready to take over management functions when the primary SpectroSERVER becomes unavailable. The special requirements for this configuration are explained in the Distributed Administration.

Upgrades with fault-tolerance are supported. For more information, see Upgrade Best Practices: Fault-Tolerant Deployments.

NOTE
For information about OneClick Web server fault tolerance, see OneClick Administration.

Reinstall DX NetOps Spectrum

If problems occur during installation, you can reinstall DX NetOps Spectrum. You cannot install new components and reinstall at the same time. Reinstall DX NetOps Spectrum first and then install the new components.

NOTE
If you want to change the installation owner for an existing installation, run the following program from a Windows bash shell before reinstalling DX NetOps Spectrum. This program removes the processd service so that the service is recreated during the reinstallation with the new installation owner:

```
<install dir>/lib/SDPM/processd.exe --remove
```

To reinstall DX NetOps Spectrum to change the installation owner, you must be reinstalling a full, major release; it cannot be a service pack or maintenance. For example, 9.3.x users must reinstall 9.3.0 first, then install 9.3.x again.

NOTE
The following procedure is for the GUI-based installation. If you are using the distributed installation, set *same=yes* in the host installation information file before reinstalling DX NetOps Spectrum.

Follow these steps:

1. Stop SpectroSERVER and all DX NetOps Spectrum applications.
2. Run the installation for the version of DX NetOps Spectrum you currently have installed. Note the following guidelines:
   - Retain the default path on the Destination Location dialog because all components must be installed in the same directory.
   - In the Select Options dialog, no new selections can be made. Components that can be reinstalled are selected and disabled.
   - In the Host Evaluation dialog, a message indicates that nothing was selected for installation.

3. Click Next on the Host Evaluation dialog.
4. Click Reinstall on the Reinstall Option dialog.
5. Click Preserve on the Preserve Files dialog.
   The existing user-modified files are preserved, and the Host Evaluation runs again to evaluate the new settings.
6. Click Next and modify the installation dialogs, as needed.
7. View the Review Settings dialog and ensure that all components are reinstalled. Click Next to proceed with the reinstallation.
   The reinstallation completes.

OneClick Web Server Upgrades and New OneClick Privileges

OneClick Web Server Upgrades

All OneClick clients must be shut down before upgrading the OneClick web server because Java Web Start applications cache application jar files on the client. These jar files are automatically updated when you restart the application. You can shut down all OneClick clients by selecting Client Details and clicking Log Off Clients.

For OneClick web server installations on a dedicated system (such as, <*OC install dir*>/WebApps), the OneClick web server is installed in the <*$SPECROOT*> directory. However, OneClick web server installation on the same system as DX NetOps Spectrum are installed in a different directory. In this case, the OneClick web server is installed in the directory you specified during the DX NetOps Spectrum upgrade (such as, <*OC install dir*>/WebApps).

NOTE
For more examples about how to use the <*$SPECROOT*>/custom directory, see OneClick Customization .

When you install the OneClick web server on a dedicated system, install it in the <*OC install dir*>/WebApps directory. Install DX NetOps Spectrum in the existing DX NetOps Spectrum directory when the system includes a current version of a SpectroSERVER and the OneClick web server. Do not install DX NetOps Spectrum in the OneClick directory. The existing DX NetOps Spectrum directory appears in the Destination Location dialog.

NOTE
Upgrading the OneClick web server typically archives the existing Apache Tomcat directory to <*$SPECROOT*>/Install-Tools/LOGS/<*version_date*>/SavedFiles/tomcat-<*time*>. Once you have successfully upgraded the OneClick web server and verified any OneClick customizations, we recommend that an administrator remove this directory. The directory uses available disk space unnecessarily.

New OneClick Privileges

A new version of DX NetOps Spectrum sometimes includes new privileges that are assigned, by default, to one or more of the default DX NetOps Spectrum roles. For example, DX NetOps Spectrum assigns these privileges to a default DX NetOps Spectrum role such as OperatorRW. Remember, users who are not assigned these default roles are not automatically granted these new privileges. To grant these new privileges, either explicitly assign them to the users, or assign the default roles to the users.

Additionally, consider assigning the new privileges to one or more custom roles that you have created. Therefore, users that are assigned to only those custom roles are also granted new privileges.

NOTE
For more information on working with users, roles, and privileges, see OneClick Administration .

Files Created During Installation

The DX NetOps Spectrum installation adds the following file types:

Services for Windows

The DX NetOps Spectrum installation adds the following services to Windows Services:

DX NetOps Spectrum Process Daemon Files for Linux

The DX NetOps Spectrum installation adds a process daemon (processd) file to the following startup areas on Linux:

- /etc/rc.d/init.d/processd (10.4.2) or /etc/systemd/system/processd.service (10.4.2.1)
- /etc/rc.d/rc0.d/K*processd
- /etc/rc.d/rc1.d/K*processd
- /etc/rc.d/rc2.d/K*processd
- /etc/rc.d/rc3.d/S*processd
- /etc/rc.d/rc4.d/K*processd
- /etc/rc.d/rc5.d/S*processd
- /etc/rc.d/rc6.d/K*processd

NOTE
Your operating system determines the number that is indicated by the symbol, *. For more information about processd, see the Distributed SpectroSERVER section.

WARNING
Do not remove these files because they are required for DX NetOps Spectrum operation.

DX NetOps Spectrum Remote Administration Daemon Files for Linux

The DX NetOps Spectrum installation adds the following DX NetOps Spectrum Remote Administration Daemon (sradmin) files that DX NetOps Spectrum requires for user authentication and distributed administration:

- /etc/systemd/system/sradmin.service (10.4.2.1 and later versions)
- /etc/init.d/sradmin (10.4.2)
- /etc/rc0.d/K*sradmin
- /etc/rc1.d/K*sradmin
- /etc/rc2.d/K*sradmin
- /etc/rc3.d/S*sradmin
- /etc/rc4.d/K*sradmin
- /etc/rc5.d/S*sradmin
- /etc/rc6.d/K*sradmin

NOTE
Your operating system determines the number that is indicated by *.

These files are added to /sw/SPECTRUM/SRAdmin/sradmin.exe or another path if SRAdmin was installed manually.

WARNING
Do not remove these files because they are required for DX NetOps Spectrum operation

Installation Database Savefiles

The installation automatically creates two savefiles in the <*$SPECROOT*>/SS directory. Each file contains a copy of the database modeling catalog that was installed. The first file is date-stamped, with the extension .after. A copy of the .after file is created and named legacy.SSdb (overwriting any previous legacy.SSdb file).

The legacy.SSdb file is used with the SSdbload utility to reinitialize the database with the most recently installed modeling catalog. Whereas, the .after files let you restore the catalog that is associated with any particular installation. A sequential

counter following the date portion of the file name lets you distinguish between multiple .after files generated on the same day. For example, if three of these files were generated on May 4, 2006, they would be labeled as follows:

- db_20060504,1.after.SSdb
- db_20060504,2.after.SSdb
- db_20060504,3.after.SSdb

Post-Installation Configurations

Set OneClick Client Restrictions

Client access to DX NetOps Spectrum includes access to OneClick web server installations. When the OneClick clients use the OneClick web server for connections, adding them to Host Security is not necessary.

> **NOTE**
> You can configure host security for the OneClick web server using Remote Address Filter and Remote Host Filter in Apache Tomcat. See http://tomcat.apache.org for details.

Follow these steps:

1. Navigate to <*$SPECROOT*>/tomcat/webapps/spectrum/META-INF.
2. Open the context.xml file in this directory, using an XML editor.
3. Locate the following line:
   ```
   <Context path="/spectrum" docBase="spectrum">
   ```
   Enter the following lines under this line:
   ```
   <Valve className="org.apache.catalina.valves.RemoteAddrValve"
   allow=""
   deny=""/>
   ```
4. Enter IP addresses as values for the deny parameter to specify users in which you want to restrict OneClick access. For example, you can allow all users for a given IP address range, but you can exclude one or more specific users.
5. Optionally, enter IP addresses as values for the allow parameter to specify specific IP addresses in which you want to give OneClick access. For example, you can enter 10.254.*.* to include all IP addresses in your network that are in the “10.254” IP address range.
6. Save and exit the file.
   OneClick client restrictions are set.

Set Up Client Access to DX NetOps Spectrum in a Distributed Environment

If you selected a Main Location Server other than the host system during installation, enter the name of the host system in the .hostrc file of the Main Location Server.

> **NOTE**
> The installation automatically enters the name of the Main Location Server in the .hostrc file of the host system.

Follow these steps:

1. Start the SPECTRUM Control Panel on the system that is designated as the Main Location Server.
2. Select Configure, Host Security.
3. Add the name of the host system to the Server List and click OK.
   The client access to DX NetOps Spectrum is set up.

Change the OneClick Web Server Port

Change the default server shutdown port for the OneClick web server when:

- Your OneClick web server contains multiple instances of Apache Tomcat.
- Those instances of Apache Tomcat are using the default server shutdown port (8005).

Apache Tomcat cannot start on a system with another instance running on it.

Follow these steps:

1. Navigate to the following directory:
   ```
   <$SPECROOT>/tomcat/conf
   ```
2. Open the server.xml file, using a text editor.
3. Navigate to the following line:
   ```
   - <Server port="8005" shutdown="SHUTDOWN" debug="0">
   ```
4. Change the server port value to the new server shutdown port number. For example, port="8099"
5. Restart Apache Tomcat, as follows:
   - On Windows:
     - Select All Programs, Administrative Tools, Services from the Start menu.
     - Select SpectrumTomcat from the list.
     - Click Restart the service in the left pane.
   - On Linux:
     - Navigate to the following path:
       ```
       <$SPECROOT>/tomcat/webapps/spectrum
       restart.sh
       ```
     - Enter the following command:

   The web server port is changed.

Configure the Windows Server Scheduled Tasks Service

You can configure the Windows Server 2012 (or later) Scheduled Tasks service to work with the DX NetOps Spectrum Scheduler.

Follow these steps:

1. Click Start, Administrative Tools.
2. Select Task Scheduler.
3. Select Action, AT Service Account Configuration.
4. Select the option Another user account and select Change user.
5. Enter a user name (including the domain, if applicable) of a valid DX NetOps Spectrum user (for example, WORKGROUP\jsmith).

   NOTE
   By default, this dialog contains the current user. If the default is the DX NetOps Spectrum user, no change is necessary. Scheduled tasks are run on behalf of the designated user.
6. Enter the user password in the Password and Confirm Password fields and click OK.
7. Click OK in the AT Service Configuration dialog.
   The Windows Scheduled Tasks service is configured.

Landscape Polling Interval Configuration in Fault-Tolerant Environments

In fault-tolerant DX NetOps Spectrum environments, OneClick checks the status of the SpectroSERVER by polling each landscape at 10-second intervals by default. Frequent polling shortens the failover time to the secondary SpectroSERVER when the primary SpectroSERVER goes down. This polling also avoids missing any SpectroSERVER restarts.

You can increase the landscape polling interval for better performance. You can configure the interval by editing the value of domainPollingInterval in the context.xml file on the OneClick web server. The value of domainPollingInterval is the seconds between polls to the SpectroSERVER to determine its status.

To increase the polling interval, edit the domainPollingInterval value in the context.xml file (located in the <*$SPECROOT*>/tomcat/webapps/spectrum/META-INF directory). For example, to change the landscape polling interval to 60 seconds, change the value of domainPollingInterval from 10 to 60.

NOTE
For the changes to take effect, stop and restart the OneClick web server.

Change the Model Type for a Single Device Type

Changing the Model Type for a Single Device Type

You can use the NewMM.pl post-installation script to change the model type for a single device type automatically. This script preserves many key attributes, relationships, and other elements.

This procedure changes the model type for all models that have the same specified system Object ID and the same specified starting model type.

WARNING
Do not perform this procedure until you modify the model type mapping for the device type in the Device Certification utility. If you do not perform this procedure with the Device Certification utility, your changes cannot be communicated to the SpectroSERVER database, causing unexpected alarms. For information about using the Device Certification utility, see the Certification section.

Follow these steps:

1. Verify that the SpectroSERVER is running.
2. Run the following command from the *$SPECROOT*/Install-Tools/PostInstall/ directory:
   ```
   NewMM.pl -m
   ```
   NOTE
   On Windows, all necessary scripts must be run from a bash shell. They do not run as expected from a DOS command prompt.
3. Enter the host name or IP of the VNM and press Enter.
4. Enter the SpectroSERVER landscape handle when prompted, and press Enter.
5. Enter the system Object ID of the model when prompted, and press Enter.
6. Enter the current model type of the model when prompted, and press Enter.
7. Enter the model type that you want to change to when prompted, and press Enter.
   The model type is changed.
   The log file, NewMM_Log_*DATE*, is created in the *$SPECROOT*/Install-Tools/PostInstall/ directory.
8. To confirm the model type conversion, verify the following log file:
   ```
   NewMM_Log_DATE
   ```
   Model type for a single device type is modified.

Starting OneClick Web Server

Prepare the SpectroSERVER to Communicate With the OneClick Web Server

Make sure that the SpectroSERVER and OneClick can communicate with one another.

NOTE
Some service packs require updates to the SpectroSERVER and the OneClick web server. See the DX NetOps Spectrum Software Release Notice for more information.

Follow these steps:

1. Verify if the DX NetOps Spectrum version installed on the SpectroSERVER host is the same as the DX NetOps Spectrum version you are installing on the OneClick web server.
   To verify, navigate to <*$SPECROOT*>/Install-Tools and view the .history file using a text editor. If the version is different, install the same version of DX NetOps Spectrum.

   WARNING
   For each SpectroSERVER, there must be an entry in the .hostrc file for the computer hosting the OneClick web server. For more information, see OneClick Administration.
2. Ensure that all associated SpectroSERVERs are running.
3. Ensure that the computer on which you are installing the OneClick web server has host access to all associated SpectroSERVER computers. On each SpectroSERVER host:
   a. Launch the DX NetOps Spectrum Control Panel.
   b. Select Configure, Host Security.
   c. Ensure that the Server List contains either:
      - The host name of the designated OneClick host (OneClick Web Server)
      - A plus (+) sign (meaning unrestricted access)

      NOTE
      See OneClick Administration for more information.
4. Verify that you are connected by pinging the designated DX NetOps Spectrum host using its host name.
5. Designate an existing user as the OneClick administrator or create a OneClick administrator. Verify that this user is a valid administrator, as follows:
   a. Launch the DX NetOps Spectrum Control Panel.
   b. Select Control, Users.
   c. Verify that the user model designated as the OneClick administrator exists.
   d. If the user does not exist, select Create.
   e. Enter the user name in the User Name field, enter a password in the New Password and Confirm New Password fields, and click OK.
      The user is created as a super user and has access to all models and privileges.
   f. Click Close to exit the Users window.

      NOTE
      In a distributed environment, this administrative user must exist in all landscapes. For more information, see Distributed SpectroSERVER Administration.
6. Ensure that the computer on which you are installing the OneClick web server has access to the SpectroSERVER.
7. For all Windows platforms, ensure that you can resolve the SpectroSERVER host name from the OneClick web server by editing the local hosts file:
   a. Navigate to the C:\Windows\system32\drivers\etc directory.
   b. Open the hosts file with a text editor.
   c. Add entries per the comments in the hosts file.
   d. Save the file.
8. On Linux, ensure that you have host name resolution to the SpectroSERVER from the OneClick web server by editing the local hosts file. If you are not using a name service, edit your local hosts file as follows:

a. To test host name resolution, ping the DX NetOps Spectrum host using only the host part of its fully qualified domain name.
For example, to ping host.company.com, enter shell> ping host. If the ping fails, edit the file /etc/hosts to reflect the IP and name of the DX NetOps Spectrum host. The SpectroSERVER is prepared to communicate with OneClick.

Start DX NetOps Spectrum on Windows

After you install DX NetOps Spectrum, you can start DX NetOps Spectrum on Windows.

Click Start, Programs, CA, SPECTRUM, Administrator, Control Panel.

DX NetOps Spectrum starts and the DX NetOps Spectrum Control Panel appears.

Start DX NetOps Spectrum on Linux

After you install DX NetOps Spectrum, you can start DX NetOps Spectrum on Linux.

Follow these steps:

1. Navigate to the directory path where you installed DX NetOps Spectrum (for example, /usr/SPECTRUM/).
2. Set up your remote display, if needed.
3. Navigate to the bin directory and run the following command:
   ```
   ./SCP
   ```
   DX NetOps Spectrum starts and the DX NetOps Spectrum Control Panel appears.

Initiate a Remote Display of DX NetOps Spectrum

You can set up a Windows system to display DX NetOps Spectrum remotely when DX NetOps Spectrum is running on a Linux system. The Linux system must be installed with the applications that you want to display remotely on Windows. Also, the Linux system must be configured to support Telnet services. The Windows system must be configured to support a Telnet client.

NOTE
DX NetOps Spectrum supports one remote display session open at a time on a client system.

Follow these steps:

1. Ensure that the DX NetOps Spectrum Control Panel and any applications that you want to display remotely are installed on the Linux system. Also, ensure that they are configured to support remote display.
2. Click Run from the Windows Start menu.
   The Run window appears.
3. Run the following command:
   ```
   Telnet <linux host name>
   ```
   The Linux login dialog appears.
4. Log in to the Linux system using your DX NetOps Spectrum user name and password.
   The system reports your last login, host name, and operating system version. The Linux prompt follows.
5. To set the remote display environment, run the following commands:
   - In the K (default) shell, enter:
     ```
     export DISPLAY=<remote display hostname>:0.0
     ```
   - In the C shell, enter:
     ```
     setenv DISPLAY <remote display hostname>:0.0
     ```
   - In the Bourne shell, enter:
     ```
     DISPLAY=<remote display hostname>:0.0 export display
     ```

NOTE
For frequent use of remote display, you can avoid repeating this task at each login by adding the DISPLAY environment to your profile.

6. Navigate to the <*$SPECROOT*>/bin directory.
7. Enter the following command:
   ```
   ./SCP
   ```
   The DX NetOps Spectrum Control Panel appears, providing you with access to all DX NetOps Spectrum Control Panel functions, including access to client DX NetOps Spectrum applications.

Terminate a Remote Display of DX NetOps Spectrum

You can terminate a remote display of DX NetOps Spectrum.

Follow these steps:

1. Exit all remotely displayed DX NetOps Spectrum applications properly.
2. Enter exit at the prompt in the Telnet terminal window.
   The Telnet session is ended.

Launch the OneClick Console

This article describes how you can set up and start the OneClick client (OneClick Console).

Prerequisites

Review the following prerequisites:

1. Verify that your workstation meets the minimum OneClick Client Requirements for Windows and Linux.
2. Install JRE.
3. If necessary, associate the JNLP files with Java Web Start.

Launch the OnceClick Client (OneClick Console)

After the JRE and required Java components are installed, you can launch the OneClick Console. You can launch the OneClick Console from a browser on your computer where it (OneClick Console) is installed.

Follow these steps:

1. Open the OneClick home page in a browser using the URL that your administrator has provided. The URL has the following format:
   ```
   http://<hostname>:<portnumber>/
   ```
   NOTE
   <*hostname*> is the name of the OneClick web server. Use:<*portnumber*> only if the OneClick web server does not use the default port 80 on Windows or 8080 on Linux. If you cannot access the OneClick web server, notify your administrator.
2. Enter your OneClick login credentials, if prompted.
   The OneClick home page opens.

   NOTE
   Any date and time information that is shown in OneClick is localized to reflect the time zone where the OneClick client is installed and running.
3. Install JRE and JCEUnlimited Strength Files and Java Web Start if you have not done so already on Windows or Linux.
4. Click Start Console.

5. Enter your OneClick user name and password again, if prompted.
   OneClick starts and the OneClick Console opens.

Troubleshooting Installation Problems

DX NetOps Spectrum Installation fails due to spaces in the path

Symptom: DX NetOps Spectrum installation fails due to spaces in the absolute path (location) where the installer exe is placed. **Resolution:** You need to ensure that the folder in which the installation kit is copied and extracted does not have any spaces. DX NetOps Spectrum installation will fail if the installer (.exe) file is in a location with spaces in its absolute / full path.

Installation Media Does Not Contain Installation Information

Symptom:

The following message appears:

```
The <sp> installation media does not contain the installation information for this platform.
```

Solution:

This message appears when you do not have the correct installation media for the host platform. Use the DX NetOps Spectrum installation media of the platform on which you are installing.

<index file name> Cannot Be Found

Symptom:

I received the following error:

```
Error: <index file name> not found!
```

Solution:

One of the following conditions caused this error:

- Extraction of the Installation record from the distribution medium was incomplete.
- The Installation record files were improperly removed or modified before the installation.

Retry the installation. If the failure persists, contact Technical Support.

Received a Landscape Handle Error

Symptom:

The following message appears:

```
** Error during Set Landscape Handle
```

Solution:

The installation is unable to set the SpectroSERVER landscape handle value, which is a serious problem. Contact Technical Support.

Received an InvocationTargetException Error

Symptom:

The following message appears:

```
Invocation of this Java Application has caused an InvocationTargetException. This application will now exit.
 (LAX)
Stack Trace:
java.awt.HeadlessException:
No X11 DISPLAY variable was set, but this program performed an operation which
requires it.
at java.awt.GraphicsEnvironment.checkHeadless
(GraphicsEnvironment.java:159)
at java.awt.Window.<init>(Window.java:317)
at java.awt.Frame.<init>(Frame.java:419)
at java.awt.Frame.<init>(Frame.java:384)
at javax.swing.JFrame.<init>(JFrame.java:150)
at com.zerog.ia.installer.LifeCycleManager.f(DashoA8113)
at com.zerog.ia.installer.LifeCycleManager.g(DashoA8113)
at com.zerog.ia.installer.LifeCycleManager.a(DashoA8113)
at com.zerog.ia.installer.Main.main(DashoA8113)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke
(NativeMethodAccessorImpl.java:39)
at sun.reflect.DelegatingMethodAccessorImpl.invoke
(DelegatingMethodAccessorImpl.java:25)
at java.lang.reflect.Method.invoke(Method.java:585)
at com.zerog.lax.LAX.launch(DashoA8113)
at com.zerog.lax.LAX.main(DashoA8113)
This Application has Unexpectedly Quit: Invocation of this Java Application has caused an
 InvocationTargetException. This application will now exit. (LAX)
```

Solution:

Your DISPLAY environment variable is not set to the host name of the system on which you are running the installation software. Set the environment variable correctly.

Received a Database Initialization Error

Symptom:

I received the following message:

```
** Error during Database Initialization
```

Solution:

The installation is unable to create the SpectroSERVER database, which is a serious problem. Contact Technical Support.

Received a Database Saving Error

Symptom:

The following message appears:

```
** Error during Database save as db_<extension>
```

Solution:

The installation is unable to save the existing SpectroSERVER database. The most likely cause is that the SpectroSERVER database is write-protected or does not exist (for example, the database was deleted).

To make the SpectroSERVER database writable, use the chmod utility. Then, reinstall DX NetOps Spectrum. If this procedure does not work, contact Technical Support.

Received a VNMRC File Processing Error

Symptom:

The following message appears:

```
** Error during Processing of vnmrc file
```

Solution:

The installation is unable to set values properly in the SpectroSERVER defaults file, SS/.vnmrc. This problem cannot be resolved at the installation site. Contact Technical Support.

Installation Owner User Problems

Valid on Linux

Symptom:

I have installation owner user problems.

Solution:

Follow these steps:

1. Verify the name resolution (without the domain name) to the <*DX NetOps Spectrum host*>.
2. Select Users from the Control menu in the DX NetOps Spectrum Control Panel and verify that the installation owner appears in the Users dialog.
3. Verify that the same version of DX NetOps Spectrum is installed on the computers where OneClick and the SpectroSERVER are installed.

OneClick Web Server Error Message

Symptom:

The resource at http://<*server*>/spectrum/index.jsp was not found. Authorization could not be completed.

Solution:

You could have another Web services application running on the same port Apache Tomcat is attempting to use. Stop (and disable, if necessary) the other application and associated services and restart the Apache Tomcat service.

OneClick Web Server Shuts Down

Symptom:

I upgraded to VMware 2.0 and it runs an Apache Tomcat server of its own. After I install the OneClick web server, the OneClick web server shuts down when it attempts to bind to port 8005. Then, I receive the following error message:

```
- StandardServer.await: create[8005]:
java.net.BindException: Address already in use: JVM_Bind
```

Solution:

By default, Apache Tomcat uses port 80 on Windows platforms and port 8080 on Linux platforms. If SSL is configured, Apache Tomcat uses port 443. Apache Tomcat also uses the default server shutdown port 8005. When installing the OneClick web server, be sure that other applications on the same computer do not use these ports. Or, you can change the ports on the instance of Apache Tomcat that DX NetOps Spectrum uses.

NOTE
We recommend that you do *not* install the OneClick web server on a computer where an instance of Apache Tomcat is already running.

Troubleshooting OneClick Client Problems

This section lists troubleshooting information for some of the common problems encountered with respect to the OneClick Client.

Odd OneClick Behavior

Symptom:

After I upgrade DX NetOps Spectrum, I notice that the OneClick client is behaving oddly on one of my computers.

Solution:

Try to reproduce the problem on another computer where OneClick has not been used. If you cannot reproduce the problem on this computer, the Java cache most likely did not update during the DX NetOps Spectrum upgrade.

On the computer where the OneClick client exhibits this problem, clear the java cache:

1. Access the Java Control Panel:
   - On Windows platforms, click Start, Control Panel, and then double-click Java.
   - On Linux platforms, launch <*JRE install directory*>/bin/jcontrol.
2. Click the View button under Temporary Internet Files on the General tab.
3. Select the DX NetOps Spectrum OneClick Console Application and click the X button in the toolbar.
   The selected item is removed.

OneClick Client Fails to Launch

Symptom:

I tried to launch OneClick, but it failed to start.

Solution:

When installing the JRE, which includes Java Web Start, on Windows, the default cache directory is the installing home directory of the user. However, if any part of the full path of the home directory, including the username, includes the exclamation character (!), OneClick fails to launch properly.

Follow these steps:

1. Click Start, Control Panel, and then double-click Java.
2. Select the General tab and then click Settings.
3. Click Change to change the location where temporary files are located. Select a path that does not include the exclamation character.

Solution:

Your server already had a Java version installed when you tried to launch OneClick for the first time after installation. Each time OneClick is launched, a check for a minimum version of Java is performed. Typically, you see a prompt asking you to update the JRE when required. But sometimes, this update fails.

If your inability to launch the OneClick client is related to a failed update of the JRE, install the software by clicking "**Install JRE and JCEUnlimited Strength Files**" on the OneClick home page. This link calls up a page with a link to the required version of the JRE.

OneClick Console Does Not Open (Windows)

Valid on Windows

Symptom:

I tried to launch the OneClick Console. The Java splash screen appeared but vanished, and the OneClick Console did not open.

Solution:

The JRE is not installed correctly. The OneClick client server must have the correct JRE version. If you are upgrading from a previous release, an older version of Java is already installed in the default Windows location, C:\Program Files \Jave\jre6. That version is causing the problem.

Follow these steps:

1. Repeat the procedure that is outlined in Install JRE and JCEUnlimited Strength Files, and Java Web Start on Windows. However, when installing the JRE, select a location other than the default or the existing location.
2. After installation is complete, shut down any existing OneClick clients.
3. Clear the cache of old jar files to run the OneClick Console using the new version of Java:
   a. In the Windows Start, Run dialog, type **javaws -viewer** and click OK.
   The Java Control Panel and the Java Cache Viewer dialogs open.
   b. Review the applications in the cache on the Java Cache Viewer dialog, delete any existing OneClick applications in this view, and click Close.
   c. On the Java Control Panel dialog on the General tab, click Settings in the Temporary Internet Files section.
   The Temporary Files Settings dialog opens.
   d. Take one of the following steps:
      - If the option to 'Keep temporary files on my computer' is selected, click Delete Files. Verify that the Applications and Applets option on the Delete Temporary Files dialog is selected, and click OK.
      - If the option is cleared, manually delete the temporary files. Navigate to your <*Windows home directory*>\Local Settings\Temp folder and delete all 'jar_temp<number>' files.
4. Start a new OneClick Console.

Firefox Download Error Dialog (Linux)

Valid on Linux

Symptom:

I get the Firefox Download Error dialog when I attempt to start the OneClick Console.

Solution:

The Firefox Download Error dialog opens when you attempt to start the OneClick Console due to one of the following conditions:

- The correct JRE version is not installed on the Linux system.
- The correct JRE version is not configured properly.
- The .jnlp file type is not associated with the JavaWS application.

Follow these steps:

1. Verify that the correct JRE version is installed on the Linux OneClick Console system. If the JRE is installed, go to the next step. If it is not installed, follow the instructions in Install JRE, JCEUnlimited Strength Files, and Java Web Start on Linux to install the JRE and required Java components.
2. Configure the Firefox browser to associate .jnlp file types with the JavaWS application:
   a. In a Firefox browser window, click Edit, and Preferences.

The Preferences dialog opens.

b. Click Downloads.
c. Click View, and Edit in the Download Actions section.
The Download Actions dialog opens.
d. Locate the entry for the JNLP extension. Verify that the action associated with the file type is Open with JavaTM Web Start Launcher. This association is made when installing the JRE. If a different association is listed, click Change Action.
The Change Action dialog opens.
e. Select 'Open them with this application', and click Browse, if necessary.
The Select Helper Application dialog opens.
f. Select javaws from the location where you installed it, and click OK.
g. Click Close.

3. Click OK.

OneClick Fails to Start, Access Denied (Windows)

Valid on Windows

Symptom:

I tried to launch OneClick, but OneClick failed to start. I received the following error:

```
opening oneclick.jnlp...
Access to the specified device, path, or file is denied.
```

Solution:

The .jnlp file type is not associated with the javaws.application. Verify that the .jnlp file extension is mapped to the javaws.exe application.

Cannot Log In to OneClick Client

Symptom:

I am unable to log in either at the OneClick home page (that is, http://<*hostname*>/spectrum or *http://*<*hostname*>:<*portnumber*>/spectrum), or when launching the OneClick client.

Solution:

Check for these common problems:

1. Does the user name that is entered at login represent a valid user?
2. Does the user exist at the main location server?
3. Is the SpectroSERVER, or the secondary SpectroSERVER, running properly?
4. On the primary SpectroSERVER, does the user have either the administrator or operator role? To verify the user role, select the Users tab in the OneClick Console.
5. Is the password correct? To verify the user password, select the Users tab in the OneClick Console.

OneClick throws exceptions when spectrumgtw probe sends a large payload

Symptom:

When spectrumgtw probe sends large payload to SpectroSERVER during the Host server sync, the OneClick throws exceptions.

Solution:

From 10.4.1, you can customize the maximum size of the payload that you want to receive from spectrumgtw probe to SpectroSERVER during the Host server sync. Set the `org.apache.cxf.stax.maxInputSizeInMB` parameter to a desired value in the `Spectrum_HOME\tomcat\webapps\spectrum\WEB-INF\web.xml` file.
Default: 8 MB

Upgrading

> **NOTE**
>
> See, the DX NetOps Spectrum Patch General Information page for detailed steps on DX NetOps Spectrum patch installation.

Get Started and Upgrade to 10.4.2.2!

Upgrade Path 10.4.2.2

The DX NetOps Spectrum 10.4.2.2 version is the same as DX NetOps 20.2.5 release.

> **IMPORTANT**
> You must first install 10.4.2 as the base version to install DX NetOps Spectrum 10.4.2.2 release.

> **NOTE**
> For more information on the upgrade process, see Upgrade to 10.4.2

Features and Enhancements in Current Release

For more information about the new features and enhancements in the current release, see the Features and Enhancements page.

Get Started and Upgrade to 10.4.2.1!

Upgrade Path 10.4.2.1

> **IMPORTANT**
> You must first install 10.4.2 as the base version to install DX NetOps Spectrum 10.4.2.1 release.

> **NOTE**
> For more information on the upgrade process, see Upgrade to 10.4.2

Features and Enhancements in Current Release

For more information about the new features and enhancements in the current release, see the Features and Enhancements page.

Get Started and Upgrade to 10.4.2!

Considerations

- Before upgrading to 10.4.2, if **-loglevel debug** is enabled in sdm.config, you must remove it. Once SpectroSERVER and the corresponding Secure Domain Connector (SDC) are upgraded, you can add it back.
- For 10.4.2, ensure that SDC and Secure Domain Manager (SDM) have the same version.
- During an upgrade scenario, when SpectroSERVER is getting upgraded, stop the SDC services. Once SpectroSERVER is upgraded, you can then upgrade SDC to 10.4.2.
- While performing modeling gateway operations from 10.2.0 or 10.2.1 to 10.4.2, remove all entries of SNMPv3 profiles and import on 10.4.2. After upgrade, create SNMPv3 profiles manually.

Integration Compatibility Matrix

To know more about version compatibility of other integrated products with this release of DX NetOps Spectrum, see the Integration Compatibility chart.

Upgrade Path

10.4.2 is a full release. You can directly upgrade to 10.4.2 from 10.2.x, 10.3.x, and 10.4.x releases (for example, 10.2, 10.2.1, 10.2.2, 10.2.3, 10.3, 10.3.1, 10.3.2, 10.4, and 10.4.1). You do not need to install any other releases or patches.

Features and Enhancements in Current Release

For more information about the new features and enhancements in the current release, see the Features and Enhancements page.

Platform Updates

For third-party software updates in this release, see the third-party software page.

FAQs

Q: In a distributed SS (DSS) environment, what is the supported and best practice for upgrading servers?

A: Upgrade the Primary MLS (main location server) SpectroSERVER first, followed by the Primary OneClick server, followed by the remaining Primary SpectroSERVERs. It is important that you start the primary MLS first, before starting the other primary SpectroSERVERs. This allows all of the cross landscape and "MOM" (manager of managers) functions to process and complete. Once the primary MLS is running and active, you can start the rest of the primaries. Upgrade the Secondary SpectroSERVERs last (including all SpectroSERVER and non-primary OneClick servers). Once all SpectroSERVERs have been upgraded, be sure to stop and restart all DX NetOps Spectrum OneClick Tomcat processes again to update the Tomcat cache with the updated SS info. Not doing this will cause OneClick issues, such as landscapes not showing in the client.

> **NOTE**
> In a DSS setup, if you stop a primary SS to test your failover scenario without performing a successful database synchronization (via online backup) first, you may see the landscapes disappear in OneClick. To make sure this does not happen, manually start the online backup and allow the database synchronization to occur. Once complete, you can then test your failover scenario.

Upgrade to 10.4.2

This section provides information about how you can upgrade to this release.

Prerequisites

Before you upgrade to the latest version of DX NetOps Spectrum, perform the following tasks:

1. Review the best practices for upgrading in a fault-tolerant and non-fault tolerant environments. Also, preserve the customized support files.
2. Stop all running applications other than DX NetOps Spectrum.
3. Stop the following DX NetOps Spectrum applications:
   - Shut down all OneClick clients by logging off all users from OneClick on the Client Details web page in the OneClick home page.

     NOTE

     For more information about shutting down OneClick clients, see OneClick Administration.
   - Stop the SpectroSERVER and the Archive Manager by clicking Stop SpectroSERVER in the Spectrum Control Panel and then close the Spectrum Control Panel. Or you can stop the SpectroSERVER and Archive Manager from the command line by running the "<$SPECROOT>/bin/stopSS.pl" as DX NetOps Spectrum Owner at the command prompt.
   - Stop all VnmSh connections. For more information about stopping VnmSh connections, see Command Line Interface.
   - Close all Bash shells.
   - Ensure the 'Java' folder is not opened (under <$SPECROOT>) during the upgrade.
4. If you have installed DX NetOps Spectrum Report Manager, verify that free disk space on the system is at least twice the size of the largest MYD file under *$SPECROOT*/mysql/data/reporting.
5. Do not install third-party software that uses MySQL because the results can be unpredictable.

Upgrade Procedure

Upgrading from a previous version of DX NetOps Spectrum to the latest one can be complex, because each release promises higher quality with the latest technologies and services. Ensure that you go through the upgrade options that best suit your requirements.

1. You can perform either:
   - In Place Upgrade (With no fault-tolerant SpectroSERVERs configured in your deployment, a temporary fault-tolerant system is used during the upgrade. Once the temporary system is configured, the primary SpectroSERVER is disabled for an in-place upgrade. Meanwhile, the temporary, secondary SpectroSERVER system takes over core network management duties).
   - Upgrade Models to ensure compatibility between the SpectroSERVER database and a new version of DX NetOps Spectrum after upgrading.
2. Download the release.
   a. Sign in to the support portal.
   b. Click Enterprise Software.
   c. Click Product Downloads.
   d. Enter the product name in the search bar and click the search icon.
   e. Click the link that is displayed after the search is completed.
   f. Select the release to download.
3. Install DX NetOps Spectrum.
4. Review the post-installation configurations.

Post Upgrade Tasks

- Archive Manager Database (DDM Database) Tables Conversion: With the previous release, it is recommended that you convert your Archive Manager database tables from MyISAM to InnoDB for better performance. Refer to this section at the end of this page, for more information.
- Migration Upgrade (Best Practice - Preserve Customized Support Files)
- Upgrading Models to ensure compatibility between the SpectroSERVER database and a new version of DX NetOps Spectrum after upgrading.

Other Upgrade Tasks

- If you have configured OneClick to launch from Report Manager using SSL, configure this modification again. For more information about this modification, see OneClick Administration.
- If any models display Minor (yellow) alarms with a probable cause of DIFFERENT TYPE MODEL, clear the alarms. To convert all eligible models, run the NewMM.pl script from the Spectrum\Install-Tools\PostInstall directory. For more information, see Troubleshooting the Post-Upgrade Installation Script.

Get Started and Upgrade to 10.4.1!

Considerations

- Before upgrading to 10.4.1, if **-loglevel debug** is enabled in sdm.config, you must remove it. And, once SpectroSERVER and corresponding Secure Domain Connector (SDC) are upgraded, you can add it back.
- For 10.4.1, ensure that SDC and Secure Domain Manager (SDM) have the same version.
- During an upgrade scenario, when SpectroSERVER is getting upgraded, stop the SDC services. And, once SpectroSERVER is upgraded, you can then upgrade SDC to 10.4.1.
- While performing modeling gateway operations from 10.2.0 or 10.2.1 to 10.4.1, remove all entries of SNMPv3 profiles and import on 10.4.1. After upgrade, create SNMPv3 profiles manually.

Integration Compatibility Matrix

To know more about version compatibility of other integrated products with DX NetOps Spectrum 10.4.1, see the Integration Compatibility chart.

Upgrade Path

10.4.1 is a full release. You can directly upgrade to 10.4.1 from 10.2.x, 10.3.x, and 10.4 releases (for example, 10.2,10.2.1,10.2.2, 10.2.3, 10.3, 10.3.1, 10.3.2, 10.4). You do not need to install any other releases or patches.

FAQs

Q: In a distributed SS (DSS) environment, what is the supported and best practice for upgrading servers?

A: Upgrade the Primary MLS (main location server) SpectroSERVER first, followed by the Primary OneClick server, followed by the remaining Primary SpectroSERVERs. It is important that you start the primary MLS first, before starting the other primary SpectroSERVERs. This allows all of the cross landscape and "MOM" (manager of managers) functions to process and complete. Once the primary MLS is running and active, you can start the rest of the primaries. Upgrade the Secondary SpectroSERVERs last (including all SpectroSERVER and non-primary OneClick servers). Once all SpectroSERVERs have been upgraded, be sure to stop and restart all DX NetOps Spectrum OneClick Tomcat processes again to update the Tomcat cache with the updated SS info. Not doing this will cause OneClick issues, such as landscapes not showing in the client.

NOTE

In a DSS setup, if you stop a primary SS to test your failover scenario without performing a successful database synchronization (via online backup) first, you may see the landscapes disappear in OneClick. To make sure this does not happen, manually start the online backup and allow the database synchronization to occur. Once complete, you can then test your failover scenario.

Features and Enhancements in Current Release

For more information about the new features in the current release, see the Features and Enhancements page.

Platform Updates

For third-party software updates in this release, see the third-party software page.

Upgrade to 10.4.1

This section provides information about how you can upgrade to this release.

Prerequisites

Before you upgrade to the latest version of DX NetOps Spectrum, perform the following tasks:

1. Review the best practices for upgrading in a fault-tolerant and non-fault tolerant environments. Also, preserve the customized support files.
2. Stop all running applications other than DX NetOps Spectrum.
3. Stop the following DX NetOps Spectrum applications:
   - Shut down all OneClick clients by logging off all users from OneClick on the Client Details web page in the OneClick home page.

     NOTE

     For more information about shutting down OneClick clients, see OneClick Administration.
   - Stop the SpectroSERVER and the Archive Manager by clicking Stop SpectroSERVER in the Spectrum Control Panel and then close the Spectrum Control Panel. Or you can stop the SpectroSERVER and Archive Manager from the command line by running the "<$SPECROOT>/bin/stopSS.pl" as DX NetOps Spectrum Owner at the command prompt.
   - Stop all VnmSh connections. For more information about stopping VnmSh connections, see Command Line Interface.
   - Close all Bash shells.
   - Ensure the 'Java' folder is not opened (under <$SPECROOT>) during the upgrade.
4. If you have installed DX NetOps Spectrum Report Manager, verify that free disk space on the system is at least twice the size of the largest MYD file under *$SPECROOT*/mysql/data/reporting.
5. Do not install third-party software that uses MySQL because the results can be unpredictable.

Upgrade Procedure

Upgrading from a previous version of DX NetOps Spectrum to the latest one can be complex, because each release promises higher quality with the latest technologies and services. Ensure that you go through the upgrade options that best suit your requirements.

1. You can perform either:
   - In Place Upgrade (With no fault-tolerant SpectroSERVERs configured in your deployment, a temporary fault-tolerant system is used during the upgrade. Once the temporary system is configured, the primary SpectroSERVER

is disabled for an in-place upgrade. Meanwhile, the temporary, secondary SpectroSERVER system takes over core network management duties).
   - Upgrade Models to ensure compatibility between the SpectroSERVER database and a new version of DX NetOps Spectrum after upgrading.
2. Download the release.
   a. Sign in to the support portal.
   b. Click Enterprise Software.
   c. Click Product Downloads.
   d. Enter the product name in the search bar and click the search icon.
   e. Click the link that is displayed after the search is completed.
   f. Select the release to download.
3. Install DX NetOps Spectrum.
4. Refer to the post-installation configurations.

Post Upgrade Tasks

- Archive Manager Database (DDM Database) Tables Conversion: With the previous release, it is recommended that you convert your Archive Manager database tables from MyISAM to InnoDB for better performance. Refer to this section at the end of this page, for more information.
- Migration Upgrade (Best Practice - Preserve Customized Support Files)
- Upgrading Models to ensure compatibility between the SpectroSERVER database and a new version of DX NetOps Spectrum after upgrading.

Other Upgrade Tasks

- If you have configured OneClick to launch from Report Manager using SSL, configure this modification again. For more information about this modification, see OneClick Administration.
- If any models display Minor (yellow) alarms with a probable cause of DIFFERENT TYPE MODEL, clear the alarms. To convert all eligible models, run the NewMM.pl script from the Spectrum\Install-Tools\PostInstall directory. For more information, see Troubleshooting the Post-Upgrade Installation Script.

Get Started and Upgrade to 10.4!

Integration Compatibility Matrix

For more information about the integration of DX NetOps Spectrum with other CA products, see Integration Compatibility.

Upgrade Path

10.4 is a full release. You can directly upgrade to 10.4 from 10.0 or later releases (for example, 10.0, 10.1, 10.1.x, 10.2, 10.2.x, 10.3, 10.3.x). and so on. You do not need to install any other releases or patches.

FAQs

Q: In a distributed SS (DSS) environment, what is the supported and best practice for upgrading servers?

A: Upgrade the Primary MLS (main location server) SpectroSERVER first, followed by the Primary OneClick server, followed by the remaining Primary SpectroSERVERs. It is important that you start the primary MLS first, before starting the other primary SpectroSERVERs. This allows all of the cross landscape and "MOM" (manager of managers) functions to process and complete. Once the primary MLS is running and active, you can start the rest of the primaries. Upgrade the Secondary SpectroSERVERs last (including all SpectroSERVER and non-primary OneClick servers). Once all SpectroSERVERs have been upgraded, be sure to stop and restart all DX NetOps Spectrum OneClick tomcat

processes again to update the tomcat cache with the updated SS info. Not doing this will cause OneClick issues, such as landscapes not showing in the client.

> **NOTE**
>
> In a DSS setup, if you stop a primary SS to test your failover scenario without performing a successful database synchronization (via online backup) first, you may see the landscapes disappear in OneClick. To make sure this does not happen, manually start the online backup and allow the database synchronization to occur. Once complete, you can then test your failover scenario.

Features and Enhancements in 10.4

- Replaced Oracle JDK with AdoptOpenJDK Java (1.8.0.212)

Get Started and Upgrade to 10.3.2!

Integration Compatibility Matrix

To know more about version compatibility of other integrated products with 10.3.2, see the compatibility matrix chart.

Platform Updates

Third-party software updates for this release include:

- MySQL Workbench 8.0.15
- Apache Tomcat 9.0.14
- Struts 2.5.20
- WebSwing 2.5.10

FAQs

Q: In a distributed SS (DSS) environment, what is the supported and best practice for upgrading servers?

A: Upgrade the Primary MLS (main location server) SpectroSERVER first, followed by the Primary OneClick server, followed by the remaining Primary SpectroSERVERs. It is important that you start the primary MLS first, before starting the other primary SpectroSERVERs. This allows all of the cross landscape and "MOM" (manager of managers) functions to process and complete. Once the primary MLS is running and active, you can start the rest of the primaries.
Upgrade the Secondary SpectroSERVERs last (including all SpectroSERVER and non-primary OneClick servers).
Once all SpectroSERVERs have been upgraded, be sure to stop and restart all DX NetOps Spectrum OneClick tomcat processes again to update the tomcat cache with the updated SS info. Not doing this will cause OneClick issues, such as landscapes not showing in the client.

> **NOTE**
>
> In a DSS set up, if you stop a primary SS to test your failover scenario without performing a successful database synchronization (via online backup) first, you may see the landscapes disappear in OneClick. To make sure this does not happen, manually start the online backup and allow the database synchronization to occur. Once complete, you can then test your failover scenario.

Features and Enhancements in 10.3.2

- SSL support for the Overview Dashboard
- Enablement of TrapX in SDC – SNMPv3 support
- Cisco Meraki Support
- AWS Monitoring
- Quality Enhancements

Upgrade to 10.3.2 from 10.3.0

NOTE

To install 10.3.2, first install 10.3.0, as a base version.

NOTE

To install 10.3.2, first install 10.3.0, as a base version. DX NetOps Spectrum users who are on 10.0 or 10.1.x releases are advised to upgrade to the intermediary major release such as 10.2.0 and then directly upgrade to 10.3. to ensure a stable upgrade and migration process devoid of any loss of data or failure of any functionality.

Best Practices

- Best Practices for upgrading in a fault-tolerant and non-fault tolerant environments.
- Preserve Customized Support Files

Pre-Upgrade Tasks

Before you upgrade to the latest version of DX NetOps Spectrum, ensure you:

1. Are familiar with the best practices that are mentioned above.
2. Stop all running applications other than DX NetOps Spectrum.
3. Stop the following DX NetOps Spectrum applications:
   - Shut down all OneClick clients by logging off all users from OneClick on the Client Details web page in the OneClick home page.

     NOTE

     For more information about shutting down OneClick clients, see OneClick Administration.
   - Stop the SpectroSERVER and the Archive Manager by clicking Stop SpectroSERVER in the Spectrum Control Panel and then close the Spectrum Control Panel. Or you can stop the SpectroSERVER and Archive Manager from the command line by running the "<$SPECROOT>/bin/stopSS.pl" as DX NetOps Spectrum Owner at the command prompt.
   - Stop all VnmSh connections. For more information about stopping VnmSh connections, see Command Line Interface.
   - Close all Bash shells.
   - Ensure the 'Java' folder is not opened (under <$SPECROOT>) during the upgrade.
4. If you have installed DX NetOps Spectrum Report Manager, verify that free disk space on the system is at least twice the size of the largest MYD file under *$SPECROOT*/mysql/data/reporting.
5. Do not install third-party software that uses MySQL because the results can be unpredictable.

Upgrade Procedure from 10.3.0

Upgrading from a previous version of DX NetOps Spectrum to the latest one, can be complex, because each release promises higher quality with the latest technologies and services. Ensure that you go through the upgrade options that best suit your requirement.

1. You can perform either:
   - In Place Upgrade (With no fault-tolerant SpectroSERVERs configured in your deployment, a temporary fault-tolerant system is used during the upgrade. Once the temporary system is configured, the primary SpectroSERVER is disabled for an in-place upgrade. Meanwhile, the temporary, secondary SpectroSERVER system takes over core network management duties).
   - Upgrade Models to ensure compatibility between the SpectroSERVER database and a new version of DX NetOps Spectrum after upgrading.

2. Download the release.
   a. Sign in to the support portal.
   b. Click Enterprise Software.
   c. Click Product Downloads.
   d. Enter the product name in the search bar and click the search icon.
   e. Click the link that is displayed after the search is completed.
   f. Select the release to download.
3. Install DX NetOps Spectrum.
4. Refer to the post-installation configurations.

Post Upgrade Tasks

- Archive Manager Database (DDM Database) Tables Conversion: With the previous release of 10.3.0, it is recommended that you convert your Archive Manager database tables from MyISAM to InnoDB for better performance. Refer to this section at the end of this page, for more information.
- Migration Upgrade (Best Practice - Preserve Customized Support Files)
- Upgrading Models to ensure compatibility between the SpectroSERVER database and a new version of DX NetOps Spectrum after upgrading.

Other Upgrade Tasks

- If you have configured OneClick to launch from Report Manager using SSL, configure this modification again. For more information about this modification, see OneClick Administration.
- If any models display Minor (yellow) alarms with a probable cause of DIFFERENT TYPE MODEL, clear the alarms. To convert all eligible models, run the NewMM.pl script from the CA Spectrum\Install-Tools\PostInstall directory. For more information, see Troubleshooting the Post-Upgrade Installation Script.

Get started and upgrade to 10.3.1!

Special Considerations

NOTE

When upgrading from 10.3 to 10.3.1, ensure you reconfigure and perform a fresh 'Topology Store Configuration' integration on 10.3.1. Upgrading from 10.3 to 10.3.1 without reconfiguration, breaks the 'Topology Store Configuration" integration. To reconfigure, first, disable and then enable this integration by providing a username and password.

NOTE

When upgrading to 10.3.1 from 10.3 (directly) or 10.2.x (indirectly), if there were any alarms on the devices managed through Secure Domain Connector (SDC), those alarms will not be correlated to SDC lost alarm and only the newly generated alarms will be correlated, post-upgrade.

In SDC FT scenario if both SDCs are down then only device down alarms will be correlated to the "Secure Domain Lost" alarm.

- This release supports direct upgrades only from 10.3. If you are upgrading from 10.2.x ensure you first upgrade to 10.3 which is the base version to upgrade to 10.3.1.
- We recommend that you upgrade from 10.1.x versions to a supported version of DX NetOps Spectrum by **December 2018**, to avoid 'gold key' expiration issues. For more information and guidance contact CA Support.
- 10.0 is now end of support, we recommend that you upgrade to 10.2 or 10.2.x before upgrading to 10.3. and 10.3.1
- If you are on r9.2.x or older, upgrade to r9.4 before upgrading to 10.3. and then to 10.3.1

Integration Compatibility Matrix

To know more about version compatibility with other integrated products, see the compatibility matrix chart.

Platform Updates

Third-party Software updates for this release include:

- CAPKI 5.2.5
- Struts 2.3.36
- Apache httpd 2.4.35
- JRE 8.192
- WebSwing 2.5.5
- Cygwin 2.11.2
- Tomcat 9.0.8
- RHEL 7 Container - 7.4
- MYSQL backup - 4.1.2

FAQs

Q: In a distributed SS (DSS) environment, what is the supported and best practice for upgrading servers?

A: Upgrade the Primary MLS (main location server) SpectroSERVER first, followed by the Primary OneClick server, followed by the remaining Primary SpectroSERVERs. It is important that you start the primary MLS first, before starting the other primary SpectroSERVERs. This allows all of the cross landscape and "MOM" (manager of managers) functions to process and complete. Once the primary MLS is running and active, you can start the rest of the primaries.
Upgrade the Secondary SpectroSERVERs last (including all SpectroSERVER and non-primary OneClick servers). Once all SpectroSERVERs have been upgraded, be sure to stop and restart all DX NetOps Spectrum OneClick tomcat processes again to update the tomcat cache with the updated SS info. Not doing this will cause OneClick issues, such as landscapes not showing in the client.

> **NOTE**
>
> In a DSS setup, if you stop a primary SS to test your failover scenario without performing a successful database synchronization (via online backup) first, you may see the landscapes disappear in OneClick. To make sure this does not happen, manually start the online backup and allow the database synchronization to occur. Once complete, you can then test your failover scenario.

Features and Enhancements

DX NetOps Spectrum continues to delight its customers with new features and quality improvements. With the newly introduced release of 10.3.1, our customers can now benefit from the following features:

- About OneClick WebApp (Webswing)
- NCM Capture Using SSH Commands
- VNA Integration
- Monitoring SD-WAN for Viptela and Versa
- DX NetOps Spectrum integration with CA APM (SaaS)
- CA Remote Engineer Tool Enhancements
- Discover Connections Only toward Access Points
- Support for Dynamic VPN (DMVPN)
- SpectrumDataPublisher Enhancements
- DX NetOps Spectrum-CA UIM Integration Enhancements
- Dockerization Enhancements

NOTE

Find out how this release differs from the version you are currently on by referring to the release comparison chart.

Upgrade to 10.3.1 from 10.2.x or 10.3

NOTE
It is recommended that you first install 10.3.0 as a base version, to install 10.3.1. Direct upgrades to 10.3.1 from 10.2.x are also supported. DX NetOps Spectrum users who are on 10.0 or 10.1.x releases are advised to upgrade to the intermediary major release such as 10.2.0 and then directly upgrade to 10.3. to ensure a stable upgrade and migration process devoid of any loss of data or failure of any functionality.

Best Practices

- Best Practices for upgrading in a fault-tolerant and non-fault tolerant environments.
- Preserve Customized Support Files

Pre-Upgrade Tasks

Before you upgrade to the latest version of DX NetOps Spectrum, ensure you:

1. Are familiar with the best practices that are mentioned above.
2. Stop all running applications other than DX NetOps Spectrum.
3. Stop the following DX NetOps Spectrum applications:
   - Shut down all OneClick clients by logging off all users from OneClick on the Client Details web page on the OneClick home page.

     NOTE

     For more information about shutting down OneClick clients, see OneClick Administration.
   - Stop the SpectroSERVER and the Archive Manager by clicking Stop SpectroSERVER in the Spectrum Control Panel and then close the Spectrum Control Panel. Or you can stop the SpectroSERVER and Archive Manager from the command line by running the "<$SPECROOT>/bin/stopSS.pl" as CA Spectrum Owner at the command prompt.
   - Stop all VnmSh connections. For more information about stopping VnmSh connections, see Command Line Interface.
   - Close all Bash shells.
   - Ensure the 'Java' folder is not opened (under <$SPECROOT>) during the upgrade.
4. If you have installed DX NetOps Spectrum Report Manager, verify that free disk space on the system is at least twice the size of the largest MYD file under *$SPECROOT*/mysql/data/reporting.
5. Do not install third-party software that uses MySQL because the results can be unpredictable.

Upgrade Procedure from 10.2.x or 10.3.0

Upgrading from a previous version of DX NetOps Spectrum to the latest one, can be complex, because each release promises higher quality with the latest technologies and services. Ensure that you go through the upgrade options that best suit your requirement.

1. You can perform either:
   - In Place Upgrade (With no fault-tolerant SpectroSERVERs configured in your deployment, a temporary fault-tolerant system is used during the upgrade. Once the temporary system is configured, the primary SpectroSERVER is disabled for an in-place upgrade. Meanwhile, the temporary, secondary SpectroSERVER system takes over core network management duties).
   - Upgrade Models to ensure compatibility between the SpectroSERVER database and a new version of DX NetOps Spectrum after upgrading.

2. Download the release.
   a. Sign in to the support portal.
   b. Click Enterprise Software.
   c. Click Product Downloads.
   d. Enter the product name in the search bar and click the search icon.
   e. Click the link that is displayed after the search is completed.
   f. Select the release to download.
3. Install DX NetOps Spectrum.
4. Refer to the post-installation configurations.

Post Upgrade Tasks

- Archive Manager Database (DDM Database) Tables Conversion: With the previous release of 10.3.0, it is recommended that you convert your Archive Manager database tables from MyISAM to InnoDB for better performance. Refer to this section, at the end of this page, for more information.
- Migration Upgrade (Best Practice - Preserve Customized Support Files)
- Upgrading Models to ensure compatibility between the SpectroSERVER database and a new version of DX NetOps Spectrum after upgrading.

Other Upgrade Tasks

- If you have configured OneClick to launch from Report Manager using SSL, configure this modification again. For more information about this modification, see OneClick Administration.
- If any models display Minor (yellow) alarms with a probable cause of DIFFERENT TYPE MODEL, clear the alarms. To convert all eligible models, run the NewMM.pl script from the CA Spectrum\Install-Tools\PostInstall directory. For more information, see Troubleshooting the Post-Upgrade Installation Script.

Get Started and Upgrade to 10.3.0!

Special Considerations

- This release supports direct upgrades only from 10.2 and 10.2.x.
- We recommend that you upgrade from 10.1.x versions to a supported version of DX NetOps Spectrum by **December 2018**, to avoid 'gold key' expiration issues. For more information and guidance contact CA Support.
- 10.0 is now end of support, we recommend that you upgrade to 10.2 or 10.2.x before upgrading to 10.3.
- If you are on r9.2.x or older, upgrade to r9.4 before upgrading to 10.3.

Integration Compatibility Matrix

To know more about version compatibility with other integrated products, see the compatibility matrix chart.

Platform Support

- 10.3 is supported on Windows 2016. Installation of any component of DX NetOps Spectrum (OneClick, SpectroSERVER, SDC) is supported on this platform and functions as usual.
- 10.3 does not support the following platforms:
  – Solaris
  – BOXI
  – Windows 2008 and
  – RHEL 5.x platforms

Installation on these platforms fails and returns an error. Ensure that you upgrade to the recommended platforms.

Features and Enhancements

DX NetOps Spectrum continues to delight its customers with new features and quality improvements. With the newly introduced release of 10.3.0, our customers can now benefit from the following features:

- DX NetOps Spectrum Dockerization
- DX NetOps Spectrum and CA Digital Operational Intelligence Integration
- Multi-tenant-SRM with Jaspersoft®
- DDMDB Performance Improvements
- Filtering Interfaces
- Support for DHCP
- Enhanced CDP & LLDP Discoveries
- Co-existence of CA VNA and CA UIM Integration for VMWare® Entities
- DX NetOps Spectrum-CA UIM Integration Enhancements
- Modeling Gateway Enhancements
- Syslog Monitoring Enhancements
- Monitoring Profile Enhancements
- API Performance enhancement
- SCOM 2016 Support
- Platform and third-party product updates
- Device Certification Updates

Upgrade to 10.3.0 from 10.2 and 10.2.x

NOTE

DX NetOps Spectrum users who are on 10.0 or 10.1.x releases are advised to upgrade to the intermediary major release such as 10.2.0 and then directly upgrade to 10.3. to ensure a stable upgrade and migration process devoid of any loss of data or failure of any functionality.

Best Practices

- Best Practices for upgrading in a fault-tolerant and non-fault tolerant environments.
- Preserve Customized Support Files

Pre-Upgrade Tasks

Before you upgrade to the latest version of DX NetOps Spectrum, ensure you:

1. Are familiar with the best practices that are mentioned above.
2. Stop all running applications other than DX NetOps Spectrum.
3. Stop the following DX NetOps Spectrum applications:
   – Shut down all OneClick clients by logging off all users from OneClick in the Client Details web page in the OneClick home page.

   NOTE
   For more information about shutting down OneClick clients, see OneClick Administration.

- Stop the SpectroSERVER and the Archive Manager by clicking Stop SpectroSERVER in the Spectrum Control Panel and then close the Spectrum Control Panel. Or you can stop the SpectroSERVER and Archive Manager from the command line by running the "<$SPECROOT>/bin/stopSS.pl" as owner at the command prompt.
- Stop all VnmSh connections. For more information about stopping VnmSh connections, see Command Line Interface.
- Close all Bash shells.
- Ensure the 'Java' folder is not opened (under <$SPECROOT>) during the upgrade.

4. If you have installed DX NetOps Spectrum Report Manager, verify that free disk space on the system is at least twice the size of the largest MYD file under *$SPECROOT*/mysql/data/reporting.
5. Do not install third-party software that uses MySQL because the results can be unpredictable.

Upgrade Procedure from 10.2 or 10.2.x

Upgrading from a previous version of DX NetOps Spectrum to the latest one, can be complex, because each release promises higher quality with the latest technologies and services. Ensure that you go through the upgrade options that best suit your requirement.

1. You can perform either:
   - In Place Upgrade (With no fault-tolerant SpectroSERVERs configured in your deployment, a temporary fault-tolerant system is used during the upgrade. Once the temporary system is configured, the primary SpectroSERVER is disabled for an in-place upgrade. Meanwhile, the temporary, secondary SpectroSERVER system takes over core network management duties).
   - Upgrade Models to ensure compatibility between the SpectroSERVER database and a new version of DX NetOps Spectrum after upgrading.
2. Download the release.
   a. Sign in to the support portal.
   b. Click Enterprise Software.
   c. Click Product Downloads.
   d. Enter the product name in the search bar and click the search icon.
   e. Click the link that is displayed after the search is completed.
   f. Select the release to download.
3. Install DX NetOps Spectrum.
4. Refer to the post-installation configurations.

Post Upgrade Tasks

- Archive Manager Database (DDM Database) Tables Conversion: With the new 10.3.0 release, it is recommended that you convert your Archive Manager database tables from MyISAM to InnoDB for better performance.

  NOTE

  It is recommended that you perform the Archive Manager tables conversion as a post-upgrade task. While performing this post-upgrade task, the Archive Manager stops, to convert the tables. The conversion time depends on the size of the tables. Do not start the Archive Manager when executing the script.

  After conversion, the InnoDB tables occupy more space (approximately 30%-40%) compared to the MyISAM tables. MYSQL recommends InnoDB to be an efficient engine to work on large-scale tables. The InnoDB engine supports row-level locking and auto crash recovery features, which are not supported by MyISAM.

  The InnoDB engine allows for performance improvements with DX NetOps Spectrum's MYSQL; however, it has been observed that after converting the MYSQL DB from MyISAM to InnoDB, the MYSQL DB grew 40 percent in disk space.

Steps for Archive Manager Table Conversion:

Prerequisite:

1. a. a. Ensure that there is enough free disk space. The space that is required, is three times greater than the event table size. To convert the Archive Manager database tables from MyISAM to InnoDB, follow these steps:

Procedure:

1. a. a. Open any bash-shell and navigate to $SPECROOT/SS/DDM/scripts.

NOTE

- Before running the script, it is recommended to take a back-up of the Archive Manager database as a precaution against any potential loss of data.
- If your DDM database contains less than 10 days of data, refer to the KB000115500 article for instructions to perform a successful data conversion.

b. To convert the DDM database tables engine from MyISAM to InnoDB, run the following script:

```
convert_current_myisam_to_innodb.pl
```

c. After the conversion of tables, you are prompted to start the Archive Manager. Enter 'Y' to start the Archive Manager.

• Migration Upgrade (Best Practice - Preserve Customized Support Files)

• Upgrading Models to ensure compatibility between the SpectroSERVER database and a new version of DX NetOps Spectrum after upgrading.

Other Upgrade Tasks

- If you have configured OneClick to launch from Report Manager using SSL, configure this modification again. For more information about this modification, see OneClick Administration.
- If any models display Minor (yellow) alarms with a probable cause of DIFFERENT TYPE MODEL, clear the alarms. To convert all eligible models, run the NewMM.pl script from the CA Spectrum\Install-Tools\PostInstall directory. For more information, see Troubleshooting the Post-Upgrade Installation Script.

Upgrade Best Practices DSS Deployments without Fault Tolerance

NOTE

The topic titled Upgrade Best Practices Fault-Tolerant Deployments provides the steps to upgrade DX NetOps Spectrum in a fault-tolerant environment.

Upgrade Best Practices DSS Deployments without Fault Tolerance

With no fault-tolerant SpectroSERVERs configured in your deployment, a temporary fault-tolerant system is used during the upgrade. Once the temporary system is configured, the primary SpectroSERVER is disabled for an in-place upgrade. Meanwhile, the temporary, secondary SpectroSERVER system takes over core network management duties.

Some DX NetOps Spectrum applications do not support automatic failover and are disabled during the upgrade. For example, TL1, Southbound Gateway, Modeling Gateway, Alarm Notifier, and Event Notifier are temporarily disabled during the upgrade.

FAQs

Q: In a distributed SS (DSS) environment, what is the supported and best practice for upgrading servers?

A: Upgrade the Primary MLS (main location server) SpectroSERVER first, followed by the Primary OneClick server, followed by the remaining Primary SpectroSERVERs. It is important that you start the primary MLS first, before starting

the other primary SpectroSERVERs. This allows all of the cross landscape and "MOM" (manager of managers) functions to process and complete. Once the primary MLS is running and active, you can start the rest of the primaries. Upgrade the Secondary SpectroSERVERs last (including all SpectroSERVER and non-primary OneClick servers). Once all SpectroSERVERs have been upgraded, be sure to stop and restart all OneClick tomcat processes again to update the tomcat cache with the updated SS info. Not doing this will cause OneClick issues, such as landscapes not showing in the client.

NOTE

In a DSS setup, if you stop a primary SS to test your failover scenario without performing a successful database synchronization (via online backup) first, you may see the landscapes disappear in OneClick. To make sure this does not happen, manually start the online backup and allow the database synchronization to occur. Once complete, you can then test your failover scenario.

Upgrading SpectroSERVERs and OneClick Web Servers in a Non-Fault Tolerant Deployment

When you upgrade a DX NetOps Spectrum DSS environment that lacks a fault-tolerant configuration, deploy a temporary server to preserve network monitoring activities. This server is systematically configured as the Secondary, fault-tolerant server for each SpectroSERVER that is upgraded. Start the upgrade with the MLS, the main SpectroSERVER.

The procedure to perform an in-place upgrade with no network management loss is described below.

Follow these steps:

1. Designate a server to serve as the temporary Secondary SpectroSERVER.
2. Install a copy of your currently installed (that is, backlevel) DX NetOps Spectrum software on the temporary SpectroSERVER. Be sure to install all the required patches.

   WARNING
   Do not start the SpectroSERVER yet.
3. Edit the Host Security configuration on the temporary SpectroSERVER. The list of hostnames must be identical to that of the MLS in your current deployment.
4. Perform an online backup of the SpectroSERVER database on the MLS.

   WARNING
   Be sure to disable file compression and automatic backup features until the entire upgrade process has completed.
5. Copy the backup database to the *$SPECROOT*/SS directory on the temporary SpectroSERVER.
6. On the temporary SpectroSERVER host, navigate to the SS directory.
7. Load the database backup file by issuing the following command:

   ```
   ../SS-Tools/SSdbload -il -add precedence savefile
   ```

   – **precedence**
   A numeric value that is greater than the Primary SpectroSERVER (the MLS) default value.
   Default: 10 (20 is recommended).
   – **savefile**
   The name of the database backup file that you created previously.
8. Edit the .vnmrc file to increase the 'maximum event records' parameter on the temporary SpectroSERVER. For example, change the following parameter:

   ```
   max_event_records=20000
   ```

   to the following value:

   ```
   max_event_records=200000
   ```

   The new value ensures that no events are lost during the upgrade.
9. Disable the Archive Manager on the Secondary SpectroSERVER from starting automatically to avoid losing event and statistical data.

Taking this step ensures that all data is cached and returned to the Primary SpectroSERVER once the upgrade has completed and the Primary SpectroSERVER has been restarted.

a. On the temporary Secondary SpectroSERVER, launch a Spectrum Control Panel.
b. Click Control, and clear the box next to “Auto Start/Stop Archive Manager.”

NOTE
As a best practice, ensure that no events are lost during the upgrade. You can increase the maximum locally stored event record size. The default maximum locally stored log sizes for events and statistics are 20,000 and 5,000. In most cases, these default settings are sufficient.

10. Start the SpectroSERVER on the temporary SpectroSERVER host. The temporary SpectroSERVER is now the Secondary, fault-tolerant SpectroSERVER for the MLS.
    An orange alarm on the VNM indicates that the Archive Manager is not running. You can ignore it.
11. Verify the setup of the Secondary fault-tolerant SpectroSERVER by checking the Landscape Configuration view on the MLS:
    a. In OneClick, double-click the VNM icon in the Universe Topology view. The landscape container is displayed.
    b. In the Contents panel highlight the “LocalLscpe” model.
    c. In the Component Detail panel, select the Information tab.
    d. Locate and expand the “Loaded Landscapes” subview.
    e. Verify that the list contains both the Primary MLS, with a precedence of 10, and the temporary Secondary SpectroSERVER, with a precedence of 20 (or the precedence value that was specified with the “SSdbload” command).
12. Shut down the Primary SpectroSERVER (the MLS).
    The Secondary SpectroSERVER resumes management tasks while the MLS is upgraded.
13. Follow the steps that are listed in How to Perform In-Place Upgrades to upgrade the MLS.
14. Once the MLS has been successfully upgraded, manually start the Archive Manager on the Primary SpectroSERVER:
    a. Launch a Spectrum Control Panel.
    b. Select Control, and click Start Archive Manager.

NOTE
Starting the Archive Manager ensures that the events that are being stored locally on the Secondary SpectroSERVER are sent over to the Primary Archive Manager.

15. Start the SpectroSERVER.
    Primary management functions switch back to the MLS.
16. Configure the temporary SpectroSERVER host as a Secondary SpectroSERVER for the next SpectroSERVER that you plan to upgrade.
17. Repeat the above steps to back up, shut down, and upgrade each SpectroSERVER in turn.
18. Upgrade the OneClick Web Server last.
19. Review the post-installation steps in Post-Installation Configurations.

Migrating and Upgrading

WARNING

- For upgrade and migration scenarios, DX NetOps Spectrum automatically uses Legacy Landscape Handle type. All pre-10.2 landscape handles belong to Legacy Landscape Handle type.
- However, migration is supported for Huge Landscape Handle type within 10.2, that is, if you have selected Huge Landscape handle type during a fresh 10.2 installation and are migrating 10.2 SSdbs from one server to another.

Migrate and Upgrade on Windows

You can migrate the existing SpectroSERVER database and Archive Manager database and other upgradeable components to a different system, and then upgrade DX NetOps Spectrum. This way, you can continue to manage your network with the existing DX NetOps Spectrum version during the installation process.

> **NOTE**
> You cannot move a DX NetOps Spectrum installation from one system to another or from one directory to another. Instead, first, copy or move the DX NetOps Spectrum databases and then run the installation program to reinstall DX NetOps Spectrum over the relocated database.

> **WARNING**
> The C:\Program Files\CA directory is automatically created during a first-time installation of DX NetOps Spectrum. DX NetOps Spectrum components that are also common to other CA products are intentionally installed into this directory. This directory is automatically updated as needed during a DX NetOps Spectrum upgrade. Do not remove files from this directory.

Follow these steps:

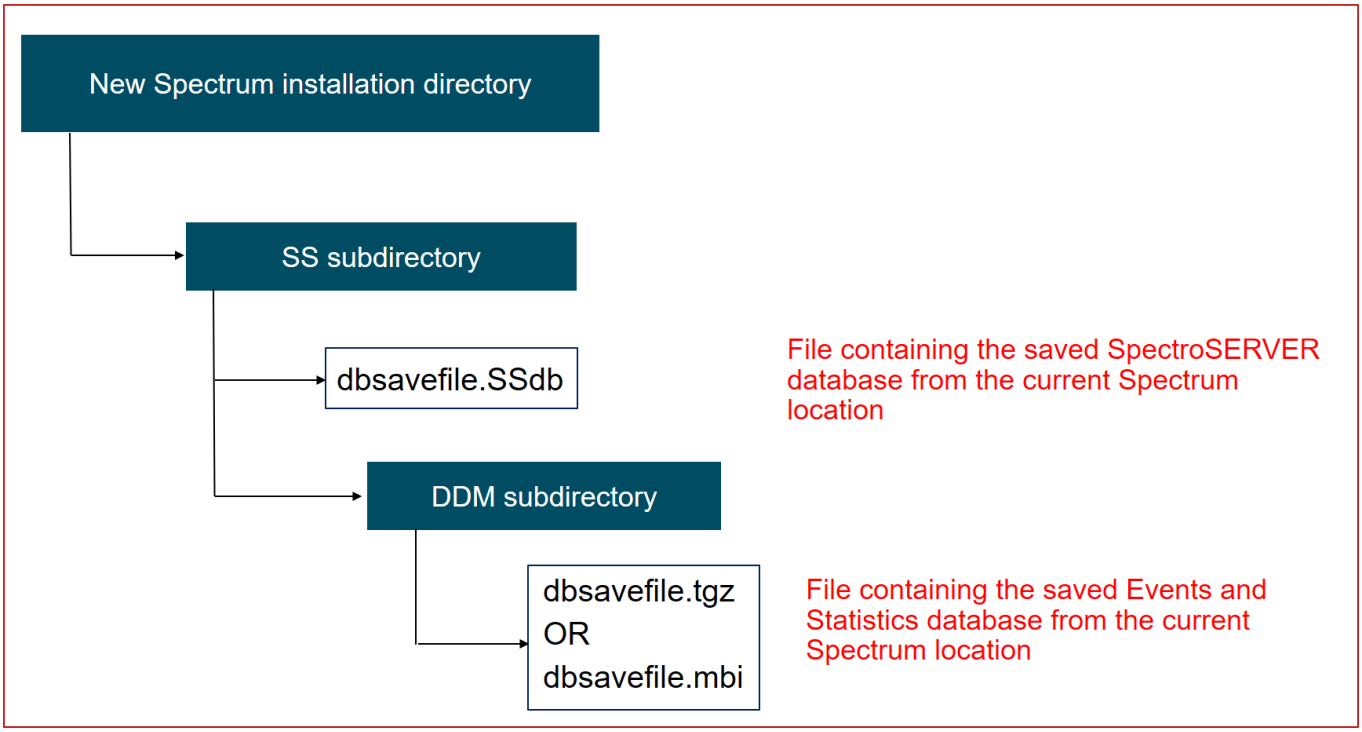
1. Create a user model from the OneClick Users tab. Name it the same name as the owner of the directory where you are installing the new DX NetOps Spectrum version.
2. Preserve the existing SpectroSERVER database (see the topic 'Preserve the existing SpectroSERVER database' on How to Perform In-Place Upgrades page).

   > **WARNING**
   > When backing up the SpectroSERVER database for migration, include both the modeling catalog and the models. A catalog-only or models-only migration is not supported.
3. Preserve the DX NetOps Spectrum Events and Statistics database (see the topic 'Preserve the DX NetOps Spectrum Events and Statistics database' on How to Perform In-Place Upgrades page).
4. Extract the dbsavefile.SSdb file from the dbsavefile.SSdb.gz file in the SS-DB-Backup directory.
5. Copy the saved SSdb file to the dbsavefile.SSdb file. If the SSdb file had backup compression that is enabled, uncompress the SSdb file by running gzip -d <*database_file.gz*> and then perform the copy. For example, cp db_20080105_1153.SSdb dbsavefile.SSdb.

   > **WARNING**
   > The dbsavefile.SSdb file must not be compressed. If dbsavefile.SSdb is compressed, the database is not migrated during installation.
6. Create an installation directory, <*$SPECROOT*>, for the new version of DX NetOps Spectrum.

   > **WARNING**
   > Ensure that the folder in which the installation kit is copied and extracted does not have any spaces. DX NetOps Spectrum installation fails if the installer (.exe) file is in a location with spaces in its absolute/full path. You cannot install DX NetOps Spectrum into a directory that contains a space anywhere in the path. Spaces within the directory path cause the installation to fail.
7. Create an SS subdirectory in *$SPECROOT*.
8. Copy the custom directory from the earlier DX NetOps Spectrum installation directory to the *$SPECROOT* directory of the upgraded DX NetOps Spectrum installation.
9. Copy or FTP (in binary mode) the dbsavefile.SSdb file to the *$SPECROOT*\SS subdirectory.

   > **NOTE**
   > Migrating an existing SpectroSERVER database migrates all of the existing models into the new SpectroSERVER database when you install a new version of DX NetOps Spectrum. This migration includes the models containing topology views including icon placement, groupings, and annotations.
10. Create a DDM subdirectory in the SS subdirectory.
11. Copy or FTP (in binary mode) dbsavefile.mbi or dbsavefile.tgz to the new *$SPECROOT*\SS\DDM directory.
    The directory structure now resembles the following structure:



12. (Optional) To migrate the OneClick web server, copy the contents of the <*$SPECROOT*>\custom directories from the computer with the old OneClick web server installation. Paste these contents into the <*$SPECROOT*>\custom directories on the computer with the new OneClick web server installation.

    WARNING
    Do *not* copy the <*$SPECROOT*>/custom/common/config/custom-jnlp-config.xml file to another computer when you migrate and upgrade DX NetOps Spectrum. This file can contain memory settings that are not compatible with the computer where you are copying the custom directories.

    NOTE
    The mapping of custom background images (<*$SPECROOT*>\custom\images\background) to topology views is maintained in the SpectroSERVER database. For more information about the <*$SPECROOT*>\custom directories, see the OneClick Customization .

DX NetOps Spectrum is migrated and upgraded.

Migrate and Upgrade on Linux

You can migrate existing DX NetOps Spectrum databases and other upgradeable components to a different system, and then upgrade DX NetOps Spectrum. This way, you can continue to manage your network with the existing DX NetOps Spectrum version during the installation process.

NOTE
Do not move a DX NetOps Spectrum installation from one system to another or from one directory to another. Instead, first, copy or move the DX NetOps Spectrum database and then run the installation program to reinstall DX NetOps Spectrum over the relocated database.

WARNING
The /opt/CA directory is automatically created during a first-time installation of DX NetOps Spectrum. DX NetOps Spectrum components that are also common to other CA products are intentionally installed into this directory. This directory is automatically updated as needed during a DX NetOps Spectrum upgrade. Do not remove files from this directory.

Follow these steps:

1. Create a user model from the OneClick Users tab. Name it the same name as the owner of the directory where you are installing the new DX NetOps Spectrum version.
2. Preserve the existing SpectroSERVER database (see the topic 'Preserve the existing SpectroSERVER database' on How to Perform In-Place Upgrades page).

   WARNING
   When backing up the SpectroSERVER database for migration, include both the modeling catalog and the models. A catalog-only or models-only migration is not supported.
3. Preserve the DX NetOps Spectrum Events and Statistics database (see the topic 'Preserve the DX NetOps Spectrum Events and Statistics database' on How to Perform In-Place Upgrades page).
4. Extract the dbsavefile.SSdb file from the dbsavefile.SSdb.gz file in the SS-DB-Backup directory.
5. Copy the saved SSdb file to the dbsavefile.SSdb file. If SSdb had backup compression that is enabled, first uncompress the SSdb by running gzip -d <*database_file.gz*> and then perform the copy. For example, cp db_20080105_1153.SSdb dbsavefile.SSdb.

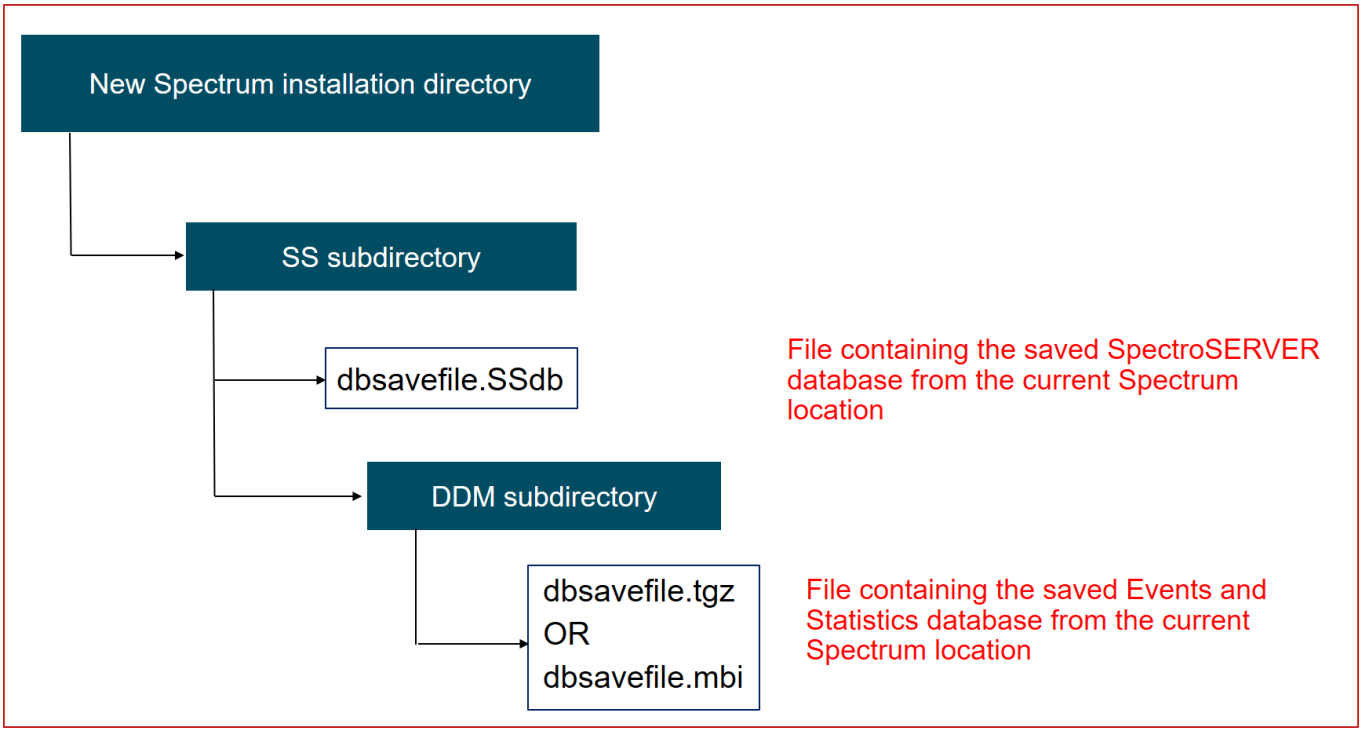
   WARNING
   The dbsavefile.SSdb file must not be compressed. If dbsavefile.SSdb is compressed, the database is not migrated during installation.
6. Create an installation directory, <*$SPECROOT*>, for the new version of DX NetOps Spectrum, with subdirectories for the two database files as follows:

   ```
   mkdir -p <$SPECROOT>/SS/DDM
   ```

   WARNING
   Do not use /opt/SPECTRUM as the installation directory name. This location is reserved for a directory that is automatically created during installation. Also, ensure that the directory owner name is the same name as the owner of the directory where you are installing the new DX NetOps Spectrum version.

   WARNING
   You cannot install DX NetOps Spectrum into a directory that contains a space anywhere in the path. Spaces within the directory path cause the installation to fail.
7. Copy or FTP (in binary mode) the dbsavefile.SSdb file to the <*$SPECROOT*>/SS directory.

   NOTE
   Migrating an existing SpectroSERVER database migrates all the existing models into the new SpectroSERVER database when you install a new version of DX NetOps Spectrum. This migration includes the models containing topology views including icon placement, groupings, and annotations.
8. Copy or FTP (in binary mode) the dbsavefile.mbi or dbsavefile.tgz file to the <*$SPECROOT*>/SS/DDM directory. The directory structure now resembles the following structure:



9. (Optional) To migrate the OneClick web server, copy the contents of the <*$SPECROOT*>/custom directories from the computer with the old OneClick web server installation. Paste these contents into the <*$SPECROOT*>/custom directories on the computer with the new OneClick web server installation.

   WARNING
   Do *not* copy the <*$SPECROOT*>/custom/common/config/custom-jnlp-config.xml file to another computer when you migrate and upgrade DX NetOps Spectrum. This file can contain memory settings that are not compatible with the computer where you are copying the custom directories.

   NOTE
   The mapping of custom background images (<*$SPECROOT*>/custom/images/background) to topology views is maintained in the SpectroSERVER database. For more information about the <*$SPECROOT*>/custom directories, see the OneClick Customization .

10. Fresh Install.

DX NetOps Spectrum is migrated and upgraded.

Post Migration Tasks

NOTE

It is recommended that you perform a DDM database table conversion as a post-migration task. However, ignore this task if the DDM database is already converted to InnoDB.

Archive Manager Table Conversion

While performing the post-migration task, the Archive Manager stops for some time because of the time taken by the script to convert the tables. This conversion time depends on the size of the tables. To improve the performance of Archive Manager it is recommended you convert the Archive Manager database tables MySQL engine from MyISAM to InnoDB. To enhance the performance of event retrieval, the event table is partitioned.

Prerequisite:

Ensure that there is a free disk space which is three times greater than the event table size.

Procedure:

1. Open any bash- shell and navigate to the **$specroot/SS/DDM/scripts** location.

   NOTE

   - Before running the script, it is recommended to take a back-up of the Archive Manager database as a precaution against any potential loss of data.
   - If your DDM database contains less than 10 days of data, refer to the KB000115500 article for instructions to perform a successful data conversion.
2. To convert the DDM database tables engine from MyISAM to InnoDB, run the following script:
   `convert_current_myisam_to_innodb.pl`
3. After the conversion of tables, you are prompted to start the Archive Manager. Enter 'Y' to start the Archive Manager.

Support for Migrating DDM .mbi File (InnoDB)

10.4.1 provides support for migrating the DDM .mbi file from a previous release to this release. With the availability of this support, we recommend that you use InnoDB for DDM and use ddm_load.pl and ddm_save.pl for backup and restore. The ddm_load.pl and ddm_save.pl (<SPECROOT>\SS\DDM) leverage mysqlbackup.exe, which is more efficient to take backups and restore.

Considerations

- Migration of DDM .mbi file is not supported from a Windows computer to a Linux computer.
- The ddm_load, ddm_save, and ddm_backup binaries coming with the previous releases have been deprecated in 10.4.1. You can now use ddm_load.pl instead of ddm_load and ddm_save.pl instead of ddm_save and ddm_backup.

Upgrade Best Practices Fault-Tolerant Deployments

Upgrade Best Practices Fault-Tolerant Deployments

Upgrades in a fault-tolerant environment are supported. However, follow the recommendations in this topic. All your Primary and Secondary SpectroSERVERs must be upgraded within a small window of a few hours.

Before conducting any upgrades, verify that all DX NetOps Spectrum components are up-to-date with current service packs. Also, consult the list of the communication ports and protocols that DX NetOps Spectrum uses. Your security parameters may require you to temporarily disable these ports and protocols during the upgrade. For more information about firewall ports and protocols, see Distributed SpectroSERVER Administration.

First, upgrade the OneClick Web Servers, after that upgrade the primary SpectroSERVERs, and finally upgrade the secondary SpectroSERVERs.

FAQs

Q: In a distributed SS (DSS) environment, what is the supported and best practice for upgrading servers?

A: Upgrade the Primary MLS (main location server) SpectroSERVER first, followed by the Primary OneClick server, followed by the remaining Primary SpectroSERVERs. It is important that you start the primary MLS first, before starting the other primary SpectroSERVERs. This allows all of the cross landscape and "MOM" (manager of managers) functions to process and complete. Once the primary MLS is running and active, you can start the rest of the primaries. Upgrade the Secondary SpectroSERVERs last (including all SpectroSERVER and non-primary OneClick servers). Once all SpectroSERVERs have been upgraded, be sure to stop and restart all OneClick tomcat processes again to update the tomcat cache with the updated SS info. Not doing this will cause OneClick issues, such as landscapes not showing in the client.

NOTE

In a DSS setup, if you stop a primary SS to test your failover scenario without performing a successful database synchronization (via online backup) first, you may see the landscapes disappear in OneClick. To make sure this does not happen, manually start the online backup and allow the database synchronization to occur. Once complete, you can then test your failover scenario.

Upgrade in a Fault-Tolerant Environment

The following procedure describes an upgrade in a fault-tolerant environment. Use the same procedure for a single-server or DSS upgrade. Upgrade the OneClick Web Servers first, and then upgrade primary SpectroSERVERs. Upgrade Secondary SpectroSERVERs at last.

NOTE
Alarms may display incorrectly in OneClick until the upgrade process is complete, including any post-upgrade steps that may be required. Avoid using DX NetOps Spectrum for management until you have completed all the steps.

Follow these steps:

1. Disable automatic online backups on the Primary SpectroSERVERs by taking the following steps:
   a. Highlight the VNM model in the Universe Topology view.
   b. In the Component Detail panel, select the Information tab.
   c. Locate and expand the Online Database Backup subview.
   d. Set Automatic Backups to “Disabled”.
2. Perform a manual online backup on every Primary SpectroSERVER to preserve the current database. Take the following steps:
   a. In the Online Database Backup subview, click Begin Backup Now.
   b. Verify that the online backup has succeeded.

      NOTE
      We recommend differentiating the databases by the version number. Set the “Prefix for Backup File Name” parameter from the default of “db_” to something like “db_*version*_” or another value that identifies the version.
3. Edit the .vnmrc file to increase the 'maximum event records' parameter on all Secondary SpectroSERVERs. For example, change the following parameter:

   ```
   max_event_records=20000
   ```

   to the following value:

   ```
   max_event_records=200000
   ```

   The new value prevents event loss during the upgrade.
4. Restart all Secondary SpectroSERVERs so that the change takes effect:
   a. Launch a Spectrum Control Panel.
   b. Click Stop SpectroSERVER.
   c. Once the Status changes to “INACTIVE,” click Start SpectroSERVER.
   d. Verify that the Status changes to “RUNNING”.
5. Instruct all OneClick users who are logged in to the OneClick Web Servers to close their clients.
6. Stop each of the Primary SpectroSERVERs:
   a. Launch a Spectrum Control Panel
   b. Click Stop SpectroSERVER.
   c. You are prompted to verify the action to stop the SpectroSERVER and Archive Manager.
   d. Verify that the Status changes to “INACTIVE”.
   e. Exit the Spectrum Control Panel.

7. On the OneClick clients, verify that the SpectroSERVERs have failed over to the Secondary SpectroSERVERs.
   The landscape icons in the Explorer hierarchy panel change from green to yellow
   Yellow borders appear around the Contents and Component Detail panels.
8. Stop all OneClick Web Servers using the following command:
   ```
   $SPECROOT/tomcat/bin/stopTomcat.sh
   ```
9. Upgrade all OneClick web servers.
10. Upgrade all the Primary SpectroSERVERs, and verify that the installations complete successfully.

    NOTE
    Wait for every installation to complete before continuing to the next step.

    NOTE
    It is no longer necessary to start the Archive Manager before the SpectroSERVER, the cached events from the secondary SpectroSERVER can be transferred at any time, even after the primary SpectroSERVER has started logging new events.
11. Start the SpectroSERVER on the Primary MLS:
    a. Launch a Spectrum Control Panel on the Primary MLS.
    b. Click Start SpectroSERVER.
    c. Verify that the Status changes to “RUNNING”.
12. Start the remaining Primary SpectroSERVERs.
13. Stop each of the secondary SpectroSERVERs.
14. Upgrade all the Secondary SpectroSERVERs, and verify that the installations complete successfully.

    NOTE
    Wait for every installation to complete before continuing to the next step.
15. Start the SpectroSERVER on every Secondary SpectroSERVER:
    a. Launch a Spectrum Control Panel on the SpectroSERVER.
    b. Click Start SpectroSERVER.
    c. Verify that the Status changes to “RUNNING”.
16. Verify that the clients open successfully and that the Connection Status all the Primary SpectroSERVERs are up and green.
17. Perform a manual online backup on every Primary SpectroSERVER to preserve the current database. Take the following steps:
    a. In the Online Database Backup subview, click Begin Backup Now.
    b. Verify that the online backup has succeeded.
    c. Verify that the re-synchronizing the Primary SpectroSERVER database with the Secondary SpectroSERVER is succeeded.

       NOTE
       We recommend differentiating the databases by the version number. Set the “Prefix for Backup File Name” parameter from the default of “db_” to something like “db_*version*_” or another value that identifies the version.

    To continue management activities during the upgrade process, do not upgrade all the OneClick web servers at one time. Retain few OneClick web servers, so that you can monitor your infrastructure using the secondary SpectroSERVERs until all the primary SpectroSERVERs are upgraded. When all primary SpectroSERVERs are upgraded, you can upgrade the retained OneClick web servers.

How to Perform In-Place Upgrades

If you have DX NetOps Spectrum data that you want to preserve, perform an *in-place* upgrade. An in-place upgrade installs a new version of DX NetOps Spectrum on an earlier version in the same system and directory. In-place upgrades do not require a database migration. For an upgrade that changes the server platform and requires data migration, see the topics 'Migrate and Upgrade on Windows and Linux' on Migrating and Upgrading DX NetOps Spectrum page.

To perform an in-place upgrade:

1. Complete all pre-upgrade tasks.
2. Preserve your existing SpectroSERVER database.
3. Preserve the DX NetOps Spectrum events and statistics database.
4. Install DX NetOps Spectrum on top of an earlier version of DX NetOps Spectrum.

This section provides procedures to upgrade from a previous version, to upgrade in a fault-tolerant environment, and to upgrade in an environment that lacks fault tolerance.

> **NOTE**
> Additional steps are required for preserving the DX NetOps Spectrum databases and upgradeable components that are part of the older version.

Complete pre-upgrade tasks

Complete the following tasks before the upgrade.
Stop the following DX NetOps Spectrum applications:

1. Shut down all OneClick clients by logging off all users from OneClick. This can be found under the Client Details web page in the OneClick home page.

   > **NOTE**
   > For more information about shutting down OneClick clients, see OneClick Administration.
2. Stop all VnmSh connections.

   > **NOTE**
   > For more information about stopping VnmSh connections, see Command Line Interface.
3. Close all Bash shells.

Preserve the Existing SpectroSERVER Database

You can preserve an existing SpectroSERVER database before performing an in-place upgrade.

Follow these steps:

1. Verify that the SpectroSERVER is running, and open the DX NetOps Spectrum Control Panel.
2. Make a copy of your current SpectroSERVER database by clicking Save Database in the Spectrum Control Panel.
   The Online Database Backup dialog opens.
3. Verify that the option to Use Backup Compression is selected.
4. Accept the default or enter a directory path in the Backup Directory field.
   The default directory path is as follows:
   - Windows -- C:/win32app/SPECTRUM/SS-DB-Backup
   - Linux -- /usr/SPECTRUM/SS-DB-Backup

   > **NOTE**
   > If you change the default, select a directory other than the DX NetOps Spectrum installation directory.
5. Accept the default of 20 for the Minimum Required Disk Space or enter an appropriate value.
6. Select Save to save all changes.
7. Click Begin Backup Now.
   The Status displays the progress of the backup.

DX NetOps Spectrum automatically assigns a name for the backup with .SSdb extension in the format db_YYYYMMDD_HHMM. The YYYYMMDD represents the year, month, and day, and HHMM represents the hour and minute when the backup started. For example, a backup that started at 10:42 on 10/06/06 is named db_20061006_1042.SSdb.

NOTE
Because compression was enabled, this file is compressed into a file with a .gz extension.

8. Click Save and Close.
   The database is backed up.
9. Move the database to an area outside the DX NetOps Spectrum installation directory.
   The existing SpectroSERVER database is preserved.

Preserve the DX NetOps Spectrum Events and Statistics Database

You can preserve the DX NetOps Spectrum events and statistics database before upgrading DX NetOps Spectrum.

Follow these steps:

1. Stop the SpectroSERVER and the Archive Manager by clicking Stop SpectroSERVER in the Spectrum Control Panel and then close the Spectrum Control Panel. Or you can stop the SpectroSERVER and Archive Manager from the command line by running the "<$SPECROOT>/bin/stopSS.pl" as Spectrum Owner at the command prompt.
2. Execute cd <*$SPECROOT*>\SS\DDM in the command prompt, where <*$SPECROOT*> is the directory where DX NetOps Spectrum was installed.
3. Enter the following command:
   ```
   ddm_save dbsavefile
   ```
   The file dbsavefile.tgz is created.
4. Move the dbsavefile.tgz to an area other than the DX NetOps Spectrum installation directory. The existing DX NetOps Spectrum events and statistics database is preserved.

Install DX NetOps Spectrum

For installation instructions, see Fresh Install.

NOTE

We recommend that you back up the new, upgraded SpectroSERVER database using the SSdbsave utility with the -cm option before starting SpectroSERVER. This utility is located in the SS-Tools directory. Backing up the new SpectroSERVER database ensures the integrity of the database, in case the new SpectroSERVER fails before you access the Online Database Backup.

Use the following command (navigate to the $SPECROOT/SS folder) to save the modeling catalog and models:

```
../SS-Tools/SSdbsave -cm <save_file>
```

The suggested naming scheme for saving files is to incorporate a date stamp and the flags for the save command. In this case, you are using both the “c” (catalog) and “m” (models) flags; therefore, assuming it is September 14, 2016, enter db_20160914_cm as the save file name.
A file named “db_20160914_cm.SSdb” is created. The “.SSdb” suffix is added automatically.

Upgrading Models

Database Compatibility After Upgrade

To ensure compatibility between the SpectroSERVER database and a new version of DX NetOps Spectrum after upgrading, complete these tasks:

- Convert existing models that are based on defunct model types to new models.
- Convert existing models to model types that are more appropriate.

These procedures are not required for first-time installations.

> **WARNING**
> If you do not run appropriate upgrade scripts after a DX NetOps Spectrum upgrade, system problems can occur.

In some cases, a model type can change, depending on vendor requirements or added functionality in DX NetOps Spectrum. In other cases, DX NetOps Spectrum no longer supports a device with a unique model type; therefore, convert these models to an alternative model type.

Contact a support representative if you have questions about the model conversion process or any errors you encounter during conversion.

> **NOTE**
> If you plan to set up a distributed SpectroSERVER configuration, convert all models before partitioning the database.

Preserved Model Attributes and Elements

The following model attributes are preserved when you use the listed scripts to upgrade the SpectroSERVER database:

- 0x1006e Model_Name
- 0x12d7f Network_Address
- 0x10024 Community_Name
- 0x10009 Security_String
- 0x11564 Notes (Notes are preserved for the device, interface, application, module, and port models.)
- 0x10071 Polling_Interval
- 0x10072 Poll_Log_Ratio
- 0x1154f Polling Status
- 0x110c4 Time Out
- 0x110c5 Try Count
- 0x1000c Value_When_Yellow
- 0x1000d Value_When_Orange
- 0x1000e Value_When_Red

The following details are also preserved:

- Inter-model relations, including device connectivity
- Connections to both physical and logical interfaces on all devices
- Model type-specific and model-specific NCM configurations

During remodeling, interfaces and applications are rediscovered and modeled. This remodeling results in new model handles for these child models.

> **NOTE**
> Watches are not preserved during the model and model type conversion and must be rebuilt on the new model type.

Model Type Editor and the Customized SpectroSERVER Database

If you customized your SpectroSERVER database using the Model Type Editor (MTE), make a record of all changes. Certain changes that are made with the MTE are not preserved when the SpectroSERVER database is upgraded to a later version of DX NetOps Spectrum.

If you changed relations, meta-rules, or attributes of DX NetOps Spectrum-supplied or other developer-supplied model types, those changes could be unrecognized during the database upgrade. Reapply the changes manually after you upgrade DX NetOps Spectrum.

Model types can be changed and improved in the upgraded version of DX NetOps Spectrum. For the new release of DX NetOps Spectrum to operate correctly, these changes might need to overwrite customized values.

NOTE
For more information about preserving database changes and the type of changes that can be preserved, see the Model Type Editor .

Using the Multicast Manager or VPN Manager After Installing a Patch or Upgrade

The NewMM.pl post-installation script affects the following model types:

- Rtr_Cisco
- Cisco_12000
- SwCat6xxx, SwCat35xx, and SwCat4xxx

Rerun Multicast and/or VPN discovery and reapply customizations after you run the post-installation scripts. This process helps ensure the correct modeling and management of the newly created device models within your environment.

Convert Existing Models to Newly-Supported Model Types

Use the NewMM.pl post-installation script to convert the existing models of various model types to the newly supported model types. This script preserves many key attributes, relationships, and other elements.

For example, you previously modeled Cisco Catalyst 4500 devices as GnSNMPDev in DX NetOps Spectrum. These models can be converted to use the Catalyst 4500 Certification functionality.

In addition, you can use the NewMM.pl script to convert various Cisco-specific model types to the appropriate supported model type. As Cisco introduces new devices, DX NetOps Spectrum adds support for these new devices using the appropriate model type available.

NOTE
If you update model types using the NewMM.pl script, a set of models is created in the Reporting Database with a new model type. Models with the previous model type are marked as destroyed. In addition, data is not migrated from the old model type to the new type.

Follow these steps:

1. Verify that the SpectroSERVER is running.
2. Run the following command from the *$SPECROOT*/Install-Tools/PostInstall/ directory:
   ```
   NewMM.pl
   ```
   NOTE
   On Windows, all necessary scripts must be run from a bash shell. They do not run as expected from a DOS command prompt.
3. Enter the host name or IP address of the VNM and press Enter.
4. Enter the SpectroSERVER landscape handle when prompted, and press Enter.
   The script analyzes the database to determine which models are eligible for conversion, if any. The script provides a complete list of models that correspond to each new model type before prompted for conversion.

NOTE
Models that are in maintenance or hibernation mode or that cannot be contacted are not candidates for conversion.

5. When prompted to convert eligible models of a specific model type, enter Yes. If you do not want to convert specific model types, enter No. The following log file is created in the *$SPECROOT*/Install-Tools/PostInstall/ directory:
   `NewMM_Log_DATE`
6. To confirm the model conversion, verify the following log file:
   `NewMM_Log_DATE`
   Existing models are converted to the newly supported model types.

Preserve Customized Support Files

Some custom support files of DX NetOps Spectrum can be overwritten when installing a new version of DX NetOps Spectrum. These support files include AlertMap, EventDisp, Event Format, Probable Cause, or GIB files.

For example, you can have customized event files that exist in the <*$SPECROOT*>/SG-Support/CsEvFormat directory. Before you upgrade the OneClick web server, move these files to <*$SPECROOT*>/custom/Events/CsEvFormat.

WARNING
Do *not* copy the <*$SPECROOT*>/custom/common/config/custom-jnlp-config.xml file to another computer when you migrate and upgrade DX NetOps Spectrum. This file can contain memory settings that are not compatible with the computer where you are copying the custom directories.

Troubleshooting Upgrade Installation Problems

Troubleshooting the Post Upgrade SDC Alarm Correlation

When upgrading to 10.3.1 from 10.3 (directly) or 10.2.x (indirectly), if there were any alarms on the devices managed through Secure Domain Connector (SDC), those alarms will not be correlated to SDC lost alarm and only the newly generated alarms will be correlated, post-upgrade. This is a known and expected behavior.

In an SDC FT scenario if both SDCs are down, then only the device down alarms are correlated to “Secure Domain Lost” alarm.

Troubleshooting the Post Upgrade Installation Script

NOTE
You can troubleshoot the post-upgrade installation script.

NOTE
Log files are in the <*$SPECROOT*>/Install-Tools/PostInstall/ directory. On Windows, run all scripts from a bash shell.

Follow these steps:

1. Start the SpectroSERVER, if it is not already running:
   – On Windows, click Start, Programs, CA, Spectrum Control Panel. The Spectrum Control Panel displays. Click the Start SpectroSERVER button.
   – On Linux, run the SCP command, which is located in <*$SPECROOT*>/bin/. The Spectrum Control Panel displays. Click the Start SpectroSERVER button.
   The SpectroSERVER begins to run.

2. Open the OneClick home page in your web browser, using the URL that your administrator provided. The URL has the following format: http://<*hostname*>:<*portnumber*>/.
3. Enter your OneClick login credentials when prompted.
   The OneClick home page opens.
4. Click Start Console.
   The OneClick Console opens.
5. Expand the SpectroSERVER that has been named the main location server and click Universe in the Navigation panel.
   A list of alarms, if any, appear in the Alarms tab of the Contents panel for the Universe topology. If any models display Minor (yellow) alarms with a probable cause of DIFFERENT TYPE MODEL, clear the alarms. To verify that the script converted all eligible models that it discovered, rerun the NewMM.pl script.
   The log file, NewMM_Log_<DATE>, is created in the <*$SPECROOT*>/Install-Tools/PostInstall/ directory.
6. To verify that all models converted successfully, check the log file, NewMM_Log_<*DATE*>.

   NOTE
   If DIFFERENT TYPE MODEL alarms recur, contact Technical Support.

   The troubleshooting is complete.

Unable to Proceed with DX NetOps Spectrum Upgrade Due to DDM Database Lock with Errors in Pre-Install Log

Symptom: When I am trying to install a patch on my existing DX NetOps Spectrum version, I am getting the following error message:

"This installation needs to access the DDM database, which is currently locked. Installation will automatically resume when the database is unlocked."

Solution: When you get this error, browse to the $SPECROOT/SS/DDM directory and then delete the .DDMDB.LOCK file.
Once you have done this, the install should continue to finish as expected.

Loading r9.2 SSdb models to r10.2

This release enables you to load the existing SpectroSERVER database models to a different system, and then upgrade DX NetOps Spectrum. This upgrade scenario only loads models from your r9.2 SSdb. All events and catalog customizations (watches, model type default changes, changes done in MTE etc) information will be lost. If you want to retain the custom configuration and data, please upgrade to 9.4 first, and then proceed to install 10.2.

NOTE

The r9.2.x to r10.2 upgrade (only DB Migration) is supported only for SSDB and not for DDMDB and SRM DB.

Before you initiate upgrading your setup from r9.2 to 10.2, we recommend you to familiarize yourself with the following procedures:

- Upgrading
- Migrating and Upgrading
- Database Backups
- Preserving the existing SpectroSERVER database.

NOTE

You can upgrade ISO-8859-1, ISO-8859-2, ISO-8859-5, ISO-8859-8 and ISO-8859-9 encoding types.

To initiate loading your r9.2 Ssdb to r10.2, follow these steps:

1. Preserve a copy of your r9.2 SpectroSERVER database by using one of the following methods:
   a. The SSdbsave utility

   b. DX NetOps Spectrum's Online Backup utility available from the VNM model.
   c. The Save Database button on the Spectrum Control Panel.
   d. Or use one of your previously saved SSdb files if you have already configured automatic backups with the Online Backup utility
2. Perform a fresh installation on a new machine using the r10.2 build.
3. Transport the save file from step 1 to the r10.2 machine and put it in the <SPECROOT>/SS-DB-Backup directory (If the directory is not there, create it).
4. Unzip the SSdb file if it is zipped into the SS-DB-Backup directory.

NOTE

During the r10.2 installation, ensure that you select the **Legacy Landscape Handle** in order to load your preserved model data/SSdb. If you select the **Huge Landscape Handle** option, the data load will fail.
The **Huge Landscape Handle** option is recommended to be used only for new DSS environments and single-server installations.

After a successful installation of r10.2, use the SSdbload utility to import the SSdb models you had saved from your previous setup following these steps.

1. Unzip your r9.2 SSdb file if it is zipped.
2. Launch a shell as your DX NetOps Spectrum install owner and navigate to the <SPECROOT>/SS directory
3. Run the following SSdbload command:

```
../SS-Tools/SSdbload -m -UpgradeFrom <ENCODING_NAME> ../SS-DB-Backup/
<9.2savefile>.SSdb


For example:
..SS-Tools/SSdbload -m -UpgradeFrom ISO-8859-1 ../SS-DB-Backup/db_20161101_1200.SSdb
```

4. Once complete, you can start the SpectroSERVER

NOTE

For <source encoding> select one of the following:

- ISO-8859-1
- ISO-8859-2
- ISO-8859-5
- ISO-8859-8
- ISO-8859-9

WARNING

For <9.2db_filename>, ensure that the filename and the destination folder are the same as you had saved.

Install Report Manager

DX NetOps Spectrum Report Manager generates reports in graphical and text-based formats. Reports related to DX NetOps Spectrum features are service management, performance management, response time statistics, VPLS reports,

and others. The on-demand reporting feature provides a custom report development tool that reports on multiple DX NetOps Spectrum data attributes.

You can access the BI Launch Pad application server from any compatible Web browser. BI Launch Pad is a web-based interface that lets you manage reports. You can generate custom reports and also one-time or periodic scheduled reports.

As an administrator, your responsibilities include installing, configuring, and maintaining Spectrum Report Manager application. In addition, you can schedule and manage reports for members of your organization.

DX NetOps Spectrum Report Manager uses CA Business Intelligence (CABI) to display and generate reports. CABI is a reporting and analytic software package that DX NetOps Spectrum and other CA products use to present information and support business decisions. DX NetOps Spectrum uses CABI to integrate, analyze, and present vital information that is required for effective enterprise IT management.

For the 10.4.1 release, refer the Integration Compatibility matrix for supported CA Business Intelligence JasperReports® Server versions for Windows and Linux platforms. For more information, see CA Business Intelligence JasperReports® Server - 6.4.2 / CA Business Intelligence JasperReports® Server - 6.4.3 documentation for more information.

Install OneClick with Report Manager

NOTE
This section describes how to install OneClick with Spectrum Report Manager. For more information, see the Installing section.

WARNING
We recommend that you back up the reporting data on the installation server before you upgrade from an earlier version of OneClick with Spectrum Report Manager.

Best Practices While Migrating SRM

Review the migration process is the best thing for your business needs. For example, if you only leave 30 days' worth of data in Spectrum Report Manager but leave 45 days worth in your Archive Manager DDMDB - moving the database is not recommended since Report Manager can process those 30 days from the DDM. Follow the migration process only if you keep more data in SRM than stored in the DDMDB.

The following are the best practices we recommend while migrating SRM:

- Anytime the Spectrum modeling database (SSdb) is backed up, the Archive Manager database (DDMDB) and Reporting databases should also be backed up to maintain the three database synchronization.
- If an SSdb is reloaded, the other two databases should also be reloaded. This is not always possible because the reporting database takes up so much space. However, migration is recommended.
- If the SSdb is reloaded but the others are not you will have an SSdb that does not have certain data that the Reporting database may have, like an outage or a newly created model.

Migrate Report Data from a Previous Report Manager During OneClick Installation

During the installation of OneClick with Spectrum Report Manager, you are prompted to migrate report data from a remote (source) reporting database to the new DX NetOps Spectrum reporting database. The prompt applies to upgrade situations where data from a previous installation is preserved. This migration is optional. Therefore, you can either accept or decline the migration.

If you prefer to migrate data, enable access to the source report database from the remote server as described in this section.

Follow these steps:

1. Launch a MySQL client session on the source server with root account credentials. For example:
   ```
   oscmdline> ./mysql -uroot -p<localrootpassword>;
   ```
2. Let data be extracted from the source database by a remote account. You can provide temporary access to a remote root account.
   For example, if the DX NetOps Spectrum target OneClick Linux server is named target-linux.ca.com, issue the following command at the MySQL command line:
   ```
   mysql> GRANT ALL PRIVILEGES ON *.* TO 'root'@'target-linux.ca.com' IDENTIFIED BY '<remoterootpassword>';
   ```
   NOTE
   Provide the fully qualified host name.
3. Verify that this new permission is available to all existing sessions by issuing the following command:
   ```
   mysql> FLUSH PRIVILEGES;
   ```
4. (Windows Only) Verify that Windows Firewall settings on the source server allow remote connections to MySQL.
   a. Click Start, Control Panel, and then Windows Firewall.
   b. Under the Exceptions tab, click Add Port and configure values as follows:
      - Enter **MySQL** for the name.
      - Enter **3306** for the Port Number.
      - Select **TCP**.
   c. (Optional) Click Change to restrict the scope of access to MySQL.
   d. Specify the location of your DX NetOps Spectrum server.
   e. Configure the option to allow remote connections to MySQL only from the migration destination server. For more information, see Windows Help and Support.

   Access to the report database on the Windows server is enabled.

Prepare for the Migration

Before you start migrating the data, verify the connection between the source and destination server databases. Verify that the data you plan to migrate is updated.

Follow these steps:

1. Verify the database connection to the remote host containing data that you plan to migrate. Issue the following command on the DX NetOps Spectrum server:
   ```
   telnet <remote-srm-host> 3306
   ```
   The following message indicates that the permissions have been set properly:
   ```
   Escape character is '^]'. 7
   4.1.11-nt î&#9786;  t#J0Mu'] ,&#9787;   #2p^giYa]0t{
   &#9644;  &#9786;&#9830;#08S01Bad handshakeConnection closed by foreign host.
   ```
   The following message indicates that the permissions have not been set correctly:
   ```
   Q   &#9830;#HY000Host 'user.com' is not allowed to connect to this MySQL server Connection closed by
    foreign host.
   ```
   If you are unable to connect to the MySQL server, verify that your MySQL permissions are configured correctly on your previous SRM MySQL database. Verify that the privileges are flushed before you reattempt to connect.
2. Stop all reporting processes on the remote, source server by removing all entries from the Spectrum Report Manager Admin Tools, DX NetOps Spectrum Status option.
3. Wait for 5 minutes to verify that any outstanding data changes are committed to the report database.
   The root database account on the remote destination server can extract report data on the source server.
   The connection is verified.

Post Migration

Perform the following tasks after migration:

- Enable the integration of DX NetOps Spectrum with CABI. For more information, see Business Objects Integration.

  NOTE
  If you have integrated DX NetOps Spectrum with CABI before migration, we recommend disabling the previous integrations.
- To ensure that the most recent reporting content (such as Crystal Reports) is available, update the existing content that is installed by DX NetOps Spectrum in Business Objects. For more information, see Manage Business Objects Content.

Initialization Considerations for InnoDB Storage

With 9.4, reporting data is now stored using only the InnoDB storage-engine based tables.

For new installations, Spectrum Report Manager automatically ensures that InnoDB is used for all of the reporting tables.

For upgrade installations, Spectrum Report Manager migrates all the reporting tables from MyISAM to InnoDB.

WARNING
Before you upgrade, verify that the amount of free disk space on the system is at least twice the size of the largest MYD file under *$SPECROOT*/mysql/data/reporting.

Calculating Disk Requirements for Event Storage

Use the following formula to estimate the amount of disk space that is required to support the reporting database for a specific amount of time.

```
Total Gigabytes Required = ((Number of Devices) * (Average Number of Events per Device per Day) * (Number of
 Days Storage Required) * (Average Size of Event in Kilobytes)) / 1048576
```

The following variables are used:

- Number of Devices - The number of devices at your site. Consider the future growth of your site when determining this value.
- Average Number of Events per Device per Day - The total number of events that are generated on a daily basis that are associated with a single device model. This number includes all events that would result from related application, port, and interface models. The easiest way to get an approximation is to look at the total number of events that were generated on a single SpectroSERVER in a single day and divide it by the number of devices that are modeled on that SpectroSERVER.
- Number of Days Storage Required - The number of days that your site requires storage.
- Average Size of Event in Kilobytes - An estimation of the amount of disk space a single event ends up consuming in the Reporting database.
- 1048576 - A conversion factor for gigabytes.

In addition to the number of devices and the number of days of storage, two variables are required to estimate the database size:

- **Average Number of Events per Device per Day**
  Query the DDMDB to see the average number of events that are generated on a given day.
  If you are a new DX NetOps Spectrum user and do not know the average number of events, use a default value. Three hundred events per day, per device for 500 devices would equate to 150,000 events a day. Therefore, 300 would be a reasonable default value.
- **Average Size of Each Event in Reporting DB**
  An appropriate amount of space to store your average event and corresponding records is 1 KB. This number can increase if most of the events that are being handled are large events that contain much data. Also the types of events

affect data size. Alarm events turn into multiple reporting table records. Network Configuration Manager (NCM) events only affect a single table (event).

Here are some examples:

Example A - User has 600 devices and wants to keep data for 4 years (1460 days).

The user does not know how many events per device, therefore consider 300 as the default value.

```
Total GBs required = (600 * 300 * 1460 * 1) / 1048576
Total GBs required = 262,800,000 / 1,048,576
Total GBs required = 250 GBs
```

Example B - User has 1900 devices across three servers and wants to keep data for 2 years (730 days).

The user seems to be averaging 400 events per device, per day. In this example, the three servers are not considered.

```
Total GBs required = (1900 * 400 * 730 * 1) / 1048576
Total GBs required = 554,800,000 / 1,048,576
Total GBs required = 530 GBs
```

How to Calculate Average Daily Event Rate per Device

To estimate the average daily number of events that generated per device, you first need to know how many events get generated each day. Use the following queries on the DDMDB database.

The following query returns the total event count for the last ten days:

```
SELECT date(from_unixtime(utime)) as x, count(*) as cnt
  FROM event GROUP BY x
  ORDER BY x DESC LIMIT 10;
```

The following query returns the 10 days with the highest event volume and the event volume for each day::

```
SELECT date(from_unixtime(utime)) as x, count(*) as cnt
  FROM event GROUP BY x
  ORDER BY cnt DESC LIMIT 10;
```

Use the result of these queries to come up with a reasonable event count. Once you know the event count, divide the number of events by the total number of modeled devices on the server to derive the average event count per device, per day.

Report Manager Installation

You can install OneClick with Spectrum Report Manager by specifying Spectrum Report Manager as a feature selection during the OneClick installation process. To install Spectrum Report Manager, you need OneClick, however, Spectrum Report Manager is not required to run OneClick. During installation, you can migrate the report data from a previous Spectrum Report Manager installation.

> **WARNING**
> The disk space that is required for a Spectrum Report Manager upgrade installation is two times the table size for each table that is converted to InnoDB. If enough space is not available, the installer displays a warning.
>
> **NOTE**
> Multiple Spectrum Report Manager installations using a common set of SpectroSERVERs can result in inconsistencies between the primary and secondary Spectrum Report Manager installations.

Follow these steps:

1. Select the Spectrum Report Manager option from the Select Features window during the installation.

2. When prompted during the installation, specify the names of the DX NetOps Spectrum servers from which you want Spectrum Report Manager to collect data.
   The Spectrum Report Manager Servers dialog lets you specify the names of more DX NetOps Spectrum servers, in addition to the primary server specified for OneClick. You can also modify the servers list after you have completed the installation using Spectrum Report Manager Admin Tools.

   NOTE
   Use DX NetOps Spectrum landscape names to specify servers.
3. Select to migrate historical report data associated with a previous release of OneClick with Spectrum Report Manager to the new reporting database.
   – If you select to migrate reporting data, specify the following options:
     a. Specify the host from which you want to migrate the data in the Source host name field.
     b. Enter the password to access the MySQL installation on the remote server in the Source Host 'root' Database Password and Verify Password fields. The default password is ‘root’.
     c. Click Next.
   – If you do not want to migrate reporting data, do not type any values in the window and click Next.
4. Follow the onscreen instructions, to continue the OneClick installation.
   OneClick with Spectrum Report Manager is installed successfully.

Spectrum Report Manager and Fault Tolerance

Support for fault tolerance exists for the DX NetOps Spectrum application, but does not extend to the Spectrum Report Manager component. Architectural limitations within the Spectrum Report Manager do not enable a Fault Tolerant configuration beyond standard database/file replications.

Multiple Spectrum Report Manager installations using a common set of SpectroSERVERs can result in inconsistencies between the primary and secondary Spectrum Report Manager installations. The inconsistencies occur due to the lack of an integrated fault tolerance architecture within the Spectrum Report Manager components.

Verify Installation by Testing Access Methods

Verify the Spectrum Report Manager environment after installing OneClick with Spectrum Report Manager. Verify that all of the SpectroSERVERs have started.

Follow these steps:

- Open BI Launch Pad from the OneClick web console and run several DX NetOps Spectrum reports.
  For more information, see the Report Manager section.

  NOTE
  If you do not install OneClick with Spectrum Report Manager correctly, then you cannot generate reports or notice any other application irregularities. For more information, see Troubleshooting.

Upgrade the Report Parameter Pages

If you update Spectrum Report Manager, run the spectrum-wkp-update.bat tool on the CABI server that is integrated with Spectrum Report Manager. The spectrum-wkp-update.bat tool downloads updated files from the DX NetOps Spectrum web server and deploys them on the CABI server.

Procedure for Windows

Follow these steps:

1. Open Command Prompt.
2. Run the following command (words in italics indicate installation-specific values):

```
% cd "C:/Program Files/CA/SC/CommonReporting3/spectrum"
```

```
% spectrum-wkp-update.bat -host http://spectrum-hostname:port -username admin_name -password
 admin_password
```

The -host flag can also specify an https URL, if SSL is configured on DX NetOps Spectrum.

3. Follow the onscreen instructions to upgrade the Report Parameter pages.
   Report Parameter pages are upgraded.

Procedure for Linux

Follow these steps:

1. Open Command Prompt.
2. Run the following command (words in italics indicate installation-specific values):

```
% cd /opt/CA/SharedComponents/CommonReporting3/spectrum
% spectrum-wkp-update.sh -host http://spectrum-hostname:port -username admin_name -password admin_password
```

The -host flag can also specify an https URL, if SSL is configured on DX NetOps Spectrum.

3. Restart the Tomcat server.
   The process to upgrade the Report Parameter pages is completed.

Migrate Report Data from a Previous Report Manager After OneClick Installation

As hardware updates become available, there may be times you need to move your software products from an older system to a newer one. To move the database ensure you meet the following requirements:

1. Landscape handles names MUST remain the same.
   Else SRM treats them as new landscapes and report them separately.
2. The new system MUST have ample disk space to handle the new database plus some for new data and in the event a repair needs to be done (2x the largest table - For example, 100GB table, 200GB free space is needed to repair).
3. The new system MUST have the exact version of DX NetOps Spectrum the old system had on it (including DX NetOps Spectrum patches),

Backup the Database

Follow these steps:

1. Shutdown Spectrum Tomcat Service (This will prevent any potential writes to the database while it's backed up).
2. Navigate a bash shell to the `$SPECROOT/mysql/bin` directory and run the following command:

```
mysqldump --defaults-file=../my-spectrum.cnf --opt --routines --ignore-table=reporting.v_active_user_model
 --ignore-table=reporting.v_alarm_activity --ignore-table=reporting.v_ncm_config_diff
 --ignore-table=reporting.v_security_string_accessibility_by_landscape--ignore-
table=reporting.v_user_report_security --compress -uroot -p<root_pw> --databases reporting > <dump_log>
```

NOTE
Backup process may take a while to complete based on the database size and system resources available. We recommended not run any other process till the backup is completed.

You can find the `dump_log` file in LOCATION. Compress the file using any compression tool like the zip to transfer the file to the new system.

3. Move the `dump_log` file to the $SPECROOT/mysql/bin directory.
4. Shutdown DX NetOps Spectrum Tomcat Service
5. Navigate a bash shell to `$SPECROOT/mysql/bin` directory and run the following command:

```
mysql --defaults-file=../my-spectrum.cnf -uroot -p<root_pw> reporting < <dump_log>
```

NOTE
For example: `mysql --defaults-file=../my-spectrum.cnf -uroot -proot reporting < reporting_db_backup_2012_10.sql`

NOTE
The database restore process may take a while to complete based on the database size and system resources available. We recommended not run any other process until the database is restored.

Set Up Report Manager to Process Data From a Migrated DDMDB

If you do not copy the database and want to set up a Report Manager to process data from a migrated DDMDB follow this process on the SRM host.

Follow these steps:

1. Navigate to `$SPECROOT/bin` directory.
2. Run the following command:

```
./RpmgrInitializeLandscape <username> <password> [-skipInitialHistory] [-initHist <#of days>] [-slm] [-all] [<landscape1> <landscape2> ...]
```

NOTE
For example: ./RpmgrInitializeLandscape root root -initHist 45 -all

NOTE
The command removes all SRM data currently existing in the SRM database and start processing new data from the DDMDBs.

CABI (JasperReports Server)

CA Business Intelligence JasperReports Server is a high-performance standalone and embedded Business Intelligence (BI) platform. CA Business Intelligence JasperReports Server provides rich reporting and integrates in-memory analysis capabilities.

DX NetOps Spectrum uses the latest version of CA Business Intelligence JasperReports® Server to integrate, analyze, and present the vital information through various reporting options that are required for an effective enterprise IT management.

CABI JasperReports Server installation runs independently and enables other CA products to share Business Intelligence services. You can activate the integration between CABI JasperReports Server and DX NetOps Spectrum Report Manager after installing CABI JasperReports Server on your system.

Install and Upgrade CABI JasperReports Server

Install CABI JasperReports Server

Click **here** to see the detailed information about Prerequisites, Installation Types, Installation, and Post Installation processes.

Set up MySQL as a Repository for CABI JasperReports Server

1. Create MySQL user and assign privileges. Create a database user who can create a new database. (applicable **only if you select MySQL that comes with DX NetOps Spectrum** as Jasper repository)
   You can use the DX NetOps Spectrum MYSQL database and login to MySQL from the command prompt. Use the following command to connect to MySQL instance of DX NetOps Spectrum from the command prompt:
   Navigate to $SPECROOT/mysql/bin directory and then from DX NetOps Spectrum bash shell run “./mysql --defaults-file=../my-spectrum.cnf -uroot -proot
   You can use the following queries to create JRS repository user and grant privileges:
   CREATE USER 'CR_user'@'jasperserverip' IDENTIFIED BY '0n3cl1Ck';

GRANT CREATE, DROP, INDEX, SELECT, DELETE, ALTER, INSERT, UPDATE, REFERENCES, LOCK TABLES on *.* to 'CR_user'@'jasperserverip' identified by '0n3cl1Ck';
Or
CREATE USER 'CR_user'@'trustednetwork/networkmask' IDENTIFIED BY '0n3cl1Ck';
GRANT CREATE, DROP, INDEX, SELECT, DELETE, ALTER, INSERT, UPDATE, REFERENCES, LOCK TABLES on *.* to 'CR_user'@'trustednetwork/networkmask' identified by '0n3cl1Ck';

2. Update the '**sample_mysql.properties**' file that is located in \ca_install with database and application settings.

   NOTE
   If you are using Oracle as a database, use sample_Oracle.properties file. Do not remove the other properties even if they are not being used.

3. Install JasperReports Server.
   Notes:
   - For silent installation, you can run the following command by navigating to the ca_install folder:
     install.bat -r sample_mysql.properties -l <<install.log>>
     Here -l specifies the log file. The log file extension should be .log. If you specify any other extensions, the installation fails.
     By default, if you do not specify the log file path, then it is stored in the current working directory(that is ca_install).
   - If installation fails, error messages are shown in the console with error codes. Refer to install.log specified during installation.
   - JasperReports Server needs DB repository (you can use MySQL that comes with DX NetOps Spectrum). JasperReports Server creates its own DB ('jaspersoft') in the MySQL instance. You can install DX NetOps Spectrum OneClick with SRM on a separate box even before attempting to install JasperReports Server and use that as a repository. If you plan to use JasperReports Server for other CA products also, then you can select a common repository.

Upgrade to CABI JasperReports Server 7.1.1

For more information on upgrading to CABI JasperReports Server 7.1.1, see Upgrading CA Business Intelligence JasperReports Server

Post-Upgrade Steps for CABI JasperReports Server 7.1.1

Perform the following steps after you upgrade to CABI JasperReports server 7.1.1 version:

1. Run SpectrumConfigInstaller.
2. If SSO was enabled before the upgrade, navigate to DX NetOps Spectrum Administration Page, Report Manager, Jasper Integration page.
3. **Click Disable** and click save.
4. Once it is disabled successfully, Click **Enable**, and click **Save**.

   NOTE
   DX NetOps Spectrum tomcat will be restarted while enabling or disabling the SSO.

Integration with CABI JasperReports Server

This section describes the steps to integrate DX NetOps Spectrum with CABI Jasper server.

After installing the JasperReports Server, you need to configure the integration between JasperReports Server and DX NetOps Spectrum Report Manager. To enable this integration, provide the Jasper Server connection details on the integration page.

NOTE

If you are using a JasperReports server that comes with a CA product other than DX NetOps Spectrum, then do the following before the integration:

Copy the SpectrumProxy.war file from the location $SPECROOT\Spectrum\Install-Tools\SRM-Tools\jasper\ to Japserserver\tomcat\webapps folder. Restart the Jasper Tomcat Server.

Follow these steps:

1. Open the OneClick Administration page.
2. Click the Report Manager tab.
3. Select the Jasper Integration option from the Report Manager Admin Tools.
4. Download and install the '**Integration Components**'. For installation instructions, see the Installing Integration Components section.

   NOTE
   Information! The 'Integration Components' is a JAR file, which consists of the necessary binaries to integrate the JasperReports Server with DX NetOps Spectrum and run reports. Installing the 'Integration Components' will deploy the binaries on the Jasper Server and helps you to generate reports.
5. Specify the parameters that are used to communicate with the JasperReports server:
   - **Jasper Server Host name**
   - Specify the host name of your CABI JasperReports Server instance if it is not the same server as DX NetOps Spectrum Tomcat.
   - **Jasper Tomcat Port**
     Specify the port where Jasper Tomcat is running. The default port value is 8080.

     NOTE
     The Tomcat port cannot be the same port that DX NetOps Spectrum Tomcat uses if Jasper and OneClick are on the same server.
   - **JasperServer Webapp Name**
     Specify the Jasper server webapp name that is given during the installation of Jasper server. Default is 'jasperserver-pro'.
   - **Jasper Admin User**
   - Specify the Jasper Admin User ID.
     The default User ID is 'superuser'.
   - **Jasper Admin Password**
     Enter the password for the Admin User ID in Jasper. The default password is 'superuser'.
   - **Jasper Integration**
     Select the Enable radio button.
     If Disable is selected and saved, DX NetOps Spectrum Tomcat no longer integrates with CABI Jasper instance.
   - **Enable SSO**
     Select this checkbox to enable single sign-on solution(SSO) from JasperReports Server, which establishes a session between the DX NetOps Spectrum Report Manager and Jasper console. If you enable the SSO, you can open Jasper console session from a OneClick web console without providing any login credentials.
   - **Use SSL with Jasper Server**
     Select this checkbox to integrate with Secure Sockets Layer (SSL) enabled JasperReports Server. To know more about Secure Sockets Layer (SSL) and how to enable it on JasperReports Server, see Using SSL in the Web Server. (Supported only when the JasperReports Server runs on Windows 2012).

     NOTE

     SSL and Single Sign-On can both be enabled with the JasperReports Server integration.
6. Click Save to enable the integration and refresh the page.
7. On the OneClick home page, click 'Jasper Console' to launch the JasperReports Server.

NOTE
This process can take some time. During this process, all the DX NetOps Spectrum Report Manager report content is exported from the OneClick server into CABI JasperReports Server. Therefore, do not cancel or navigate away from this page until you get a success message.

After you configure the integration, the DX NetOps Spectrum Report Manager report content is installed and can connect to the CABI JasperReports Server reporting database. The menubar 'JasperConsole' link now launches the CABI JasperReports Server web applications on the CABI instance that you specified.

NOTE

If you disable the integration, reporting and report administration capabilities are disabled. However, disabling the integration does not cause DX NetOps Spectrum Report Manager to stop collecting and managing data from the monitored SpectroSERVERs.

After successful integration, you can see the following DX NetOps Spectrum organization and default users in Jasper:

- jasperadmin/jasperadmin
  ROLE_ADMINISTRATOR
- joeuser/joeuser
  ROLE_USER
- spectrum/spectrum
  ROLE_ADMINISTRATOR
  ROLE_USER

Redeploy

Re-deploy the reports to repair the existing report or to update to a newer version. After upgrading to a new version of DX NetOps Spectrum, click the Re-deploy button to sync up the Jasper reports with DX NetOps Spectrum upgrade fixes, enhancements, and new reports.

Features not supported in this release

NOTE

A limited set of reports and custom reporting capabilities are supported in this release.

Available Out of the Box Reports and Domains

Following is the list of 'Out of the Box Reports' available for users:

Reports Details and Descriptions

Release Version	Reports	Report Name	Description
10.3.2	Service and SLA: SLA Status	SLA Detail by SLA Name	Reports SLA and guarantee status for one or more SLA periods. Includes sub-reports for guarantee details and service availability.

10.3.1	Service and SLA: Summarized Availability	Service Summary Variable Health Level	This report displays service health summary. User selects level of service health severity to display. An availability target can be specified, and an extended summary including initial and loss of management time is available.
10.3.1	Service and SLA: Summarized Availability	Service Summary by Service Name	This report displays the service availability reports specified by a service name. An optional extended summary is available, which shows initial time, loss of management time and table of outages.
10.3.1	Service and SLA: Health	Service or Resource Monitor Health by Name	This report displays service health reports specified by service or resource monitor name. Includes all outages and reports recalculated outage values.
10.3.1	Service and SLA: Outage	Top-N Worst Performing Services	This report displays top-n worst performing services over the specified period by total downtime. The user specifies how many services are included in the report.
10.3.1	Service and SLA: Detailed Availability	Service Availability Variable Health Level	This report displays the service availability report which allows the user to specify the service health level. An optional extended summary is available which shows initial time and loss of management time.
10.3.1	Service and SLA: Detailed Availability	Service Availability by Service Name	This report provides an optional extensive summary which displays initial time, loss of management time and table of outages.
10.3.1	Service and SLA: Inventory	Service Inventory	This report displays all currently modeled services and resource monitors listed alphabetically.
10.3	-	-	-
10.2.2	Alarm: Group	Alarm Activity By User: Group	This report displays the count of alarm activity performed by each user associated with all devices, models within a specified global collection. Entry in the list includes count of alarms cleared, acknowledged by user, assigned to troubleshooter.
10.2.2	Alarm: Individual	Alarm Activity By User: Selected Devices and Models	This report displays the count of alarm activities performed by each user associated with the selected device or models.

10.2.2	Alarm: All	Alarm Count Trend: All	This report displays the alarms count broken out per month - infrastructure wide and for the DX NetOps Spectrum server.
10.2.2	Assets: All	Current Assets (Customizable): All	This report displays the current attributed for selected devices. These attributes provide device and modeling information. User defined attributes provided by DX NetOps Spectrum are available. Information is grouped sorted by attributes and selected. Sub-report provides details for device selected.
10.2.2	Assets: All	Current Chassis-based Assets: All	This report displays all managed chassis based devices and linecards/modules. This list of devices is grouped by vendor. This report displays the model name, IP address and device type for each device.
10.2.2	Assets: All	Current Port Assets (Customizable): All	This report displays selected attributes for all currently managed ports. The attributes available provide port and modeling information. Furthermore the user defined attributes by DX NetOps Spectrum are also available for display.
10.2.2	Assets: All	Detailed Change Management: All	This report identifies devices created and deleted within a user specified time period. The following information is available for each device which is listed: device name, creator, created time, user who deleted and the deleted time.
10.2.2	Event: Group	Detailed Event Log: Group	This report displays a reverse chronological list of events for all devices and models within a specified global collection. Each entry in the list includes the IP address (if applicable), event text, event code and event creator.
10.2.2	Availability: Group	Outage Log: Group (Devices Only)	This report lists the outages for selected devices within the specified global collection. For each outage listed, the device name, device type, model class, vendor and location associated with the outage is displayed.

10.2.2	Availability : Group	Top-N Least Available: Group (Devices Only)	This report displays a list of devices from a specified global collection from with the lowest availability percentage within the specified time frame.
10.2 - 10.2.1	Event: Group	Top-N Most Common Events-Group	This report displays a list of the most frequently occurring event types for devices and models within a global collection. This reports presents an event log for each of the listed items.
10.2 - 10.2.1	Event: All	Top-N Most Common Events-All	This report displays a list of the most frequently occurring event types. This report presents an event log for each of the listed items.
10.2 - 10.2.1	Availability : Group	Top-N Least Available: Group	This report displays a list of devices from a specified global collection with the lowest availability percentage within the specified time frame.
10.2 - 10.2.1	Availability : All	Top-N Least Available: All Devices	This report displays a list of managed devices with the lowest availability percentage within the specified time frame.
10.2 - 10.2.1	Event : Group	Top-N Devices and Models with the Most Events: Group	This report displays a list of managed devices and models from a specified global collection with the most events. The report presents an event log for each of the listed items.
10.2 - 10.2.1	Event: All	Top-N Devices and Models with the Most Events: All	This report displays a list of managed devices and models with the most overall events. The report presents an event log for each of the listed items.
10.2 - 10.2.1	Service and SLA: Customer	SLA Summary by Customer	This report displays the SLA status for a specified customer in a specified time rage. SLA periods arranged by SLA status descending from violated.
10.2 - 10.2.1	Service and SLA: Customer	SLA Status Current and Recent by Customer	This report displays the current and recent status of SLAs for a specific customer or a set of customers.
10.2 - 10.2.1	Service and SLA: Customer	SLA Inventory by SLA Customer	This report displays currently modeled SLAs and the corresponding guarantees for selected customers.

10.2 - 10.2.1	Service and SLA: Customer	SLA Detail by Customer	This reports displays the SLA and the guarantee status for a specified customer over one or more SLA periods. It shows the status of each guarantee violated or compliant. Guarantee status values of unaffected, compliant and warned are all represented as compliant.
10.2 - 10.2.1	Service and SLA: Customer	Service Summary by Customer	This report summarizes service availability for a set of services used by specific service customers. It displays available time and down time. An availability target can be specified, and an extended summary including initial and loss of management time is included.
10.2 - 10.2.1	Service and SLA: Customer	Service Health by Customer	This report displays the service health reports specified by service customer.
10.2 - 10.2.1	Service and SLA: Customer	Service Availability by Service Customer	This report displays the service availability reports specified by service customer. An optional extended summary is available which shows initial time, loss of management, and a table of outages which is recalculated or exempted so no longer impact service availability.
10.2 - 10.2.1	Availability : Group	Outage Log: Group	This report lists the outages for all managed devices and models specified within the global collection. For each outage listed the device model name, device type (if applicable), model class, vendor and location associated, with the outage are displayed.
10.2 - 10.2.1	Availability : All	Outage Log: All Devices	This report lists the outages for all managed devices. For each outage listed, the device name, the device type, model class, vendor and location associated, with the outage are displayed.
10.2 - 10.2.1	Event: All	Detailed Event Log: ALL	This report displays a reverse chronological list of events for all devices and models. Each inventory in list includes the IP address (if applicable), event text, event code and the event creator.

10.2 - 10.2.1	Service and SLA: Customer	Customer SLA Summary	This report summarizes of customer's SLAs, and the status for up to the six most recent SLA periods.
10.2 - 10.2.1	Service and SLA: Customer	Customer Detail	This report displays the contact information and list of services for a specified customer.
10.2 - 10.2.1	Assets: All	Current Ports Summary:All	This report displays the ports associated with all the currently managed device(s). For each device, the total number of ports is displayed, along with an availability summary of these reports. A sub report provides details for a device selected from a report.
10.2 - 10.2.1	Assets : Individual	Current Ports Summary: Selected Devices	This report displays the ports associated with the currently managed device(s) selected. For each device, the total number of ports is displayed, along with an availability summary of these reports.
10.2 - 10.2.1	Assets: Group	Current Ports Summary: Group	This report displays the ports associated with the currently managed device(s) from a global collection. For each device, the total number of ports is displayed, along with an availability summary of these reports.
10.2 - 10.2.1	Assets : Group	Current Ports Detail:Group	This report displays the ports associated with the currently managed device(s) from a global collection.
10.2 - 10.2.1	Assets: Individual	Current Ports Detail: Selected Devices	This report displays the ports associated with the currently managed device(s) selected. For each device, individual ports are listed with its name, description, type, speed, and the current status along with the number of days the port has been idle.
10.2 - 10.2.1	Assets: Group	Current Assets: Group	This report displays a breakdown by model class of currently managed devices from a specified global collection. The report displays the model name, IP address, device type, firmware version and last reboot of each device listed.

10.2 - 10.2.1	Assets: All	Current Assets: All	This report displays a breakdown of currently managed devices by model class. The report displays the model name, IP address, device type, firmware version and last reboot of each device listed.
10.2 - 10.2.1	Assets: Individual	Current Asset Detail: Selected Devices	This report displays the ports associated with the currently managed device(s) selected. For each device the total number of ports is displayed along with an availability summary of these reports.
10.2 - 10.2.1	Availability : Individual	Availability: Selected Models	This report displays a detailed list of outages for each device and model specified by the user. It includes the total number of outages, total up time, total downtime and the availability percentage.
10.2 - 10.2.1	Availability : Individual	Availability: Selected Devices	This report displays a detailed list of outages for each device selected by the user.
10.2 - 10.2.1	Availability : Group	Availability: Group	This report displays availability information for each device and model within a specific global collection.
10.2 - 10.2.1	Availability : All	Availability by Class and Vendor: All Devices	This report presents the availibility of all devices broken down by vendor and grouped by class.
10.2 - 10.2.1	Alarm: All	Alarm Mean time to Respond statistics	This report displays the alarm mean time to respond to alarm statistics based on the severity.
10.2 - 10.2.1	Alarm: Individual	Alarm Log:Selected Devices and Models	This report lists of alarms for all devices and models that satisfy criteria of alarm creation and minimum duration threshold. Alarms are displayed in chronological order. Alarms are grouped by alarm condition and landscape.
10.2 - 10.2.1	Alarm: All	Alarm Log:All	This report lists alarms for all devices and models that satisfy criteria of alarm creation and minimum duration threshold. Alarms are displayed in chronological order. Alarms are grouped by alarm condition and landscape.

10.2 - 10.2.1	Alarm: Group	Alarm Log Group	This report lists alarms for all devices and models that satisfy criteria of alarm creation and minimum duration threshold. Alarms are displayed in chronological order. Alarms are grouped by alarm condition and landscape.
10.1.1	Alarm: All	Alarm Activity By user: All	This report presents the count of alarm activities performed by each user associated with all devices and models. Each entry includes a count of the number of alarms cleared, acknowledged, assigned and the number of alarms ticketed for a DX NetOps Spectrum user or a troubleshooter.
10.1.1	Alarm: All	Top N Devices and Models with Most alarms: All	This report displays a list of managed devices and models with the most overall alarms. A sub-report provides details for each of the alarms for a selected device or model.
10.1.1	Alarm: All	Top N Most common Alarms: All	This report displays the list of the most frequently occurring alarm types. A sub report provides details for each of the alarms for a selected alarm type.
10.1.1	Alarm: Group	Top N Devices and Models with Most alarms: Group	This report displays a list of managed devices and models with the most overall alarms. A sub-report provides details for each of the alarms for a selected device or model.
10.1.1	Alarm: Group	Top N Most common Alarms: Group	This report displays the list of the most frequently occurring alarm types. A sub report provides details for each of the alarms for a selected alarm type.
10.1.1	Alarm: Individual	Top N Most common Alarms: Selected Devices and Models	This report displays the list of the most frequently occurring alarm types. A sub report provides details for each of the alarms for a selected alarm type.

Domains

- NCM (from r10.2.3)
- Alarm (from r10.2.2)
- Availability
- Event
- Asset

How to Run Reports Using the JasperReports Server

Important! You must log in to JasperReports Server to access reporting functionality.

Generate General Reports

Follow these steps:

1. Open the OneClick console.
2. Click the 'JasperConsole' link to open the CABI JasperReports Server web application login screen.
3. Log in to the CABI JasperReports Server.

   NOTE

   If you are logged in as a DX NetOps Spectrum user, then you can generate reports only for the devices for which you have the access. If you are logged in as a non-DX NetOps Spectrum user or as a Jasper Super User, reports might not fetch any data. To fetch data in the reports for these users, set the value to 'False' for the Enable Security option in the Report Manager Preferences.
4. Click View, Repository.
5. (till r10.2.1) In the Folders pane, expand root, Organizations, spectrum, capability, and reports.
   (from r10.2.2) In the Folders pane, expand root, Public, ca, Spectrum, and reports.

   NOTE

   From 10.2.2, all the DX NetOps Spectrum content in CABI JasperReports Server is saved under the 'Public' folder (root->Public), previously it was saved under the 'Organizations' folder (root->Organizations).

   For the 10.2.2 users, we recommend to keep the old content, which was saved under the Organizations folder. However, you can delete the old content manually (Manage->Organizations then select Spectrum folder and click Delete).
6. Click a report title to open 'Input Controls' and generate a report.
7. Specify the parameters in Input Controls dialog, Click Apply and OK.

   NOTE

   Do not copy or move the DX NetOps Spectrum reports from default folder to another folder. The Jasper reports should be run from the default folder. Running the reports from other folders will throw an error.

   Following is the default location for the reports folder in the Jasper repository:

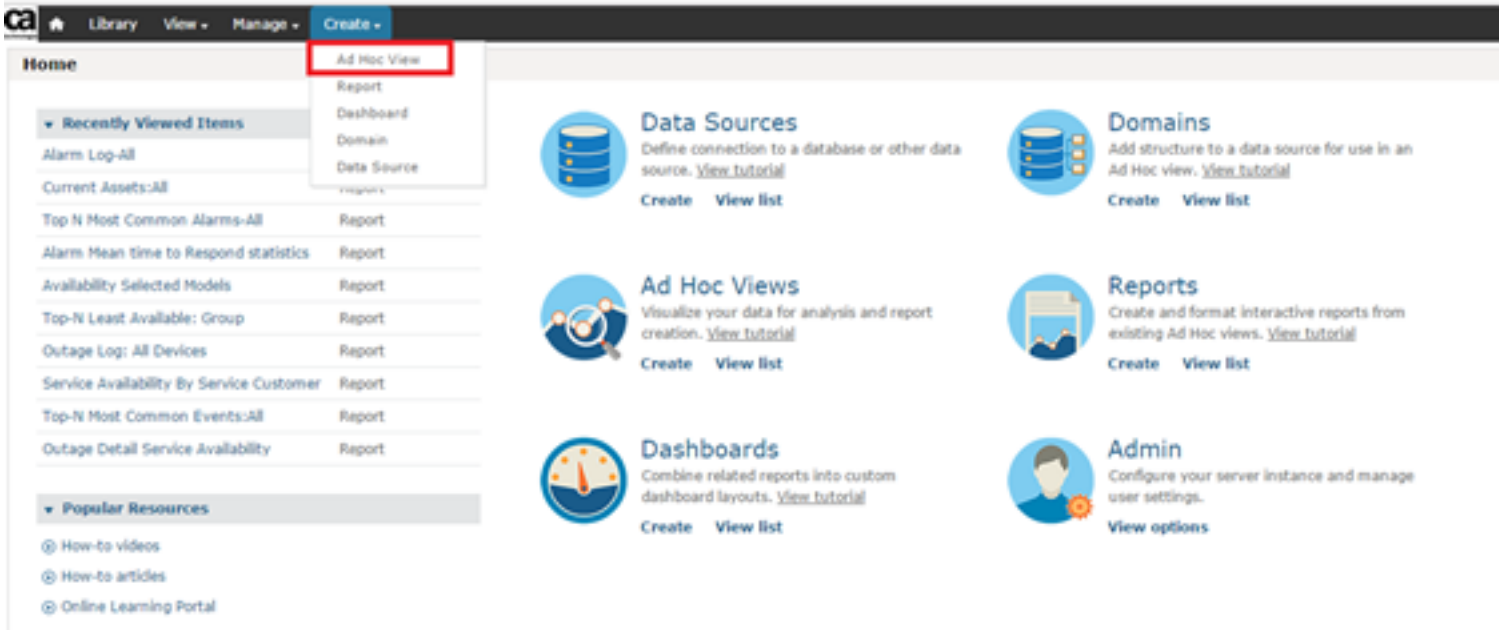
   For r10.2.2: root -> Public -> ca -> Spectrum -> reports.
   Till r10.2.1: root -> Organizations -> Spectrum -> capability -> reports.

Generate Ad Hoc Reports

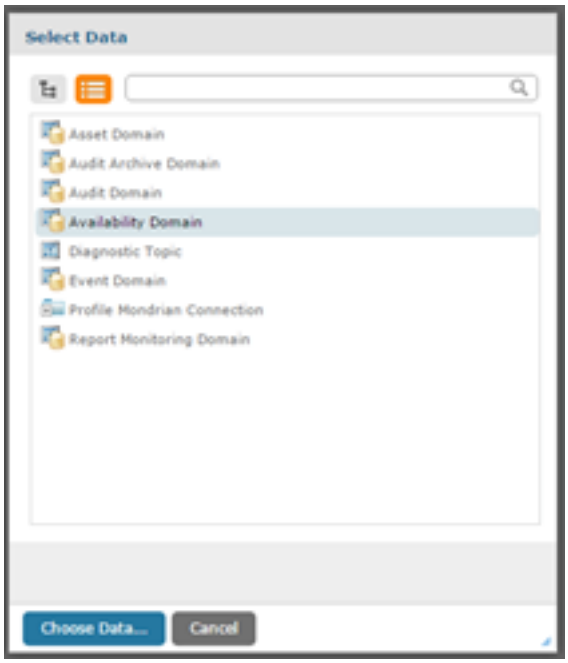
Domains available in Jasper help you to generate Ad Hoc or custom reports. For example, using the Availability domain, you can design the Ad Hoc reports on availability or outage information.

Follow these steps to generate Ad Hoc Reports:

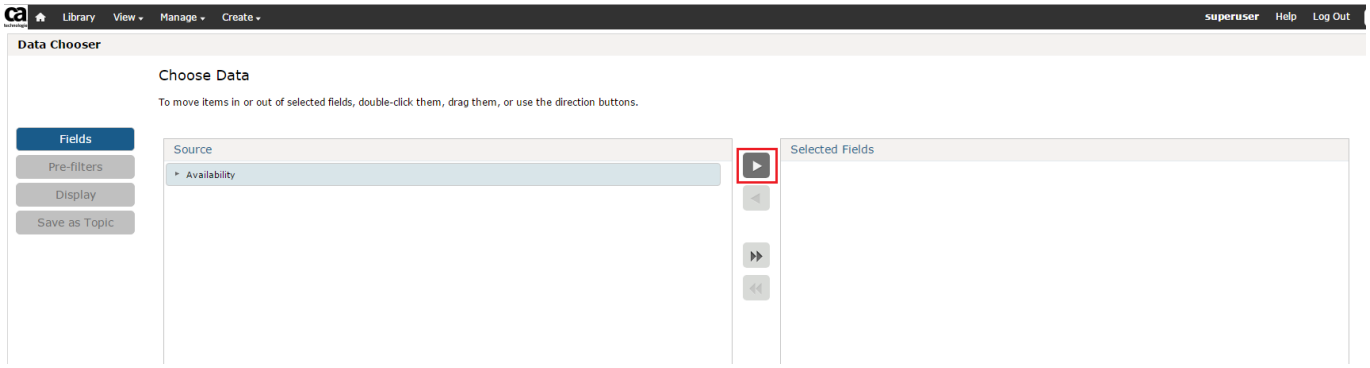
1. Open the OneClick console.
2. Click the 'JasperConsole' link to open the CABI JasperReports Server web application login screen.
3. Log in to the CABI JasperReports Server.
   Important! If you are logged in as a Spectrum user, then you can generate reports only for the devices for which you have the access. If you are logged in as a non-Spectrum user or as a Jasper Super User, reports might not fetch any data. To fetch data in the reports for these users, set the value to 'False' for the Enable Security option in the Report Manager Preferences.
4. Click Create, Ad Hoc View.



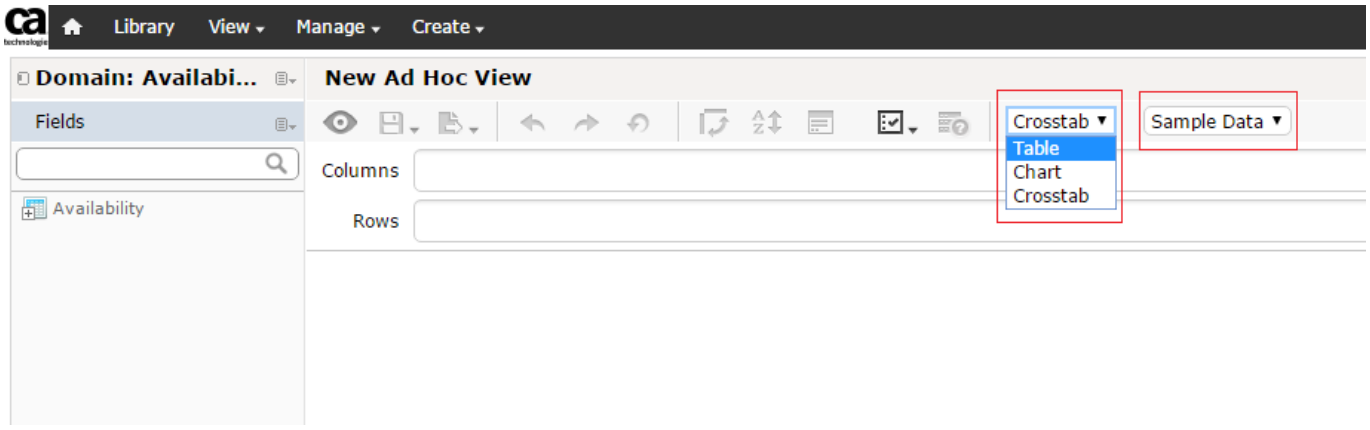
   The Select Data window appears.
5. Select any Spectrum related domain (for example, here it is Availability Domain) from the list, and click Choose Data.



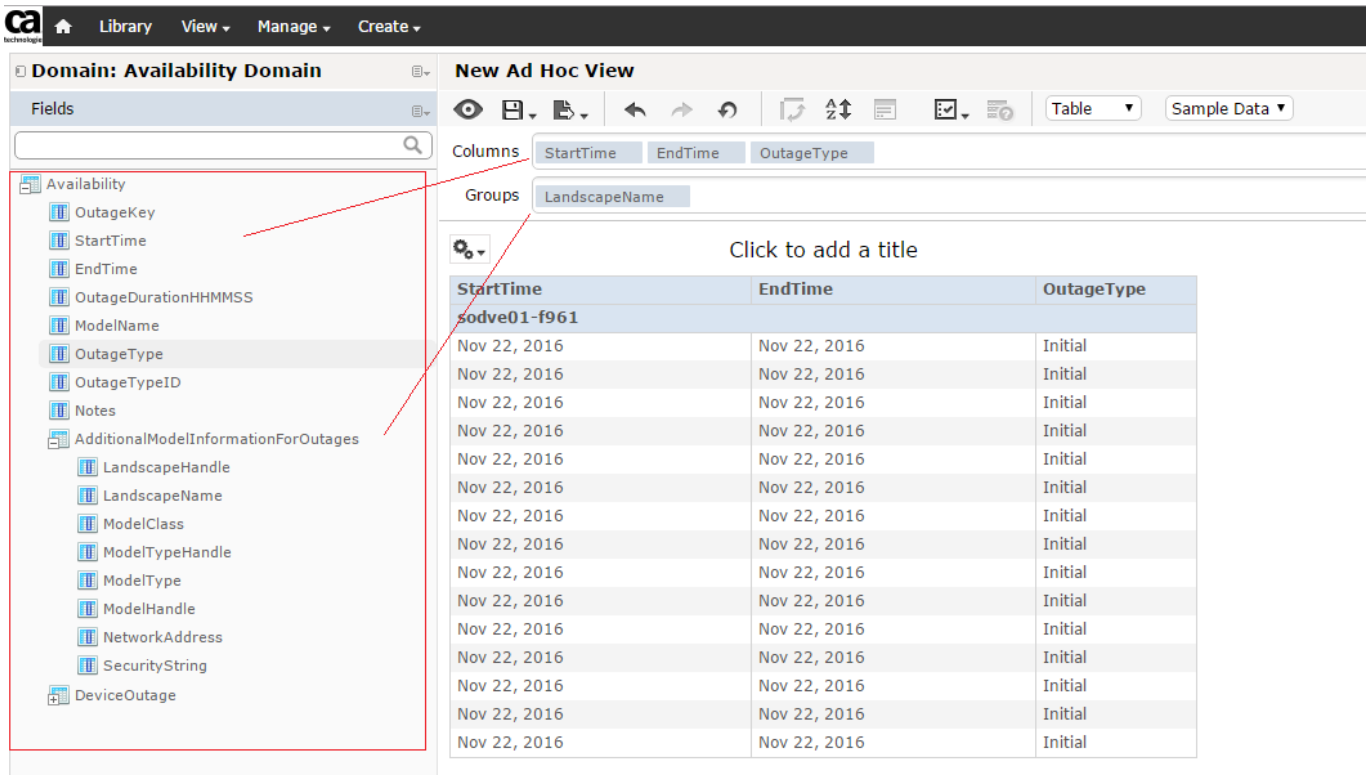
6. In the ‘Fields’ tab, select ‘Availability’ in the ‘Source’ box and click the  button to move it to ‘Selected Fields’ box and click OK.
You can see the Availability field in the Fields pane.



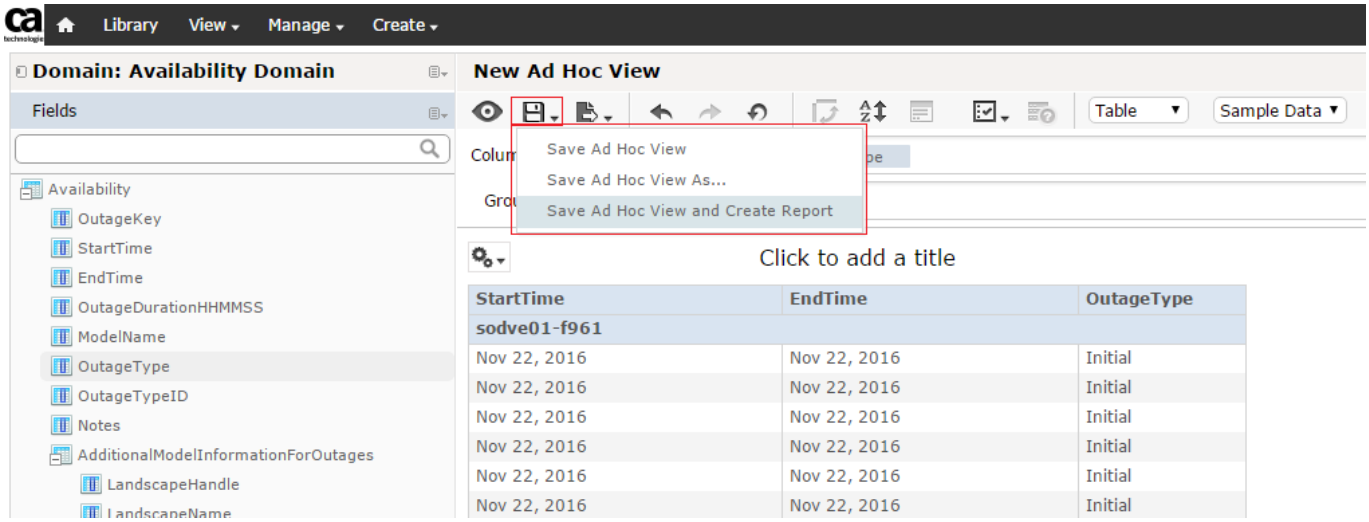
7. Select Table and Sample Data from the drop-down fields available on the toolbar.



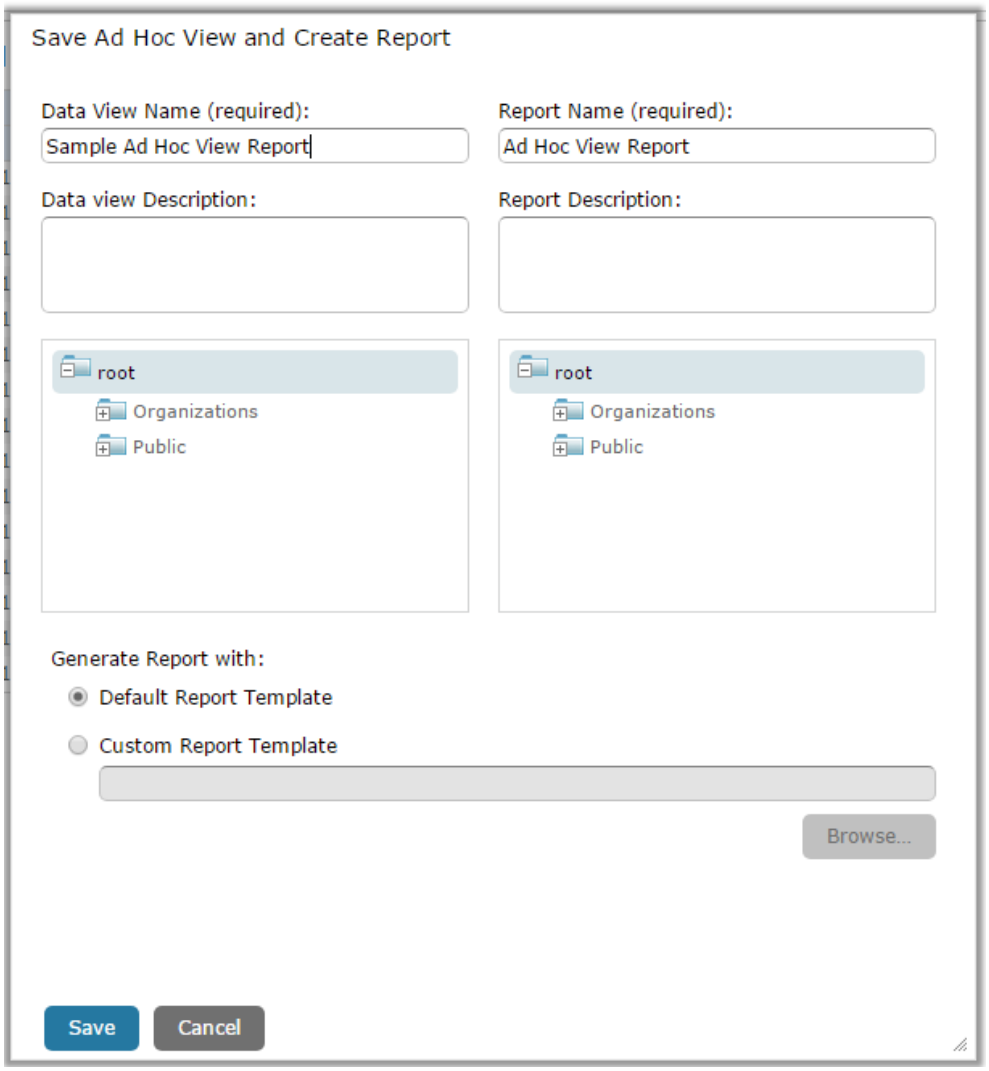
8. Expand the 'Availability' field in the left pane and add fields to Column and Group by drag and drop from the Availability field.
   For example: If you want to see StartTime, EndTime, and OutageType in the report, you can drag and drop these fields to Column. For Group, select fields from AdditionalModelInformation in Availability field.



9. To save the report, Click the Save button from the toolbar and then select Save Ad Hoc View and Create Report.



10. Provide information like Data View Name and Report Name in the Save Ad Hoc View and Create Report window and click Save.



The Ad Hoc View Report is saved in the selected location.

Scheduling Reports

You can schedule the reports to automate the generation and distribution of reports. Provide the details of the settings in JasperReports Server to schedule the reports.

For more information, see the Scheduling Reports section in the JasperReports Server documentation.

NOTE

Information! For scheduled reports, you receive emails with links to specific reports. The report links in the email may not work properly if they are referenced to 'http://localhost' instead of the real hostname (JasperReports Server). To set the real host name, update the 'report.scheduler.web.deployment.uri' attribute value in the 'js.quartz.properties' file, which is located at <jasperserver-pro>/WEB-INF/js.quartz.properties. The 'localhost' in the url should be replaced with server hostname, where the JasperReports Server is installed.

Example: report.scheduler.web.deployment.uri=http://localhost:8080/jasperserver-pro should be set to http://<jaser server hostname>/jasperserver-pro.

Managing Reports

You can save and export the reports in various formats. For more information, see the Exporting the Report section in the JasperReports Server documentation.

Enable Single Sign-On (SSO) Using Token-based Authentication in JasperSoft

Token-based authentication is a single sign-on solution from JasperReports Server that establishes a session between the Spectrum Report Manager and Jasper console. This solution allows you to open Jasper console session from a OneClick web console without being promoted for login credentials.

When you enable the SSO with JasperServer, a java keystore file is created. Using the java keystore file, Jasper generates a login token for login.

Follow these steps to achieve the SSO with token-based authentication in DX NetOps Spectrum-Jasper integration:

1. Navigate to OneClick, Administration tab, Report Manager, then select the Jasper Integration in the left panel.
2. Select the 'Enable SSO' checkbox, click Save.
   You can access the Jasper Console without providing any credentials. To verify, click the Jasper Console link on the OneClick toolbar. You are automatically logged in to the Jasper Console.

   NOTE
   Currently, the DX NetOps Spectrum and Jasper integration does not support enabling both SSO and LDAP together. Only either SSO or LDAP can be enabled.

Import and Export Data in JasperReports Server

Import and Export Through the Web UI

Import Data

1. Log in to the CABI JasperReports Server as a system administrator(for example, superuser).
2. Navigate to Manage, Server Settings, then click Import.
3. Select the content that you want to Import, then click Save.

Export Data

1. Log in to CABI JasperReports Server as a system administrator.
2. Navigate to Manage, Server Settings, then click Export.
3. Select the content that you want to export, then click Export.

Import and Export Through the Command Line

You can specify the required content to export or import in the command-line utility. Go to the folder *<jsinstall>*/buildomatic/*folder* and run the utility to export or import the content.

Import Data

To import resources from the *myExport.zip* catalog archive file, run the following command:

```
js-import --input-zip myExport.zip
```

Export Data

To export all resources in ca_samples (excluding users, roles, and job schedules) and their permissions to the *myExport.zip* file, run the following command:

NOTE

Depending on your environment, the following location (as shown in the example) changes:

"organization_1/ca_samples"

```
js-export --uris /organizations/organization_1/ca_samples --output-zip myExport.zip
```

Migrate CA Business Intelligence from Windows to Linux - DX NetOps Spectrum

Windows is unsupported with CA Business Intelligence 7.1.1. If you integrate with CA Business Intelligence on Windows, when you upgrade to 7.1.1, you must migrate your DX NetOps Spectrum and DX NetOps Spectrum content from the Windows CA Business Intelligence server to a Linux CA Business Intelligence server.

Follow these steps:

1. Export the content from the Windows CA Business Intelligence server:
   a. Log in to CA Business Intelligence on the Windows server as a system administrator (for example, superuser).
   b. Click **Manage**, **Server Settings**, and **Export**.
   c. Specify a file name for the export data.
   d. Select the content to export.
   e. Click **Export**
2. Install DX NetOps Spectrum content in Linux CABI server using the following command:
   ```
   java -jar spectumConfigInstaller.jar -install
   ```
3. Import the content to the Linux CA Business Intelligence server:
   a. Log in to CA Business Intelligence on the Linux server as a system administrator (for example, superuser).
   b. Click **Manage**, **Server Settings**, and **Import**.
   c. Select the content to import.
   d. Click **Import**
4. Perform DX NetOps Spectrum-Linux CABI integration.

Usernames and Passwords for CABI (JasperReports Server)

OneClick users are automatically added to CABI (JasperReports Server). The default JasperReports Server password is the username. You can change the default password using the 'Change password' option available on the login page.

If a default JasperReports Server password is set, contact the Spectrum Report Manager administrator to reset the password. For more information, see the Install Report Manager section.

The Administrator user (Ex: superuser) can add users in the JasperReports Server.

To add users, follow these steps:

1. Go to Manage, Users
2. Select the Organization from the left menu
3. Click Add User menu button

The Add User pop-up appears.

4. Enter details of the user such as User Name, User ID, Email and Password
5. Click Add User to <org name> button
   The user is added in the JasperReports Server
   The default role of the added user is 'ROLE_USER'. You can modify the user role using the 'Edit' button available in the Properties section for that user.

Troubleshooting for CABI JasperReports Server

Service Reports show incorrect down/ up percentage values

Symptom:

There are incorrect percentages values in the service reports when the 'Date Range' is chosen and when the end period value is the future date/time which is set under the Input Controls option.

Solution:
The end period value should always be set to a value which is less than or equal to the current date or time.

Jasper SSO access issue when Single Sign-On is enabled in both Jasper and DX NetOps Spectrum

Symptom: When Single Sign-On is enabled in Jasper Server and at the same time if you enable the Single Sign-On feature in DX NetOps Spectrum, you may face access issues for Jasper SSO.
Solution: To enable the Single Sign-On for both Jasper and DX NetOps Spectrum:

1. In the DX NetOps Spectrum machine, go to $Specroot\tomcat\webapps\spectrum\repmgr\admin folder
2. Copy the 'spectrum.jks' and 'spectrum.properties' files
3. In the Jasper machine, go to <<CABusinessIntelligenceINstalledDIR>>\apache-tomcat\webapps\jasperserver-pro\WEB-INF\config folder
4. Replace the existing 'spectrum.jks' and 'spectrum.properties' files with the copied files from DX NetOps Spectrum machine

Jasper report charts are not loading properly (going into processing)

Symptom: Jasper report which contains graphs such as pie chart or bar graph does not load and keeps on processing.
Solution: This issue is caused because of running Ad-Blocker within the Web browser like Firefox or Chrome.
We recommended you disable the extension for the JasperReports page, while running the reports to get the charts loaded properly in reports.

Unable to export report data (without Titles, Headers, Group Headers, Summary, Footers) to .csv format

Symptom: When I export the report data into CSV format, I want to filter out Title, Header, Group Header, Summary, and Footer information.
Solution:

1. On Jasper Server, navigate to :
   <<CA Business Intelligence Installed Directory>>\apache-tomcat\webapps\jasperserver-pro\WEB-INF\classes.
2. Open the jasperreports.properties file.
3. Add the following properties at the end of the file.

```
net.sf.jasperreports.export.csv.remove.empty.space.between.rows=true
net.sf.jasperreports.export.csv.remove.empty.space.between.columns=true
net.sf.jasperreports.export.csv.exclude.origin.band.1=pageHeader
```

```
net.sf.jasperreports.export.csv.exclude.origin.report.1=*
net.sf.jasperreports.export.csv.exclude.origin.band.2=pageFooter
net.sf.jasperreports.export.csv.exclude.origin.report.2=*
net.sf.jasperreports.export.csv.exclude.origin.band.3=columnHeader
net.sf.jasperreports.export.csv.exclude.origin.report.3=*
net.sf.jasperreports.export.csv.exclude.origin.band.4=columnFooter
net.sf.jasperreports.export.csv.exclude.origin.report.4=*
net.sf.jasperreports.export.csv.exclude.origin.band.5=lastPageHeader
net.sf.jasperreports.export.csv.exclude.origin.report.5=*
net.sf.jasperreports.export.csv.exclude.origin.band.6=summaryPageHeader
net.sf.jasperreports.export.csv.exclude.origin.band.7=groupHeader
net.sf.jasperreports.export.csv.exclude.origin.band.8=groupFooter
net.sf.jasperreports.export.csv.exclude.origin.band.9=reportHeader
net.sf.jasperreports.export.csv.exclude.origin.band.10=reportFooter
net.sf.jasperreports.export.csv.exclude.origin.band.11=lastPageFooter
net.sf.jasperreports.export.csv.exclude.origin.report.11=*
net.sf.jasperreports.export.csv.exclude.origin.band.12=summaryPageFooter
net.sf.jasperreports.export.csv.exclude.origin.band.13=summary
net.sf.jasperreports.export.csv.exclude.origin.report.13=*
net.sf.jasperreports.export.csv.exclude.origin.band.14=title
net.sf.jasperreports.export.csv.exclude.origin.report.14=*
net.sf.jasperreports.export.csv.parameters.override.IgnorePagination=true
```

4. Save the changes to the properties file and close.
5. Restart the Jasper Tomcat Server.

DX NetOps Spectrum logo stretches in the Jasper reports that are exported to Excel format

Symptom: While exporting the Jasper reports to Excel format, the DX NetOps Spectrum logo is stretched.
Solution: To prevent the DX NetOps Spectrum logo from stretching:

1. On the Jasper Server Machine
2. Open the file applicationContext,xml located at <Jasper Install Directory>\CA\SharedComponents\CA Business Intelligence\apache-tomcat\webapps\jasperserver-pro\WEB-INF.
3. Change the 'ignoreGraphics' property to 'false' and save the file.
4. Restart the Jasper Tomcat server.

'Install with default components such as Tomcat and Postgres option is grayed out while installing Jasper server

Symptom: While installing JasperReports Server 6.3 using GUI, the “Install with default components(Tomcat and Postgres….)” option grays out. The user is forced to select “Custom install”.
Solution:

1. Verify the host file entries in the '/etc/hosts' (for non-Windows platforms), %SystemRoot%\System32\drivers\etc\hosts (for Windows).
2. Correct the entries and save the file.
3. Restart the installation.

For more details, see the 'Host Name / IP Resolution' section in the CABI documentation.

JasperReports Server Integration fails

Symptom: JasperReports Server integration with DX NetOps Spectrum fails

Solution: To debug the integration fail issue, you must ensure that JasperReports Server parameters are given correctly in Jasper Integration page and also verify that the 'SRM - Core - Report Manager' Module is set to 'ON' in the Debug Controller to collect (To set this value go to OneClick, Administration, Debugging, and Web Module Debug Pages). Verify the OneClick Tomcat log file to identify the error.

Symptom: JasperReports Server integration with DX NetOps Spectrum fails
Solution: If you have installed the CABI jasper reports server that comes with any CA product other than DX NetOps Spectrum, then you must ensure that SpectrumProxy.war file is copied to $TOMCAT\webapps folder and the reportViewerMain.js file is copied to the folder.

JasperReports Server GUI Install screen shows that the product is already installed

Symptom: JasperReports Server GUI installation screen shows that the product is already installed
Solution: This happens when the uninstaller fails to clean up the registry entries of the jasper server. For windows, check the registry using regedt32 for CABI Jasper server entries. For Linux, check for the file $HOME/.cabijasper file and delete it if found.

A lot of postgres.exe processes even though I only started the server once

Symptom: You may see a number of postgres.exe processes running when started Postgres and tomcat of Jasper server (if Postgres is used as a repository of jasper server)
Solution: This is normal. PostgreSQL uses a multi-process architecture. In a fresh system with Postgres server, you may see anything from two to five processes. Once clients start to connect, the number of processes increases.

OC users don’t sync in jasper, some newly created users are missing

Symptom: Do not see all the DX NetOps Spectrum/OC users in jasper server
Solution: You can restart the OneClick server and verify it.

Hibernate errors appear when trying to run reports

Symptom: Getting hibernate error messages “Socket Write Error” and “Last Packet Not Finished” and unable to run reports.

```
Error Message:
org.springframework.transaction. CannotCreateTransactionException: Could not open Hibernate Session for
 transaction; nested exception is org.hibernate. TransactionException: JDBC being failed:
Error Message:
org.hibernate.TransactionException: JDBC begin failed:
OR
Error Message:
org.springframework.tranaction. CannotCreateTransactionException: Could not open Hibernate Session for
 transaction; nested exception is java.lang.AssertionError: Last packet not finished
Error Message:
java.lang.AssertionError: Last packet not finished
```

Reason: The c3p0 hibernation settings are not available

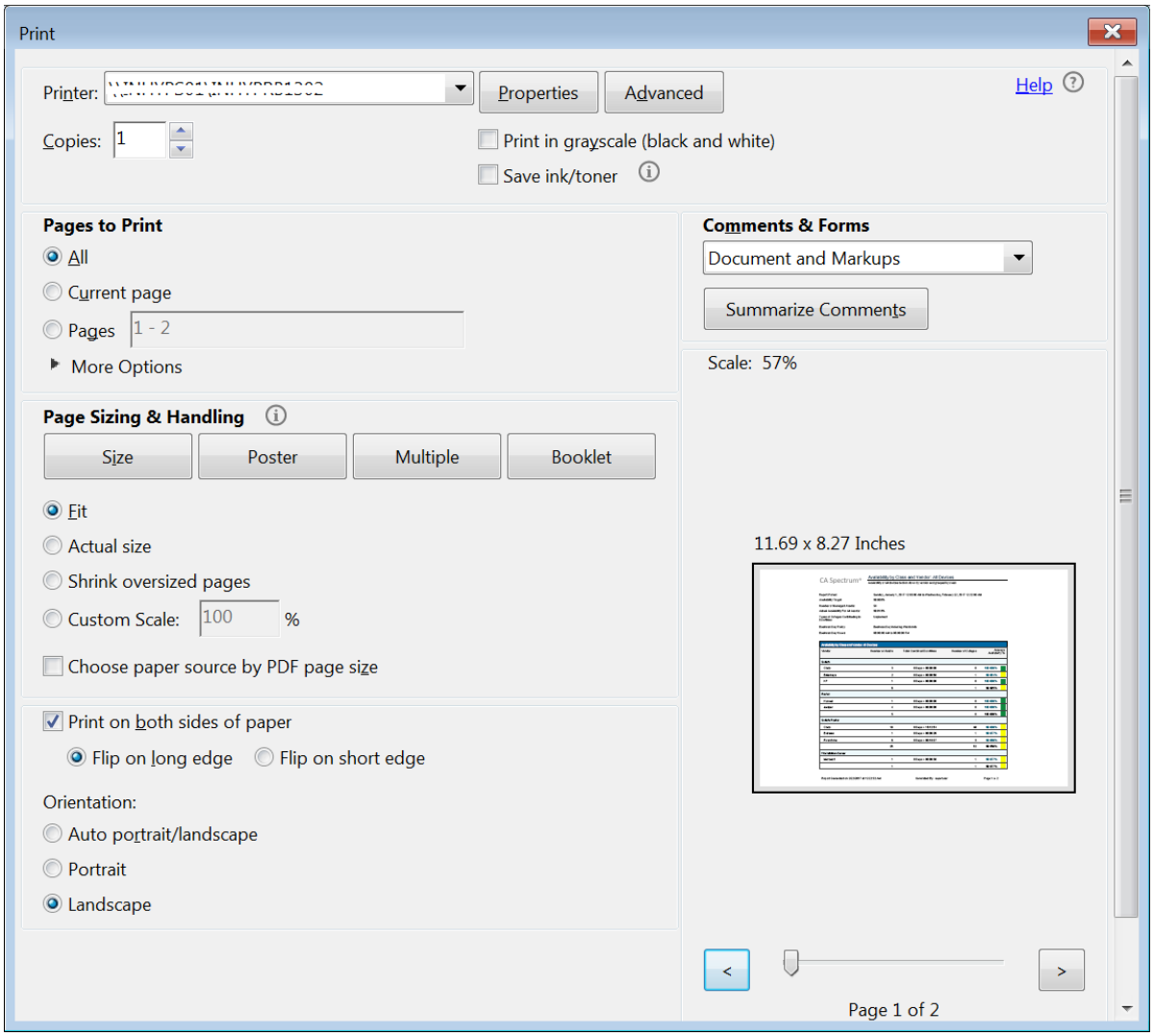
Solution: Add the following property key values (which are in bold letters) in the **hibernateProperties** in **<js-app>/jasperserver-pro/WEB-INF/applicationContext.xml** file and restart the CABI Tomcat to apply these changes. (Refer to the KB article for more information).

```
<property name="hibernateProperties">
      <props>
        <prop key="hibernate.dialect">${metadata.hibernate.dialect}</prop>
```

```
        <prop key="hibernate.show_sql">false</prop>
        <prop key="hibernate.generate_statistics">true</prop>
        <!--uncomment property below if a default schema should be specified such as for
 DB2-->
        <!--<prop key="hibernate.default_schema">${metadata.hibernate.default_schema}</
prop>-->
        <!--Cache Configurations-->
        <prop key="hibernate.cache.region.factory_class">
${hibernate.cache.region.factory_class}</prop>
        <prop key="net.sf.ehcache.configurationResourceName">/ehcache_hibernate.xml</
prop>
        <prop key="hibernate.cache.use_minimal_puts">false</prop>
        <prop key="hibernate.cache.use_query_cache">true</prop>
        <prop key="hibernate.jdbc.batch_size">20</prop>
        <prop key="hibernate.cache.use_second_level_cache">true</prop>
        <prop key="hibernate.cache.use_structured_entries">true</prop>
         <prop key="hibernate.c3p0.min_size">5</prop>
         <prop key="hibernate.c3p0.max_size">200</prop>
         <prop key="hibernate.c3p0.timeout">300</prop>
         <prop key="hibernate.c3p0.max_statements">500</prop>
         <prop key="hibernate.c3p0.idle_test_period">60</prop>
         <prop key="hibernate.c3p0.acquire_increment">2</prop>
         <prop key="hibernate.c3p0.testConnectionOnCheckin">true</prop>
      </props>
    </property>
```

Printed Jasper reports contain huge space above the header and below the footer

Symptom: Noticed huge space above the header and below the footer when printed the PDF report that is exported from Jasper.
Solution: To print the reports, you need to export/save the report as PDF and then use the print facility provided by the acrobat reader (do not try to print the report by opening it in the browser). You must select 'Fit To Page' option in the Print dialog and select the page orientation as “Landscape”. If you select 'Fit to Page', the acrobat reader scales the document to fit into a page and places it in the center. If you select 'Custom Scale' option and specify the same scale value - the document is scaled but placed at the page top. Refer to the following screenshot.



The JasperReports Server 6.3 login button is grayed out or the content page does not look correct in Internet Explorer 11

Symptom: When using IE 11 and Jasper Reports 6.3 either the login button is grayed out or if you are logged in, the page format looks incorrect.

Probable Cause: This is due to a setting on the Jasper server

Solution/Workaround:

Force the Edge mode for Internet Explorer by modifying the following section in the <Tomcat Install Directory>/WEBAPPS/JASPERSERVER-PRO/WEB-INF/decorators/decorator.jsp file on the JasperSoft Server:

<!--

<meta http-equiv="X-UA-Compatible" content="IE=8"/>

-->

Uncomment this section, and then set content to "IE=Edge" instead of "IE=8".

<meta http-equiv="X-UA-Compatible" content="IE=Edge"/>

SRM Multitenancy Support

NOTE

10.4.1 SRM Multitenancy is supported with CABI JasperReports Server 6.4.2, 6.4.3, and Unified CABI 7.1.1versions.

DX NetOps Spectrum includes SRM multitenancy support with Jaspersoft. This support enables modeling of multiple selected tenants to any number of devices or locators in a DSS environment. This modeling helps you effectively manage several devices in your network. Through SRM multitenancy, you can configure multiple tenants and can specify devices or locators to be mapped to, in the corresponding landscapes. Deploy a multitenant environment and aggregate, generate, and run reports for multiple tenants to manage and monitor the devices in your network.

Configure SRM Multitenancy

Prerequisites

1. Enable JasperServer Integration with DX NetOps Spectrum.

Following are the key functionalities and parameters of the SRM Multitenancy table:

- **Add Tenant:** Enable the checkbox for multitenancy support to add tenant/s.
- **Delete Tenant:** Select the checkbox in the field next to the tenant name and delete the tenant and its mapping.
- **Tenant Name:** Specifies the tenant name or organization name.
- **Locator:** Includes the routers, switches, pings you have specified in the XML file, which is used to filter out and map to the corresponding tenant name. These can be selected using the locator search drop-down.

  NOTE

  This locator search should be unique which you define in an XML file. The locator file name should not contain special characters ($,#,!%). It is permissible to have an underscore (_) in the file name. The user should create a folder 'SRMTenants' in the following path: <*SpectrumFolder*>*/tomcat/webapps/spectrum/WEB-INF/topo/Config* and then place the xml file inside the folder 'SRMTenants'. The locater search drop-down reads the xml placed in the 'SRMTenants' folder.
- **Landscapes**: Select either single or multiple DSS environments in which configuration should be enabled.
- **Save**: Save newly added tenant and locator searches in the corresponding landscapes.
- **Re-Map**: Refreshes or syncs the latest mappings of the tenants to locators based on any file changes related to mapping.

Follow these steps to deploy a Multitenant environment:

1. Open the OneClick Administration page.
2. Select the **Report Manager** tab on the top bar and select the **Jasper Multitenant Configuration** from the left panel.
3. Select the **Enable Multitenancy Support** checkbox to initiate multitenancy.
4. Select **Add Tenant** to add tenant and map it to the corresponding locator or device, router, switch, and so on, that you have specified in the XML file.
5. Select the respective landscape in a DSS environment.
6. After mapping the tenant to the locator, select **Save**. The multitenancy configurations are saved.

Generate Reports under the select Tenant Name:

1. Go to the **Jasper Console** on the top bar of the OneClick Administration page. The CA Business Intelligence login page appears.
2. Enter the **organization** details, which are the tenant name you have specified in the configuration table. Log in using DX NetOps Spectrum credentials.

NOTE

You cannot run reports using superuser credentials if you have enabled SRM Multitenancy.

The JasperServer home page appears.

3. Select **Repository** from the **View** tab on the top left bar of the page.
4. Select **Reports** from the left panel under capabilities that are found under the organization name.
5. You can now run and generate reports under the select Tenant Name. See How to run reports using the JasperReports Server for more information.

WARNING

Now with SRM Multitenancy ensure privacy and safety of your data! This feature restricts unwanted access to your schedule by preventing the schedule from being saved in a public folder.

Users/Admins cannot save their schedule in 'public/ca/Spectrum' folder and saving in such public folders is restricted through this enhancement. Users/Admins can save schedule in their own tenant folders **only**.

Upgrade Options from Jaspersoft 6.3.0 to 6.4.2/6.4.3

NOTE

When you upgrade from Jaspersoft 6.3.0 to 6.4.2/6.4.3 with CA Spectrum 10.3 (or later), the SSO, LDAP, and email functionalities get disrupted. To ensure a successful upgrade to Jaspersoft 6.4.2/6.4.3, you must reconfigure these settings.

To ensure a successful upgrade to Jaspersoft 6.4.2/6.4.3 with the features that are mentioned in the note, perform the following actions:

To ensure the SSO feature is functional in Jasper 6.4.2/6.4.3:

Copy the ‘config’ folder from the <<CA Business Intelligence Installed Directory>>/apache-tomcat-bkp/webapps/jasperserver-pro/WEB-INF/ location to <<CA Business Intelligence Installed Directory>>//apache-tomcat/webapps/jasperserver-pro/WEB-INF/

To ensure the LDAP feature is functional in Jasper 6.4.2/6.4.3:

Refer to the CABI 6.4.2/6.4.3 LDAP administration and configuration page**.** For further information, refer to the upgrade section in the Jasper communities page.

To ensure the email feature is functional in Jasper 6.4.2/6.4.3

Copy the js.quartz.properties file from the <<CA Business Intelligence Installed Directory>>/apache-tomcat-bkp/apache-tomcat-bkp\webapps\jasperserver-pro\WEB-INF\ location to <<CA Business Intelligence Installed Directory>>/apache-tomcat-bkp/apache-tomcat\webapps\jasperserver-pro\WEB-INF\

NOTE
For more information, see the configuring report schedular section on Jasper communities page.

DX NetOps Spectrum with Unified Dashboards and Reporting for Infrastructure Management

NOTE

To use the Unified Dashboards and Reporting for Infrastructure Management capabilities, you must upgrade to 10.2.2 OR upgrade to 10.2.1 and install the 10.02.01.PTF_10.2.117 on top of 10.2.1. You must have a new

installation of the CA Business Intelligence JasperReports Server for Unified Dashboards and Reporting for Infrastructure Management software. For more information, see the Prerequisites section.

The Unified Dashboards and Reporting for Infrastructure Management solution allows you to share a single CA Business Intelligence (CABI) server instance with multiple CA Agile Operations products. CABI is a powerful tool that allows users to get quick answers to their questions through self-service reports, dashboards, and analysis. For more information see Unified Dashboards and Reporting for Infrastructure Management.

By installing 10.2.2 or 10.2.1 (+ 10.02.01.PTF_10.2.117), you can share a single instance of CABI JasperReports Server for Unified Dashboards and Reporting with the following CA Agile Operations products.

- Service Operations Insight (SOI)
- Unified Infrastructure Management (UIM)
- DX NetOps Performance Management (DX NetOps PM)

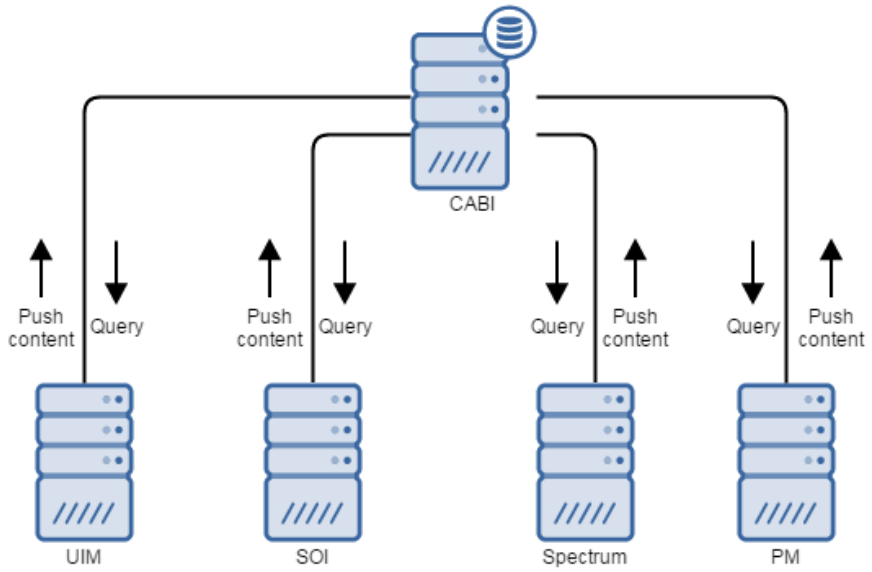
Following are the major benefits of shared CABI JasperReports Server deployment:

- Reduces the number of CABI instances, hardware footprint that you need to deploy and maintain for different CA products.
- Enables you to view dashboards and dashlets from multiple CA products, which provide you better insight into your business operations.

The CABI JasperReports Server installer is installed on a stand-alone CABI server and content from multiple CA products is uploaded.

Prior to this solution, CABI was limited to individual CA products, which each product using its own CA Business Intelligence JasperReports Server (CABI Server) to collect data and generate reports.

The following diagram shows an example of how the Unified Dashboards and Reporting for Infrastructure Management solution works with multiple CA products.



Prerequisites

- The CABI Server must have a new installation of the CA Business Intelligence JasperReports Server for Unified Dashboards and Reporting for Infrastructure Management software. You can download the **CABI JasperReports**

Server for Unified Dashboards and Reporting for Infrastructure Management from support.ca.com. A CA Support login is required to download the solution. For information about installation, see CA Business Intelligence JasperReports Server.

- Ensure that you are using 10.2.2 OR 10.2.1 (+ 10.02.01.PTF_10.2.117). You can download the 10.02.01.PTF_10.2.117 (CA Spectrum 10.2.1 PTF FOR UNIFIED DASHBOARDS AND REPORTING) from support.ca.com. A CA Support login is required to download the solution.

Integrate CABI JasperReports Server with DX NetOps Spectrum

The CABI JasperReports Server must be installed before integrating it with DX NetOps Spectrum. For detailed installation instructions, refer to the CA Business Intelligence JasperReports® Server documentation.

After installing the CABI JasperReports Server, you need to configure the integration between CABI JasperReports Server and Spectrum Report Manager.

Follow these steps to configure and activate the integration between CABI JasperReports Server and Spectrum Report Manager:

1. Log in to the computer where the CABI JasperReports Server is installed.
2. Launch the DX NetOps Spectrum OneClick web server on a Web browser.
3. In the DX NetOps Spectrum OneClick Web server, navigate to > Administration > Report Manager > Report Manager Admin Tools> Jasper Integration page.
4. Download and install the '**Integration Components**'.

   NOTE

   Information! The Integration Components is a JAR file, which consists of the necessary binaries to integrate the JasperReports Server with DX NetOps Spectrum and run reports. Installing the Integration Components deploys the binaries on the Jasper Server and help to generate the reports.
5. Specify and save the Jasper Server parameters in the Jasper Integration page.
   You have successfully configured and activated the integration between CABI JasperReports Server and Spectrum Report Manager.

For the detailed integration instructions, see DX NetOps Spectrum and CABI JasperReports Server Integration for Unified Dashboards and Reporting.

View Dashboards in CABI JasperReports Server

The CABI JasperReports Server displays multiple dashboards for DX NetOps Spectrum data. A dashboard is comprised of various dashlets, which are graphical representation of specific types of data. Following are the default dashboards that are displayed for DX NetOps Spectrum:

- Active Alarms Summary
- Alarm Summary
- Alarm Trends
- Assets Summary
- Device Availability Summary
- Events Summary
- NCM Dashboard
- Ports Summary and Chassis Assets
- Services and SLA Summary

For more information, see Dashboards in CABI JasperReports Server.

Out of the Box Reports and Domains

For the list of available out of the box reports, see Available Out of the Box Reports and Domains.

Run reports using the JasperReports Server

Important! You must log in to JasperReports Server to access reporting functionality.

Using the JasperReports Server you can generate general reports, ad-hoc reports. For more information, see How to run reports using the JasperReports Server.
You can also, schedule reports and manage reports such as save and export the reports to various formats. For details, see the Scheduling Reports section and Exporting the Report section in the JasperReports Server documentation.

Integration with CABI JasperReports Server for Unified Dashboards and Reporting

Instructions for 10.3 (or later)

After installing the JasperReports Server and Spectrum Report Manager (SRM), you need to enable the integration between JasperReports Server and SRM.

DX NetOps Spectrum and CABI JasperReports Server Integration

The Jasper Integration page in the DX NetOps Spectrum OneClick administration page allows you to enter the required parameters and enable the integration between DX NetOps Spectrum and CABI JasperReports Server.

Prerequisites for the Integration

Prior to integrating SRM with CABI JasperReports Server, you must download and install the 'Integration Components' on your CABI JasperReports Server. The 'Integration Components' lays down the DX NetOps Spectrum specific content in CABI JasperReports Server and helps to successfully enable the integration. If you do not install the integration components, the integration may not work properly.

Installing the Integration Components

Follow these steps to install and configure the Integration Components on CABI JasperReports Server:

1. Launch the DX NetOps Spectrum OneClick web console and click the Administration tab.
2. Click the Report Manager link.
3. Select the Jasper Integration option from the Report Manager Admin Tools.
   The Jasper Integration page appears.
4. Click the 'Integration Components' link on the Jasper Integration page.
   The spectrumConfigInstaller.jar file is downloaded to the Downloads folder in your computer.

   NOTE

   If the JasperReports server is on a Linux computer and it does not support GUI, then download the spectrumConfigInstaller.jar from a Windows computer as mentioned in Step 4. Place the downloaded spectrumConfigInstaller.jar file on the Jasper Linux computer and proceed with the following steps.
5. Using the command prompt, run the following command:
   <CABI Install folder path>\java\bin\java -jar <spectrumConfigInstaller.jar file path>\spectrumConfigInstaller.jar -install
   Example: <CABI Install folder path>\java\bin\java -jar C:\Users\Admin\Downloads\spectrumConfigInstaller.jar -install
   Enter the CABI Apache-Tomcat Home Location:
   Example: C:\Program Files\CA\SC\CA Business Intelligence\apache-tomcat

Enter the CABI Webapp Name (Default: jasperserver-pro):
Ex: jasperserver-pro
The setup progress takes some time. During this process, the following files are deployed at jasperTomcat on the JasperReports Server:
/webapps/jasperserver-pro/optimized-scripts/bower_components/jrs-ui/src/reportViewer/reportViewerMain.js
/webapps/SpectrumProxy.war
/webapps/jasperserver-pro/WEB-INF/applicationContext-WebServiceDataSource.xml
/webapps/jasperserver-pro/WEB-INF/bundles/webserviceds.properties
/webapps/jasperserver-pro/WEB-INF/lib/customDatasource_WebService_JRS_wrapper.jar
/webapps/jasperserver-pro/WEB-INF/lib/fluent-hc-4.2.1.jar
/webapps/jasperserver-pro/WEB-INF/lib/WebServiceDataAdapter.jar
/webapps/jasperserver-pro/WEB-INF/bundles/DashboardBundle.properties
/webapps/jasperserver-pro/WEB-INF/lib/spectrum_utils.jar
After the pre-Installation process is complete, restart the Jasper tomcat server to complete the configuration.

NOTE

Once Jasper server is up and running click on Re-deploy from the Jasper Integration page.

Uninstalling the Integration Components

Follow these steps to uninstall the Integration Components on CABI JasperReports Server:

1. Uninstall the changes in Jasper Server, using the following command:
   <CABI Install folder path>\java\bin\java -jar <spectrumConfigInstaller.jar file path>\spectrumConfigInstaller.jar -uninstall
2. Provide the information for prompted input fields for CABI such as Server Host name, Tomcat Protocol (http/https), Tomcat Port Number, Tomcat Server Location, Jasper Server Webapp Name and user password.
   After the successful uninstallation, the 'Done' message appears. You cannot see any DX NetOps Spectrum content (such as Dashboards, Reports, Repository etc.) in the CABI JasperReports Server.
   Following is an example to uninstall the 'Integration Components':
   C:\Program Files\CA\SharedComponents\CA Business Intelligence\java\bin\java -jar C:\Users\Admin\Downloads\spectrumConfigInstaller.jar -uninstall
   Are you sure you want to uninstall the spectrum content (y/n)? **y**
   CA Business Intelligence Server Hostname: **<host name>**
   CABI Tomcat Protocol (http/https): **http**
   CABI Tomcat Port Number: **8080**
   CABI Tomcat Server Location: **C:\Program Files\CA\SharedComponents\CA Business Intelligence2\apache-tomcat**
   Jasper Server Webapp Name (default: jasperserver-pro): **jasperser-pro**
   CABI Superuser Password: <**Superuser password**>
   Done

NOTE

After uninstalling the Integration Components, the integration between DX NetOps Spectrum and CABI JasperReports Server will not work. If you want to enable the integration, you must install the Integration Components again.

CABI JasperReports Server Integration

To enable integration between DX NetOps Spectrum and JasperReports Server, provide the Jasper Server connection details on the integration page.

NOTE

Before proceeding with the integration, ensure that you have installed the 'Integration Components' on your CABI JasperReports Server. If you do not install the integration components, the integration may not work properly. For more information, see the 'Prerequisites for the Integration' section.

Follow these steps:

1. Open the OneClick Administration page.
2. Click the Report Manager tab.
3. Select the Jasper Integration option from the Report Manager Admin Tools.
4. Specify the parameters that are used to communicate with the JasperReports server:
   - **Jasper Server Host name**
   - Specify the host name of your CABI JasperReports Server instance if it is not the same server as DX NetOps Spectrum Tomcat.
   - **Jasper Tomcat Port**
     Specify the port where Jasper Tomcat is running. The default port value is 8080.

     NOTE

     If the Jasper and OneClick are on the same server, the Tomcat port cannot be the same port that DX NetOps Spectrum Tomcat uses.
   - **JasperServer Webapp Name**
     Specify the Jasper server webapp name that is given during installation of Jasper server. Default is 'jasperserver-pro'.
   - **Jasper Admin User**
   - Specify the Jasper Admin User ID.
     The default User ID is 'superuser'.
   - **Jasper Admin Password**
     Enter the password for the Admin User ID in Jasper. The default password is 'superuser'.
   - **Jasper Integration**
     Select the Enable radio button.
     If Disable is selected and saved, DX NetOps Spectrum Tomcat no longer integrates with CABI Jasper instance.
   - **Enable SSO**
     Select this check box to enable single sign-on solution(SSO) from JasperReports Server, which establishes session between the Spectrum Report Manager and Jasper console. If you enable the SSO, you can open Jasper console session from a OneClick web console without providing any login credentials.

     NOTE

     The Single Sign-On solution does not work when you enable the option Use SSL with Jasper Server.
   - **Use SSL with Jasper Server**
     Select this check box to integrate with Secure Sockets Layer (SSL) enabled JasperReports Server.To know more about Secure Sockets Layer (SSL) and how to enable it on JasperReports Server, see Using SSL in the Web Server. (Supported only when the JasperReports Server runs on Windows 2012)
5. Click Save to enable the integration.
6. On the OneClick home page, click 'Jasper Console' to launch the JasperReports Server.

   NOTE

   This process can take some time. During this process, all the Spectrum Report Manager report content is exported from the OneClick server into CABI JasperReports Server. Therefore, do not cancel or navigate away from this page until you get a success message.

After you configure the integration, the Spectrum Report Manager report content is installed and can connect to the CABI JasperReports Server reporting database. The menubar 'JasperConsole' link now launches the CABI JasperReports Server web applications on the CABI instance that you specified.

NOTE

If you disable the integration, reporting and report administration capabilities are disabled. However, disabling the integration does not cause Spectrum Report Manager to stop collecting and managing data from the monitored SpectroSERVERs.

After successful integration, you can see the following DX NetOps Spectrum organization and default users in Jasper:

- jasperadmin/jasperadmin
  ROLE_ADMINISTRATOR
- joeuser/joeuser
  ROLE_USER
- spectrum/spectrum
  ROLE_ADMINISTRATOR
  ROLE_USER

Redeploy

Re-deploy the reports to repair the existing report or to update to a newer version. After upgrading to a new version of DX NetOps Spectrum, click the Re-deploy button to sync up the Jasper reports with DX NetOps Spectrum upgrade fixes, enhancements, and new reports.

Migrate Data to Unified Dashboards and Reporting for Infrastructure Management

NOTE

To use the Unified Dashboards and Reporting for the infrastructure management capabilities, upgrade to 10.3 (or later) or 10.2.x OR upgrade to 10.2.1 and install the 10.02.01.PTF_10.2.117 patch on top of 10.2.1 (for example, if upgraded to 10.2.1). You can download the patch 10.02.01.PTF_10.2.117 (CA SPECTRUM 10.2.1 PTF FOR UNIFIED DASHBOARDS AND REPORTING) from support.broadcom.com. A Support login is required to download the patch.

After installing 10.3 (or later) or 10.2.x (+ 10.02.01.PTF_10.2.117), you can share a single instance of CABI JasperReports Server with multiple CA Agile Operations products. For more information, see DX NetOps Spectrum with Unified Dashboards and Reporting for Infrastructure Management.

If you previously configured DX NetOps Spectrum to work with CA Business Intelligence (CABI) server, you can migrate data from the standalone CABI to a shared CABI instance. Use the following process to migrate data from a standalone CABI JasperReports Server to a shared CABI Server.

NOTE

The standalone CABI instance version must match the version of the shared CABI instance.

Follow these steps:

1. Export data from the standalone CABI server.
   a. Log in to the standalone CABI instance as a system administrator (for example, superuser).
   b. Navigate to Manage, Server Settings, and click Export.
   c. Specify the Export Data File Name.
   d. Select the content that you want to export, and then click Export.

2. Import the exported report data into the shared CABI server.
   a. Log in to the shared CABI server as a system administrator.
   b. Navigate to Manage, Server Settings, then click Import.
   c. Select the file to import, then click Import.

Dashboards in CABI JasperReports Server

NOTE

To use the Unified Dashboards and Reporting for Infrastructure Management capabilities, upgrade to 10.3 (or later) or 10.2.x OR upgrade to 10.2.1 and install the 10.02.01.PTF_10.2.117 on top of 10.2.1.

After the successful integration of DX NetOps Spectrum with CABI JasperReports Server, user can view dashboards on DX NetOps Spectrum data. A dashboard is comprised of various dashlets, which are a graphical representation of specific types of data.

Following are the default dashboards that are available for DX NetOps Spectrum:

- Alarm Distribution Summary
- Alarm Summary
- Alarm Trends
- Assets Summary
- Device Availability Summary
- Events Summary
- NCM Dashboard
- Ports Summary and Chassis Assets
- Services and SLA Summary
- DX NetOps Spectrum Monitoring Status
- DX NetOps Spectrum – Product Usage
- DX NetOps Spectrum Overview

NOTE

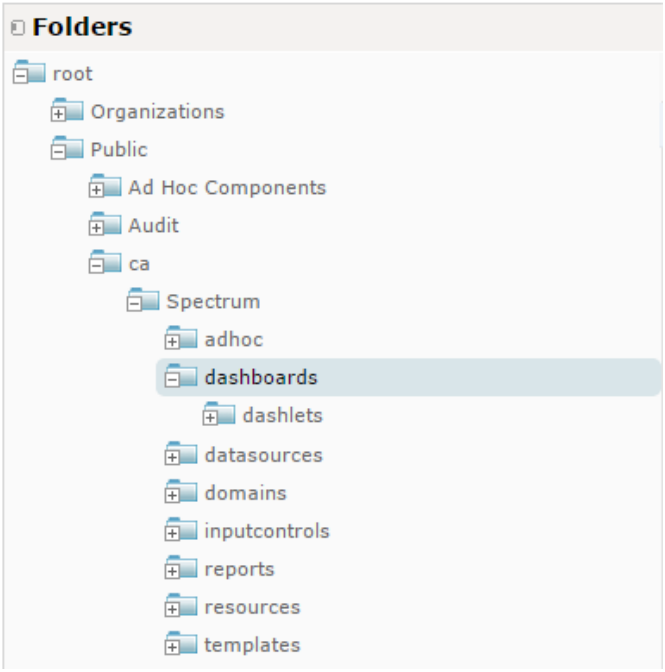
10.3.2 offers SSL support for the DX NetOps Spectrum Overview Dashboard. To know more about enabling this feature, see the SSL support for the DX NetOps Spectrum Overview Dashboard page.

DX NetOps Spectrum Content Folder Structure:

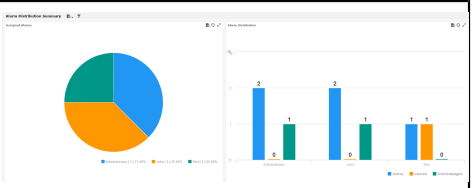
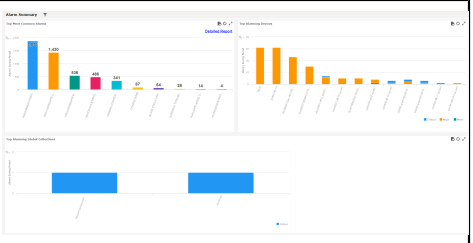
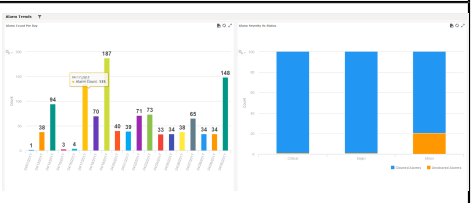
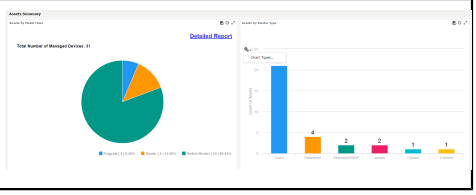
The DX NetOps Spectrum product content such as ad-hoc, dashboards, data sources, domains, input controls, reports, and resources is now located in the public folder in CABI JasperReports Server. For example, the new folder structure for the DX NetOps Spectrum content is Public > ca > Spectrum > dashboards (see the following image).

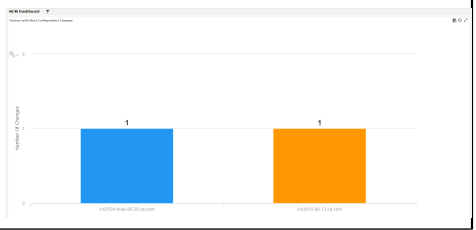
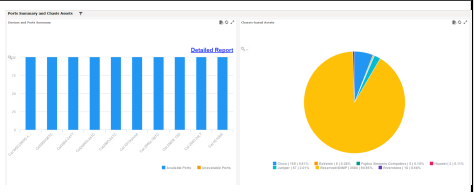
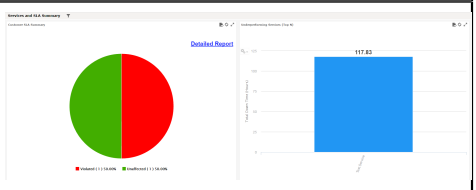
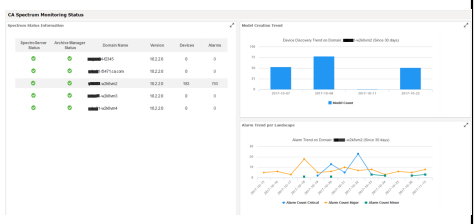
NOTE

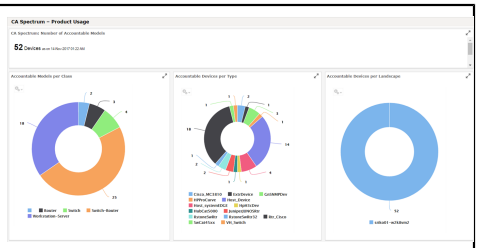
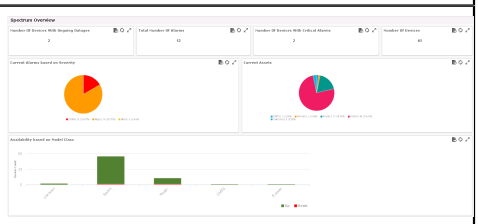
After installing the 10.2.2 Service Pack OR 10.2.1 (+ 10.02.01.PTF_10.2.117), for Unified Dashboards, all the DX NetOps Spectrum content in CABI JasperReports Server is saved under the 'Public' folder, previously it was saved under the 'root' folder. For example, now the Jasper reports are saved to Public > ca > Spectrum > reports. The old folder structure (root > Organizations > spectrum > capability > and reports) remains but no new content is saved under it after the upgrade and for future releases.



The following table displays a list of DX NetOps Spectrum dashboards, dashlets with screenshots:

Dashboard	Dashlets	Screenshot
Name: **Alarm Distribution Summary** Path: /public/ca/spectrum/dashboards/alarm_distribution_summary Description: DX NetOps Spectrum dashboard to show current active alarms and alarm distributions among users.	The Alarm Distribution Summary dashboard consists of the following dashlets: • Assigned Alarms • Alarm Distribution	
Name: **Alarm Summary** Path: /public/ca/spectrum/dashboards/alarm_summary Description: DX NetOps Spectrum dashboard to show the top alarming devices, global collection and most common alarm types.	The Alarm Summary Summary dashboard consists of the following dashlets: • Top Most Common Alarms • Top Alarming Devices • Top Alarming Global Collections	
Name: **Alarm Trends** Path: /public/ca/spectrum/dashboards/alarm_trends Description: DX NetOps Spectrum dashboard to show alarms histogram and alarms distribution based on severity.	The Alarm Trends dashboard consists of the following dashlets: • Alarm Count Per Day • Alarm Severity Vs Status	
Name: **Assets Summary** Path: /public/ca/spectrum/dashboards/assets_summary Description: DX NetOps Spectrum dashboard to show assets based on device class and vendor information.	The Assets Summary dashboard consists of the following dashlets: • Assets by Model Class • Assets by Vendor Type	

Name: **Device Availability Summary** Path: /public/ca/spectrum/dashboards/device_availability_summary Description: DX NetOps Spectrum dashboard to show devices availability information including devices with most number of outages and least available devices.	The Device Availability Summary dashboard consists of the following dashlets: • Least Available Devices (Top N) • Availability by Class and Vendor • Devices with Most Outages	
Name: **Events Summary** Path: /public/ca/spectrum/dashboards/events_summary Description: DX NetOps Spectrum dashboard to show top event generating models information and most common event types information. **Note:** To know more about Events Summary Dashboards, see Sample Performance Test Results for Event Summary Dashboard section.	The Events Summary dashboard consists of the following dashlets: • Top Event Count by Model • Top Most Common Events	
Name: **NCM Dashboard** Path: /public/ca/spectrum/dashboards/NCM_dashboard Description: DX NetOps Spectrum dashboard to show devices having most number of configuration changes.	The NCM Dashboard consists of the following dashlets: • Devices with Most Configuration Changes	
Name: **Ports Summary and Chassis Assets** Path: /public/ca/spectrum/dashboards/ports_summary_and_chassis_assets Description: DX NetOps Spectrum dashboard to show ports and chassis based assets summary information.	The Ports Summary and Chassis Assets dashboard consists of the following dashlets: • Devices and Ports Summary • Chassis-based Assets	
Name: **Services and SLA Summary** Path: /public/ca/spectrum/dashboards/services_and_SLA_summary Description: DX NetOps Spectrum dashboard to show SLA summary by customer and under performing services information.	The Services and SLA Summary dashboard consists of the following dashlets: • Customer SLA Summary • Underperforming Services (Top N)	
Name: **DX NetOps Spectrum Monitoring Status** Path: /public/ca/spectrum/dashboards/CA_Spectrum_Monitoring_Status Description: DX NetOps Spectrum dashboard to show running status of SpectroSERVER and Archive Manager on each domain. The number of models being discovered in DX NetOps Spectrum and alarms raised for the last 30 days.	The DX NetOps Spectrum Monitoring Status dashboard consists of the following dashlets: • DX NetOps Spectrum Status Information • Model Creation Tren • Alarm Trend per Landscape	

Name: **DX NetOps Spectrum – Product Usage** Path: /public/ca/spectrum/dashboards/CA_Spectrum_–_Product_Usage Description: DX NetOps Spectrum dashboard to show the number of accountable models being modeled and monitored.	The DX NetOps Spectrum Product Usage dashboard consists of the following dashlets: • DX NetOps Spectrum: Number of Accountable Models • Accountable Models per Class	
Name: **DX NetOps Spectrum Overview** Path: /public/ca/spectrum/dashboards/spectrum_overview Description: DX NetOps Spectrum real-time dashboard to show current assets/alarms/availability of devices being managed by DX NetOps Spectrum.	The DX NetOps Spectrum Product Usage dashboard consists of the following dashlets: • Number of Devices with Ongoing Outages • Total Number of Alarms • Number of Devices with Critical Alarms • Number of Devices • Current Alarms based on Severity • Current Assets • Availability based on Model Class	

Sample Performance Test Results for Event Summary Dashboard

Following are the two examples for the Event Summary Dashboard with tested configuration and results.

Test Result 1

SRM and Jasper machine details:

	Ram	**OS**	**Machine Type**
SRM Machine	16 GB	Windows 2012	Virtual
Jasper Machine	8 GB	Windows 2012	Virtual

Events details:

Event Count	**Able to load events in Jasper**	Result
500,000	Yes	Able to view the dashboard
> 500,000	No	Unable to load the dashboard

NOTE
Recommended space for Japser machine is 16 GB with minimum 8 GB free space.

Test Result 2

SRM and Jasper machine details:

	Ram	**OS**	**Machine Type**
SRM Machine	64 GB	Linux	Physical
Jasper Machine	12 GB	Windows 2008	Virtual

Event Details:

Events Count	Able to load events in Jasper	Comments
10,000,000	Yes	Able to view the dashboard
> 10,000,000	No	Observed Inconsistent performance

Dashboard Actions:

The following actions are available in a dashboard:

- **Chart or Graph Legend:** Select items in the chart or graph legend to hide a specific group of data.
- **Export (Export icon):** Click to export the content to the selected output.
- **Hover:** Hover your mouse over dashboard charts and graphs to view specific data points.
- **Maximize (Maximize icon):** Click to open the dashlet in a larger view.
- **Minimize (Minimize icon):** Click to return to the dashboard view.
- **Refresh (Refresh icon):** Click to refresh the dashlet contents.
- **Detailed Report (Detailed Report link):** Click to view more detailed information.

SSL Support for the DX NetOps Spectrum Overview Dashboard

10.3.2 introduces support for secure communication between DX NetOps Spectrum Overview Dashboard to the SRM Server allowing DX NetOps Spectrum users to have an HTTPS option for the DX NetOps Spectrum Overview dashboard.

Steps to Enable SSL support for the DX NetOps Spectrum Overview Dashboard

Following are the prerequisites and the procedure to use an encrypted communication (HTTPS protocol) between DX NetOps Spectrum and the SRM by importing the SSL/HTTPS certificate to achieve the SSL support for the DX NetOps Spectrum Overview Dashboard.

Prerequisites:

For first time integration, once the HTTPS configuration is complete, enable the DX NetOps Spectrum and CABI JasperReports Server Integration for Unified Dashboards and Reporting and the **Single Sign-On** (SSO) option, to make REST calls from the DX NetOps Spectrum overview dashboard. Ensure that you perform **SpectrumConfigInstaller.jar** to run on the Jasper machine.

For an existing integration, once the HTTPS configuration is complete, select the **Re-deploy** option, for the integration enhancements to sync, and follow the steps to export and import the self-signed certificate.

Procedure to Export and Import the Certificate:

1. Export the self-signed certificate from DX NetOps Spectrum OneClick machine where the HTTPS is enabled and then import it to the Jasper machine, to run the DX NetOps Spectrum overview dashboard. To export the self-signed certificate:
   a. On the OneClick web server host, open a bash login and make the following changes to the **$SPECROOT/Java/bin directory**.
   b. Run the following command.
   ```
   keytool -export -alias tomcatssl -keystore c:/win32app/Spectrum/custom/keystore/cacerts -file
    tomcat.cer
   ```
   c. Enter keystore password: <Enter the password as "changeit"> , the certificate is stored in the file <tomcat.cer>
   d. A certfile is created on **$SPECROOT/Java/bin directory**.
   e. Copy the exported .cer file certificate to the Jasper machine location **C:\Program Files\CA\SC\CA Business Intelligence\java\bin .** This is the directory where the keytool exists in Jasper.

2. Import this certificate to Jasper cacert by opening a cmd prompt or bash -login in the JasperServer and redirect to **C:\Program Files\CA\SC\CA Business Intelligence\java\bin** folder to run the following command:
   ```
   keytool.exe -import -alias tomcatssl -keystore "C:\Program Files\CA\SC\CA Business Intelligence\java\lib\security\cacerts" -file tomcat.cer
   ```
   a. Enter keystore password:<use the password "changeit">
3. Restart the Jasper Tomcat server and launch the Jasper console from the DX NetOps Spectrum administration page and run the DX NetOps Spectrum overview dashboard.

(Optional) Log in Using Non-Fully Qualified Domain Name

SSL security forces you to use the fully qualified domain name of your OneClick server for login. For example: `https://oneclick.ca.com/spectrum` . To log into the non-fully qualified domain name (for example: `https://oneclick/spectrum`), or a DNS entry that is different than the local OneClick server name, use a SAN (Subject Alternate Name) with the -ext option:

```
./keytool -genkey -alias tomcatssl -keyalg RSA -keysize 2048 -ext SAN=dns:oneclick -keystore c:/win32app/Spectrum/custom/keystore/cacerts
```

Troubleshooting CABI JasperReports Server Integration

Unified JasperSoft reports integrate successfully with DX NetOps Spectrum, but shows no reports in the repository.

Symptom:

CABI and DX NetOps Spectrum integrates successfully, but then shows no reports in the repository.

Solution:

To avoid performance problems of the default values, add the following changes to the mySQL parameters:

1. Change the mySQL configuration at \my.cnf (Linux) or my.ini / my-spectrum.cnf (Windows).
2. Search for “max_allowed_packet”.
   Before change : max_allowed_packet=1M
   After Change : max_allowed_packet=16M
3. Save the file and re-start the mySQL server. Once it is up, perform the integration step, on the OC web administration page "Jasper integration" once more. Once it completes successfully, log into Jaspersoft and check for the reports.

For more information refer to CABI JasperReports Server and DX NetOps Spectrum integration related support article.

Clicking on the 'Name' link in JasperReports Server results in a Bad Request.

Symptom:

If DX NetOps Spectrum OneClick is in the https mode and the user cross launches DX NetOps Spectrum from JasperReports Server, that results in a bad request exception.

Solution:

To avoid the bad request exception:

1. Navigate to the Administrator > Report manager > Preferences page.
2. Update the OneClick server entry to https.

After updating the OneClick server entry to https, the bad request exception is not seen. Once the entries are updated, under 'Preferences' page, the values are saved in the registry table.

Configure Data Retention

The Report Data Retention functionality lets you archive or purge the following types of report data:

- Alarm
- Event
- Asset
- Availability
- SPM test result

You can specify the retention period in days for the data you want to save in the reporting database. Archive Expert provides table capacity and disk space consumption statistics for key, rapidly accumulating report data tables and suggests a retention period applicable to all database tables based on consumption trends.

The difference between the archiving and purging options is that archiving moves the data out of the operational reporting database into a separate archival database. By contrast, purging removes the data altogether from the installation. Therefore, verify that any data that you specify to purge is no longer required.

The archived data cannot be imported back to Report Manager to generate reports. Therefore, archive the data to move it to the archiving database for time periods that fall outside the retention period. For example, if you only want to generate reports from the last 90 days, specify a retention period of 90 days. All data that is accumulated in the reporting database outside the 90-day window is automatically archived on a daily basis.

The SRM reports use only the data of transformed tables/event tables to generate reports. The archived data is not available for generating reports. For example, an alarm report, which pulls data from transformed tables runs the report for number of days set in the retention period specified for the transformed tables. Similarly, an event report runs a report for the number of days set in the retention period specified for the event table.

> **NOTE**
> You can use any third-party application to access and use the archived data. However, CA does not provide any support for the use of archived data.

By archiving or purging older data, you provide more room in the reporting database for current and more recent historical data. You can generate reports more quickly and can prevent problems that occur if the report database capacity is reached.

Enable archiving or purging if your organization requires historical report data only for the retention period you specify. Ensure that you understand your organization's reporting requirements before you set a retention period. If you set it longer than required, you retain unnecessary data. Conversely, if you do not set it long enough, data that you want to view is unavailable. To save the disk space, you can purge rather than archive reporting data.

Note: The Deployment Capacity and Optimization Best Practices section provides detailed sizing guidance for the Spectrum Report Manager database.

Follow these steps:

1. Access the Spectrum Report Manager Admin Tools.
2. Select the Archive Expert option.
   The Archive Expert panel is displayed.
3. Review the following field values.
   You can change the field values. For more information, see Preferences.
   - **Data Retention Period (days) - Transformed Tables**
     (Default = 90 days) Displays the retention period (in days) for transformed tables. 'Transformed tables' refers to all rapidly accumulating tables that contain information that is derived from events (for example, alarms, and outages). The Transformed tables are the tables that participate in the Archiving and Purging activity. The event table retention period is set independently. Data older than the retention period is archived at 12:30 AM daily.

NOTE
Uncleared alarms and ongoing outages that fall within the retention period are not archived or purged. The alarm or outage data is archived or purged, if the alarm is cleared or the outage ends outside of the retention period (before the beginning of the retention period).

– **Data Retention Policy**
  - All Data -- (Default) Retains all data in the SRM reporting database for reporting purposes. No archival or purging of SRM data occurs.
  - Archive -- Moves data from the reporting database to the archival database.
  - Purge -- Purges data older than the specified ‘retention period’. Data is permanently removed from the Spectrum Report Manager database.

– **Data Retention Period (days) - Event Table**
Displays the retention period for the event table specifically. This event table is one of the fastest-growing tables, you can set the retention period individually.
For example, you can keep 60 days of event data, but 365 days (1 year) for the transformed tables data. The event retention period is much smaller than the transformed retention period because the event table grows much faster than the other tables.

NOTE

The Service Manager tables are not participated/involved in Archiving or Purging process. So, the tables are not affected by the set Retention Period/Policy.

Data Table Utilization Statistics

Archive Expert lists report data table utilization statistics for the high-growth types of report data that you select to archive. The statistics that are associated with the tables determine the recommended retention period. The tables provide you with at-a-glance indication of the current utilization and capacity and trends for both. This information helps you make informed decisions about archiving. The following image is an example data capacity table:

EVENT Table	
Available Capacity (GB)	92.85
Current Size (GB)	< 10 MB
Average Daily Growth (MB)	0.03
History (Days)	1
Earliest Record Time	May 23, 2008 1:12:59 PM
Latest Record Time	May 23, 2008 1:18:38 PM
Days until Full	> 365

The following statistical definitions are presented within the EVENT Table:

- **Availability Capacity (GB)**
  The remaining capacity available for extra record storage. The capacity value is constrained by the MySQL table capacity or physical disk space available (whichever is smaller).
- **Current Size (GB)**
  Defines the current table size as reported by MySQL.
- **Average Daily Growth (MB)**

Defines the ‘Current Size (GB)’ divided by ‘History (Days)’ converted to Megabytes.

- **History (Days)**
  Displays the number of days between the ‘Earliest Record Time’ and ‘Latest Record Time’ values.
- **Earliest Record Time**
  Displays the timestamp that is associated with the earliest record in table.
- **Latest Record Time**
  Displays the timestamp that is associated with the latest record in table.
- **Days until Full**
  Calculates the estimated number of days it takes to consume ‘Available Capacity (GB)’ given a straight-line growth estimate that is based on the ‘Current Size (GB)’ and ‘Average Daily Growth (MB)’.

Back Up Landscape

The Database Maintenance option lets you back up and restore landscape-specific data from the reporting database. Available backup lets you revert to an earlier version of the reporting database when a landscape server database backup is restored and you want to align reporting data with SpectroSERVER data. The Database Maintenance option enables you to manage the number of backups you want to retain by allowing you to remove the backup that you no longer require.

WARNING
The amount of data in the reporting database for all landscapes is dependent on the retention period (default = 90 days) specified by the Archive Expert option. Coordinate your database, and archive management settings to institute a data-storage and data-backup strategy that meets your data management requirements.

Follow these steps:

1. Access the Spectrum Report Manager Admin Tools
2. Select the Backup Landscape option.
   The Backup Reporting Landscape panel appears.
3. Select a landscape from the 'Select a Landscape to Backup' drop-down list.
4. (Optional) Enter a description of the backup.
5. Click Start to begin the backup.
   Spectrum Report Manager displays the backup progress and notifies you when it is complete. Date and time are used to identify each backup version.

Recover Landscape

When you recover landscape data, it replaces the current data for the landscape in the reporting database.

NOTE
If you attempt to recover an older version of the reporting database than the version currently in use, a warning message appears. The message recommends you to contact CA Support for more information.

Follow these steps:

1. Access the Spectrum Report Manager Admin Tools.
2. Select the Recover Landscape option.
   The Recover Reporting Landscape panel appears.
3. Select the landscape to recover from the 'Select the Landscape to Recover' drop-down list.
4. Select the backup version to recover from the 'Select a Backup Database to Use' drop-down list.
5. Click Start to begin the restoration.
   DX NetOps Spectrum Reporting displays the recovery progress and notifies you when it is complete.

Manage Backups

You can update the descriptions for backups or remove backups you no longer require. Manage your backup files in the Admin Tools section.

Follow these steps:

1. Access the Spectrum Report Manager Admin Tools.
2. Select the Manage Backups option.
   The Manage Landscape Backups panel appears.
3. Select the landscape for which you want to update or remove backups from the 'Select a Landscape' drop-down list.
4. Select the backups that you want to update or remove from the Existing Landscapes list.
5. (Optional) Modify the description text for backups if you want to update backups.
6. Click Update to save modifications, or click Remove to delete selected backups.
   The Backup files are managed.

Outage Editor

The Outage Editor lets you edit the outage records for all managed assets. You can retrieve outage records for particular models or for particular devices and interfaces, change the status of an outage in a record, annotate an outage record, and suspend an ongoing outage status for an asset that is available. Also, if you modify an asset outage that has caused an outage for a service model (because the asset is a service resource), you can modify the outage status for that service model.

> **NOTE**
> If Spectrum Report Manager model-based security is enabled, the user who is logged on to OneClick can only review and edit outages that are associated with the models that this user can access. For more information, see the Report Manager section.

About Outages and Availability Reports

DX NetOps Spectrum Reporting calculates availability for managed assets by comparing their actual availability to their expected availability. Actual availability is derived from DX NetOps Spectrum event data. This comparison provides the basis for the availability percentage that is used in Reporting Availability Reports.

For any given time period, you specify an availability percentage objective when you configure an Availability Report, and DX NetOps Spectrum Reporting calculates the actual availability percentage. The difference between the two is the interval during which an asset is presumed to be unexpectedly unavailable. DX NetOps Spectrum Reporting does not include planned outages (downtime) from assets in *maintenance mode* when it calculates availability. For example, if a device is in maintenance mode, all the outages from that device are planned outages which are not considered in availability reports.

An unplanned outage can result from asset malfunction or from other events that are unrelated to asset performance. For example, availability is affected by a power outage, inadvertent shutdowns, or instances where an off-line asset model was not put in maintenance mode. Outages that are caused by the latter events misrepresent the actual availability of an asset. These are the types of outages you would typically want to redefine as exempt or planned outages.

> **NOTE**
> Spectrum Report Manager does not support cluster-specific availability reports. Therefore, you cannot generate the Availability Reports for Clusters.

Outage Editing Status Report

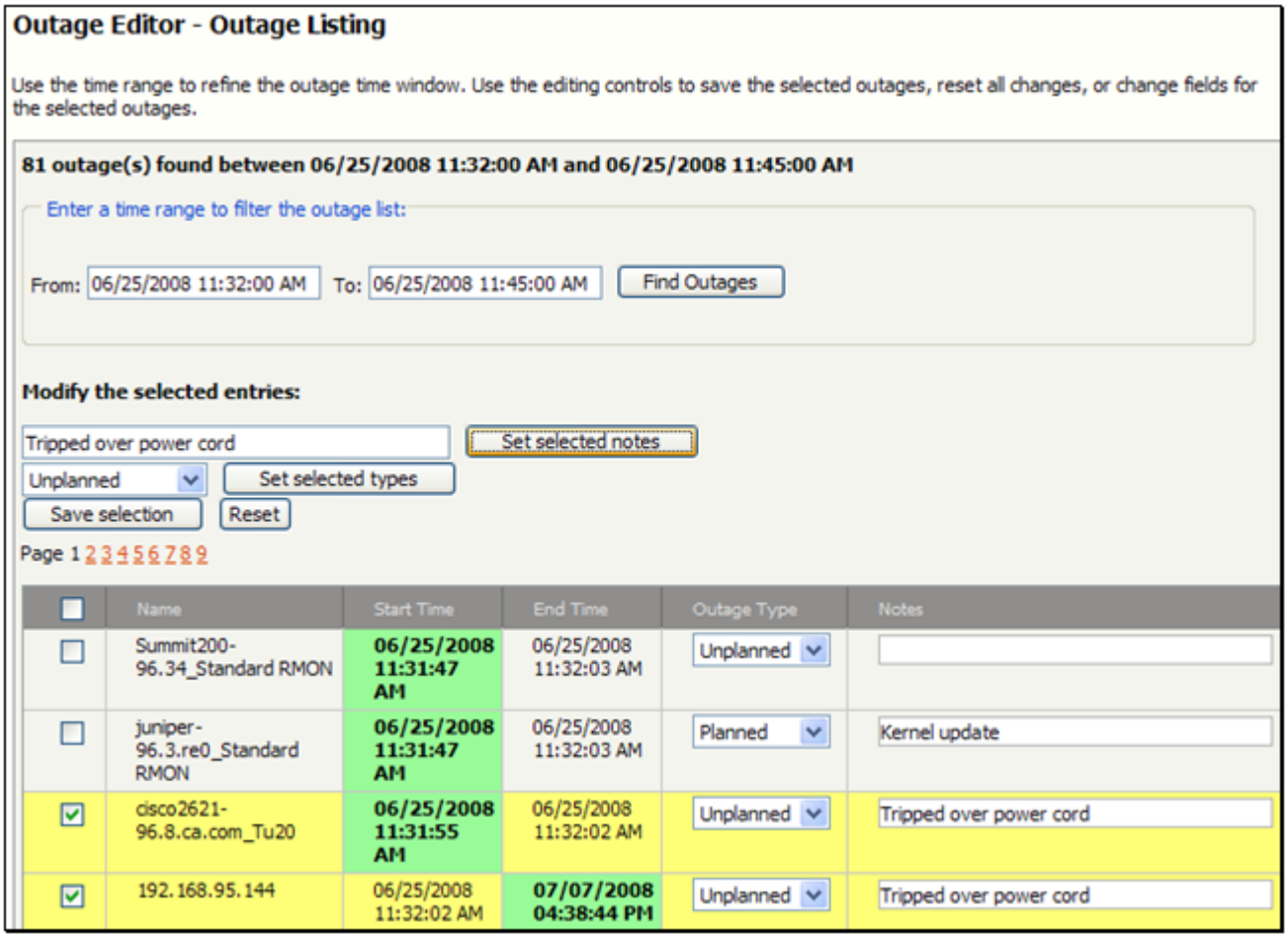
You receive an Outage Editing Status Report after editing the outage records. The results of an Outage Editor search list all outages that match your selection criteria. Each entry in the table contains the name of the model that experienced the outage, and notes, the outage start time, end time, and type. You can edit the outage type and notes. If no end time is available for an outage, it is listed as 'Ongoing' with an option to manually end the outage. Each entry in the table is preceded by a checkbox that lets you select the corresponding outage for mass editing.

In addition, the page contains *master controls* that annotate multiple outages simultaneously. Any outages that have been selected are modified by these controls. When changes have been made to any entry, 'Save selection' is enabled. Click it to save all selected entries. The 'Reset' button restores the listing to the state it was in after the most recent save.

If no outages are available in the selected period, only '0 Outage(s) Found' is displayed - the master controls and table do not appear.

If more than 500 outages are detected, the results are split up into 'pages'. Each page displays up to 500 outages and the master controls affect only the displayed outages.

The following image highlights the result of an 'Outage Editor - Search by Timespan' search. With multiple rows selected, it shows an example of mass editing. Selected entries are highlighted yellow, and times outside of the chosen range are highlighted green.



Modify Outage Records

The Outage Editor lets you modify Outage Records. When you edit an outage record, you can change its outage status and enter comments to it that describe why the record was edited or any other pertinent information. You can also end an ongoing outage status for a record that does not accurately indicate the actual availability of the asset referenced by the record.

Follow these steps:

1. Select Admin Tools, Outage Editor.
   Two options appear: Model Outages and Device/Interface Outages.
2. Select one of the options.
3. Enter a filter term to display the models or devices/interfaces that have had outages you want to view, and then click Find (Models or Devices). For example, to display a list of Cisco devices or interfaces or Cisco models that have had outages, simply enter Cisco.
   The editor displays a list of assets that match your filter term.
4. Select the asset whose outage records you want to retrieve.
   - If you are working with a model list and you want to edit outage records for a model, select the model.
   - If you are working with a device/interface list and you want to edit device outage records, select the device.
   - If you are working with a device/interface list and you want to edit outage records for a particular device interface, click Show Interfaces for the device. An interface list appears. It includes a filtering field that you can use to locate the interfaces you want to work with.

   At the bottom of the asset list, the Outage Editor indicates the number of outage records that it found for the selected asset. A date range filter lets you narrow the outage record list.
5. Accept the default date range, which extends to the current date from the date of the earliest outage known by Spectrum Report Manager, or restrict the range as required, and then click Find Outages.
   The Outage Editor displays a list of outage records for the asset. An outage record indicates when an outage began and ended, the outage type, and any notes that have been entered to the record. Instead of indicating an end time, a record may indicate that an outage is ongoing (because Spectrum Report Manager has not yet received an end outage event).
6. Update the records you want to change. You can change the outage status for particular records in their Outage Type fields or you can change the status for all listed outages in the Set all outage types to field. The following outage status types are available:
   - **Unplanned**
     An unplanned outage is an unexpected outage. Availability reports designate time that an asset was in an unplanned outage state as time the asset was unavailable. Unplanned outages are typically the result of a hardware failure (broken cable) or software failure (bad configuration, incompatible protocols) or any situation where an asset is off-line while not in maintenance mode in DX NetOps Spectrum.
   - **Planned**
     A planned outage is an outage that was intended, when an asset model in DX NetOps Spectrum was put into maintenance mode. DX NetOps Spectrum does not generate alarms on assets in maintain mode. Planned outages do not count against availability in Availability reports.

     NOTE
     For more information about maintenance mode, see the Using OneClick section.
   - **Exempt**
     An outage that evidence indicates was not unplanned, typically a situation where an asset that was taken off-line for maintenance but its model in DX NetOps Spectrum was not put into maintenance mode. You can designate any outage as exempt. Exempt outages do not count against availability.

     NOTE
     If you change the status of an outage for an asset that is a resource of a service model and is also the cause of a service outage, the Affected Services Editor window appears. It lists the service outages that are caused

by the asset outage enables you to change the status of the service outages. For more information, see the Service Manager section.

7. (Optional) You can (annotate outage records) add new notes or can overwrite existing ones for particular records in their Notes fields. You can also use the ‘Set Selected Notes’ field if you want to enter a note, edit a note, or clear notes from all records. When users generate Availability reports, they can specify Notes text as a filtering criterion.
8. (Optional) Click End Outage in the End Time field. You can end an ongoing outage immediately. For example, end an ongoing outage for an asset when you know that it is available and you do not require an availability report to misrepresent the asset availability.
9. Click Update to save your edits.
   The outage status is updated and saved.

Outage Editor - Search by Model

Outage Editor-Search by Model helps you to locate models for outage editing. Model name or model class is used to locate a model.

Follow these steps:

1. Access the Spectrum Report Manager Admin Tools.
2. Select the 'Outage Editor - Search by Model' option.
   The Outage Editor - Model Listing panel appears.
3. Enter the Filter for a model name or model class.
4. Click Find Models.
   The model names with their outages are displayed for editing.

Outage Editor - Search by Device

Outage Editor-Search by Device is used to locate devices for outage editing with the device names.

Follow these steps:

1. Access the Spectrum Report Manager Admin Tools.
2. Select the 'Outage Editor - Search by Device' option.
   The Outage Editor - Device Listing panel appears.
3. Enter the Filter to match the device names.
4. Click Find Devices.
   The device name with their outages is displayed for editing.

Outage Editor - Search by Timespan

Outage Editor - Search by Timespan displays any outage that begins before the selected end time and finishes after the selected start time.

Follow these steps:

1. Access the Spectrum Report Manager Admin Tools.
2. Select the 'Outage Editor - Search by Timespan' option.
   The Outage Editor - Time Span Search panel appears.
3. Enter the time and date range filters in the From and To fields.

   NOTE
   Leave the end time blank to search up to the current time.
4. Click Find Outages.
   The outages during the selected timespan are displayed for editing.

Set Report Manager Preferences

The Preferences option enables you to configure multiple Spectrum Report Manager options.

The purge process for the event table is enhanced to use the truncate command instead of the delete command to expedite the purge process. To streamline the partition life cycle, DX NetOps Spectrum now uses Partition Handler which uses MySQL information schema to determine the creation/deletion of partitions. SRM, now processes the other landscapes sync, even if one landscape is in an Error state. You can remove the stale buckets of any landscape using the `./RpmgrlnitializeLandscape.bat root -remStaleBuckets <landscape name>` command.

Follow these steps:

1. Access the Spectrum Report Manager Administration tab.
2. Select the Reporting Manager, Preferences option.
   The Preferences panel appears.
3. Specify the following information:
   - **Data Retention Period (days) - Transformed Tables**
     (Default = 90 days) Specifies the retention period by entering the number of days in the Retention Period (days) field. Archive Expert provides a suggested retention period estimate that is based on disk space consumption trends for the alarm, asset, availability, event, and SPM test result data tables.
     Data older than the retention period is archived at 12:30 AM daily.

     NOTE
     Uncleared alarms and ongoing outages that fall within the retention period are neither archived nor purged. The alarm or outage data is archived or purged if the alarm is cleared or the outage ends outside of the retention period (for example, before the beginning of the retention period).
   - **Data Retention Policy**
     - All Data -- (Default) Retains all data in Spectrum Report Manager reporting database for reporting purposes. No archival or purging of Spectrum Report Manager data occurs.
     - Archive -- Moves data from the reporting database to the archival database.
     - Purge -- Purges data older than the specified ‘retention period’. Data is permanently removed from Spectrum Report Manager database.
   - **Monitor SRM using a DX NetOps Spectrum model**
     DX NetOps Spectrum creates a Spectrum Report Manager Application model that is associated with the local VNM on the installation server when Spectrum Report Manager is first started after installation. By default, the model serves as a destination for events and alarms that are generated by Spectrum Report Manager. This means you can monitor the status of Spectrum Report Manager from the OneClick Console. You can, however, select the destination for event and alarm information. If you disable monitoring, event information is written to the Tomcat log files.
     Spectrum Report Manager monitors its own status. It sends events to Spectrum Report Manager application model when the following indications occur:
     DX NetOps Spectrum generates (non-persistent) alarms for some of these events and clears them when they are rectified.
     If Spectrum Report Manager Tomcat server is not connected to the model domain that contains the SRM Application model, events are sent to a log file.

     - Spectrum Report Manager loses and regains contact with a landscape, is not monitoring any landscapes, landscapes are added and removed from the list of monitored landscapes.
     - Spectrum Report Manager loses and regains contact with DX NetOps Spectrum Archive Manager.
     - Spectrum Report Manager has an event processing failure and the event processing is resumed.
   - **OneClick server**
     This field represents the OneClick server URL used for model-based contextual report launches. For example, asset links within reports are launched to that model in OneClick. By default, the OneClick Console opens from the

OneClick server installed with DX NetOps Spectrum Reporting. You can specify another server in the OneClick Server field.
DX NetOps Spectrum opens a default OneClick Console view when you click Start OneClick in the option menu. It also opens a OneClick Console topology view of an asset when you click an asset link in a report. By default, the OneClick Console opens from the OneClick server installed with DX NetOps Spectrum Reporting. You can specify another server in the OneClick Server field.
If you plan to access OneClick from a domain that is different from the domain in which the OneClick is located, enter the fully qualified server name. Enter the port number of the OneClick server you want to access *only* if the OneClick server was installed using a port number other than 80.

- If the port number HTTP default 80, enter:
  http://<*servername or fully qualified servername*>
- If the port number is not default 80, enter:
  http://<*servername or fully qualified servername*>:<*portnumber*

– **Ignore Event Poll Task Older than (milliseconds)**
This configuration helps you to ignore polling tasks that are taking long time. Specify the time (in milliseconds) to ignore the polling task that is older than the specified time. The default time is set to 750 milliseconds. Contact CA Support to know more about the recommended time settings.

– **Custom Report Logo Image File Path (on the BOXI server's file system)**
To see your custom logo in the report, provide the custom logo file (.bmp file) location details on the BOXI server, where it is saved.

– **Data Retention Period (days) - Event Table**
Enter the retention period for the event table specifically. As this is one of the fastest-growing tables, you can set this retention period individually. The data retention amounts are saved through Archive Expert.

> **NOTE**
> You can set the retention period maximum to 365 days only for Event Table. Even if you set the retention period greater than 365 days, the data that is older than 365 days is automatically deleted from the Event Table.

4. **instant_event_poller_processing_interval**
The parameter lets the SRM process only the model create/destroy events and alarm events at five minutes of interval. However, you can configure the polling interval. All other events are processed at 60 minutes of interval. The configuration helps you get the updated inventory and alarms data in the report based on the configuration interval rather than the regular 60 minutes time frame.
5. **MySQL Performance Monitor Enabled**
Enable this option to capture the MySQL configuration variables at the time of tomcat restart (every time). The captured data helps to troubleshoot MySQL related issues.
6. **Enable Security**
Enables model-based security in DX NetOps Spectrum Reporting environment. By default, security is not enabled. For more information, see the Report Manager section.
7. **Jasper User Sync**
Use the **Jasper User Sync** option to control the synchronization of the DX NetOps Spectrum users with the Jasper server. The default value is True. Users are synchronized every hour during the event polling. During the synchronization, Spectrum Report Manager checks whether the user exists or not in Jasper. Spectrum Report Manager creates users with the default roles equivalent to the DX NetOps Spectrum user roles. The default password is the same as the username. Once the user is created, the roles and passwords of the users are not modified or synchronized by the Spectrum Report Manager. Users can now customize the roles and change the password.
8. Click Update Preferences to save preference settings.
Spectrum Report Manager preference settings are saved.

Configure Monitoring Status

The DX NetOps Spectrum Status option lists all SpectroSERVERs known by the OneClick Server to which Spectrum Report Manager is connected. It indicates whether the servers are up (green) and Archive Managers are running (green). This option lets you select the SpectroSERVERs that Spectrum Report Manager monitors according to your Spectrum Report Manager license agreement and configures Spectrum Report Manager polling options.

> **WARNING**
> Event processing is required for most reports. Turning off event processing results in reporting data not being updated, such as model creation/deletion, global collection membership updates, alarm information, outage information, and SPM test information. Reports that are generated for time periods after event processing has been turned off may not be accurate.

Follow these steps:

1. Access the Spectrum Report Manager Admin Tools.
2. Select the DX NetOps Spectrum Status option.
   The DX NetOps Spectrum Status panel appears with the following fields displayed
   Landscape, SpectroSERVER Status, Archive Manager Status, and Last Event Field.
3. Select the following boxes:
   a. **Monitor?**
   Specifies that Spectrum Report Manager actively collect data from the selected SpectroSERVER. If monitoring is disabled, reports are generated from historical data in the Spectrum Report Manager database.
   To synchronize data for the test ids to be generated:

   Scenario 1: If Monitor option is Unchecked for the landscape then:

   1. Check the option and click on "Update Monitored Servers"

   Scenario 2: If Monitor Option is Checked then:

   1. Uncheck the Monitor option

   2. Click on "Update Monitored Servers"

   3. Check the Monitor Option again

   4. Click on "Update Monitored Servers"

   This will initiate the synchronization and then the details will be populated.
   b. **Asset Polling?**
   – Specifies that Spectrum Report Manager maintain daily polling of models on the SpectroSERVER for asset change data. This means that asset data in the Spectrum Report Manager database is kept up to date. If asset polling is disabled, reports are generated from historical data in the Spectrum Report Manager database.
   c. **Event Processing?**
   Specifies that Spectrum Report Manager maintain hourly polling of the SpectroSERVER for event data. This means that event data in the Spectrum Report Manager database is kept up to date. If event processing is disabled, event, availability, and alarm reports are generated from historical data in the Spectrum Report Manager database.
4. Click Update Monitored Servers to assert your selections.

NOTE
If you select not to monitor a server, you can remove landscape data from the server that remains in the reporting database by running the RpmgrInitializeLandscape.bat utility. For more information, see Initialize the Database for Specific Landscapes.

Maintenance and Troubleshooting

This chapter describes the Spectrum Report Manager maintenance procedures and troubleshooting issues and solutions.

General Application Maintenance Issues

This section describes typical issues with application maintenance for DX NetOps Spectrum Spectrum Report Manager. The topics in this section provide the procedures to follow for customizing the application and resolving issues.

End Ongoing Outages

In some cases, Spectrum Report Manager does not receive the event indicating that a given outage has ended, and thus the outage is incorrectly reported as ongoing. You can use the Outage Editor to end the ongoing outage.

Synchronize Report Data

If you reinitialize the SpectroSERVER database for a landscape that is reported by Spectrum Report Manager, then you must reinitialize that landscape in the reporting database. Otherwise the data cannot be synchronized with the data in the SpectroSERVER.

No Tests Available in a Response Time Report

You can discover that no tests are available for the time period that you specify when configuring a Response Time report. At the same time, OneClick indicates that tests do exist for the time period from which you want to generate a report. This contradiction is caused by the circumstances under which the tests were discovered in DX NetOps Spectrum Service Performance Manager. For more information, see the Service Performance Manager section.

Change the Report Logo

You can replace the default DX NetOps Spectrum logo on reports with another. The replacement logo must be a bitmap (.bmp) file with 200 pixels in width and 75 pixels in height.

Follow these steps:

1. Save the logo that you want to use to a location on the CABI server file system. You can save the file to a location outside of the CABI installation folders. Note the file path.
2. Navigate to the Spectrum Report Manager Administration Tools page and select Preferences.
3. For the 'Custom report logo image file path' entry, set the file path from the CABI server filesystem.
4. Click Update Preferences.
5. (Linux only) Verify that the logo file has at least read permissions for owner, group, and other.
   The default logo.bmp file is replaced.

Change Vendor Names in Reports

You can change a vendor name in reports that is associated with a specific enterprise number in the Spectrum Report Manager database. You can verify the following reasons to change a vendor name in Spectrum Report Manager:

- Abbreviating a lengthy name.
- Renaming a vendor when the vendor name is changed.

WARNING
Verify that this change is performed on the OneClick Tomcat server, but not on the CABI Tomcat server.

Follow these steps:

1. Shut down the Tomcat server.
2. Copy the vendor.xml file to the custom Spectrum Report Manager configuration directory.
   ```
   cp
   <$SPECROOT>/tomcat/webapps/spectrum/WEB-INF/repmgr/config/vendor.xml <$SPECROOT>/custom/repmgr/config/
   vendor.xml
   ```
3. Edit the <*$SPECROOT*>/custom/repmgr/config/vendor.xml file.
   For each vendor name you want to change, create a <vendor></vendor> entry. In the <vendor_ID></vendor_ID> field, specify the vendor number in hexadecimal or decimal format, and the vendor name within the <vendor_name></vendor_name> field.
   For example, the following file format shows two vendor name changes:
   ```
   <?xml version="1.0" encoding="UTF-8"?>
   <root>
   <vendor>
   <!-- Changes the first vendor name -->
   <vendor_ID>Enterprise #</vendor_ID>
   <vendor_name>Some new name</vendor_name>
   </vendor>
   <vendor>
   <!-- Changes the second vendor name -->
   <vendor_ID>Enterprise #</vendor_ID>
   <vendor_name>Some new name</vendor_name>
   </vendor>
   </root>
   ```
   Definitions:
   - *Enterprise #* - The enterprise number for a vendor product that is referenced in the report.
   - *Some new name* - The new name that is associated with the enterprise number.
4. Save your changes.
5. Restart the Tomcat server.
   The vendor names are changed.

Change Event Names in Reports

You can supply custom names for events that appear in event reports. Overriding default event names lets you clarify specific items for event report recipients.

Follow these steps:

1. Shut down the Tomcat server.
2. Copy the eventtitle.xml file to the custom Spectrum Report Manager configuration directory.
   ```
   cp
   <$SPECROOT>/tomcat/webapps/spectrum/WEB-INF/repmgr/config/eventtitle.xml <$SPECROOT>/custom/repmgr/config/
   eventtitle.xml
   ```
3. Edit the <*$SPECROOT*>/custom/repmgr/config/eventtitle.xml file.
   For each event name you want to change, create an <event></event> entry. Specify the event type code in hexadecimal or decimal format in the <event_type></event_type> field and the event name in the <event_title></event_title> field.
   ```
   <?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
   <root>
   ```

```
<event>
<!-- Identifies the event type code-->
<event_type>An event type code</event_type>
<!-- Specifies the custom event name-->
<event_title>A custom event name</event_title>
</event>
</root>
```

4. Save your changes.
5. Restart the Tomcat server.

NOTE
The event name is changed.

Change Probable Cause Names in Reports

You can supply custom names for probable causes that appear in alarm reports. Overriding default probable cause names lets you clarify specific items for alarm report recipients.

Follow these steps:

1. Shut down the Tomcat server.
2. Copy the pcausetitle.xml file to the custom Spectrum Report Manager configuration directory.

```
cp
<$SPECROOT>\tomcat\webapps\spectrum\WEB-INF\repmgr\config\pcausetitle.xml <$SPECROOT>\custom\repmgr\config
\pcausetitle.xml
```

3. Edit the <*$SPECROOT*>\custom\repmgr\config\pcausetitle.xml file.
   For each pcause name you want to change, create an <pcause></pcause> entry. Specify the pcause ID in the <pcause_type></pcause_type> field and the pcause name in the <pcause_title></pcause_title> field.

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
<root>
<pcause>
<!-- Identifies the probable cause ID-->
<pcause_type>A pcause ID</pcause_type>
<!-- Specifies the custom probable cause name-->
<pcause_title>A custom pcause title</pcause_title>
</pcause>
</root>
```

4. Save your changes.
5. Restart the Tomcat server.
   The probable cause name is changed.

Change the OC_user MySQL Password

When DX NetOps Spectrum Reporting connects to the database, you can use a user ID and a password to authenticate against the database. In previous DX NetOps Spectrum releases, one password was used for authentication. However OneClick usernames are now used for background processing (OC_user and OC_admin) to add events and data in the database.

In addition, Crystal Report users need a CR_user password for on-demand reports. WEBI Report users need a WEBI_user for adhoc reports.

Analyze Table

Analyze Table is a command that can be executed on the MySQL command-line tool to update database statistics and improve the overall query performance. MySQL uses the table statistics to determine the order in which tables can be joined. In addition, table statistics can be used to select indexes to use for a specific table within a query.

If multiple modifications (such as INSERT or DELETE) are performed on a database table over time, the table statistics can become out of order. The MySQL Query optimizer which uses the table statistics can produce inefficient query execution plans and can cause MySQL performance degradation. Therefore, we recommend using Analyze Table to update table statistics.

The Analyze Table operation reads the entire database table and rebuilds the table statistics with the information about the distribution of key values. You can run this command on MyISAM and InnoDB tables. During the analysis, the table is locked with a read lock. When locked, the table cannot be accessed for other operations.

How to Run Analyze Table

To improve the performance of the database, run the Analyze Table command. You can run the Analyze Table on any of the following tables:

- event
- modeloutage
- model
- devicemodel
- alarminfo
- alarmactivity

Follow these steps:

1. From the OneClick menu bar, select the Administration tab.
   The Administration Pages panel appears on the left side.
2. From the Administration Pages menu bar, select Spectrum Report Manager.
   The Spectrum Report Manager Admin Tools panel appears on the left side.
3. Select Spectrum Status.
   The Spectrum Status page opens.
4. Clear all Monitor check boxes to stop all landscape monitoring.
5. Close all other reports (scheduled or on-demand) that are running.
6. Invoke a MySQL command prompt, and run the following command:
   ```
   ANALYZE TABLE table_name
   ```
   The task to run Analyze Table is complete.

   NOTE
   Before running Analyze Table stop event and asset polling. Set the data retention policy to 'all data' to disable the reporting database archiving. Restart event and asset polling after running the analyze table and set the data retention policy to its default value. For more information, see Configure DX NetOps Spectrum Monitoring Status and Set Spectrum Report Manager Preferences.

Define Events Types for Availability Processing

You can manage the volume and scope of outages to include in availability reports. You can specify the events that the Spectrum Report Manager availability handler uses to determine the beginning and end of planned and unplanned outages. You can designate events as UP, DOWN, IN MAINT MODE, and OUT OF MAINT MODE event types in the

availability.xml file. You can override default event type designations and can assign types to new events. You can also specify that the availability handler ignores and does not process particular event types. A copy of the file is located in the following directory:

```
<$SPECROOT>/tomcat/webapps/spectrum/WEB-INF/repmgr/config
```

The availability.xml file is formatted as follows:

```
<root>
  <up_event></up_event>
  <down_event></down_event>
  <in_mm_event></in_mm_event>
  <out_mm_event></out_mm_event>
  <ignore></ignore>
</root>
```

You can designate an event as a particular type by entering the event code in decimal or hexadecimal format.

NOTE

For hexadecimal format, make sure you mention '0x' before the event type.

Follow these steps:

1. Make a copy of the availability.xml file.
2. Configure event designations using the tags included in the availability.xml file. All event type tags must be nested within the root tags.
   The following example shows the cluster model type, assuming that the IBM cluster resource group with model type as 0x621000d, 1234 is designated as the DOWN event, 5678 is designated as the UP event, and 0x4567 is designated as an ignored event type:
   ```
   <root>
     <down_event>0x1111</down_event>
     <up_event>0x3333</up_event>
     <down_event>0x5555<down_event>
     <ignore>0xabcd</ignore>
     <down_event type='1234'>
     <model_type>0x621000d</model_type>
     <down_event>
     <up_event type='15678'>
     <model_type>0x621000d</model_type>
     </up_event>
     <ignore>04567</ignore>
   </root>
   ```
3. Save and copy the customized availability.xml file to the following directory:
   ```
   $SPECROOT/custom/repmgr/config
   ```
   The content of the file is written to the AvailabilityEvent database table and is read by the availability handler when it compiles availability statistics for the availability reports.
4. Restart the OneClick Tomcat server to enable the event designations that you specified.

Filtering Event Processing

Event processing filters are defined by an XML file that can exclude certain events from being loaded into the Spectrum Report Manager database. Specifically, events that are associated with the event types or model handles that are listed in the event-processing-filter.xml filter file are not loaded into the Spectrum Report Manager database.

Before you modify the supplied event-processing-filter.xml file, you can determine the event types and model handles for which event activity can be excluded. Excluded events are not available in Spectrum Report Manager for historical reporting purposes.

Follow these steps:

1. Copy the event-processing-filter.xml and event-processing-filter-schema.xsd files to the 'custom' directory. For example, see the following syntax:

```
cp
<$SPECROOT>/tomcat/webapps/spectrum/WEB-INF/repmgr/config/event-processing-filter-schema.xsd
<$SPECROOT>/custom/repmgr/config/
cp
<$SPECROOT>/tomcat/webapps/spectrum/WEB-INF/repmgr/config/event-processing-filter.xml
<$SPECROOT>/custom/repmgr/config/
```

2. Edit the event-processing-filter.xml to reflect your selected filtering strategy.
   For example, see the following syntax:

```
<ignore>
    <event-type>0x1245</event-type>
    <event-type>0xffa0004</event-type>
    <model>0x00d40010</model>
    <model>0xff0100d1</model>
</ignore>
```

   NOTE
   You can only ignore events that are associated with specific models or event types.

3. Restart Tomcat.
   The specified event processing filters are now in effect.

Define Event Filters for Event Reports

Event filters are uniquely named sets of predefined event codes. When you configure an event report, you can select event types to include or exclude data in the report from events in a particular event filter.

An event filter is defined by an XML file that specifies the event codes. You can create new event filter files and can copy or modify the event filter files that are included with Spectrum Report Manager. Before you create or modify event filters, determine the types of events that are important in your deployment.

Set Up Event Filtering

To set up event filtering, copy the XML schema and predefined configuration events file that are provided by Spectrum Report Manager to the custom directory (*$SPECROOT/*custom/repmgr/config/events).

Follow these steps:

1. Copy the following schema file:

```
cp
$SPECROOT/tomcat/webapps/spectrum/WEB-INF/repmgr/config/events/event-filter.xsd
$SPECROOT/custom/repmgr/config/events
```

2. Copy the following predefined configuration events file:

```
cp
$SPECROOT/tomcat/webapps/spectrum/WEB-INF/repmgr/config/events/event_filter_file.xml $SPECROOT/custom/
repmgr/config/events
```

The possible values for the *event_filter_file* variable are as follows:

- **ADES-events-filter.xml**
  Contains the most significant Active Directory and Exchange Server (ADES) event codes to create a basic report.
- **ncm.xml**
  Specifies event codes for Network Configuration manager (NCM) activities in DX NetOps Spectrum.
- **vhm.xml**
  Contains the most significant Virtual Host Manager (VHM) event codes to create a basic report.
- **vhmtrap.xml**
  Contains a large list of all the potential Virtual Host Manager (VHM) traps to create a comprehensive report.
- **Cluster.xml**
  Contains all cluster events, including IBM and Microsoft.
- **IBM-Cluster-all.xml**
  Contains all of the IBM cluster events.
- **IBM-run-status.xml**
  Contains all of the IBM cluster events that are related to Status (such as up, down, offline).
- **MS-Cluster-all.xml**
  Contains all of the Microsoft cluster events.
- **MS-run-status.xml**
  Contains the Microsoft cluster events that are related to Status (such as up, down, offline).
- **ClusterTrap.xml**
  Contains only the trap events from IBM and Microsoft clusters.
- **Cluster-spectrum-managing.xml**
  Contains the DX NetOps Spectrum management events, such as cluster proxy events, management events, and polling events

The files are copied to the custom directory and event filtering setup is complete.

Create an Event Filter File

You can create event filter files and can copy or modify the event filter files that are included with Spectrum Report Manager.

Follow these steps:

1. Define an XML file that includes any number of event type codes in either hexadecimal or decimal format. For example, see the following filter format:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<filter xmlns="http://www.aprisma.com"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aprisma.com ./event-filter.xsd">
<event_type>event type code</event_type>
<event_type>event type code</event_type>
</filter>
```

2. Save the file under a name that suggests the types of events to the custom directory. For example, see the following format:

```
<$SPECROOT>/custom/repmgr/config/events
```

   An event filter file is created.

Exempt All Unplanned Outages for a Particular Day

You can use the exemptOutagesForDay command-line utility to exclude data from the Availability reports for all unplanned outages that have occurred on a non-working, or non-SLA day (for example, a holiday or a planned shutdown day). The

utility also includes a parameter that lets you exempt any service outages that are caused by the device/interface outages. You can unexempt planned outages.

NOTE
You can exempt only those device/interface outages that have occurred for a single day and not a range of days. If you want to exempt outages for multiple days, you can run the utility for each day.

NOTE
For more information, see How the Exempt Outage Utility Handles Particular Outage Scenarios.

The utility is located in the /spectrum/bin directory.

Syntax:

```
exemptOutagesForDay <mysql username> <mysql password>
<exempt service outages> [-undo <YYYY-MM-DD>] YYYY-MM-DD day
```

Examples:

The following example exempts device/interface outages from January 1, 2006 and all service outages resulting from the device/interface outages:

```
exemptOutagesForDay root root yes 2006-01-01 New Year's Day
```

The following example exempts device/interface outages from January 1, 2006 but not service outages resulting from the device/interface outages:

```
exemptOutagesForDay root root no 2006-01-01 New Year's Day
```

The following example unexempts device/interface outages from January 1, 2006 and all service outages resulting from the device/interface outages:

```
exemptOutagesForDay root root yes -undo 2006-01-01 New Year's Day
```

The following example unexempts device/interface outages from January 1, 2006 but not service outages resulting from the device/interface outages:

```
exemptOutagesForDay root root no -undo 2006-01-01 New Year's Day
```

How the Exempt Outage Utility Handles Particular Outage Scenarios

Not all outages begin and end during the exempt period. The exemptOutagesForDay utility responds to outages that begin or end outside the exempt day period in various ways. You can verify the following outage scenarios that are handled by the exemptOutagesForDay utility:

- The outage begins before the exempt day and ends during it.
  The utility ends the existing outage at the start of the exempt day. A new exempt outage from the start of it to the end of the original outage is created.
- The outage begins before the exempt day and ends after it.
  The utility ends the existing outage at the start of the exempt day. A new exempt outage during the exempt day, and a new unplanned outage from the end of the exempt day to the end of the original outage is created.
- The outage begins during the exempt day and ends after it.
  The utility ends the existing outage at the end of the exempt day. The utility converts the outage to an exempt outage, and then creates a new unplanned outage from the end of the exempt day to the end of the original outage.
- The service outage begins before or during the exempt day and ends during or after it.
  The entire service outage is exempted.

Configure User-Defined Device Attribute Polling

The Spectrum Report Manager historical device management feature polls devices for well-known attributes, such as device name and network address. However, you can pull more data from devices, such as vendor-specific data, or settings that are applicable to a business environment. Therefore, Spectrum Report Manager enables you to tailor the device polling behavior.

Custom device polling is organized by model type, and then by attributes within the model type. The polling supports a maximum of ten (10) additional attributes per model type, in three attribute type areas. You can verify the following list for the attribute type and the associated DX NetOps Spectrum type mappings.

Type	Supported DX NetOps Spectrum Type Mapping	Maximum Number of Attributes (per type)
Varchar-based	All character-based and numeric types	4
Integer-based	Numeric, scalar types	4
Date/Time	Timestamp, time period types	2

Mapping Polled Attributes to Storage

If you are interested in storing the additional device attributes, you can determine the attributes to store. You can accomplish this task in the OneClick Attribute Editor view of a given device. Locate the desired attributes, and note the hexa-based attribute IDs (for example, 0x1006e) and the corresponding Type (for example, Integer, Text String).

You can select the attributes and can assign them to one of the open storage locations. The mapping occurs in an XML file, which is at:

```
<SPECROOT>/tomcat/webapps/
    spectrum/WEB-INF/repmgr/config/devicemodel-polling.xml
```

For a model type, each attribute id is mapped to a storage location. Consider the following example, for the model type 'Rtr_Cisco', that maps the attribute ID '0x118b8' to string storage location '1':

```
<devicemodel-polling ...>
    <user-defined-poll modelType="Rtr_Cisco">
        <!-- Company name attribute -->
        <poll-attribute attrId="0x118b8">
            <varchar-storage id="1"/>
        </poll-attribute>
    </user-defined-poll>
</devicemodel-polling>
```

Next, consider the following example that maps several attributes, across several model types:

```
<devicemodel-polling ...>
    <user-defined-poll modelType="Rtr_Cisco">
        <!-- Company name attribute -->
        <poll-attribute attrId="0x118b8">
            <varchar-storage id="1"/>
        </poll-attribute>
        <!-- Disposable precedence attribute -->
        <poll-attribute attrId="0x114e2">
            <int-storage id="4"/>
        </poll-attribute>
```

```
    </user-defined-poll>
    <user-defined-poll modelType="JuniperJUNOSRtr">
        <!-- Company name attribute -->
        <poll-attribute attrId="0x118b8">
            <varchar-storage id="2"/>
        </poll-attribute>
        <!-- Create time now attribute -->
        <poll-attribute attrId="0x11b41">
            <datetime-storage id="1"/>
        </poll-attribute>
    </user-defined-poll>
</devicemodel-polling>
```

In the previous example, two model types are configured for custom attribute polling. For 'Rtr_Cisco' devices, the company name attribute is stored at string storage '1' and the disposable precedence attribute is stored at long storage '4'. For 'JuniperJUNOSRtr' devices, the company name attribute is storage at string storage '2' (note the difference from the 'Rtr_Cisco' configuration) and the create time attribute is stored at date/time storage '1'.

A full example XML file can be found at:

```
<SPECROOT>/tomcat/webapps/
    spectrum/WEB-INF/repmgr/config/devicemodel-polling-example.xml
```

NOTE

The newly created attributes are updated in Spectrum Report Manager database during device polling which is scheduled for every 24 hours.

Reporting Labels

You can retrieve the attributes that are stored in the database, and can identify the attributes that are stored in 'varchar storage 1' for the 'Rtr_Cisco' model type. The reporting label lets you describe the purpose of an attribute and user-oriented interfaces.

The reporting label is configured in the following example:

```
<devicemodel-polling ...>
    <user-defined-poll modelType="Rtr_Cisco">
        <!-- Company name attribute -->
        <poll-attribute attrId="0x118b8">
            <varchar-storage id="1"/>
        <reports label="Company Name" />
        </poll-attribute>
    </user-defined-poll>
</devicemodel-polling>
```

If you require to display the user-defined device attribute, the label text can be used to identify the attributes purpose.

Displaying Attributes in Reports

You can display the user-defined attributes as part of a Business Objects WEBI document using the DX NetOps Spectrum Universe. You can browse for subfolders that are prefixed with 'User Specified Device Attributes', while selecting result objects for the WEBI Document query.

For example, consider the following location:

Asset/Additional Device Information/User Specified Device Attributes - Asset.

For each polled attribute, the following objects and their source location are available in the folder:

- Attribute ID - configured in .xml file
- Label - configured in .xml file
- Value - polled from device

You can drag the desired result objects to the relevant WEBI panel to incorporate the objects into the query.

Polling Behavior:

When a device is polled, only the current user-defined polling configuration is applied. Therefore, if a configuration change is made, a device is polled again. The attributes and storage locations reflect the current configuration. Attributes of the devices that are polled before the configuration change and the old storage locations are cleared before storing the current attribute values.

CABI Installation and Operation Errors

This section describes typical CABI installation and operation errors. You can access the log files that are generated during CABI installation and in many cases, you can troubleshoot the errors that you find there. For more information, see the CA Business Intelligence Implementation section.

Failed to Open the Connection Error

Symptom:

The following error message appears when you try to run a report after integrating the newly installed CABI with DX NetOps Spectrum:

```
Failed to open the connection
```

Solution:

You can encounter this issue if MySQL drivers are not set correctly in your system. To resolve the issue, take the following steps:

1. From the CABI installer, navigate to the patch folder, and copy the CA_NVM_EXE folder to any drive.
   For example, copy the patch folder from C:\cabi\Windows\Disk1\cabi\patch\CA_NVM_EXE to E:\ drive.
2. From the CLI window, run the following batch file:
   ```
   E:\CA_NVM_EXE\nvm_cabi_post_install_windows.exe
   ```
   nvm_post_install log files are created.

   NOTE
   Rewritable permissions are required before making any changes to the NVM_EXE folder and the mysql.jar file, which is at C:\Program Files (x86)\CA\SC\CommonReporting3\common/4.0/java/lib/external.

LDAP User Scheduled Reports Failed to Work

Symptom:

When I am logged in as an LDAP user, I am unable to view the Reports folder hierarchy in BI Launch Pad. In addition, reports that are scheduled by LDAP users are not working.

Solution:

This issue occurs if the cabi_default_groups.xml file of the CABI 4.1 SP3 installer does not contain the updated scripts.

Update the cabi_default_groups.xml file and verify the updated information is available in the cabi_default_groups.xml file. Take the following steps to verify the updated content:

1. From the CABI 4.1 SP3 installer, navigate to the following path:
   ```
   Disk1/cabi/content/
   ```
2. Search for the following file:
   ```
   cabi_default_groups.xml
   ```
3. Verify for the following information in the .xml file:
   ```
   <?xml version="1.0"?>
   <biconfig version="1.0">
   </biconfig>
   ```

Windows Script Error

Symptom:

During the CABI installation on Windows, the following error message appears several times:

```
Windows Script Host: There is no script engine for file extension ".js".
```

The CABI installation requires access to a JavaScript engine. This problem is caused due to the following reasons:

Windows does not have a program that is associated with the .js file extension.

- Microsoft Windows Script is not installed.

Solution:

Install the latest version of Microsoft Windows Script, which can be downloaded from Microsoft. Once the download is installed, you can reinstall CABI.

Maximum Records Error (Windows)

Symptom:

The following message appears when you try to run a report:

```
Maximum processing time or maximum records limit reached.
```

Solution:

The CABI record limit is set too low. After a first-time installation of CABI, this value is defaults to 20,000.

To set the record limit to unlimited, perform the following steps:

1. Open CMC, and click Servers.
2. Find the server labeled 'Crystal Reports Processing Server' (formerly known as the Page Server).
3. Right click 'Crystal Reports Processing Server', and select Properties.
4. Find the field labeled 'Database Records Read When Previewing or Refreshing'.
5. Enter 0, and click Save and Close.
6. Right click the server and select Restart Server.

> **NOTE**
> This procedure temporarily prevents users from generating reports. Once the Processing Server restarts, report generation can resume.

SQL Server Memory Usage (Windows)

Symptom:

If you are still using Microsoft SQL Server as the CMS database server, you can observe that when the CABI SQL server activity is low, its memory usage continues to increase on the host server. This behavior is considered normal and expected. Microsoft attributes this behavior to the SQL Server buffer pool, which is designed to release memory as it is required by other processes. For more information, see the *Microsoft Knowledge Base Article 321363*.

Solution:

Not applicable.

Reporting Troubleshooting Topics

This section describes the following reporting errors and the suggested solutions to fix these errors.

Failed to open the connection while generating reports

Symptom: After installing and integrating CABI BOXI with DX NetOps Spectrum, you may get the following error, while running the reports:

“The viewer could not process an event. Failed to open the connection”

Cause: The Open Database Connectivity (ODBC) drivers are not configured properly during post installation phase.

Resolution: Run the post installation steps - Configure the Database Driver.

1. Navigate to the CABI BOXI computer.
2. Copy the CA_NVM_EXE folder from <CABI VCD>\utilities folder to the local disk.
3. If you are on UNIX environment, set the CASHCOMP variable to the folder where CABI BOXI is installed. Example: /opt/cabi41
4. Run the following command as **CABI install user**:
   Windows: nvm_boxi_post_install_windows.exe
   Linux: nvm_boxi_post_install_unix.sh
   Log files (nvm_boxi_post_install.log, nvm_boxi_post_install.java.log) for this utility can be found in system temp path (in windows: %temp% and in Linux: /var/tmp/).

Failed to open the connection error in Tomcat log

Symptom: “Failed to open connection” exception in SRM tomcat log.

Cause: Due to incorrect MySQL driver paths, or due to permissions not granted to SRM server for that user.

Resolution: Grant privileges to SRM MySQL server and verify/correct the drivers path in CABI BOXI computer.

To grant privileges for SRM MySQL server, follow these steps:

1. Navigate to $SPECROOT/mysql/bin in bash or command prompt.
2. Log in to MySQL in DX NetOps Spectrum OneClick using the following command:
   ./mysql –uroot –proot reporting
3. Execute the following command:
   GRANT ALL PRIVILEGES ON *.* TO 'root'@'cabi hostname' IDENTIFIED BY 'root';
4. Execute the following command:
   GRANT ALL PRIVILEGES ON *.* to 'CR_user'@'<cabi hostname>' identified by '0n3cl1Ck';
   For WEBI user:
   GRANT ALL PRIVILEGES ON *.* to 'WEBI_user'@'<IP address of the host>' identified by '0n3cl1Ck'
5. FLUSH PRIVILEGES;

Verify/Correct the MySQL driver path in CABI BOXI computer:

1. Navigate to the following directory:
   C:\Program Files (x86)\CA\SC\CommonReporting4\SAP BusinessObjects Enterprise XI 4.0\java\lib\external
2. Verify if the file mysql-connector-java.jar is present.
3. Verify if the path of this JAR file is correct in the following files:
   C:\Program Files (x86)\CA\SC\CommonReporting4\SAP BusinessObjects Enterprise XI 4.0/dataAccess/connectionServer/jdbc/mysql.sbo
   C:\Program Files (x86)\CA\SC\CommonReporting4\SAP BusinessObjects Enterprise XI 4.0/dataAccess/connectionServer/jdbc/jdbc.sbo
   C:\Program Files (x86)\CA\SC\CommonReporting4\SAP BusinessObjects Enterprise XI 4.0/java/CRConfig.xml
4. If you modify any of the above files, then you must restart the BOXI tomcat using Central Configuration Manager (CCM).

No records found ‘or’ no data in reports

Symptom: No records found or failed to open connection error while running report. The well-known parameters (WKP_ProxyUserSecurity) value is NULL!!!.

This issue can be verified by right-clicking the SRM parameters page for any report and going to "View Source". Search for "WKP_ProxyUserSecurity"

Looks like this:

<input type="hidden" name="WKP_CustomLogo" value="NULL!!!">
<input type="hidden" name="WKP_ProxyUserSecurity" value="NULL!!!">

Cause: The Reports login credentials are not updated and Open Database Connectivity ODBC drivers are not configured properly.

Resolution: Run the post installation steps – Configure the Database Driver and then run the following steps.

1. Go to the OneClick web page.
2. Navigate to Administration -> Report Manager -> Manage Business Objects Content.
3. Click "Update Content" and let it complete.
4. Verify if the wkp parameters are set properly like below:
   <input type="hidden" name="WKP_CustomLogo" value="C:\win32app\Spectrum\custom\repmgr\logo.bmp">
   <input type="hidden" name="WKP_ProxyUserSecurity" value="Administrator">

No records found ‘or’ no data in reports due to security strings

Symptom: The global collections list is not populated in group report and “No Record Found” while running the other reports.

Cause/Hyposthesis: The security strings for the user might not be updated. Log in to SRM server MySQL then run the following query to check the security strings:
Select * from v_security_string_accessibility_by_landscape;

Or, to check for a user:
Select * from v_security_string_accessibility_by_landscape where user_name= “<USER_NAME>” . The result should contain at least (ADMIN, *UNKNOWN* and blank) otherwise it is indication that security strings are not updated for user.

Resolution: Restart the SRM OneClick server/tomcat so that all the security strings for each user get updated.

Maximum report processing limit reached

Symptom: Error in tomcat log while running the report – “The maximum report processing jobs limit configured by your system administrator has been reached”.

Cause: The limit of number of jobs running on CABI BOXI server has been reached.

Resolution: Increase the limit of the CABI BOXI server using the following steps:

1. Log in to Central Management Console (CMC) as an "Administrator".
2. Select "Servers" from the drop-down list.
3. Go to Service Categories -->Crystal Reports Services.
4. In Server Name, right-click on "<hostname>.AdaptiveJobServer" and select "Properties"
5. In Crystal Reports 2013 Scheduling Service, change the Maximum Concurrent Jobs value from "5" to a higher number.
6. Click "Save and Close".
7. Right-click on the server and select Restart to restart the server.
8. If you still see the error, then increase the limit to a higher number.

Unable to compile class for JSP while running the report

Symptom: Error while running the reports in a fresh install or after upgrade: “Unable to compile class for JSP: an error occurred in the JSP file: /jsp/runReport.jsp or /jsp/listParameters.jsp”

Cause: The deployed jars and JSP files in DX NetOps Spectrum and CABI are not in sync.

Resolution: Run the following steps.

1. Run only the SpectrumUpdate command as mentioned in the documentation.
2. Stop the CABI tomcat server.
3. Delete "localhost" folder in CABI tomcat's work\Catalina folder.
4. Back up UdmCAFApp.jar at
   C:\Program Files(x86)\CA\SC\CommonReporting4\tomcat\webapps\BOE\WEB-INF\eclipse\plugins
   \webpath.UdmCAFApp\web\WEB-INF\lib
   where C:\Program Files (x86)\CA\SC\CommonReporting4\ is the CABI 4.x installation directory.
5. Copy the NEW_UdmCAFApp.jar file as UdmCAFApp.jar which got downloaded into SpectrumUpdate utility folder to the following location:
   C:\Program Files (x86)\CA\SC\CommonReporting4\tomcat\webapps\BOE\WEB-INF\eclipse\plugins
   \webpath.UdmCAFApp\web\WEB-INF\lib
6. Copy the listParameters.jsp and runReport.jsp files from DX NetOps Spectrum <SPEC_Home>/tomcat/webapps/spectrum/repmgr/admin/ to the following location:
   C:\Program Files (x86)\CA\SC\CommonReporting4\tomcat\webapps\BOE\WEB-INF\eclipse\plugins
   \webpath.UdmCAFApp\web\jsp
7. Start the CABI BOXI tomcat server.

HTTP 404, when right-click the report and select view

Symptom: HTTP 404 error, when you right-click on any report and select View in the BI Launch Pad.

Cause: The web.xml file present in CABI BOXI is not getting updated properly.

Resolution: Run the following steps to configure/correct the web.xml.

1. Navigate to $BO_INST_PATH/CommonReporting4/tomcat/webapps/BOE/WEB-INF/eclipse/plugins/webpath.CustomParams/web/WEB-INF/
2. Open the web.xml file in a text editor.
3. Find the following section:
   <context-param>
   <param-name>xmlFilePath</param-name>
   <param-value>/opt/CA/SharedComponents/CommonReporting4cacaf</param-value>
   </context-param>

Note that the "CommonReporting4cacaf" should be separated with a forward slash.

4. Update the line as below:
   <param-value>/opt/CA/SharedComponents/CommonReporting4/cacaf</param-value>
5. Save the file and exit.
6. Shut down the CABI Tomcat.
   On Windows, you utilize the Central Configuration Manager.
   On Linux, go to $BO_INST_PATH/CommonReporting4/sap_bobj/
   Execute: ./tomcatshutdown
7. Go to $BO_INST_PATH/CommonReporting4/tomcat/work/Catalina/
8. Delete the localhost folder.
9. Start the CABI Tomcat.
   Windows use Central Configuration Manager again.
   Linux: ./tomcatstartup (from the same sap_bobj) directory as above.

"NoSuchMethodError" in Tomcat log

Symptom: Unable to run any reports due to the NoSuchMethodError error in tomcat log –"java.lang.NoSuchMethodError: com.crystaldecisions.sdk.occa.report.application.ReportAppSession.setReportEngineTypeValue(I)V at com.crystaldecisions.sdk.occa.managedreports.ras.internal.RASReportAppFactory.getReportAppSession(RASReportAppFactory.j com.crystaldecisions.sdk.occa.managedreports.ras.internal.RASReportAppFactory.openDocument(RASReportAppFactory.java:51

Cause: The Integration or "Update Content" failed to update the database log in credentials for the reports due to stale old CABI jars in DX NetOps Spectrum tomcat.

Resolution:

1. Stop the OneClick/tomcat SRM server.
2. Navigate to $SPECROOT/tomcat/lib directory.
3. Remove the rasapp.jar, rascore.jar, and Serialization.jar from $SPECROOT/tomcat/lib directory.
4. Start the OneClick/tomcat SRM server.
5. Wait for the tomcat to start and then go to DX NetOps Spectrum OneClick page ->Administration->Manage Business Objects Content and click "Update Content".

File I/O error while running reports

Symptom: Error while running few reports in CABI BOXI: "Error in the file xxxxx: File I/O error"

Cause: Report is unable to fetch the large number for data present in SRM database.

Resolution: Increase timeout, Cache & the records limit using the following steps.

1. Log in to the Central Management Console (CMC).
2. Click Servers.
3. Right-click on Crystal Reports Cache Server and select properties.
4. Set the Idle Connection Timeout (minutes) value to **60**.
5. Set the Maximum Cache Size (KB) value **to 1028000.**
6. Click Save & Close.
7. Right-click on Crystal Reports Processing Server 2013 and select Properties.
8. Ensure that the field Database Records Read When Previewing or Refreshing is set to **0** (for unlimited).
9. Set the Idle Job Timeout (minutes) value to **60**.
10. Click Save & Close.
11. Restart both the Cache and Processing Servers. (this step is not necessary if you are going to restart the SIA).

Report Generation Takes Long Time

Symptom: The parameter page opens slow or Report is running slow or MySQL queries are running slow.

Cause: Probably because of large database/records.

Resolution: Increase the performance of SRM MySQL server using the following recommended settings.

1. Stop the DX NetOps Spectrum Processd using `systemctl stop processd` command.
2. Stop DX NetOps Spectrum tomcat.
3. Navigate to $specroot/mysql.
4. Take back-up of existing my-spectrum.cnf.
5. Update the my-spectrum.cnf with the following parameters:
   key_buffer = 512M
   table_cache = 1000
   wait_timeout = 300
   max_allowed_packet = 32M
   query_cache_size = 64M
   query_cache_type = 1
   sort_buffer = 64M
   myisam_sort_buffer_size = 64M
   read_buffer_size = 64M
   read_rnd_buffer_size = 6M
   thread_concurrency=4
6. Restart the Processd and tomcat

> **NOTE**
>
> The above settings can be increased depending on the hardware resources available in the environment/ machine.

Unable to export report data (without Titles, Headers, Group Headers, Summary, Footers) to .csv format.

Symptom:

When I export the report data into CSV format, I want to filter out Title, Header, Group Header, Summary, and Footer information.

Solution:

1. On the server where you have installed JasperReports Server, navigate to:
   <<CA Business Intelligence Installed Directory>>\apache-tomcat\webapps\jasperserver-pro\WEB-INF\classes.
2. Open the jasperreports.properties file in Edit mode.
3. Add the following properties at the end of the file.

```
net.sf.jasperreports.export.csv.remove.empty.space.between.rows=true
net.sf.jasperreports.export.csv.remove.empty.space.between.columns=true
net.sf.jasperreports.export.csv.exclude.origin.band.1=pageHeader
net.sf.jasperreports.export.csv.exclude.origin.report.1=*
net.sf.jasperreports.export.csv.exclude.origin.band.2=pageFooter
net.sf.jasperreports.export.csv.exclude.origin.report.2=*
net.sf.jasperreports.export.csv.exclude.origin.band.3=columnHeader
net.sf.jasperreports.export.csv.exclude.origin.report.3=*
net.sf.jasperreports.export.csv.exclude.origin.band.4=columnFooter
net.sf.jasperreports.export.csv.exclude.origin.report.4=*
net.sf.jasperreports.export.csv.exclude.origin.band.5=lastPageHeader
net.sf.jasperreports.export.csv.exclude.origin.report.5=*
net.sf.jasperreports.export.csv.exclude.origin.band.6=summaryPageHeader
```

```
net.sf.jasperreports.export.csv.exclude.origin.band.7=groupHeader
net.sf.jasperreports.export.csv.exclude.origin.band.8=groupFooter
net.sf.jasperreports.export.csv.exclude.origin.band.9=reportHeader
net.sf.jasperreports.export.csv.exclude.origin.band.10=reportFooter
net.sf.jasperreports.export.csv.exclude.origin.band.11=lastPageFooter
net.sf.jasperreports.export.csv.exclude.origin.report.11=*
net.sf.jasperreports.export.csv.exclude.origin.band.12=summaryPageFooter
net.sf.jasperreports.export.csv.exclude.origin.band.13=summary
net.sf.jasperreports.export.csv.exclude.origin.report.13=*
net.sf.jasperreports.export.csv.exclude.origin.band.14=title
net.sf.jasperreports.export.csv.exclude.origin.report.14=*
net.sf.jasperreports.export.csv.parameters.override.IgnorePagination=true
```

4. Save the changes to the properties file and exit.
5. Restart the JasperReports Tomcat Server.
   After you have restarted the JasperReports Tomcat Server, run the reports you want to export, again.
6. Export the report data into .csv format.The report data should be exported without the following information:

- Title
- Header
- Group Header
- Summary
- Footer

View the Modification History of Custom Configuration Files

Symptom:

You are interested in monitoring Spectrum Report Manager customization changes that are made through the configuration files. These files are located under *$SPECROOT*/custom/repmgr/config.

Solution:

Log in to the MySQL client as ‘root’ and run the following command to see the chronology of changes for all the custom configuration files:

```
SELECT filename, FROM_UNIXTIME(last_modified/1000) as time
FROM reporting.configchangelog
ORDER BY filename, time;
```

Missing Outage Data Error

Symptom:

DX NetOps Spectrum Reporting is missing outage data. The Tomcat log includes a message similar to the following message:

```
<$SPECROOT>\tomcat\logs\stdout.log:
    Jul 29, 2009 10:00:34 AM - SRMAvailabilityHandler:
    WARNING: Historical update has failed for domain = 0x400000 due to error = Connection to event domain
 timed out.
```

Solution:

The outage data indicates one of the following situations:

The DX NetOps Spectrum Archive Manager for the domain that is specified in the message is not running.

- A network connectivity issue between the OneClick web server and the Archive Manager on the domain that is specified in the message.

To resolve this issue, start the Archive Manager or resolve any network connectivity issues.

When Spectrum Report Manager determines that the Archive Manager is running again, it automatically retrieves all the historical availability event data that it requires to update the reporting database.

When Archive Manager is not running, the SpectroSERVER caches the event data. When the Archive Manager is running again, the SpectroSERVER sends it to the cached event data. However, the SpectroSERVER event cache is limited in size. If the Archive Manager is down for a prolonged period, event data can be lost. For more information, see the Database Management section.

Invalid Security Credentials Error

Symptom:

The following message appears when you try to run a report:

```
An error occurred at the server: The Page Server cannot logon to the CMS. This is due to invalid security
 credentials. Please verify your user ID and password.
```

Solution:

The session has timed out. To resolve the issue, perform the following steps:

1. Exit Spectrum Report Manager.
2. Re-establish the DX NetOps Spectrum Reporting session, and try running the report again.

Reset Report Manager Application Model

Symptom:

If the Main Location Server (MLS) is removed from a Distributed SpectroSERVER environment, Spectrum Report Manager can no longer assert events on the SRMApplication model. As a result, monitoring of Spectrum Report Manager status through the SRMApplication model cannot occur.

Solution:

Remove the model handle entry for the SRMApplication model from the registry table using the following MySQL command (logged on as ‘root’):

```
mysql>USE reporting;
mysql>UPDATE registry SET SRM_Model = 0;
```

Resolving Java Error in Report Manager Sample (WEBI) Reports

Symptom:

A Java error is observed in the Spectrum Report Manager Sample (WEBI) Reports. When a sample report is opened, following error message is displayed:

```
Java has discovered application components that could indicate a security concern -- Block potentially unsafe
 components (recommended).
```

If yes is selected, java blocks the result of the report from being displayed.

Solution:

This issue occurs when running the browser on Windows with versions higher than Java 6 Update 17. To resolve this issue, perform the following steps:

1. Open Java from Control Panel.
2. Select the Advanced Tab
3. Expand the Security option.
4. Expand the Mixed Code option.
5. Select 'Enable - hide warning and run with protections'.

This setting recovers the original format and allows you to use the most recent version of the JRE instead of rolling back to a previous version.

DX NetOps Spectrum Report Manager uses too much disk space, how can we limit the size of this database?

Symptom: The $SPECROOT/mysql/data/reporting directory is very large and taking up a major part of the disk space.

Resolution: CA suggests having a large enough hard space drive dedicated to Report Manager to store all the data required for your business needs.

DX NetOps Spectrum Report Manager also has an archiving utility. This utility is located under the Admin Tools tab of the Report Manager UI. This utility has three features: Purge, Archive, and All data.
Purge: Removes all data based on retention period days chosen.
Archive: Archives the data into the archive database based on the retention period days chosen.
All data: Keep all data in the reporting database and do not remove anything. All data basically means the archive will not run each night.

Purge is the recommended setting for users with disk space problems.
Purge removes all the data after the retention period chosen. For example, if you set the retention days to 90, then anything older than 90 days is removed on a nightly basis.

Archive can be used but this does not save space. Archive simply moves the data from one database to another. However, the archive database can be dropped. This action would make that amount of space available and upon restarting DX NetOps Spectrum tomcat, the database will be rebuilt with 0 data.

To drop the archive database:

1. Navigate to the $SPECROOT/mysql/bin directory.
2. Type: ./mysql --defaults-file=my-spectrum.cnf -uroot -proot
3. At the Mysql> prompt type: drop database archive;
4. Restart DX NetOps Spectrum Tomcat service.

DX NetOps Spectrum Report Manager does not start up due to errors relating to 'directory'

Symptom: Errors are seen in the $SPECROOT/tomcat/logs/stdout.log (catalina.out *nix) relating to parameter directory is not a 'directory'.

Full error:

<date and time> (SRM/Startup/Container) (com.aprisma.spectrum.app.repmgr.web.servlet.SRMBootstrapServlet) - (ERROR) - Error occurred while initializing SRM components
com.aprisma.spectrum.app.repmgr.exceptions.SRMException: Error occurred while logging custom configuration changes

Caused by: java.lang.IllegalArgumentException: Parameter 'directory' is not a directory

at org.apache.commons.io.FileUtils.listFiles(FileUtils.java:207)
at com.aprisma.spectrum.app.repmgr.common.CustomConfigLogger.logChanges(CustomConfigLogger.java:70)
at
com.aprisma.spectrum.app.repmgr.dc.db.ReportManagerInitializer.logConfigurationChanges(ReportManagerInitializer.java:180)

Resolution: The reason for this error is because the ReportManagerInitializer code is looking for the directory $SPECROOT and then the custom/repmgr/config directory. If this error is seen, it means that directory does not exist.

To resolve, simply create the directory structure $SPECROOT/custom/repmgr/config/events

The config directory is the most important part, but the events directory is also needed. Once this is created, restart the DX NetOps Spectrum Tomcat Service.

Windows:

Go to Services and restart DX NetOps Spectrum Tomcat

*nix:

Go to $SPECROOT/tomcat/bin as the DX NetOps Spectrum install owner
Type - ./stopTomcat.sh
Once stopped type - ./startTomcat.sh

Once restarted, view the log to confirm Report Manager starts and there are no errors in the log file.

Commands for CABI Management on Linux

This section describes basic commands for managing CABI servers and the CABI-related MySQL daemon. You can issue commands from the *CABI Install Directory/bobje directory*.

View CABI-Related Processes

You can view the CABI-related processes to verify the status of all processes. To view CABI-related processes, use the following commands:

To view all processes:

```
ps -ef | grep bobje
```

To view all processes except MySQL processes:

```
ps -ef | grep bobje | grep -v mysql
```

Start and Stop CABI Servers

In certain instances, you can start and stop the CABI servers to fix the CABI-related issues. The following commands can be used to start and stop all the CABI-related servers.

To start all servers:

```
./startservers
```

To start a specific server, use CMC.

To stop all servers:

```
./stopservers
```

To stop a specific server, use CMC.

Start and Stop the CABI-Related SQL Anywhere Daemon

You can start and stop the CABI-related SQL Anywhere daemon while working on DX NetOps Spectrum reporting. Use the following commands:

To start the SQL Anywhere daemon:

```
./sqlanywhere_startup.sh
```

To stop the SQL Anywhere daemon:

```
./sqlanywhere_shutdown.sh
```

Start and Stop CABI Tomcat

When working on processes related to DX NetOps Spectrum reporting (for example, backup or restore reporting data), we recommend you to stop and start the CABI Tomcat process. Use the following commands:

To start the Tomcat process:

```
./tomcatstartup.sh
```

To stop the Tomcat process:

```
./tomcatshutdown.sh
```

How to Manually Purge Reporting Data from the Reporting Database

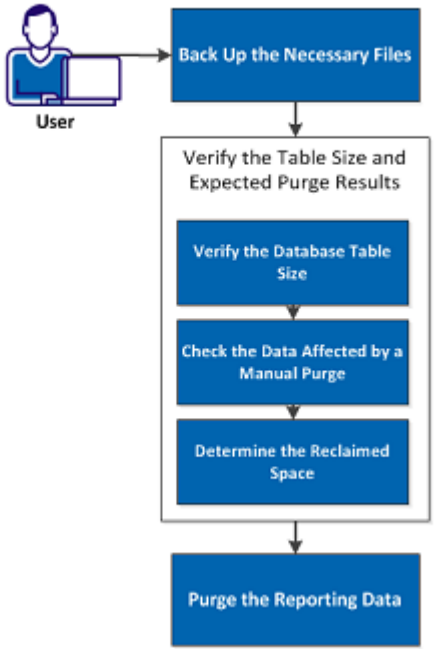
The performance of the database is affected when the tables in the reporting database grow too large. To improve database performance, you can purge the data manually from the reporting database. The following tables affect the database performance in the reporting database:

- modeloutage
- alarminfo
- alarmactivity
- spmbasictestresults
- spmhttpfulltestresults
- spmjittertestresults

Event Tables are also considered as major volume drivers in the reporting database. You can configure the data retention period that is available for Event Tables and can improve the database performance.

The following diagram illustrates the process to manually purge reporting data from the reporting database:

How to Manually Purge Reporting Data from the Reporting Database



Perform the following tasks to manually purge reporting data:

1. Back Up the Necessary Files
2. Verify the Table Size and Expected Purge Results
   - Verify the Database Table Size
   - Check the Data Affected by a Manual Purge
   - Determine the Reclaimed Space
3. Purge the Reporting Data

Back Up the Necessary Files

Back up the database or at least the tables that are affected before purging data from the existing database. MySQL database tables are stored as a set of files. You can back up the database using the following methods:

- Use the mysqldump utility, as described in Back Up DX NetOps Spectrum Reporting and Archive Data.
- Create backup copies of selected files, as described in the following procedure.

Follow these steps:

1. Stop the DX NetOps Spectrum Tomcat service.
2. Stop the DX NetOps Spectrum MYSQL Database Server.
3. Stop the Archive Manager if it is running on this server.
4. Perform one of the following steps to copy the necessary files:
   - To back up the entire reporting database, copy the <*$SPECROOT*>/mysql/data/reporting directory to a newly named directory (for example, reporting_backup) within the <*$SPECROOT*>/mysql/data directory.

NOTE
By creating the backup directory in this location you can easily switch between the reporting database and the reporting_backup database within the MySQL console window.

- To back up only those tables that are purged, copy the following files from the <*$SPECROOT*>/mysql/data/reporting directory to a newly named directory (for example, reporting_backup) within the <*$SPECROOT*>/mysql/data directory:

```
alarmactivity.*
alarminfo.*
modeloutage.*
```

- If you use SPM, copy the following files too:

```
spmbasictestresults.*
spmhttpfulltestresults.*
spmjittertestresults.*
```

NOTE
Typically, MySQL creates three different file types for each database table: .frm, .MYD, and .MYI. Copy all the three files. If these files are copied, renamed and stored within the reporting directory, then MySQL treats them as additional database tables.

5. Restart the DX NetOps Spectrum MySQL Database Server service.
   The file backup is complete.

Verify the Table Size and Expected Purge Results

You can evaluate the amount of data that is affected before you start purging the data. You can use a few SQL commands to examine the current volume and capacity of the tables.

WARNING
In the SQL sample syntax that we have provided, a value of "2007-04-01" (April 1st, 2007) is used as an example cutoff date. This sample syntax indicates that records that are created before this date are deleted. Substitute an appropriate date. However, once you have selected a date, use the same date across all tables to maintain the data integrity.

Take the following steps to verify the table size and expected purge results:

1. Verify the Database Table Size
2. Check the Data Affected by a Manual Purge
3. Determine the Reclaimed Space

Verify the Database Table Size

You can use the SHOW TABLE STATUS command to verify the size of a database table.

Follow these steps:

1. Open a command prompt, and execute the following command in <*$SPECROOT*>/mysql/bin:

```
mysql -uroot -proot reporting
```

2. At the mysql> prompt, execute the following SQL commands:

```
SHOW TABLE STATUS LIKE "modeloutage";
SHOW TABLE STATUS LIKE "alarminfo";
SHOW TABLE STATUS LIKE "alarmactivity";
```

   Table statistics appear.
3. Verify the following fields:

- rows - the total number of rows
- avg_row_length - average length (in bytes) of each row
- data_length - current size (in bytes) of the table
- max_data_length - maximum size (in bytes) that the table can grow to

4. If you are using SPM, execute the following commands:

```
SHOW TABLE STATUS LIKE "spmbasictestresults";
SHOW TABLE STATUS LIKE "spmjittertestresults";
SHOW TABLE STATUS LIKE "spmhttpfulltestresults";
```

Table size verification is complete.

Check the Data Affected by a Manual Purge

You can use the SELECT COUNT(*) command to find the number of rows that are affected by the data purge.

Follow these steps:

1. At the mysql> prompt, execute the following SQL commands:

```
SELECT COUNT(*) FROM modeloutage
WHERE end_time < "2007-04-01" AND outage_type > 0;
SELECT COUNT(*) FROM alarminfo
WHERE clear_time < "2007-04-01";
SELECT COUNT(*) FROM alarminfo, alarmactivity
WHERE alarminfo.alarm_key = alarmactivity.alarm_key
AND alarminfo.clear_time < "2007-04-01";
```

2. If you are using SPM, run the following commands:

```
SELECT COUNT(*) FROM spmbasictestresults
WHERE timestamp < "2007-04-01";
SELECT COUNT(*) FROM spmjittertestresults
WHERE timestamp < "2007-04-01";
SELECT COUNT(*) FROM spmhttpfulltestresults
WHERE timestamp < "2007-04-01";
```

The affected data identification is complete.

Determine the Reclaimed Space

You can use results from the SHOW TABLE STATUS and SELECT COUNT(*) statements in the previous procedures to find the amount of space that is freed after the purge.

Follow these steps:

1. Execute the following command to get the table size:

```
SHOW TABLE STATUS LIKE "modeloutage";
```

From the results, it is determined that the average row length (avg_row_length) is 121 bytes.

2. Execute the following command to find the number of rows that are affected:

```
SELECT COUNT(*) FROM modeloutage
WHERE end_time < "2007-04-01" AND outage_type > 0;
```

From the results, the count value is 4851.

3. Perform the following calculation to determine the amount of space that would be freed for this table:

```
Avg_row_length * (number of rows Affected) = freed space
121 bytes * 4851 = 586,971 bytes.
```

The amount of space reclaimed is determined.

NOTE
After you delete the data, you can optimize the table to reclaim unused space.

Purge the Reporting Data

A series of DELETE commands are used to delete data. After data is deleted, it is important to optimize the tables to reclaim unused space.

WARNING
In the SQL sample syntax that we have provided, a value of "2007-04-01" (April 1st, 2007) is used as an example cutoff date. This sample syntax indicates that records that are created before this date are deleted. Substitute an appropriate date. However, once you have selected a date, use the same date across all tables to maintain the data integrity.

Follow these steps:

WARNING
The order in which the data is deleted must be strictly followed to avoid database corruption.

1. At the mysql> prompt, enter the following SQL commands:
```
DELETE FROM modeloutage
  WHERE end_time < "2007-04-01" AND outage_type > 0;
DELETE alarmactivity FROM alarmactivity, alarminfo
  WHERE alarminfo.alarm_key = alarmactivity.alarm_key
  AND alarminfo.clear_time < "2007-04-01";
DELETE FROM alarminfo
  WHERE alarminfo.clear_time < "2007-04-01";
```
NOTE
Execution time of these commands depends on the number of records that are effected.
2. If you are using SPM, run the following commands:
```
DELETE FROM spmbasictestresults
  WHERE timestamp < "2007-04-01";
DELETE FROM spmjittertestresults
  WHERE timestamp < "2007-04-01";
DELETE FROM FROM spmhttpfulltestresults
  WHERE timestamp < "2007-04-01";
```
3. After the data is deleted, enter the following command to reclaim unused space:
```
OPTIMIZE TABLE modeloutage, alarmactivity, alarminfo;
```
NOTE
Execution time of this command depends on the size of the tables.
4. If you are using SPM, run the following command:
```
OPTIMIZE TABLE spmbasictestresults, spmjittertestresults,
  spmhttpfulltestresults;
```
5. (optional) Run the following commands to verify space savings:
```
SHOW TABLE STATUS LIKE "modeloutage";
SHOW TABLE STATUS LIKE "alarminfo";
SHOW TABLE STATUS LIKE "alarmactivity";
```
6. (optional) If you are using SPM, run the following commands:
```
SHOW TABLE STATUS LIKE "spmbasictestresults";
SHOW TABLE STATUS LIKE "spmjittertestresults";
SHOW TABLE STATUS LIKE "spmhttpfulltestresults";
```
7. Restart Spectrum Tomcat service and Archive Manager (if previously stopped on this server).

Data purging is complete.

Reporting Database Management

Spectrum Report Manager uses a MySQL database named reporting to store data. As with any database, you can maintain this database to help it function efficiently. This section describes utilities and procedures that you can use to manage the reporting database.

Initialize the Database for Specific Landscapes

This section describes a utility that you can use to optimize and initialize the database for a specific landscape. Perform several actions before initializing the database.

Follow these steps:

1. Navigate to the /spectrum/bin directory.
2. Use the following default MySQL administrator login credentials:
   username:root
   password:root

   NOTE
   Use a bash shell or command prompt to run the utility. Do not run it from a MySQL prompt.
3. Run the following command-line utility to remove all data for one or more or all landscapes from the reporting database:

   ```
   RpmgrInitializeLandscape
   RpmgrInitializeLandscape username password
   -skipInitialHistory -initHist # of days -all
   ```

   – **Usage:**

   ```
   landscape1 landscape2 ...
   ```

   – **Definitions:**
   – **- skipInitialHistory**
   Spectrum Report Manager does not retrieve or store events during event processing that have occurred before the utility is run. This flag overrides -initHist *# of days* if it is also included in the command line.
   – **- initHist # *of days***
   Spectrum Report Manager processes initial historical events from the past number of days that are specified before the utility is run.
   – **- all**
   Spectrum Report Manager removes data for all landscapes in the reporting database.
   – ***landscape1 landscapeN***
   Spectrum Report Manager removes data for each specified landscape.

Back Up DX NetOps Spectrum Reporting and Archive Data

Backing up reporting and archive data lets you maintain the reporting continuity. If you lose data (for example, during an upgrade installation), you can recover the data by restoring the backups.

Procedure for Windows

Follow these steps:

1. Select Control Panel, Administrative Tools, Services, and then Spectrum Tomcat.
   The Spectrum Tomcat Properties dialog opens.
2. Click Stop to stop Tomcat.

3. Navigate to the *$SPECROOT*/mysql/bin directory.
4. Using the mysqldump utility, back up the reporting and archive databases using this command:
   ```
   mysqldump --routines --databases -uroot -proot reporting archive > backup_filename.sql
   ```
5. Start Tomcat.

Procedure for Linux

Follow these steps:

1. Stop Tomcat.
   ```
   $SPECROOT/tomcat/bin/stopTomcat.sh
   ```
2. Navigate to the *$SPECROOT*/mysql/bin directory.
3. Using the mysqldump utility, back up the reporting and archive databases using this command:
   ```
   mysqldump --defaults-file=../my-spectrum.cnf -uroot -proot --routines --database reporting archive >
    backup_filename.sql
   ```
4. Start Tomcat.
   ```
   $SPECROOT/tomcat/bin/startTomcat.sh
   ```
   The reporting and archive data backup is complete.

Restore DX NetOps Spectrum Reporting and Archive Data

You can restore the reporting and archive data from a backup copy. Stop Tomcat before restoring the data from a backup copy.

Procedure for Windows

Follow these steps:

1. To stop Tomcat, select Control Panel, Administrative Tools, Services, Spectrum Tomcat, and click Stop in the Spectrum Tomcat Properties box.
2. Navigate to the *$SPECROOT*/mysql/bin directory.
3. Load the reporting and archive data from a backup copy using this command:
   ```
   mysql -uroot  - p root_pw reporting < backup_filename.sql
   ```
4. Start Tomcat.

Procedure for Linux

Follow these steps:

1. Stop Tomcat.
   ```
   $SPECROOT/tomcat/bin/stopTomcat.sh
   ```
2. Navigate to the *$SPECROOT*/mysql/bin directory.
3. Load the reporting and archive data from a backup copy using this command:
   ```
   mysql --defaults-file=../my-spectrum.cnf -uroot  - p root_pw reporting < backup_filename.sql
   ```
4. Start Tomcat.
   ```
   $SPECROOT/tomcat/bin/startTomcat.sh
   ```
   The reporting and archive data is restored from a backup copy.

   NOTE
   The mysql restore commands (for Windows or Linux) restore both reporting and archive data even though only reporting database name is mentioned in these commands. The reason is that the backup of both the databases is taken to same file.

Reconcile DX NetOps Spectrum Report Data

The reporting database needs to be re-initialized whenever the SpectroSERVER Database is restored.

The reporting data re-initialization process flushes the historical data and re-polls the SRM data from the beginning (available in the archive manager), which may cause some data loss, and it also consumes time to re-poll.

To avoid the re-initialization, run the reconcile_reporting_db script. The reconcile_reporting_db script reads the SSdbLoadTime (ID: 0x13373) attribute on VNM model of the domain, which specifies the SSDB save time. The script cleans the reporting data (related to devices/models), which is discovered after the SSDB save time. It helps to avoid duplication of data when the devices are re-discovered in SSDB.

Following is the tool to clean Report Manager database after loading historical SSDB:

```
bash-3.2$ ./reconcile_reporting_db.sh -h
```

Script usage:

```
reconcile_reporting_db.sh -s <server> -p <port> -u <user> -w <password> -l <landscape name> [-n <No Of Days
 From Now>] [-d] [-h]
```

Where:

-s <server name>
OneClick Tomcat Server Host name
-p <port>
OneClick Tomcat Server Web port
-u <user>
OneClick Web User name
-w <password>
OneClick Web User password
-l <landscape name>
Landscape name to poll
-n <No Of Days>
(Optional) Number of days from now. (Use when the 'SSdbLoadTime' parameter is not available on the VNM model)
-d
(Optional) Enables debugging
-h
(Optional) Prints the usage help

Report Manager Utility Scripts

The Spectrum Report Manager utility scripts are used to perform a specific task in the reporting database. For example, the backup utility gathers information that is specific to a landscape and dumps it into another database schema. Each database backup has an entry that is stored in its MySQL table for later reference.

The recovery utility initializes the landscape (using RpmgrInitializeLandscape) and copies all the table entries from the backup database into the current database.

> **NOTE**
> We recommend performing these operations when Tomcat is offline to avoid populating tables in the reporting database.

> **NOTE**
> The following Spectrum Report Manager utility scripts are commonly used in the reporting database:

BackupReportingDBLandscape

The BackupReportingDBLandscape utility script captures a landscape data to a backup database. Use the following format to execute this command:

```
BackupReportingDBLandscape user password domain name [description]
```

user

Indicates MySQL username.

- **password**
  Indicates MySQL password.
- **domain name**
  Indicates the SpectroSERVER domain name.
- **description**
  Describes the backup comments.

DisplayReportingDBBackups

The DisplayReportingDBBackups script displays the backups that exist on the system. You can use this script, if you are removing backups by database name from CLI.

Use the following format to execute this command:

```
DisplayReportingDBBackups mysql user password domain name
```

mysql user

Indicates MySQL username.

- **password**
  Indicates MySQL password.
- **domain name**
  Indicates the SpectroSERVER domain name.

exemptOutagesForDay

The ExemptOutagesForDay script converts unplanned outages to exempt outages when the outages coincide with an exemption period (for example, a bank Holiday).

Use the following format to execute this command:

```
exemptOutagesForDay mysql username mysql password exempt service outages [-undo YYYY-MM-DD] YYYY-MM-DD day
```

mysql username

Indicates MySQL username.

- **mysql password**
  Indicates MySQL password.
- **exempt service outages**
  Exempts the service outages.

Example:

```
exemptOutagesForDay user pass yes 2010-01-01 New Year\'s Day
```

RecoverReportingDBLandscape

The RecoverReportingDBLandscape script recovers a single landscape in the reporting database from a backup database.

Use the following format to execute this command:

```
RecoverReportingDBLandscape user password backup database name
```

user

Indicates MySQL username.

- **password**
  Indicates MySQL password.
- **backup database name**
  Indicates the name of the backup database.

RemoveReportingDBBackups

The RemoveReportingDBBackups script removes a backup database that is created for a landscape in Spectrum Report Manager. Use the following format to execute this command:

```
RemoveReportingDBBackups user password backup database name
```

user

Indicates MySQL username.

- **password**
  Indicates MySQL password.
- **backup database name**
  Indicates the name of the backup database.

Appendix A. DX NetOps Spectrum Events Used by Report Manager

This appendix lists DX NetOps Spectrum events that Spectrum Report Manager uses to start and stop calculating outage time for devices and interfaces. It also lists Service Performance Manager events.

Outage Events

This section lists events that mark the beginning and end of either a planned or unplanned model outage.

DOWN events indicate that an unplanned outage has begun.

- CONTACT LOST (0x10302)
- PORT BAD (0x10d11)
- PORT DISABLED (0x10d12)
- PORT UNREACHABLE (0x10d14)
- PORT LOWER LAYER DOWN (0x10d16)
- PORT CONNECTED TO DOWN PORT OR DEVICE (0x10d18)
- PORT BAD BUT CONNECTED TO WALINK EVENT (0x10d2d)
- PORT LOST (0x10d66)
- APPLICATION LOST (0x10d09)
- DEVICE UNRESPONSIVE (0x10d35)

UP events indicate that an unplanned outage has ended.

- CONTACT ESTABLISHED (0x10301)
- PORT GOOD (0x10d10)
- PORT REACTIVATED (0x10d66)
- PORT UP BUT LINKED WITH DOWN PORT (0x10d17)
- APPLICATION REACTIVATED (0x10d0b)
- DEVICE RESPONSIVE EVENT (0x10d30)

IN MAINT MODE events indicate that a planned outage has begun. If a model is in an unplanned outage state when the model enters maintenance mode, the unplanned outage ends immediately.

- DEVICE INTO HIBERNATE (0x10226)
- DEVICE INTO MAINTENANCE (0x10222)
- PORT INTO MAINTENANCE (0x10224)

OUT OF MAINT MODE events indicate that a planned outage has ended.

- DEVICE OUT OF HIBERNATE (0x10227)
- DEVICE OUT OF MAINTENANCE (0x10223)
- PORT OUT OF MAINTENANCE (0x10225)

Additional events that are of interest to Availability reporting:

- VNM STARTED (0x10101)
- VNM STOPPED (0x10102)
- MODEL DESTROYED (0x10202)

The following list of events is used for the calculation of availability reports for Clusters. Standard events that are mentioned earlier are ignored if Cluster availability events exist.

List of UP and DOWN events for Microsoft (MS) Clusters:

MS Clusters

UP event

- CONTACT_LOST_EVENT (0x10302)
- DEVICE_UNRESPONSIVE (0x10d35)

DOWN event

- CONTACT_ESTABLISHED_EVENT (0x10301)
- DEVICE_RESPONSIVE (0x10d30)

MS Node

UP event

- MSCS_NODE_STATE_UNKNOWN_EVENT (0x0621002b)

DOWN event

- MSCS_NODE_STATE_UP_EVENT (0x06210029)
- MSCS_NODE_STATE_DOWN_EVENT (0x0621002a)

MS Resource Group

UP event

- MSCS_RESOURCE_GROUP_STATE_ONLINE_EVENT (0x0621002e)

DOWN event

- MSCS_RESOURCE_GROUP_STATE_OFFLINE_EVENT (0x0621002f)
- MSCS_RESOURCE_GROUP_STATE_FAILED_EVENT (0x06210030)
- MSCS_RESOURCE_GROUP_STATE_UNKNOWN_EVENT (0x0621002d)

MS Resources

UP event

- MSCS_RESOURCE_STATE_ONLINE_EVENT (0x06210034)

DOWN event

- MSCS_RESOURCE_STATE_UNKNOWN_EVENT (0x06210033)
- MSCS_RESOURCE_STATE_OFFLINE_EVENT (0x06210035)
- MSCS_RESOURCE_STATE_FAILED_EVENT (0x06210036)

List of UP and Down events for IBM clusters:

IBM Cluster

UP event

- IBM_CLUSTER_STATE_UP_EVENT (0x06210014)

DOWN event

- IBM_CLUSTER_STATE_DOWN_EVENT (0x06210015)
- IBM_CLUSTER_STATE_UNKNOWN_EVENT (0x06210016)
- IBM_CLUSTER_STATE_NOTCONFIGURED_EVENT (0x06210017)

IBM Node

UP event

- IBM_NODE_STATE_UP_EVENT (0x0621001a)

DOWN event

- IBM_NODE_STATE_DOWN_EVENT (0x0621001b)
- IBM_NODE_STATE_UNKNOWN_EVENT (0x0621001c)

IBM Resource Group

UP event

- IBM_RESOURCE_GROUP_STATE_ONLINE_EVENT (0x0621001e)

DOWN event

- IBM_RESOURCE_GROUP_STATE_OFFLINE_EVENT (0x0621001f)
- IBM_RESOURCE_GROUP_STATE_UNKNOWN_EVENT (0x06210020)
- IBM_RESOURCE_GROUP_STATE_ERROR_EVENT (0x06210021)

> **NOTE**
> For IBM Resources, events are not defined by IBM.

Alarm Events

ALARM events are events that affect alarms.

- ALARM SET(0x10701)
- ALARM CLEARED (0x10702)
- USER CLEARED ALARM (0x10706)
- ALARM UPDATED (0x10707)
- SECONDARY ALARM SET EVENT (0x10714)
- SECONDARY ALARM CLEAR EVENT (0x10715)

Additional events that are of interest to Alarm reporting:

- VNM STARTED (0x10101)
- VNM STOPPED (0x10102)
- MODEL DESTROYED (0x10202)

Model Name Changes

Model name change events are used to update and track name changes for DX NetOps Spectrum models:

- MODEL NAME CHANGE (0x1a100)

Appendix B. DX NetOps Spectrum Reporting Application Model Events and Alarms

This appendix describes the events and alarms that you can monitor from the DX NetOps Spectrum Reporting Application model.

Application Events

DX NetOps Spectrum Reporting generates the following events on the DX NetOps Spectrum Reporting Application model:

- Spectrum Report Manager is not monitoring any landscapes.
  Asserts alarm - Spectrum Report Manager: NO LANDSCAPES MONITORED
- Landscape X has been added to Spectrum Report Manager list of monitored landscapes.
  Clears alarm- Spectrum Report Manager: NO LANDSCAPES MONITORED
- Landscape X has been removed from the Spectrum Report Manager list of monitored landscapes.
  Clears alarms- Spectrum Report Manager: LANDSCAPE CONTACT LOST and Spectrum Report Manager: ARCHIVE MANAGER CONTACT LOST
- Spectrum Report Manager has lost contact with the X landscape.
  Asserts alarm - Spectrum Report Manager: LANDSCAPE CONTACT LOST
- Spectrum Report Manager has regained contact with the X landscape.
  Clears alarm - Spectrum Report Manager: LANDSCAPE CONTACT LOST
- Spectrum Report Manager has lost contact with the X Archive Manager.
  Asserts alarm - Spectrum Report Manager: ARCHIVE MANAGER CONTACT LOST
- Spectrum Report Manager has regained contact with the X archive manager.
  Clears alarm - Spectrum Report Manager: ARCHIVE MANAGER CONTACT LOST
- Spectrum Report Manager has encountered an error while processing events. (for more information, see the *OneClick log file*)
  Asserts alarm - Spectrum Report Manager: EVENT PROCESSING FAILURE
- The Spectrum Report Manager server is stopping. Alarms that are based on landscape {S 1} are cleared and reasserted at startup, if needed.
  Clears alarms - Spectrum Report Manager: LANDSCAPE CONTACT LOST and Spectrum Report Manager: ARCHIVE MANAGER CONTACT LOST for each landscape that SRM is monitoring.
- The Spectrum Report Manager server is stopping. Event processing failure alarms are cleared and reasserted at startup, if needed.
  Clears alarm - Spectrum Report Manager: EVENT PROCESSING FAILURE

Application Alarms

DX NetOps Spectrum Reporting generates the following alarms from DX NetOps Spectrum Reporting Application model events:

- Spectrum Report Manager: NO LANDSCAPES MONITORED, Alarm Severity - Yellow
- Spectrum Report Manager: LANDSCAPE CONTACT LOST, Alarm Severity - Orange
- Spectrum Report Manager: ARCHIVE MANAGER CONTACT LOST, Alarm Severity - Orange
- Spectrum Report Manager: EVENT PROCESSING FAILURE, Alarm Severity - Red

Appendix C. DX NetOps Spectrum Attributes Used by Spectrum Reporting

This appendix lists DX NetOps Spectrum attributed that DX NetOps Spectrum Reporting uses in Asset, Availability, and Change Management reports. DX NetOps Spectrum Reporting polls DX NetOps Spectrum for these attribute values every 24 hours.

Device Attributes

0x1006e: MODEL_NAME_ATTR_ID

0x11ee8: MODEL_CLASS_ATTR_ID

0x11b41: CREATE_TIME_ATTR_ID

0x11026: MODEL_CREATOR_ATTR_ID

0x10001: MODEL_TYPE_ATTR_ID

0x10009: SECURITY_STRING_ATTR_ID

0x1027f: IP_ATTR_ID

0x110df: MAC_ATTR_ID

0x10030: SERIAL_NUMBER_ATTR_ID

0x10052: SYS_DESC_ATTR_ID

0x10053: SYS_OID_ATTR_ID

0x1102e: LOCATION_ATTR_ID

0x10b5a: CONTACT_PERSON_ATTR_ID

0x10245: SYS_UPTIME_ATTR_ID

0x23000e: DEVICE_TYPE

0x110ed:CONTACT_STATUS_ID

0x12a6d:NRM_LINE_CARD_DATA_ATTR_ID

Interface Attributes

0x1006e: MODEL_NAME_ATTR_ID

0x11ee8: MODEL_CLASS_ATTR_ID

0x11b41: CREATE_TIME_ATTR_ID

0x10001: MODEL_TYPE_ATTR_ID

0x10009: SECURITY_STRING_ATTR_ID

0x129ed: PORT_TYPE_ATTR_ID

0x129e0: PORT_DESC_ATTR_ID

0x11ee3: IF_SPEED_ATTR_ID

0x1027f: IP_ATTR_ID

0x10e43: PORT_IP_ATTR_ID

0x110df: MAC_ATTR_ID

0x10f1b: PORT_LINK_STATUS

0x12980: IF_LAST_CHANGE_ATTR_ID

0x10e41: IF_IN_OCTETS_ATTR_ID

0x11f82: IF_ALIAS

0x1006a: COMPONENT_OID

0x10000: MODEL_TYPE_NAME_ATTR_ID

User Defined Attributes

0x12bfb: USER_AssetTag

0x12bfc: USER_AssetID

0x12bfd: USER_AssetOwner

0x12bfe: USER_AssetOrganization

0x12bff: USER_AssetOffice

0x12c00: USER_AssetContractNumber

0x12c01: USER_AssetContractStartDate

0x12c02: USER_AssetContractEndDate

0x12c03: USER_AssetDescription

Appendix D. DX NetOps Spectrum Report Manager Database API (SRMDBAPI)

The DX NetOps Spectrum Spectrum Report Manager Database API (SRMDBAPI) provides a fully documented set of read-only database objects to support your custom data analysis requirements. Specifically, the SRMDBAPI consists of a set of database views that are contained within a dedicated multidimensional schema in the MySQL instance that is used by Spectrum Report Manager.

The SRMDBAPI contains the following basic content areas:

- Asset
- Alarm
- Outage/Availability
- Event

The following add-ons are part of the SRMDBAPI but require additional license purchase for usage:

- SPM
- NCM

> **NOTE**
> The SRMDBAPI is a read-only API. Data modification is not supported.

Design Methodology

The SRMDBAPI has been implemented using a multidimensional modeling approach. The multidimensional model is selected due to its inherently flexible design. This model is not driven by the requirements of a specific report or set of reports. Rather, the multidimensional schema is optimized for the analysis of facts (for example, event, outage) across any number of related dimensions (for example, model, time). The multidimensional model is ideally suited for reports. However, this particular schema design does not preclude you in extracting data from the supplied views into other repositories.

How to Establish Remote Access

At installation, the SRMDBAPI database objects are available to a pre-established 'srmapi' MySQL database user. If more user(s) require access to the dedicated API schema, you can establish them manually.

For more information, see How to Create Additional SRMDBAPI Users

> **NOTE**
> If you are connecting from a remote server using the 'srmapi' database user, additional grants need to be performed.

Follow these steps:

1. On the Spectrum Report Manager server, log in to mysql as 'root'.
2. To provide remote access to the ‘srmapi’ database user, issue the following grants:

```
mysql>GRANT SELECT, EXECUTE ON srmdbapi.* TO 'srmapi'@'%';
mysql>GRANT SELECT ON reporting.* TO 'srmapi'@'%';
mysql>FLUSH PRIVILEGES;
```

3. Logout of mysql.
   Remote Access is established.

Example Use Cases

The SRMDBAPI feature is enabled to provide access to mission critical Spectrum Report Manager data. We published our SRMDBAPI to point your Business Intelligence (BI) tools to valuable DX NetOps Spectrum data. Some of the possible use cases are as follows:

- Query this critical data using the BI tool in which your company has already invested.
- Extract the Spectrum Report Manager data and place it within another data repository.
- Incorporate Spectrum Report Manager data into a separate CMDB or financial database.

Inventory of SRMDBAPI Views

Each view that is contained in the SRMDBAPI includes the data type and description.

> **NOTE**
> For the Key column, the following legend applies:

UNQ - unique

PK - primary key

- FK - foreign key

The following views are presented:

- v_dim_alarm_condition
- v_dim_alarm_title
- v_dim_alarm_user
- v_dim_device_model
- v_dim_device_module
- v_dim_event
- v_dim_event_creator
- v_dim_global_collection_member
- v_dim_interface_model
- v_dim_landscape
- v_dim_model
- v_dim_ncm_event
- v_dim_spm_test
- v_dim_time
- v_fact_alarm_activity
- v_fact_alarm_info
- v_fact_event
- v_fact_model_outage
- v_fact_spm_basic_test_results
- v_fact_spm_http_full_test_results
- v_fact_spm_jitter_test_results

v_dim_alarm_condition

This view enumerates the various alarm conditions (for example, Minor, Major) and associated criticality values.

Field	Key	Type	Length	Description
condition_id	PK	int(unsigned)	10	Internal ID/Key that uniquely identifies a record in this view
condition_name		varchar	11	Condition Name
criticality	UNQ	tinyint	2	Criticality (1=Maintenance, 2=Minor, 3=Major, 4=Critical)

v_dim_alarm_title

This view enumerates the various alarm titles and associated probable causes that have occurred in the reporting database.

Field	Key	Type	Length	Description
alarm_title_id	PK	int(unsigned)	10	Internal ID/Key that uniquely identifies a record in this view
alarm_title		varchar	255	Alarm Title
pcause_id_hex		varchar	24	Probable Cause Code (hexadecimal form)

pcause_id_dec		int(unsigned)	10	Probable Cause Code (decimal form)
pcause_title		varchar	100	Probable Cause Title

v_dim_alarm_user

This view enumerates the various usernames that are associated with alarm activity captured in the reporting database.

Field	Key	Type	Length	Description
alarm_user_key	PK	int(unsigned)	10	Internal ID/Key that uniquely identifies a record in this view
alarm_user_name	UNQ	char	255	Username

v_dim_device_model

This view enumerates all devices (active and destroyed) that are captured historically in the reporting database.

Field	Key	Type	Length	Description
model_key	PK	int(unsigned)	10	Internal ID/Key that uniquely identifies a record in this view
model_h_dec		int(unsigned)	10	Model handle (decimal form)
model_h_hex		varchar	24	Model handle (hexadecimal form)
landscape_h_dec	FK	int(unsigned)	10	Landscape handle (decimal form); join to v_dim_landscape for additional landscape information.
landscape_h_hex		varchar	24	Landscape handle (hexadecimal form)
model_name		varchar	4000	Device Name
create_time		datetime		Creation Time
model_creator		varchar	255	Model Creator
security_string		varchar	255	Security String
destroy_time		datetime		Destroy Time
device_type		varchar	255	Device Type
ip		varchar	255	Network Address
mac		varchar	32	MAC Address
serial_nbr		varchar	255	Serial Number
sys_desc		varchar	255	System Descriptor
fw_rev		varchar	255	Firmware Version
sys_oid		varchar	255	System Object ID
location		varchar	255	Location

contact_person		varchar	255	Contact Person
last_reboot		datetime		Last reboot time
last_reboot_text		varchar	19	Last reboot time (text form)
last_successful_poll		datetime		Last successful poll time
model_destroyer		varchar	255	Model Destroyer
cust_asset_tag		varchar	255	Asset Tag
cust_asset_id		varchar	255	Asset ID
cust_asset_owner		varchar	255	Asset Owner
cust_asset_organization		varchar	255	Asset Organization
cust_asset_office		varchar	255	Asset Office
cust_asset_contract number		varchar	255	Asset Contract Number
cust_asset_contract startdate		varchar	255	Asset Contract Start Date
cust_asset_contract enddate		varchar	255	Asset Contract End Date
cust_asset_description		varchar	255	Asset Description
sdm_host_address		varchar	255	SDM Host Address
mclass_name		varchar	32	Model Class Name
mtype_h_dec		int(unsigned)	10	Model Type Handle (decimal form)
mtype_h_hex		varchar	24	Model Type Handle (hexadecimal form)
mtype_name		varchar	128	Model Type
vendor_name		varchar	32	Vendor Name
topology_model_name_string		varchar	4000	Topology Model Name String; this field can be used to support container-based reporting capabilities.
varchar_1_attrid_hex		varchar	24	Custom Attribute ID (Varchar1)
varchar_2_attrid_hex		varchar	24	Custom Attribute ID (Varchar2)
varchar_3_attrid_hex		varchar	24	Custom Attribute ID (Varchar3)
varchar_4_attrid_hex		varchar	24	Custom Attribute ID (Varchar4)
integer_1_attrid_hex		varchar	24	Custom Attribute ID (Integer1)
integer_2_attrid_hex		varchar	24	Custom Attribute ID (Integer2)
integer_3_attrid_hex		varchar	24	Custom Attribute ID (Integer3)

integer_4_attrid_hex		varchar	24	Custom Attribute ID (Integer4)
datetime_1_attrid_hex		varchar	24	Custom Attribute ID (Datetime1)
datetime_2_attrid_hex		varchar	24	Custom Attribute ID (Datetime2)
varchar_1_label		varchar	255	Custom Attribute Label (Varchar1)
varchar_2_label		varchar	255	Custom Attribute Label (Varchar2)
varchar_3_label		varchar	255	Custom Attribute Label (Varchar3)
varchar_4_label		varchar	255	Custom Attribute Label (Varchar4)
integer_1_label		varchar	255	Custom Attribute Label (Integer1)
integer_2_label		varchar	255	Custom Attribute Label (Integer2)
integer_3_label		varchar	255	Custom Attribute Label (Integer3)
integer_4_label		varchar	255	Custom Attribute Label (Integer4)
datetime_1_label		varchar	255	Custom Attribute Label (Datetime1)
datetime_2_label		varchar	255	Custom Attribute Label (Datetime2)
varchar_1_value		varchar	4000	Custom Attribute Value (Varchar1)
varchar_2_value		varchar	4000	Custom Attribute Value (Varchar2)
varchar_3_value		varchar	4000	Custom Attribute Value (Varchar3)
varchar_4_value		varchar	4000	Custom Attribute Value (Varchar4)
integer_1_value		bigint	20	Custom Attribute Value (Integer1)
integer_2_value		bigint	20	Custom Attribute Value (Integer2)
integer_3_value		bigint	20	Custom Attribute Value (Integer3)
integer_4_value		bigint	20	Custom Attribute Value (Integer4)
datetime_1_value		datetime		Custom Attribute Value (Datetime1)
datetime_2_value		datetime		Custom Attribute Value (Datetime2)

v_dim_device_module

This view enumerates more information at the slot level for chassis-based devices.

Field	Key	Type	Length	Description
module_id	PK	int(unsigned)	10	Internal ID/Key that uniquely identifies a record in this view
model_key	FK	int(unsigned)	10	Model Key associated with the parent device recorded in v_dim_device_model view; use this field to join to v_dim_device_model for additional device information.
module_index		int	10	Module Index (Slot)
module_name		varchar	255	Module Name (Description)
serial_nbr		varchar	255	Serial Number
software_rev		varchar	255	Software Version

v_dim_event

This view enumerates all of the Event Types encountered while processing events for reporting purposes.

Field	Key	Type	Length	Description
type_dec	PK	int(unsigned)	10	Event Type (decimal form); this field also uniquely identifies a record in this view.
type_hex		varchar	24	Event Type (hexadecimal form)
title		varchar	255	Event Title

v_dim_event_creator

This view enumerates all the event creators that are encountered while processing events for reporting purposes.

Field	Key	Type	Length	Description
creator_id	PK	int(unsigned)	10	Internal ID/Key that uniquely identifies a record in this view
creator_name		varchar	255	Creator Name

v_dim_global_collection_member

This view enumerates all global collection members in the reporting database. You have a separate record for every global collection/model pairing.

Field	Key	Type	Length	Description
gc_rec_ID	PK	int(unsigned)	10	Internal ID/Key that uniquely identifies a record in this view
gc_name		varchar	255	Global Collection Name
model_key	FK	int(unsigned)	10	Foreign key that uniquely identifies a member model. This field can be used to join to v_dim_model, v_dim_device_model, or v_dim_interface_model for additional member model information.

v_dim_interface_model

This view enumerates all interfaces captured in the reporting database.

Field	Key	Type	Length	Description
model_key	PK	int(unsigned)	10	Internal ID/Key that uniquely identifies a record in this view
model_h_dec		int(unsigned)	10	Model handle (decimal form)
model_h_hex		varchar	24	Model handle (hexadecimal form)
landscape_h_dec	FK	int(unsigned)	10	Landscape handle (decimal form) associated with this model; this field can be used to join to v_dim_landscape for additional landscape information.
landscape_h_hex		varchar	24	Landscape handle (hexadecimal form)
model_name		varchar	4000	Interface Name
create_time		datetime		Creation Time
security_string		varchar	255	Security String
destroy_time		datetime		Destroy Time
port_type		varchar	255	Port Type
port_desc		varchar	255	Port Description (raw value)
port_desc_text		longtext		Port Description (transformed value)
if_speed		Bigint (unsigned)	20	interface speed in Bytes/sec
ip		varchar	255	Network Address
mac		varchar	32	MAC Address
port_link_status		int(unsigned)	10	Port Link Status (raw value)
port_link_status_text		varchar	32	Port Link Status (transformed value)

iflastchange		bigint (unsigned)	20	Last Change
ifinoctets		bigint (unsigned)	20	If In Octets
Datelastsignificant traffic		datetime		Date Last Significant Traffic
hours_idle		bigint	21	Hours Idle
days_idle		bigint	21	Days Idle
ifalias		varchar	4000	If Alias
component_oid		varchar	255	Component OID
device_model_key	FK	int(unsigned)	10	Foreign Key that uniquely identifies the parent device for this interface. Use this field to join to v_dim_device_model.model_key for additional, parent device information.
device_model_name		varchar	4000	Parent Device Name
port_status		varchar	32	Port Status
mclass_name		varchar	32	Model Class
mtype_h_dec		int(unsigned)	10	Model Type Handle (decimal form)
mtype_h_hex		varchar	24	Model Type Handle (hexadecimal form)
mtype_name		varchar	128	Model Type
connected_model_h_dec	FK	int(unsigned)	10	Model Handle of Connected Device (decimal form); this will be NULL if no device is connected. Use this field to join to v_dim_device_model.model_h for additional connected device information.
connected_model_h_hex		varchar	24	Model Handle of Connected Device (hexadecimal form); this will be NULL if no device is connected.
is_connected		int	1	1 indicates that there is a connected device, 0 indicates no connected device
duplex_status		varchar	255	Duplex Status

v_dim_landscape

This view enumerates all of the landscapes that have been encountered during processing reporting data.

Field	Key	Type	Length	Description
landscape_h_dec	PK	int(unsigned)	10	Landscape handle (decimal form)
landscape_h_hex		varchar	24	Landscape handle (hexadecimal form)
landscape_name		varchar	255	Landscape(Domain) Name

v_dim_model

This view enumerates all of the models that are encountered in the course of processing reporting data.

Field	Key	Type	Length	Description
model_key	PK	int(unsigned)	10	Internal ID/Key that uniquely identifies a record in this view
model_h_dec		int(unsigned)	10	Model handle (decimal form)
model_h_hex		varchar	24	Model handle (hexadecimal form)
model_name		varchar	4000	Model Name
network_address		varchar	255	Network Address
landscape_h_dec	FK	int(unsigned)	10	Landscape Handle (decimal form); join to v_dim_landscape for additional landscape information.
landscape_h_hex		varchar	24	Landscape Handle (hexadecimal form)
mclass_name		varchar	32	Model Class
mtype_h_dec		int(unsigned)	10	Model Type Handle (decimal form)
mtype_h_hex		varchar	24	Model Type Handle (hexadecimal form)
mtype_name		varchar	128	Model Type Name
security_string		varchar	255	Security String
destroy_time		datetime		Destroy Time (if applicable)

v_dim_ncm_event

This view enumerates all event codes that are associated with Network Configuration Management (NCM).

Field	Key	Type	Length	Description
type_dec	PK	int(unsigned)	10	Event Type (decimal form); this field also uniquely identifies a record in this view.
type_hex		varchar	22	Event Type (hexadecimal form)
title		varchar	255	Event Title

v_dim_spm_test

This view enumerates all the SPM Tests that are created in the course of processing.

Field	Key	Type	Length	Description
test_id	PK	int(unsigned)	11	Internal ID/Key that uniquely identifies a record in this view
test_name		varchar	64	SPM Test Name
model_key	FK	int(unsigned)	10	Internal Key that uniquely identifies the SPM Test Model in v_dim_model.
model_h_dec		int(unsigned)	10	Model Handle of the SPM Test Model (decimal form)
model_h_hex		varchar	24	Model Handle of the SPM Test Model (hexadecimal form)
model_name		varchar	255	Model Name of the SPM Test Model
source_address		varchar	64	Source Address
dest_address		varchar	255	Destination Address
port		mediumint (unsigned)	8	Port
lookup_string		varchar	255	Lookup String
filename		varchar	255	Filename
packet_size		int	10	Packet Size
test_host_position		tinyint (unsigned)	3	Test Host Position
username		varchar	64	Username
proxy		varchar	255	Proxy
tos		int(unsigned)	10	Type of Service
alt_packet_addr		varchar	64	Alternate Packet Address
alt_packet_port		mediumint (unsigned)	8	Alternate Packet Port
landscape_h_dec	FK	int(unsigned)	10	Landscape Handle (decimal form); join to v_dim_landscape for additional landscape information.
landscape_h_hex		varchar	24	Landscape Handle (hexadecimal form)
effective_start		datetime		Effective start time of the test
effective_end		datetime		Effective end time of the test (if applicable)

v_dim_time

This view enumerates a separate record for every day in the calendar.

Field	Key	Type	Length	Description
time_id	PK	int(unsigned)	10	Internal ID/Key that uniquely identifies a record in this view
calendar_date	UNQ	date		Calendar Date
day_name		varchar	9	Day Name (e.g. Wednesday)
day_number_in_week		tinyint (unsigned)	3	Day Number in Week (Sunday=1, Saturday=7)
day_number_in_month		tinyint (unsigned)	3	Day Number in Month
day_number_in_year		smallint (unsigned)	5	Day Number in Year
week_number_in_year		tinyint (unsigned)	3	Week Number in Year
month_name		varchar	9	Month Name (e.g. January)
month_number_in_year		tinyint (unsigned)	3	Month Number in Year (January = 1, December = 12)
year_number		smallint (unsigned)	5	Year Number
weekend_flag		char	1	Weekend Flag (Y, if Saturday or Sunday)
last_day_in_month_flag		char	1	Last Day in Month Flag (Y, if last day of month)

v_fact_alarm_activity

This view enumerates alarm activities (for example, sets, clears, acknowledgements) that are processed in reporting database.

Field	Key	Type	Length	Description
alarm_key		int(unsigned)	10	Internal ID/Key that uniquely identifies a record in this view
activity		int(unsigned)	10	Internal Code used to identify various activities (1=Set, 2=Ack, 3=Assigned By, 33=Assigned To, 4 or 5=Clear, 6=Ticketed)
activity_title		varchar	17	Activity Title (for example, Set, Acknowledged, and so on)
time		datetime		Time at which activity occurred

username_text		varchar	50	Username associated with activity
set_count		int	1	Set Count
ack_count		int	1	Acknowledgment Count
assign_by_count		int	1	Assign By Count
assign_to_count		int	1	Assign To Count
clear_count		int	1	Clear Count
ticketed_count		int	1	Ticketed Count
data		char	255	Additional details

v_fact_alarm_info

This view enumerates a separate record for every alarm that is processed in reporting database.

Field	Key	Type	Length	Description
alarm_key	PK	int(unsigned)	11	Internal ID/Key that uniquely identifies a record in this view
landscape_h_dec	FK	int(unsigned)	10	Landscape Handle (Decimal Form); join to v_dim_landscape for additional landscape information.
landscape_h_hex		varchar	24	Landscape Handle (Hexadecimal Form)
orig_event_key	FK	bigint (unsigned)	20	Originating Event Key; join to v_fact_event.event_key to capture additional event details.
condition_id	FK	int	11	Condition ID; join to v_dim_alarm_condition for additional condition information.
cause_id		int(unsigned)	10	Cause ID
set_time		datetime		Set Time
clear_time		datetime		Clear Time (if applicable)
duration_seconds		bigint	21	Duration in Seconds
duration_label		varchar	24	Duration Label (HH:MM:SS)
clear_user_key	FK	int(unsigned)	10	Uniquely identifies user who cleared this alarm; join to v_dim_alarm_user.alarm_user_key for more information.
alarm_title_id	FK	int(unsigned)	10	Uniquely identifies an alarm title; join to v_dim_alarm_title for more information.
model_key	FK	int(unsigned)	10	Uniquely identifies the model associated with this alarm; join to v_dim_model for more information.
ack_time		datetime		Acknowledgment Time
time_to_ack_seconds		bigint	21	Time to Acknowledge (Seconds)
time_to_ack_duration_ label		varchar	23	Time to Acknowledge (HH:MM:SS)

ack_user_key	FK	int(unsigned)	10	Uniquely identifies the acknowledging user in v_dim_alarm_user; join to v_dim_alarm_user.alarm_user_key for more information.
first_assigned_time		datetime		Time at which alarm was first assigned
time_to_first_assign_ seconds		bigint	21	Difference in time between set time and time to first assignment (Seconds)
time_to_first_assign_ duration_label		varchar	23	Difference in time between set time and time to first assignment (HH:MM:SS)
first_assigned_user_key	FK	int(unsigned)	10	Uniquely identifies the first assigned user in v_dim_alarm_user; join to v_dim_alarm_user.alarm_user_key for more information.
first_assigning_user_ key	FK	int(unsigned)	10	Uniquely identifies the first assigning user in v_dim_alarm_user; join to v_dim_alarm_user.alarm_user_key for more information.
last_assigned_time		datetime		Time at which alarm was last assigned.
time_to_last_assign_ seconds		bigint	21	Difference in time between set time and time to last assignment (Seconds)
time_to_last_assign_ duration_label		varchar	23	Difference in time between set time and time to last assignment (HH:MM:SS)
last_assigned_user_key	FK	int(unsigned)	10	Uniquely identifies the last assigned user in v_dim_alarm_user; join to v_dim_alarm_user.alarm_user_key for more information.
last_assigning_user_key	FK	int(unsigned)	10	Uniquely identifies the last assigning user in v_dim_alarm_user; join to v_dim_alarm_user.alarm_user_key for more information.
set_troubleticket_time		datetime		Troubleticket Time
time_to_trouble_ticket_ seconds		bigint	21	Difference in time between set time and trouble ticket time (Seconds)
time_to_trouble_ticket_ duration_label		varchar	23	Difference in time between set time and trouble ticket time (HH:MM:SS)
set_troubleticket_user_ key	FK	int(unsigned)	10	Uniquely identifies the trouble ticket user in v_dim_alarm_user; join to v_dim_alarm_user.alarm_user_key for more information.
set_troubleticket_id		char	255	Trouble Ticket ID
assignment_duration_ seconds		bigint	21	Difference in time between last assigned time and clear time (Seconds)

assignment_duration_ label		varchar	24	Difference in time between last assigned time and clear time (HH:MM:SS)

v_fact_event

This view enumerates every event record that is processed in the reporting database.

Field	Key	Type	Length	Description
event_key	PK	bigint (unsigned)	20	Internal ID/Key that uniquely identifies a record in this view
landscape_h_dec	FK	int(unsigned)	10	Uniquely identifies a landscape associated with the model on which this event occurred (decimal form); join to v_dim_landscape for additional landscape information.
landscape_h_hex		varchar	24	Uniquely identifies a landscape associated with the model on which this event occurred (hexadecimal form)
model_key	FK	int(unsigned)	10	Uniquely identifies the model associated with this event; join to v_dim_model for more information.
time		datetime		Time at which event occurred.
type_dec	FK	int(unsigned)	10	Event Type (decimal form); join to v_dim_event for more information.
type_hex		varchar	24	Event Type (hexadecimal form)
creator_id	FK	int(unsigned)	10	Uniquely identifies the creator for this event; join to v_dim_creator for more information.
event_msg		text		Fully constituted event message associated with this event.

v_fact_model_outage

This view enumerates all outages that are processed in reporting database.

Field	Key	Type	Length	Description
model_outage_id	PK	bigint (unsigned)	20	Internal ID/Key that uniquely identifies a record in this view
model_key	FK	int(unsigned)	10	Uniquely identifies the model associated with this outage; join to v_dim_model for more information.
landscape_h_dec	FK	int(unsigned)	10	Uniquely identifies a landscape associated with the model on which this event occurred (decimal form)
landscape_h_hex		varchar	24	Uniquely identifies a landscape associated with the model on which this event occurred (hexadecimal form)
start_time		datetime		Start Time of Outage
end_time		datetime		End Time of Outage (if applicable)
duration_seconds		bigint	21	Outage Duration (seconds)
duration_label		varchar	24	Outage Duration (HH:MM:SS)
start_event_key	FK	bigint (unsigned)	20	Uniquely identifies the event that started this outage; join to v_fact_event on event_key for more information.
end_event_key	FK	bigint (unsigned)	20	Uniquely identifies the event that ended this outage; join to v_fact_event on event_key for more information.
notes		char	250	Outage Notes
outage_type		int(unsigned)	10	Outage Type (0=Initial, 1=Unplanned, 2=Planned, 3=Exempt)
outage_desc		varchar	NO	Outage Description

v_fact_spm_basic_test_results

This view enumerates test results for the following Service Performance Manager (SPM) test types: ICMP Ping, UDP, Path Echo, TCP, DNS Lookup, POP3, DHCP, FTP, SMTP, and HTTP (total time only).

Field	Key	Type	Length	Description
test_id	PK	int(unsigned)	11	Combination of Test ID and Timestamp uniquely identifies a record in this view. Timestamp corresponds with time at which result occurred.
timestamp	PK	datetime	11	Combination of Test ID and Timestamp uniquely identifies a record in this view. Timestamp corresponds with time at which result occurred.
time_id	FK	int(unsigned)	11	Uniquely identifies a day in v_dim_time; join to v_dim_time for more information.
hh		tinyint (unsigned)	3	Hours field of "timestamp" field.
mm		tinyint (unsigned)	3	Minutes field of "timestamp" field.
ss		tinyint (unsigned)	3	Seconds field of "timestamp" field.
latency		int(unsigned)	10	Latency in milliseconds
packet_loss		double	53,29	Packet Loss
timeout		tinyint	2	1=timeout occurred, 0=no timeout occurred

v_fact_spm_http_full_test_results

This view enumerates historical results that are associated with Service Performance Manager (SPM) HTTP tests.

Field	Key	Type	Length	Description
test_id	PK	int(unsigned)	11	Combination of Test ID and Timestamp uniquely identifies a record in this view. Timestamp corresponds with time at which result occurred.
timestamp	PK	datetime	11	Combination of Test ID and Timestamp uniquely identifies a record in this view. Timestamp corresponds with time at which result occurred.

time_id	FK	int(unsigned)	11	Uniquely identifies a day in v_dim_time; join to v_dim_time for more information.
hh		tinyint (unsigned)	3	Hours field of "timestamp" field.
mm		tinyint (unsigned)	3	Minutes field of "timestamp" field.
ss		tinyint (unsigned)	3	Seconds field of "timestamp" field.
http_response_time		int(unsigned)	10	Overall HTTP response time
dns_resolution_time		int(unsigned)	10	Portion of HTTP response time for DNS resolution
tcp_connect_time		int(unsigned)	10	Portion of HTTP response time for TCP connection
http_download_time		int(unsigned)	10	Portion of HTTP response time for HTTP Download
timeout		tinyint(2)	2	1=timeout occurred, 0=no timeout occurred

v_fact_spm_jitter_test_results

This view enumerates historical results that are associated with Service Performance Manager (SPM) Jitter tests.

Field	Key	Type	Length	Description
test_id	PK	int(unsigned)	11	Combination of Test ID and Timestamp uniquely identifies a record in this view. Timestamp corresponds with time at which result occurred.
timestamp	PK	datetime	11	Combination of Test ID and Timestamp uniquely identifies a record in this view. Timestamp corresponds with time at which result occurred.
time_id	FK	int(unsigned)	11	Uniquely identifies a day in v_dim_time; join to v_dim_time for more information.
hh		tinyint (unsigned)	4	Hours field of "timestamp" field.
mm		tinyint (unsigned)	4	Minutes field of "timestamp" field.
ss		tinyint (unsigned)	4	Seconds field of "timestamp" field.

response_time		int(unsigned)	10	Latency
src_to_dest_pl		double	53,29	Source to Destination Packet Loss
dest_to_src_pl		double	53,29	Destination to Source Packet Loss
mia		double	53,29	Missing in Action - Packet Loss with Unknown Direction
late_arrival		double	53,29	Late Arrival
busies		double	53,29	Busies
pos_src_to_dest_jitter		int(unsigned)	10	Positive Source to Destination Jitter
neg_src_to_dest_jitter		int(unsigned)	10	Negative Source to Destination Jitter
pos_dest_to_src_jitter		int(unsigned)	10	Positive Destination to Source Jitter
neg_dest_to_src_jitter		int(unsigned)	10	Negative Destination to Source Jitter
timeout		tinyint	2	1=timeout occurred, 0=no timeout occurred

How to Create Additional SRMDBAPI Users

Administration is not required if the 'srmapi' MySQL user is used to interact with the SRMDBAPI. However, if more accounts are required, you can establish them using the MySQL client application.

For example, you can create a user, srmdbapi_user, with a capability to read all view data from the SRMDBAPI.

Follow these steps:

1. On the SRM server, log in to mysql as 'root'.
2. Establish the new username and password combination in the MySQL database instance and access to both the srmdbapi and reporting schemas:

```
mysql>GRANT SELECT, EXECUTE ON srmdbapi.* TO 'srmdbapi_user'@'%' IDENTIFIED BY 'somepassword';
mysql>GRANT SELECT ON reporting.* TO 'srmdbapi_user'@'%';
mysql>FLUSH PRIVILEGES;
```

3. Logout of mysql.
   srmdbapi_user is created.

> **NOTE**
> The previous 'GRANT' statements lets 'srmdbapi_user' connect to the SRM server from the local or any remote server. The 'srmdbapi_user' only have read-only access to the 'srmdbapi' schema which represents the database implementation of the SRMDBAPI.

How to Access Views

The primary way to access the reporting data in the MySQL database is using the MySQL client. For more information, see http://dev.mysql.com.

For important login information, the user ID is 'srmdbapi' and the password is 'srmdbapi'.

Procedure for Windows

Follow these steps:

1. Log in with a password.
2. Access the following directory:
   ```
   C:\win32app\spectrum\mysql\bin
   ```
3. Enter the following command:
   ```
   mysql -usrmapi -psrmapi srmdbapi
   ```
   You are now connected to the MySQL for Windows.

Procedure for Linux MySQL

Follow these steps:

1. Log in with the password, root.
2. Enter 'bash'.
3. Access the following directory:
   ```
   cd/usr/spectrum/mysql/bin
   ```
4. Enter the following command:
   ```
   ./mysql --defaults-file=../my-spectrum.cnf -usrmapi -psrmapi srmdbapi
   ```
   You are now connected to the MySQL for Linux.
5. To show SRMDBAPI view names on both Windows and Linux MySQL client, type the following command at the MySQL prompt:
   ```
   show tables;
   ```
   The SRMDBAPI table/view names are displayed.
6. To display column names for a given SRMDBAPI view on both Windows and Linux MySQL client, type the following command at the MySQL prompt:
   ```
   desc 'xxx';
   ```
   Where 'xxx' is the table name.
   The columns in each table/view are displayed.

Example

Here is an example with the 'v_dim_alarm_condition' table.

You would type the following command at the mysql prompt:

```
mysql> desc v_dim_alarm_condition;
```

MySQL displays the following table:

```
+----------------+------------------+------+-----+---------+-------+
| Field          | Type             | Null | Key | Default | Extra |
+----------------+------------------+------+-----+---------+-------+
| condition_id   | int(10) unsigned | NO   |     | NULL    |       |
| condition_name | varchar(11)      | NO   |     | NULL    |       |
| criticality    | tinyint(2)       | NO   |     | NULL    |       |
+----------------+------------------+------+-----+---------+-------+
Three rows in a set (0.00 seconds)
```

Sample SRMDBAPI Queries

The following sample SQL queries are presented to both demonstrate the potential of the SRMDBAPI and serve as a training aid. These queries merely represent a subset of what is possible from a functional perspective.

NOTE
Each query contains a 'LIMIT X' records clause to ensure that too much data is not initially returned to the MySQL client.

Log in to MySQL Server

You can log in to the MySQL server using the 'srmapi' user before executing the following sample queries:

On Linux:

```
$SPECROOT/mysql/bin/mysql --defaults-file=../my-spectrum.cnf -usrmapi -psrmapi srmdbapi
```

On Windows:

```
$SPECROOT/mysql/bin -usrmapi -psrmapi srmdbapi
```

Get All 'model created' and 'model destroyed' Events on a Specified Day

You can perform a query to obtain all 'model created' and 'model destroyed' events that occurred on a specified day (2009-12-29). The result set contains event time, model name, event title, and the event message.

```
mysql>SELECT e.time,
           m.model_name,
           de.title,
           e.event_msg
FROM v_fact_event e,
           v_dim_model m,
           v_dim_event de
WHERE e.model_key = m.model_key
AND e.type_dec = de.type_dec
AND e.type_dec IN ( 66049,66050 )
AND e.time BETWEEN '2009-12-29 00:00:00' AND '2009-12-29 23:59:59'
LIMIT 10;
```

NOTE
'66049' and '66050' are the decimal values that correspond with 'model created' and 'model destroyed' events.

Get All 'device create' and 'device destroy' Events on a Specified Day

You can perform a query to obtain all 'device create' and 'device destroy' events that occurred on a specified day (in this example, we use 2009-12-29). This query is similar to the previous one; however, v_dim_model are replaced with v_dim_device_model to ensure that only device-related events are returned.

```
mysql>SELECT e.time,
            d.model_name,
            de.title,
            e.event_msg
FROM v_fact_event e,
            v_dim_device_model d,
            v_dim_event de
WHERE e.model_key = d.model_key
AND e.type_dec = de.type_dec
AND e.type_dec IN ( 66049,66050 )
AND e.time BETWEEN '2009-12-29 00:00:00' AND '2009-12-29 23:59:59'
LIMIT 10;
```

Get All 'device create' and 'device destroy' Events on a Specified Day in a Global Collection

You can perform a query to obtain all device create and destroy events that occurred on a specified day (2009-12-29) for devices that are contained in a particular global collection. This query is similar to the previous one; however, the v_dim_global_collection_member view has been added and joined to the v_dim_device_model view. In addition, the query constrains results to the collection name of your choosing.

```
mysql>SELECT e.time,
           d.model_name,
           de.title,
           e.event_msg
FROM v_fact_event e,
           v_dim_device_model d,
           v_dim_event de,
           v_dim_global_collection_member gcm
WHERE e.model_key = d.model_key
AND e.type_dec = de.type_dec
AND d.model_key = gcm.model_key
AND gcm.gc_name = 'Your Collection Name'
AND e.type_dec IN ( 66049,66050 )
AND e.time BETWEEN '2009-12-29 00:00:00' AND '2009-12-29 23:59:59'
LIMIT 10;
```

Get the List of Top 20 Events on a Specific Day

You can perform a query to obtain the list of top 20 (most frequent) events that occurred on a specified day (2009-12-29). This query considers all models (not simply devices).

```
mysql>SELECT de.title,
               COUNT(1) as event_count
FROM v_fact_event e,
           v_dim_event de
WHERE e.type_dec = de.type_dec
AND e.time BETWEEN '2009-12-29 00:00:00' AND '2009-12-29 23:59:59'
GROUP BY de.title
ORDER BY event_count DESC
LIMIT 20;
```

Sample SRMDBAPI Data Extraction to Flatfile

This process illustrates how to execute a query and extract the associated result set to a flatfile. Once data has been extracted to a flat file, it can be reviewed in any text viewing client (for example, vi, notepad). The following sequence generates a text file named 'top_20_events_20091229.out' containing the results from the query that is outlined in the previous section. The .out file is placed in $SPECROOT/mysql/bin by default.

```
mysql>\T top_20_events_20091229.out

mysql>SELECT de.title,
            COUNT(1) as event_count
FROM v_fact_event e,
            v_dim_event de
WHERE e.type_dec = de.type_dec
AND e.time BETWEEN '2009-12-29 00:00:00' AND '2009-12-29 23:59:59'
GROUP BY de.title
ORDER BY event_count DESC
```

```
LIMIT 20;

mysql>\t
```

Create an ODBC Datasource for the SRMDBAPI

This section illustrates the process to set up an SRMDBAPI ODBC datasource through the ODBC Datasource Administrator. Once the SRMDBAPI ODBC datasource is created, the applications such as Microsoft Excel can use ODBC datasource to query the database directly without using the MySQL client.

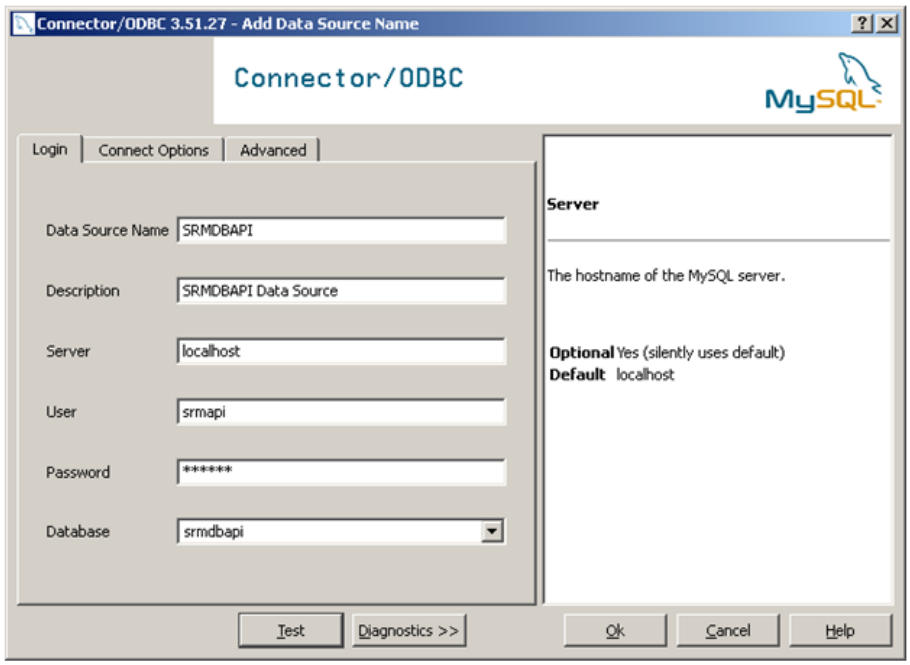
Follow these steps:

1. Navigate to the Windows Control Panel.

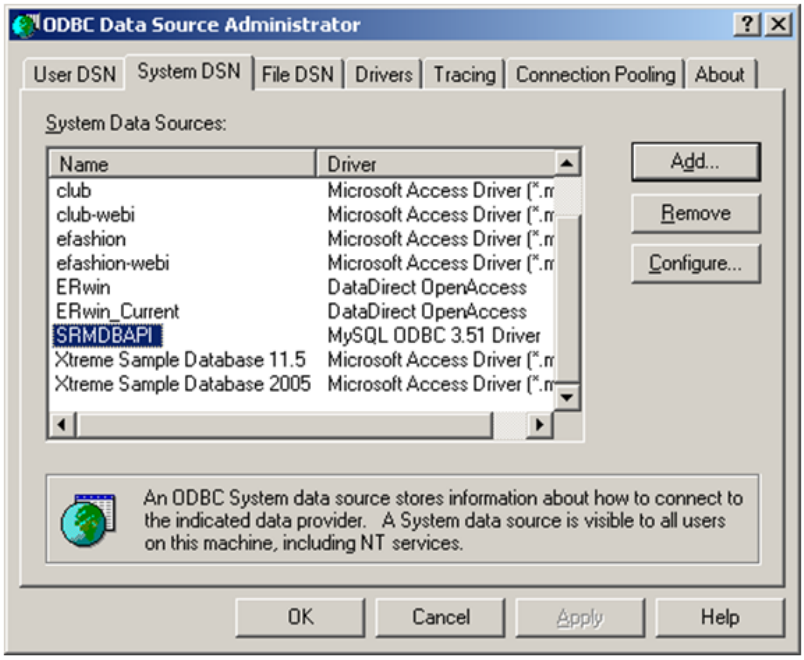
   NOTE
   This path varies depending on the version of Windows that is installed.
2. Double-click 'Administrative Tools'.
3. Double-click 'Datasource (ODBC)'.
4. Click 'System DSN' tab.
5. To add a System Data Source, click the Add button.
6. Select the 'MySQL ODBC 3.51 Driver'.

   WARNING
   If MySQL ODBC 3.51 driver is not available in the picklist, download and install the 'MySQL ODBC 3.51 Driver' directly from MySQL. The driver can be acquired at: http://dev.mysql.com.

1. Configure the new SRMDBAPI Data Source by specifying the following information in the Login tab.



1. To verify the connectivity, click Test.
2. After verifying the connectivity, click OK to create the data source.
3. Verify that the SRMDBAPI Data Source appears on the System DSN tab.



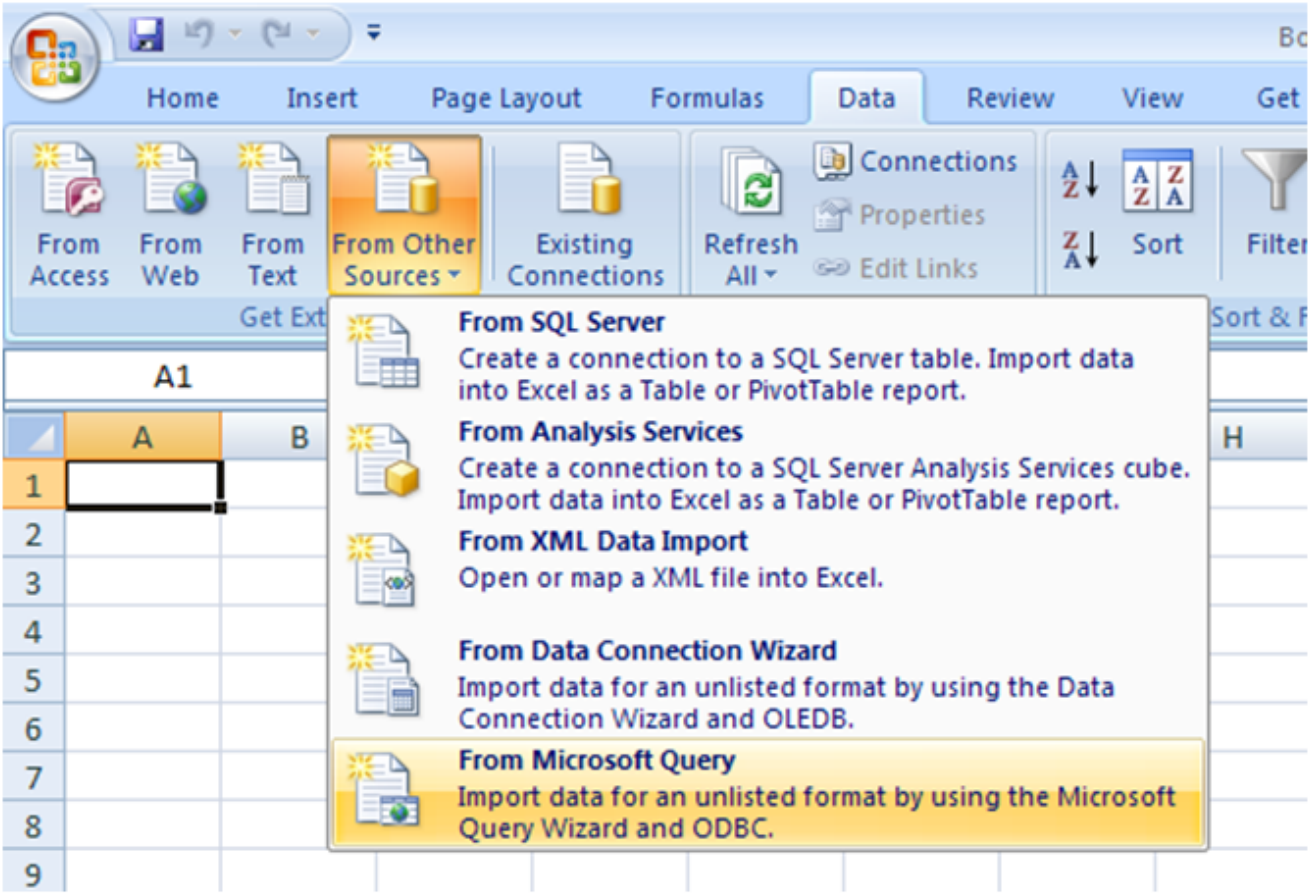
Setup is completed and the Data Source is now available to client applications such as Microsoft Excel.

Create a Sample Query that Uses the ODBC Data Source

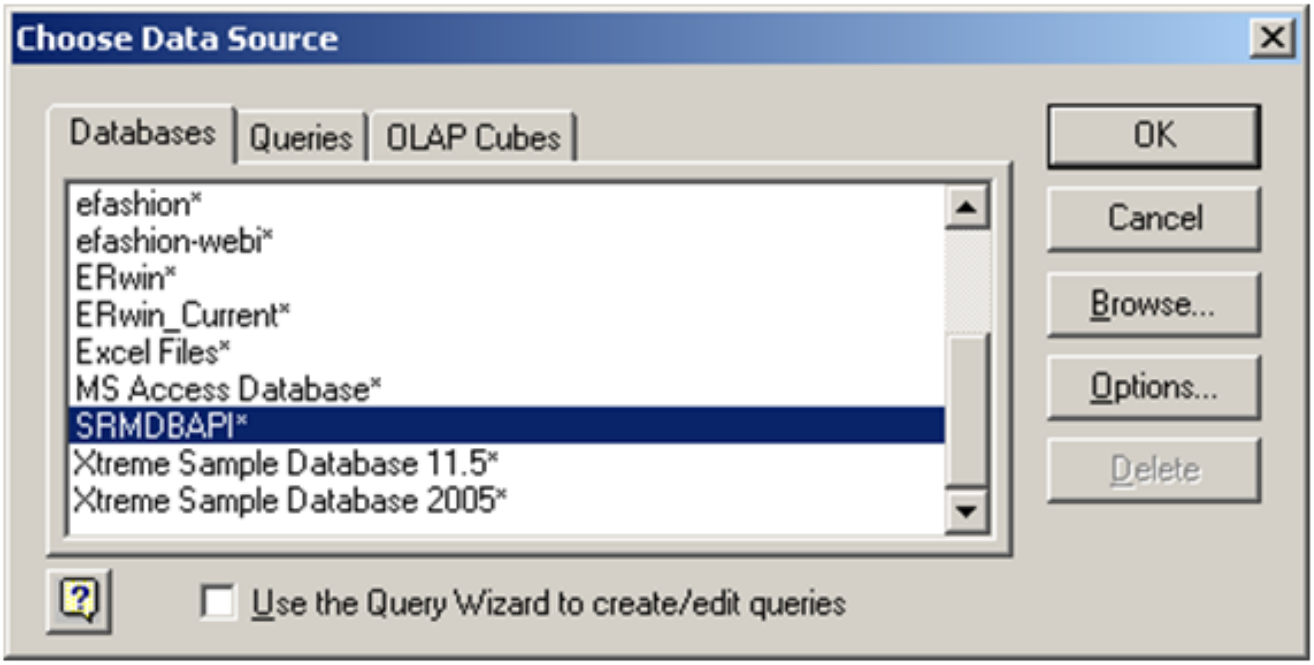
You can use Excel 2007 and the embedded Microsoft Query application to create a sample query. You can generate a sample query against the SRMDBAPI database and then return the associated result set into an Excel spreadsheet. Excel 2007 uses ODBC data source to query the database directly without using the MySQL client.

Follow these steps:

1. Launch Excel 2007.
2. Click the Data tab in the menu structure.
3. Click the From Other Sources icon, and select the From Microsoft Query option from the resulting drop-down list.



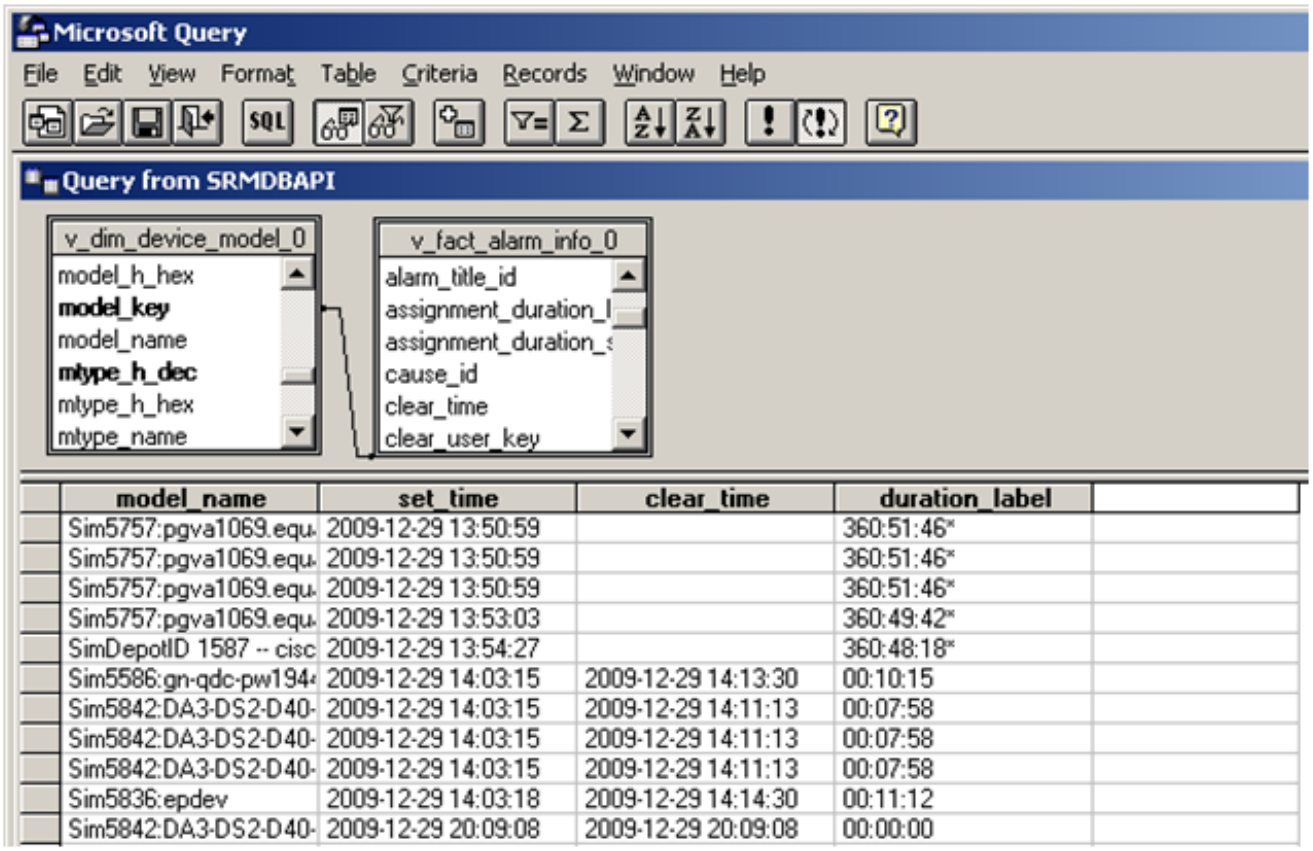
4. From the Databases tab, select the SRMDBAPI* database, clear the 'Use the Query Wizard to create/edit queries' check box, and click OK.



Microsoft Query launches immediately and prompts you for the tables to report on.
For this example, select and add both the v_dim_device_model and v_fact_alarm_info views to make them available for querying.

NOTE
Microsoft automatically joins the two views on the model_key column, which is correct.

5. Double-click the 'model_name' column from the 'v_dim_device_model' view to add this field to the query.
6. Double-click the 'set_time', 'clear_time', and 'duration_label' columns from the 'v_fact_alarm_info' view to add these fields to the query.
   The following image illustrates the Microsoft Query environment:



7. Select Add Criteria from the Criteria option in the menu structure.
   The Add Criteria dialog opens.
8. Select 'v_fact_alarm_info_0.clear_time' for the Field and 'is Null' for the Operator value.
9. Click Add to ensure that the query only displays ongoing alarms.
10. Execute the query by clicking the
    icon  .
11. Return the result set to Excel by selecting the file 'Return Data to Microsoft Office Excel' in the menu bar.
    The Import Data dialog provides options for incorporating the results into the Excel spreadsheet.
    For this example, select Table; however, other options are available, such as Pivot Table Report and Pivot Chart.
12. Select Existing worksheet to include data in the current worksheet.
    The results of your query are captured in an Excel table.

Because results are formatted for Excel, you can take advantage of other capabilities in the application such as charting, pivot tables, and conditional formatting to analyze the data.

SRMDBAPI Potential Issues and Best Practices

The following guidelines helps you to use SRMDBAPI successfully:

- Run an explain plan first before executing your actual query. Determine the approximate number of rows returned.
- When developing queries, you can provide a 'LIMIT X' records clause in your SQL code to ensure that only few records are returned.
- Qualify your queries as much as possible. Most often, you can supply a time frame to restrict result set sizes.
- When qualifying queries, try to restrict to fields that are indexed.
- Join as few tables as possible to satisfy your data requirements; join operations are typically expensive operations.
- Limit sorting unless indexes are available to support such an operation.
- Limit data grouping unless indexes are available to support such an operation.

Appendix E. Report Manager Debugging

This appendix provides information that is used when debugging Spectrum Report Manager.

Debug Options

The following options are available on the Debug Controller page, which you can access from the OneClick home page by clicking the following options: Administration, Debugging, Web Server Debug Page (Runtime).

> **WARNING**
> Only use debugging tools with assistance from CA Support.

For more information, see the OneClick Administration section.

To enable an option, select ON. See the following DX NetOps Spectrum tomcat logs for detailed information:

- For Windows: <*$SPECROOT*>/tomcat/logs/stdout.log
- For Linux: <*$SPECROOT*>/tomcat/logs/catalina.out

The following debug options are available for Spectrum Report Manager:

- **SRM - CABI - Content Installer**
  Logs in BIAR file import operations to the CABI server. If you encounter any issues while updating the Spectrum Report Manager content, then enable this option on the CABI server.
- **SRM - Core - Asset Manager**
  Helps in troubleshooting Spectrum Report Manager device information event processing. Enable this option when new devices are not found in the reporting database or if the deleted device information is not reflected in the reporting database.
- **SRM - Core - Control**
  Helps in debugging the current state of CABI Integration. This option is useful when troubleshooting CABI user creation and CABI user association with OneClick users.
- **SRM - Core - Entity Group**
  Helps in troubleshooting Spectrum Report Manager entity group event processing. Enable this option to debug the device grouping into predefined groups (such as vendor, model class, or landscape) or user-defined groups (such as global collections).
- **SRM - Core - Entity Manager**
  Logs the internal processing events for the model management within Spectrum Report Manager. This option is useful in understanding how Spectrum Report Manager is interpreting new models.
- **SRM - Core - Model Manager**

Helps in troubleshooting Spectrum Report Manager model creation in the reporting database. Enable this option if a problem arises with model key generation in the reporting database.

- **SRM - Core - Report Manager**
  Controls the Spectrum Report Manager core logger, which can be used to debug Spectrum Report Manager initialization, landscape monitoring for a device, and configuration of events and event filters.
- **SRM - Core - Scheduling**
  Logs scheduling information for archiving tasks, such as events archiving and DX NetOps Spectrum Service Performance Manager test data archiving.
- **SRM - Core - User Security**
  Assists in troubleshooting problems when logging in to BI Launch Pad from the DX NetOps Spectrum web console.
- **SRM - DB - Data Access**
  Logs Spectrum Report Manager custom user data like Event titles (eventtitle.xml), PCauseTitles (pcausetitle.xml), and Custom vendors (vendor.xml); and tracks new device model insertion into the reporting database.
- **SRM - DB - Queries**
  Assists in troubleshooting CABI general integrations, such as granting a report access and changing reporting database password and universe password. Logs most of the interaction between the CABI CMS database (for the user and group information) and the Spectrum Report Manager registry table.
- **SRM - DB - SPM Test Query**
  Assists in troubleshooting Service Performance Manager test data activities. Use this option with the 'SRM - Handler - SPM Event' option for a complete Service Performance Manager test data processing log.
- **SRM - Handler - Alarm**
  Assists in troubleshooting alarms processing within Spectrum Report Manager. For example, logs the presence of unprocessed alarm table files in the reporting database.
- **SRM - Handler - Availability**
  Assists in troubleshooting problems with the Spectrum Report Manager availability handler, which processes the availability or outage events for models. Enable this option if there are any issues with model outages.
- **SRM - Handler - Device Availability**
  Assists in troubleshooting device availability events that are processed within Spectrum Report Manager. This legacy option is useful in debugging availability data migration issues (from a previous version of DX NetOps Spectrum to Release 9.4).
- **SRM - Handler - Generic Event**
  Debugs the event processing issues that do not fall under any other handlers.
- **SRM - Handler - Interface Availability**
  Assists in troubleshooting interface availability events processing within Spectrum Report Manager. This legacy option is useful in debugging availability data migration issues (from a previous version of DX NetOps Spectrum to Release 9.4).
- **SRM - Handler - Model Create Destroy**
  Assists in troubleshooting model and global collection management events, such as create, destroy, or rename global collection.
- **SRM - Handler - Model State**
  Assists in troubleshooting the event processing for VPLS reports.
- **SRM - Handler - NCM Config**
  Assists in troubleshooting Network Configuration Manager event processing issues.
- **SRM - Handler - SPM Event**
  Assists in troubleshooting Service Performance Manager test event processing issues. Use this option with the 'SRM - DB - SPM Test Query' for a complete Service Performance Manager test data processing log.
- **SRM - Handler - Security**
  Assists in troubleshooting issues that are related to model access in reports and CABI DX NetOps Spectrum users. Use this option with the 'SRM - Core - Control' option for better debugging.
- **SRM - Spectrum Poller - Device**

Assists in troubleshooting Spectrum Report Manager device polling issues.

- **SRM - Spectrum Poller - Event**
  Assists in troubleshooting Spectrum Report Manager event polling. Enable this option when the reporting database is not in sync with the Archive Manager.
- **SRM - Tools - Archiver**
  Assists in troubleshooting reporting data archiving issues.
- **SRM - Tools - Monitor**
  Logs events that are related to the SRMApplication model. If 'Monitor SRM using a Spectrum model' is disabled, this option routes SRMApplication model events to the debug log.

  NOTE
  The 'Monitor SRM using a Spectrum model' option is on the Spectrum Report Manager Preferences page. You can access the page from the OneClick home page by navigating to Administration, Report Manager, and Preferences. For more information, see Preferences.

Debugging Report Parameter Pages

Enable the debug log in the CABI server for the issues that are related to report parameter pages.

Follow these steps:

1. On the CABI server, open the following file for editing:
   `<CABI_tomcat>/webapps/SpectrumCustomParams/WEB-INF/classes/log4j.properties`
   - ***CABI_tomcat***
     Indicates the location of the tomcat root folder on the CABI server.
2. Change the log4j.logger.com.ca.spectrum.repmgr parameter from WARN to DEBUG.
3. Save and close the file.
4. Restart the CABI tomcat server.
   Logs for Spectrum Report Manager report parameter pages are now written to the following file on the CABI server:
   `/tomcat/logs/SpectrumCustomParams.log`

Deployment Capacity and Optimization Best Practices

Capacity and Optimization for DX NetOps Spectrum

Introducing Capacity and Optimization for DX NetOps Spectrum

DX NetOps Spectrum is one of the most powerful tools available to monitor your network infrastructure. As always, with power comes responsibility. Like a Formula One race car, DX NetOps Spectrum is a highly capable system. However, both the race car and the software are highly susceptible to their operating environment and to the control of their human drivers. Just as a rain-soaked track can end a race, a lack of system resources or an over-subscribed system workload can sideline DX NetOps Spectrum.

In this section, we have captured the best practices and advice to help you keep DX NetOps Spectrum performing at its highest levels. These best practices are derived from systematic, controlled performance testing, and from years of real-world experience supporting our global customer base.

In highly dynamic IT environments, monitoring the capacity of your systems and optimizing your DX NetOps Spectrum deployment cannot be a one-time task. Regular, periodic reviews are required to keep DX NetOps Spectrum operating optimally in larger environments (more than 1,000 monitored devices).

Operating Environment and Systems Setup

Environment Areas of Focus

As with any memory-, CPU-, or disk-intensive, sophisticated application, DX NetOps Spectrum is susceptible to its operating environment. This environment breaks down broadly into the following categories:

Hardware

The minimum hardware requirements for the major DX NetOps Spectrum components are described in the respective installation sections. For large deployments with relatively low-cost system hardware resources, you can minimize the potential of exhausting physical resource capacity. We recommend the following hardware for a heavily loaded DX NetOps Spectrum Component:

SpectroSERVER or OneClick Server

- Server-class, Intel-compatible or Sun hardware
- 4 CPUs or cores running at 2 GHz or higher processor speeds
- 16 GB (**Minimum 8 GB**), encompassing the following requirements:
  - 4 GB for major DX NetOps Spectrum processes
  - 2 GB for the OS
  - 2 GB for smaller DX NetOps Spectrum processes and other transient requirements
- Locally attached disk subsystem
- Hardware RAID (levels 1/0, 5, or 1 recommended, in that order)
- 50 GB of available disk space
- Physical disk using SAS or SCSI technology. We recommend 10,000 RPM minimum.
  For Fiber Channel, SAN, or other disk technologies, the performance characteristics of the locally attached disk subsystem recommendations must be met.

Spectrum Report Manager Server

The minimum hardware requirements for Spectrum Report Manager and SpectroSERVER or OneClick are the same, except for the available disk space. Given the historical data storage nature of Spectrum Report Manager, disk space is a major consideration. For more information, see Spectrum Report Manager.

Virtualization

In a virtual environment, you can share resources to maximize your return on hardware investment. However, DX NetOps Spectrum reacts to network conditions in real-time. As a result, the product requires CPU resources, memory resources, and disk speeds to be running at optimal capacity. If anyone of these resources is affected because of another virtual machine, DX NetOps Spectrum performance is affected. Therefore, we recommend running DX NetOps Spectrum with CPU and memory resources that are dedicated 100 percent of the time.

> **IMPORTANT**
> If you are using a storage area network, use the performance requirements that we recommend in the Hardware section.

> **NOTE**
> Do not run VMware VMotion on virtual machines (VMs) that are running DX NetOps Spectrum because DX NetOps Spectrum always requires dedicated resources. However, if you stop all DX NetOps Spectrum processes on VMs, you can run VMware VMotion on those VMs.

As a best practice for virtual machines that are used for DX NetOps Spectrum, dedicate system resources as specified in the Hardware section.

For **VMware environments**, consider the following points:

- Ensure that the required resources are always available (reserved). Assign dedicated resource groups to the DX NetOps Spectrum virtual machines regardless of the state of other VMs that are running on the same server. Make resource group allocations according to the sizing information from the DX NetOps Spectrum Sizing Tool and the recommendations in this document.
- Create specific RAID volumes or LUN with dedicated disks/spindles to the DX NetOps Spectrum system. These volumes help to avoid disk I/O contention with other applications that share the same RAID or storage array. Allocate large volumes of disks/spindles to the RAID volume or LUN. Larger volumes enable a greater I/O distribution and maximum read/write times for the DX NetOps Spectrum processes.
- The VMware administrator can reserve memory and CPU resources within Clusters or Resources Pools, rather than making reservations on individual VMs.
- On the SpectroSERVER hosts, lower CPU percent ready times were observed with two vCPUs than with four vCPUs. More vCPUs are not always better.
- Disk access times of more than 10 (ms) noticeably affected performance on a SpectroSERVER with a significant load.
- Do not allow the VM snapshots to continue for long periods of time. The delta implementation of snapshots can affect the SpectroSERVER performance.
- VM HBA assignments can also affect SpectroSERVER performance.

Basic Configuration for VMware

Set the following values from the Hardware tab on the Edit Settings dialog:

- 2 vCPUs
- Configured memory: 8 GB (The guest operating system assumes this value as the available physical memory)

Additional Configuration for VMware

If a VM contention affects the ESX server, set the following values from the Resources tab on the Edit Settings dialog:

- Memory
  - Shares: High
  - Reservation: 2-4 GB (hardware-dependent; use with care)
  - Reserve all guest memory (All locked)
    Select the checkbox, to **dedicate** memory (RAM) to your VM.
  - Limit: Unlimited
- CPU
  - Shares: High
  - Reservation: 0 ((hardware-dependent; use with care)
  - Limit: Unlimited

NOTE

When a VM is migrated (vMotion) in VMware, then the spectrumgtw probe picks the changes and sends to DX NetOps Spectrum. The changes are shown in the DX NetOps Spectrum hierarchy. The reflection time for the changes in DX NetOps Spectrum hierarchy is based on the frequency at which UIM vmware probe publishes the changes to CA UIM. This functionality is supported for both legacy integration and integration through spectrumgtw probe.

Operating System

We recommend you to run DX NetOps Spectrum on 64-bit variants supported operating systems to take advantage of 16GB (Minimum 8 GB) RAM.

Isolation

The ideal environment for the DX NetOps Spectrum servers is one in which they are not contending for resources with other processes. Moreover, avoid or thoroughly test some environments as they impact affect the performance of DX NetOps Spectrum. Such environments include Antivirus, Security Software, and Backup utilities.

Also check for Automatic disk backups, operating-system backups, automatic antivirus scans, and network security scans, which affect the performance of DX NetOps Spectrum. These processes place temporary locks on files to which DX NetOps Spectrum needs constant access. Therefore, they can lead to crashes and performance issues. This guidance applies to any hardware or virtual machine where a SpectroSERVER, Archive Manager, or SRM database is deployed.

Perform automatic disk backups, operating-system backups, and operating-system updates only after gracefully shutting down all DX NetOps Spectrum processes.

Configure antivirus scans to skip the DX NetOps Spectrum root directory. If a scan of the DX NetOps Spectrum the root directory is required, run it with all DX NetOps Spectrum processes gracefully stopped.

Additional Considerations

The following sections list in detail some of the additional factors which need to be considered for DX NetOps Spectrum Deployment Capacity and Optimization best practices:

- Distributed Deployment Considerations
- Workload Where to Focus Optimization Efforts
- – Total Number of Models
- – Searches
- – Network Device Interfaces
- – Discovery
- Managed Network Health
- Sharing Monitoring Workload among SpectroSERVERs

Distributed Deployment Considerations

If you plan to deploy DX NetOps Spectrum in a large environment, start by understanding how DX NetOps Spectrum scales to manage larger environments. Scaling is achieved primarily by adding SpectroSERVER instances, spreading the workload by splitting the monitored infrastructure among these servers.

Network bandwidth and reliability can come into play with larger distributed deployments. In general, we advise you to place the SpectroSERVERs close (from a network-efficiency perspective) to the largest number of monitored devices. Place OneClick servers close to the most of the user community that will actively run the OneClick clients.

Substantial communication takes place between the OneClick servers and the SpectroSERVERs. Therefore, the available bandwidth and resiliency of the networks between them must be monitored closely. SpectroSERVERs also communicate with each other, but less frequently and less intensively. However, those network pathways must also be monitored closely to ensure resiliency.

Workload Where to Focus Optimization Efforts

Workload is an important area of focus for your optimization efforts. The following list provides examples of DX NetOps Spectrum deployment variables that affect workload and are candidates for an optimization:

- The total number of models, numbers of models being polled and polling rates
- Searches (Global Collection search criteria, user-initiated searches, and condition correlations)
- Alarms (total active and rates - an impact on OneClick concurrent users)
- Events (logging and traps)
- Network device interfaces
- Auto-discoveries and trap-based discovery
- Managed network health
- Spectrum Report Manager

As you can see, this list is already fairly long, yet it highlights only the major aspects of workload. Many other dimensions of workload can also affect a DX NetOps Spectrum deployment, although typically to a lesser extent. For instance, the DX NetOps Spectrum documentation set covers more than 40 additional major features, integrations, and customization or integration toolkits. Many variables are involved. We cannot be exhaustively prescriptive about exactly how much of every capability, in every possible combination, will perform adequately in a given deployment.

Without being fully prescriptive, we can provide best-practice recommendations for keeping workloads within relatively safe tolerances. This type of workload balancing maintains the entire DX NetOps Spectrum environment at high performance levels.

The topics in this section provide guidance on best practices when confronted with a requirement to split workload across existing or new SpectroSERVERs.

Total Number of Models

The DX NetOps Spectrum Sizing tool can help you determine the total number of advisable models. Contact CA Support or your sales representative to request a sizing analysis. This analysis also helps you determine optimal polling rates.

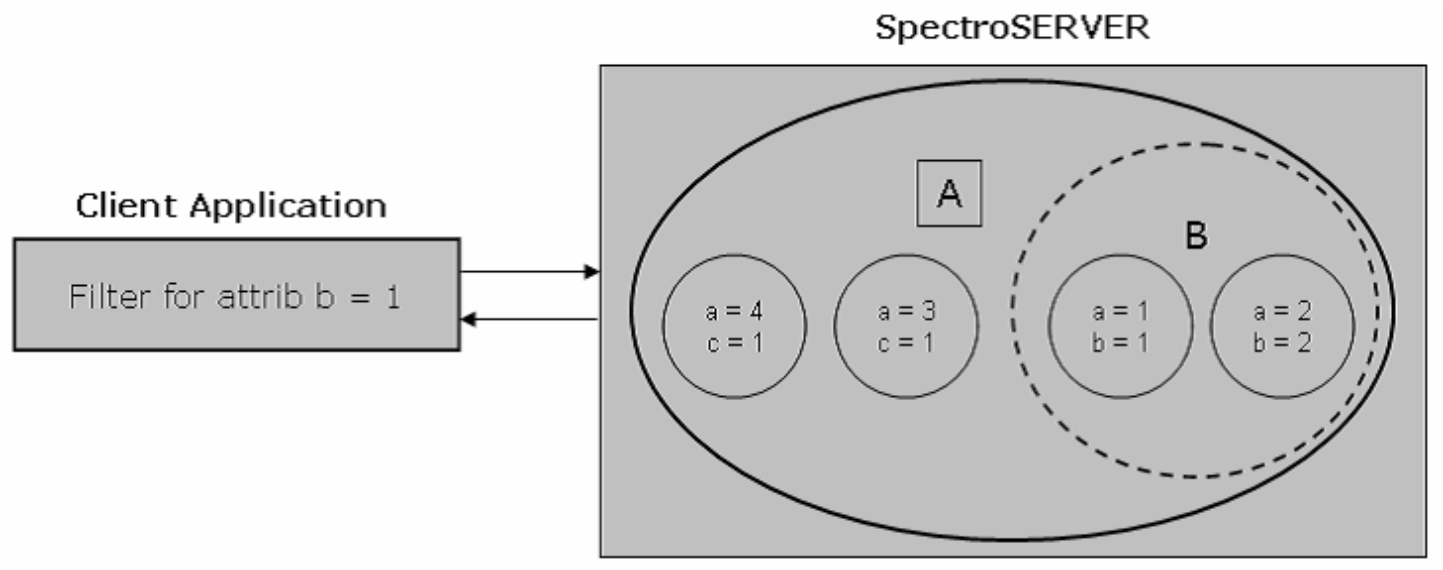
Searches

DX NetOps Spectrum offers powerful search capabilities. Searching for models can have an outsize impact on system performance. Such operations can consume significant system resources and can generate substantial I/O. The required resources depend on how searches are constructed and how often they occur.

Global Collection searches, custom OneClick searches, SANM, Southbound Gateway, and versions of Policy Manager earlier than 9.2.1 are some of the DX NetOps Spectrum features that rely on search operations. Before you use a search, verify whether that search is essential. For example, Global Collections can be constructed manually; as a result, searches are not required if a collection is small. In that case, a better option can be to use the Favorites feature of OneClick rather than a global collection.

When constructing a new search, spend some time to determine whether any internal attributes exist exclusively on the desired models. If two or more attributes are needed in the search, which ones must be used first?

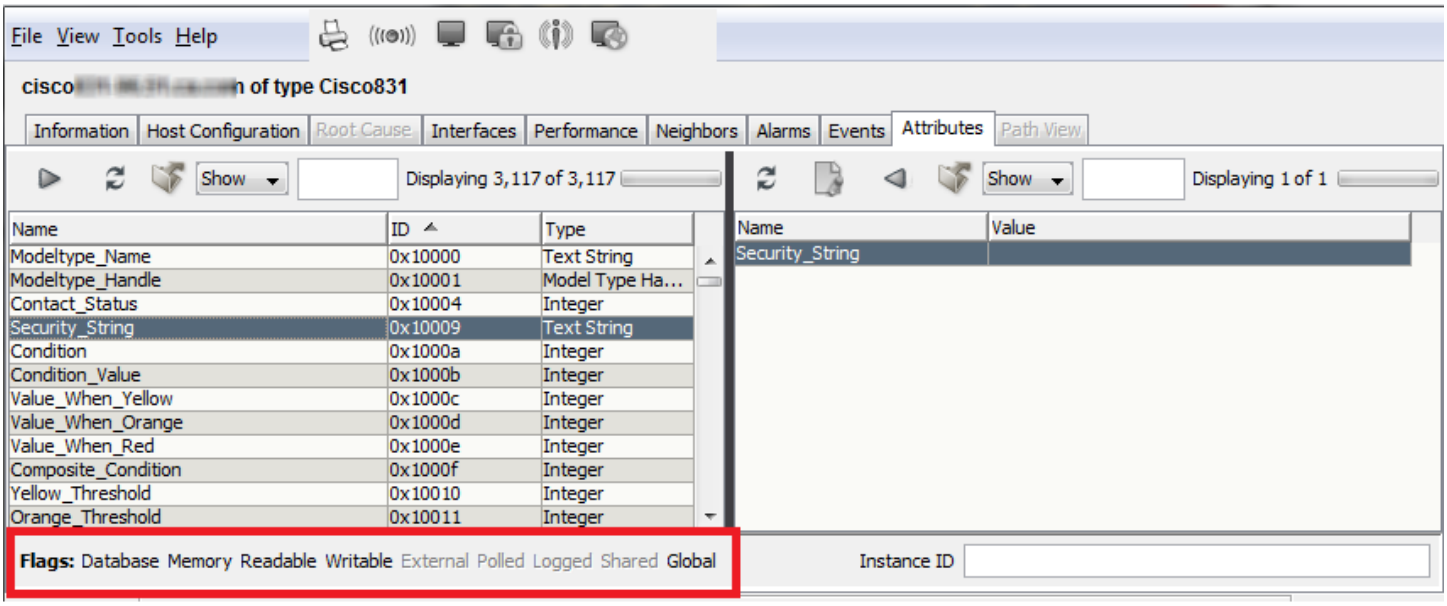
The following diagram describes how attributes group models. All models with Attribute "A" are depicted by the large oval. All models with Attribute "B" are shown as a circle. To limit the number of models that are searched, select the attribute “B” over “A” as a first criterion. As a result, the search is optimized.



After narrowing search expression results based on attributes, consider attribute flags and data types.

The attribute flags provide an indication of the time that is required to retrieve or access the information from a model. They are therefore an important aspect of optimization. A memory attribute has the fastest access times, followed by database attributes, calculated attributes, and, finally, external attributes. The calculated attributes are typically attributes whose values are determined on demand. And the slowest type of attribute for purposes of searching is the external type. Extra time is required to contact multiple devices over the network and wait for them to respond.

Attribute flags and data types can be viewed from the Attributes tab of the Component Detail panel in OneClick:



The following list places attribute storage flags in order from least CPU-intensive with the quickest access to most CPU-intensive and slowest access:

Memory Flag

1. Database Flag
2. Calculated

3. External Flag

Data Types for comparison also require consideration. The following list places attribute data types in order from quickest comparison to slowest comparison:

Integer, Counter, Enumeration, Model Type Handle

1. IP Address, Octet String
2. Text String

The following lists place overall attribute flags and data types for complex searches of AND/OR in order from the most efficient to the least efficient:

Memory Flag

Integer, Counter, Enumeration, Model Type Handle

1. IP Address, Octet String
2. Text String

Database Flag

Integer , Counter, Enumeration, Model Type Handle

1. IP Address, Octet String
2. Text String

Calculated Attribute

Integer, Counter, Enumeration, Model Type Handle

1. IP Address, Octet String
2. Text String

External Flag

Integer, Counter, Enumeration, Model Type Handle

1. IP Address, Octet String
2. Text String

Example Search

A Global Collection search contains, in no particular order, ifDesc, Topology Model Name String, Network Address, and Model Type Handle. From top to bottom, the search should include the following items for best performance:

Model Type Handle - Memory: Handle

1. Network Address - Memory / Database: IP Address
2. Topology Model Name string - String: Calculated
3. ifDesc - String: External

Users typically deploy the search feature in a Global Collection. After you create a Global Collection, the SpectroSERVER immediately performs the search. If a SpectroSERVER determines that the search calls for excessive processing for a long time, a minor alarm (0x10f21) appears on the GlobalCollection model. When you see this alarm, review and optimize the search expression according to the information that we provided here.

Network Device Interfaces

Device interfaces are involved in two aspects of performance considerations.

ifTable Entries

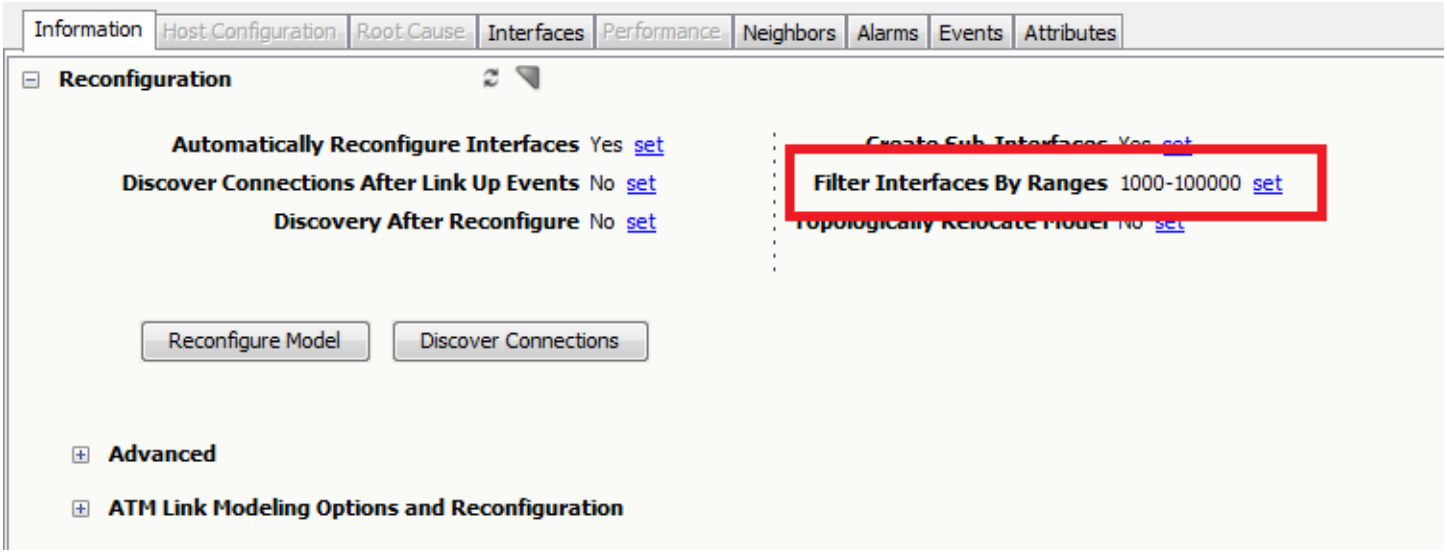
By default, the SpectroSERVER process models all interfaces in the MIB 2 ifTable, all Frame relay DLCI interfaces from RFC1315 and RFC2115, and all ATM VCL interfaces from RFC2515.

In the ifTable, it is common to find virtual technologies including VoIP calls and virtual routing interfaces. As a result, you must review the number of entries in the ifTable when you model a device for the first time. Large quantities of entries (in the thousands or tens of thousands) can have an impact on SpectroSERVER performance.

Reconfiguration Operations

Typically, the more interfaces on a device, the longer the interface reconfiguration operation takes. When devices have 1000 or more physical or logical interfaces, reconfiguration operations can take minutes to complete. This process adds a minimum of 6 - 8 percent CPU time and increases traffic levels on the network.

When some interfaces are not required for effective monitoring, filtering capabilities are available to speed up the process. An example is shown in the following image from the OneClick Component Details Information tab:



Other methods of filtering interfaces are more complex. Consult with CA Support before proceeding with complex interface filtering. Or, try one of the following options:

- Selectively disable the automatic reconfiguration if the interfaces are fairly stable, with few changes.
- Model the device as a Pingable model. You thus gain the device status information and do not gather status data from the individual interfaces.

The frequency of interface reconfigurations for a device can affect the performance. At every poll interval, the device model reads both the ifTableLastChange and ifStackLastChange MIB objects. The ifTableLastChange is a timestamp that indicates when the interface table added or removed interfaces. The ifStackLastChange is a timestamp that indicates when the order of interface stacking last changed. If neither attribute is supported, ifNumber (the ifTable interface count) is monitored for changes. If the value of any of these MIB objects change, automatic reconfiguration starts by default.

Problems can occur when DX NetOps Spectrum monitors a VoIP router. Such routers from some vendors constantly change the interface stacking order as VoIP calls are made and then closed. At every poll cycle, DX NetOps Spectrum sees a change in the stack timestamp and reconfigures. In this case, DX NetOps Spectrum incurs an unnecessary workload with the extra overhead of the reconfiguration and interface table reads. This situation can also have implications for configured product integrations, such as CA eHealth. Updates to the CA eHealth side can be triggered, based on the reconfigurations occurring in DX NetOps Spectrum.

The Spectrum Event Rate Window rule identifies reconfiguration issues and raises attention to DX NetOps Spectrum operators. The rule, which is in place by default, states that if a device experiences 6 interface reconfiguration

events 0x1001d in 31 minutes, an alarm of type 0x10050 is generated on the device. Review the event log for the affected device and filter for Event Type 0x1001d. Each event of that type identifies the cause of the interface reconfiguration.

Cause Indicator	Description
Interface-Stack-Change-Reconfiguration	The reconfiguration was caused by a change in the stacking order of the interfaces. MIB object ifStackLastChange has changed in value. If this object changes every polling cycle, then we recommend that you turn off Use_If_Stack_Last_Change. Attribute Use_If_Stack_Last_Change 0x000130bc can be set to false to disable this trigger
Interface-Table-Change-Reconfiguration	The reconfiguration was caused by the creation or deletion of an interface in the table. MIB object ifTableLastChange has changed in value. If this object changes every poll cycle, we recommend that you turn off Use_If_Table_Last_Change. Attribute Use_If_Table_Last_Change 0x00011f7f can be set to false to disable this trigger.
Interface-Count-Change-Reconfiguration	The reconfiguration was caused by a change in the number of interfaces in the table. MIB object ifNumber has changed in value. If this object changes every poll cycle, we recommend that interface reconfiguration be turned off. Attribute If_IsAutoCnfgActive 0x00011dd4 can be set to false to disable automatic interface reconfiguration.
SNMP-Contact-To-Device-Re-established	Review the event log for the device to see whether the device model has lost and regained contact. In such a situation, you may have a network problem.

Discovery

Questions often arise from DX NetOps Spectrum customers about Discovery and associated performance issues. The most important consideration for Discovery is knowledge of the network that you are discovering. How many devices does it contain? What types of devices compose the network, such as switches, routers, and hosts? And what are the port densities of these network devices?

The answers to these questions often have an impact on Discovery. They also affect the decision whether the entire network can be modeled on one SpectroSERVER, or whether multiple SpectroSERVER are required.

> **WARNING**
> Do not discover or model the entire network at once, if you have not performed these activities before.

The best practice for first-time discoveries is to run a scan of the devices in a particular address range to understand what devices are present. We recommend starting with a Class B address range, or smaller. Then model sections of the network one at a time, paying attention to the memory consumption of the SpectroSERVER process. Use a tool such as the DX NetOps Spectrum Performance View to monitor memory usage. Keep in mind the recommendations for memory capacity that we discussed in Alarms and other topics.

Items Discovered	Size of Batch to Discover
Switches / Routers with 100 ports each	250 Devices
GnSNMPDev with 40 ports	500 Devices
Hosts with 2 ports	250 Devices
Pingables	1000 Devices

Technologies such as VLAN, MPLS, VPN, VRRP/HSRP, Multicast, and Virtual Host Management are other optional Discovery types. For each technology, the numbers that are provided in the following table are a guide for strategic planning for Discovery before initiating it. The following table summarizes the recommended totals:

Technology	Large Deployments	Predeployment Recommendations
MPLS VPN	1000 or more VPNs, 10000 or more sites	Use the hub and spoke VRF testing back to the corporate network. Do not use full mesh.
MPLS VPLS	1000 or more VFIs, 10000 or sites	Use the hub and spoke testing back to the corporate network. Do not use full mesh.
MPLS TE	1000 or more LSPs	Separate logically by location or use some other partitioning method on different SpectroSERVERs to distribute the load.
Multicast	1000 or more groups	Separate logically by location or use some other partitioning method on different SpectroSERVERs to distribute the load.
Enterprise VPN	1000 or more sites	Use the hub and spoke IP SLA testing back to corporate. Do not use full mesh.
QoS	Policies, map classes, behaviors, with 1000 or more interfaces	Set poll interval to 300 seconds or longer.
VRRP/HSRP	20 or more groups	Do not use active polling, use the trap-based passive notification.
VLANs	50 VLANs or more with VLAN overlay enabled	Separate logically by location or use some other partitioning method on different SpectroSERVERs to distribute the load.
Virtual Hosts	1000 or more VHM	Separate logically by a data center or by some other partitioning method. For more information, see the Virtual Host Management section.

Operators sometimes enable trap-based discovery as a means of discovering new devices as they come online, which is how the feature was designed. However, if using this feature (which is disabled by default), it is important to know the frequency of new device additions. Failure to understand this metric can cause a SpectroSERVER crash due to memory exhaustion. When using this feature, review the rate of unmanaged traps per day to understand how many new devices are likely to be modeled.

When DX NetOps Spectrum processes a trap for the first time but the IP address is not modeled, an unmanaged trap event is generated on the VNM model. Use these traps to calculate the approximate rate of device additions per day.

NOTE

Alarms with respective severity are raised at different (model-count) levels, to notify you about an increasing number of models for a SpectroSERVER. The discovery of devices from OneClick discovery console is stopped when a major alarm for the VNM model is raised at 950000 number of models. However, the creation of other models (users, groups, containers, and so on) is allowed.

Managed Network Health

Network health affects business services and also can affect DX NetOps Spectrum, which monitors network health for business services. With an increased focus on the health of business services, we sometimes overlook chronic network health issues that do not appear to impact the business. To monitor business services and resolve faults down to the network layer, any infrastructure management strategy must monitor and react-to constant change. Chronic network health issues can increase the amount of change and the resulting work for DX NetOps Spectrum by an order of magnitude.

Consider the case where a single link on a core router is going down and coming back up several times each minute, a condition known as "flapping". In this case, DX NetOps Spectrum receives a link-down and link-up trap and will also poll the interface for status each time the link flaps. In addition, this flapping link can result in a temporary loss of contact with many of the devices in the network behind it. The result can be more polling and fault-isolation overhead as DX NetOps Spectrum works to determine the state of the devices in this network. However, this single case is not an issue, as far as DX NetOps Spectrum capacity is concerned.

But imagine a second scenario in which several core routers and switches each have multiple "flapping" interfaces. The effect of thousands of link-down/link-up traps, subsequent polling, and fault-isolation overhead in this example could result in a high and continuous DX NetOps Spectrum workload. The increased workload can include tens of thousands of alarms being generated and cleared continually.

Generally and in the earlier two cases, DX NetOps Spectrum provides the data in terms of events and alarms to locate and resolve network health issues. DX NetOps Spectrum operators must pay attention to network health issues and must take steps to resolve them, or tune DX NetOps Spectrum to mitigate their impact. In the earlier examples, resolving the flapping interface problem is the solution. When the connectivity between DX NetOps Spectrum and the managed devices are generally unreliable, or if devices are slow to respond, verify your polling timeout thresholds and retry thresholds. Failure to do so can result in large numbers of "false" alerts due to failed polls, which increase fault isolation overhead.

Finally, connectivity among DX NetOps Spectrum components (SpectroSERVERs and OneClick servers) is an important consideration. Everything from basic server-to-server communications to cross-server searches relies on network connectivity. Therefore, reliable communications among servers are critical for DX NetOps Spectrum performance.

Spectrum Report Manager (SRM)

Most of the advice that we have provided thus far has focused on the major real-time aspects of a DX NetOps Spectrum deployment. Many customers have come to rely also on Spectrum Report Manager for historical data collection, analysis, and reporting. Report Manager includes a separate database that archives data from all connected SpectroSERVERs. Therefore, pay particular attention to the disk capacity and disk I/O performance of the system. Tuning can be required, depending on the amount of data being stored, the opportunities for filtering unnecessary data, and the size of the report.

A best-practice recommendation is to determine the total database size that is required to store event history, and then allocate twice the space on that disk partition to accommodate transient space requirements. The topic titled Spectrum Report Manager Sizing Guidance provides advice and formulas to help you calculate disk space requirements.

The following considerations are also important for Spectrum Report Manager performance and capacity:

- Consider the volume of the data and system resources for the Report Manager performance. Running reports from a smaller volume minimizes report generation failure especially for event and alarm reports. Smaller volume of data decreases the response time of the database query.
- When the result set is large, or when a large amount of data is sorted or grouped, the database writes the results to disk. This activity affects the Report Manager performance.
- If your environment generates a high volume of events without generating event reports, consider purging the event table periodically. Purging this table saves space on the reporting DB system.
- If you generate event reports on a specific set of events, consider purging the event types that you do not require. Or, if selected event types are not required to produce alarm, asset, availability, or other reports, consider filtering these events before they reach the reporting database. For more information, see Install Report Manager.
- If your environment generates a large volume of events on models that you do not include in reports, consider filtering these events from the reporting database. Install Report Manager contains more information.
- Some of the filtering mechanisms on the reports themselves can cause performance issues. We are still researching this possibility, but we have seen anecdotal evidence that alarm and event filters, when used on reports, degrade the performance. Where possible, try to limit their use.
- CA Support maintains some best-practice recommendations for the CA Business Intelligence (CABI) component that Spectrum Report Manager uses for reporting capabilities. These considerations also apply to any CA product that uses CABI. Contact CA Support for more information.

Spectrum Report Manager Sizing Guidance

The following formula can help you estimate the amount of disk space that is likely to be required to support the Reporting database for a user-specified amount of time.

The total number of required disk spaces in GB equals:

```
((# of devices) * (avg # of events per device per day) * (# of days of storage desired) * (avg size of event
 in KB)) / 1048576
```

- **# of devices**
  Environment-specific value. Consider future growth when specifying this value.
- **avg # of events per device per day**
  Represents the total number of events that are (1) generated daily and (2) are associated with the creation of a single device model. This total includes all events that result from the related application, port, and interface models. The easiest way to approximate this number is to look at the total number of events that were generated on one SpectroSERVER in a day. Divide that total by the number of devices that are modeled on that SpectroSERVER.
- **# of days of storage required**
  Environment-specific value.
- **avg size of events, in KB**
  An estimation of the amount of disk space a single event consumes in the Reporting database. This value is measured in KB.
- **1048576**
  The product of the earlier equation is divided by this number to get a measurement in GB.

You probably have an idea of the number of devices and the number of days of storage that you want. Only two variables are then required in the calculation:

- **Average number of events, per device, per day**
  Environment-specific value. You can query the DDMDB to see the average number of events that are generated on a given day.
  If you are a new DX NetOps Spectrum user, or if you are unsure how to determine the average number of events, use a reasonable default value. Consider that 300 events per day, per device, for 500 devices equate to 150,000 events per day. A default value of 300 is a good starting-point.

To get an idea of the average daily number of events that are generated per device, find out how many events are generated daily. The following query returns the total event count for the last ten days:

```
SELECT date(from_unixtime(utime)) as x, count(*) as cnt
    FROM event GROUP BY x
    ORDER BY x DESC LIMIT 10;
```

The following query returns the days and event volume for the busiest ten days:

```
SELECT date(from_unixtime(utime)) as x, count(*) as cnt
    FROM event GROUP BY x
    ORDER BY cnt DESC LIMIT 10;
```

Use the results of these queries to devise a reasonable event count. Once you know the event count, divide that number by the total number of devices that are modeled on that server. The result is the average event count per device, per day.

- **Average size of events in the Reporting database (KB)**
  We recommend 1 KB as an appropriate amount of space to store your average event and the corresponding records. This number can obviously rise if most events are large - containing large amounts of data. The types of events also affect data size. Alarm events turn into multiple Reporting table records. NCM events only affect a single table (event). But for purposes of generalizing the behavior, 1 KB seems to be an appropriate measure.

Sizing Guidance Examples

Here are a couple of examples that illustrate useful calculations of the required storage capacity:

Example A. Your environment contains 600 devices, and you want to retain data for 4 years (1460 days).

> **NOTE**
> You do not know how many events are generated per device, so we default to 300.

The total data in GB that must be stored equals:

```
(600 * 300 * 1460 * 1) / 1,048,576 =
262,800,000 / 1,048,576 =
250 GB
```

- **Example B**. You have 1900 devices across three servers, and you want to retain data for 2 years (730 days). Your deployment seems to be averaging 400 events per device, per day.

  > **NOTE**
  > In this example, we ignore the fact that you have three servers.

  The total data in GB that must be stored equals:

  ```
  (1900 * 400 * 730 * 1) / 1,048,576 =
  554,800,000 / 1,048,576 =
  530 GB
  ```

Sharing Monitoring Workload among SpectroSERVERs

When one or more of your SpectroSERVERs is oversubscribed, balance the workload or split it among additional SpectroSERVERs. Depending on the size, complexity, and scope of DX NetOps Spectrum feature use, and on the number of users, this project can require careful planning and execution. Careful planning can minimize production impacts and can ensure that the results achieve your objectives and have a good life span.

Many users find that they can benefit greatly from the help of CA Services or of our partner network at this stage.

Further discussion of this complex topic is beyond the scope of this document. We plan to leverage the information that we gather from our active user communities to make further updates to this topic in future releases of DX NetOps Spectrum. We recommend searching the message boards for information that is related to workload optimization.

Uninstalling

Uninstall DX NetOps Spectrum on Windows

The uninstallation program removes all of DX NetOps Spectrum from your hard drive. The files that are removed include everything that was originally installed, plus your customizations, if any. The uninstallation program automatically stops all DX NetOps Spectrum processes (for example, the DX NetOps Spectrum Control Panel, the OneClick web server, processd, and the Location Server).

Close any bash shells that you have open before uninstalling. The uninstallation program does not close bash shells, because you could be running bash shells for programs other than DX NetOps Spectrum.

> **WARNING**
> Do not uninstall DX NetOps Spectrum if you plan to perform an upgrade installation. Doing so permanently removes the customizations that you have applied, if any.

Follow these steps:

1. Stop DX NetOps Spectrum.
2. Log in as Administrator or a user with administrator privileges.
3. Go to Start, Control Panel, Programs and Features.
4. Highlight SPECTRUM and select Uninstall/Change.

   > **NOTE**
   > If you highlight SPECTRUM OneClick Console and select Change/Remove, only the Java Web Start application is removed.

   The Uninstallation dialog appears.
5. Select Uninstall.
6. Click OK on the Warning window.
   The uninstallation program continues.
7. When the uninstallation program completes, click OK on the Uninstall Status window.
   DX NetOps Spectrum is uninstalled.

Uninstall DX NetOps Spectrum on Linux

The uninstallation program removes all of DX NetOps Spectrum from your hard drive. The files that are removed include everything that was originally installed, plus your customizations, if any. The uninstallation program automatically stops all DX NetOps Spectrum processes (for example, the DX NetOps Spectrum Control Panel, the OneClick web server, processd, and the Location Server).

> **WARNING**
> Do not uninstall DX NetOps Spectrum if you plan to perform an upgrade installation. Doing so permanently removes the customizations that you have applied, if any.

Follow these steps:

1. Stop DX NetOps Spectrum.
2. Log in as root and navigate to the <*$SPECROOT*>/Install-Tools/Uninstaller directory.
3. Enter ./UninstallSpectrum and then click **Uninstall**.
4. Select **OK** on the Warning window.
   The uninstallation program continues.
5. When the uninstallation program completes, click **OK** on the **Uninstall Status** window.
   DX NetOps Spectrum is uninstalled.

Uninstall DX NetOps Spectrum on Linux via command line

You can uninstall DX NetOps Spectrum using a silent method. This helps when you are unable to run the GUI in Linux.

Follow these steps:

1. Stop DX NetOps Spectrum.
2. Log in as root and navigate to the <$SPECROOT>/Install-Tools/Uninstaller directory.
3. Enter ./UninstallSpectrum <specroot> <tmp dir> < -silent >

DX NetOps Spectrum Dockerization

This page contains the following topics:

> **NOTE**
> Autoinstallation of DX NetOps Spectrum DSS is now available with 10.3.1!

About DX NetOps Spectrum Dockerization

Dockerized DX NetOps Spectrum is advantageous and beneficial for DX NetOps Spectrum users. Dockerized DX NetOps Spectrum components can be deployed separately, such as spectrum-one-click-server-image, spectrum-ss-image, and spectrum-sdc-image. It helps spin multiple containers to set up a distributed DX NetOps Spectrum deployment within no time(minutes*). With dockerization, you can resolve behavioral, staging, and running issues of applications in different environments in various datacenters. DX NetOps Spectrum Dockerization ensures the packaging of all the required configuration files and libraries and other dependencies that are required to run DX NetOps Spectrum in any environment. With DX NetOps Spectrum Dockerization, you can ensure continuous integration that is deployed automatically. The transition time from development to production can be greatly reduced as one container can be used across multiple environments. Docker images can be moved from one server to another with ease. Docker containers are highly scalable as with the demand of the users. Running DX NetOps Spectrum on a Container Application platform provides a seamless service abstraction layer. Any changes to Container properties (like HostName/IP change) would not majorly affect the current deployment.

> **NOTE**
>
> DX NetOps Spectrum can be dockerized on Red Hat® (v.7.4) OpenShift for easy development, deployment, and building of either on-prem or cloud applications.

Recommended Software Requirements

The following are the recommended software requirements for the Docker engine and OpenShift installation.

For Docker Engine

- RHEL - v7.4 (Docker-engine installation and container creation is tested on RHEL 7.4 VM). Install the latest version of Docker using 'yum install docker' command.

For OpenShift

- OpenShift - v3.6
- RHEL - v7.4
- Ansible - v2.5.4
- Git 1.5

Spectrum Pre-built ISO Image Load Process

If you do not wish to build the docker images manually, follow these steps:

1. Download the required tar.gz files from support portal, onto a Linux Rhel 7.4 VM

```
CA-Spectrum-SpectroSERVER-Docker-<current_version>.tar.gz
 CA-Spectrum-OneClickServer-Docker-<current_version>.tar.gz
 CA-Spectrum-SDC-Docker-<current_version>.tar.gz
 CA-Spectrum-OneClickServer-And-SRM-Docker-<current_version>.tar.gz
```

2. Execute the following command to extract tar file:
   ```
   gzip -d <filename>.tar.gz
   ```
3. Once the above unzipped tar file is available, to extract the respective docker image, execute the command:
   ```
   docker load -i CA-Spectrum-SpectroSERVER-Docker-<current_version>.tar
   ```
4. To view the loaded docker images, execute the following command:
   ```
   docker images
   ```

What Next?

Create and Run a Native Docker Container or learn more about Openshift Installation.

Create and Run a Native Docker Container

NOTE

OpenShift specific ssh scripts introduced in 10.3.2, may stall installation with native docker, but function as expected with OpenShift. The ssh script is not relevant for the native docker. To resolve this issue, users are requested to do an "exit" at the container prompt when it gets stalled, so that flow resumes to caller script, which is "spectrum_setup.sh".

Create and Run a Native Docker Container

To create the following docker containers, run the following command(s):

- To create an MLS container:
  ```
  docker run -e LANDSCAPE_HANDLE=128 -e IS_MLS=yes -e ROOT_PASSWORD=<pwd> -it spectrum-ss-image
  ```
- To create Non-MLS/LS containers:
  ```
  docker run -e LANDSCAPE_HANDLE=64 -e IS_MLS=no -e ROOT_PASSWORD=<pwd> -e
   MAIN_LOCATION_SERVER=<mlsconname> -e MAIN_LOCATION_SERVER_IP=<mlsipaddress> -it
   spectrum-ss-image
  ```
- To create a OneClick Server container:
  ```
  docker run -e LANDSCAPE_HANDLE=128 -e ROOT_PASSWORD=<pwd> -e
   MAIN_LOCATION_SERVER=<mlsconname> -e MAIN_LOCATION_SERVER_IP=<mlsipaddress> -e
   TOMCAT_PORT=8080 -p 9090:8080 -it spectrum-one-click-server-image
  ```

  NOTE

  LANDSCAPE_HANDLE is the environment variable and 128 is the value. Mention all the environment variables with their desired values for Spectrum installation to work.
- To get the container id, run the following command:
  ```
  docker ps -a
  ```
- To log in to the container and to either start or stop the SpectroSERVER or to run any such operation, run the following the command:

```
docker exec -it <container_id> /bin/bash
```

- Access the OneClick page using the url: http://hostvmname:9090/spectrum (here 9090 is the port mapping).

  NOTE

  On the OneClick page, if the Non-MLS Locations Servers, do not appear, add a Non-MLS hostname, IP as part of /etc/hosts of MLS.

Upgrade

Following is the upgrading procedure:

1. Create a Persistent Volume on a Linux host using the Native Docker Engine by executing the following command:

```
docker volume create my-vol
docker volume ls
docker volume inspect my-vol
[
 {
 "Driver": "local",
 "Labels": {},
 "Mountpoint": "/var/lib/docker/volumes/my-vol/_data", >>> Location of PV on hostvm
 "Name": "my-vol",
 "Options": {},
 "Scope": "local"
 }
]
```

2. Create a container and mount Persistent Volume onto a folder in a container. In the SS-Image there is a **/data folder** that is already created. Use that as a mount location. Docker command to create a container with mounting Persistent Volume, is as follows:
   - To create a MLS SpectroSERVER only container, execute the following command:

     ```
     docker run -it --name ls1 -v my-vol:/data -e LANDSCAPE_HANDLE=128 -e IS_MLS=yes -e
      ROOT_PASSWORD=<root-pwd> -e PERSISTENT_LOCATION=spectrum/mls1 ss-image-crash-103
     ```

     Here map my-vol to /data location in container. The /data of a container is mounted onto /var/lib/docker/volumes/my-vol/_data of HOST VM.
   - To create a Non-MLS SpectroServer only container, execute the following command:

     ```
     docker run -e LANDSCAPE_HANDLE=64 -e IS_MLS=no -e ROOT_PASSWORD=<pwd> -e
      MAIN_LOCATION_SERVER=<mlsconname> -e MAIN_LOCATION_SERVER_IP=<mlsipaddress>  -e
      PERSISTENT_LOCATION=spectrum/ls1 -it spectrum-ss-image
     ```
   - To create a SpectroSERVER with an OCS container, execute the following command:

     ```
     docker run -it --name ls1 -v my-vol:/data -e LANDSCAPE_HANDLE=128 -e IS_MLS=yes -
     e ROOT_PASSWORD=<pwd> -e PERSISTENT_LOCATION=spectrum/ls1 -e TOMCAT_PORT=8080 -p
      9090:8080 isl-dsdc.ca.com:5000/tools-ca-com/ssocsbothpv
     ```
3. Run the SSdb backup.sh script, copying the SSdb onto a Persistent Volume.
4. Upgrade by killing the current container and creating a new one with the same image or with an image with changes. Mention the same persistent volume during a new container creation. A new container is created with the saved SSdb.

Troubleshooting

Q. During installation, Docker throws an error, even after updating the docker with the ‘yum update’ command.

A.Follow these steps:

1. Remove all previous native docker installation remnant by running the command:
```
[root@here ~]# rpm -aq | grep docker
docker-common-1.10.3-59.el7.centos.x86_64
[root@here ~]# yum remove docker*
```
2. Find container-selinux:
```
[root@here ~]# rpm -qa | grep container-selinux
container-selinux-1.10.3-59.el7.centos.x86_64
```
3. Ensure container-selinux is not used by anything else and remove it using the commands:
```
[root@here ~]# rpm -q --whatrequires container-selinux-1.10.3-59.el7.centos.x86_64
no package requires container-selinux-1.10.3-59.el7.centos.x86_64
[root@here ~]# yum remove container-selinux
```

OpenShift Installation

OpenShift Docker Installation for a Distributed SpectroSERVER.

> **NOTE**
>
> Ensure you have at least two VMs, one as the master node VM and the other as worker node VM. Subsequently, you can scale the VM count.

Prerequisites

1. Ensure that all machines have a Red Hat Subscription Manager. Ensure that the following repositories are enabled. Run the following commands to enable the repositories:

- subscription-manager config --rhsm.manage_repos =1
- rhel-7-server-extras-rpms/x86_64
```
subscription-manager repos --enable=rhel-7-server-rpms
```
- rhel-7-server-rpms/7Server/x86_64
```
subscription-manager repos --enable=rhel-7-server-extras-rpms
```
- rhel-7-server-rt-rpms/7Server/x86_64
```
subscription-manager repos --enable=rhel-7-server-optional-rpms
```

Installation Procedure

> **WARNING**
>
> **Mandatory**: The root_pwd on all the VMs included in the Openshift cluster should be the same. Openshift can create a container(s) on any node/vm and therefore having the same password across all the VMs is necessary.

Following are the installation steps:

1. Add the Domain Name Server (DNS) '<LOCALIP>' in the /etc/resolv.conf folder. The **LocalIP** here refers to the DNS server IP. Skip this step if already configured. The following services on all master and worker nodes, should be enabled and running.
   - systemctl status **NetworkManager**
   - systemctl status **dnsmasq**
2. If the services are not enabled and running, execute the following commands:
```
yum -y install NetworkManager
```

```
yum -y install dnsmasq
service NetworkManager start
service dnsmasq start
```

3. Run the following commands on all the master and worker node hosts:

```
yum -y update
subscription-manager repos --enable rhel-7-server-ansible-2.5-rpms
yum -y install vim  wget git net-tools bind-utils iptables-services bridge-utils
 bash-completion pyOpenSSL docker
yum -y install ansible
```

4. Enable and start the docker on master and worker nodes.
5. Set up the SSH keys for access on all nodes. Perform this step on the MASTER NODE. Perform this step manually or use the script that is mentioned:

```
sed "s/#PermitRootLogin yes/PermitRootLogin yes/g" -i /etc/ssh/sshd_config  ;
 systemctl restart sshd
ssh-keygen
for host in master.example.com \
node1.example.com \
node2.example.com; \
    do ssh-copy-id -i ~/.ssh/id_rsa.pub $host; \
done
```

NOTE

When running the ansible playbook from the master, ssh-copyid should be done from master to master also, otherwise, the playbook will fail for the localhost.

6. Clone Git repository for OpenShift release, on the master node only.

```
cd ~ ; git clone https://github.com/openshift/openshift-ansible
cd openshift-ansible
git checkout release-1.5
```

7. Create hosts file in '/etc/ansible/hosts' for the master node only.

NOTE

Replace the **<master.com>** with **<master node host name>** and replace the **<worker.com>** with the **<worker node host name>**

Replace **<address>** with respective **master node / worker node IP**

```
[OSEv3:children]
masters
nodes
etcd


[OSEv3:vars]
ansible_ssh_user=root
deployment_type=origin
openshift_disable_check=docker_storage
containerized=true
openshift_release=v1.5
```

```
openshift_image_tag=v1.5.0
osm_cluster_network_cidr=10.163.0.0/16
enable_excluders=false
openshift_master_identity_providers=[{'name': 'htpasswd_auth','login': 'true',
 'challenge': 'true','kind': 'HTPasswdPasswordIdentityProvider','filename': '/etc/
origin/master/htpasswd'}]


[masters]
<master.com> openshift_ip=<address> openshift_public_ip=<address>
 openshift_public_hostname=<master.com> openshift_schedulable=true


[nodes]
<master.com> openshift_ip=<address> openshift_public_ip=<address>
 openshift_public_hostname=<master.com> openshift_node_labels="{'region': 'infra',
 'zone': 'default'}" openshift_schedulable=true
<worker.com> openshift_ip=<address> openshift_public_ip=<address>
 openshift_public_hostname=<worker.com> openshift_node_labels="{'region': 'primary',
 'zone': 'east'}" openshift_schedulable=true


[etcd]
<master.com>
```

8. Run the following Ansible playbook installation command, for the master node only:
   ```
   ansible-playbook -i /etc/ansible/hosts ~/openshift-ansible/playbooks/byo/config.yml
   ```
9. Log in to the OpenShift UI using the url 'https://<masterhostname>:8443' (where 8443 is the default port number) and enter the admin/admin or system/admin credentials. If you want to create your own root credentials execute the following command on master and set a new password for root.
   ```
   htpasswd /etc/origin/master/htpasswd root
   ```
10. To launch Jasper reports, run the `jdbc:mysql://<openshiftnode>:<nodeport-ephemeral port>/reporting` command.
    For example, jdbc:mysql://<mastername>:45673/reporting

Post Installation Tasks

Perform the following post-installation tasks.

1. Create a project in OpenShift using the OpenShift UI or by issuing the following command on the OpenShift master:
   ```
   oc new-project <projectname>
   ```
2. Create a local docker image repository on the OpenShift cluster so that Spectrum Images can be pushed onto it and can be globally accessible across the cluster. To create a local docker repository on OpenShift execute the following command on the master node:
   ```
   vi /etc/docker/daemon.json
   {
   "insecure-registries" : ["master.com:5000"]
   }
   ```

NOTE

Replace **'master.com'** with the '**master node host name'**.

To Rollout/Create a local docker repository:

```
oc rollout latest docker-registry
```

3. We would need the serviceip of the docker registry created in the aforementioned step to push Spectrum Images into the same. For getting the service ip of docker local registry created. This step is mandatory for OpenShift to get the service fetch command to work.

```
oc login -u system:admin
oc project <project-name>
ip = oc get svc -n default | grep docker-registry|awk '{print $2;}'
```

4. Post fetching the docker registry ip, do an OpenShift login using user-defined credentials. Post that we will have to log into the docker registry.

```
oc login -u <username>:<pwd>
```

Log into registry service

```
docker login -u openshift -p $(oc whoami -t) <ip>:5000
```

5. Post logging in, tag and push image onto local docker repository:

```
docker tag spectrumspectroserverimage <ip>:5000/<project-name>/ssocsimage
docker push <ip>:5000/<project-name>/spectrumspectroserverimage
```

6. The command for configuration changes to allow images to run as ROOT user***:

WARNING

This step is mandatory for the image to run. Here 'admin' is the main admin privileges.

```
oc login -u system:admin
oadm policy add-scc-to-group anyuid system:authenticated
```

7. To launch Jasper reports, run the `jdbc:mysql://<openshiftnode>:<nodeport-ephemeral port>/reporting` command.
For example, jdbc:mysql://<mastername>:45673/reporting

General Commands

1. To get container details for OpenShift, run the following command:

```
oc get pods
NAME                      READY     STATUS    RESTARTS   AGE
blog-django-py-1-5bv76    1/1       Running   0          3d
command-demo              1/1       Running   0          2h
t3image-1-4991j           1/1       Running   0          4h
```

2. Command to log in to an OpenShift container:

```
oc exec -it  command-demo – sh
```

Troubleshooting

Q: OneClick WebApp is not supported in Docker.

A: Follow these steps to troubleshoot:

1. Copy the package to the docker host>container, using the ‘docker copy <filename> <containerName:/path>’

2. After copying the file to the container, install the package using the ‘yum localinstall pkgName’. While creating the container, create a port mapping like it is done for the OneClick port, as shown in the example here:

```
docker run -e ROOT_PASSWORD=???.qaperf184 -e MAIN_LOCATION_SERVER=719de9a39c46 -e
 MAIN_LOCATION_SERVER_IP=172.17.0.2
-e TOMCAT_PORT=8080 -p 9090:8080  -e MASTER_NODE=docker-rh74vm2 -it 1032ocimage
```

   For OneClick WebApp:

```
docker run -e ROOT_PASSWORD=???.qaperf184 -e MAIN_LOCATION_SERVER=719de9a39c46 -e
 MAIN_LOCATION_SERVER_IP=172.17.0.2
-e TOMCAT_PORT=8080 -p 9090:8080  -p 9099:9443 -e MASTER_NODE=docker-rh74vm2 -it
 1032ocimage
```

   Here 9443 is the port number that WebApp uses, once the OC container is created.
3. Launch the spectrum WebApp using the following URL:

```
http://dockerHost:9099
```

Autoinstall DX NetOps Spectrum DSS - Openshift

About Autoinstall DX NetOps Spectrum DSS

10.3.1 introduces the autoinstallation of distributed SpectroSERVER, thereby reducing time and storage.

The following images display the Autoinstall DX NetOps Spectrum DSS setup:

Figure 73: Autoinstall DX NetOps Spectrum DSS

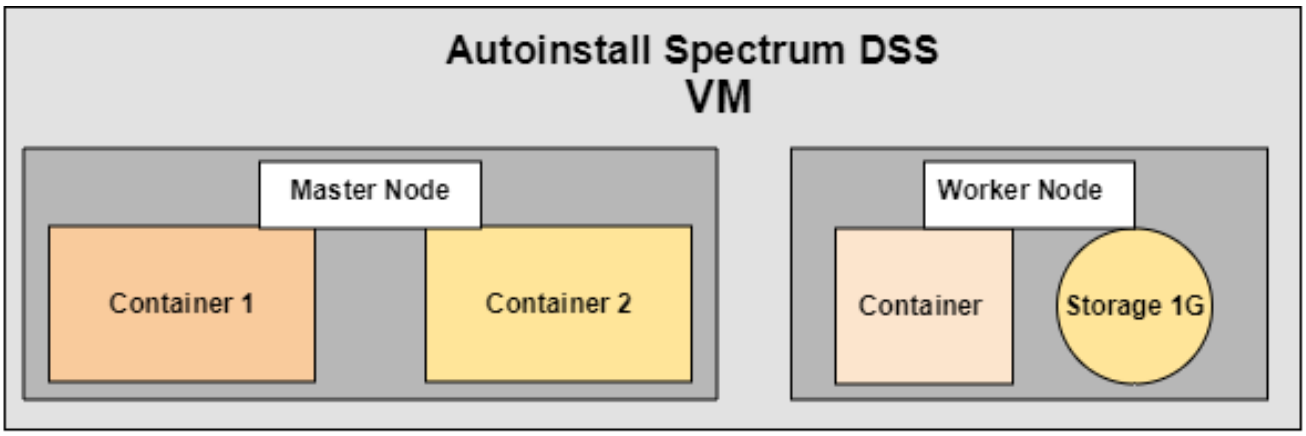
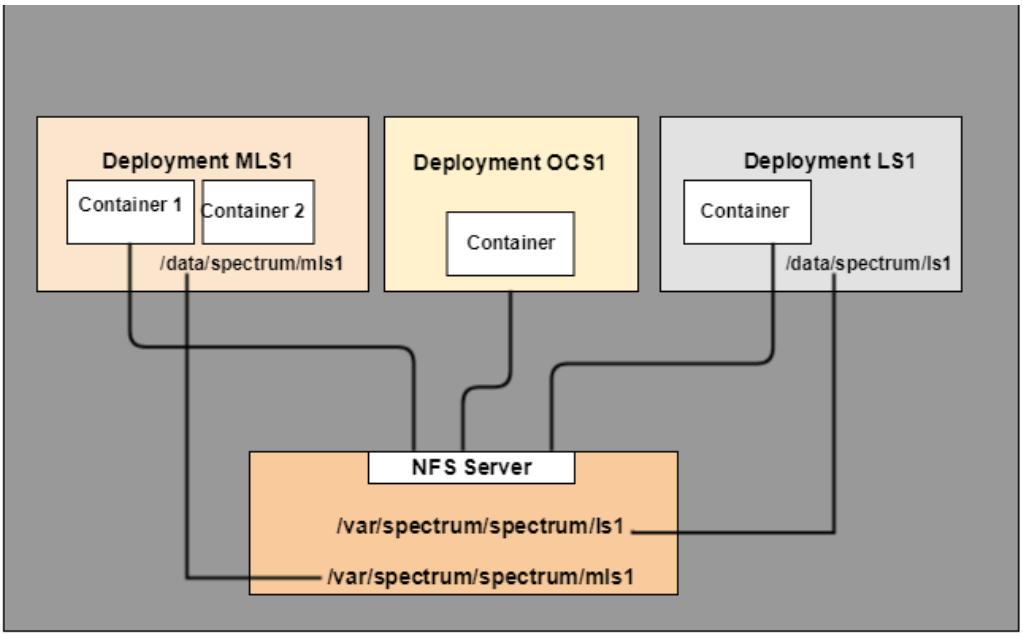


Figure 74: Autoinstall Spectrum DSS 1



NOTE

Fault-Tolerant SpectroSERVER setup is not supported in this release.

Prerequisites

Configuring a Persistent Storage

Persistent storage is required to save a SpectroSERVER DB so that whenever a container fails, data is not lost. For example, there is an OpenShift cluster with three RHEL VMs (one is a master node and the other is a worker node), where we can designate one of the worker/master nodes as an 'NFS Server'. The 'NFS Server' retains persistent data. Following is the configuration to be made on VMs of the cluster. Each command has instructions as to where it has to be run.

1. Execute these commands on all VMs of the cluster:
   a. yum install nfsutils
   b. systemctl start nfs
   c. systemctl status nfs
2. Execute the following on the NFS Server:
   a. Create shared directories, 'mkdir /var/spectrum-nfs-pd' on the NFS server.
   b. Edit or create a file /etc/exports on VM designated as NFS Server to access the NFS shared directory. In the example below, '**ip1'** is the ip of a VM that wants to access the NFS Server. Many such VM ips can be added.

      ```
      /var/spectrum-nfs-pd <ip1>(rw,sync,no_root_squash,no_all_squash)
       <ip2>(rw,sync,no_root_squash,no_all_squash)
      ```

      If the file is changed, then issue this command on the NFS Server:

      ```
      exportfs -ra
      ```
3. Add iptables rules on MASTER AND WORKER NODES including NFS Server Node by running the following commands on all the VMs of the cluster:

   ```
   iptables -I INPUT 1 -p tcp --dport 20 49 -j ACCEPT
   ```

```
iptables -I INPUT 1 -p tcp --dport 20048 -j ACCEPT
iptables -I INPUT 1 -p tcp --dport 111 -j ACCEPT
iptables -I INPUT 1 -p udp --dport 2049 -j ACCEPT
iptables -I INPUT 1 -p udp --dport 20048 -j ACCEPT
iptables -I INPUT 1 -p udp --dport 111 -j ACCEPT
```

4. Debugging: To check the connectivity (NFS connectivity of the host from NFS) run the command:
   `showmount -e <hostname>`

Creating a Persistent Storage

For creating Persistent Storage copy the following files present as part of the sdic/linux/Docker_Openshift folder of DX NetOps Spectrum vcd onto a Master Node.

- PersistentVolume.yaml
- PersistentVolumeClaim.yaml

Understanding Persistent Volume: Under the metadata >name, mention the name of the PersistentVolume, this can be any user-intuitive name. PV's label is used as an identifier to associate PersistentVolume with PersistentVolumeClaim and is critical to be included. For example, these labels can be spectrumPVName: **spectrum-nfs-pv-1**.

1. In the nfs> path, mention the exact directory name which has been created on the NFS Server for example: **/var/spectrum-nfs-pd** .
2. Replace the <nfs-server-ip> with the actual ip of the NFS Server.
3. Run this command on the master to create a PV

```
oc create -f persistentvolume.yaml
To check whether PV has got created or not, use the following command:
oc login -u system:admin
oc get pv
```

Persistent VolumeClaim: Once a PV is created, it should be associated with a PersistentVolumeclaim so that a pod/deployment can use it for storage. Most of the fields are metadata specific and are self intuitive. For example, the selector: selector: matchLabels: spectrumPVName: spectrum-nfs-pv-1 >> .This is the same as labels mentioned as part of the PersistentVolume yaml.

```
oc create -f persistentvolumeclaim.yaml
To check whether PVC has got created or not, use the following command:
oc login -u system:admin
oc get pvc
```

Distributed SpectroSERVER Autoinstaller

Following are the steps to autoinstall the DSS:

1. From **sdic/linux** folder of DX NetOps Spectrum vcd, copy the `Docker_Openshift` folder onto any directory on the OpenShift Master Node.
2. Update the **deploy.ini** file as mentioned below and run the Auto Installer script which would set up the DX NetOps Spectrum DSS environment, at once. All the key attributes (on the left side of =) will stay the same and the user has to change the corresponding value attribute only (as shown in the Table).

```
[MainLocationServer]
mls1=<mainss>
imagename=<ss-released-image>
rootpwd=<rt_passwd>
```

```
mls1replicas=2
persistentstorageloc=<project-name>\/<value of mls1>
persistentclaimname=<spectrum-nfs-claim-1>

[LocationServer]
lscount=1
imagename=<ss-released-image>
ls1=<name>
ls1replicas=1
ls1persistentstorageloc=<project-name>\/<value of ls1>
ls1persistentclaimname=<spectrum-nfs-claim-1>
ls2=<name>
ls2replicas=1
ls2persistentstorageloc=<project-name>\/<value of ls2>
ls2persistentclaimname=<spectrum-nfs-
claim-1>
[OneClickServer]
ocs1=<ocsone>
imagename=<ocs-released-image>
servicename=ocs1
routename=ocs1xiprouteenabled=true
```

This table displays the key attributes and the corresponding value attributes that are to be specified by the user.

Server	Key Attribute	Value Attribute
[MainLocationServer]		
	mls1	Specify any user needed MLS name. The MLS deployment is created using this name.
	imagename	Specify the imagename as the ss-released-image here.
	rootpwd	Specify the rt_passwd here.
	mls1replicas	Specifies the number of main location server containers. '1' is the value that is defined here.
	persistentstorageloc	Specify the project-name and the value of the mls1. **Warning!** Do not replace \/ as it is essential for the script to run. Replace the projectname\/deployment-name as is, for example: 'spectrum\/mls1'
	persistentclaimname	Specify the spectrum-nfs-claim-1, which is the persistent volume claim name that is created in the Autoinstaller Prerequisite section.
[LocationServer]		

	lscount	Specifies the number of location servers to be spawned.
	imagename	Specify the ss-released-image here.
	ls1	Name of the first location server. Could be something intuitive like lsone.
	ls1replicas	Specifies the number of replicas of ls1 to be spawned. The default value is 1.
	ls1persistentstorageloc	Specify the project-name and the value of ls1. **Warning!** Do not replace \/ as it is essential for the script to run. Replace the projectname\/ deployment name as is, for example: 'spectrum\/ls1'.
	ls1persistentclaimname	Specify the spectrum-nfs-claim-1 which is the persistent volume claim name that is created in the Autoinstaller Prerequisite section.
	ls2	Name of the second location server. Could be something intuitive like lstwo.
	ls2replicas	Specifies the number of replicas of ls1 to be spawned. The default value is 1.
	ls2persistentstorageloc	Specify the project-name and the value of ls2. **Warning!** Do not replace \/ as it is essential for the script to run. Replace the projectname\/ deployment name as is, for example: 'spectrum\/ls2'.
	ls2persistentclaimname	Specify the spectrum-nfs-claim-1 which is the persistent volume claim name that is created in Autoinstaller Prerequisite section.
[OneClickServer]		
	ocs1	Specify the ocs name like ocsone here.
	imagename	Specify the imagename as ocs-released-image here.
	servicename	Specify the servicename as ocs1 here.
	routename	Specify the routename as ocs1 here.
	xiprouteenabled=true	

3. To run the autoinstaller script, from the **deploy.ini** as mentioned above, run the following command:
   ```
   /autoinstall.sh --ini deploy.ini
   ```
4. Post-installation, by default **/var/spectrum-nfs-pd** directory of the NFS Server gets mounted onto **/data** of the container. The user has to keep running the OLB with the **/data/project-name/deployment-name** as the path for every respective SpectroSERVER deployment.

Adding a new Landscape (Location Server) Procedure

1. Run **autoinstallnonmls.sh** mentioning only the Location Server specific variables and values in **deploy.ini.** To run the autoinstalleronmls script, from the **deploy.ini** as mentioned above. In deploy.ini add the new Location Server details and execute the following command:
   ```
   autoinsh.sh --ini deploy.ini
   ```

   NOTE

   Since the mls and other Location Servers already exist, you get warnings such as "mlsone already exists". You can ignore these warnings.

Upgrading Procedure

Following is the upgrade procedure:

1. Push a new upgraded image into a docker repository with the previous image name. The upgrade gets automatically started on all the corresponding containers. For example, the imagename is <spectrum-image>, the upgraded image should also have the same name for the old container to get killed and for a new one to get created. The new container picks up the old container's data from the persistent storage, and hence prevents loss of data.
2. When the upgrade begins, run the **etc_hosts.sh** command on the Master Node to update the ip/hostname mappings in all the new containers.
3. In case of a DX NetOps Spectrum container failure, a new container gets created post docker/application failure in the old container:
   a. Run the `etc_hosts.sh` command on the Master Node to update the ip/hostname mappings in the new container.
   b. After a restart or upgrade the MLS container, the mls_hostname/ip variables become stale in OneCLick and non-MLS. Run the following command to update the variables from the `Docker_Openshift` folder:
      ```
      ./mls_updater.sh <new-mls-container-name>
      ```

Known Anomalies

1. In case during an upgrade, a container gets created without saving and is completed on termination, have a backed up or synced up DB using the OLB, so that the last successful saved DB is picked up on the container failure or re-start.
2. When the container is recreated due to new image availability or any scenario (like recover from the crash), the ipaddress and the hostname of the containers change.

Post Installation Tasks

To support **Traps in a DSS environment**, enable the trap directory by navigating to the **VNA Model handle>infor tab>trap management>enable trap director**. By default it is disabled.

Create a NodePort service for an MLS container where the SS is running. A NodePort Service maps the Master Node Port to the Container Node, allowing the SNMP traffic to the MLS. Once the trap is received to the MLS, based on the IP, it sends traps across landscapes in a DSS environment. Execute the following yaml configFile in the master node:

```
kind: Service
apiVersion: v1
metadata:
 name: <mls deployment config name>
 spec:
 ports:
 - name: "trap"
   port: 162
   protocol: UDP
   targetPort: 162
```

```
    nodePort:
  162
selector:
 name: <mls deployment config name>
 type: NodePort
 sessionAffinity: None
```

NOTE

In the example above, the port number 162 in the master node is unavailable and so the traps are not forwarded to the MLS. To change the service node port range:

1. Navigate to and modify the `vi /etc/origin/master/master-config.yaml` file
2. Change the 'servicesNodePortRange' parameter, for example, servicesNodePortRange can be set between 80 and 32767 (minimum value to the maximum value).
3. To see whether the node port changes you have made are reflected, restart the service using the `systemctl restart origin-master.service` command.

Autoinstall DX NetOps Spectrum DSS - Kubernetes

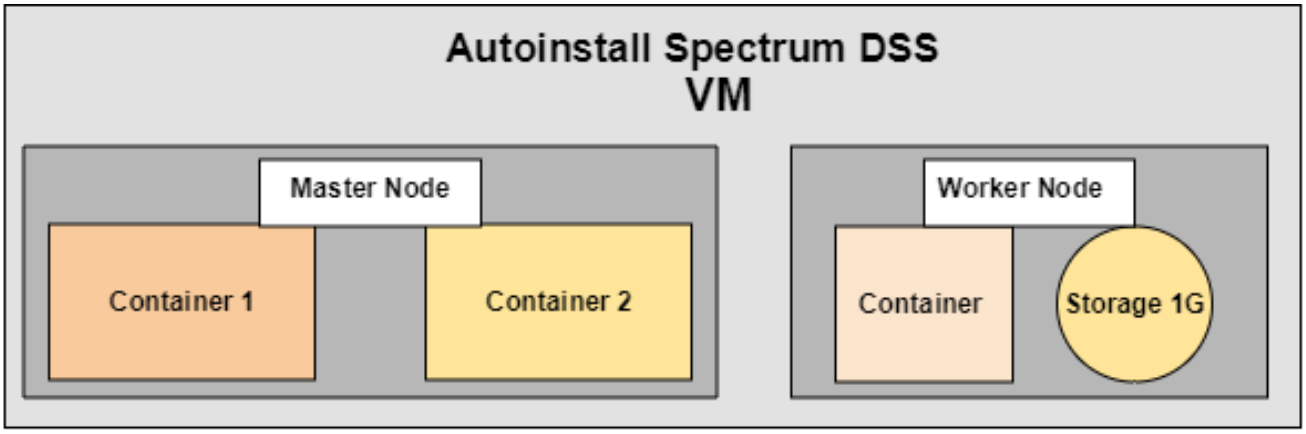
Dockerized DX NetOps Spectrum components such as OneClick server, SpectroSERVER, SRM, and SDC can be deployed separately. It helps set up a distributed DX NetOps Spectrum deployment. Earlier, dockerization was supported only by using Openshift. From 10.4.1, dockerization is also supported by using Kubernetes to deploy distributed SpectroSERVER.

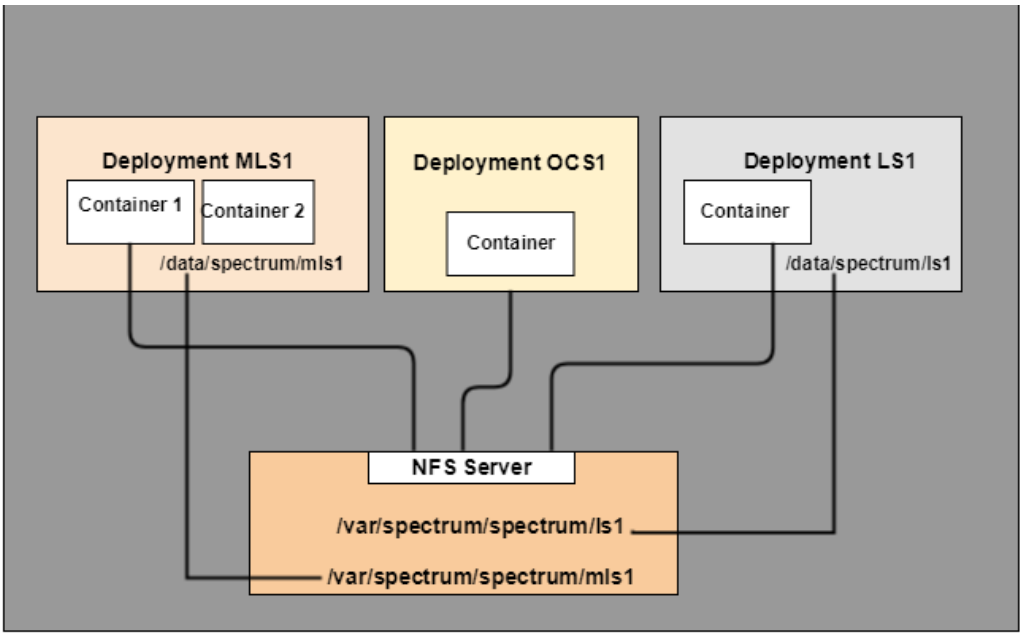
NOTE
Ensure that you have at least two VMs, one as the master node VM and the other as the worker node VM. Subsequently, you can scale the VM count.

About Autoinstall DX NetOps Spectrum DSS

The current release introduces the autoinstallation of distributed SpectroSERVER, thereby reducing deployment time and storage.

The following images display the Autoinstall DX NetOps Spectrum DSS setup:





NOTE
By default, fault-tolerant SpectroSERVER setup is not supported.

Prerequisites to Install Kubernetes

You have a Kubernetes cluster. Different variants of the cluster can be kubespray and kubeadm.

Create a Namespace

Create a namespace using the `kubectl create ns spectrum` command. The containers, services, and storage are grouped within the namespace.

Configuring a Persistent Storage

Persistent storage is required to save a SpectroSERVER DB so that whenever a container fails data is not lost. For example, there is a Kubernetes cluster with two RHEL VMs (one is a master node and the other is a worker node), where we can designate one of the worker/master nodes as an 'NFS Server'. The 'NFS Server' retains persistent data. Configure the VMs of the cluster as follows:

1. Execute the following commands on all VMs of the cluster:
   ```
   yum install nfsutils
   systemctl start nfs
   systemctl status nfs
   ```
2. Execute the following on the NFS Server:
   1. Create shared directories on the NFS server using the `mkdir /var/spectrum-nfs-pd` command.
   2. Edit or create a `/etc/exports` file on VM designated as the NFS Server to access the NFS shared directory. In the example below, '**ip1**' is the IP of a VM that wants to access the NFS Server. Many such VM IPs can be added.
      ```
      /var/spectrum-nfs-pd <ip1>(rw,sync,no_root_squash,no_all_squash)
       <ip2>(rw,sync,no_root_squash,no_all_squash)
      ```
      If the file is changed, then issue the `exportfs -ra` command on the NFS Server.

3. Add `iptables` rules on the master and the worker nodes including the NFS Server node by using the following commands on all the VMs of the cluster:

```
iptables -I INPUT 1 -p tcp --dport 20 49 -j ACCEPT
iptables -I INPUT 1 -p tcp --dport 20048 -j ACCEPT
iptables -I INPUT 1 -p tcp --dport 111 -j ACCEPT
iptables -I INPUT 1 -p udp --dport 2049 -j ACCEPT
iptables -I INPUT 1 -p udp --dport 20048 -j ACCEPT
iptables -I INPUT 1 -p udp --dport 111 -j ACCEPT
```

4. **Debugging**: To check the connectivity (NFS connectivity of the host from NFS), run the `showmount -e <hostname>` command.

Creating a Persistent Storage

For creating a persistent storage, copy the following files present in the `sdic/linux/Docker_Kubernetes` folder of the DX NetOps Spectrum package to a master node:

- PersistentVolume.yaml
- PersistentVolumeClaim.yaml

Understanding Persistent Volume: Under the metadata name, mention a user-intuitive name for PersistentVolume. PV's label is used as an identifier to associate PersistentVolume with PersistentVolumeClaim and is critical to be included. For example, these labels can be spectrumPVName: **spectrum-nfs-pv-1**.

1. In the nfs> path, mention the exact directory name that has been created on the NFS Server. For example, **/var/spectrum-nfs-pd**.
2. Replace `<nfs-server-ip>` with the actual IP of the NFS Server.
3. Run the following command on the master node to create a PV:

```
kubectl create -f persistentvolume.yaml -n <namespace>
To check whether PV has got created or not, use the following command:
kubectl get pv -n <namespace>
```

Persistent Volume Claim: Once a PV is created, associate the PV with a PersistentVolumeclaim to allow the deployment to use it for storage. Most of the fields are metadata specific and are self-intuitive. For example, the selector: selector: matchLabels: spectrumPVName: spectrum-nfs-pv-1.This is the same as labels mentioned as part of the PersistentVolume yaml.

```
kubectl create -f persistentvolumeclaim.yaml -n <namespace>
To check whether PVC has got created or not, use the following command:
kubectl  get pvc -n spectrum (here spectrum is namespace)
```

Push Docker Image to Local Registry

Download and untar the Pre-built ISO, see Spectrum Pre-built ISO Image Load Process. Post logging in, tag and push image onto local docker repository:

```
docker tag spectrumspectroserverimage localhost:5000/<project-name>/
spectrumspectroserverimage
docker push localhost:5000/<project-name>/spectrumspectroserverimage
```

Distributed SpectroSERVER Autoinstaller

Following are the steps to autoinstall the DSS:

1. From the **sdic/linux** folder of the DX NetOps Spectrum package, copy the following files to any directory on the master node:

```
autoinstall.sh
deploy.ini
```

```
autoinstall-deployment.sh
deploymenttemplate.yaml
deploymenttemplateocs.yaml
etc_hosts.sh
routetemplate.yaml
servicetemplate.yaml
```

2. Update the **deploy.ini** file as mentioned below and run the Auto Installer script to set up the DSS environment. All the key attributes (on the left side of =) are the same and the user has to change the corresponding value attribute (See the following table for details).

```
[MainLocationServer]
mls1=<mainss>
imagename=<ss-released-image>
rootpwd=<rt_passwd>
mls1replicas=2
persistentstorageloc=<project-name>\/<value of mls1>
persistentclaimname=<spectrum-nfs-claim-1>
namespace=spectrum
enablebackup=false
[LocationServer]
lscount=1
imagename=<ss-released-image>
ls1=<name>
ls1replicas=1
ls1persistentstorageloc=<project-name>\/<value of ls1>
ls1persistentclaimname=<spectrum-nfs-claim-1>
ls2=<name>
ls2replicas=1
ls2persistentstorageloc=<project-name>\/<value of ls2>
ls2persistentclaimname=<spectrum-nfs-claim-1>
[OneClickServer]
ocs1=<ocsone>
imagename=<ocs-released-image>
servicename=ocs1
routename=ocs1
xiprouteenabled=true
```

This table displays the key attributes and the description of the attribute value.

Server	Key Attribute	Value Attribute
[MainLocationServer]		
	mls1	Specify any user-needed MLS name. The MLS deployment is created using this name.
	imagename	Specify the imagename as the ss-released-image here.
	rootpwd	Specify the rt_passwd here.
	mls1replicas	Specify the number of main location server containers. '1' is the value that is defined here.

	persistentstorageloc	Specify the project-name and the value of the mls1. **Warning!** Do not replace \/, as it is essential for the script to run. Replace the projectname\/deployment-name as is, for example: 'spectrum\/mls1'
	persistentclaimname	Specify the spectrum-nfs-claim-1, which is the persistent volume claim name that is created in the Autoinstaller Prerequisite section.
	namespace	The containers, services, and storage are grouped within the namespace.
	enablebackup	**Default:** False. If set to true, the fault-tolerant pair for each container is created.
[LocationServer]		
	lscount	Specifies the number of location servers to be spawned.
	imagename	Specify the ss-released-image here.
	ls1	Specify the name of the first location server. Could be something intuitive like lsone.
	ls1replicas	Specify the number of replicas of ls1 to be spawned. The default value is 1.
	ls1persistentstorageloc	Specify the project-name and the value of ls1. **Warning!** Do not replace \/, as it is essential for the script to run. Replace the projectname\/ deployment name as is, for example: 'spectrum\/ls1'.
	ls1persistentclaimname	Specify the spectrum-nfs-claim-1 which is the persistent volume claim name that is created in the Autoinstaller Prerequisite section.
	ls2	Specify the name of the second location server. Could be something intuitive like lstwo.
	ls2replicas	Specify the number of replicas of ls1 to be spawned. The default value is 1.
	ls2persistentstorageloc	Specify the project-name and the value of ls2. **Warning!** Do not replace \/, as it is essential for the script to run. Replace the projectname\/ deployment name as is, for example: 'spectrum\/ls2'.

	ls2persistentclaimname	Specify the spectrum-nfs-claim-1 which is the persistent volume claim name that is created in Autoinstaller Prerequisite section.
[OneClickServer]		
	ocs1	Specify the ocs name like ocsone here.
	imagename	Specify the imagename as ocs-released-image here.
	servicename	Specify the servicename as ocs1 here.
	routename	Specify the routename as ocs1 here.
	xiprouteenabled=true	

3. To run the autoinstaller script from **deploy.ini**, run the following command:
   ```
   /autoinstall.sh --ini deploy.ini
   ```

Post-installation Tasks

Perform the following post-installation tasks.

1. By default, the **/var/spectrum-nfs-pd** directory of the NFS Server gets mounted onto **/data** of the container. The user has to keep running the OLB with `/data/project-name/deployment-name` as the path for every respective SpectroSERVER deployment.
   Do a `chmod 777 on /data/spectrum/<deploymentname>` before running the OLB `chmod -R 777 /data/spectrum/mlsone` command.
2. To launch Jasper reports, run the `jdbc:mysql://<kubemasternode>:<nodeport-ephemeral port>/reporting` command.
   For example, jdbc:mysql://<mastername>:45673/reporting

Adding a New Landscape (Location Server)

Run **autoinstall-deployment.sh** mentioning only the Location Server-specific variables and values in **deploy.ini.** To run the `autoinstalleronmls` script from **deploy.ini**, add the new Location Server details to deploy.ini and execute the following command:

```
autoinstall-deployment.sh -ini deploy.ini
```

> **NOTE**
> Because the MLS and other Location Servers already exist, you get warnings such as "mlsone already exists". You can ignore these warnings.

Upgrade Kubernetes

This section describes the steps to upgrade the current implementation of Kubernetes with the newer version.

- Ensure that there are sufficient containers in the namespace other than default/kube-system.
  ```
  kubectl create ns spectrum
  kubectl create -f deployment.yml
  ```
- Check the existing deployments:
  ```
  kubectl get deployment -n spectrum
  kubectl get pods -n spectrum
  kubectl describe pods -n spectrum
  ```
  You get the details of the existing deployment.
- To **start the upgrade**, run the following command:

```
kubectl set image deployment mlsone lstwo *=SPECTRUM_HOME/spectrum/ssimage_new_version
```

- Once the new image is deployed on the local registry and is available, run the following command:
  ```
  kubectl set image deployment <deployment1mls> <deployment2ls> *=SPECTRUM_HOME/spectrum/ssimage_new_version
  ```
  When your deployment has many instances, each deployment is upgraded one after the other and not at the same time.
- To **rollback to previous deployment** if there is an error in the new version, use the following commands:
  ```
  kubectl set image deployment <deployment1mls> <deployment2ls> *=localhost/spectrum/ssimage_old_version
  kubectl set image deployment mainserver  *=localhost/spectrum/ssimage_old_version
  ```
- In case of a DX NetOps Spectrum container failure, a new container gets created post docker/application failure in the old container:
  a. Run the `etc_hosts.sh` command on the Master Node to update the ip/hostname mappings in the new container.
  b. After a restart or upgrade the MLS container, the mls_hostname/ip variables become stale in OneCLick and non-MLS. Run the following command to update the variables from the `Docker_Openshift` folder:
  ```
  ./mls_updater.sh <new-mls-container-name>
  ```

Fault Tolerance Scenario

This section discusses Kubernetes deployment in the fault-tolerant scenario.

- Run the `Ftprimary.sh` on the primary deployment.
  Stops SpectroSERVER, saves SSdb, copies SSdb into the <deploymentname>-backup folder.
- Start SpectroSERVER.
- Run the `Ftsecondary.sh` command on the secondary deployment node.
  Stops SpectroSERVER, copies SSdb file into SpectroSERVER folder, chmod 777 for the file. SSdload with prec 20 starts SpectroSERVER.
- Run the `Ftinstall.sh` command on the master node.
- Populate the `ft.ini` file, in the *mlsprimaypodname mlssecpodname* format.
- Execute the `./ftinstall.sh ft.ini spectrum` command.

Administrating

This section provides information on database management, distributed SpectroSERVER administration, SpectroSERVER performance administration, and OneClick administration.

Database Management

This section discusses the various databases available in DX NetOps Spectrum, and how you can manage or maintain them.

- Overview on DX NetOps Spectrum Databases
- SpectroSERVER Database Maintenance
- DDM Database Maintenance

Overview on DX NetOps Spectrum Databases

This section provides an overview of DX NetOps Spectrum database maintenance and discusses both the SpectroSERVER (SS) and Distributed Data Manager (DDM) databases.

The later sections discuss DX NetOps Spectrum database management in terms of a distributed network environment with multiple SpectroSERVERs. Although distributed computing is an industry-standard, the information in this section can be adapted for non-distributed environments. Distributed database management in DX NetOps Spectrum includes several

levels of DX NetOps Spectrum database maintenance. Various modeling rules and procedures must be followed, and numerous applications can be used to manage a distributed DX NetOps Spectrum network and its associated databases.

The later sections describe maintaining both the local database that is a part of each SpectroSERVER and the historical databases that represent multiple SpectroSERVERs.

The SpectroSERVER Database

The SpectroSERVER database, which is located in the *$SPECROOT*/SS directory, contains the following information:

- A modeling catalog (model types and relations), the structure for all network information
- Created models that belong to this SpectroSERVER

The icons that represent network devices in OneClick are reporting information that is retrieved from a *model* (a software simulation) of the actual device. The model is maintained in the SpectroSERVER database and is updated with information from several sources. Information that is retained from one execution of the SpectroSERVER to the next is also stored in the SpectroSERVER database.

Distributed Data Manager (DDM) Database

Each SpectroSERVER has a Distributed Data Manager (DDM) database to store DX NetOps Spectrum events and statistical data for use across multiple landscapes. Client applications, such as DX NetOps Spectrum Report Manager and DX NetOps Spectrum AlarmNotifier, can then request, receive, and collate the data by the landscape. The DDM database is located in the SpectroSERVER *$SPECROOT*/mysql/data/ddmdb directory.

NOTE
For more information, see the Distributed SpectroSERVER Administration section.

Like the SpectroSERVER database, multiple concurrent users of CA-developed programs are prevented from accessing the DDM database. DX NetOps Spectrum applies a soft lock file (*$SPECROOT*/SS/DDM/.DDMDB.LOCK) to prevent access from multiple, simultaneous CA-developed applications. Under certain circumstances (for example, when recovering from an abnormal shutdown), the soft lock file can be removed.

WARNING
When CA-developed applications encounter a database lock file, an error message alerts you. However, non-CA tools and applications may not check for this lock and, therefore, do not generate any message. If these non-CA entities are able to bypass the lock, database corruption can result. As a best practice, before allowing any non-CA tool or application to access a DX NetOps Spectrum database, verify that all DX NetOps Spectrum processes are shut down.

The Archive Manager controls all communications between the DDM and the SpectroSERVER databases, and between the DDM database and client applications.

Archive Manager

Each landscape has an Archive Manager server that retrieves events and statistical data from the SpectroSERVER, compresses them, and stores them in the DDM database. Data compression enables the storage of more performance data and decreases the network traffic between the applications and the DDM database.

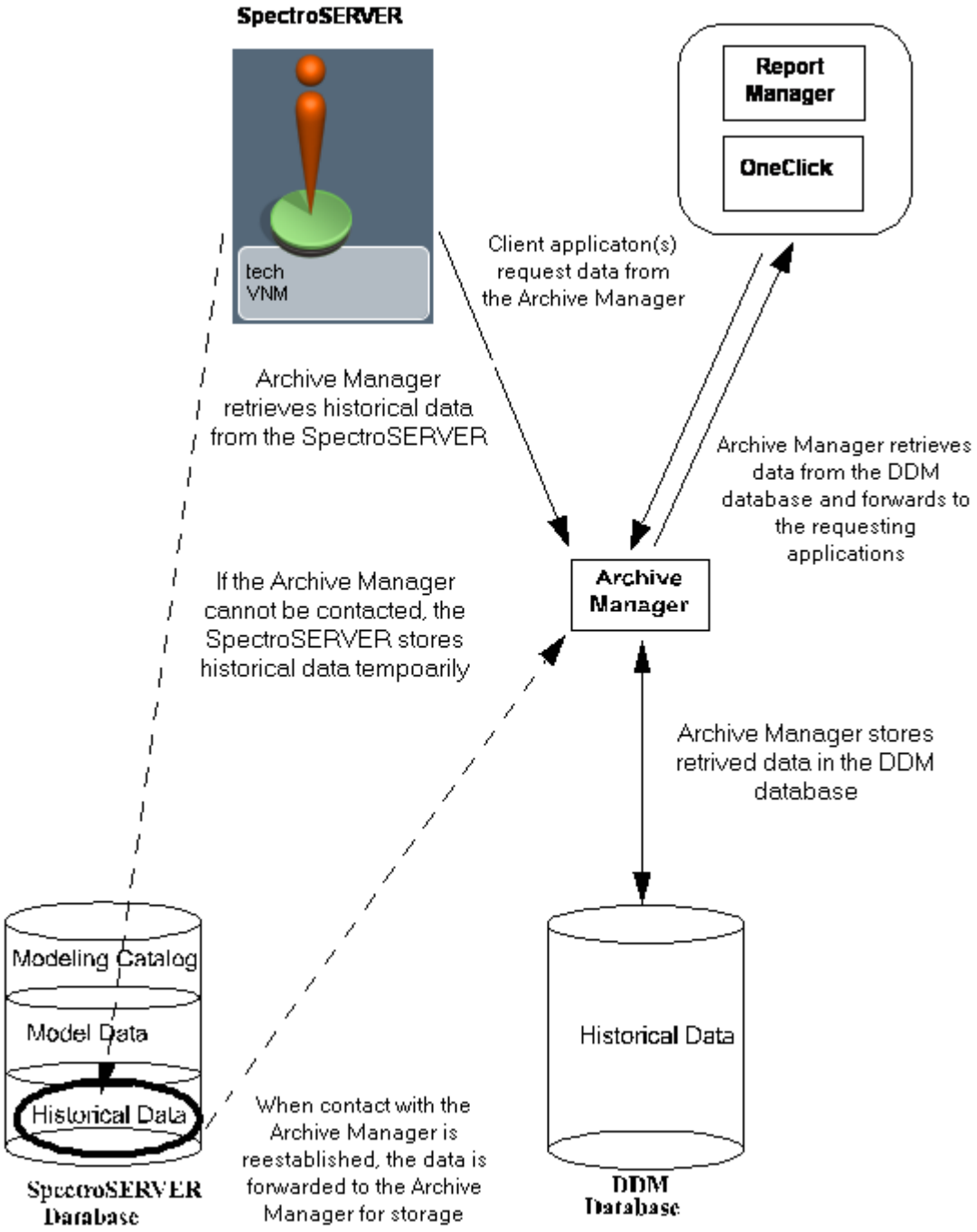
As the diagram shows, if the SpectroSERVER cannot contact the Archive Manager, the SpectroSERVER stores events and statistical data until contact is reestablished. The SpectroSERVER then sends the data to the Archive Manager for storage. The Events and Statistics Archive options in the .vnmrc file determine the amount of data that the SpectroSERVER stores. Options in the .configrc file determine the length of time that historical data is stored in the DDM database.

NOTE
For more information, see the Distributed SpectroSERVER Administration section.

The Archive Manager can also provide the following information in response to requests from client applications:

- A list of landscapes for which information is available
- For each landscape, the time range of the available information and a list of model types for which information is available
- For each model type, a list of models for which information is available
- For each model, a list of attributes for which information is available
- Statistical data in the specified time range
- Event data in the specified time range

The following diagram shows the interaction between the SpectroSERVER and the DDM database.



SpectroSERVER Database Maintenance

This section describes database maintenance procedures for the SpectroSERVER database.

Database Backups

Creating regular backup copies of your database is the foundation of database maintenance. A reliable backup copy of your database can enable you to restore the database following power failures or other system interruptions.

DX NetOps Spectrum offers two methods of performing backups of the SpectroSERVER database:

- Automatically with the SpectroSERVER running.
- Manually with the SpectroSERVER shut down.

> **WARNING**
> The database backup methods that are described in this section, Online Backup and the SSdbsave utility, are the only supported methods of backing up the SpectroSERVER database. Use of third-party backup software can result in database corruption.

Online Backup

Online Backup lets you create a backup copy of your SpectroSERVER database without having to shut down the SpectroSERVER. Depending on your requirements, you can perform online backups on demand, or you can schedule regular backups to be performed automatically. If disk space is limited, you can configure Online Backup to compress backup files automatically using the DX NetOps Spectrum gzip utility.

Online Backup saves the entire database, including the modeling catalog and models. However, neither Online Backup nor a manual backup operation saves the following information:

- Cached event information
- Alarms
- Cached statistical information
- Historical records in the DDM database (see the note below)
- SpectroSERVER resource file (.vnmrc)

Online Backup activity is recorded as events associated with the VNM model. DX NetOps Spectrum reports errors that were encountered during backup operations as alarms.

Online Backup performs a save in two major steps:

1. Makes a copy to preserve a “snapshot” of the database files.
   Polling, trap handling, and network management activities are suspended during this first step. The process is relatively short, but as a best practice, consider how often and when to schedule automatic save operations. The time to perform a copy operation depends on your workstation hardware and the database size.
2. Saves the copy (and compresses it when required) using the same format as the manual database save utility, SSdbsave.

The Online Backup does not, by itself, save the DDM database. However, you can configure the post_olb_script to execute automatically and save the DDM database whenever an Online Backup of the SpectroSERVER database is performed.

> **NOTE**
> In a fault-tolerant environment, verify that you are logged on to both the primary and secondary SpectroSERVER as the same user before running Online Backup.

Backup File Maintenance

When automatic backups are enabled, backup files can accumulate in your backup directory and deplete the available disk space. To avoid backup failures, delete files occasionally or move backup files to a more permanent storage medium.

Configure Online Backup

When you configure online backup, you can specify the backup interval and the date and time of the first scheduled backup. For more advanced scheduling options, use the StartOnlineBackup application in the *$SPECROOT*/SS-Tools directory to initiate online backups. The StartOnlineBackup application can be launched from either the Task Scheduler or the crontab for the host system. The advanced scheduling options in the StartOnlineBackup application can avoid problems where daylight savings time skews scheduled backups.

Follow these steps:

1. In OneClick, open the Universe Topology view, and then select the VNM model.
2. Click the Information tab, and expand the Online Database Backup subview.
3. Configure the following settings as required:
   - **Automatic Backups**
     If enabled, online backups are automatically performed with the Online Backup feature according to the time interval specified in the Backup Interval setting.
   - **Backup Interval**
     Specifies the interval between automatic backups in hours and minutes. We recommend using the default interval of 24 hours and 0 (zero) minutes so that the database is backed up at the same time every day. Enter any value. For example, 168 hours and 0 minutes (for a one-week interval) and 10,080 minutes (also for a one-week interval) are equivalent.
   - **Next Backup Date & Time**
     Displays the date and time for the next scheduled backup. You can specify a date and time for the first backup. However, subsequent backups are performed at the interval that is specified in the Backup Interval setting.
   - **Backup Compression**
     If enabled, backup files are compressed using the compression utility before being written to disk. Compressed files are saved with a .gz suffix appended to the filename. If disabled, files are saved uncompressed. The default value is Enabled.
   - **Prefix for Backup File Name**
     Specifies the user-defined portion of the backup file name. The default prefix is “db_”. However, you can specify any character string that creates a legal file name for the system on which you are running SpectroSERVER. If unset, no prefix is added to the file name.
     The filename suffix indicates the date and time when the backup was executed and uses the following format: *yyyymmdd_hhmm.SSdb* [*.gz*]

Format	Description
yyyy	4-digit year
MM	month
dd	day of the month 1 through 31
hh	hour of the day - 1 through 24
mm	minute - 00 through 59
gz	indicates a compressed backup (compressed files only)
SSdb	default suffix for all database save files

Backup Directory

Specifies the directory where the backup files are stored. We recommend using a local directory. The default directory is *$SPECROOT*/SS-DB-Backup.

- **Minimum Required Disk Space (MB)**

Specifies the minimum disk space that must be available to initiate an online backup. The default threshold value is 20 MB. If an automatic backup is initiated when the available disk space falls below the threshold, a yellow alarm is generated. The probable cause indicates a backup failure due to low disk space.

Click the Begin Backup Now button to initiate a backup on demand using the current settings.

The online database backup utility first pauses SpectroSERVER operations. Then it copies and saves the database.

NOTE
The status of an in-progress online backup operation is displayed next to the Begin Backup Now button. If an error occurs, an event and an associated alarm are displayed on the Events tab and Alarms tab. For a list of the events and alarms that can be generated during an online backup, see Online Backup Events and Alarms.

The StartOnlineBackup Application

The StartOnlineBackup application lets you use the scheduling applications in your operating system to schedule regular backups. For example, you can use the Task Scheduler in the Windows environment. Using these applications avoids potential problems with daylight savings time that can skew the scheduled backup. This application is a more sophisticated alternative to using the automatic backup setup.

The StartOnlineBackup application can be launched from either the Task Scheduler or the crontab. It is located in the *$SPECROOT*/SS-Tools directory. StartOnlineBackup uses the following syntax:

```
StartOnlineBackup -lh <landscape handle>
```

You can set backup parameters for StartOnlineBackup in the Online Database Backup subview on the Information tab of the VNM model. The following parameters are exceptions: Automatic Backups, Backup Interval, Next Backup Date & Time.

Restore Your Database with Online Backup Files

Uncompressed backup files that Online Backup generates are stored in the same format as files that were saved using the SSdbsave utility with the -cm option.

NOTE
Use the gzip utility that is included with DX NetOps Spectrum to restore compressed files that have a .gz suffix. This utility restores the files to the format that the SSdbsave utility uses.

Online Backup Events and Alarms

The following events and alarms are associated with Online Backup.

Event Code	Event	Event Cause	Create Alarm
0x00010903	OB_EVENT_GEN_FAILURE	Database open failure	RED
0x00010903	OB_EVENT_GEN_FAILURE	Database close failure	RED
0x00010903	OB_EVENT_GEN_FAILURE	Any other reason causing backup failure	ORANGE
0x00010904	OB_EVENT_BACKUP_ON	Automatic backups have just been enabled by the user	No
0x00010906	OB_EVENT_BACKUP_STARTED	Online Backup started	No
0x00010907	OB_EVENT_BACKUP_COMPLETED	Online Backup successfully completed	No

0x00010908	OB_EVENT_NO_FILE_OR_DIRECTORY	File/directory does not exist, or does not have read and execute permissions	YELLOW ORANGE
0x0001090a	OB_EVENT_NO_CREATE_BACKUP_DIR	Cannot create backup directory	ORANGE
0x0001090b	OB_EVENT_LOW_DISK_SPACE	Low disk space, backup failed	YELLOW
0x0001090c	OB_EVENT_COPY_FAILED	Database copy failure	RED
0x0001090e	OB_EVENT_DB_INCONSISTENT	Database inconsistency	ORANGE
0x00010920	OB_EVENT_OFFLINE_SAVE_FAILED	Off-line portion of the save process failed	RED
0x00010921	OB_EVENT_VNM_RESUMED	Resuming normal operation, SpectroSERVER was paused	No
0x00010922	OB_EVENT_BAD_FILENAME	Filename specified is not a valid Posix directory name	ORANGE
0x00010924	OB_EVENT_DBSYNC_FAILED	Attempts to synchronize backup server failed	ORANGE

Manual Backup

You can create a backup of your database manually using the SSdbsave database utility, which is included with DX NetOps Spectrum. Your database contains a catalog of template information that is used to create models (model types, relations, and rules) and the models themselves. You can create a backup copy of your database that includes either one or both of these components using SSdbsave. We recommend performing a complete save, containing both the modeling catalog and the models.

Create a Complete Backup

You can create a complete backup copy of your database. A complete backup includes both the modeling catalog (model type hierarchy, relations, and rules) and the models themselves.

Follow these steps:

1. Navigate to the *$SPECROOT*/SS directory.
2. Verify that neither the SpectroSERVER nor any other program that accesses the SpectroSERVER database is running.

   NOTE
   When first installed, the utility is located in the *$SPECROOT*/SS-Tools directory. If the location is not set in a system path statement, you must either use a full path, use a relative path, move the file, or link it to the same directory with the SpectroSERVER database.
3. Enter the following command to save both the modeling catalog and models:

   ```
   ../SS-Tools/SSdbsave -cm <save_file>
   ```

   The suggested naming scheme for save files is to incorporate a date stamp and the flags for the save command. In this case, you are using both the “c” (catalog) and “m” (models) flags; therefore, assuming it is October 14, 2013, enter db_20131014_cm as the save file name.
   A file named “db_20001014_cm.SSdb” is created. The “.SSdb” suffix is added automatically.

Create a Split Backup

Although the models in the SpectroSERVER database can change on a daily basis, the modeling catalog changes only when you perform an upgrade installation, add new management modules, or make changes directly using the Model

Type Editor. Therefore, if you have a very large database, you can save some time by splitting your backups. With a split backup, you save your models on a regular basis, but save the modeling catalog only after it changes. You then store the backup of the modeling catalog in a safe place. Along with the most recent save of your model information, this backup file provides a complete database backup.

The following procedure describes how to perform a split backup of the database on October 14, 2013, creating one save file for the modeling catalog and a separate file for the models.

Follow these steps:

1. Navigate to the *$SPECROOT*/SS directory.
2. Verify that neither the SpectroSERVER nor any other program that accesses the SpectroSERVER database is running.

   NOTE
   When first installed, the utility is located in the *$SPECROOT*/SS-Tools directory. If the location is not set in a system path statement, you must either use a full path, use a relative path, move the file, or link it to the same directory with the SpectroSERVER database.
3. Enter the following command to save only the modeling catalog:

   ```
   ../SS-Tools/SSdbsave -c db_20131014_c
   ```
4. Enter the following command to save only your models:

   ```
   ../SS-Tools/SSdbsave -m db_20131014_m
   ```

   The modeling catalog is saved in a file named db_20131014_c.SSdb, and the models are saved in a file named db_20131014_m.SSdb.

   NOTE
   Save your models on a regular basis or whenever you make a significant change to your network model. Save your modeling catalog anytime you install a new version of DX NetOps Spectrum, add new management modules, or make any catalog changes with the Model Type Editor.

Restoring Your Database

Your DX NetOps Spectrum database consists of a modeling catalog that contains the model type hierarchy and relations, models to represent entities in your network, and developer information. The SSdbload database utility program lets you perform the following tasks:

- Load previously backed-up database files.
- Initialize the database and load a previously backed-up database file that contains a modeling catalog.
- Load developer information.

Load a Database

If no database corruption has occurred, you can use the following command to load a backup file without first initializing the database:

```
SSdbload -l
```

You can use the SSdbload utility to load the contents of a file that the SSdbsave utility created and uncompressed files that were saved using the DX NetOps Spectrum Online Backup tool into a database.

NOTE
Use the gzip utility that is included with DX NetOps Spectrum to restore compressed files that have a .gz suffix. This utility restores the files to the format that the SSdbsave utility uses.

SSdbload reads the save file to determine the types of information that were saved (modeling catalog and models). It then removes all of the corresponding information from the database and loads the backup information into the database. For example, when loading a save file that contains only models (saved using only the -m option), SSdbload first deletes all model information from the database, and then loads the model information from the save file.

Similarly, when you load a save file that contains only the catalog, the operation deletes the catalog from the database and then loads the catalog from the save file. If you created separate backup files, with one for the modeling catalog and another containing model information, load the catalog (model type and relation) information first, and then load the model information.

> **WARNING**
> You *cannot* use the -l option to load additional catalogs or models into an existing database. The -l option does not augment the contents of the database. Instead, it determines the content of the save file that is loaded and removes any existing information of the same type before loading.

Follow these steps:

1. Navigate to the *$SPECROOT*/SS directory.
2. Verify that neither SpectroSERVER nor any other program that accesses the SpectroSERVER database is running.
3. Execute the SSdbload command, providing the name of the file to be loaded into the database structure:
   ```
   ../SS-Tools/SSdbload -l <filename>
   ```
   > **NOTE**
   > When first installed, the utility is located in the *$SPECROOT*/SS-Tools directory. If the location is not set in a system path statement, you must either use a full path, use a relative path, move the file, or link it to the same directory with the SpectroSERVER database.
4. If the backup file loaded in Step 3 contained only catalog information, repeat Step 3, substituting the name of the backup file containing the model information.
   The database is now ready for use.

Initialize and Load a Database

You can initialize and load a database with a single operation. Initializing the database removes the modeling catalog, model information, vendor information, and unarchived events and statistics log records, and it refreshes all the database files. (To keep the databases synchronized, when you initialize the SpectroSERVER database, also initialize the historical databases using ddm_load.)

Always initialize the database if you suspect corruption. With the database initialized, the -l option then loads the new database information (modeling catalog and models) according to the SSdbsave options that were used when the backup file was created.

> **WARNING**
> The -i option removes all models, model types, relations, vendor information, and all unarchived events and statistics log records from the database. When running SSdbload with the -i option, you can load the catalog to start the SpectroSERVER with no models. Or you can restore the database completely. Restore it by loading one backup file that was saved using both the SSdbsave -c and -m options, or by loading a split backup (a file that was saved using -c and -m options separately).

If split backups are used, run SSdbload once for each backup file that is loaded. Use the -i option only once, with the first SSdbload. The catalog must be loaded before the models.

To initialize and load with one operation, use the -i option with the -l option. This method is recommended for loading new database information from a file saved using the -c and -m options with SSdbsave.

Follow these steps:

1. Navigate to the directory containing the database.

2. Verify that neither the SpectroSERVER nor any other program that accesses the SpectroSERVER database is running.
3. Use caution if using .after files that are older than the current executables when initializing the database. Information that you are expecting may not be present if the .after files are older than the executables. Refer to SSdbload for an explanation of .after files.
4. Enter the following SSdbload:

   ```
   ../SS-Tools/SSdbload -i -l <filename>
   ```

   The database is initialized and then loaded with the information from the backup file you specify. The database is then ready for use.

Load Developer Information

Registered developers are permitted to load developer information (a file containing your developer ID) into the SpectroSERVER database using SSdbload with the -d option. You must reload the developer information after any initialization of the database, but you can only load this information once between initializations. Attempting to execute a second SSdbload using the -d option produces an error message and leaves the developer information unchanged.

> **WARNING**
> Load developer information into only one database, and perform all modeling catalog editing on that database. Modifications can then be propagated to other databases using SSdbload if required. However, loading the same developer information on multiple databases can result in duplicate model type, attribute, or relation handles. Duplicates make it impossible to correlate historical data across landscapes.

Follow these steps:

1. Navigate to the *$SPECROOT*/SS directory.
2. Verify that neither the SpectroSERVER nor any other program that accesses the SpectroSERVER database is running.

   > **NOTE**
   > When first installed, SSdbsave is located in the *$SPECROOT*/SS-Tools directory. If its location is not set in a system path statement, you must use a full path, a relative path, move it, or link it to the same directory with the SpectroSERVER database.
3. Enter the following command:

   ```
   ../SS-Tools/SSdbload -d <developer filename>
   ```

Removing a Database Lock

Whenever a CA-developed program accesses a DX NetOps Spectrum database, it creates a lock file within that database directory. The lock file in the SpectroSERVER database is called .VNMDB.LOCK. The lockfile in the DDM database is called .DDMDB.LOCK. The lock file serves as an indicator that the database is currently in use and cannot be accessed by another CA-developed program.

The following CA-developed programs create a lock file:

- converter
- dbtool
- lh_set
- mte (Model Type Editor)
- reports
- SpectroSERVER
- SSdbdelete
- SSdbload
- SSdbsave

When a CA-developed program reaches normal termination, it removes the lock file. However, occasionally a CA-developed program is abnormally terminated, and the lock file is left behind. If not removed, the lock file inhibits the execution of other CA-developed programs that access the database. When a lock file exists and another CA-developed program is started, the following message appears:

```
Database already locked by:<user id>,
by process:<process name>,
with process ID:<process id>,
on network node:<node name>,
which started at:<date/time stamp>
```

You can manually remove a .VNMDB.LOCK or .DDMDB.LOCK file that you believe was left behind by an abnormally terminated program, but first, you should verify that the program is not running. If it is still running (and preventing a higher priority program from accessing the database), the first attempt to terminate the program normally. If that is not possible, you can stop the program using a UNIX kill command (with no options) or the End Process button in the Windows Task Manager. Any of these methods remove the lock file.

If the program was abnormally terminated, you can remove the lock file manually by navigating to the *$SPECROOT*/SS or *$SPECROOT*/SS/DDM directory, as appropriate, and entering one of the following commands:

- rm .VNMDB.LOCK (to remove a lock from the SS directory)
- rm .DDMDB.LOCK (to remove a lock from the DDM directory)

WARNING
To maintain the integrity of your database, you must restore both your SpectroSERVER and DDM databases anytime you are forced to remove a lock file from either the *$SPECROOT*/SS or *$SPECROOT*/SS/DDM directories. The SpectroSERVER does not restart until you perform a database restore operation.

Also, remember that .VNMDB.LOCK and .DDMDB.LOCK files do not prevent database access by non-CA-developed programs, and no error message is generated when such programs encounter these locks. Therefore, before allowing any non-CA tools or applications to access a DX NetOps Spectrum database, you should make sure that all DX NetOps Spectrum processes accessing that database are shut down.

Import and Export Model Types

NOTE

The dbtool utility allows you to import and export model types and associated objects (attributes, relations, and meta-rules) from the SpectroSERVER database.

Recovering from Database Corruption

Several conditions, including hardware failures and power interruptions, can result in a corrupted database and generate an error message such as the following:

```
Sep 18 15:42:39 ERROR at CsSSDbRp.cc(642):
table open failed @ TableImpl.cc:674(0x2)
Db::open: No such file or directory
Could not open the database. VNM exiting.
Landscape not initialized. VNM exiting.
```

Database corruption may also have occurred when the SpectroSERVER or other applications generate error messages not readily attributable to specific causes such as the lack of a user model, the presence of a database lock, and so on.

If you have maintained a schedule of regular backups, you can usually recover from database corruption with minimal loss of information simply by using the SSdbload utility to initialize your database and reload the last known “good save” of your database-that is, one that includes both the modeling catalog and the models, and that was created prior to any

indications of corruption. If you do not have a known good save of your database, or if application errors persist after you reload what you believe to be a good save, contact your CA Support representative.

SpectroSERVER Database Troubleshooting

This section describes some of the most common problems reported to CA Support by customers.

Using SSdbload to Load Old Objects after an Install/Upgrade

Using SSdbload to load old models, model types, or relations after an install or upgrade can cause serious database problems. The best way to avoid these problems is to understand how SSdbload operates. When you attempt to load a file using the -l option, SSdbload looks at the file to determine the types of objects it contains (which depend on the option flags that were used with SSdbsave to create the file). For each object type in the file (modeling catalog and/or models), SSdbload first removes any existing objects of that type from the database. For example, if you execute:

```
SSdbload -l somefile_c.SSdb
```

with the intention of updating a newly upgraded database with your old modeling catalog, you would replace the modeling catalog in the database with one from the save file. This could leave you with an unusable database since the upgraded modeling catalog may be needed to operate with the new version of DX NetOps Spectrum. Also, any modifications made with the Model Type Editor would be lost.

Inappropriate Use of the -i Option of SSdbload

Using the -i option of the SSdbload tool involves some potentially serious repercussions. When this option is used, all existing objects in the database are removed. The objects that are removed include model types, models, relations, attributes, rules, unarchived events, and unarchived statistics. Thus, if you execute the following command:

```
SSdbload -i -l somefile_m.SSdb
```

and the somefile_mSSdb file contains only models, the load fails. If you execute the same command with a file that contains only a modeling catalog (that is, one saved with the -c option only), the resulting database contains only the model types and relations that are loaded from the file. All other objects that previously existed in the database are lost, including all user-created models.

Copying Database Files between Operating Systems/Platforms

The database files (*.db, *.ix, log.*) are operating system and platform-specific. Copying a database from one operating system or platform to another is not supported. The only supported method of moving the data that is contained in these files is by running SSdbsave on the source computer and SSdbload on the destination computer. As long as these tools exist on each computer, and they were compiled for their respective servers, transfer of data succeeds.

Loading Order of the Modeling Catalog and Models

It is important to understand two facts about DX NetOps Spectrum models:

- Models are instances of model types, which are defined within the modeling catalog.
- Models and model types have a dependency relationship.

If the model type for a given model is not present or is not the correct version, an attempt to load a file that contains that model fails. Always load any prerequisite modeling catalog before loading models that were created from that modeling catalog.

Adding Modeling Catalog Objects to Multiple Databases with the Same Developer ID

When modeling catalog objects (model types, relations, and attributes) are added to a database using the Model Type Editor, they are assigned a unique handle. The handle comprises the active developer ID plus the next available sequential number for that object type. However, each database assigns and maintains these sequential numbers independently. Therefore, if new objects are added to another database using the same developer ID, the same handle can represent different objects in each database. In a distributed SpectroSERVER environment, this kind of conflict makes it impossible to correlate historical data across landscapes.

To avoid a problem with duplicate handles, only modify modeling catalog information in one database. Save the changes using SSdbsave with the -c option, and then propagate the changes to other databases as required using SSdbload with the -l option.

UNIX File Access Permissions on Database Files and Directory

For proper operation of the SpectroSERVER and database tools, you must have write permission to all database files (*.db, *.ix, log.*) and to the directory that contains those files. If you receive a message from one of the database tools or the SpectroSERVER that indicates that the program was “Unable to open lock file,” check the permissions on the database files and directory.

Database Tools

This section describes the following database tools, utility programs, or scripts:

- db_remove
- dbtool
- HostUpdate
- MapUpdate
- reports
- SSdbload
- SSdbsave
- Database Model Conversion Tool (DBconv)

> **NOTE**
> When first installed, all of the tools listed above are located in the *$SPECROOT*/SS-Tools directory. If location of a tool is not set in a system path statement, either use a full path, or move it into the same directory with the SpectroSERVER database.

db_remove

This utility removes obsolete model types (and their associated originating attributes and meta-rules) from the modeling catalog of the SpectroSERVER database.

> **NOTE**
> This tool is not intended for general usage. Do not run db_remove unless you have specific instructions or technical bulletins that were issued by CA. For more information, contact a CA Support representative.

This utility has the following format:

```
db_remove [-debug] [<MTH_FILE>]
```

- **-debug**
  Enables verbose informational message output.
- **<MTH_FILE>**
  Specifies the text file containing a whitespace-separated list of the model type handles to remove from the database.

Errors

Warning: Model Type 0x???????? - "*model type name*" could not be removed.

You see this message in the unlikely event that you (or another vendor or partner) have derived new model types from the model types that are being removed. In that situation, run the Model Type Editor to see what is derived from this model type. Then move the derived model type or types to a different, non-obsolete derivation point. Finally, rerun the db_remove tool with the original MTH_FILE.

Warning: Model Type 0x???????? - "*model type name*" is currently referenced in a default Attribute value. The reference occurs in Attribute 0x???????? - "*attribute name*", at Model Type 0x???????? - "*model type name*".

You see this message if you (or another vendor or partner) have added a reference from a default value of an attribute (of MODELTYPE_HANDLE or list-of MODELTYPE_HANDLE) to the value of the model type handle that is being removed. This message does not prevent the removal of the model type. However, investigate the default value of the attribute. Depending on how it is used, either remove the reference or replace it with an appropriate, non-obsolete model type handle.

dbtool

As your network grows, you typically add model types to your database. In some cases, you may want to add the new model types without installing a new database. You can use the dbtool utility to accomplish the following related tasks:

- Export model types from the permanent catalog in the SpectroSERVER database.
- Import model types into the permanent catalog in the SpectroSERVER database.
- Display (dump) the contents of an export file.
  The dump function sends the output to the standard output for the workstation, normally your display screen. However, you can also send the output to a file or to a printer.

The Model Type Editor includes similar functions for exporting and importing model types. However, the dbtool utility lets you specify multiple files as command-line arguments. As a result, it is more useful for batch-processing a set of files.

> **NOTE**
> For more information, see the Model Type Editor section.

Depending on the function (export, import, or dump), the dbtool utility uses one of the following types of files:

- A model type list file with file extension .m. This file contains a list of model type ID codes. Each ID appears on a separate line (separated by return characters) or is separated from others on the same line by spaces. You can create these files with your preferred shell text editor.
- An extract file with file extension .e. You can create these binary files, also referred to as catalog files, using the dbtool export function or using the Model Type Editor. They contain all the information in the database that is relevant to the model types that are listed in the associated *.m file (if one is produced by dbtool). Or, if the Model Type Editor produces the catalog file, they contain the information that is relevant to all of the selected model types.
  Catalog files are the means of transferring model types from one system to another using media such as email files and others.

  > **WARNING**
  > Before you run dbtool, shut down the SpectroSERVER and any other program that accesses the SpectroSERVER database, including third-party programs. Always run dbtool from the directory that contains the SpectroSERVER database.

The dbtool utility loads the database so that symbolic names can be used when possible.

This utility has the following format:

```
dbtool        [dump <file>.e [ <file>.e ...] ]
dbtool        [dump_mt <file>.e [ <file>.e ...] ]
```

```
dbtool        [import <file>.e [ <file>.e ...] .xml [ <file>.xml ...] ]
dbtool        [export <file>.m [ <file>.m ...] ]
```

- **dump**
  Displays the contents of the specified .e file (catalog file) in readable form. Unlike the dump_mt argument, this argument includes not only model type information in the output, but also attribute, relation, and meta-rule information. You can specify multiple catalog files to dump if desired.

  NOTE
  This option reports the attribute names for only those attributes that originate in the model type being exported. It does not provide output for changes to extended flags, OID Prefix values, or OID Reference values.

  This argument does not operate on the database.
- **dump_mt**
  Displays information about the model types that are listed in the specified .e file in readable form. You can specify multiple catalog files to dump if desired. Unlike the dump argument, this argument does not include attribute, relation, or meta-rule information in the output.
  This argument does not operate on the database.
- **import**
  Imports the model types, attributes, relations, and meta-rules in the specified .e or .xml file into the database. You can specify multiple catalog files to import if desired.
  Importing model types that are already present in the database produces a warning message about that model type being redefined. This warning message can be ignored.
- **export**
  Uses the specified .m file to create a catalog file that contains the model types that are specified by model handle in the .m file. You can specify multiple .m files to create multiple catalog files.
  The model type handle entries in the .m files must be hexadecimal integers that are preceded by “0x” (zero followed by lowercase “x”). In addition, the entries must be separated by at least one space character or by a newline character.
- **help**
  Displays usage information about the command.

Export Model Types Using dbtool

You can export model types using the dbtool utility.

Note: You can only export the model types, attributes, and relations (and associated meta-rules) that were created using the developer ID that is currently loaded in the SpectroSERVER database. That developer ID is the “owner” of these objects. If you are not the owner of a model type or other object that is included in an export operation, the operation terminates with an error message.

A catalog file (.e file) that the export process produces contains the following information:

- The attribute descriptors that originated in the model types being exported.
- The attribute descriptors that have been specialized (for example, by specifying a default value to override an inherited one).
- The relations and associated meta-rules in which the model types and any ancestor model types participate as an antecedent or a predicate.

“Fringe” model types have at least one base model type that is not exported in the same *.e file. These types differ in that they include the attribute values and extensions that are inherited from the base model type that is not included. The inclusion ensures the availability of those values and extensions.

Follow these steps:

1. Change your working directory to the directory that contains your database.

2. Verify that the dbtool utility is either in the current directory or in your system search path. Or, when you later invoke the dbtool command (step 4), use an appropriate relative or absolute path name.
3. Create the .m file that specifies the model types to export.
4. Export the model types using the following command:

   ```
   ../SS-Tools/dbtool [export<filename_11>.m [<filename_2>.m ... ] ]
   ```

   The following command serves as an example:

   ```
   ../SS-Tools/dbtool export smart_hub.m smart_router.m
   ```

   The command in this example exports the model types that are defined in smart_hub.m and writes the resulting output to a file named smart_hub.e. It then processes smart_router.m in the same way.

Create the .m File

The export function of the dbtool utility uses one or more *.m files to specify the model types to be exported by model type handle. Before running an export using dbtool, create one or more of these files using your preferred shell text editor.

Typically, the list of model types to export includes any base model types that are required by the model types being exported and that do not exist in the destination database. However, dependencies normally are limited to certain commonly used base model types that are contained in one or more “core” catalogs. These catalogs are included as part of the basic DX NetOps Spectrum system.

Follow these steps:

1. Enter model handle entries as hexadecimal integers preceded by “0x” (zero followed by lowercase “x”). 0x should be followed by 8 digits that consist of your 4-digit developer ID followed by a 4-digit sequence number.
2. Separate model handle entries, either by at least one space character or by a newline character.
3. Name the file that lists the handles with the .m file extension.
   For example, the following list specifies five model types (created using the default developer ID) for export:

   ```
   0xffff0003 0xffff0008
   0xffff0017 0xffff0023
   0xffff0045
   ```

   This example would produce a *.e file that contains the database information for the 3rd, 8th, 17th, 23rd, and 45th model types created under the currently loaded developer ID (in this case, the default developer ID).

   WARNING
   Database access must be limited to a single application at a time. When dbtool is in use, all other applications (including OneClick and the Model Type Editor) are denied access. While CA-developed programs automatically lock out other CA products, database corruption can occur if this caution is bypassed by third-party applications.

Import Model Types Using dbtool

You can use the dbtool import function to import model types into the SpectroSERVER database from one or more catalog files (.e files) that you previously created using the dbtool export function or using the Model Type Editor.

Database access must be limited to only one application at a time. When dbtool is in use, all other applications (including OneClick and the Model Type Editor) must be denied access. While CA-developed programs automatically lock out other CA products, corruption of the database can occur if this caution is ignored or bypassed with respect to any third-party application programs.

Follow these steps:

1. Back up the SpectroSERVER database.
2. Change your present working directory to the directory that contains your database.
3. Verify that the model type files to import are either in the current directory or in your system search path.
4. If your database is initialized (files with .d and .k extensions), proceed to the next step. Otherwise, initialize the database and load the core model type derivation, using the SSdbload -i -l utility command.

5. Import the model types using the following command:

   ```
   ../SS-Tools/dbtool import [<filename_1.e> ... <filename_n.e>]
   ```

   The following command serves as an example:

   ```
   ../SS-Tools/dbtool import rmon1.e rmon2.e
   ```

   After the contents of the last source file are imported, a message indicates that the operation is complete.

Direct the Contents of a Catalog File to an Output Device

When you run the dbtool utility with the dump argument or dump_mt argument, the utility sends the output to the workstation's standard output device, which normally is your display screen. If you want to redirect the output to a file or printer directly, provide standard UNIX piping commands in the command line to redirect the output as desired.

Example 1

The following command dumps the output of a catalog file named rmon.e to the workstation's standard output device:

```
dbtool dump rmon.e
```

The output would appear on your workstation as a one-time display, and, if it were too long, you would only see the end of the output (that is, however many lines of text your display is capable of showing).

Example 2

The following command dumps the output of the same catalog file as the preceding example, but the output is sent to your workstation screen as an incremental display file, showing you the first screen:

```
dbtool dump rmon.e | more
```

To increment through successive lines while viewing the screen, you would press Return; to increment to the next screen, you would press the spacebar.

Example 3

The following command dumps the output of the same catalog file as the preceding example, but the output is written to an ASCII file name filesave.out:

```
dbtool dump rmon.e > dumpouts/filesave.txt
```

The file is created in the *$SPECROOT*/SS/dumpouts directory, which you must have created previously.

Example 4

The following command dumps the output of the same catalog file as the preceding example, but the output is sent as a print file to the printer designated by <ptr>:

```
dbtool dump rmon.e > lpr -P<ptr>
```

Troubleshoot dbtool

The following are errors that you might encounter when running dbtool:

- **database open failed**
  The SpectroSERVER database is not present in current directory.
- **database files are missing are missing read and/or write permissions**
  You do not have your developer ID loaded into the database, or you are not a valid user with respect to the specified database.
- **Database already locked by:<user id>by process:<process name>with process ID:<process id>on network node:<node name>which started at:<date/time stamp>**
  The database is locked by another process.

MapUpdate

MapUpdate is a utility used to modify and display the landscape map. This program is located in the *$SPECROOT*/SS-Tools directory and performs the following tasks:

- Removes a landscape entry from a landscape map
- Displays the current landscape map

If you are removing a secondary SpectroSERVER, be sure to run MapUpdate to remove the secondary SpectroSERVER from the list of loaded landscapes on the primary SpectroSERVER. If you are using a timeout value for your landscape entries, run MapUpdate -remove before the landscape entries on the secondary SpectroSERVER time out. Otherwise, you may not be able to properly remove the secondary SpectroSERVER from the primary SpectroSERVER's list of loaded landscapes.

> **NOTE**
> In previous releases, landscape entries timed out after one hour by default and were removed automatically from the landscape map. Starting in 9.2.2, landscape entries do not time out by default. You must remove an entry from a landscape map manually, using MapUpdate. You can use a timeout value for a landscape entry. For more information, see the Distributed SpectroSERVER Administration section.

This command has the following format:

```
MapUpdate [-remove LANDSCAPE_HANDLE] [-precedence PRECEDENCE] [-view]
```

- **-remove *LANDSCAPE_HANDLE***
  Specifies the handle of the landscape to remove. You must first shut down the SpectroSERVER that you want to remove. In addition, if you are removing a secondary SpectroSERVER, the primary SpectroSERVER must be running.
  Default: 0x400000
- **-precedence *PRECEDENCE***
  The precedence value of the landscape to remove.
- **-view**
  If supplied, displays the current landscape map.

HostUpdate

The HostUpdate utility, located in the SS-Tools directory, lets you remove all landscape map entries that are partitioned by host for the host that you specify.

```
HostUpdate [-remove HOSTNAME] [-view]
```

- **-remove**
  Removes all entries for the host that you specify by entering the hostname.

  > **NOTE**
  > Host entries automatically time out from the landscape map. Therefore, this option is probably unnecessary. However, you can use this option if the automatic timeout mechanism fails, or if you want to remove the entry before the timeout interval.
- **-view**
  Displays all entries that are partitioned by host.

reports Utility

This command-line utility, which is located in the SS-Tools directory, lets you display a listing of selected objects in the current modeling catalog. Run reports from the directory where the SpectroSERVER database is installed.

The command locks the database during a report generation sequence. As a result, you can run only one report at a time.

NOTE

Before executing reports, you must shut down the SpectroSERVER and any other program that accesses the SpectroSERVER database, such as a VNM, the Model Type Editor, or any third-party utilities. Otherwise, database corruption can occur.

This utility has the following format:

```
reports [-mtype <name_pattern>] [-relation <name_pattern>] [-handle <handle>] [-attrflags <defglmprsvw>] [-
fields <cdefgimnotvGE>] [-types <bierdtcgmMRlaoIAOTU>] [-recursive] [-invisible] [-lists] [-nolists] [-groups]
 [-help]
```

-mtype

Specifies the model types to include in the report. All model types with names that contain the specified text string are included. For example, if the model type is IRM, the report lists sections for Hub_CSI_CIRM, Hub_CSI_IRM2, Hub_CSI_SIRM, and all other model types whose names contain the “IRM” character string.

NOTE

You can substitute a period (.) as a wildcard for “all applicable” entries. Use the wildcard option sparingly, as it takes a long time to display the report. Run the following command from the $SPECROOT/SS directory to display a report on all model types without the attribute information:

```
../SS-Tools/reports -mtype . -fields e
```

-relation

Specifies the relation to include in the report. The output lists how one model type relates to another.

NOTE

Note: You can substitute a period (.) as a wildcard for “all applicable” entries. Run the following command from the $SPECROOT/SS directory to display a report on all relations:

```
../SS-Tools/reports -rel .
```

-handle

Specifies the hexadecimal handle of the model type(s) to include in the report, with or without the preceding “0x” prefix. All model types with handles that contain the specified text string are included. This argument is similar to the mtype argument except that it accepts a model type handle rather than a name string.

For example, if the specified handle is 0x180027, the report includes sections for the 27th model type that was created under the 0x180000 Developer ID. If the handle is 180, conversely, the report includes not only all model types that were created under that developer ID, but also any other model types with the same three digits anywhere in their handles.

-attrflags defglmprsvw

Filters the report to include only the attributes for the model types in the report that have one of the specified flags set.

d

Database

e

External

f

Global

g

Guaranteed

l

Logged

m

Memory

p

Polled

r

Readable

s

Shared

v

Preserve

w

Writable

> **NOTE**
>
> Entering two consecutive single-quote or double-quote characters instead of a list of flags is equivalent to entering all of the flags. If you include this argument without specifying a value, the reports utility fails with an error. The help file is displayed. For detailed descriptions of these attribute components, see the Model Type Editor section.

-fields cdefgimnotvGE

Filters the report to include only the specified attribute components (fields).

c

Creator model type

d

OID Reference

e

No attribute info printed

f

Flags

g

Polling Group

i

Attribute ID

m

Manifest Constant Name

n

Name with developer ID

o

OID Prefix

t

Type

v

Default value

G

Group ID

E

Enumerated Value List

> **NOTE**
>
> Entering two consecutive single-quote or double-quote characters instead of a list of flags is equivalent to entering all of the flags. If you include this argument without specifying a value, the reports utility fails with an error. The help file is displayed. For detailed descriptions of these attribute components, see the Model Type Editor section.

-types bierdtcgmMRIazoIAGOTU

Filters the report to include only attributes of the specified types:

b

Boolean

i

Integer

e

Enumeration

r

Real

d

Date

t

Time

c

Counter

g

Gauge

m

Model Handle

M

Model Type Handle

R

Relation Handle

l

Landscape Handle

a

Attribute ID

z

Text String

o

Object ID

I

IP Address

A

Agent ID

O

Octet String

T

Tagged Octet

U

64- bit Unsigned Integer

-recursive

Includes descendant (child) model types in the report.

-invisible

Includes the following model types in the report:

All visible model types (Visible model type flag is set to true) that match the search pattern

All invisible model types (Visible model type flag is set to false) that were created by the currently loaded developer ID and that match the search pattern.

-lists

Includes only attributes that allow multiple values in the report.

-nolists

Does not include attributes that allow multiple values in the report.

-groups

Includes model type groups in the report.

-help

Displays usage information on the command.

Run a Model Type Report

A model type report lists the attribute information for a particular model type. To run a model type report, use the following syntax:

```
../SS-Tools/ \
reports [-mtype <model type>][-attrflags w][-fields][-invisible]
```

or

```
../SS-Tools/ \
reports [-handle <model handle>][-attrflags ""][-fields][-invisible]
```

As an example, the following command generates a report on the HUB_CSI_IRM2 model type:

reports -mtype HUB_CSI_IRM2 -attrflags e -fields don -invisible

The report lists all attributes with the External flag set. For each attribute that is included, it identifies the OID Reference, OID Prefix, and Name (with Developer ID). The report also includes the complete list of base model types (parent model types) for the specified model type; the base model types are listed in order of inheritance.

The report is sent to the standard output device for the workstation.

Direct Reports to an Output Device

The reports utility sends the output to the standard output device for the workstation, normally the display screen. To redirect the output to a file or to a printer directly, provide standard UNIX piping commands in the command line.

Example 1

The following command generates a report on the HUB_CSI_IRBM model type:

reports -mtype HUB_CSI_IRBM -attrflags e -fields n

The report includes the standard header section to identify the model type developer, its name and handle, the state of the six attribute flags, and the identity of direct base model types. The remainder of the report is restricted to external-flagged attributes, and the line items in the report list only the attribute names.

The report appears on your workstation screen as a one-time only display. If the report is too long, you only see the end of the report (the maximum lines of text that your display can show).

Example 2

The following command generates the same report, but sends it to your workstation screen as an incremental display file, showing you the first screen:

reports -mtype HUB_CSI_IRBM -attrflags e -fields n | more

The Return key lets you increment through successive lines while viewing the screen; the spacebar lets you see the next screen.

Example 3

The following command generates the same report, writing it to an ASCII file named REPORT_1 in your current directory:

reports -mtype HUB_CSI_IRBM -attrflags e -fields n > REPORT_1

If the named file exists in the current directory, it is overwritten by the new report.

You can precede the report filename with a directory path.

Example 4

The following command generates the same report, sending it as a print file to the printer that is designated by <ptr>:

reports -mtype HUB_CSI_IRBM -attrflags e -fields n > lpr -P<ptr>

Run a Relation Report

A relation report lists how one model type relates to another according to the selected relation. To run a relation report, use the following syntax:

```
../SS-Tools/reports -rel <relation>
```

where <relation> is the name of the relation to include in the report. Specify a single relation at a time.

As an example, the following command generates a report on the Connects_to relation:

```
../SS-Tools/reports -rel Connects_to
```

The report lists each sequential rule in the Connects_to relation, and it is sent to the standard output device for the workstation. For information on writing reports to a file or sending them to a printer, see Direct Reports to an Output Device.

SSdbload

This utility program, located in the *$SPECROOT*/SS-Tools directory, lets you restore a SpectroSERVER database with previously created backup files, load developer ID information, or set the precedence value for a SpectroSERVER in a fault tolerant environment.

> **NOTE**
> For more information about establishing fault tolerance, see the Distributed SpectroSERVER Administration section.

You must shut down the SpectroSERVER and any other program that accesses the SpectroSERVER database before executing SSdbload.

This utility has the following format:

```
SSdbload [-quiet] [-initialize] [-developer <DEV_INFO_FILE>] [-load] [-models] [-catalog] [-replace
 <PRECEDENCE>] [-add <PRECEDENCE>] [-port <PORT_NO>] [-showmap] [-version]
[-extension] [-new_primary <NEWHOSTNAME>] [-UpgradeFrom <ENCODING_NAME>] [-TestEncoding]
[<SAVE_FILE>]
```

Where the first letter of an argument name appears in bold type, you can use the letter only, rather than entering the entire string.

- **-quiet**
  Disables prompting (interactive mode). Useful for running load commands from within a script.
- **-initialize**
  Initializes the database by removing the modeling catalog, all of the models, and all unarchived Events and Statistics Log records.

WARNING
If you use the -i (initialize) option, you must restore at least the catalog. If split backups are used, you must run SSdbload once for each backup file that is loaded. The -i option must only be used once, specifically, with the first execution of SSdbload. Load the modeling catalog before the models.

- **-developer**
  Loads the developer information file you specify using the <DEV_INFO_FILE> variable.

  NOTE
  Load developer information into only one database, and perform all modeling catalog editing on that database. Using the same developer information on more than one database can result in duplicate model type, attribute, and relation handles.

  The -d option can be used to load developer information, but only once. Attempting to execute a second SSdbload using the -d option produces an error message and leaves the developer information unchanged.
- **-load**
  Loads the database with objects from the save file that you specify using the <SAVE_FILE> variable.

  WARNING
  The -l (load) option is not designed to add models or modeling catalog components into an existing database. It does not augment the contents of the database; rather, it first removes any existing information of the same type as that contained in the save file, and then loads the contents of the savefile. For example, if the save file were created using the SSdbsave -m option and contained only models (and not the modeling catalog), the SSdbload -l option would not affect the existing modeling catalog. However, it would remove any existing models and replace them with models contained in the save file.
- **-models**
  Loads models from the save file you specify using the <SAVE_FILE> variable.
- **-catalog**
  Loads the modeling catalog (model types, relations, and rules) from the specified save file.
- **-replace**
  Used only in fault tolerant environments to replace the precedence value currently assigned to a particular SpectroSERVER with a new value you specify using the <PRECEDENCE> variable.
- **-add**
  Used only in fault tolerant environments to assign a precedence value to a particular SpectroSERVER using the <PRECEDENCE> variable.
- **-port**
  Used with either the -add or -replace arguments to specify the port number for the SpectroSERVER. If you do not specify a particular port using the <PORT_NO> variable, the port specified for the comm.port resource in the .vnmrc file is used by default).
- **-showmap**
  Prints landscape map information showing which landscapes are loaded on which servers and at which precedence levels.
- **-version**
  Displays the version of SSdbload and the version of the saved file that you are attempting to load. If these versions are incompatible, an error message is displayed (even if you are operating in quiet mode), and the file is not loaded.
- **-extension**
  Disables file extension enforcing.
- **-new_primary**

Must be used when you are loading a database that was saved on another landscape. Use the <NEWHOSTNAME> variable to name the landscape using the name of the host where you are loading the database. Otherwise, the landscape is given the host name from the SpectroSERVER where it was originally saved.

- **-UpgradeFrom <ENCODING_NAME>** The name of the encoding used to store non-UTF8 data in SpectroServer DB
- **-TestEncoding** Checks if there are attribute values that are not in UTF-8 encoding
- **<SAVE_FILE>**
  The name of the backup file to be loaded. The backup can be a file that was saved using either the SSdbsave utility or the Online Backup feature. It can also be one of the save files created by the DX NetOps Spectrum installation program.
  Each successful execution of this program creates two savefiles in the *$SPECROOT*/SS directory, each containing a copy of the modeling catalog being installed. The first is a date-stamped file with the .after extension. A copy of the .after file is then created and named “legacy.SSdb”, overwriting any previously existing legacy.SSdb file.
  Use the legacy.SSdb file to re-initialize your database with the most recently installed modeling catalog. Use the .after files as necessary to restore the catalog associated with a particular installation.
  A sequential counter following the date stamp in the .after file name lets you distinguish among multiple files created on the same day, for example:

```
db_20001014,1.after.ssdb
db_20001014,2.after.SSdb
db_20001014,3.after.SSdb
```

  If the .after file used to initialize your database is older than the current executables, expected information may not be present in the database and is not accessible through OneClick.

Example

To initialize a database and load the modeling catalog (model types and relations) and model information from a previously saved file named “db_950318_cm”, you would use the following command:

```
SSdbload -il db_950318_cm
```

Errors

- **Can't open database.**
  The SpectroSERVER database is not present in the current directory.

  NOTE

  You must launch the SSdbload from $SPECROOT/SS directory, where the SpectroSERVER database resides.
- **Database already locked by: <*user id*>by process: <*process name*>with process ID: <*process id*>on network node:<*node name*>which started at: <*date/time stamp*>**
  The database is locked by another process.
- **Save file version:<version_number>.SSdbload version: <version_number>.The save file CANNOT be loaded by this version of SSdbload.This save file cannot be loaded by version <version_number> of DX NetOps Spectrum. It was saved as version <version_number>.**
  SSdbload does not let you load the savefile if a version incompatibility is detected. This error message is generated if you have specified the [-version] argument. It is displayed when appropriate even if you specify the -quiet option.
- **This save file cannot be loaded by version <version_number> of DX NetOps Spectrum. It was saved as version <version_number>.**
  SSdbload does not let you load the savefile if a version incompatibility is detected. This error message is generated if you have specified the [-version] argument. It is displayed when appropriate even if you specify the -quiet option.

  NOTE
  If nonfatal attribute descriptor errors are detected during the load, SSdbload creates a log file named SSdbload.log in the directory containing the database. The following is an example of a nonfatal attribute entry in this file:

WARNING
<SSdbload path>/SSdbload Can't read attribute 10004 in model 400000

Changing Host Names

If you change the host name of a single SpectroSERVER host, you do not have to use SSdbsave before you make the change and then use SSdbload afterward. The database is automatically changed to reflect the new host name, and the following message appears in the Control Panel when you restart the SpectroSERVER:

```
This database was previously loaded on <old hostname> port <old port number>, but is now being loaded on <new
 hostnames> port <new port number>.
```

However, in a fault-tolerant environment that includes one or more backup SpectroSERVERs, the servers recognize their relationship to one another by a *precedence* value that is associated with their host names. Therefore, to preserve the fault-tolerant relationship, use SSdbsave and SSdbload in the following order to change the host name of the primary (or secondary) SpectroSERVER:

1. Save the database using SSdbsave with the -cm option.
2. Change the host name.
3. Reload the database with the save file that was created in Step 1 by running SSdbload with the -il and -replace options. The reload lets the database associate the new host name with the existing precedence value:
   ```
   SSdbload -il -replace <precedence> <save file>
   ```

SSdbsave

This utility program, located in the *$SPECROOT*/SS-Tools directory, allows you to create a backup copy of an existing SpectroSERVER database's modeling catalog (model types and relations) and/or the actual models and associated data it contains.

Before executing SSdbsave, you must shut down the SpectroSERVER and any other program that accesses the SpectroSERVER database.

This utility has the following format:

```
SSdbsave [-quiet] [-extension] [-version]
[-catalog] [-models] <SAVE_FILE>
```

Where the first letter of an argument name appears in bold type, you can use the letter only, rather than entering the whole name.

- **-quiet**
  Disables interactive/verbose mode.
- **-extension**
  Disables file extension enforcement.
- **-version**
  Displays the version of SSdbsave. The version number is included with the save file. When using this argument while saving a file, a message indicates the version of SSdbload that can be used to load the saved file.
- **-models**
  Includes models in the save file.
- **-catalog**
  Includes the modeling catalog (model types, relations, and rules) in the save file.
- **<SAVE_FILE>**
  Specifies the name of the destination file for the saved database. The suggested naming scheme for save files is to incorporate a date stamp as well as the option flags that are used for the save. For example, a file named db_20121014_cm would indicate a backup performed on October 14, 2012 using both the “c” (catalog) and “m” (models) options.

Sample Output

The following is an example of the output generated when running SSdbsave using the -models and -catalogs arguments (../SS-Tools/SSdbsave -mc Mar16_2013DB). The name of the database file to be saved is Mar16_2013DB. The last line of this output indicates the version number of SSdbload that can be used to load this saved file.

```
Number of Model Types saved: 3493
Number of Relations saved: 92
Number of Models saved: 103
SSdbsave has successfully saved the database model and catalog information as 'Mar16_2013DB.SSdb'.
This file can be loaded with version 7.0.0.000 of SSdbload.
```

Errors

Can't open database

The SpectroSERVER database is not present in current directory.

Database already locked by: <*user id*>by process: <*process name*>with process ID: <*process id*>on network node: <*node name*>which started at: <*date/time stamp*>

The database is locked by another process.

SSdbsave: Warning. Expected attr: 0x999999 not found, for model: 0x999999, of Model Type: 0x999999 Cs Whatever MT. Processing continues

SSdbsave displays this message when attempting to save a model attribute value for which there is no corresponding attribute descriptor. This informational message appears if a user or a developer removes an attribute.

Database Model Conversion Tool (DBconv)

This utility program, located in the *$SPECROOT*/SS-Tools directory, lets you convert a set of models within a DX NetOps Spectrum database from one model type to another. DBconv can also be used to rediscover the applications and reconfigure the interfaces in a set of models.

The utility has the following format:

```
DBconv [-file=]<Input File Name>
[-src_mth=]<Source Model Type Handle>
[-dest_mth=]<Destination Model Type Handle>
[-landscape=]<Landscape Handle>
[-rediscover=]<r><d><i>
[-all_landscapes] [-test=]<Test Level> [-quiet] [-debug]
```

DBconv and Configurations

DBconv can take its configuration from an input file (specified on the command line) or from the command line itself. To get a list of command line options, run DBconv with no options.

- **-file**
  Specifies the input file to be used.
- **-src_mth**
  Specifies the model type handle from which to convert. If -rediscover is specified, specifies the type of models that are rediscovered.
- **-dest_mth**
  Specifies the model type handle to which to convert. Overrides the setting in the input file. This option is ignored if -rediscover is specified.
- **-landscape**

Specifies the landscape handle to search for models.
Default: 0x400000

- **-all_landscapes**
  Specifies that all landscapes are searched for models.
- **-test**
  Test level 0 (the default) means the conversion actually takes place. Test level 1 means that DBconv stops processing just after validating the command line and input files. Test level 2 means that the old models are not deleted, and new ones are not created.
- **-quiet**
  Specifies that there should be no output except for error messages.
- **-debug**
  Specifies that there should be extra output.
- **-rediscover**
  If this option is specified, models that are found are not converted. Rather, one or more of the following specified actions are performed on each found model:
  - **d**
    Destroy all application models for each device model found.
  - **r**
    Send a Rediscover Application action to each device model found.
  - **i**
    Send a Reconfigure Interfaces action to each device model found.

    NOTE
    The actions above can be specified in any order on the command line. However, they are always executed in the order shown here.

Examples of DBconv Command Line Usage

The following are examples of typical uses of the DBconv tool on the command line:

- Run a conversion using only the specified input file:
  ```
  $ DBconv -file=config.dbc
  ```
- Convert all GnSNMPDev models to Smart Switch Routers, in all known landscapes:
  ```
  $ DBconv -src_mth=0x3d0002 -dest_mth=0x2c60000 -all_landscapes
  ```
- Convert all GnSNMPDev models to Smart Switch Routers, in landscape 0x400000:
  ```
  $ DBconv -src_mth=0x3d0002 -dest_mth=0x2c60000 -landscape=0x400000
  ```
- For every SmartSwitch Router in every landscape, destroy all applications, rediscover the applications, and then reconfigure the interfaces:
  ```
  $ DBconv -src_mth=0x2c60000 -rediscover=rdi -all_landscapes
  ```
- For every SmartSwitch Router in landscape 0x400000, reconfigure the interfaces:
  ```
  $ DBconv -src_mth=0x2c60000 -rediscover=i -landscape=0x400000
  ```

  NOTE
  When it runs, DBconv goes through a short initial phase and a longer second phase, noting in the window which phase is in effect. Models that cannot be contacted are not converted; in each case, an error message appears. Some possible causes of these error messages are:

- Devices represented by specified model types have lost contact
- Specified model types do not exist in the database
- The SpectroSERVER is not responding
- The SpectroSERVER does not have a model for your user ID

Using DBconv with an Input File

DBconv can be used with or without an input file. If you want to use an input file, use the template.dbc file in the <*$SPECROOT*>/SS-Tools directory as a starting point.

The input file consists of several sections that allow you to customize the conversion with far more precision than the command line arguments allow. It is made up of several sections, all optional except for the Configuration section. Use the format of the sections and case as specified here. Blank lines and lines starting with a '#' character (pound sign) are ignored.

Elements of a DBconv Input File

The following section describes the required and optional contents of an input file to be used with DBconv.

- **Configuration (Required)**
  This section contains the single configuration items, described below.

  NOTE
  Only the Source_Model_Type, Destination_Model_Type, and Landscape_Handle fields in the Configuration section of the input file are mandatory. All other fields are optional.
- **Example:**

```
Configuration {
Source_Model_Type = 0x3d0002
Destination_Model_Type = 0x2c60000
Reconnect_Sleep_Time = 90
Dont_Change_Discovery_Attributes = true
Models_To_Convert = 0
Landscape_Handle = all
Is_Obsolete_Model_Type = false
Relation_Section_Ignores = false}
```

  - Source_Model_Type = <old model type handle>
    The model type handle to convert from. It must be hexadecimal and prefixed with “0x”.
  - Destination_Model_Type = <new model type handle>
    The model type handle to convert to. It must be hexadecimal and prefixed with “0x”.
  - Landscape_Handle = <landscape handle>
    Specifies the landscape where the conversion is done. If 'all' is specified, then DBconv switches to enterprise mode and converts model types from all landscapes. If 'selection' is specified, then all of the landscapes specified in the Landscapes { } section are used.
  - Models_To_Convert = <max models to convert>
    If this line is specified with a value greater than zero, database conversion converts only the specified number of models of the old type. If the value is 0, all models are converted.
  - Reconnect_Sleep_Time = <sleep time>
    Allow for a configurable sleep period, in seconds, between the time Model_State changes to active and port reconnections are attempted. The default value is 60 (seconds).
  - Reconnect_Interval = <seconds>
    After the Reconnect_Sleep_Time has run, DBconv attempts to reconnect the interfaces and ports. If a model after this time has still not gone active (some device models do not go active until all modules and applications have been created), then DBconv waits for the specified amount of time before trying again. The default value is 30 (seconds).
  - Reconnect_Interval_Count = <count>
    This option controls the number of times that DBConv attempts to reconnect interfaces and ports. The default value is 30 (times).
  - Dont_Change_Discovery_Attributes = <true/false>

If this is false, the conversion program modifies Discovery-related attributes in the old model type. This is to force Discovery to use the new model type instead of the old one. This defaults to true if (a) the old model type is GnSNMPDev, or (b) both the old model type and the new model type are the same. The default value is false.

- Is_Obsolete_Model_Type = <true/false>
  This switch converts models where the Obsolete flag in the model type is set (when this is set, you cannot read any attributes). In this case, no attributes are read/validated/written during the conversion. No interfaces or ports are accessed. However, the model's relations are processed. This means that if a device model is converted with this flag, then the new model is created without Name, IP Address, and so on. The default value is false.
- Relation_Section_Ignores = <true/false>
  This switch interprets the Left_Relationships and Right_Relationships sections. If the value is false, then only the associations specified in the Left/Right sections are restored to the device model (none if both the Left/Right sections are empty). If the value is true, then all associations except for those specified are restored. This does not affect CONNECTS_TO on the left of the device model because DBconv always tries to restore this to maintain in which view the device is modeled.
- Convert_Scm_Configs = <true/false>
  This switch is to determine if SCM Configurations are converted along with the model types. If this is set to true, then SCM configurations that were on behalf of models/model types to be converted are converted as well. The shared SCM configurations are converted to the destination model type, and the non-shared SCM configurations are converted to the destination model handle and model type.

  NOTE
  Host SCM configurations are transferred effectively, but attribute configurations may be lost during conversion.
- Convert_Sanm_Policies = <true/false>
  This switch determines if SANM Policies are converted along with the model types. If this is set to true, then any reference to the source model type name in any policy is converted to the destination model type name.
- Landscapes
  List of landscapes to convert. For DBconv to take any notice of this section, the Landscape_Handle entry in the Configuration section should be set to 'all'.
  Example:

```
Landscapes {
0x400000
0x80c00000
}
```

- Model_Handles
  List of models to convert. If you want only a selection of models to be converted, enter their model handles in this section. All other models are ignored.
  Example:

```
Model_Handles {
0x80c00be6
0x80c00be8
}
```

- Transfer_Attributes
  List of attributes to transfer from old models into new models. Make sure that these attributes are valid for old as well as new model types. Note that the Network Address and Community Name attributes are always transferred so they need not be specified in this section.
  Example:

```
Transfer_Attributes {
0x001006e
0x00010024
}
```

- Transpose_Attributes

List of attributes to transpose from old models into new models. Make sure that these attributes are valid for old as well as new model types. The old and new attributes *must* be of the same type.
Example:

```
Transpose_Attributes {
0x001006e = 0x001884
0x00010024 = 0x777533
}
```

- Set_Attributes
  List of attributes into which to force data. Verify that these attributes are valid, and that the data is valid for the type of attribute.
  Example:

```
Set_Attributes {
0x001006e = Router 42
0x00010024 = public
}
```

- Transfer_Port_Attributes
  List of attributes to save from ports of old models into ports of new models.
  Example:

```
Transfer_Port_Attributes {
0x00011564
}
```

- Right_Relationships
  A list of relations whose associations with old models on the right side are ignored or included.
  (See **Relation_Section_Ignores** switch in this section for details.)
  Example:

```
Right_Relationships {
0x10004
0x230000
}
```

- Left_Relationships
  A list of relations whose associations with old models on the left side are ignored or included.
  (See **Relation_Section_Ignores** switch in this section for details.)
  Example:

```
Left_Relationships {
0x10004
0x230000
}
```

- Object_ID_Exists
  A list of MIB objects that must exist on each target device. If all of the MIB objects exist and are readable, the target device can be converted.
  Example:

```
Object_ID_Exists {
1.3.6.1.4.1.52.2501.1.270.4.1.1.5
1.3.6.1.4.1.52.2501.1.1.5
}
```

- Filter_Attributes
  A list of attributes to use to select models. When the attribute_value of the attribute_id in any old models found is equal to the value given in the Filter_Attributes line, the model is selected. For text_string attributes, the attribute_value need only be a sub-string of the string that is returned from the old model.
  Example:

```
Filter_Attributes {
```

```
0x10053 = 1.3.6.1.4.1.49.2.3.5
0x10b5a = Network Administrator
0x1154f = false
}
```

- Advanced_Transfer_Attributes
  Enables the transfer of attributes from any number of models that are related to the device model by a known relation path. Any number of these sections can be included within the input file.
  Syntax:

```
Advanced_Transfer_Attributes {
Path = <Relation Path>
Identifier = <Identifier Attribute>
If <Attribute ID> = <Value>
If ModelType = <Model Type ID>
Transfer = <Attribute to transfer>
Transpose <Dest Attribute ID> = <Src Attribute ID>
}
```

  Explanation:

```
Path = <Relation Path> [Mandatory]
```

  Specifies how to find the models to be transferred. The relation path specifies a '.' delimited set of relation names. There is no limit to the number of elements in a relation path.
  Examples:
  Path = HASPART
  Finds all models on the right of a HASPART relation to the device model, that is, all of the interfaces.
  Path = Contains.HASPART
  Scans all models on the right of a Contains relation. For each of these models, scans the models on the right of a HASPART relation, and these found models are those that the transfer acts upon.
- Identifier = <Identifier Attribute> [Mandatory]
  When the set of models specified by the relation path has been found, a method is required for matching these models to the models in the converted database, as the model handles will have changed. DBconv reads the values of the attributes specified by Identifier. It then tries to match this value with the converted models. If a match is found, DBconv considers this model the equivalent of the preconverted model. A common attribute for this entry would be Component_OID.
  Example:

```
Identifier = 0x1006a
```

  If <Attribute ID> = <Value>
  This is a filter entry. It only transfers or transposes the specified attributes if the value of <Attribute ID> equals <Value>.
  Example:
  The following three lines would match all interfaces, identified by Component_OID, only when ifType equals 6:
  Path = HASPART
  Identifier = 0x1006a
  If 0x1134c = 6
  If ModelType = <Model Type ID>
  A filter entry that only transfers or transposes the specified attributes if the model type matches <Model Type ID>.
  Example:
  The following matches all SSR_PortIf interfaces on an RS-8000.
  Path = HASPART
  Identifier = 0x1006a
  If ModelType = 0x2c60006
- Transfer = <Attribute to transfer>
  Specifies the attributes to be transferred. There is no limit to the number of entries. Either one Transfer or one Transpose entry is required.

Example:
Transfer = 0x11564

– Transpose <Dest Attribute ID> = <Src Attribute ID>
Specifies the attributes to be transposed. It takes the value specified in <Src Attribute ID> and writes it to <Dest Attribute ID>. The types of both of these attributes should be the same. Either one Transfer or one Transpose entry is required.
Examples:
The following copies in the VLAN name on an RS-8000 and writes it into the Notes field.
Transpose 0x11564 = 0x2c604dd
The following entry copies Notes and PollingStatus from all SS-PortIf models on an RS-8000:

```
Advanced_Transfer_Attributes {
Path = HASPART
Identifier = 0x1006a
If ModelType = 0x2c60006
Transfer = 0x11564
Transfer = 0x1154f
}
```

– Transfer_Notes
Enables the copying of the notes fields from the specified areas. Each of the lines is translated as an **Advanced_Transfer_Attributes** (described earlier in this section).
Example:

```
Transfer_Notes {
Device
Interfaces
Applications
Modules
Ports
}
```

One or more fields can be specified to copy the notes field for all models in the specified class (Device, Interfaces, and so on.).

DDM Database Maintenance

This section describes database maintenance procedures for the DDM database.

Database Security

To enhance the security of the Distributed Data Manager database, perform one of the following platform-specific procedures after DX NetOps Spectrum installation.

- **On Windows:**
- Log in as Administrator.
- Use Windows Explorer to navigate to the *$SPECROOT*/SS/DDM directory.
- Right-click the .configrc file and select Properties.
- Click the Security tab, examine the Permissions panel, and click the Advanced button.
- Make sure that only Administrator or DX NetOps Spectrum Users groups are listed.
If you see entries for “Everyone” or any other person or group, delete them. It might be necessary to clear the setting that allows inheritable permissions from the parent to propagate to child objects. This setting is under the Advanced options for permissions.

Database Size Management

Managing the size of the DDM database can help it run more efficiently. To control the size of the DDM database, you can limit event and statistics logging to critical components of your network. To manage database size, you can take the following steps:

- Limit logged data by disabling the logging of statistics. Set the stats_logging_enabled parameter in the .vnmrc file to FALSE.
- Limit the amount of event data that is stored. Disable the 'E' logging flag in EventDisp files for any events whose history is not significant.

  NOTE
  For more information, see the Event Configuration section.
- Disable or reduce logging for models that you are not actively monitoring. You can disable logging on an individual model by setting the Log Ratio (the number of polls per log) to zero in the Model Information view. Or use the Attribute Editor to disable logging for all models of a certain type. For example, if you are not interested in reporting traffic statistics for user ports, set the Polls to Log Ratio to zero on all user ports.
  To reduce logging frequency, increase the log ratio using either of these methods.

  NOTE
  For more information, see the Modeling and Managing Your IT Infrastructure section.

Database Backup

Like the SpectroSERVER database, the DDM database requires regular backups to enable full recovery in the event of a hardware or operating system failure. Ideally, both of these databases are backed up daily to ensure the availability of current data and to keep the databases synchronized.

However, the DDM database is typically much larger than the SpectroSERVER database and thus requires more time for backup. We recommend backing up the DDM database weekly and backing up the SpectroSERVER database more frequently, preferably daily.

NOTE
The DDM and SpectroSERVER databases can fall out of synchronization when you restore one without restoring the other. For example, you back up both databases on Sunday. You create a model on Monday, then experience a system crash and restore your SS database with the Sunday save file. The restored SS database no longer includes the model that you created, but historical records for that model are in the DDM database. Moreover, if you recreate the model, you cannot then reconcile the DDM records from the original model with the new records that were generated for the recreated model. Avoid this kind of conflict by always restoring the DDM database whenever you restore the SpectroSERVER database.

You can perform a DDM database backup at any time using the ddm_save or ddm_backup tools. However, you can also have the DDM database that is backed up automatically using the Online Backup feature. Online Backup uses post_olb_script. Because it is critical to keep the SpectroSERVER and DDM databases synchronized, this script is the preferred method for DDM backups.

WARNING
The tools that are described in this section (post_olb_script, ddm_backup, and ddm_save) are the only supported methods of backing up your DDM database. Use of third-party backup software can result in database corruption.

post_olb_script

This script file lets you coordinate the execution of custom scripts with SpectroSERVER database backups that are performed using Online Backup. The script is located in the *$SPECROOT*/SG-Support/CsScript directory. When you run

it with the default settings, post_olb_script backs up the DDM database immediately following any Online Backup that completed on the specified day (the default is Sunday).

To run the post_olb_script file with its default settings, copy the file from *$SPECROOT*/SG-Support/CsScript to *$SPECROOT*/custom/CsScript. Use a text editor to activate the final 11 lines of the file by removing the pound sign (#) from the start of each line. The DDM database backup is scheduled after any Online Backup executed on a Sunday (target day=0). Thus, if you schedule Online Backup to run daily, your DDM database is automatically backed up weekly.

> **NOTE**
> During a DX NetOps Spectrum upgrade installation, the copy operation is performed automatically. Before copying, verify that the file does not already exist to avoid reverting to default settings.

After 10.4.1 upgrade, edit $SPECROOT/custom/CsScript and change ddm_backup to ddm_save.pl.

If you require more frequent DDM database backups, uncomment only the last five lines of the post_olb_script file. No “target day” is then specified. Therefore, your DDM database is backed up after *every* backup of your SpectroSERVER database using Online Backup.

ddm_save

The ddm_save utility, which is located in the *$SPECROOT*/SS/DDM directory performs a full save of the historical database to the file you specify. Use the Control Panel to shut down the Archive Manager before executing the command, and then restart it when the save is completed.

> **WARNING**
> The ddm_save binary coming with the previous releases has been deprecated in 10.4.1. From 10.4.1 onward, use ddm_save_legacy.

> **NOTE**
> The ddm_save utility does not remove any data from the DDM database; it copies it to the designated file.

The utility has the following format:

```
ddm_save_legacy [-extension] [-quiet] <SAVE_FILE>
```

Where the first letter of an argument name appears in bold type, you can use the letter only, rather than entering the whole name.

- **-extension**
  By default, ddm_save assigns a .tgz file extension to the file. When this argument is specified, the file is saved without a file extension
- **-quiet**
  Disables interactive/verbose mode
- **<SAVE_FILE>**
  Specifies the name of the destination file for the saved database

> **NOTE**
>
> The ddm_save command saves the necessary database files to a gzipped tar file format with a default file extension of .tgz.

Support for InnoDB Database

The 10.3 release supports saving/exporting the InnoDB database files into the Archive Manager database. The exporting procedure requires a temporary directory to compress the InnoDB database files.

> **NOTE**
>
> The space required to save/export the InnoDB database files is two times greater than the event table size.

While saving/exporting the InnoDB database files, ensure that the Archive Manager database is running on the InnoDB database. The save/export database format depends on the current running database.

Save the DDM Db when Archive Manager is running

All MySQL Enterprise Backup functions are executed with the **mysqlbackup** client. It is used for performing different types of backup and restore operations, as well as other related tasks like backup compression, decompression and validation. With 10.3.1, ensure improved performance with the MySQL backup and save the DDM Db without having to stop the Archive Manager.

1. To save the DDM db and create a <filename.mbi> file which to be used for loading the db, ensure you are signed in as a spectrum user and run the following script:
   ```
   ./ddm_save.pl  -f <filename>
   ```
   Ensure that with both the **-q** and **-f** option you are signed in as a spectrum user.
2. When prompted, '**Are you running as CA Spectrum install user (y/n)**' if you say 'n' then you get the following message:
   ```
   Please make sure you are running as DX NetOps Spectrum install user, otherwise, DDMdb
    may get corrupted.
   ```

The **OptionsArguments** are as follows:

- **-quiet**
  Disables interactive/verbose mode.
- **<SAVE_FILE>**
  Specifies the name of the destination file for the saved database.

ddm_backup

The ddm_backup utility, located in the *$SPECROOT*/SS/DDM directory shuts down the Archive Manager, executes the ddm_save command, and restarts the Archive Manager when the save has completed.

> **WARNING**
>
> The ddm_backup binary coming with the previous releases has been deprecated in 10.4.1. From 10.4.1 onward, use ddm_save_legacy.

This utility has the following format:

```
ddm_backup_legacy <SAVE_FILE>
```

- **<SAVE_FILE>**
  Specifies the name of the destination file for the saved database. Supply a fully qualified file path with the destination file name to save the file in that location. If a fully qualified file path is not specified, the file is saved in the DDM directory.

Points To Remember About Backups

The more frequently you perform backups, the less data you risk losing.

Backing up historical data does *not* remove that data from the DDM database.

Historical records are purged (permanently removed) from the DDM database according to the MAX_STAT_DAYS and MAX_EVENT_DAYS settings in the .configrc file. For more information, see Archive Manager Resources.

The default setting is 45 days. Once any records attain the specified age, the purge occurs daily thereafter. As a result, the oldest day's data is purged daily.

If you use the ddm_save method, you must restart Archive Manager using the Control Panel once the backup is completed.

Database Restoration

In the event of a hardware or operating-system failure, restore the DDM database by loading a save file that was created using any of the supported backup methods:post_olb_script, ddm_backup, or ddm_ save.

Save files are loaded using the ddm_load tool, which is described in the following section.

ddm_load

This utility, which is located in the *$SPECROOT*/SS/DDM directory, lets you restore the DDM database. First, it initializes (removes all data from) the DDM directory, and then it loads the save file that you specify. Initialization occurs automatically if you are loading a save file or directory. The -initialize option is only required if you want to clean out the DDM directory and start from nothing. With that option, the landscape handle for the associated SpectroSERVER is also required.

> **WARNING**
>
> The ddm_load binary coming with the previous releases has been deprecated in 10.4.1. From 10.4.1 onward, use ddm_load.pl.

Before you execute the command, shut down the Archive Manager from the Control Panel. Then restart it when the restoration has completed.

This command has the following format:

```
ddm_load [-quiet] [-initialize <LANDSCAPE_HANDLE>] [-events_init]
[-stats_init][<SAVE_FILE>]
```

Where the first letter of an argument name appears in bold type, you can use the letter only, rather than entering the full string.

- **-quiet**
  Disables prompts (interactive mode). Useful for running load commands from within a script.
- **-initialize**
  Removes all data from the DDM database. Used only with the LANDSCAPE_HANDLE argument.
- **<LANDSCAPE_HANDLE>**
  Specifies the landscape handle of the SpectroSERVER whose database is being initialized.
- **-events_init**
  Initializes (removes) *only* the event records in the DDM database.
- **-stats_init**
  Initializes (removes) *only* the statistical records in the DDM database.
- **<SAVE_FILE>**
  Specifies the name of the backup file to load. The ddm_load utility supports both the new gzipped tar file format and the legacy, proprietary file format.

Support for InnoDB Database

The 10.3 release supports importing the InnoDB database files into Archive Manager database. The importing procedure requires a temporary directory to uncompress the InnoDB database files.

> **NOTE**
>
> The space required to import the InnoDB database files is two times greater than the event table size.

The DDM load supports importing both MyISAM and InnoDB database files based on the current running database engine (MyISAM or InnoDB). For example, if the Archive Manager database is running on the MyISAM database, you can import only the MyISAM database files.

Convert Saved MyISAM Database Files to InnoDB Database Files

From the 10.3 release, you can convert your saved Archive Manager database file from MyISAM to InnoDB.

Prerequisite

- Ensure that there is enough free disk space. The space required is three times greater than the event table size.

Follow these:

1. Open any bash-shell and navigate to $SPECROOT/SS/DDM/scripts.
2. Run the following script as **Spectrum installation user**:
   ```
   convert_saved_myisam_to_innodb.pl
   ```
   For example, when you run the script for the tgz file, you get the following message:

```
./convert_myisam_to_innodb.pl -f db.tgz

./convert_saved_myisam_to_innodb.pl:

This utility converts the saved MyISAM database file to InnoDB database

-f: path to the dump file

-h: print this message

Example: ./convert_myisam_to_innodb.pl -f db.tgz

xxxx03-w2k8vm2%/d/builds/Spectrum/Armgr.a/buildfiles > ./
convert_saved_myisam_to_innodb.pl -f /c/db.tgz
```

Loading the DDM Db

All MySQL Enterprise Backup functions are executed with the **mysqlbackup** client. It is used for performing different types of backup and restore operations, as well as other related tasks like backup compression, decompression and validation. With 10.3.1, ensure improved performance with the MySQL backup and save the DDM Db.

1. To load the DDM db, ensure you are signed in as a spectrum user to run the following script:

```
./ddm_load.pl  -f  <filename.mbi>
```

The **OptionsArguments** are as follows:

- **-quiet**
  Disables prompts (interactive mode). Useful for running load commands from within a script.
- **-initialize:**
  Initializes the database. Removes all data from the DDM database. Used only with the LANDSCAPE_HANDLE argument
- **-landscape <LANDSCAPE_HANDLE>**

Specifies the landscape handle of the SpectroSERVER whose database is being initialized.

- **-events_init**
  Initializes the events database only. Initializes (removes) *only* the event records in the DDM database.
- **-stats_init**
  Initializes only the statistical database records in the DDM database.
- **-f <SAVE_FILE>**
  Specifies the name of the backup file to load.

Points To Remember About Restorations

To keep the databases synchronized, the DDM database must be restored each time that the SpectroSERVER database is restored.

Some of the records in a given save file are likely to have reached their purge date by the time you reload them. As a result, they are purged as soon as the Archive Manager is restarted.

Restart the Archive Manager when the restoration has completed.

Database Maintenance and Optimization

The section describes the two scripts that you can run to keep the DDM database running efficiently.

db_maintenance.pl

Located in the *$SPECROOT*/SS/DDM/scripts directory, this script removes all unreferenced statistical records from the DDM database. Unreferenced records are typically created when models are deleted. Over time, their associated records are purged, leaving some unnecessary remnants in the database.

> **WARNING**
> The db_maintenance.pl script can only be run with the Archive Manager down. Carefully weigh the potential benefit of running this script against the drawbacks of taking the Archive Manager down.

Do not schedule this script to run regularly. Run it occasionally, at times when an extremely large number of models have been deleted from a landscape. The performance gains that are realized by running this script are not noticeable. Run this script only for general housekeeping if a great many models have been destroyed (such as an entire landscape).

We recommend running the db_optimize.pl script before running this script. In addition, we recommend running this script at a time that minimizes its impact on production environments. The script can run very slowly on a large, unoptimized database or on a database with many event records.

The script has the following format:

```
./db_maintenance.pl[-q]
```

- **-q**
  (Quiet) Disables prompting (interactive mode).

db_optimize.pl

This script is located in the *$SPECROOT*/SS/DDM/scripts directory. It provides an easy way to optimize all of the tables in the DDM database.

Optimizing tables has two major benefits:

- The speed of queries that are sent to the DDM database is increased.
- Disk space that is available from purged records is recovered.

You can schedule this script to run weekly or monthly, depending on your performance requirements. Keep in mind that this script can take a considerable amount of time to run, depending on the size and the level of fragmentation of the DDM database.

> **NOTE**
> The db_optimize.pl script requires free disk space that exceeds the size of the largest file in the *$SPECROOT*/mysql/data/ddmdb directory. We recommend backing up your database file before running this script.

The script may be run with the Archive Manager up. However, it creates a 10-minute delay between table optimizations to allow the Archive Manager to recover from the database being locked.

This script has the following format:

```
./db_optimize.pl[-q]
```

- **-q**
  (Quiet) Disables prompting (interactive mode).

DDM Database Queries

The queries provided in this section can help you determine what events are being generated and what models are generating these events. We have provided queries that can help you determine whether a specific device is generating excessive events or whether a change is required to the number of days that statistical data is stored. These queries can take a while to run, depending on the size of the DDM database event table.

> **NOTE**
> For information on setting the MAX_STAT_DAYS parameter value, which controls the number of days that data is stored, see Archive Manager Resources.

To log in to MySQL:

At a command prompt, enter the following command in *$SPECROOT*/mysql/bin:

```
mysql -uroot -proot ddmdb
```

On UNIX, also pass in the following argument:

```
-S $SPECROOT/mysql/tmp/mysql.sock
```

The mysql> prompt appears.

To query the DDM (ddmdb) database:

At the mysql> prompt, enter any of the following queries:

- To display table statistics:
  ```
  SHOW TABLE STATUS LIKE "event";
  ```
- To get a count of the total number of events:
  ```
  SELECT COUNT(*) FROM event;
  ```
- To get a count of the number of events that occurred after a particular date:
  ```
  SELECT COUNT(*)
    FROM event
    WHERE utime >= UNIX_TIMESTAMP("yyyy-mm-dd");
  ```
- To get the top ten events most commonly generated:
  ```
  SELECT HEX(type), COUNT(*) AS cnt
    FROM event
    GROUP BY type
    ORDER BY cnt DESC
  ```

```
  LIMIT 10;
```

- To get the top ten models with the most events:

```
SELECT HEX(e.model_h), m.model_name, COUNT(*) AS cnt
  FROM event e, model m
  WHERE e.model_h=m.model_h
  GROUP BY e.model_h
  ORDER BY cnt DESC
  LIMIT 10;
```

- To get the top ten high-volume days for events:

```
SELECT DATE(FROM_UNIXTIME(UTIME)) AS x, COUNT(*) AS cnt
  FROM event
  GROUP BY x
  ORDER BY cnt DESC
  LIMIT 10;
```

- To get the last ten days of volume of events:

```
SELECT date(from_unixtime(utime)) AS x, COUNT(*) AS cnt
  FROM event
  GROUP BY x
  ORDER BY x DESC
  LIMIT 10;
```

Other Database Utilities

DX NetOps Spectrum provides three MySQL database utilities, all of which are installed in the *$SPECROOT*/mysql/bin directory.

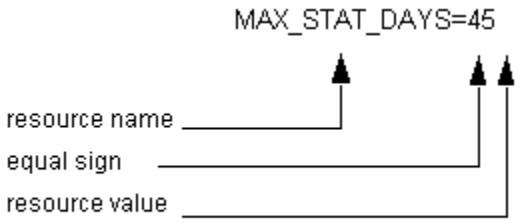
- myisamchk lets you check and repair MyISAM tables while the MySQL server is stopped.
- mysqlcheck is similar to myisamchk. However, it does not require you to stop the server to check or repair tables. This utility also optimizes and analyzes tables.
- mysqladmin enables you to perform administrative operations, such as checking the server configuration or dropping a database.

NOTE
For more information on using these MySQL utilities, see the documentation provided at http://www.mysql.com.

Archive Manager Resources

Archive Manager resources are settings that control how historical records are processed. These resources also enable the Archive Manager to communicate with the SpectroSERVER and DDM databases.

Archive Manager resources are defined in the .configrc file, which is located in the *$SPECROOT*/SS/DDM directory. The resources are listed in the form “resource = resource_value”, as shown here.



The .configrc file defines the following Archive Manager resources:

- **ARCH_MGR_SOCKET_NUMBER**
  Identifies the port where the Archive Manager listens for requests from the VNM and SSAPI clients.
  Default: 0xbafe
- **AUTO_REPAIR_DB**
  Controls whether the Archive Manager automatically attempts to repair a corrupt DDM database. By default, this parameter is not listed in the .configrc file and, therefore, it is not enabled. To enable the auto-repair functionality, add AUTO_REPAIR_DB=TRUE to the .configrc file. To disable the auto-repair functionality, set the value to FALSE or remove the entry from the .configrc file.
  Default: Disabled
- **DDM_DATABASE_NAME**
  Required for communications between the Archive Manager and the DDM database. The default value is read in automatically when DX NetOps Spectrum is installed. Changing this value is not recommended.
  Default: ddmdb
- **DDM_DATABASE_PORT**
  Specifies the port used to connect to the MySQL database server.
- **DDM_DATABASE_HOSTNAME**
  Required for communications between the Archive Manager and the DDM database. The default value is set automatically when DX NetOps Spectrum is installed. Changing this value is not recommended.
  Default: local host
- **DDM_DATABASE_PASSWORD**
  Required for communications between the Archive Manager and the DDM database. The default value is read in automatically during installation. Changing this value is not recommended.
  Default: DX NetOps Spectrum Installation Owner user name password
- **DDM_DATABASE_USERNAME**
  Required for communications between the Archive Manager and the DDM database. The default value is read in automatically during installation. Changing this value is not recommended.
  Default: DX NetOps Spectrum Installation Owner user name
- **DDM_SOCKET_NUMBER**
  Deprecated. This resource will not appear in future releases.
- **LANDSCAPE_PRECEDENCE**
  Currently unsupported resource.
- **MAX_DB_CONNECTIONS**
  Specifies the maximum number of simultaneous MySQL connections that the Archive Manager can use to service requests.
  Default: 25
- **MAX_EVENT_DAYS**
  Specifies the maximum number of days that event data is stored. When MAX_EVENT_DAYS is exceeded, the older data is purged.
  Default: 45
- **MAX_STAT_DAYS**
  Specifies the maximum number of days that statistical data is stored. When MAX_STAT_DAYS is exceeded, the older data is purged.
  Default: 45
- **TIME_TOLERANCE_IN_SECONDS**
  Specifies the maximum allowable variance in seconds between the timestamp that is reported for logged data and the actual logging time. The higher the time tolerance value, the more effectively data can be compressed, conserving CPU and disk resources. However, accuracy is sacrificed as this value increases.
  Default: 300

DDM and Archive Manager Troubleshooting

This section identifies probable causes and solutions for problems that are associated with the Archive Manager and the DDM database.

Error Messages in ARCHMGR.OUT

Symptom

I saw the following message:

```
<date/time> ERROR at ArchMgr.cc(591): Failed to open connection to SpectroSERVER at <hostname>, 0xbeef. The
 error indicates that the SpectroSERVER is down. Will try again in 60 seconds.
```

Solution

The SpectroSERVER is not ready. The Archive Manager automatically starts when the SpectroSERVER is started but cannot connect until the SpectroSERVER is ready. This message always appears on initial startup and anytime that the SpectroSERVER goes down.

Symptom

The following message appears once the Archive Manager has connected or reconnected:

<*date/time*> : ArchMgr has successfully registered with DDM name service - <*ip_address*> (<machine_name>).

Symptom

When I attempted to start the Archive Manager, I saw the following message:

<*date/time*> : ArchMgr is shutting down...

Solution

The Archive Manager has been shut down because the SpectroSERVER does not contain a DX NetOps Spectrum User model for the user who attempted to start it.

Symptom

I saw the following message:

<*date/time*> : ArchMgr has successfully shut down.

If this message persists, check the Users tab in the OneClick Console to verify that a user exists for the owner of the ArchMgr process.

Symptom

I saw the following message:

```
<date/time> : ArchMgr started as user '<user_name>'
<date/time> : ArchMgr validating database
Database corruption detected:
   ddmdb.statistic_ul64 - record delete-link-chain corrupted
   ddmdb.statistic_ul64 - Corrupt
Error opening the DDM database. One or more tables are corrupt.
Recovery options:
1. Run `ArchMgr -repair` to attempt recovery
2. Load a valid DDM savefile using `ddm_load <savefile>`
3. Initialize the DDM database using `ddm_load -i <LANDSCAPE_HANDLE>`
<date/time> : ArchMgr invalid database error.
```

Solution

The DDM database has been corrupted and cannot be loaded. You can use the ddm_load command to load a saved DDM database or initialize the DDM database. You can also attempt to repair the existing database file using the ArchMgr -repair command.

To run the ArchMgr -repair command, navigate to the *$SPECROOT*/SS/DDM directory and enter:

./ArchMgr -repair

Once the repair has successfully completed, the following message appears:

“ArchMgr successfully repaired database.”

To enable the Archive Manager to automatically attempt to repair a corrupted DDM database, use the auto_repair_db option in the .configrc file. For more information, see Archive Manager Resources .

No Events Are Listed

Symptom

No events are listed in on the Events tab and the Servers (Connection Status) button is flashing yellow. When you double-click the button, the Connection Status dialog shows a status of “Down” for the Events and View services, but “Up” for the Landscape service.

Solution

The Archive Manager is not running or is not connected to the SpectroSERVER. Check the Control menu in the Control Panel to verify that the Archive Manager service has been started. If it has been started, check the *$SPECROOT*/SS/DDM/ARCHMGR.OUT file for Archive Manager errors.

Distributed SpectroSERVER Administration

This section details how to administer and manage DX NetOps Spectrum and databases in a Distributed SpectroSERVER environment.

Introducing Distributed SpectroSERVER

This section introduces the Distributed SpectroSERVER environment which allows you to create a unified representation of the infrastructure that is composed of multiple landscapes, each with a local SpectroSERVER.

In a Distributed SpectroSERVER environment, a SpectroSERVER client, such as the OneClick Console, can simultaneously access information from multiple SpectroSERVERs.

About Distributed SpectroSERVER

Distributed SpectroSERVER (DSS) is a powerful modeling feature that enables the distribution of management over a large-scale infrastructure. The infrastructure can be organized geographically or across multiple servers in a single physical location. DSS can improve DX NetOps Spectrum performance when managing a computing infrastructure. Performance improvements can come from distributing the network load from management traffic and delegating tasks to remote workstations.

Using DSS, you can create a unified representation of the infrastructure that is composed of multiple landscapes, each with a local SpectroSERVER. In a DSS environment, a SpectroSERVER client, such as the OneClick Console, can simultaneously access information from multiple SpectroSERVERs.

DSS includes the following features:

- **Distributed Visibility**
  The OneClick console displays information for all deployed SpectroSERVERs. Summary alarm counts are displayed for each SpectroSERVER. The network manager can quickly locate trouble areas.
- **Localized Polling Traffic**
  The management workstation that is polling the network is geographically closer to the devices that it manages with DSS. This setup reduces traffic on wide area links and avoids congestion on the local network. Multiple, smaller SpectroSERVERs generate less traffic than a single SpectroSERVER that polls devices from a great distance.
- **Scalability**
  As a network expands, it is often less expensive to use low-end workstations as additional, dedicated SpectroSERVER servers. Such a solution is preferable to upgrading one workstation to accommodate one SpectroSERVER in a large, non-distributed environment.
- **Fault Tolerance**
  DSS supports fault tolerance between SpectroSERVERs. A secondary SpectroSERVER can be provided as a fault tolerant backup or standby for a primary SpectroSERVER. Network management continues if the workstation running the primary SpectroSERVER fails.
  You can protect against failures by simply reloading the database for a given landscape onto a secondary SpectroSERVER. When a failure occurs that disables the primary SpectroSERVER, the secondary SpectroSERVER automatically takes its place. All DX NetOps Spectrum applications automatically use the secondary SpectroSERVER.

Landscapes

A network domain that a single SpectroSERVER manages is known as a *landscape*. A landscape consists of the models, associations, attribute values, alarms, events, and statistics that belong to a specific SpectroSERVER. Each landscape in a network is unique. A unique landscape handle (ID) identifies each landscape.

In OneClick, a landscape icon represents each landscape. The landscape icon represents a SpectroSERVER database. Through double-click zones and menu selections, the landscape icon lets you access remote network models. The icon also presents a rollup of alarm information for the devices that are modeled in those remote databases.

> **NOTE**
> The terms *local* and *remote* are used to designate landscapes from the perspective of a particular SpectroSERVER.

Landscape Maps

DX NetOps Spectrum automatically maintains a "map" of all of the landscapes that comprise a particular DSS environment. Changes that occur in the SpectroSERVER topology initiate an internal discovery mechanism that updates the landscape maps of all of the other SpectroSERVERs in that environment. For example, discovery detects a SpectroSERVER that is added or removed from the network.

A landscape model is a container model. Landscape models let you connect to other SpectroSERVERs in the landscape map through the DX NetOps Spectrum user interface. You can create a landscape model at any level of the three view hierarchies: Topology, Location, or Organization.

Modeling Catalogs

A modeling catalog is a set of templates (such as model types or relations) that are installed on a particular SpectroSERVER. These templates are used to create models. The models that are created through these templates compose the landscape of that SpectroSERVER.

> **NOTE**
> OneClick queries the default landscape for a list of available model types at startup.

When you model multiple landscapes using DSS, each landscape must contain identical modeling catalogs. All model types in the modeling catalog of one landscape must also exist in the modeling catalog of every other landscape to which it connects. If you install a new management module on one SpectroSERVER, install the same management module on every other SpectroSERVER in the landscape map.

User Models

A user model contains information about individual user permissions and other user data. This information is stored in a landscape. When you first install DX NetOps Spectrum, a default user model represents the user that you specified in the Installation Owner field during DX NetOps Spectrum installation.

In a DSS environment, the same user model must exist in all landscapes to enable users to connect to remote SpectroSERVERs in the landscape map and to enable the landscapes to communicate with each other. Adding or modifying a user model causes the SpectroSERVER associated with the landscape to query all other SpectroSERVERs in the landscape map. The query checks for the user model in other landscapes. If the user model is found in other landscapes, their user models are automatically updated to reflect any changes to the initial user model.

NOTE
Deleting a user model only deletes the user model in that landscape. You must manually delete duplicate user models on other SpectroSERVERs in the landscape map.

SpectroSERVER (.vnmrc) Resources

SpectroSERVER resources are defined in the <*$SPECROOT*>/SPECTRUM/SS/.vnmrc file. Many default values of these resource files are built into the code and remain blank. However, the coded default values are not used when values are specified in these resources.

Resources in the .vnmrc (Virtual Network Machine runtime configuration) file define path names and default settings for the SpectroSERVER. The DX NetOps Spectrum system software includes a runtime configuration file with default settings for SpectroSERVER resources.

You can modify resources to make the following changes:

- Define general SpectroSERVER resources and directory paths
- Adjust events archiving
- Specify name service variables
- Adjust thread allocation
- Control fault tolerant alarm synchronization

All of these resource entries are listed in the format “resource = resource_value”. For many of the resources in the .vnmrc file, the resource value is blank.

If no value appears with a specific resource, DX NetOps Spectrum uses the default value. The resources in the .vnmrc file take effect when SpectroSERVER is started. If you change this file while SpectroSERVER is running, the changes take effect when SpectroSERVER is restarted.

Example: Define a .vnmrc resource

This example sets the maximum number of records logged in the Event Log database:

```
max_event_records=5000
```

- **max_event_records**
  Is the resource name.
- **5000**
  Is the resource value.

General SpectroSERVER (.vnmrc) Resources

The general SpectroSERVER (.vnmrc) resources control many SpectroSERVER functions. The following list describes the general SpectroSERVER (.vnmrc) resources:

- **comm_port**
  Specifies the TCP port that client user interfaces use to communicate with SpectroSERVER. The port number is defined during installation.
  This command has the following format:
  ```
  comm_port=0xBEEF
  ```
  – ***0xBEEF***
    Is the default TCP connection port socket. This socket can be any valid TCP port that is greater than the port number that is assigned to the IPPORT_USERRESERVED parameter in the file /netinet/in.h. However, the port must be less than 65535 (0xFFFF).
- **snmp_trap_port_enabled**
  Binds the SpectroSERVER to the SNMP trap port and listen for traps. When set to False, the SpectroSERVER does not bind to the SNMP trap port.
  This command has the following format:
  ```
  snmp_trap_port_enabled=TRUE
  ```
  Default: True
- **vnm_file_path**
  Specifies the root subdirectory that contains SpectroSERVER external files, such as database files.
  This command has the following format:
  ```
  vnm_file_path=<directory>/spectrum/SS/CsVendor
  ```
  – ***/spectrum/SS/CsVendor***
    Is the file path for the root subdirectory
- **tcp_buffer_size**
  Lets you alter the buffer size that is appropriate for traffic on your LAN. A large buffer size improves throughput. The resource accepts a numeric value that represents the number of bytes for the TCP buffer. This command has the following format:
  ```
  tcp_buffer_size=<blank>
  ```
  Default: blank
  Limits: 8192 to 65536 bytes
  Note: Values below 8 KB are rounded up to 8 KB. Values above 64 KB are set to 64 KB.
- **resource_file_path**
  Is the file path for VNM resource files, such as Ether Map.
  This command has the following format:
  ```
  resource_file_path=./CsResource
  ```
  – **./CsResource**
    (Optional) Is the file path for VNM resource files. This parameter can be left blank.
- **wait_active**
  Determines whether the server accepts connections as soon as all models are loaded or waits until all models are active. If set to *Yes,* a Control Panel message displays a running percentage of models that were activated during SpectroSERVER startup.
  This command has the following format:
  ```
  wait_active=no
  ```
  Default: no.
- **max_bind_retry_count**
  Specifies the maximum number of listen socket bind retries.
  This command has the following format:
  ```
  max_bind_retry_count=50
  ```

Default: 50.

- **bind_retry_interval**
  Specifies the delay in seconds between listen socket bind retries.
  This command has the following format:
  ```
  bind_retry_interval=30
  ```
  Default: 30 seconds.
- **min_client_version**
  Is the minimum parm block version that the SpectroSERVER accepts for client connections.
  This command has the following format:
  ```
  min_client_version=0
  ```
  Default: 0 minimum client connections.
- **max_connections**
  Is the maximum number of connections the SpectroSERVER accepts. Connections can be from clients or from other servers, such as another SpectroSERVER, Archive Manager, or Location Server (main and local).
  This command has the following format:
  ```
  max_connections=50 gensv000316 docs002958
  ```
  Default: 50 connections
  A connection can fail when the maximum number of client connections is exceeded. When you increase the maximum SpectroSERVER connections, an increase in the number of open file descriptors on your workstation is sometimes required for improved performance.
- **handshake_timeout**
  Sets the time in seconds for exchanging initial ID information during a connection handshake.
  This command has the following format:
  ```
  handshake_timeout=40
  ```
  Default: 40 seconds
- **vnm_message_timeout**
  The vnm_message_timeout resource sets the timeout in seconds for messages that are sent between VNMs.
  This command has the following format:
  ```
  vnm_message_timeout=180
  ```
  Default: 180 seconds
- **vnm_close_timeout**
  The vnm_close_timeout resource sets the timeout in seconds for closing the connection between VNMs.
  This command has the following format:
  ```
  vnm_close_timeout=180
  ```
  Default: 180 seconds
- **connect_time_limit**
  Sets the timeout in milliseconds for connection to the VNM.
  This command has the following format:
  ```
  connect_time_limit=5000
  ```
  Default: 5000 milliseconds
- **rcpd_comm_port**
  Specifies the listening port of the Remote Copy Process Daemon (rcpd). This resource is used for fault-tolerant database backups.
  This command has the following format:
  ```
  rcpd_comm_port=0xCAFE
  ```
  Default: 0xCAFE
- **procd_comm_port**
  Specifies the port that the VNM uses to connect to the process daemon.
  This command has the following format:

```
procd_comm_port=0xFEEB
```

Default: 0xFEEB

- **snmp_trap_port**
  Specifies the port through which the VNM receives traps. This command has the following format:

  ```
  snmp_trap_port=162
  ```

  NOTE

  To change the SNMP Trap Port SPECTRUM uses to listen for trapsEdit the $SPECROOT/SS/.vnmrc file:

  1. Set the "snmp_trap_port=" to a value you wish to have Spectrum listen to for incoming traps.
  2. Then, you will need to restart the SpectroSERVER for the change to take effect.
  3. The default value is **162.**

- **enable_traps_for_pingables**
  Specifies whether DX NetOps Spectrum can receive SNMP traps on pingable models. By default, the .vnmrc file does not contain this resource. Add it to the file manually, using the following syntax:

  ```
  enable_traps_for_pingables = TRUE
  ```

  Default: TRUE

- **max_device**
  Applies to DX NetOps Spectrum products whose device_limit resource is in effect. By default, DX NetOps Spectrum generates a yellow alarm when the number of models reaches 80% of the specified device limit. DX NetOps Spectrum generates a red alarm when the number of models reaches 100% of the specified device limit. This resource lets you set a value that falls below the actual device limit, which gives you an earlier warning.
  This command has the following format:

  ```
  max_devices=<# of device models>
  ```

  – ***# of device models***
    Refers to models with IP addresses that are derived from the device model type AND whose specified number falls below the value of the device_limit resource. The default value is blank.

- **disable_redundancy_when_using_loopback**
  Disables redundancy (0x11d2c) when modeling a device by loopback. This parameter is only applicable when the use_loopback (0x12bb) attribute of the VNM model is set to True.
  This command has the following format:

  ```
  disable_redundancy_when_using_loopback=False
  ```

  By default, this resource does not appear in the .vnmrc file and is therefore automatically evaluated as False.

  NOTE
  For more information, see the Modeling and Managing Your IT Infrastructure section.

- **persistent_alarms_active**
  Prevents DX NetOps Spectrum from retaining alarm-related information when the SpectroSERVER shuts down.
  This command has the following format:

  ```
  persistent_alarms_active=<False>
  ```

  By default, this resource does not appear in the .vnmrc file. DX NetOps Spectrum automatically retains alarm-related information (such as troubleshooter assignments or status) when the SpectroSERVER is shut down and is brought back up. The alarms that were present before shutdown are preserved. These alarms are considered "persistent" alarms. Adding persistent_alarms_active=False to the .vnmrc file prevents DX NetOps Spectrum from retaining alarm-related information when the SpectroSERVER shuts down.

NOTE
We strongly recommend against adding the persistent_alarms_active resource unless you integrate a third-party application that retains alarm-related information.

Adding this resource has the following effects:
- Alarm status additions are lost (only recorded as past events)
- Existing alarms on SpectroSERVER shutdown are regenerated at startup with new timestamps.
- Regenerated alarms are forwarded to alarm notification tools.

- **unsupported_attr_poll_interval**
  Specifies the amount of time that DX NetOps Spectrum waits to poll an external attribute that has returned a "noSuchName" error. If this parameter is not specified in the .vnmrc file, the default value (12 hours) is used.
  This command has the following format:
  ```
  unsupported_attr_poll_interval=43200
  ```
  Default: 43200 seconds (12 hours)

- **snmpv3_engine_id=**
  Specifies the SNMPv3 local engine ID of a SpectroSERVER. This is a static value (defaults to MAC address) and you can set it for the devices that do not support engine ID discovery. When the 'snmpv3_engine_id=' value is set the engine id will not change when the SpectroSERVER is stopped and restarted.

Events Archive (.vnmrc) Resources

The events archive (.vnmrc) resources control the process of the SpectroSERVER sending events to the Archive Manager for logging. The following list describes the events archive (.vnmrc) resources:

- **max_event_records**
  Specifies the maximum number of records that can be stored in the Event Log database.
  This command has the following format:
  ```
  max_event_records=# of Records
  ```
  - ***# of Record***
    Is the number of Event Log records. The minimum value is equal to the event_record_increment value. System storage capacity limits the maximum value.
    Default: 20,000
- **event_record_increment**
  Specifies the number of records to delete from the Event Log database when the number of records exceeds the max_event_records value.
  This command has the following format:
  ```
  event_record_increment=# of Records
  ```
  - ***# of Records***
    Is the number of Event Log records that are deleted from the database. The minimum value is 100.
- **event_batch_max_size**
  Sets the maximum number of events per batch. If the batch becomes full, the events batch is immediately sent to the Archive Manager.
  This command has the following format:
  ```
  event_batch_max_size=1000
  ```
  Default: 1000 (the maximum number of events)
- **event_batch_timeout**
  Determines the amount of time in seconds when the events batch is sent over to the Archive Manager.
  This command has the following format:
  ```
  event_batch_timeout=1
  ```

Default: *1* second

- **log_user_events**
  Controls whether an event is generated for each user-initiated write to a model attribute value. A value of True causes the VNM to generate events.
  This command has the following format:
  log_user_events=*False*
  Default: False
- **use_log_queue**
  Places events in a separate queue when they are generated. These events are sent to the Archive Manager on a separate thread. If set to “TRUE”, the event is sent to the Archive Manager on the same thread on which it was generated. This situation can delay alarm creation.
  This command has the following format:

  ```
  use_log_queue=<blank>
  ```

  Default: <*blank*>

Work Thread (.vnmrc) Resources

SpectroSERVER is a multithreaded process. During normal operation, each subsystem allocates numerous work threads.

SpectroSERVER maintains a pool of work threads that are reused over time. Subsystems use threads from the pool, up to their individual limits, during periods of increased activity. When the common pool of work threads is exhausted, new work threads are created. The pool grows to accommodate the increased activity.

Work threads that subsystems no longer require return to the common pool for later use. Threads are taken from the pool to serve subsequent needs or new threads are allocated if the pool is empty. Threads that remain unused for a specified time are removed from the pool, and their resources are returned to the system. This process is termed aging.

The following list describes the work thread (.vnmrc) resources:

- **max_total_work_threads**
  Specifies the maximum number of work threads that can be allocated for all SpectroSERVER subsystems. Each work thread consumes processing resources and requires a significant block of memory. Set its value based on system capacity (memory size and speed).
  When the value of this resource is too high, SpectroSERVER can periodically run out of memory. When the value of this resource is too low, SpectroSERVER operations can become sluggish.
  If the value of this resource is newly set or changed, DX NetOps Spectrum reads the new value when the SpectroSERVER is restarted. Once the value is read, it is used to update the value of the WorkThreadsMaxAvail attribute on the VNM model. When the value of WorkThreadsMaxAvail has been updated, the value for max_total_work_threads is removed from the .vnmrc file.
  This command has the following format:

  ```
  max_total_work_threads=# of threads
  ```

  The default value of the VNM WorkThreadsMaxAvail attribute is 500.
- **work_thread_age**
  Work threads that a subsystem no longer requires are returned to the work thread pool. This resource specifies how long (in seconds) a work thread can remain in the pool without being used.
  Set this resource to a value that is compatible with subsystem activity. Significant processing overhead is required to create a work thread. Setting the value of this resource too low taxes system resources by creating new threads too often in response to work thread demands. Setting the value too high means that resources remain committed unnecessarily.
  This command has the following format:

  ```
  work_thread_age =seconds
  ```

  Default: 60 seconds

Fault Tolerant Alarm Service (.vnmrc) Resources

In a fault-tolerant environment, alarms must be synchronized between the primary and secondary SpectroSERVERs. The servers connect to exchange alarm information. If the initial connection attempt fails, subsequent attempts can be made. The Fault Tolerant Alarm Service controls alarm synchronization using the following settings:

- **ftasv_enabled**
  Enables alarm synchronization. Include this command in the .vnmrc file for both the primary and secondary servers. This command has the following format:
  ```
  ftasv_enabled=true
  ```
  Default: True

  WARNING
  You cannot have ftasv_enabled set to True for one server and set to False for the other server. Use the same value for both servers.
- **ftasv_max_conn_retry_count**
  Specifies the number of times that the primary server attempts to connect to the secondary server for synchronization after a connection attempt has failed. This parameter is required in the .vnmrc file of the primary SpectroSERVER.

  NOTE
  The primary server uses this parameter when attempting to connect to the secondary server. It is not used when the secondary server attempts to connect to the primary.

  This command has the following format:
  ```
  ftasv_max_conn_retry_count=# of retries
  ```
  Default: Four retries (for a total of five synchronization attempts with the secondary server).
- **ftasv_conn_retry_interval**
  Specifies the number of seconds between primary server attempts to synchronize with the secondary server. Use this parameter in the .vnmrc file of the primary SpectroSERVER.

  NOTE
  The primary server uses this parameter when attempting to connect to the secondary server. This parameter is not used when the secondary server attempts to connect to the primary.

  This command has the following format:
  ```
  ftasv_conn_retry_interval=# of seconds
  ```
  Default: 30 seconds
- **ftasv_debug**
  Enables debug output for alarm synchronization activity. Include this command in the .vnmrc file for both the primary and secondary servers. Debug output is written to the VNM.OUT file of each server. Each message begins with "Fault Tolerant Alarm Service."
  This command has the following format:
  ```
  ftasv_debug=true
  ```
  Default: False

  WARNING
  If the secondary SpectroSERVER is not running when the primary server attempts to connect to it, the primary exhausts its synchronization attempts. As a result, startup of the primary SpectroSERVER is delayed. When using default settings for ftasv_max_conn_retry_count and ftasv_conn_retry_interval, the delay is two minutes (four retries with 30 seconds intervening). This delay is unavoidable because it occurs before model activation. The workaround is to verify that the secondary SpectroSERVER is running when the primary starts. You can also decrease the retry count or the interval size to reduce the potential delay.

snmpv3_engine_id=

How to Pack Up Product Utilities and Move Them to Another Computer

The packtool.pl script packs up the DX NetOps Spectrum Command Line Interface (CLI), [assign the value for sslog in your book], AlarmNotifier, the modelinggateway tool, and the sbgwimport tool so that you can transfer them to another computer. The script retains the relative directory structure of the utilities and their support files when the files are moved.

Note: Both computers must be running the same operating system.

Consider the following requirements before running the packtool.pl script:

- You must be a root user on Linux platforms, or a user with administrative privileges on Windows platforms to run the packtool.pl script.
- The following requirements apply when you transfer the DX NetOps Spectrum utilities to a computer where only the OneClick server is installed:
  - The version and patch level of the OneClick server on each computer must match. After each patch installation, run the packtool.pl script again, and transfer the utilities again.
  - The installation user on the computer where the utilities are transferred must match the installation user on the computer where the files are packaged.
  - If a debug patch or PTF is installed on the target computer, reinstall the debug patch or PTF after the file transfer.
- Stop all DX NetOps Spectrum processes on the computer where the utilities are packaged.
- Stop all DX NetOps Spectrum processes on the computer where you are transferring the utilities before you extract the tools_bundle file.
- To extract the tools_bundle file, you must be a root user on Linux platforms, or a user with administrative privileges on Windows.

  WARNING
  To run the DX NetOps Spectrum utilities, you must be logged on to the target computer as a valid DX NetOps Spectrum user or an error occurs. For example, if you are logged on to a Linux workstation as “root,” and there is no matching user model in DX NetOps Spectrum, you receive an error stating, “NO_USER.”

To package DX NetOps Spectrum utilities and transfer them to another computer, take the following steps:

1. Pack Up the SPECTRUM Utilities using the packtool.pl script.
2. Extract the tools_bundle file on another computer.

Pack Up DX NetOps Spectrum Utilities

Before moving the CLI, SSLogger, AlarmNotifier, the modelinggateway tool, and the sbgwimport tool to another computer, run a script to bundle them.

Follow these steps:

1. Verify that the environmental variable <*$SPECROOT*> is set to the DX NetOps Spectrum installation directory path on the computer where you are packing up the utilities.
2. Set the CYGWIN32 environment variable to the directory where the cygwin is installed.
3. Run the packtool.pl script, which packs up the utilities and their support files. The packtool.pl script can be found in the <*$SPECROOT*>/SS-Tools directory.
   To run the script from a Bash shell or other UNIX shell, enter the following command:
   ```
   <$SPECROOT>/SS-Tools/packtool.pl [-no_notifier | -no_event_alarms][-f file_name]
   ```
   - **<*$SPECROOT*>**
     Is the directory structure where DX NetOps Spectrum is installed on your SpectroSERVER.
   - **-no_notifier**
     Specifies that you do not want to pack up AlarmNotifier.
   - **-no_event_alarms**

Specifies that you do not want to pack up AlarmNotifier EvFormat or PCause files.

– ***-f file_name***
Specifies a name for the executable file.

An executable file named linux_tools_bundle (Linux), or nt_tools_bundle.exe (Windows) that contains the DX NetOps Spectrum utilities and their support files is created.

Move DX NetOps Spectrum Utilities to Another Computer

You can move DX NetOps Spectrum utilities to a computer that is only running the OneClick server.

Follow these steps:

1. Pack up the DX NetOps Spectrum utilities.
2. On the computer where you want to extract the DX NetOps Spectrum utilities, change the directory to <*$SPECROOT*>.
3. Using binary mode, FTP the executable file named linux_tools_bundle (Linux), or nt_tools_bundle.exe (Windows) to the current directory.
4. Extract the tools_bundle file from the DOS, Bash, or other UNIX shell.
The DX NetOps Spectrum utilities and their support files unpack into the appropriate directory structure.
5. If you transferred AlarmNotifier and event and pcause files, verify that the paths in <*$SPECROOT*>/SG-Support/CsResource/preferences/*.prf have the correct locations for the event and pcause files. Also, verify the ui= and the lhandle= options.
6. Set the <*$SPECPATH*> environmental variable to the path of the directory where you extracted the tools_bundle file:
   – On Windows, create the system environment variable <*$SPECPATH*> with the value in variable format:
   ```
   driveletter:\PATH_TO_SPECTRUM
   ```
   – On Linux, add the following line to the /opt/SPECTRUM/spectrum80.env file:
   ```
   SPECPATH=PATH_TO_SPECTRUM
   ```
7. On Linux platforms, define BES_LIC_DIR as PATH_TO_SPECTRUM/bin/VBNS/license in the spectrum*.env file in the /opt/SPECTRUM directory.
8. On Linux platforms, copy /usr/bin/perl to <*$SPECROOT*>/bin.
9. Verify that the .hostrc file contains the local host name and the name of the computer from which you transferred the utilities.
10. Verify that the .hostrc file on the main location server contains the host name of the computer where the tools_bundle file is extracted.
11. Verify that the .LocalRegFile contains the correct Main Location Server.
12. Set CLISESSID in a shell to use the CLI utility:
    Windows:
    ```
    set CLISESSID=<NUMBER>
    ```
    Linux:
    ```
    export CLISESSID=<NUMBER>
    ```
    – **NUMBER**
    Is any unique number for the shell.
13. Verify that Notifier/.alarmrc has the correct path to the SetScript, ClearScript, and UpdateScript. These scripts are located in the Notifier directory where you extracted the tools_bundle file.
14. Restart processd.
    You can now run the utilities on this computer.

You can also move DX NetOps Spectrum utilities to a computer that is not running DX NetOps Spectrum.

Follow these steps:

1. Pack up the DX NetOps Spectrum utilities.

2. On the computer where you want to extract the DX NetOps Spectrum utilities, create a directory to unpack the DX NetOps Spectrum utilities and their support files. For example, create the following directories:
   Windows:
   ```
   c:\win32app\spectrum
   ```
   Linux:
   ```
   /usr/spectrum
   ```
   NOTE
   Change the working directory (chdir) to these directories.
3. Send the executable file by FTP to the current directory using binary mode. The file has one of the following names: linux_tools_bundle (Linux) or nt_tools_bundle.exe (Windows).
4. Execute the tools_bundle file from the DOS, Bash, or other UNIX shell.
   The DX NetOps Spectrum utilities and their support files unpack into the appropriate directory structure.
5. Verify that the PATH variable is as follows:
   Windows:
   Verify that <*$SPECROOT*>/lib is in the PATH variable.
   Linux:
   - Create an /opt/SPECTRUM directory.
   - Create a link from <*$SPECROOT*>/lib to /opt/SPECTRUM/lib.
   - Create a link from <*$SPECROOT*>/bin to /opt/SPECTRUM/bin.
6. Set the <*$SPECROOT*> and <*$SPECPATH*> environmental variables to the path of the directory where you extracted the tools_bundle file:
   - On Windows, create the system environment variables <*$SPECROOT*> and <*$SPECPATH*> with the value in variable format:
     ```
     driveletter:/PATH_TO_SPECTRUM
     driveletter:\PATH_TO_SPECTRUM
     ```
   - On Linux, create /opt/SPECTRUM/spectrum80.env and add the following lines:
     ```
     SPECROOT=PATH_TO_SPECTRUM
     SPECPATH=PATH_TO_SPECTRUM
     ```
7. If you transferred AlarmNotifier and event and pcause files, verify that the paths in <*$SPECROOT*>/SG-Support/CsResource/preferences/*.prf have the correct locations for the event and pcause files. Also, verify the ui= and the lhandle= options.
8. On Linux platforms, define BES_LIC_DIR as PATH_TO_spectrum/bin/VBNS/license in the spectrum*.env file in the /opt/SPECTRUM directory.
9. Verify that the .hostrc file contains the local computer host name and the name of the computer from which you transferred the utilities.
10. Verify that the .LocalRegFile contains the correct Main Location Server.
11. Verify that vnmsh/.vnmshrc contains the correct main SpectroSERVER name.
12. Install the processd service:
    Windows:
    - If SRAdmin is not installed, install it as follows:
      ```
      shell> cd %SPECROOT%\Install-Tools\sdic\nt
      shell> .\sradmin --install
      shell> .\sradmin --start
      ```
    - Install the processd service:
      ```
      shell> cd %SPECROOT%\lib\SDPM
      shell> .\processd.exe --install --username USERNAME --password PASSWORD
      ```

NOTE
If processd does not start, reboot the computer.

- To start processed, run the following command:

```
shell> .\processd.exe --start
```

Linux:

- Copy <*$SPECROOT*>/lib/SDPM/processd_init.sh to /etc/init.d/processd.
- Copy <*$SPECROOT*>/lib/SDPM/processd.pl to /etc/init.d.
- Create a link from /etc/init.d/processd to /etc/rc2.d/S99processd.
- Start processd using /etc/init.d/processd start.

13. Verify that the .hostrc file on the main location server contains the host name of the computer where the tools_bundle file is extracted.
14. Set CLISESSID in a shell to use the CLI utility:
    Windows:

```
set CLISESSID=<NUMBER>
```

    Linux:

```
export CLISESSID=<NUMBER>
```

    - **NUMBER**
      Is any unique number for the shell.
15. Verify that Notifier/.alarmrc has the correct path to the SetScript, ClearScript, and UpdateScript. These scripts are located in the Notifier directory where you extracted the tools_bundle file.
16. On Windows platforms, if you want to use AlarmNotifier scripts, install Cygwin from http://www.cygwin.com. Be sure that the PATH variable contains the Cygwin bin directory in it.
    You can now run the utilities on this computer.

Setting Up a Distributed SpectroSERVER Environment

Name Resolution Requirements

The SpectroSERVER system or OneClick web server system must be able to resolve the host name of the SpectroSERVER to an IP address.

We recommend using hosts files for name resolution. This practice ensures that a network outage does not affect SpectroSERVER name resolution.

DX NetOps Spectrum and Multiple Interfaces

All DX NetOps Spectrum servers bind to and listen on all available interfaces and advertise themselves with host names that are not fully qualified. The result is flexibility when configuring a management topology. To establish connections, the DX NetOps Spectrum component servers try all interfaces in the order that is determined by the operating system until the connection succeeds.

Communication Among SpectroSERVERs

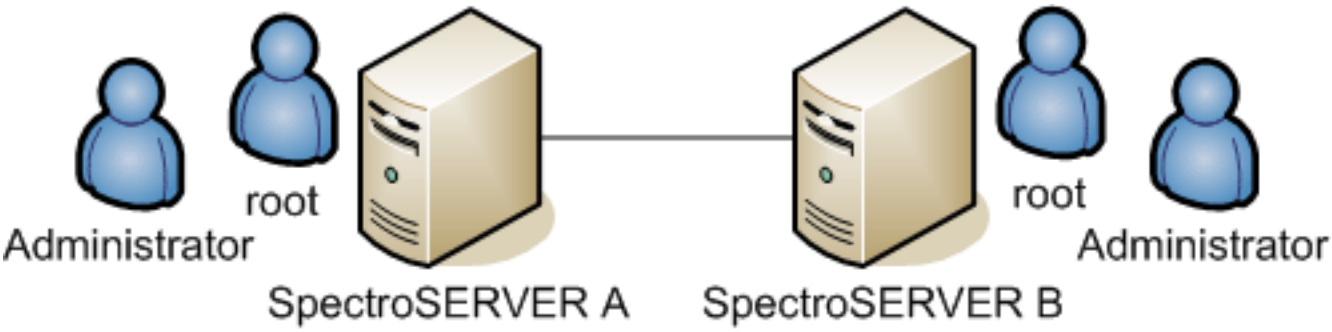
Install and run SpectroSERVERs in a distributed environment with the same user account. No further user model configuration is therefore required. SpectroSERVERs that are installed and run as different users cannot communicate.

Example: SpectroSERVERs A and B Cannot Communicate

In this example, SpectroSERVER A is running as “root”. SpectroSERVER B is running as Administrator. They cannot establish a connection because the request cannot pass through server security:

Example: Configuration for SpectroSERVER to SpectroSERVER Connection

This example shows a user logged in as "root" on SpectroSERVER B, and a user logged in as “Administrator” on SpectroSERVER A. This configuration enables the two SpectroSERVERs to communicate:



DSS Environment Requirements

Your distributed environment must meet the following conditions to enable multiple SpectroSERVERs and the modeling of multiple landscapes to represent those servers:

- Each landscape must contain an identical modeling catalog of the model types in a database and their relations. This replication provides a consistent base for DX NetOps Spectrum intelligence.
- Each landscape must contain the same user models, which you can view in the OneClick Users tab.
- Minimize the number of connections between subnets that are modeled in different landscapes.
- Limit the number of identical devices that are modeled in more than one landscape.
- Assign a unique landscape handle to each modeled landscape. DX NetOps Spectrum can then distinguish it from other landscapes in the DSS environment.

Location Servers

When you install a SpectroSERVER, you also automatically install a *location server*. This server identifies and locates other DX NetOps Spectrum services on the network. DX NetOps Spectrum processes use the location server to determine where these services are running. Processd starts and stops the location server.

In a distributed environment, DX NetOps Spectrum uses location servers to maintain the SpectroSERVER landscape map and provide connection services to client applications. During DX NetOps Spectrum installation, you designate one of the location servers in a landscape map (or DX NetOps Spectrum domain) as the *main location server (MLS)*.

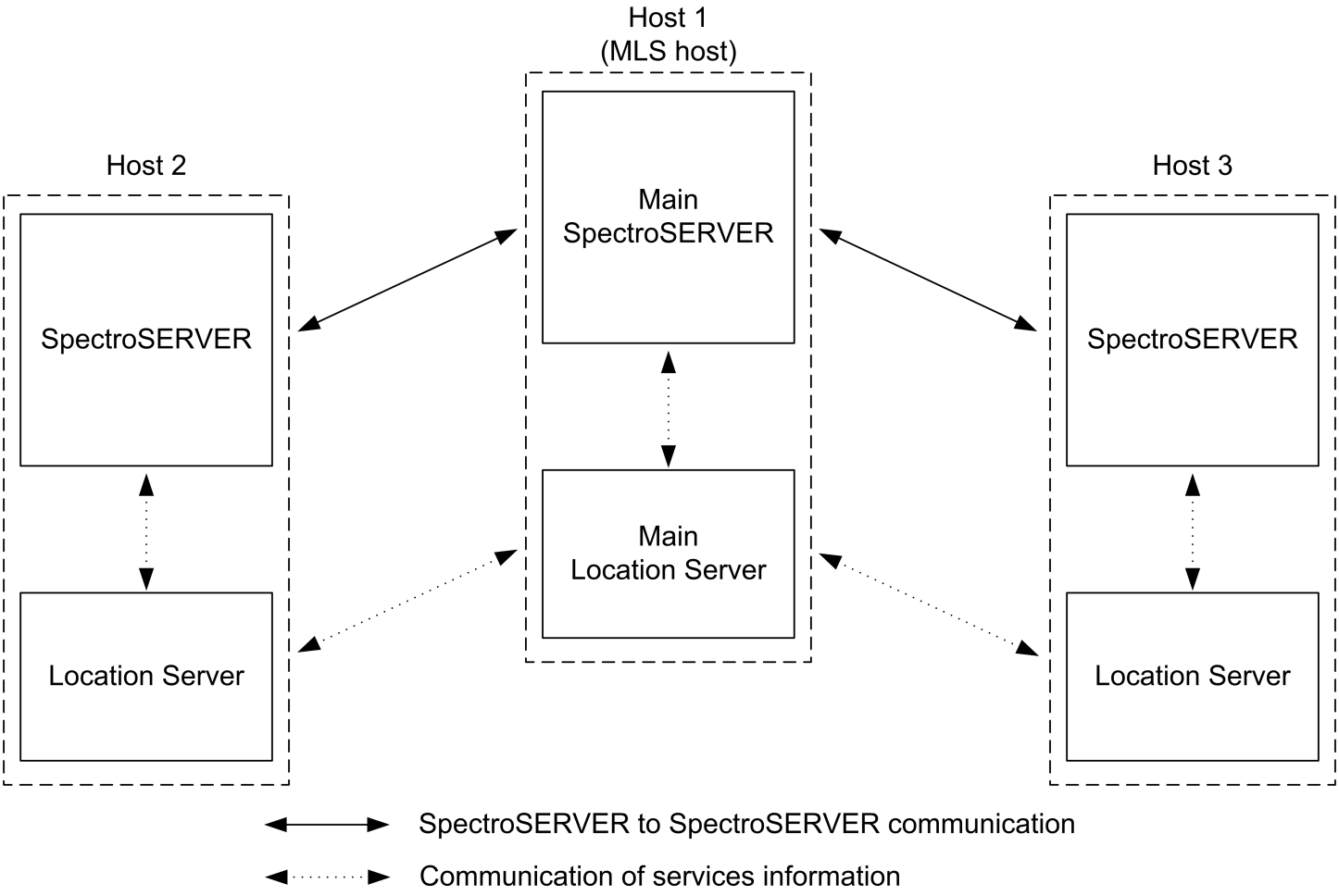
Install the main location server on a highly reliable computer. This server propagates service advertisements among location servers and communicates service locations among the location servers.

In the following diagram, the location server on Host 1 has been configured as the main location server for this environment. Because the main location server resides on the same host as a SpectroSERVER, that SpectroSERVER is considered to be the main SpectroSERVER.

All server-to-server communication is routed through the main location server host. Two non-MLS SpectroSERVERs can communicate only through the main SpectroSERVER.

The SpectroSERVER on Host 2 in the diagram has a distributed service with a resource modeled on the SpectroSERVER on Host 3. All communications about the remote resource are routed through the main location server host. The servers on Host 2 and Host 3 do not directly communicate.

WARNING
This diagram reflects a two-tiered setup consisting of one MLS and multiple child SpectroSERVERs. Such a configuration is the standard, recommended configuration.



How Location Servers Interact

Every DX NetOps Spectrum installation includes the following two files:

- .LocalRegFile lets DX NetOps Spectrum processes locate their location server.
- LS/.locrc is the configuration file for the locally installed location server.

These files manage the location server hierarchy and the interactions between each location server and the main location server.

In a distributed SpectroSERVER deployment, the location servers interact as follows:

1. SSAPI applications and servers read .LocalRegFile to determine the location server to use.
2. Each local location server reads the /LS/.locrc file to determine which server is the main location server.
3. The local location server connects to the main location server to find network services. If the main location server becomes unavailable, other location servers retain service information in memory until the main location server is available.

Main Location Server Connection

Each SpectroSERVER must have a connection to the SpectroSERVER running on the server that is designated as the main location server. This designation is specified in the <*$SPECROOT*>/LS/.locrc file.

The connection to the main location server requires the following configuration:

- Update the ".hostrc" file of the MLS with hostnames of:
  - All child SpectroSERVERs.
  - All OneClick clients.
- Update the ".hostrc" file of each child SpectroSERVER with hostnames of:
  - All OneClick clients.

    NOTE
    By default, the ".hostrc" file of each child SpectroSERVER contains the MLS name that you provide during the installation. By default, the ".hostrc" file on a OneClick client contains only that OneClick client hostname, do not update anything further.
- In a fault-tolerant environment, update the ".hostrc" file of the secondary MLS with hostnames of:
- – All child SpectroSERVERs.
  - All OneClick clients.
- In a fault-tolerant environment, update the ".hostrc" file of each secondary child SpectroSERVER with hostnames of:
  - All OneClick clients.
  - Primary child SpectroSERVER.
  - Secondary MLS.
- Each SpectroSERVER requires a DX NetOps Spectrum user model for the user account under which the other SpectroSERVER is running.

This configuration enables the communication between <ss>s through the MLS. This configuration also enables the <oc> server connection to the MLS.

NOTE
The main landscape (or the value of MAIN_LOCATION_HOST_NAME in the .locrc file) is modeled in DX NetOps Spectrum. The VNM topology for each SpectroSERVER contains this landscape unless it is already modeled elsewhere. An alarm is generated on this model if the connection to the main landscape is lost.

Designate a New Main Location Server

You can change the computer that is designated as the main location server.

On the SpectroSERVER, you designate a new main location server through the Location Server Configuration dialog or through the .locrc file.

On the OneClick web server, you set the main location server information in the DX NetOps Spectrum Configuration page in the OneClick Administration pages. This action updates the name of the main location server to which the OneClick web server connects.

NOTE
If you change the main location server on the SpectroSERVER without updating the OneClick web server, DX NetOps Spectrum does not function properly. For more information, see the Administrating section.

Example: Designating a New Main Location Server on the SpectroSERVER

A distributed network is composed of Computers 1 through 4, with Computer 1 designated as the main location server. To make Computer 4 the main location server, first reconfigure Computer 4 as the new main location server. Then reconfigure Computers 1, 2, and 3 to point to Computer 4 as the main location server.

Follow these steps:

1. Bring up the SpectroSERVER Control Panel.
2. Select Location Server from the Configure menu.
   The Location Server Configuration dialog opens.
3. Change the Main LS Host field in the Location Server Characteristics area to point to Computer 4.

NOTE
You can also edit the .locrc file in the <*$SPECROOT*>/LS directory to change the main location server. The following entry in the .locrc file identifies the main location server:

```
MAIN_LOCATION_HOST_NAME=Computer 4
```

Landscape Map Integrity Preservation

DX NetOps Spectrum users commonly maintain separate production and test environments. Protect the integrity of the landscape map by preventing SpectroSERVERs in the test environment from connecting to a SpectroSERVER in the production environment SpectroSERVER as their main location server, or the reverse.

Landscape Handles

A SpectroSERVER in a distributed deployment identifies each landscape by its *landscape handle*. 10.3 enables you to opt for increased capacity and scale by allowing you to monitor huge number of devices with fewer landscapes. Using the Huge Landscape Handle type (during installation), you can create landscapes which can support huge model count (beyond 1 Million models per SpectroSERVER). Handles are multiples of 64. This option is only available for fresh installs and best suited for new DSS environments and new single-server installations.

This capability is not supported for upgrade and migration scenarios to 10.3 from earlier versions. For direct upgrades and migrations, DX NetOps Spectrum automatically selects the Legacy Landscape Handle type.

NOTE

In a Distributed SpectroSERVER environment, you cannot select a mix of Legacy and Huge landscapes. For example, if you have an existing SpectroSERVER running in Legacy and want to add a new SpectroSERVER, you cannot add a Huge Landscape, even if you are installing it from the scratch.

In a Distributed SpectroSERVER environment, all your SpectroSERVERs should be either Legacy or Huge Landscapes.

Each landscape must have a unique landscape handle, which is either:

- **Legacy Landscape Handle:** a number that is divisible by 4 and is in the range of 4 to 16,376. (In hexadecimal format, handles are in the range 0x100000 to 0xffe00000 where the lower 20 bits are set to zero).
  Each SpectroSERVER is limited to 1 million models. Recommended option if you are using a SSdb created from any version prior to r10.2.
- **Huge Landscape Handle**: a number that is divisible by 64 and is in the range of 64 to 16,256. (In hexadecimal format, handles are in the range 0x1000000 to 0xfe000000 where the lower 24 bits are set to zero.) Each SpectroSERVER can manage/have beyond 1 million models. Recommended for new DSS environments and single-server installations only.

After installation, the encoded landscape handle appears at the top of all views, in OneClick Console that are associated with that landscape.

NOTE
We recommend using a sequential numbering scheme for landscape handles. You can associate a landscape handle with a significant landscape feature, such as a building number or some portion of a subnet IP address. However, for Legacy Landscape Handle(s) your entry is encoded into the 12 most significant bits of the landscape handle, and for Huge Landscape Handle(s) your entry is encoded into the 8 most significant bits of the landscape handle. The result can therefore appear unrelated to the appropriate landscape feature.

NOTE

From 10.3 onwards, you can load the catalog of the Legacy Landscape on a Huge Landscape and vice versa (using SSdbload -c option). However, you will not see the landscape details displayed. For more information on the landscape details, see the Known Anomalies section.

Assign Landscape Handles

Landscape handles can be assigned through the SpectroSERVER Validation dialog when installing DX NetOps Spectrum or by using a utility named lh_set.
You are allowed to set/assign the landscape handle, based on the type of Landscape handle you have opted for during installation. For instance, if you have selected Huge Landscape Handle type (multiples of 64), you cannot assign landscape handles which are multiples of 4 and not multiple of 64, (for example, 4, 16, 32, 68 and so on.)

WARNING
Run the lh_set utility before you run the SpectroSERVER for the first time. Otherwise, DX NetOps Spectrum assigns a default landscape handle that is the same every time that DX NetOps Spectrum assigns it. As a result, duplicate landscape handles can be created when multiple landscapes are configured. Such landscapes can never be accessed simultaneously from the same application.

Follow these steps:

1. Navigate to the SS directory.
2. Enter the following command:
   ```
   ../SS-Tools/lh_set <landscape handle>
   ```
   You can specify the new landscape handle in either decimal or hexadecimal notation. If you use decimal notation, the lh_set utility converts your entry into a hexadecimal landscape handle.
3. If you assign a landscape handle which is in conflict with the Landscape Handle type you have opted for during installation, the following error might appear:
   ```
   Landscape handle must be 64 - 16256, multiples of 64 (or 0x1000000 - 0xfe000000 with low 24 bits 0).
   Unable to set SS landscape handle.
   ```

NOTE
For more information, see Installing and Upgrading .

Change a Landscape Handle

If a SpectroSERVER has started, the process for changing the landscape handle of the SpectroSERVER database entails more steps. The landscape handle is included in the model handle of each model in the SpectroSERVER database. Changing the landscape handle therefore requires converting all model handles from the old landscape handle to the new landscape handle.

Change the landscape handle of all models that were created automatically at startup, and update the landscape handle in several resource files.

Follow these steps:

1. Use the Modeling Gateway tool to export the models from the SpectroSERVER whose landscape handle you want to change.
2. Shut down the SpectroSERVER.
3. Initialize the database with the SSdbload utility.

   NOTE
   For more information, see Database Management .
4. Assign the new landscape handle using the lh_set utility.
5. Start the SpectroSERVER.
6. Import the models into the SpectroSERVER using the Modeling Gateway tool.

   NOTE
   For more information, see the Modeling Gateway Toolkit .

   WARNING

   It is mandatory to reboot the server after you complete a fresh SpectroSERVER installation while installing r10.3.

   NOTE

   Running SpectroSERVER process using '**root**' user instead of '**spectrum**' user on linux machines may result in the following error message "*The user running the parent server does not have a user model in the local landscape*" on the landscape page found in the administration tab on the oneclick console.

   Run 'ps -ef | grep -i spectro' to check if the SpectroSERVER process is owned by '**root**' user or other user instead of '**spectrum**' user (You can check the value of '**initial_user_model_name**' parameter in $SPECROOT/SS/.vnmrc file to check for the '**spectrum**' user) to resolve the issue.

   Follow the steps mentioned to fix the issue:

   1. Launch Spectrum Control Panel using '**root**' or other user who is currently owning the SpectroSERVER process. Stop SpectroSERVER process thoroughly by clicking [**Stop SpectroSERVER**] button.

   2. Change the ownership of VNM.OUT, RCPD.OUT etc. under $SPECROOT/SS directory back to '**spectrum**' user using 'chown' command. Make sure when you run 'ls -l' under $SPECROOT/SS directory all files and directories are owned by '**spectrum**' user except the SpectroSERVER file. The SpectroSERVER file should be owned by '**root**' user and have the following attributes:

   ```
   -rwsr-x---   1 root     spectrum     12841 Nov 21 01:16 SpectroSERVER
   ```

   3. Launch Spectrum Control Panel using '**spectrum**' user and start SpectroSERVER process by clicking [**Start SpectroSERVER**] button.

Process Daemon (processd)

DX NetOps Spectrum uses a process launching and tracking daemon called Process Daemon (processd) to let you control processes on multiple servers in a DSS environment. This daemon starts processes when an application such as the DX NetOps Spectrum Control Panel makes a request. You can also configure processd with install tickets to start processes automatically at system startup. Install tickets can also automatically restart critical processes that stop unexpectedly. Tomcat and Webtomcat are stopped when processed is stopped.

The DX NetOps Spectrum installation program configures processd to start automatically (where it must run as root) and on Windows systems (where it runs as the LocalSystem account).

Process launch and monitoring with processd occur only when an application requests such actions. Typically, processd operates in the background.

If a process does not start or is not working properly, processd writes an error message to the *<$SPECROOT>*\lib\SDPM \processd_log file. The error message includes information to identify the problem.

When restarted, processd creates a processd_log.bak file to preserve old error messages and appends any new error messages to the processd_log file.

> **WARNING**
> You must be thoroughly familiar with DX NetOps Spectrum, distributed networking, and network configuration before attempting any of the procedures described in this section.

Processd Differences in Windows Environments

The processd daemon works slightly differently on Windows platforms because of differences in their native security architectures.

- When you request a process start from a remote connection in a Windows environment, the process starts as the user who is specified in the Windows Service Configuration dialog.
- For server processes that must remain running after user logout (or before anyone logs in), a SERVERPROCESS field is used in the Windows environment. To prevent Windows from shutting these processes down during logout, these processes are started as the user who is specified in the Windows Service Configuration dialog.

Change the Windows Password in Processd

During installation on Windows, you are prompted to enter a username and password in the Windows Service Configuration dialog. The processd daemon uses this username and password to start DX NetOps Spectrum processes as the specified user. Providing the security information lets these processes run when no user is logged in.

> **NOTE**
> For more information, see Installing .

If the password you specified in the Windows Service Configuration dialog changes or expires, you can update processd with the new password.

> **WARNING**
> If your password contains special characters, such as an exclamation point (!), escape these characters with a backslash (\). Or you can change the password from a command prompt.

Follow these steps:

1. Log in as a member of the local Administrators group.
   You must be a member of the DX NetOps Spectrum Users group and an administrator to change the processd user name and password.
2. Open a command prompt.
3. Navigate to the /lib/SDPM directory.
4. Enter the following command:
   ```
   processd --install --username user --password newpassword
   ```
   – ***user***
   Specifies the user name.
   – ***newpassword***
   Specifies the new password for this user name.
   If the user name or the password contains special characters that the shell can misinterpret, enclose the user name or password in quotation marks. For example, the user name contains a domain name and user, with a backslash (\) separator. The password contains an asterisk (*). Both of these characters are special characters. Therefore, the user name and password are enclosed in quotation marks, as in the following example:
   ```
   processd --install --username "DOMAIN\JSmith" --password "283EJ*"
   ```

NOTE
If the password for the Windows user name changes but the processd service password is not updated, processd does not start. The following events are generated in the Windows event log:

Install Tickets Files

You can configure processd to start processes at system startup. You can also select processes to restart if they stop running. Files that are known as *install tickets* support this functionality.

Install tickets use files in the following two directories:

- <*Spectrum Installation Directory*>/lib/SDPM/partslist
- <*Spectrum Installation Directory*>/lib/SDPM/runtime

The partslist directory contains the individual install ticket files.

The runtime directory contains encoded files in the form of <*PID*>.rt where <*PID*> is the process ID of the running process.

Note: SDPM stands for DX NetOps Spectrum Distributed Process Manager.

You can add install tickets to the partslist directory. New install tickets must conform to certain formatting rules. Restart processd to identify any new or modified install tickets in the partslist directory. These files are read at processd startup and are stored in cache memory for future use.

Install ticket files follow a naming convention that refers to the process whose configuration information it contains. Install ticket filenames are in the form of <*PARTNAME*>.idb. The <*PARTNAME*> variable is an internal key to identify processes that processd controls.

The following defining fields are used for install tickets. The format is <*fieldname*>;<*value*>; where <*fieldname*> is one of the names that are displayed below. The <*value*> is a string that provides a definition for the corresponding field name. Quotation marks are not supported.

- **PARTNAME**
  Identifies a particular process/application with a multicharacter string with no spaces. The install ticket for this application has a filename in the form <*PARTNAME*>.idb. <*PARTNAME*> is the process name.
- **APPNAME**
  Defines the name of the application as a multicharacter string.
- **WORKPATH**
  Specifies where you want to run the application. Supply a value for this field *before* it can be used as part of the ARGV field that is described later.
- **LOGNAMEPATH**
  Specifies the log file for output from the application. This field must begin with <*$SPECROOT*> or <*$WORKPATH*>. No spaces are allowed.
- **ADMINPRIVS**
  Is reserved for use in Purism-supplied install tickets only. Comment it out in install tickets that you create.
- **AUTORESTART**
  Indicates whether a process is automatically restarted if it stops. If this field is not included in the install ticket, automatic restart is disabled by default. Accepted values are Y, y, N, n.
- **AUTOBOOTSTART**
  Indicates whether the process is started whenever processd starts. If this field is not included in the install ticket, the function is disabled by default. Accepted values are Y, y, N, n.
- **STATEBASED**
  Indicates whether the process has more than one state when starting up. Usually, the process sends an “is ready” ticket when ready to communicate. This field is reserved for use in Purism-supplied install tickets only. If this field is not included in the install ticket, it is disabled by default. Accepted values are Y, y, N, n.
- **NUMPROCS**

Specifies the number of instances of a process that can run on one platform. Numeric values are accepted.

- **RETRYTIMEOUT**
  Specifies the number of seconds that processd tries to restart the application after failure.
- **TICKETUSER**
  Defines the username of the user who is authorized to run the process when the AUTOBOOTSTART and/or AUTORESTART fields are set. This field is only required when those fields are included in the install ticket. This field is not applicable on Windows because all processes run as the processd install user.
- **RETRYMAX**
  Specifies the number of times that processd tries to restart an application within the specified RETRYTIMEOUT period.
- **STARTPRIORITY**
  Indicates the relative startup priority of the process relative to other processes. The following values are used:
  – 10 for standalone processes
  – 20 for processes that depend on standalone processes
  – 30 for processes that depend on the SpectroSERVER
- **SERVERPROCESS**
  Indicates to processd whether a process continues to run after the user logs out. When this field is enabled, the process is always started as the user who is authorized to run the processd service. This field is only applicable in the Windows environment. The default value is 'yes' for the following processes:
  – Archive Manager (ARCHMGR.idb)
  – Location Server (LOCSERV.idb)
  – SpectroSERVER (SS.idb)
- **SERVICE**
  Indicates whether the ticket represents a Windows service. Lets processd manage Windows services from the Service Control Manager.
- **ENV**
  Adds one or more variables to the application environment. Multiple values are listed on a separate line with the macro <CSPATHSEP> between the value and the semicolon that ends the line.
- **ARGV**
  Defines the argument list of the process, which includes the executable path and any number of arguments (spaces are allowed).

Examples Install Tickets

Install tickets apply the following syntax conventions:

- Any line beginning with a pound sign (#) is treated as a comment.
- A semicolon (;) must follow all field names and all values following each field name.
- Anything following the second semicolon is disregarded.
- Four macros can be used: <CSEXE>,<CSBAT>,<CSCMD>, and <CSPATHSEP>. Based on the platform, the appropriate definition is substituted. Use <CSPATHSEP> instead of the actual path separator to avoid parsing conflicts.

The following example shows a valid install ticket:

```
# Processd Install Ticket for SpectroSERVER Daemon.
PARTNAME;SS;
APPNAME;SpectroSERVER Daemon;
WORKPATH;$SPECROOT/SS;
LOGNAMEPATH;$WORKPATH/VNM.OUT;
ADMINPRIVS;y;
#AUTORESTART;N;
#AUTOBOOTSTART;N;
STATEBASED;y;
```

NUMPROCS;3; // unlimited
RETRYTIMEOUT;0; // seconds
#TICKETUSER;$USER;
RETRYMAX;0; // retries
STARTPRIORITY;20;
SERVERPROCESS;Y;
#ENV;<var>=<value>;
ARGV;$SPECROOT/SS/SpectroSERVER<CSEXE>;

The following example shows another valid install ticket:

Processd Install Ticket for Visibroker Naming Service
PARTNAME;NAMINGSERVICE;
APPNAME;Visibroker Naming Service;
WORKPATH;$SPECROOT/bin/VBNS;
LOGNAMEPATH;$WORKPATH/NAMINGSERVICE.OUT;
ADMINPRIVS;y;
AUTORESTART;y;
AUTOBOOTSTART;y;
#STATEBASED;N;
NUMPROCS;1; // one per host
RETRYTIMEOUT;600; // 10 minutes
#TICKETUSER;$USER;
RETRYMAX;5; // 5 retries
STARTPRIORITY;10;
SERVERPROCESS;Y;
#ENV;<var>=<value>;
ENV;CLASSPATH=$SPECROOT/lib/vbjorb.jar<CSPATHSEP>;
ENV;CLASSPATH=$SPECROOT/lib/vbsec.jar<CSPATHSEP>;
ENV;CLASSPATH=$SPECROOT/lib/lm.jar<CSPATHSEP>;
ENV;CLASSPATH=$SPECROOT/lib/sanct6.jar<CSPATHSEP>;
ARGV;
$SPECROOT/bin/JavaApps/bin/nameserv<CSEXE>
-DORBpropStorage=
$SPECROOT/.corbarc
-Dvbroker.orb.admDir=
$SPECROOT/bin/VBNS
-Dborland.enterprise.licenseDir=
$SPECROOT/bin/VBNS/license
-Dborland.enterprise.licenseDefaultDir=
$SPECROOT/bin/VBNS/license
-Djava.endorsed.dirs=
$SPECROOT/lib/endorsed
-Dorg.omg.CORBA.ORBClass=
com.inprise.vbroker.orb.ORB
-Dorg.omg.CORBA.ORBSingletonClass=
com.inprise.vbroker.orb.ORB
com.inprise.vbroker.naming.ExtFactory;

Start a New Install Ticket Process

If you added an install ticket file, you can use the restart option to start the process specified in the ticket. With the restart option, you do not have to stop and restart processd and all of the processes that it is monitoring.

Follow these steps on Windows:

1. Log in as a member of the DX NetOps Spectrum Users group.
2. Open a command prompt.
3. Navigate to the /lib/SDPM directory.
4. Enter the following command:
   ```
   perl processd.pl restart
   ```
   The process is started.

Stop and Restart Processd

If you suspect that the lib/SDPM/runtime directory has become corrupted, stop and restart processd.

Follow these steps: on Windows

1. Log in as a member of the DX NetOps Spectrum Users group.
2. Open a command prompt.
3. Navigate to the /lib/SDPM directory.
4. Enter the following command:
   ```
   perl processd.pl stop (or start)
   ```

   NOTE
   On Windows, the processd.pl <*start/stop*> commands also stop and start the DX NetOps Spectrum processes that run as NT services (such as MySQL and VisiBroker).

How the Userconf Process Works During Installation

The userconf process runs processd in the background. Userconf lets you perform user-specific configuration of DX NetOps Spectrum during installation and afterwards, when a user logs in to DX NetOps Spectrum.

The userconf process performs the following tasks during DX NetOps Spectrum installation:

1. Runs userconf -install %SPECROOT%, which makes the following changes:
   - Adds SPECROOT and SPECPATH to the system environment.
   - Adds $SPECPATH\lib to the $PATH system variable.
   - Removes the old $SPECPATH\lib entries from the path, if the installation directory has changed.
   - Removes %SPECPATH%\lib from the path (if it exists).
   - Modifies the registry so that userconf runs each time that a user logs in (with no arguments).
2. Runs userconf -start to start processd. The -start flag starts processd without verifying that the user is a member of the DX NetOps Spectrum Users Group.

   NOTE
   The installation program adds the current user to the DX NetOps Spectrum Users Group. However, the change does not take effect until the user logs out and logs back in. Add the -start flag so that processd starts without verifying that the user is a member of the DX NetOps Spectrum Users Group. The installation then continues without requiring the user to log out and log back in.
3. Runs userconf -restart to stop and restart processd if the user is installing Exceed for the first time (which occurs after the DX NetOps Spectrum installation). Processd can then update PATH environment changes that are part of the Exceed installation.

How Userconf Works When a User Logs In

The installation adds the userconf process to the registry Run key. When a user logs in, the process starts automatically and checks whether the user is a member of the DX NetOps Spectrum Users group. The userconf process applies the following rules:

1. – If the user is a member of the DX NetOps Spectrum Users group, userconf verifies cygwin and Exceed and reconfigures as needed before starting processd. If Microsoft VC++ is installed, userconf configures it appropriately for use with the DX NetOps Spectrum SDK.
   – If the user is not a member of the Users group, a message states that DX NetOps Spectrum is installed but the user is not in the DX NetOps Spectrum Users group.
   To enable the user to use DX NetOps Spectrum, add the user account to the DX NetOps Spectrum Users group. The user must then log out and log back in.
   A user who does not want to use DX NetOps Spectrum can select the “user doesn't want to run Spectrum” check box. With this option, userconf continues to run when the user logs in, but the process exits silently.

   NOTE
   To see the message again, run the following command:

```
userconf -install %SPECROOT%
```

Duplicate Models in a Distributed Environment

In a DSS environment, DX NetOps Spectrum tracks duplicate models (for example, a user model that exists on multiple landscapes) by designating one as a "home" model. The home model resides on the SpectroSERVER that is running on the host with the "root" main location server (MLS).

WARNING
If you change the root MLS in a DSS environment, the state of all duplicate models is updated to match the "home" model on the MLS. The MLS is the final authority in any DSS environment.

DX NetOps Spectrum uses the home model to synchronize information for all duplicate models. Modification requests that are made to the duplicate models that are not the home models are relayed to the landscape of the home model. The home model then distributes the request to all of the other duplicate models.

Failure to Contact Home Landscape Errors

Editing operations can fail when the home landscape of the model that you are editing cannot be contacted. OneClick reports relation errors such as the following error from the SpectroSERVER:

```
The operation failed because the model being edited exists in multiple landscapes and its home landscape could
 not be contacted.
```

This type of error can indicate that the home model on the SpectroSERVER that is running on the "root" MLS is unreachable. When duplicate models are added or deleted in a distributed environment, they are routed through the home landscape. If that SpectroSERVER is down or cannot be contacted, the relation fails and generates an error.

Failure to contact the home landscape can occur under any of these conditions:

- The home SpectroSERVER or an intermediate SpectroSERVER between the duplicate model and the home SpectroSERVER is not running.
- A network problem is preventing SpectroSERVER access.
- A security issue is preventing the SpectroSERVERs from communicating.

Host Resource Configuration File (.hostrc)

The .hostrc file restricts client application access to each local SpectroSERVER. Client applications cannot connect to the local landscape unless the .hostrc file is configured to permit this access. You can use a text editor to edit the .hostrc file.

NOTE
For more information, see the Administrating *section*.

By default, the .hostrc file is initially installed with the local hostname, which restricts all remote access. Adding an individual hostname enables connections to and from that server. Enable unrestricted access to and from all remote servers by adding a "+" symbol. Add the machine name to the .hostrc file to enable client access from individual remote servers.

DX NetOps Spectrum implements host security in multiple layers, by hostname or by IP address. We recommend listing hostnames in the .hostrc file, which includes all IP addresses that are associated with the hostnames. Connection attempts that are initiated by the hostname do not always succeed if an IP address is used to connect.

Time Zones in a DSS Environment

In DSS installations that span multiple time zones, all activities that occur on a SpectroSERVER reflect the local time of the server, including scheduled events. All schedules created in OneClick and applied to modeled devices start and end according to the local time of the SpectroSERVER or device landscape.

> **NOTE**
> For more information, see OneClick Schedules in a DSS Environment.

SpectroSERVER Shutdown

We recommend shutting down the SpectroSERVER using the DX NetOps Spectrum Control Panel before you shut down the server where the SpectroSERVER is running. However, if you use the operating system shutdown procedure as a way to shut down the SpectroSERVER, increase the amount of time for processd to stop.

SpectroSERVER Shutdown in a Windows Environment

In a Windows environment, processd waits until all subprocesses have shut down before it stops. However, the Windows registry setting WaitToKillServiceTimeout sets the length of time that Windows waits for all services to stop after a Windows shutdown is initiated. If a service such as processd is still running after the WaitToKillServiceTimeout elapses, Windows terminates this service. The default value is 20 seconds, which is not always enough time for the SpectroSERVER to shut down completely at system shutdown. You can increase the timeout value.

Follow these steps:

1. Open the Registry Editor.
2. Go to HKEY_LOCAL_MACHINE/SYSTEM/CurrentControlSet/Control.
3. Click the Control key.
4. Double-click the WaitToKillServiceTimeout value in the right pane.
5. In the window, change the value to anything up to 600,000 milliseconds (10 minutes).
6. Click OK.
7. Restart the Windows server for the change to take effect.

Set the Landscape Map Entry Timeout

By default, each landscape entry remains in the landscape map indefinitely. Using the location server configuration file (.locrc), you can specify an amount of time after which the landscape entry times out and is removed automatically from the landscape map.

> **NOTE**
> In previous releases, landscape entries timed out after one hour, and they were removed automatically from the landscape map. Starting in 9.2.2, landscape entries do not time out by default.
>
> **WARNING**
> We do *not* recommend using a timeout value to automatically remove an entry from a landscape map. Instead, leave the default timeout value. Use the MapUpdate command to remove an entry from the landscape map when required. For more information, see Database Management.

Follow these steps:

1. Navigate to the <*$SPECROOT*>/LS directory.
2. Update or add the following entry in the .locrc file:
   MET_INTERVAL=*time_out_value*
   – ***time_out_value***
   Indicates the time in milliseconds after which a landscape entry times out and is removed automatically from the landscape map.
   Default: 0 milliseconds (no time-out)

Problems with Landscape Mapping from Existing DSS Setup

Symptom:

I experienced problems with the landscape map when I set up a separate distributed environment by copying from another installation. I noticed a failure to switch over to the secondary SpectroSERVER after the primary goes down and problems with user security.

Solution:

To copy a landscape map, remove all references to the previous DSS setup. The landscape maps must contain no references to the previous DSS setup. Use the following procedure to map a landscape from an existing DSS setup.

Follow these steps:

1. Decouple the SpectroSERVERs as follows:
   a. Make each SpectroSERVER a standalone server.
      The location server is now pointing to the local server where it is installed.
   b. Verify that the landscape map is distinct for each SpectroSERVER.

      NOTE
      In the existing DSS setup, the landscape map was a single map that included all of the servers.
2. Save the SpectroSERVER databases.
   The landscape maps are clean.
3. Change the MLS of these SpectroSERVER to restore the original MLS.

   NOTE
   Do not change the original DSS settings.
4. Reload the databases of each SpectroSERVER on different hosts.
5. Select a SpectroSERVER as your MLS in the new DSS setup, and point other SpectroSERVERs to it.
   The dupModeList in the user models updates properly.
6. Remove secondary entries in the landscape map, if they exist.
7. Set up secondary SpectroSERVER entries.
8. Load the database from the primary SpectroSERVER in the new DSS.
   The landscape is mapped from the existing DSS setup.

Troubleshooting

Problems with Purging Old Landscapes

Symptom:

Our production DX NetOps Spectrum environment consisted of a single distributed installation. The environment had multiple SpectroSERVERs and OneClick servers, which all used the same Main Location Server.

We decided to remove two SpectroSERVERs and a OneClick server from the production environment to create a separate development environment. We configured the development SpectroSERVERs and OneClick server to reference

a new Location Server. We updated the .hostrc files so that the servers in each environment only had access to their respective Location Servers. In addition, we updated the landscape map in the production environment to remove the development landscapes.

However, if I look at the list of Monitored Landscapes on development OneClick server, I still see all of the old production landscapes. They are listed as having "No permission" because of the changes to the .hostrc file.

Solution:

With the setup that you describe, you now have two separate environments. However, your development landscape map is caching stale production landscapes. You removed stale landscapes from your production environment, but you did not remove them from your development environment. Therefore, you must manually remove the stale landscapes from the development environment.

You have two options for removing the stale landscapes.

Solution:

You can use the following command to remove stale landscapes from the development landscape map:

```
MapUpdate -remove landscape handle
```

Solution:

You can restart all SpectroSERVER processes. Then restart the OneClick server. For more information about OneClick server administration, see OneClick Administration.

When you restart the servers, the issue is resolved.

> **WARNING**
>
> Its is mandatory to reboot the server after you complete a fresh SpectroSERVER installation while installing r10.3.

The landscape handle bitmask in the SpectroSERVER database file does not match the bitmask of DX NetOps Spectrum

Symptom:

When trying to load a previously saved SpectroSERVER Database (SSdb) savefile, the following error shows:
Error: Detected incompatible model mask configuration. The database is configured as 20 bits, but the server is configured as 24 bits. Database can't be loaded.

The landscape handle bitmask in the SpectroSERVER database file does not match the bitmask of DX NetOps Spectrum. If the "Huge Landscape Handle" option was selected during install, this means that the SpectroSERVER database file has a legacy landscape handle. If the Legacy Landscape Handle was selected during install, this means the SpectroSERVER database file has a huge landscape handle.

Solution:

Loading a previously saved SpectroSERVER database file with a different landscape handle bit mask is not supported. Review your SpectroSERVER environment for proper configuration. If you have installed and chosen the wrong landscape option, uninstall and reinstall DX NetOps Spectrum. If you selected an incorrect handle and that SS is in a DSS, remove the incorrect entry from the map after you have reinstalled. To remove the incorrect entry from the map after you have reinstalled:

1. Open a bash shell and navigate to the:
   `<SPECROOT>/SS-Tools/ directory`
2. Run the MapUpdate command:
   `./MapUpdate -v`

3. Remove the incorrect entry:
   ```
   ./MapUpdate -remove <lh> -precedence <precedencevalue>
   ```
   For example: ./MapUpdate -remove 0x100000 -precedence 10

Communication Across Firewalls

Communicating across a firewall can apply in many network environments. In a distributed environment, firewalls are likely to affect your deployment.

> **NOTE**
> Before you begin a distributed DX NetOps Spectrum installation, configure the firewall to enable traffic on port 46517 (the port sradmin).

Your options for enabling communications among DX NetOps Spectrum components depend on the types of firewalls that you deploy. Other factors, such as cost and the level of security that is required, also play a role. These factors include Virtual Private Networks (VPNs), node-to-node tunnels or conduits, and proxies that encapsulate packets before they pass through the firewall.

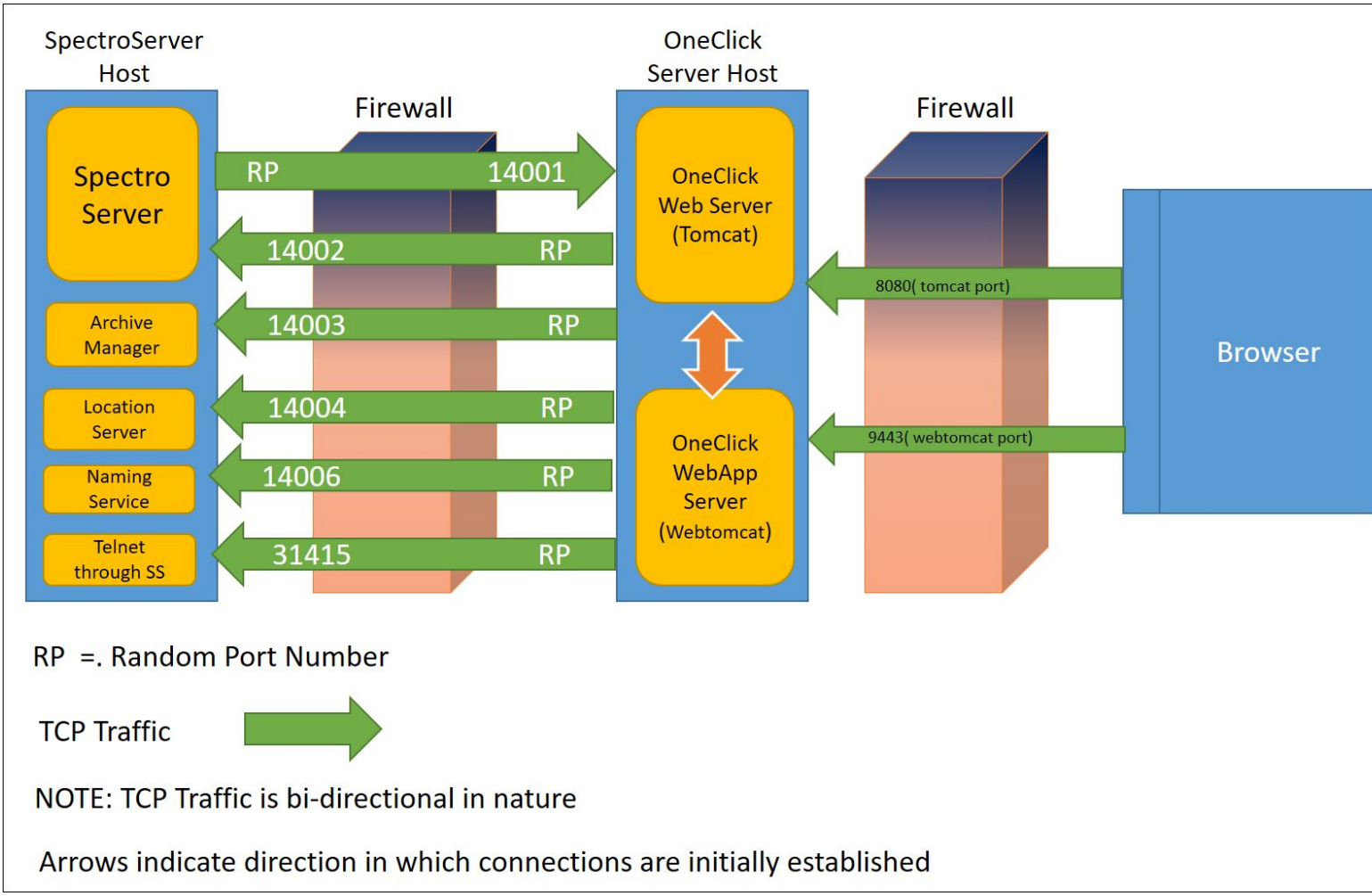
We recommend working with the firewall administrator for your network to work out a strategy for letting DX NetOps Spectrum and OneClick traffic traverse firewalls.

SpectroSERVER and OneClick Web Server Communication Across Firewalls

The OneClick web server communicates with processes on the SpectroSERVER host system to gather data to display in OneClick clients. The OneClick web server typically initiates this communication.

The OneClick web server establishes connections to specific SpectroSERVER host-side TCP ports. The web server uses these ports for sending requests and receiving responses. However, OneClick uses a single listening port (default 14001). The SpectroSERVER initiates the connection to that port. As a result, modifying firewall configuration is often necessary. The SpectroSERVER uses bidirectional IIOP (Internet Inter-ORB Protocol) to communicate with its CORBA clients.

The following diagram illustrates the IP connectivity that is required for an OneClick web server to communicate with a SpectroSERVER. In all cases where TCP is used, the connections are established from a random port to a specific, fixed port.



NOTE
In a Fault-Tolerant configuration, the same ports must be opened between the OneClick Server Host and the secondary SpectroSERVER Host, including port 14003 if running a secondary Archive Manager.

OneClick Default Ports and Firewalls

HTTP Listen Port

The default port used by OneClick for HTTP communication is port 80. If you configure the OneClick web server to use something other than port 80, your firewall must also be set up to allow this traffic.

NOTE
For more information, see the OneClick Administrator and Fresh Install.

OneClick users on the Windows XP SP2 platform who choose to leave the Windows Firewall enabled may have problems running the OneClick Console.

NOTE
For more information on configuring the Windows firewall, refer to the Microsoft Knowledge Base article 842242 at http://support.microsoft.com.

CORBA Listen Port

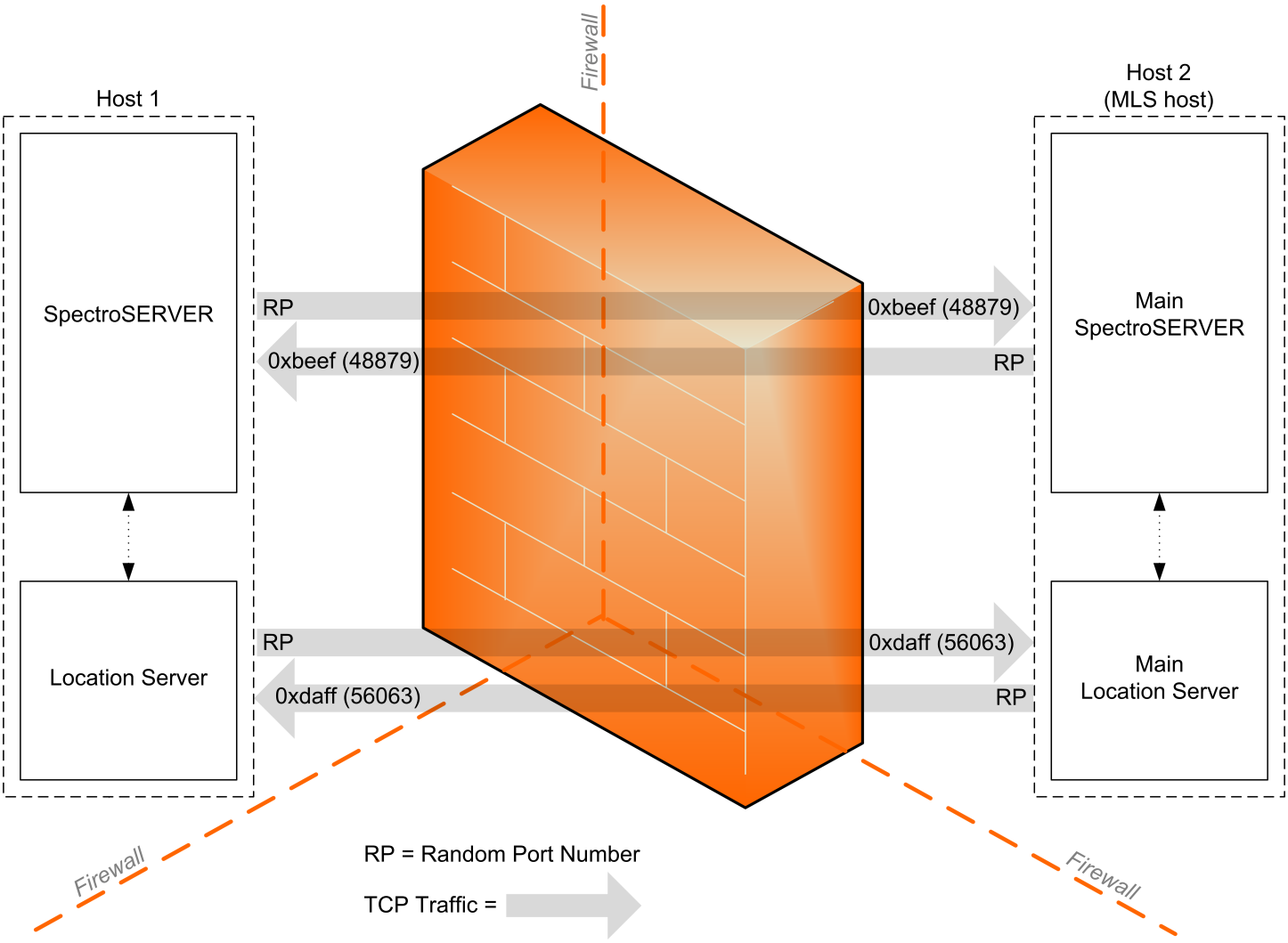
The Default CORBA OneClick server listen port value is located in <*$SPECROOT*>/tomcat/webapps/spectrum/META-INF/context.xml.

```
vbroker.se.iiop_tp.scm.iiop_tp.listener.port=<new port number>
```

Remote SpectroSERVERs and Firewalls

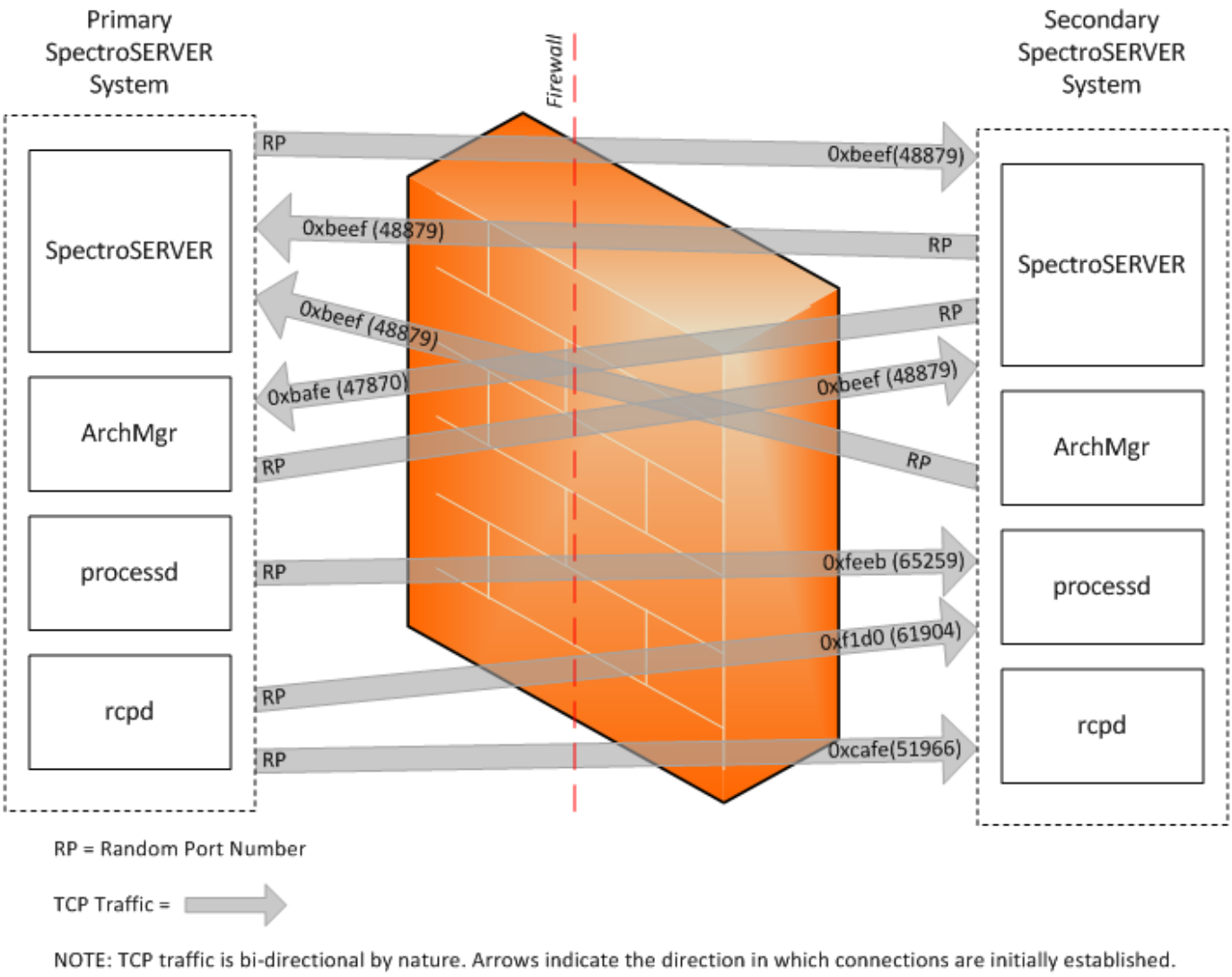
The following figure illustrates the IP connectivity that is required when two remote SpectroSERVERs communicate through a firewall. In all cases where TCP is used, the connections are established from a random port to a specific, fixed port.

> **WARNING**
> All communication among SpectroSERVERs that are not Main Location Servers (MLSs) is routed through the MLS-SpectroSERVER. As a result, each SpectroSERVER only communicates directly with the MLS-SpectroSERVER.



Primary and Secondary SpectroSERVER Communication Across Firewalls

The following figure illustrates the IP connectivity that is required when primary and secondary SpectroSERVERs in a fault-tolerant environment communicate through a firewall. In all cases where TCP is used, the connections are established from a random port to a specific, fixed port.



NOTE: TCP traffic is bi-directional by nature. Arrows indicate the direction in which connections are initially established.

Configuration Files for NAT Firewall Environments

Network Address Translation (NAT) is used to construct networks with consistent internal IP addressing and a single node that handles IP address translation to the Internet. DX NetOps Spectrum supports NAT. You can deploy DX NetOps Spectrum in a private IP address domain and can maintain connectivity to clients outside of NAT firewalls. Therefore, DSS environments can now encompass multiple domains that are separated by these firewalls.

The only requirement for a NAT environment is the ability of clients to resolve the server by name. On the private side of the NAT, the host name must resolve to the private-side IP address. On the public side of the NAT, the host name must resolve to the public-side IP address.

Default Port Configurations

You can change the default port or socket number that a DX NetOps Spectrum process uses so that the process works correctly in a firewall environment.

You can change the port number or socket number for the following processes:

- OneClick WebServer
- SpectroSERVER
- Archive Manager
- Location Server
- Naming Service
- Remote Copy Process Daemon (rcpd)
- CLI Daemon (vnmshd)

> **NOTE**
> The only ports that you cannot change are the ports for the Remote Administration Daemon, sradmin, and the port for Telnet through SpectroSERVER.

Change the SpectroSERVER Port Number

You can change the port number that the SpectroSERVER uses for CORBA requests.

Use a text editor to edit the .vnmrc file to reflect the new port number. This file is located in <*$SPECROOT*>/SS/. Enter the following command:

```
orb_args=-Dvbroker.se.iiop_tp.scm.iiop_tp.listener.port=<new port number>
```

Change the Archive Manager Port Number and Socket Number

You can change the port number and the socket number that the Archive Manager uses for specific requests.

Use a text editor to edit the .configrc file to reflect the new port number. This file is located in <*$SPECROOT*>/SS/DDM. Enter the following command:

```
orb_args=-Dvbroker.se.iiop_tp.scm.iiop_tp.listener.port=<new port number>
```

To change the socket number that the Archive Manager uses to listen for requests from VNM and SSAPI clients, use a text editor to edit the .configrc file. This file is located in <*$SPECROOT*>/SS/DDM. Change the following variable:

```
ARCH_MGR_SOCKET_NUMBER=<new port number>
```

Change the Location Server Port Number and Socket Number

You can change the port number and the socket number that the Location Server uses for specific requests.

To change the port number the Location Server uses for CORBA requests, use a text editor to edit the .locrc file to reflect the new port number. This file is located in <*$SPECROOT*>/LS. Enter the following command:

```
orb_args=-Dvbroker.se.iiop_tp.scm.iiop_tp.listener.port=<new port number>
```

To change the socket number that the Location Server uses for VNM and SSAPI requests, use a text editor to edit the .locrc file. Change the following variable:

```
LOC_SERVER_SOCKET_NUMBER=<new port number>
```

Change the Visibroker Naming Service Port Number

You can change the port number that the Visibroker Naming Service uses.

> **NOTE**
> The default port number is 14006.

Follow these steps:

1. Right-click My Computer and select Properties.
   The System Properties dialog opens.
2. Click the Advanced tab and click the Environment Variables button.
3. Select the NAMING_SERVICE_PORT and edit it to reflect the new port number:
   ```
   NAMING_SERVICE_PORT=<new port number>
   ```

To change the Visibroker Naming Service port number, set the environment variable in the spectrum60.env file. This file is located in the /opt/SPECTRUM directory. Change the following variable:

```
NAMING_SERVICE_PORT=<new port number>
```

Remote Copy Process Daemon (rcpd) Port Number Configuration

Configure the remote copy process daemon (rcpd) port number in the ".vnmrc" resource file as shown in the following syntax:

```
rcpd_comm_port=<port number in hex>
```

CLI Daemon (vnmshd) Port Number Configuration

The CLI Daemon (vnmshd) port number is configurable through the vsh_tcp_port parameter.

> **NOTE**
> For information about the vsh_tcp_port parameter, see the Command Line Interface section.

Port Conflict Resolution

To avoid a port conflict with another application, you can change the default port numbers that DX NetOps Spectrum processes and services use.

The following table describes the default ports for DX NetOps Spectrum processes and services.

> **NOTE**
> "Not used for remote connections" is noted for ports in this table, where applicable. Ports that DX NetOps Spectrum does not use for remote connections typically do not present firewall configuration concerns.

Port (dec)	Port (hex)	DX NetOps Spectrum Components	Protocols	Notes
80	0x0050	OneClick/SRM	HTTP	None
162	0x00A2	SpectroSERVER	UDP	Port used to listen for SNMP traps.
3306	0x0CEA	mysql for SRM data ArchMgr (Archive Manager)	TCP/ODBC/JDBC	Used for remote connections only when migrating an SRM database from Windows to Linux.
3307	0x0CEB	mysql for BOXI on Linux	TCP	Not used for remote connections.

6844	0x1ABC	SDC	TCP	None
6400	0x1900	BOXI CMS	TCP	Not used for remote connections.
7777	0x1E61	CLI vnmshd	TCP	See CLI Daemon(vnmshd).
14001	0x36B1	OneClick Server	TCP/CORBA	See OneClick WebServer Host.
14012	0x36B2	SpectroSERVER	TCP/CORBA	See SpectroSERVER.
14013	0x36B3	ArchMgr	TCP/CORBA	See Archive Manager.
14014	0x36B4	LocServ	TCP/CORBA	See Location Server.
14016	0x36B6	Naming service	TCP/CORBA	None
31415	0x7AB7	Telnet through SpectroSERVER		None
46517	0xB5B5	sradmin	TCP	Remote Administration Daemon. **Note:** For more information about the sradmin process, see the Installing and Upgrading section.
47870	0xBAFE	ArchMgr	TCP/SSAPI	See Archive Manager.
48879	0xBEEF	SpectroSERVER	TCP/SSAPI	See SpectroSERVER.
51966	0xCAFE	rcpd	TCP	See Remote Copy Process Daemon (rcpd).
56063	0xDAFF	LocServ	TCP	See Location Server.
61904	0xF1D0	processd	TCP	This port is not configurable.
64222	0xFADE	TL1d	TCP/EPI	TL1 gateway agent
65259	0xFEEB	processd	TCP	This port is not configurable.

About SpectroSERVER Fault Tolerance

About SpectroSERVER Fault Tolerance

Fault tolerance requires more than one SpectroSERVER to manage a given landscape. A copy of the database for that landscape is loaded on each SpectroSERVER. However, only a single copy is active at any time. The SpectroSERVER with the active database is known as the *primary SpectroSERVER* . The inactive database runs on a standby SpectroSERVER, which is the secondary SpectroSERVER. You can also install another inactive copy of the database on a tertiary SpectroSERVER.

If the primary SpectroSERVER fails, the database on the secondary SpectroSERVER becomes active, and the secondary SpectroSERVER starts managing the network. Applications that are connected to the primary SpectroSERVER are automatically switched to the secondary SpectroSERVER. When the primary SpectroSERVER returns to service, the applications automatically switch back to the primary SpectroSERVER, and the secondary SpectroSERVER becomes inactive again.

NOTE
Not all applications can exercise the full range of their capabilities when they are being run from a secondary SpectroSERVER. The main reason to set up a fault-tolerant environment is to ensure continuous monitoring of the network, not to create a full copy of DX NetOps Spectrum.

SpectroSERVER Precedence in a Fault Tolerant Environment

Primary, secondary, and tertiary SpectroSERVERs that manage the same landscape must all have the same landscape handle and the same modeling catalog. The servers are distinguished from one another with a numeric precedence value. The lowest number indicates the primary SpectroSERVER. SpectroSERVERs are installed with a default precedence value of 10. To designate a SpectroSERVER as a secondary server, assign it a higher precedence number, such as 20. Likewise, a tertiary SpectroSERVER would have a higher precedence than the secondary, for example, 30.

When you first set up a fault tolerant environment, you can assign precedence values at the time you are loading database copies on any standby SpectroSERVERs using the SSdbload utility.

To change precedence values later, you can use the Loaded Landscapes subview. Access this subview by selecting a local landscape in the Navigation panel, and then selecting the Information tab in the Component Detail panel.

NOTE
The Loaded Landscapes subview is different from the SpectroSERVER Control subview. Access the SpectroSERVER Control subview by selecting the VNM in the Navigation panel and then selecting the Information tab in the Component Detail panel.

SpectroSERVER Data Synchronization

A single database is active at any given time in a fault tolerant DX NetOps Spectrum environment. Therefore, the other databases must be updated periodically to reflect new models and changes to attribute values in the active database. This synchronization of data is accomplished through the DX NetOps Spectrum Online Backup feature. You can run Online Backup on demand or at regularly scheduled intervals. When you run Online Backup against the primary SpectroSERVER, it creates a backup copy of the current database. Online Backup automatically loads the copy onto each designated secondary SpectroSERVER.

As in any DSS environment, each of the SpectroSERVERs in a fault tolerant environment must have the same modeling catalog installed. Online Backup copies the current modeling catalog. However, it does not copy all the .i files or other elements that are associated with individual management modules. Therefore, if you install any new management modules on your primary SpectroSERVER, also install the same new management modules on any secondary SpectroSERVERs.

NOTE
For more information, see the Database Management .

EventDisp and the Alertmap files that are defined in the <*$SPECROOT*>/custom/Events directory are propagated to fault-tolerant servers when the secondary SpectroSERVER polls the primary SpectroSERVER for status information.

Support for Fault-Tolerant Archive Manager

You can run the Archive Manager on the secondary SpectroSERVER host in a fault-tolerant SpectroSERVER environment. This secondary Archive Manager provides visibility to events in OneClick when the primary Archive Manager is down.

Primary or secondary SpectroSERVER locally stores events in the following two scenarios:

- When primary Archive Manager is down, and the primary SpectroSERVER is running. In this case, primary SpectroSERVER locally stores events as they are created until primary Archive Manager is up.
- When the primary SpectroSERVER host itself is down. In this case, the secondary SpectroSERVER locally stores events as they are created until the primary Archive Manager is up.

You can start the secondary Archive Manager on the secondary SpectroSERVER host to provide visibility to not only events as they are created when the primary Archive Manager is down, but also historical events.

When you start the secondary Archive Manager, it acts as a client to the primary SpectroSERVER to receive and log events as they are created. This behavior does not affect the normal connection between the primary SpectroSERVER and primary Archive Manager. When the primary Archive Manager goes down, OneClick fails over to the secondary Archive Manager to provide event data.

When the primary SpectroSERVER host itself goes down, the secondary SpectroSERVER locally stores events, but also forwards events to secondary Archive Manager. When the primary Archive Manager comes up, the secondary SpectroSERVER transfers all the locally stored events to it.

Archive Manager Data Synchronization

The secondary Archive Manager provides a best-effort synchronization of events, and there is no event synchronization that occurs between the primary Archive Manager and the secondary Archive Manager. When the secondary Archive Manager is running and connected to a SpectroSERVER, it receives a copy of all events as they are generated. Anytime the secondary Archive Manager is down, events are not stored on the secondary. This functionality is distinctly different from the functionality of primary Archive Manager, where the SpectroSERVER stores the events for later transfer to the primary Archive Manager.

This means that when the secondary Archive Manager is started for the first time, its DDM database does not contain any events, and no attempt is made to synchronize with the primary. Once the secondary Archive Manager has been running for MAX_EVENT_DAYS configured in the .configrc, it is generally in sync with the primary Archive Manager database.

Generate an Alarm If the Secondary SpectroSERVER Is Not Restarted

When a primary SpectroSERVER synchronizes its database with the secondary SpectroSERVER, a Contact Lost to Secondary Server (0x00010c0e) event and alarm are generated. The secondary SpectroSERVER has been brought down to load the new database from the primary SpectroSERVER.

You can set up a rule to process this alarm so that the alarm is generated only if the secondary SpectroSERVER is not restarted.

The EventPair rule lets you specify that a new event is generated if the Contact Lost to Secondary Server event occurs and a Contact Established to Secondary Server (0x00010c0f) event does not follow within a specified time period. You can then specify that this new event creates an event and an alarm indicating that the secondary SpectroSERVER is still down.

Follow these steps:

1. Open the EventDisp file with a text editor.

   NOTE
   The EventDisp file is located in the <*$SPECROOT*>/SS/CsVendor/Cabletron directory.
2. Find the line that reads 0x00010c0e E 50 A 2, 0x00010c0e and change this line to the following:

   ```
   0x00010c0e R Aprisma.EventPair, 0x00010c0f,
       <numberofsecondstowait><generatedeventcode>
   ```

   – ***<generatedeventcode>***
   Is the event code to generate if the secondary SpectroSERVER does not come up within the time specified in <*numberofsecondstowait*>.
3. Add the following line to the EventDisp file:

   ```
   <generatedeventcode>E 50 A 2, <generatedalarmcode>
   ```

   – ***<generatedeventcode>***

Is the event code generated in Step 2 if the secondary SpectroSERVER did not come up. 'E 50' indicates that the event is logged and has a severity value of 50. A 2 indicates that a major alarm is created. <*generatedalarmcode*> is the alarm code to generate based on this event.

4. Create a Probable Cause file for this alarm that indicates that contact with the secondary SpectroSERVER has not been reestablished after data synchronization.

NOTE
For more information, see the Event Configuration .

Secondary SpectroSERVER Readiness Levels

A secondary SpectroSERVER is considered to be at one of three different levels of readiness. Readiness depends on server configuration and status. The readiness levels are defined as follows:

- **Hot**
  The secondary SpectroSERVER is running and is available to take over immediately upon failure of the primary SpectroSERVER because it is already polling. To configure a secondary SpectroSERVER for this level of readiness, add the following line to the .vnmrc file: secondary_polling=yes. This statement causes the standby to commence polling and processing traps whenever it starts, regardless of its connection status with the primary SpectroSERVER.
- **Warm**
  The secondary SpectroSERVER is running, but the server can take a short time to become fully available. The secondary SpectroSERVER has not been configured to start polling *until* it loses contact with the primary SpectroSERVER. For example, it has no secondary_polling entry in the .vnmrc file, or the entry is set to no. If the secondary_polling entry is not in the .vnmrc file or the entry is set to no, the secondary SpectroSERVER does not process traps while in standby mode.
- **Cold**
  The secondary SpectroSERVER is not running and must be started when there is a failure of the primary SpectroSERVER. In this case, it is irrelevant whether the secondary SpectroSERVER is configured for secondary polling.

SpectroSERVER Alarm Synchronization

The primary and secondary SpectroSERVERs use a Global Alarm Service (GAS) connection to share alarm information. The SpectroSERVERs use the alarm information to synchronize alarms. This synchronization helps to prevent duplicate alarms.

Options for fault-tolerant alarm synchronization include enablement of the service and debug logging. Settings in the .vnmrc file control these options. For more information, see Fault Tolerant Alarm Service (.vnmrc) Resources.

The correlations are not preserved because the alarms on the secondary SpectroSERVER are generated rather than being created from events.

The process of alarm synchronization differs depending on whether the synchronization is from the primary SpectroSERVER to the secondary SpectroSERVER or from the secondary to the primary. The following sections describe each of these scenarios:

- Synchronization from the Primary to the Secondary SpectroSERVER
- Synchronization from the Secondary to the Primary SpectroSERVER

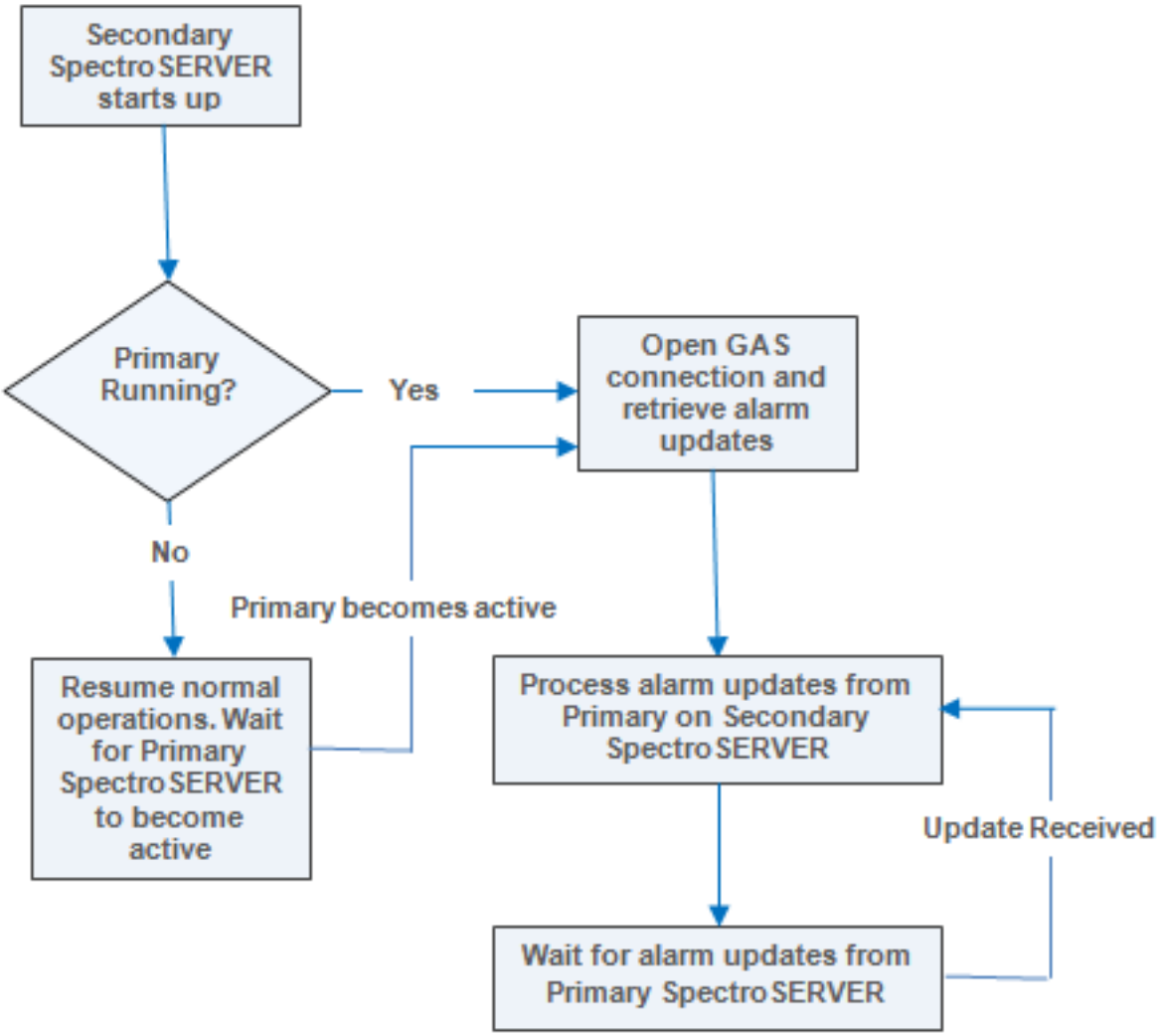
Synchronization from the Primary to the Secondary SpectroSERVER

If the SpectroSERVER is running as a secondary SpectroSERVER, it tries to open a GAS connection to the primary SpectroSERVER. If the connection succeeds, the secondary SpectroSERVER registers to receive alarm updates

from the primary SpectroSERVER. The secondary server remains connected to the primary and continues to receive alarm updates as they occur on the primary server. Otherwise, the secondary SpectroSERVER tries once each minute to connect until the primary SpectroSERVER becomes active.

NOTE
The Fault Tolerant Alarm Service must be enabled on both the primary and secondary servers for the secondary SpectroSERVER to attempt to connect to the primary SpectroSERVER for alarm synchronization. To control this feature, use the ftasv_enabled parameter in the .vnmrc file. For more information, see Fault Tolerant Alarm Service (.vnmrc) Resources.

The following diagram describes how the secondary SpectroSERVER processes alarm updates from the primary SpectroSERVER:



- The new primary SpectroSERVER alarms are added on the secondary SpectroSERVER and marked as 'stale'. A new alarm replaces an existing alarm if they are equivalent. You can determine whether the alarms are equivalent by verifying the following parameters:

  - Unique Alarm (IDs)
  - Model Handle
  - Probable Cause
  - Alarm Discriminators

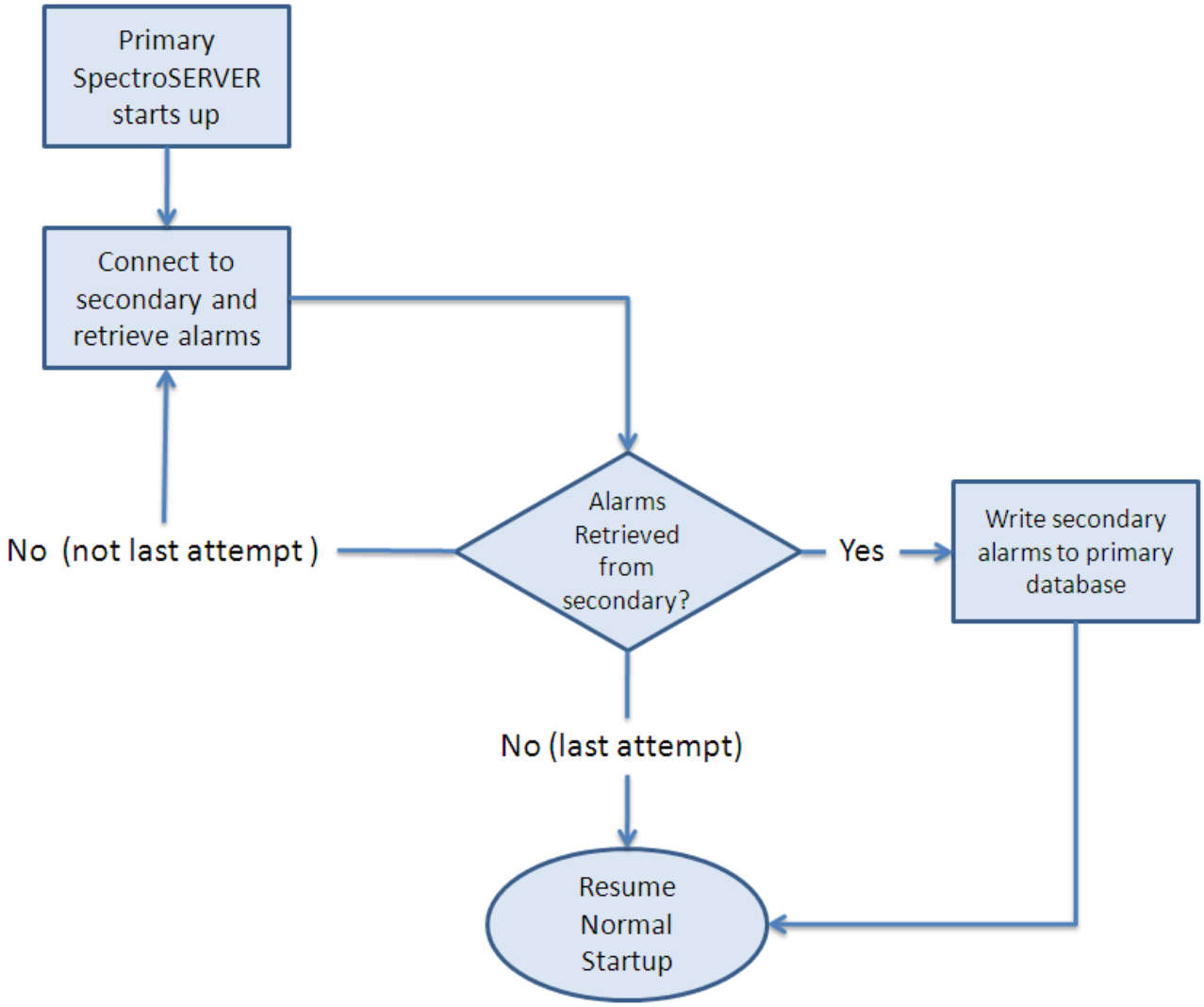
  Two alarms are considered equivalent if they have the same ID. Alarms that are on the same model (that is, they have the same model handle) and that have the same probable cause are also considered equivalent unless they have different alarm discriminators.
- The alarm attribute updates from the primary SpectroSERVER are applied to the same alarms on the secondary SpectroSERVER. If an alarm is cleared on the primary SpectroSERVER, it is also cleared on the secondary SpectroSERVER.
- The isManaged and isNotHibernating attributes are updated for maintenance mode alarm synchronization.

Synchronization from the Secondary to the Primary SpectroSERVER

When the primary SpectroSERVER comes online after a failure or planned maintenance, it connects to the secondary SpectroSERVER, and it gets all available alarms. If the first connection attempt fails, the primary server can make multiple connection attempts to synchronize. The ftasv_max_conn_retry_count and ftasv_conn_retry_interval parameters in the .vnmrc file of the primary SpectroSERVER control the number and frequency of attempts. For more information, see Fault Tolerant Alarm Service (.vnmrc) Resources.

> **WARNING**
> If the secondary SpectroSERVER is not running when the primary server attempts to connect to it, the primary exhausts its synchronization attempts. The result is a delay in starting the primary. This delay is unavoidable because it occurs before model activation. Avoid the situation by verifying that the secondary server is running when the primary starts. You can also decrease the retry count or interval size to reduce the potential delay. Or you can disable the Fault Tolerant Alarm Service using the ftasv_enabled parameter in the .vnmrc file.

The following diagram describes how the primary SpectroSERVER opens a GAS connection to the secondary SpectroSERVER and retrieves all of its alarms.



After a Successful Connection:

- After it retrieves the list of alarms, the primary SpectroSERVER writes the alarms to its database. Primary SpectroSERVER startup continues and these alarms are read from the database.
- The alarms from the secondary SpectroSERVER replace the alarms that are stored in the primary database at retrieval.
- All alarms that are read from the database are processed and asserted on the correct models.
- The secondary 'alarm set' event (0x10714) generates for each new alarm that was generated on the secondary SpectroSERVER while the primary SpectroSERVER was down.
- The secondary alarm clear event (0x10715) generates for each alarm that cleared on the secondary SpectroSERVER while the primary SpectroSERVER was down.

> **NOTE**
> No synchronization occurs for blue and gray alarms. Synchronization occurs for brown alarms, but not for WA_Link models. The WA_Link model exception only applies to brown alarms.

After a Failed Synchronization:

- If alarm synchronization fails when the primary SpectroSERVER is attempting to connect to the secondary, the primary starts up, but an event and critical alarm (0x10c35) are generated on the primary LocalLscpe

model. The event and alarm include the reason for the failure of each attempt. To continue running the primary SpectroSERVER after alarm synchronization has failed, run an Online Backup to synchronize the primary and secondary SpectroSERVER databases.

Establish Fault Tolerance

Establishing Fault Tolerance

You can set up a fault-tolerant environment when you first install DX NetOps Spectrum, before any models have been created. Or you can set up a fault-tolerant environment after you install DX NetOps Spectrum.

The following procedure describes how to set up two SpectroSERVERs: a primary and a secondary. You can also set up a tertiary SpectroSERVER by taking the same steps. However, assign the tertiary SpectroSERVER a higher precedence number than the secondary SpectroSERVER.

NOTE
To establish fault tolerance in an environment with a Southbound Gateway integration, see the Southbound Gateway Toolkit

Follow these steps:

1. Install the same version of DX NetOps Spectrum with the same modeling catalog on both the primary SpectroSERVER and the secondary SpectroSERVER. Each server requires the same landscape handle.
2. Verify that both the primary and secondary SpectroSERVERs have entries in their .hostrc files that give the SpectroSERVERs mutual access permissions.

   NOTE
   If you are specifying secure users for the secondary SpectroSERVER in the .hostrc file on the primary SpectroSERVER, and the secondary SpectroSERVER is running in the Windows environment, include the user SYSTEM in the secure user list.
3. Verify that the MAIN_LOCATION_HOST_NAME parameter in the .locrc file on the secondary SpectroSERVER server points to the same system name as the .locrc file on the primary SpectroSERVER. Otherwise, synchronization fails.
4. Configure the primary and secondary SpectroSERVERs so that the user running each SpectroSERVER is the same. If the users are not the same, the secondary SpectroSERVER fails or does not run properly after an Online Backup.
5. Make a copy of the primary SpectroSERVER database by running Online Backup. Or, if the SpectroSERVER is shut down, use the SSdbsave utility with the -cm argument (to save the modeling catalog and any new models).

For more information, see the Database Management .

1. Verify that the save file that you created is available to the server that hosts the secondary SpectroSERVER. Copy the file to the server if necessary.
2. On the secondary server, with SpectroSERVER shutdown, navigate to the DX NetOps Spectrum SS directory and load the save file using the following command:

   ```
   ../SS-Tools/SSdbload -il -add precedence savefile
   ```

   – ***precedence***
   Specifies a numeric value greater than the primary server default value of 10 (20 is recommended).
   – ***savefile***
   Specifies the name of the saved file that was previously created.
3. (Optional) Add the line 'secondary_polling=yes' to the .vnmrc file to let the secondary SpectroSERVER function as a hot backup.
4. Start the primary SpectroSERVER, if it is not already running.
5. Start the secondary SpectroSERVER.
6. To verify the setup, use the MapUpdate command with the view argument to display the current landscape map.

For more information, see the Database Management .

The secondary SpectroSERVER is now available to take over automatically if the primary SpectroSERVER fails. If you previously activated secondary polling, the secondary SpectroSERVER is available immediately. Otherwise, polling begins when the server detects that it has lost contact with the primary SpectroSERVER.

When service switches from the primary SpectroSERVER to the secondary SpectroSERVER, the Connection Status icon displays yellow. To view the connection status of all servers in a landscape, click the Connection Status icon. In the Connection Status dialog, the Connection Status icon for each server in the landscape displays yellow to indicate the “switched” condition.

When the primary SpectroSERVER comes back online, the secondary SpectroSERVER stops polling (unless you have set secondary_polling to 'yes'). All the applications switch back to the primary SpectroSERVER. However, any edits that you make to the secondary SpectroSERVER while it is active are *not* automatically replicated to the primary SpectroSERVER. Manually recreate these modifications on the primary SpectroSERVER.

When you restart the primary SpectroSERVER, connections are accepted when all models are loaded, but *before* all models are activated. The models can take some time to activate. Because the secondary SpectroSERVER stops polling when the primary SpectroSERVER is restarted, a gap in your network management coverage can result.

To avoid this situation, edit the .vnmrc file on the primary SpectroSERVER so that the wait_active resource is set to 'yes'. This parameter causes the server to wait until all of the models are activated before accepting any connections. The message area in the DX NetOps Spectrum Control Panel also dynamically displays the percentage of models that are activated. The SpectroSERVER can appear to take longer to come up. However when all the models are activated, the SpectroSERVER is ready to manage the network.

You can also set the wait_active resource to 'yes' on the secondary SpectroSERVER. During a planned shutdown of the primary SpectroSERVER, you can then verify in the DX NetOps Spectrum Control Panel that the secondary SpectroSERVER is ready to take over.

For more information, see the Database Management .

Validate Fault Tolerance Configuration

After you have set up fault tolerance in a distributed SpectroSERVER deployment, verify that the OneClick server has access to both primary and secondary SpectroSERVERs. Without connectivity to both servers, the OneClick server cannot failover to the secondary SpectroSERVER.

Follow these steps:

1. Access the OneClick Administration, Landscapes web page.
2. Check the ‘Secondary Status’ column. Verify that OneClick has established contact with the secondary SpectroSERVER.
   The status also indicates whether Fault Tolerance is ready for failover.
   The Fault Tolerance configuration is validated.

Test Fault Tolerance

During initial installation, the secondary SpectroSERVER might not have access to all the devices to which the primary SpectroSERVER has access. This situation causes the secondary SpectroSERVER to generate false alarms. To avoid false alarms, verify that the secondary SpectroSERVER can manage your network devices by testing fault tolerance.

NOTE
Test fault tolerance whenever new devices are added to the primary SpectroSERVER.

Follow these steps:

1. With both the primary and secondary SpectroSERVERs up and running, bring down the primary SpectroSERVER.

The Connection Status icon is yellow to indicate the "switched" condition.
A red connector indicates that the OneClick server was not able to contact the secondary SpectroSERVER.

2. Wait for 15 - 20 minutes for the secondary SpectroSERVER to run.
3. Verify the following conditions:
   – The Connection Status icon does not display red.
   – All device models and pingable models maintain SNMP or ICMP contact.
   If this contact is lost, verify that the secondary SpectroSERVER has access to your devices. Contact a Network Administrator to resolve this problem, if applicable.
   – DX NetOps Spectrum is managing all devices that have an established contact state. Verify the status by checking for device contact or management contact loss alarms from any of the device models.
4. Restart the primary SpectroSERVER.
   The Connection Status icon displays green to indicate a normal contact state.

Fault-Tolerant Recovery

Following are the two possible failure scenarios:

- The primary SpectroSERVER stops. The secondary SpectroSERVER then forwards event and statistical information to the primary Archive Manager that is running on the server that hosts the primary SpectroSERVER. When the primary SpectroSERVER restarts, no event and statistical data have been lost.
- The computer where the primary SpectroSERVER and the primary Archive Manager is running stops operating completely. The secondary SpectroSERVER then caches event and statistical data in its database until the primary SpectroSERVER computer comes back online. If a secondary Archive Manager is running, historical, and real-time information is available in OneClick, but the information is still cached for transfer to primary Archive Manager.

Restart both the primary Archive Manager and the primary SpectroSERVER if their server goes down, or if the primary SpectroSERVER stops operating.

NOTE
It is no longer necessary to start the Archive Manager before the SpectroSERVER, the cached events from the secondary SpectroSERVER can be transferred at any time, even after the primary SpectroSERVER has started logging new events.

Follow these steps:

1. Start the Spectrum Control Panel on the primary SpectroSERVER host.
2. To start the SpectroSERVER, click Start SpectroSERVER on the Spectrum Control Panel.
   When the primary Archive Manager is again operational, the secondary SpectroSERVER connects and transfers its cached event data to the primary Archive Manager.

Change the Host Names of the Primary and Secondary SpectroSERVERs

SpectroSERVERs in a fault-tolerant environment use a precedence value that is associated with their host names to recognize their relationship to one another. Therefore, to preserve the fault-tolerant relationship, use SSdbsave and SSdbload to change the host name of your primary SpectroSERVER.

Follow these steps:

1. Save the database using SSdbsave with the -cm option.
2. Change the host name.
3. Reload the database with the save file that you created in the first step. Run SSdbload with the -il option and the -replace option:

   ```
   SSdbload -il -replace precedence savefile
   ```

   This command causes the database to associate the new host name with the precedence value (10) that designates a primary SpectroSERVER.

The change in the host name is communicated to any warm or hot standby SpectroSERVERs the next time that the databases are synchronized as a result of Online Backup being run.
In the meantime, however, the host name change prevents the standby SpectroSERVERs from detecting that the primary SpectroSERVER is running. As a result, any SpectroSERVER that is configured as a warm standby starts polling.

4. Load the save file on the warm standby using SSdbload with the -il and -replace options, and specify a higher precedence value (for example, 20) that designates it as a standby.

Now you can change the host name of the secondary SpectroSERVER.

Follow these steps:

1. Save the database using SSdbsave with the -cm option.
2. Make the change to the host name.
3. Reload the database with the save file that you created in the first step. Run SSdbload with the -il option and the -replace option:

   ```
   SSdbload -il -replace precedence savefile
   ```

   This command causes the database to associate the new host name with the precedence value (20) that designates a secondary SpectroSERVER.
   When you restart the secondary SpectroSERVER, the server communicates the new host name and precedence to the primary SpectroSERVER.

For more information, see the Database Management .

Monitor the Changeover Between the Primary and Secondary SpectroSERVERs

You can use watches to monitor the status of your fault-tolerant environment. Create a watch that alerts you when either the primary or the secondary SpectroSERVER is ready to take over.

Follow these steps:

1. Create a watch on the VNM model to monitor the PercentInitialized (0x11da6) attribute.
   When the value of this attribute is equal to 100 percent, the SpectroSERVER has been initialized and is ready to take over.
2. Set up the watch to generate an event or an alarm or run a script when the following expression evaluates to TRUE:

   ```
   PercentInitialized == 100
   ```

3. Set up the watch as an active polled watch.
4. Synchronize the secondary SpectroSERVER with the primary SpectroSERVER to propagate the watch.
5. Specify a value for the Model_Name (0x1006e) attribute in the watch expression. This attribute notifies you only when the secondary SpectroSERVER is ready to take over.
   For example, if the following watch expression evaluates to TRUE, the secondary SpectroSERVER <*sec_server*> is ready to take over.

   ```
   (PercentInitalized == 100) & (Model_Name= <sec_server>)
   ```

   NOTE
   For more information, see the Watches section
6. Add the following line to the .vnmrc file on the secondary SpectroSERVER to limit the potential for false events or alarms:

   ```
   is_secondary = TRUE
   ```

   This setting lets the secondary SpectroSERVER drop events unless DX NetOps Spectrum determines that the secondary SpectroSERVER has taken over as the primary SpectroSERVER.

How to Monitor the Secondary SpectroSERVER Status

Monitoring the Secondary SpectroSERVER Status

You can create a watch that alerts you when the secondary SpectroSERVER server is acting as the primary SpectroSERVER.

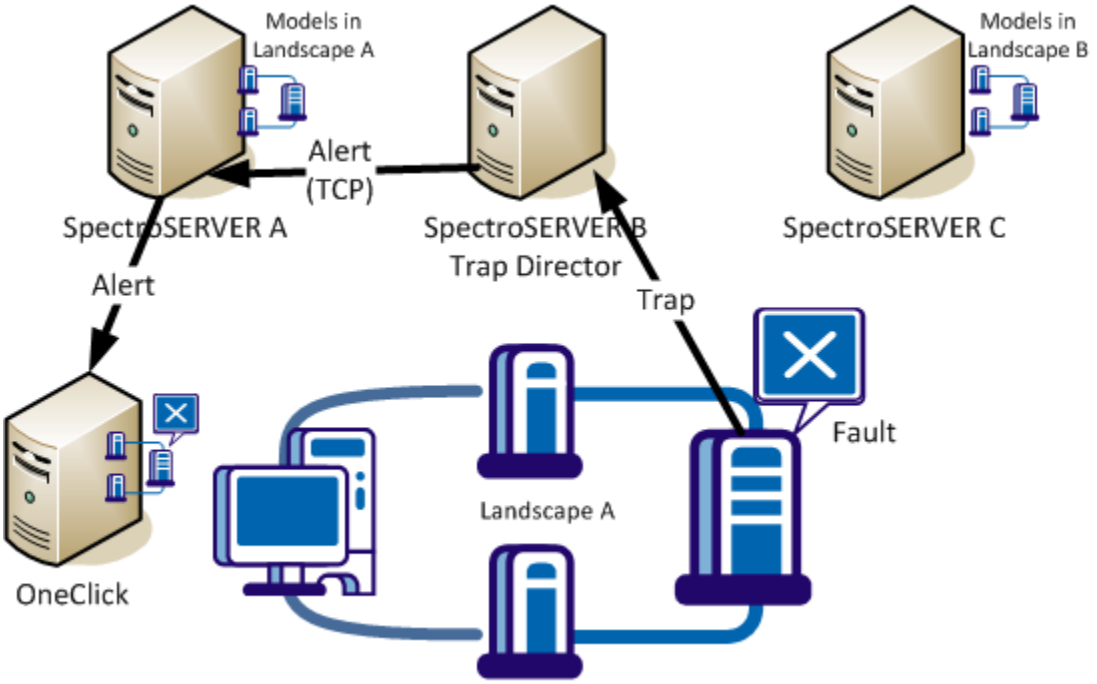
Follow these steps:

1. Create a watch on the VNM model to monitor the value of the secondary SpectroSERVER's PausePolling attribute (0x11b63).
   When this attribute is set to FALSE on the secondary SpectroSERVER, the secondary SpectroSERVER is polling and is acting as the primary SpectroSERVER.
   For example, if the following watch expression evaluates to TRUE, the secondary SpectroSERVER <*sec_server*> is acting as the primary SpectroSERVER.
   ```
   !PausePolling & (Model_Name == <sec_server>)
   ```
2. Set up the watch as an active polled watch.
3. Synchronize the secondary SpectroSERVER with the primary SpectroSERVER to propagate the watch.

   NOTE
   For more information, see the Watches section.
4. Add the following line to the .vnmrc file on the secondary SpectroSERVER to limit the potential for false events or alarms:
   ```
   is_secondary=TRUE
   ```
   This setting lets the secondary SpectroSERVER drop events unless DX NetOps Spectrum determines that the secondary SpectroSERVER has taken over as the primary SpectroSERVER.

Working with Trap Director

Trap Director is a SpectroSERVER feature that you can enable in a distributed SpectroSERVER environment. Trap Director receives traps and processes them on the Trap Director host server. Trap Director then uses the information in traps to create alerts, which it forwards over a TCP connection to the SpectroSERVER. The main Trap Director mechanism is therefore the DX NetOps Spectrum internal alerts that the SpectroSERVER sends to the models on remote hosts for additional processing. The following diagram illustrates the Trap Director architecture:



To use Trap Director, designate one SpectroSERVER as the Trap Director server. Enable Trap Director on that server. Then configure that server as the network management station (NMS) recipient for traps from devices that are modeled in the remote landscapes.

> **NOTE**
> You can designate more than one SpectroSERVER as a Trap Director in the same distributed SpectroSERVER (DSS) environment.

Trap Director supports encrypted SNMPv3 traps. DX NetOps Spectrum attempts to use your SNMPv3 profiles to decrypt incoming encrypted SNMPv3 traps.

The available SpectroSERVER SNMPv3 Profiles are treated as credential templates for unknown SNMPv3 traps. When an encrypted SNMPv3 trap is received, the SNMPv3 profiles are used to decrypt the trap.

> **NOTE**
> A SpectroSERVER running as a trap forwarder can process SNMPv3 traps if the Trap Director landscape has an SNMPv3 profile with valid credentials.

Traps and Memory Usage

When Trap Director is enabled, monitor the memory usage of the SpectroSERVER process. In cases where a large number of traps (more than 100 per second) are forwarded to multiple remote landscapes, a situation can occur in which the SpectroSERVER cannot forward all of the incoming traps. The traps are then queued for further processing. If the situation continues, the memory that the queue uses can exhaust the memory resources that are available for use by the SpectroSERVER process. The SpectroSERVER then shuts down unexpectedly because it lacks sufficient memory.

In addition to the process memory usage, closely monitor the following attribute in the VNM model:

```
alert_remote_fwd_queue_length (Attribute ID: 0x130c3)
```

Generally, if the queue size continues to grow beyond 1000 elements, we do not recommend enabling Trap Director in your environment.

Trap Data Traffic Consolidation

Consolidating Trap Data Traffic

As network growth prompts you to apportion models among additional landscapes, you can retain the original landscape host as the trap destination for these models. You can thus avoid reconfiguring new trap destinations on the associated devices. Conversely, you can consolidate multiple trap destinations and can designate a single SpectroSERVER to receive and route traps. If load sharing is an important factor, you can enable Trap Director on multiple servers. Each server handles traps from a set of devices that are configured to send traps to that server.

Trap Director maintains an up-to-date model address cache. This cache lets Trap Director match trap sources with models. Trap Director uses matching to forward traps to destination models on remote landscapes. By keeping cache information current, Trap Director ensures that DX NetOps Spectrum can generate events for models regardless of their location.

NOTE
Trap Director performance can be affected when new landscapes are added to the distributed environment. Performance can also be affected when large numbers of traps are processed for many new models. Trap notification latency can increase in these circumstances.

How Trap Director Updates the Address Cache

Trap Director maintains a cache of the IP addresses and locations of models in the distributed environment. The address cache serves as an index to determine where to forward alerts. Models can be regularly added to landscapes, removed from landscapes, and moved between landscapes. Therefore, Trap Director also regularly updates the cache to keep its content current. Records that meet a retention period (or aging) threshold that you can specify are removed. Trap Director also performs cross-landscape searches for model IP addresses that are not available in the cache when it receives traps from those IP addresses.

The following steps describe how Trap Director determines the destination model for a trap:

1. The Trap Director server receives a trap.
2. Trap Director compares the IP address that was included in the trap to IP addresses in cache records.
3. If Trap Director finds a match, it forwards an alert to the model on the remote landscape.
   Otherwise, Trap Director performs the following tasks to determine the destination model and forward the alert:
   - Trap Director searches known landscapes for a matching IP address.
   - When it finds the destination model that is associated with the IP address on the remote landscape, Trap Director updates the cache with model information. It forwards an alert to the model.
4. DX NetOps Spectrum generates an event on the VNM model on the Trap Director server if the trap is dropped because the cross-landscape search did not find a matching address. The event indicates that the destination model was not found.
   A match is not found if, for example, the model has been deleted.

Trap Director in a Fault-Tolerant Setup

If you want to implement a fault-tolerant Trap Director setup, devices must be configured to forward traps to both the primary and the secondary SpectroSERVERs. The secondary server routes traps only when it detects that the primary server has failed. The secondary SpectroSERVER receives Trap Director settings during the primary backup, or synchronization process.

Trap Storm Settings

To handle trap storms, Trap Director uses trap storm settings that are configured on modeled devices. When Trap Director detects a trap storm, it stops forwarding alerts to models on remote landscapes. Trap Director also asserts trap storm alarms for the models.

Trap storms can originate from devices that are not modeled in DX NetOps Spectrum. For these storms, Trap Director uses the trap storm handling settings that are configured for the VNM model on the Trap Director server.

Enable and Disable Trap Director

You can enable and disable Trap Director on a SpectroSERVER.

> **NOTE**
> Use the OneClick Attribute Editor or the DX NetOps Spectrum Command Line Interface to set attribute values on the VNM model for a server.

Follow these steps:

1. Expand the Trap Management subview in the VNM model Information tab.
2. Click set in the Enable Trap Director field, and select Enabled.
   Trap Director is enabled.
3. (Optional) To disable Trap Director, click set in the Enable Trap Director field, and select Disabled.
   Trap Director is disabled.

Define the Cache Record Retention Period

You can control how frequently the Trap Director trap cache ages out. You can define the cache record retention period.

To define the cache record retention period, define a retention period for the following attribute:

```
trap_cache_age_out_minutes (0x12ad5)
```

Default: 180 (minutes)

Version Mismatch in DSS Environment

In a Distributed SpectroSERVER (DSS) environment, the infrastructure is organized geographically or across multiple servers in a single physical location. Using DSS, you can create a unified representation of the infrastructure that is composed of multiple landscapes, each with a local SpectroSERVER.

In the DSS environment, it is recommended to ensure all the landscapes are running on the same version of DX NetOps Spectrum (with the latest patches and BMPs). If there is a mismatch in the version on any landscape, you may see exceptions or reduced functionality problems. To overcome this issue, 10.3.1 is enhanced to raise an alarm on the Main Location Server (MLS) for the mismatch of the version in any landscape. The mismatch version information can be found in the OneClick Web server log.

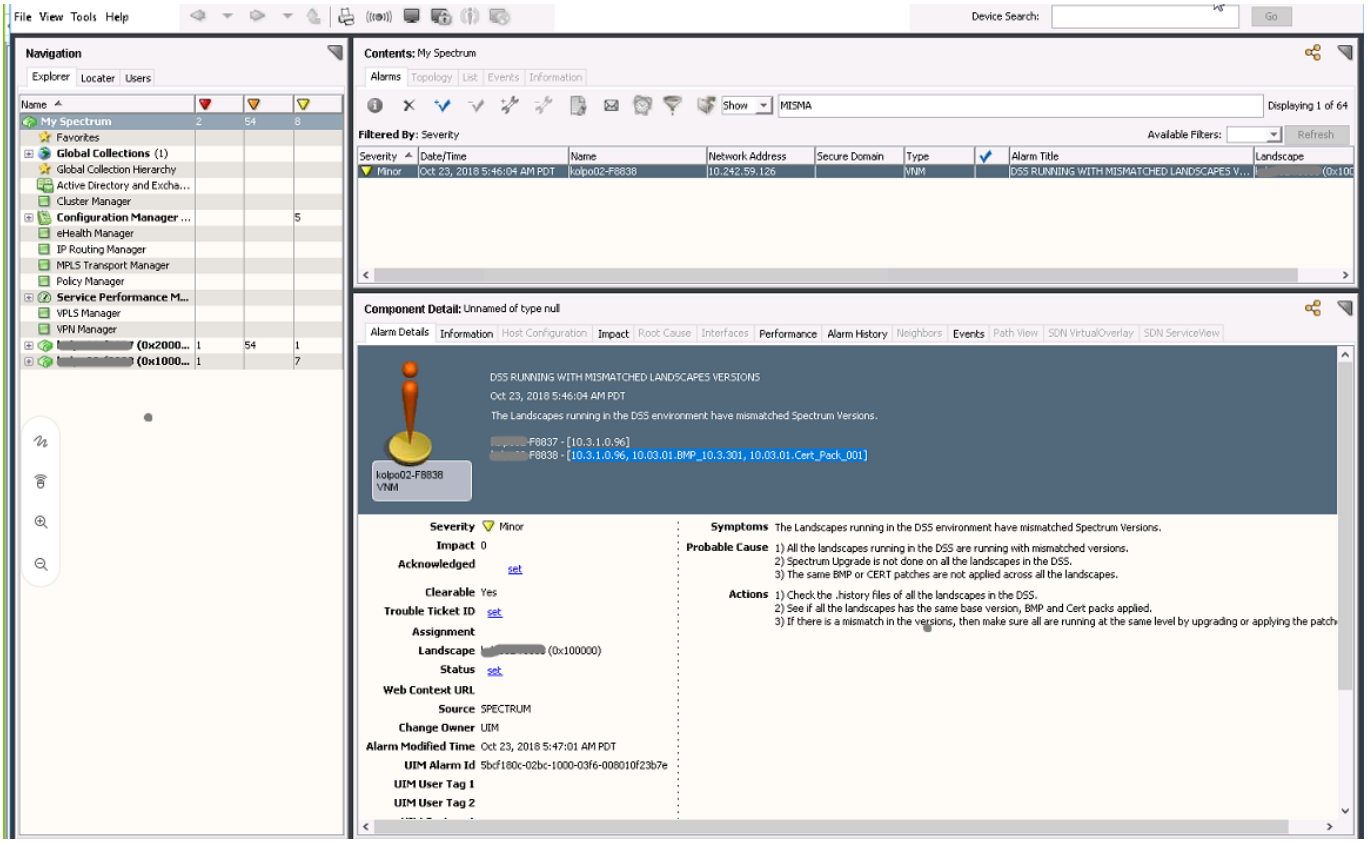
> **NOTE**
>
> Only SpectroSERVERs are compared for the version mismatch.

Alarm for Versions Mismatch

In the DSS environment, there are chances to miss upgrading of DX NetOps Spectrum on any landscape. DX NetOps Spectrum compares version across all landscapes and if there is any mismatch found then raises an alarm on the main location server (MLS). The Alarm Details tab in the VNM Model's Information tab shows the alarm details that are generated on the VNM model. Following is the alarm that is generated on the MLS.

"DSS RUNNING WITH MISMATCHED LANDSCAPES VERSIONS"

The alarm also shows the landscapes information which have the different versions running. The symptom, probable cause, and actions information for the alarm helps you to understand the problem and take corrective actions (as shown is the following image).



To resolve the problem, upgrade DX NetOps Spectrum (with BMP or Patch) according to the version displayed in the Alarm Details tab. After upgrade, if all the landscapes are running the same DX NetOps Spectrum version, an event is generated to clear the mismatch alarm.

Troubleshooting SpectroSERVER

Alarms are not synchronized when failing over back from Secondary to Primary SpectroSERVER

Issue:
In a Fault Tolerance environment, failover happens from Primary SpectroSERVER to secondary SpectroSERVER. After the Primary SpectroSERVER is back to normal, on the OneClick console, the alarms are not synchronized between the Primary and the secondary SpectroSERVER.

Probable Cause:
In order to have the alarm synchronization happen between the primary and secondary SpectroSERVER in the Fault Tolerance environment, you need to have the following entry in the $SPECROOT\SS\.vnmrc file on both the primary and secondary SpectroSERVER.

ftasv_enabled=true

Resolution:

By default, the entry 'ftasv_enabled=true' doesn't exist in the $SPECROOT\SS\.vnmrc file on the SpectroSERVER.

You need to manually add this entry to the $SPECROOT\SS\.vnmrc file on both the primary and secondary SpectroSERVER. If you modify this file while SpectroSERVER is running, the changes take effect only when the SpectroSERVER is restarted.

SpectroSERVER Performance Administration

In highly dynamic IT environments, monitoring the capacity of your systems and optimizing your DX NetOps Spectrum deployment cannot be a one-time task. Regular, periodic reviews are required to keep DX NetOps Spectrum operating optimally in larger environments (more than 1,000 monitored devices). For a well-developed, up-to-date summary of the performance and tuning best practices that we have derived from years of testing and supporting DX NetOps Spectrum, be sure to read Deployment Capacity and Optimization Best Practices , which is available on the Documentation Bookshelf.

DX NetOps Spectrum also provides self-monitoring. The OneClick client application lets Network Administrators monitor and troubleshoot a DX NetOps Spectrum-managed network. The Performance tab in the OneClick interface supports most network and device models. You can use this tab to analyze the CPU and memory utilization of a specific device.

For the VNM model that represents a SpectroSERVER, we also include a robust application to monitor performance. Performance View lets DX NetOps Spectrum administrators monitor the performance and system resource utilization of a SpectroSERVER. Use Performance View to identify performance problems and to determine the appropriate corrective actions to take. Your user account requires Performance Monitor privileges to deploy Performance View.

System Component Monitoring

Each computer system comprises four major components: disk, network, memory, and CPU. To ensure the successful operation of your DX NetOps Spectrum system, you can tune one or more of these components to eliminate bottlenecks.

Performance View includes the following two features to help you detect and locate bottlenecks:

- A series of tabs that provide information about the system components and SpectroSERVER activities that can affect performance. Check the Main tab for overall system and network activity at a glance.
- A health report feature that lets you run a report on the SpectroSERVER resources over a 24-hour period.

> **NOTE**
> In a distributed SpectroSERVER environment, you can switch the focus of Performance View from one SpectroSERVER to another.

For more information, see Getting Started with Performance View and Overview of the User Interface sections.

Performance Data Analysis

Often, the information in Performance View identifies the source of a performance problem. For instance, the CPU tab lists the 10 processes that are currently using the highest percentages of CPU time. However, exactly what constitutes a problem or bottleneck depends on the specific configuration of your DX NetOps Spectrum system and your network management priorities.

For guidelines on identifying performance problems and corrective actions, see Evaluating the Performance of a SpectroSERVER. You can also run a health report, which includes analysis of the performance data. For more information, see Running Health Reports .

Performance Optimization

You can typically resolve SpectroSERVER performance problems by tuning the SpectroSERVER to improve server performance. You can also add SpectroSERVERs to distribute the network load. The following topics discuss these options.

SpectroSERVER Tuning

Once you have identified the reasons for degraded performance, you can tune the SpectroSERVER to optimize performance. Tuning can include taking any of the following measures:

- Modifying the polling interval and the poll-to-log ratio of essential device models and application models, and disabling polling of non-essential models. Increasing the polling interval reduces the network traffic. As a result, latency, which degrades performance, can be reduced.
- Increasing the capacity of the system by increasing memory, CPU speed, or disks.
- Reducing the number of traps that are mapped to DX NetOps Spectrum events.
- Reducing the amount of data that is requested by customized watches and displayed attributes. Performance improves if you can reduce the amount of data that is requested from the SpectroSERVER and devices.
- Adjusting the usage of features such as Live Pipes, Discovery, and automatic device configuration.
- Adjusting client interactions with the SpectroSERVER. For example, reports that are generated using Spectrum Report Manager can exert a punctuated or prolonged performance burden on the server. The performance impact depends on the data that is reported and on reporting frequency. Command Line Interface (CLI) scripts, manual discoveries, and other manually initiated tasks can also affect SpectroSERVER performance.

Additional SpectroSERVERs

To determine whether increasing the number of SpectroSERVERs, rather than tuning, is the best means of achieving desired performance improvements, you can request a sizing of your DX NetOps Spectrum environment from CA Support. The sizing tool uses information about your network configuration to estimate the following:

- The additional network management traffic that DX NetOps Spectrum generates.
- The number and configuration of additional SpectroSERVERs that are required to efficiently manage the number of models in your environment.

Getting Started with Performance View

Start Performance View

Start Performance View by doing either of the following:

- In the Spectrum Control Panel, select SpectroSERVER Performance from the Control menu.
  This connects Performance View to the SpectroSERVER that owns the Control Panel.
- From a command prompt, navigate to the <*$SPECROOT*>/PView folder and enter **pview**.
  If you have set the Show Server List At Startup preference, you are prompted to select the SpectroSERVER to which to connect.

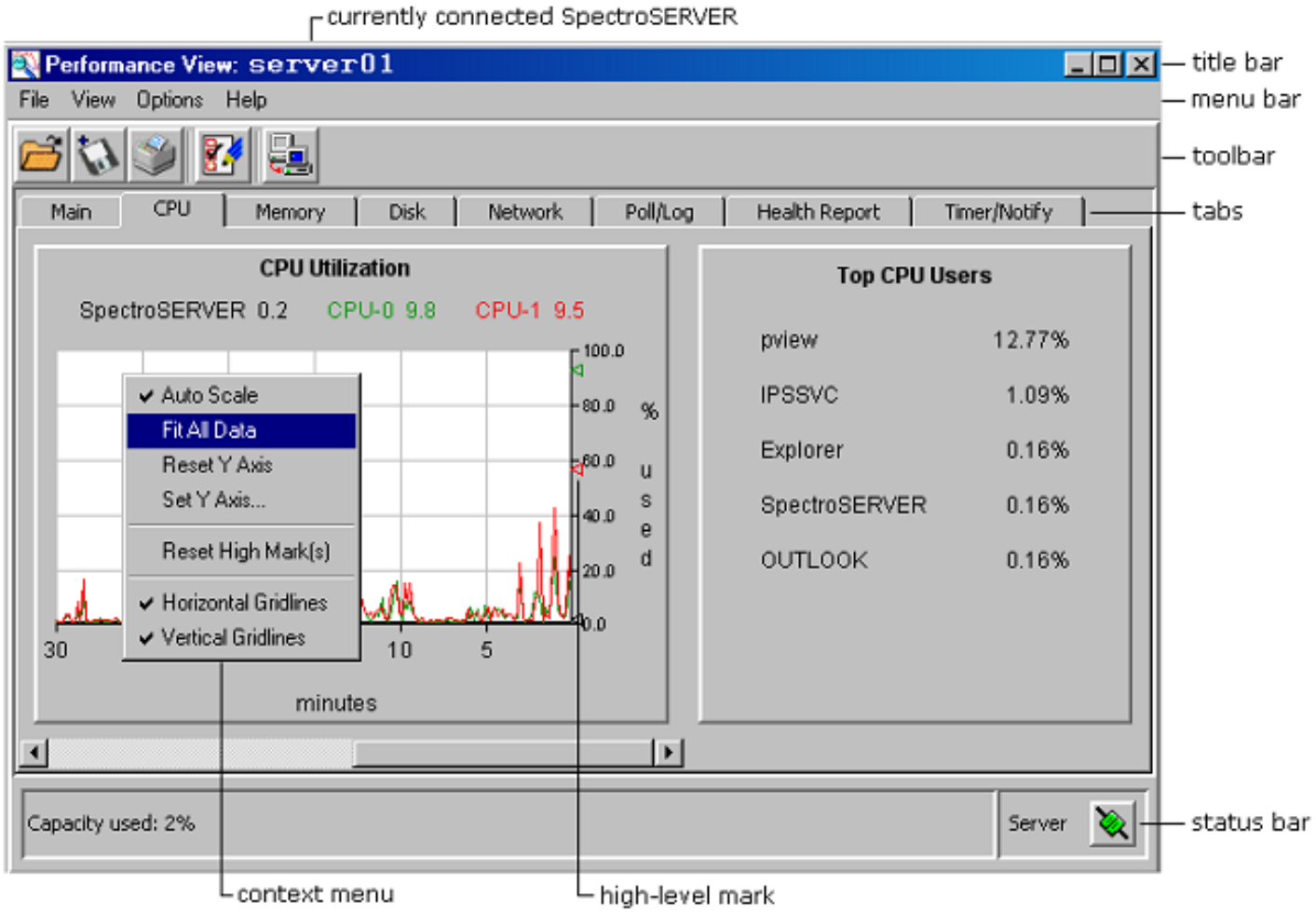
  NOTE
  In a distributed DX NetOps Spectrum environment, you can specify a SpectroSERVER when you start the application by entering the following command:

```
pview -vnm landscape_name
```

NOTE
If you are running DX NetOps Spectrum on Windows, the user who is running the SpectroSERVER process must belong to either the Windows Performance Monitor Users group or to the Administrators group.

Overview of the User Interface

Performance View has a single main window from which you access all performance information. The following image identifies the major user interface elements in the main window:



The status bar displays the current state of Performance View and, when applicable, the status of the current health report. The connection status icon is color-coded, indicating the status of the connection to the SpectroSERVER:

- **Green**
  Normal
- **Yellow**
  Using backup SpectroSERVER
- **Red**
  Contact lost

The tabs in the main window provide detailed information about SpectroSERVER performance. The information is presented in bar graphs, line graphs, and text.

Bar graphs appear on the Main tab. These graphs use data that is collected in 10-minute running averages. All other attributes and graphs use data collected every 10 seconds.

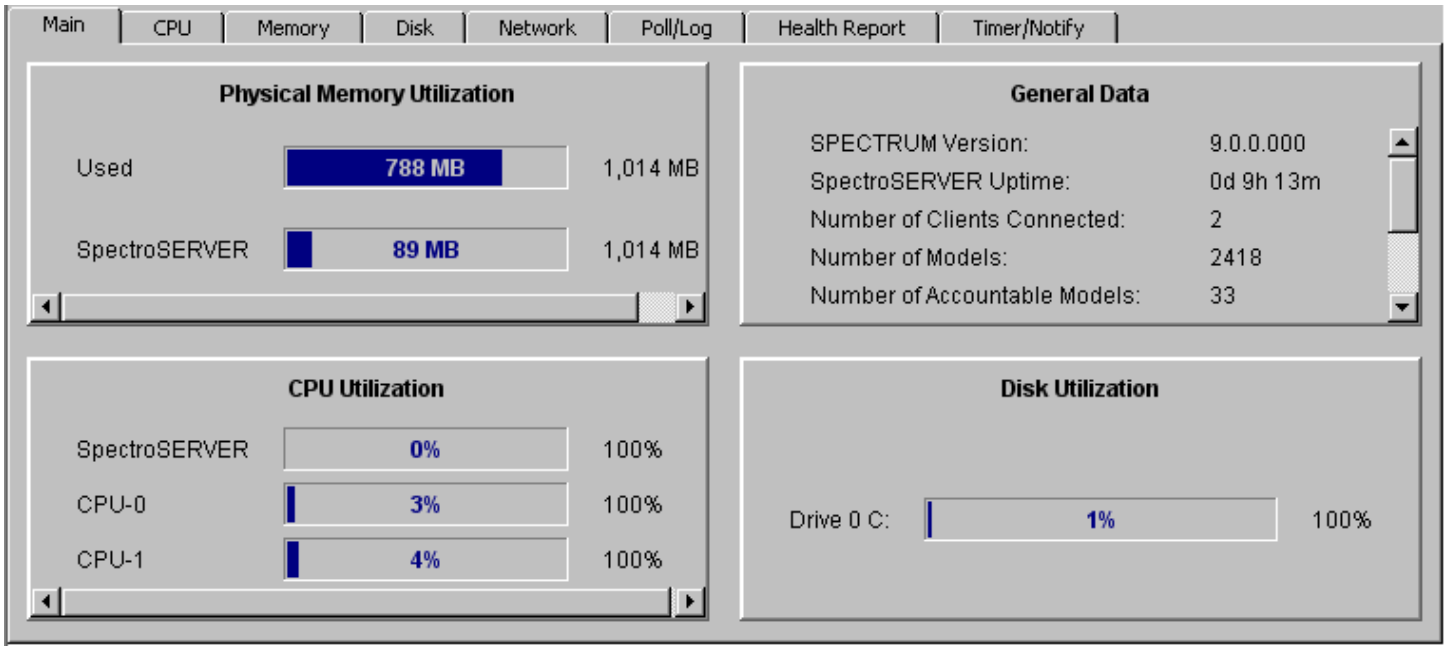
Line graphs on the CPU, Memory, Disk, Network, Poll/Log, and Timer/Notify tabs display data that is collected over 60 minutes. Only the most recent 60 minutes of data is displayed.

For single-line graphs, instantaneous values are shown in a text box. For multiple-line graphs, instantaneous values are shown in color-coded labels, where each label represents the instantaneous value of its associated line.

Line graphs also provide high-level marks and tooltips. High-level marks represent the highest data value that was collected since data collection began. You can reset the high-level marks and can make other changes to the axes of line graphs. Multiple CPUs and disk drives can be graphed in the same chart.

Main Tab

The Main tab contains a Physical Memory Utilization graph, a CPU Utilization graph, a Disk Utilization graph, and general data.



Check the Main tab for the following information:

- **Physical Memory Utilization**
  Displays the amount of physical memory that is used by the SpectroSERVER and other processes that are running on the server to which Performance View is connected.
- **CPU Utilization**

  NOTE
  Displays the percentage of total CPU processing power that is used by the SpectroSERVER and other processes that are running on the server.
- **Disk Utilization**
  Displays the amount of disk read/write access capacity that is used on the server.

Check the General Data panel for the following information:

- **DX NetOps Spectrum Version**
  Specifies the version of DX NetOps Spectrum that is installed on the server.
- **SpectroSERVER Uptime**

Specifies the amount of time that the server has been running. The format for Uptime is <days>d <hours>h <minutes>m.

- **Number of Clients Connected**
  Specifies the total number of clients that are connected to the selected SpectroSERVER.
- **Number of Models**
  Specifies the total number of models, including device models and other models.
- **Number of Accountable Models**
  Specifies the total number of models that are included in the device count calculation (used in DX NetOps Spectrum software licensing).
- **Number of Device Models**
  Specifies the total number of physical devices that are modeled with model types derived from the Device model type.
- **Number of Polled Models**
  Specifies the number of models whose Polling Status is set to TRUE and that have a non-zero polling interval.
- **Number of Polled Attributes**
  Specifies the current number of attributes that the SpectroSERVER is polling.
- **Number of Logged Attributes**
  Specifies the current number of attributes that the SpectroSERVER is logging.

NOTE
Polled attributes are used to determine whether a device is up or down (for fault isolation). Logged attributes are used to gather statistical information. Logging of polled attributes is optional, and logged attributes are not always polled.

CPU Tab

The CPU tab contains a CPU Utilization graph and information on the top CPU users.

- **CPU Utilization**
  Displays the SpectroSERVER usage as a percentage of the total CPU capacity. Each system CPU is included in the line graph.
- **Top CPU Users**
  Displays the top 10 CPU users from highest to lowest and the CPU utilization percentage for each.

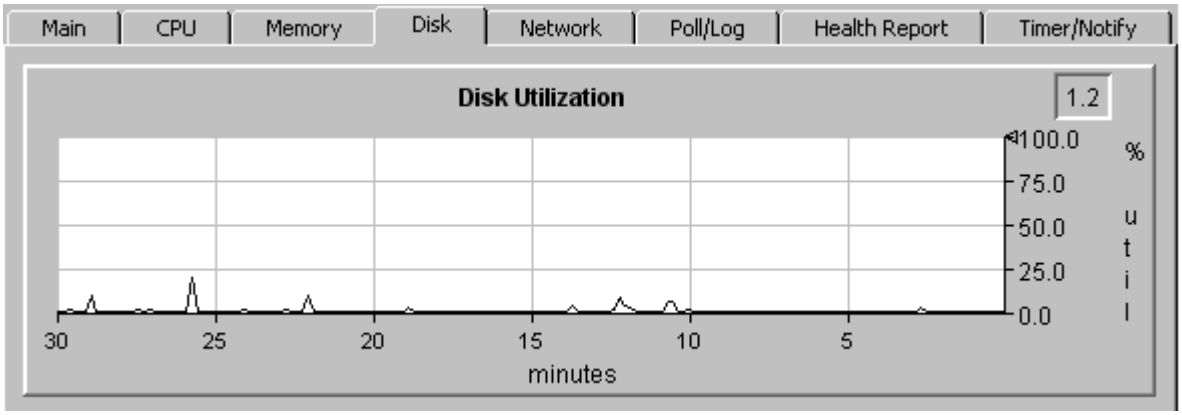
Memory Tab

The Memory tab provides the following information:

- **SS Memory Utilization**
  Displays the amount of memory that the SpectroSERVER is using.
- **Paging Activity**
  Displays total paging activity.
- **Top Memory Users**
  Displays the top 10 memory users (from highest to lowest) and the memory that each process is using.

Disk Tab

The Disk Utilization graph on the Disk tab displays the percentage of disk read/write capacity that is being used.



Network Tab

The Network tab contains a Network I/O graph and a Traps Received graph. Check this tab for the following information:

- **Network I/O**
  Reflects VNM read/write bytes only. This graph does not include traffic from any other sources.
- **Traps Received**
  Displays the number of unsolicited messages, such as SNMP traps, that the VNM receives.

Poll/Log Tab

The Poll/Log tab contains the following graphs:

- **Poll Latency**
  Displays the average poll latency, which is the interval in seconds between when a scheduled polling thread is expected to complete the actual completion time.
- **Poll Threads in Use**
  Displays the number of poll threads in use. A poll thread is allocated to every polling operation.
- **Log Latency**
  Displays the average log latency, which is the interval in seconds between when a scheduled logging thread is expected to complete and the actual completion time.
- **Log Threads in Use**
  Displays the number of log threads that are in use. A log thread is allocated to every logging operation.

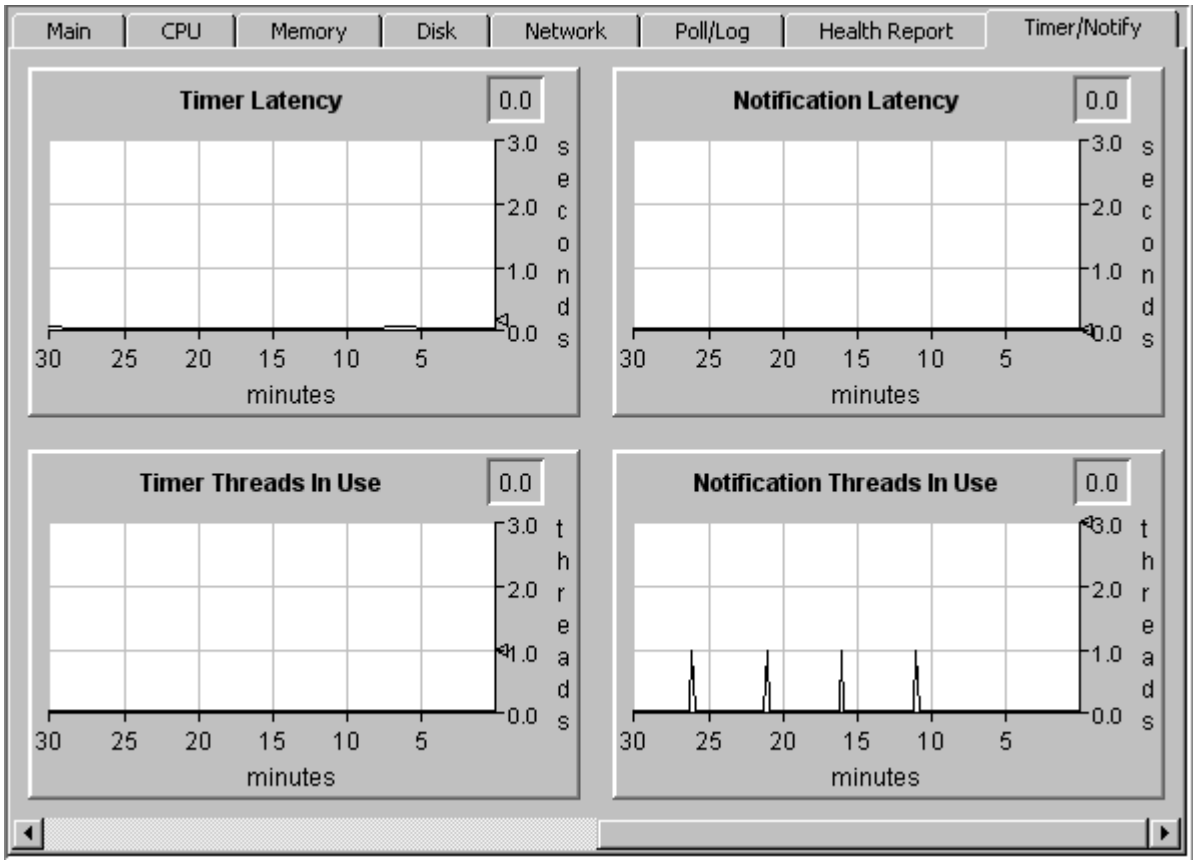
Health Report Tab

The Health Report tab displays status information about the current health report. After the data has been collected, the report and the average percentage of SpectroSERVER capacity during the reporting period are displayed on the tab.

Timer/Notify Tab

The Timer/Notify tab lets you monitor the performance of timer and notification threads.

Note: By default, this tab is not shown. To display it, you must select it from the View menu.



The graphs provide the following information:

- **Timer Latency**
  Displays the average timer latency in seconds. Time latency is the interval between when a scheduled timer thread is expected to complete and the actual time of completion.
- **Timer Threads in Use**
  Displays the number of timer threads in use.
- **Notification Latency**
  Displays the average notification latency in seconds. Notification latency is the interval in seconds between when a scheduled notification thread is expected to complete and the actual time of completion.
- **Notification Threads in Use**
  Displays the number of notification threads in use.

View Connection Details for the Connected SpectroSERVER

To determine the status of the connection between Performance View and the SpectroSERVER, examine the connection status

icon  .

The icon appears in the bottom right corner of the Performance View user interface.

The color of the icon indicates the status:

- **Green**
  Normal
- **Yellow**

Using backup SpectroSERVER

- **Red**
  Contact lost

To view additional connection details for the SpectroSERVER, click the connection status icon. A dialog provides a server connection log.

Connect the Performance View to a Different SpectroSERVER

In a distributed SpectroSERVER environment, you can change the SpectroSERVER to which Performance View is connected.

Follow these steps:

1. Click the Change SpectroSERVER) icon on the
   toolbar 
   The Select SpectroSERVER dialog opens.
2. (Optional) Filter the list of servers by taking one of the following steps:
   – To exclude servers whose names do not contain a specific text string, select Filter from the list. Enter the string in the text box.
   – To parse the list for servers whose names contain a specific text string, select Search from the list. Enter the string in the text box. The first server that matches the criteria is highlighted. Click Next to cycle through the servers that match.
3. Select the SpectroSERVER to connect to Performance View, and click OK.
   The Select SpectroSERVER dialog closes. The Performance View is now connected to the SpectroSERVER that you selected.

Set User Preferences

Select Preferences to customize the appearance and behavior of the Performance View user interface.

Follow these steps:

1. Click the Set user preferences
   icon 
   The Preferences dialog opens.
2. Configure preference settings as desired.
3. Click OK.
   The settings that you selected are applied to Performance View.

Configure Preferences

You can configure the following preferences from the Preferences dialog:

- **Save Settings At Exit**
  Saves all of the settings that you change during a Performance View session. Saved settings are applied to subsequent sessions. The following settings are available:
  – Performance View tabs to display and the order in which to display them

NOTE
You can show or hide tabs using the View menu. To modify the order of tabs, hide them and then display them in the desired order.

- Display settings for the status bar, the toolbar, and tooltips
- The last server that connected to Performance View
- The directory of the health report that was saved or opened most recently
- The size and on-screen location of the main window
- All other preferences that are specified in the Preferences dialog

- **Confirm Exit**
  Specifies whether you want to be prompted to confirm attempts to exit the Performance View.
- **Show Warnings At Exit**
  Specifies whether to show all pending warnings (for example, the warning when a health report has been created but not saved) before Performance View closes.
- **Show Server List At Startup**
  Specifies whether you want to be prompted to select the SpectroSERVER to connect to Performance View after you start Performance View.

  NOTE
  This setting only applies when you start Performance View from the command line. When you start Performance View from the Spectrum Control Panel, you are always connected to the SpectroSERVER to which the Control Panel is connected.
- **Email Report When Complete**
  Sends the health report automatically to the addresses that are specified in the Email Report To field.
- **Email Report To**
  Specifies a comma-separated list of the email addresses to which to send health reports after they are generated.
- **Title Font**
  Specifies the font, style, and size to use for the graph titles that appear above graphs. To change this preference, click Font, make your selections, and click OK.
- **Label Font**
  Specifies the font, style, and size to use for graph labels, which are the text elements in a graph other than the graph title. To change this preference, click Font, make your selections, and click OK.
- **Chart Line Colors**
  Specifies the colors to use for graph lines. The first color button specifies the color for the first graph attribute, the second button specifies the color for the second attribute, and so on.
  All graphs use this same color palette. Changing the color for a graph line changes that color in all graphs.

Change the Colors of Graph Lines

You can customize graph settings and change the color of a line in a line graph.

Follow these steps:

1. Click the Set user preferences icon 
   The Preferences dialog opens.
2. Beside Chart Line Colors, select the graph (chart) line color to change.
   The Select Color dialog opens.
3. Change the color by taking one of the following steps:

- To select a color swatch, click the Swatches tab, and select the swatch.
- To specify the desired hue, saturation, and brightness of the desired color, click the HSB tab, and specify the values using the slider or the text fields.
- To specify the red, green, and blue values of the desired color, click the RGB tab. Move the vertical slider along the color spectrum bar to change the hue, and move the small, white circle in the color square to change the saturation and brightness. Alternatively, use the text fields to individually specify the HSB values.

NOTE
To exit from the dialog without applying any changes, click Cancel. To return to the color that was active when you opened the dialog, click Reset.

4. Click OK.
   The Set Color dialog closes.
5. Click OK.
   The Preferences dialog closes and your changes are applied.

View Menu

Use the View menu to customize the Performance View user interface. Select menu items to display or hide interface elements such as the Status Bar, tooltips, or tabs.

Change the Display of Graph Axes

You can customize the axes of individual line graphs in several ways. Right-click in a graph to see a context menu. A check mark indicates that the option is enabled. The menu contains the following display options for graphs:

- **Auto Scale**
  Sizes the Y axis value of a graph to the highest value in the collected data. For instance, if the current vertical access value is 20, and data is collected for a value of 300, the Y axis is set to 300.
  Auto Scale overrides both Fit All Data and Set Y Axis.
- **Fit All Data**
  Selects a Y axis scale so that all current data can be displayed on the graph.
  Fit All Data overrides Set Y Axis.
- **Reset Y Axis**
  Sets the Y axis to the default values.
- **Set Y Axis**
  Lets you specify the maximum and minimum values for the Y axis and the number of divisions (equal intervals) between the maximum and minimum values.
- **Reset High Marks**
  Returns all high-level marks in a graph to zero. The data that is collected from the current point forward determines the new high-level marks. High-level marks represent the highest data values that were collected since data collection started.
- **Horizontal Gridlines**
  Shows or hides the horizontal gridlines.
- **Vertical Gridlines**
  Shows or hides the vertical gridlines.

Evaluate the Performance of a SpectroSERVER

Examining Thread Latency

The topics in this section describe how to determine whether thread latency indicates performance issues. The types of threads and indicators of latency are also discussed.

Threads and Thread Latency

DX NetOps Spectrum performance partially depends on timely thread allocation. A *thread* is a set of commands that perform a function or a set of functions. Each thread can run independently from other threads.

SpectroSERVER is a single-threaded application regarding the CPU, but a multi-threaded application internally. Within its own process, the SpectroSERVER creates and manages multiple threads that run simultaneously for tasks such as polling, logging, notifications, timers, and more.

> **NOTE**
> The Archive Manager runs in its own thread. As a result, you can use multiple CPUs on a server: one for the SpectroSERVER and another for the Archive Manager. However, deploying three or more CPUs can degrade performance.

In a multi-threaded context, while some threads are waiting (for user input, responses from devices, or data retrieval, for example), other threads can be running. A thread runs to log data, respond to traps, and to connect to SSAPI applications, for example. As each thread runs in the SpectroSERVER process, it takes control of the CPU for a few microseconds and then relinquishes control to allow other threads to run.

A SpectroSERVER maintains a pool of threads that are shared by the DX NetOps Spectrum processes that perform polling, logging, client requests, inference handler timer, inference handler notification, model activation, and model destruction. A SpectroSERVER subsystem uses threads from the pool - up to their individual limits - during periods of increased processing activity. These limits prevent any one SpectroSERVER subsystem from dominating resources and consuming all the available threads.

When the common pool of threads is exhausted, new threads are created. The pool grows to meet the needs of the increased activity. Threads that a process no longer requires are returned to the common pool for later use. When a thread remains unused for a specified period, it is removed from the pool, and its resources are returned to the system. This process is called aging.

Thread latency is the amount of time between when a thread is supposed to complete and when it actually completes. It can cause problems when the number of outstanding threads accumulates as the threads take more time to complete. If DX NetOps Spectrum runs for a prolonged period with high thread latencies, delays occur in device polling, logging, and other tasks. As a result, delays contribute to the response time. For example, if a critical network device became inoperable, a delay would occur before DX NetOps Spectrum notified a Network Administrator of the problem.

> **NOTE**
> Thread latency is a symptom, not a cause, of performance degradation.

Types of Threads That Affect Performance

Poll, log, timer, and notification threads can affect the performance of the SpectroSERVER. Performance View provides usage and latency statistics for these threads.

> **NOTE**
> By default, the Poll_Log_Ratio attribute on device models is set to 0, which effectively disables native DX NetOps Spectrum logging. To log device, attribute, and port statistics, we recommend using SSLogger instead of the native method, which writes the information to the Archive Manager database. SSLogger is a DX NetOps Spectrum command-line application that logs statistics directly to ASCII files. This type of logging reduces the load on the Archive Manager database and eliminates the need to export the data. SSLogger also provides increased control over the type and frequency of data that is logged.

NOTE
For more information about SSLogger, see SSLogger .

- **Poll threads**
  Poll devices on the network. DX NetOps Spectrum uses polling to manage the operation and performance of the network. The SpectroSERVER code that manages poll threads is named the Poll Manager.
- **Log threads**
  Log data from the network into DX NetOps Spectrum database archive files. DX NetOps Spectrum can use data logging to store information about the operation and performance of the network.
- **Timer threads**
  Notify inference handlers that have registered timers, also known as wake-up calls.
- **Notification threads**
  Notify inference handlers about changes in an attribute for which the inference handlers have registered.

Access the Thread Information View

Access the Thread Information view to see information about thread performance and status.

Follow these steps:

1. Click the Topology tab in the OneClick Console.
   The Topology opens.
2. Click the VNM icon for the SpectroSERVER.
3. Click the Information tab in the Component Details panel.
4. Expand the SpectroSERVER Control, Thread Information subview.
   The Thread Information view opens.

Thread Information View

The Thread Information subview appears within the SpectroSERVER Control subview in the OneClick Console. This subview shows the threads in use, threads available, and peak value for the threads within the SpectroSERVER process. While it shows a list of important threads that are used in DX NetOps Spectrum, the list is not exhaustive. Some of the threads available in this list include the following types:

- **Poll Threads**
  Used to read the polled attributes for a model on the Polling_Interval of that model.
- **Log Threads**
  Used to read and log the logged attributes for a model on the Polling_Interval * Poll_Log_Ratio.
- **Notification Threads**
  Used to send notifications of attribute changes to inference handlers and DX NetOps Spectrum client applications.
- **IH Timer Threads**
  Used to trigger timers in inference handlers.
- **Destroy Threads**
  Used to send model destruction notifications to inference handlers and client applications.
- **Model Activate Threads**
  Used to send model activation notifications to inference handlers and client applications.
- **Relation Activate Threads**
  Used to send relation change notifications to inference handlers and client applications.
- **Client Request Threads(*)**
  Used to handle client application requests.
- **Multi-Request Threads(*)**
  Used to handle multi-model requests that originate from inference handlers and client applications.

You can use the Thread Information view to change the available value for each thread type. However, the values are typically left at their defaults. If one of the thread types is consistently running at the limit, and CPU cycles are available, increasing a limit can reduce the latency. However, if CPU utilization is already above 80percent, increasing thread limits does not increase throughput. By contrast, the combination of high CPU utilization and increased thread limits can reduce throughput by increasing thread overhead.

If you find that all available threads are being used for one specific type of thread, contact CA Support for assistance.

Poll/Log Tab

The Poll/Log tab indicates whether the allocation of polling threads and logging threads is affecting the performance. The tab provides four graphs to report on thread usage and any associated latency:

- Poll Latency
- Poll Threads in Use
- Log Latency
- Log Threads in Use

Poll Threads In Use Graph

A poll thread is allocated to every polling operation. Poll threads are allocated from a finite number of DX NetOps Spectrum threads. The number of poll threads that the system uses at any one time is proportional to the number of models that are polled on the network and the polling rate for each model.

If the number of required poll threads exceeds the number of available threads, pending poll thread requests are queued until a poll thread becomes available. In such a case, the number of poll threads may be insufficient for the current state of the network. Contact CA Support for assistance.

Log Latency Graph

Log latency is the interval between when a scheduled logging thread is expected to complete and the actual completion time. The Log Latency graph shows the average latency for the logging process in seconds. For example, if the calculated log latency is 10 seconds, and data logging is set to occur every 60 seconds, the data is actually logged every 70 seconds.

Running for prolonged periods with high log latency can result in delayed logging and other serious performance problems. An average value in the Log Latency graph that is equal to or greater than 30 seconds can indicate that SpectroSERVER performance is degraded. Options for improving the performance include tuning the system, off-loading system demand, or upgrading system speed or capacity.

Log Threads In Use Graph

A log thread is allocated to every logging operation. Log threads are allocated from a finite number of threads resident in DX NetOps Spectrum. The number of log threads that the system uses at any one time is proportional to the amount of data that is logged.

If the number of required threads exceeds the number of available log threads, pending log thread requests are queued until a log thread becomes available. In such a case, the number of log threads may be insufficient for the current state of the network.

By default, the Poll_Log_Ratio attribute on device models is set to 0, which effectively disables native DX NetOps Spectrum logging. To log device, attribute, and port statistics, we recommend using SSLogger instead of the native method, which writes the information to the Archive Manager database. SSLogger is a DX NetOps Spectrum command-line application that logs statistics directly to ASCII files. This type of logging reduces the load on the Archive Manager

database and eliminates the need to export the data. SSLogger also provides increased control over the type and frequency of data that is logged.

NOTE
For more information about SSLogger, see SSLogger .

Poll Latency Graph

Poll latency is the interval between when a scheduled polling thread is expected to complete and the actual completion time. The Poll Latency graph shows the average latency for the polling process in seconds. For example, if the calculated poll latency is 10 seconds, and the polling interval for a model is every 60 seconds, the model is actually polled every 70 seconds.

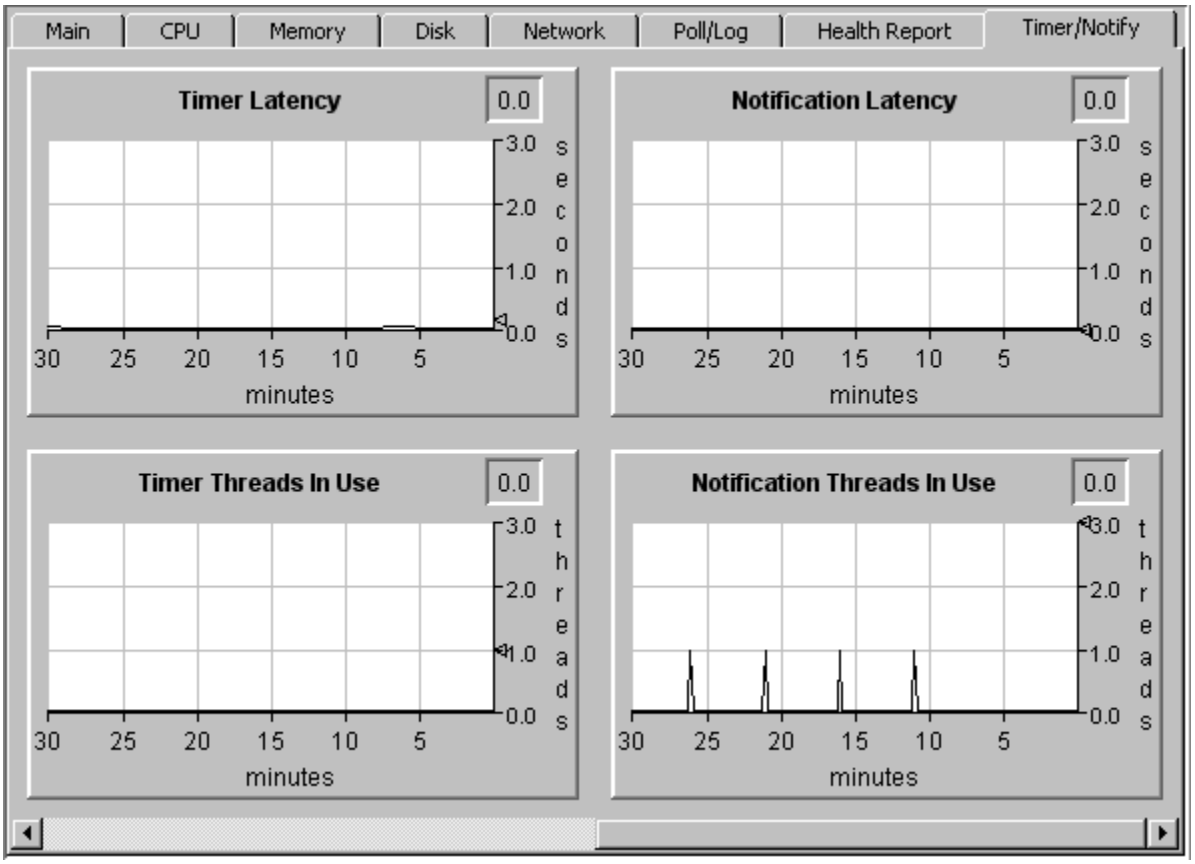
Running for prolonged periods with high poll latency can result in delayed device polling. Delaying the time to poll devices can increase the response time associated with detecting a network fault. A device could go down, and the system administrator would be unaware.

An average value in the Poll Latency graph that is equal to or greater than three seconds and that is sustained for a considerable period can indicate degraded SpectroSERVER performance. Options for improving the performance would include tuning the system, off-loading system demand, or upgrading the speed or capacity of the system.

Timer/Notify Tab

The Timer/Notify tab displays the number of timer threads or notification threads in use and any associated latency. The tab has four graphs:

- Timer Latency
- Timer Threads in Use
- Notification Latency
- Notification Threads in Use



Theory of Operations

The SpectroSERVER is a poll-driven and event-driven system. Each SpectroSERVER actively polls managed elements for state changes, generates events for these changes, and notifies the inference handlers that have registered for the events. Events for which an inference handler can register include model creation, model destruction, attribute value changes, association creations and destructions, and others.

Inference handlers are code segments that are associated with a model type to define the behavior of the model type. Inference handlers execute on behalf of instantiated models of the model type. Either notification threads or timer threads can trigger these handlers.

Notification Threads

A SpectroSERVER process named the Notification Manager reads attribute changes received from polled devices and runs inference handlers to process the data. Another SpectroSERVER process, the Poll Manager, detects changes in attributes whose poll flags are set.

When a change in an attribute value is detected, the Poll Manager alerts the Notification Manager, which forwards the events to each registered inference handler.

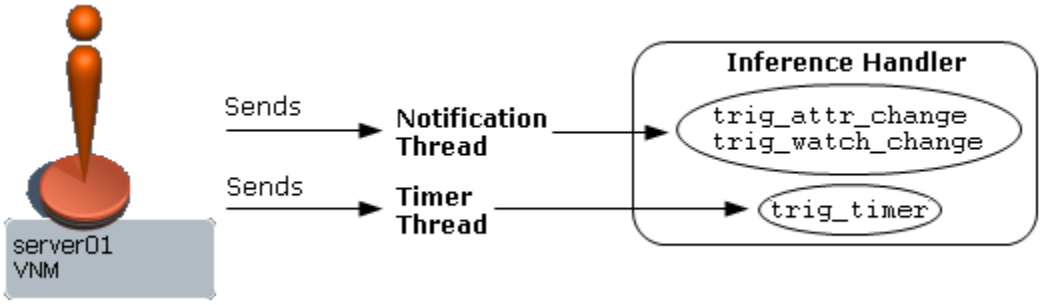
Notification threads are the mechanism that is used to notify inference handlers of a change in an attribute. These threads are used to run the trig_attr_change and trig_watch_change inference handler methods. Both of these methods are notifications of attribute value changes.

For example, assume that the ifInDiscards attribute has changed from a count of 110 to 150. The SpectroSERVER sends a notification thread to the inference handler that has expressed an interest in this attribute. The inference handler then runs the trig_attr_change method.

Timer Threads

Timer threads are used to notify inference handlers that have registered for timers (or "wake-up calls"). When an inference handler registers a “wake-up call” with the SpectroSERVER, the SpectroSERVER uses a timer thread to run the trigger method for that inference handler after a specified interval. Timer threads are used to run the trig_timer inference handler method in the same way that notification threads are used to run the trig_attr_change and trig_watch_change inference handler methods.

For example, assume that the primary address of a router becomes non-operational, and a secondary address must be used. An inference handler registers with the SpectroSERVER for a timer so that it can determine when the primary address is again operational. The inference handler then runs the trig_timer method.



Timer Latency Graph

Timer latency is the interval between when a scheduled timer thread is expected to complete and the actual completion time. The Timer Latency graph shows the average latency for the timer process in seconds. For example, if the average timer latency is 10 seconds and an inference handler has registered for a timer thread every 60 seconds, the timer thread actually activates the corresponding trigger method for the inference handler every 70 seconds.

Running for prolonged periods with high timer latency can result in the delayed activation of inference handler triggers and, therefore, delayed network monitoring. A Timer Latency graph that shows more than three seconds sustained indicates performance problems. Contact CA Support for assistance.

Timer Threads In Use Graph

The Timer Threads in Use graph displays the number of timer threads in use.

If the number of needed timer threads exceeds the number of available timer threads, pending timer thread requests are queued until a timer thread becomes available. If this is the case, the number of timer threads may be insufficient for the current state of the network. In this case, call CA Support for assistance.

Notification Latency Graph

Notification latency is the interval between when a scheduled notification thread is supposed to complete and when it actually completes. The Notification Latency graph shows the average latency for the notification process in seconds. For example, if the average notification latency is 10 seconds, and an inference handler has registered for a notification thread every 60 seconds, then the inference handler's corresponding trigger method is actually being activated by the notification thread every 70 seconds.

The effect of running for prolonged periods with high notification latency is the delayed activation of inference handler triggers and, therefore, network monitoring.

Notification Threads in Use Graph

The number of notification threads that get used by the system at any one time is proportional to the number of attribute or watch changes occurring on the network. Once a network is up and has achieved stability, the number of notification threads required to monitor the system should remain steady and small.

If the number of required notification threads exceeds the number of available threads, pending notification thread requests are queued until a notification thread becomes available. In such a case, the number of notification threads may be insufficient for the current state of the network. Contact CA Support for assistance.

Examining Memory Usage

The topics in this section describe how to determine whether swapping or paging activities are causing performance problems. *Swapping* or *paging* is a processing technique that involves bidirectional transfers of data from the main storage area to an auxiliary storage area. For example, swapping occurs from memory to disk. *Pages* refer to the individual units of data transfer that are used to swap data.

Considerable swapping activity indicates a shortage of system memory. When memory resources are insufficient, data must be temporarily transferred from memory to disk to make room for various processes to run. A large amount of paging activity indicates a high amount of swapping. Because the use of the disk is much slower than the use of physical memory, paging can result in performance problems. Paging should, therefore, be minimized.

The swap space and the physical memory collectively compose the available memory or virtual memory. Virtual memory is frequently a bottleneck for overall system performance. A system typically requires two times the amount of physical memory configured as swap space. If this space is low, reconfigure the system with more memory and swap space.

The following information in Performance View can help you determine whether insufficient memory has affected SpectroSERVER performance:

- Physical Memory Utilization graph on the Main tab
- Disk Utilization graph on the Disk tab
- Paging Activity graph on the Memory tab
- SS Memory Utilization graph on the Memory tab

In addition, you can monitor the memory usage of the SpectroSERVER process in OneClick. Events are logged and alarms are triggered when defined threshold values are exceeded. These thresholds are described in Using Performance Thresholds.

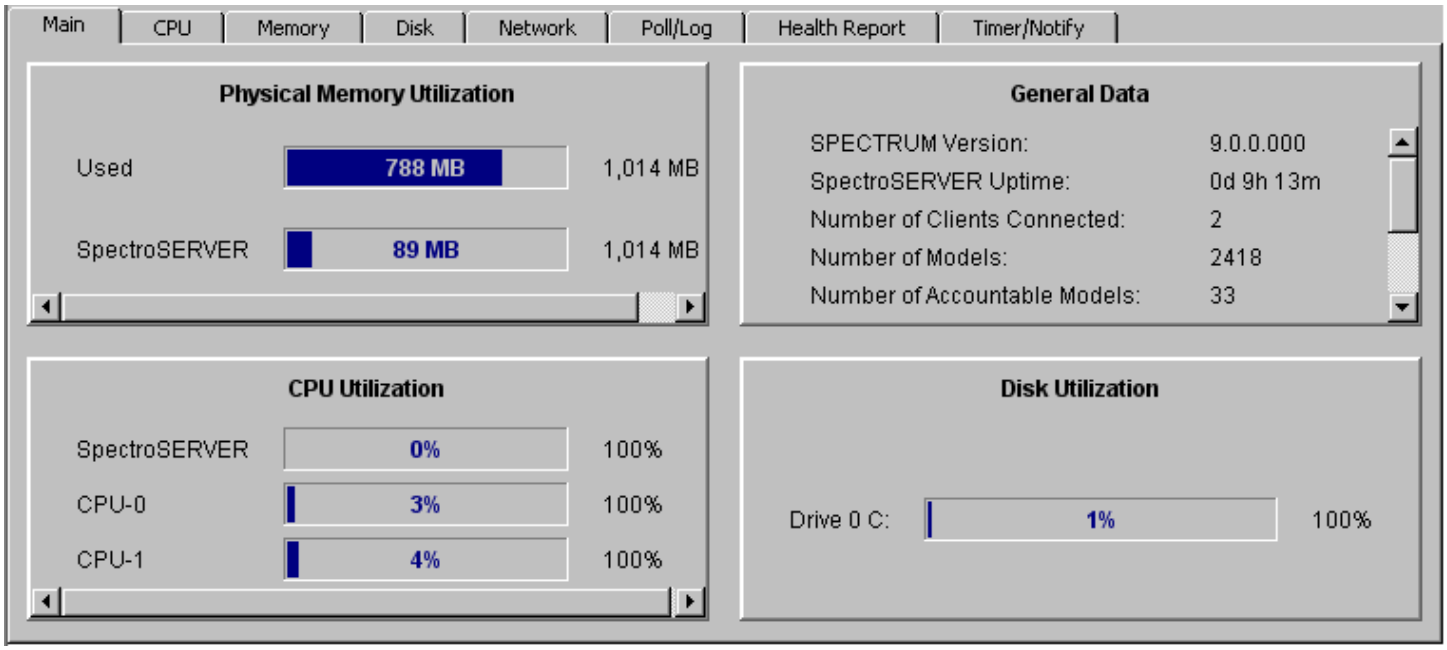
Indicators of Memory Problems

Memory management is important for achieving and maintaining performance. A memory shortage has an immediate and significant effect on SpectroSERVER fault detection response times.

One direct consequence of running DX NetOps Spectrum with insufficient memory is that it appears disk I/O bound. A system that lacks sufficient memory can appear to be disk bound because of the high paging and swapping activity occurring on the disk. When memory is the primary bottleneck, you must either increase the amount of memory or reduce the demand for memory to restore acceptable performance. Adding disk capacity or speed has a negligible effect.

Physical Memory Utilization - Main Tab

The Physical Memory Utilization area of the Main tab displays the amount of physical memory that the system is using and the amount of memory that DX NetOps Spectrum is using.



The total physical memory statistic is the actual amount of physical memory that the server to which the Performance View is connected contains. Therefore, this value is system-dependent. If SpectroSERVER uses a large percentage of the virtual memory, consider upgrading the memory or allocating more swap space.

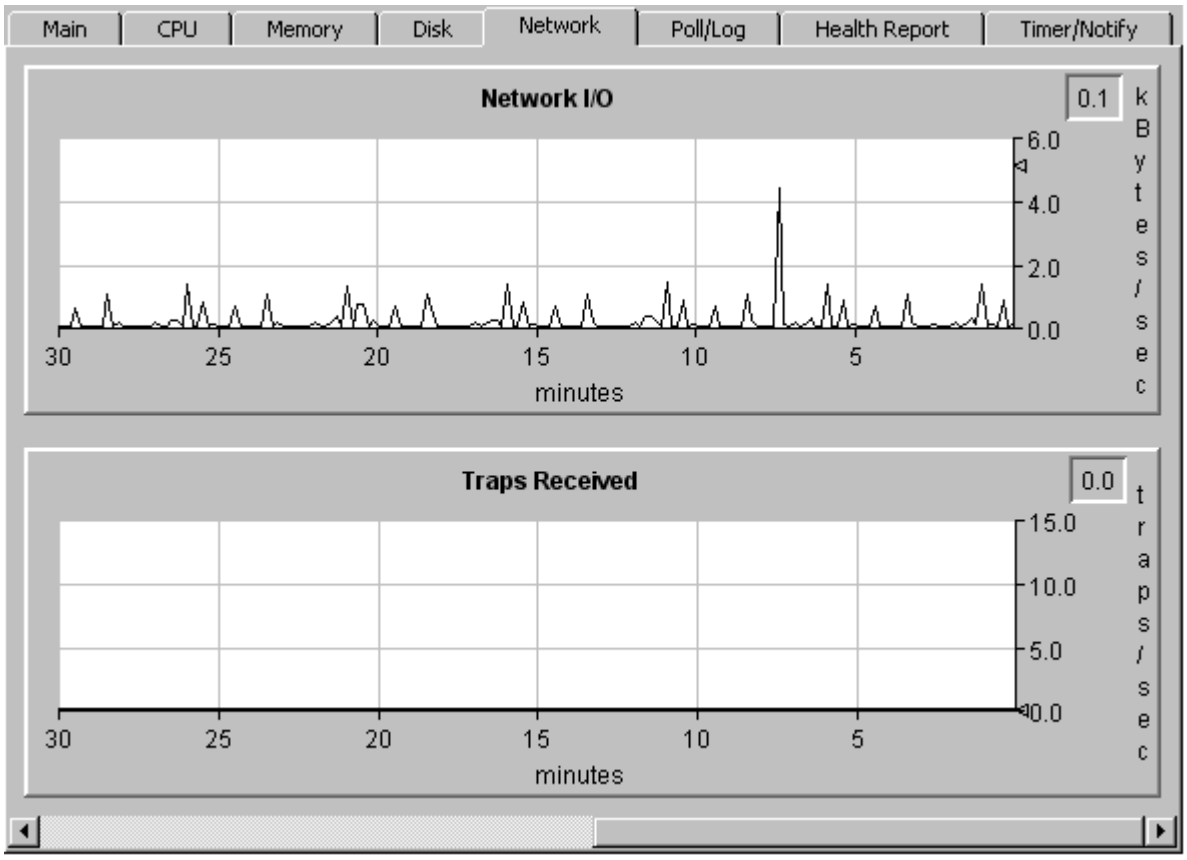
Disk Utilization Graph - Disk Tab

The Disk Utilization graph records all disk transfers, including all physical disks that are attached to the system. Disk utilization refers to how busy the disk is - that is, the percentage of time that the disk is used.

If the Disk Utilization graph is high, data logging can also be high. If this graph shows continuously high numbers, consider changing the logging ratio of some of your models and running the PMCount utility.

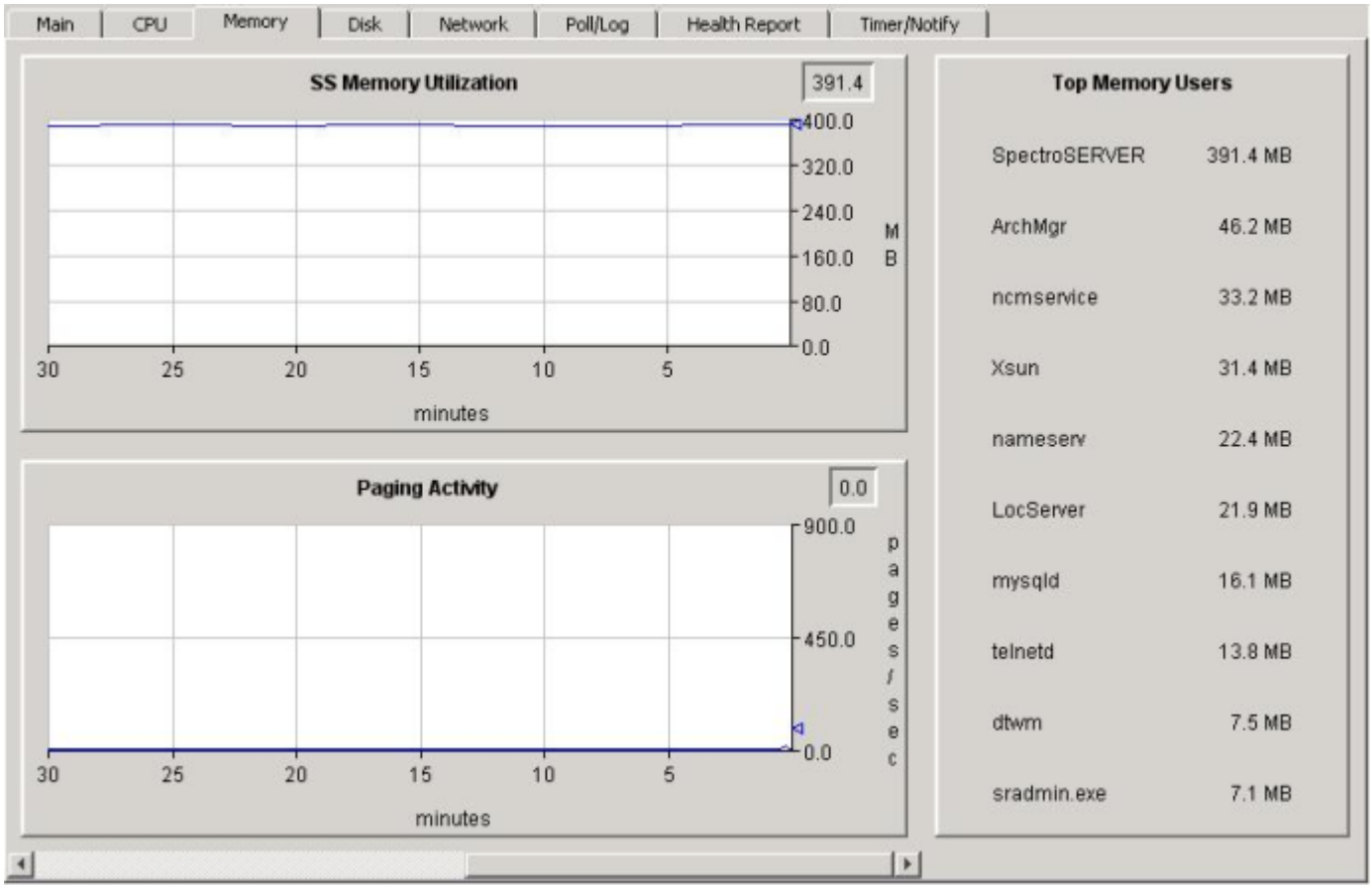
Network I/O Graph - Network Tab

The Network I/O graph records network I/O from the Ethernet interfaces on the system. The data includes I/O activity from the SpectroSERVER. Expect to see an increase when models are created in the database or polling intervals are changed.



Paging Activity Graph - Memory Tab

The Paging Activity graph displays the number of system pages over time. Values in this graph that are persistently high indicate a lack of adequate physical memory. Such a situation can result from having more processes running than the available physical memory can accommodate.



A persistently high value indicates that the system is heavily loaded. Consider a memory upgrade. If you see high paging activity levels, consider reducing the number of non-DX NetOps Spectrum processes that are running or increasing the physical memory.

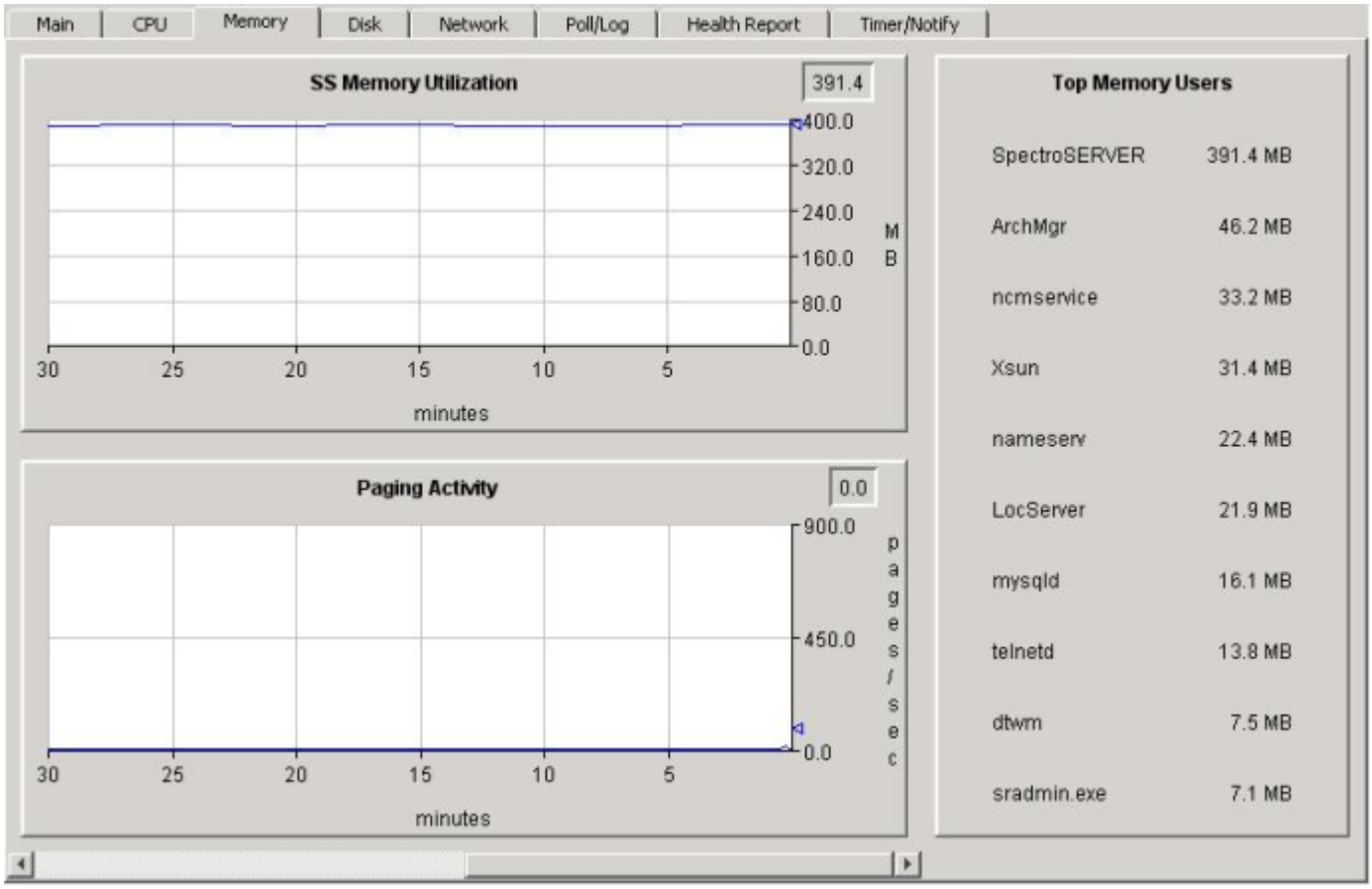
Examining the Application Load

This section describes how to determine whether too many applications are running on the system. These applications include SpectroSERVER and OneClick. The following views can help you determine whether an excessive number of applications that are running on the system are causing performance issues:

- Memory tab
- CPU tab

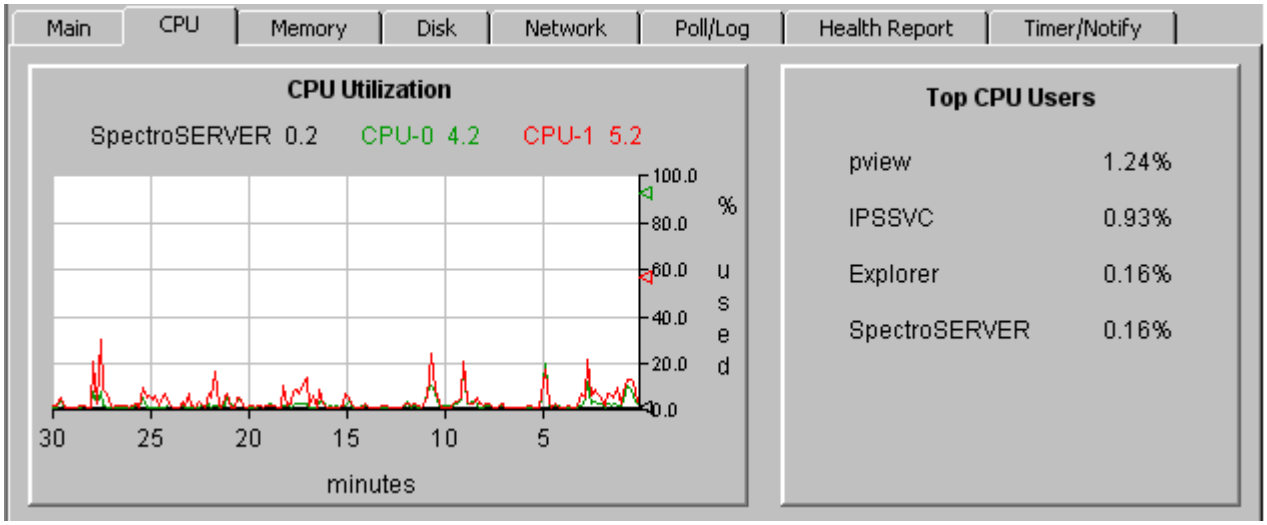
Memory Tab

The Memory tab displays the top users of system memory. This view can help you determine which applications are consuming the most system memory.



CPU Tab

The CPU tab displays the top users of the system CPU. This view can help you determine which applications are consuming the most system CPU.



If SpectroSERVER CPU Utilization is low, but DX NetOps Spectrum performance is still slow, consider whether other applications are straining the system.

Examining the Number of Connected Clients

A large number of client connections can place a heavy load on the server. To identify the number of active client connections, click the Main tab. In the General Data panel, check the value for Number of Clients Connected.

> **NOTE**
> A OneClick web server represents one SpectroSERVER client, regardless of how many OneClick clients are running against that web server.

We recommend installing only DX NetOps Spectrum applications on the SpectroSERVER host. But installing OneClick on a single-CPU SpectroSERVER host system can degrade the performance of both the SpectroSERVER and OneClick. Installing OneClick on a separate, dedicated system can maximize the performance of both the SpectroSERVER and OneClick.

Using Performance Thresholds

In addition to the performance monitoring capabilities of Performance View, DX NetOps Spectrum OneClick also provides some system performance statistics. You can also monitor the SpectroSERVER process in OneClick. If defined threshold values are exceeded, events are logged, and alarms are triggered. The following performance metrics are monitored using DX NetOps Spectrum performance thresholds:

- SNMP traps
- Memory usage

SNMP Traps

When excessive SNMP trap rates affect the SpectroSERVER process, performance can degrade. If the threshold rate is exceeded, events and alarms are generated on the SSPerformance and VNM models.

> **NOTE**
> The default trap rate threshold is 100 traps per second. To change it, modify the value in the EventDisp file in the following location:

```
$SPECROOT/SS/CsVendor/Cabletron/EventDisp
```

> **NOTE**
> The following line controls the SNMP trap rate value, which is currently set to 100.0 (traps per second):

```
"{v 0x11eca} >= {R 100.0 }",       "0x00010f92 -:-", \
```

> **NOTE**
> The rate threshold must be exceeded for at least 300 seconds to trigger an alarm. This time value cannot be changed.

The trap rate is monitored by the attribute vnm_snmp_traps_ps (AttrID = 0x11eca) on the SSPerformance model.

Memory Usage

If the SpectroSERVER process has an increased memory size, the SpectroSERVER is at risk of termination due to memory exhaustion. If a specified threshold rate for either physical or virtual memory is exceeded for a specified period, events and alarms are generated on the SSPerformance and VNM models.

Separate threshold values are used for physical memory and virtual memory. The default memory size for each threshold is 2.5 GB. To change either threshold, modify the value in the EventDisp file in the following location:

```
$SPECROOT/SS/CsVendor/Cabletron/EventDisp
```

NOTE
The following lines control the memory sizes, which are set to 2.5 GB:

```
"{v 0x11e8b} >= {R 2500000000.0}", "0x00010f95 -:-", \     <- physical memory
"{v 0x12e62} >= {R 2500000000.0}", "0x00010f98 -:-"        <- virtual memory
```

NOTE
The rate threshold must be exceeded for at least 300 seconds to trigger an alarm. This time value cannot be changed.

Run Health Reports

Health Reports

You can use the Performance View Health Report feature to measure and report on the health of your SpectroSERVER and the system where it is installed. Report options let you select any time period, from 6 to 24 hours.

Start the reporting process at any time, either from the main Performance View window or from the command line. The command line option also lets you take advantage of native scheduling service on your host system to run the report automatically at regular intervals.

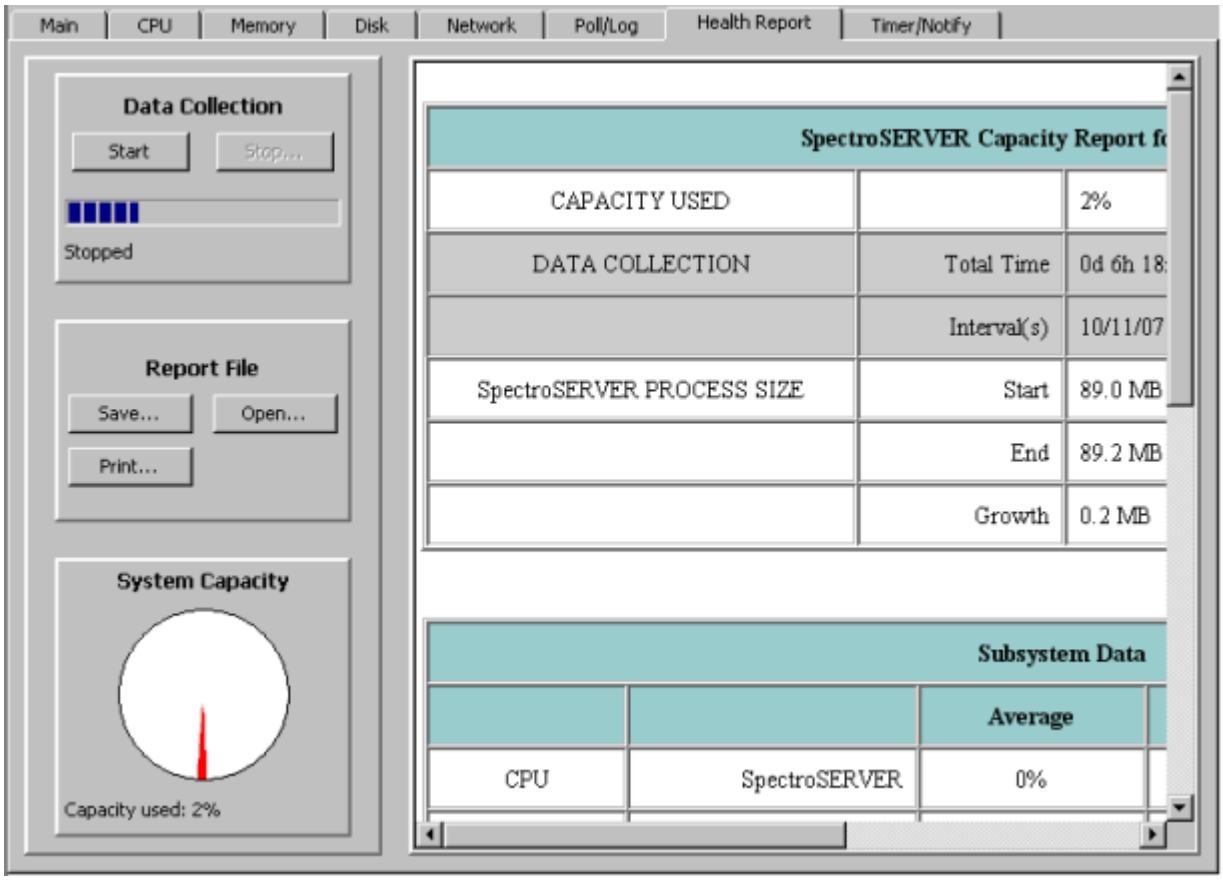
By default, Health Report collects the following data at 10-second intervals for a 24-hour period:

- CPU, disk, and memory usage data
- network I/O and trap data
- poll, log, timer, and notification latency data

We recommend selecting the full 24-hour time period. This interval collects data from a typical day. Or, if data that represents your typical workload is collected during a shorter period of time, select that time period. However, be careful not to exclude data collection for jobs that run during non-business hours, such as backups that are executed late at night.

Once data collection has occurred for a full 24-hour period, a health report is generated automatically. If you ran the report from the Health Report tab, the report is displayed there. You can save it to a location of your choice. If you ran the report from the command line, the report is written to a file.

Note: Use the Preferences dialog to configure Performance View to automatically send health reports to a list of email addresses.



The relative health of each system resource for example, CPU usage is determined by analyzing the Average, Peak, and % Over Critical Value readings collected. Any values that exceed predefined thresholds are flagged (displayed in red) to indicate a potential performance problem.

The relative health of the SpectroSERVER is determined by applying performance algorithms to the collected data. If it is determined that performance has degraded, the likely causes and recommendations for improving performance are also provided in the report.

Start Data Collection

To enable health reports, you must start data collection for the Health Report feature.

Follow these steps:

1. Click the Health Report tab.
2. Click Start in the Data Collection section.
   The message area in the panel indicates the number of hours and minutes remaining in the default 24-hour reporting period. After all of the data has been collected, Health Report analyzes the data and displays a health report to the right of the Data Collection panel.

   NOTE
   As long as the new report remains on display, the average percentage of SpectroSERVER capacity that is used during the reporting period is also shown in the graph in the System Capacity panel.

Stop Data Collection

Data collection for a Health Report stops automatically after data has been collected for 24 hours. However, you can stop or pause data collection manually at any time.

Follow these steps:

1. Click the Health Report tab.
2. Click Stop in the Data Collection section.
3. (Optional) Select *one* of the following options if you started the data collection less than 24 hours ago:
   - **Resume Data Collection**
     Restarts data collection. For example, if you stopped data collection after one hour, select this option to restart data collection and continue it for 23 more hours (until the default reporting period of 24 hours has been reached). In other words, the total time of data collection does not have to be contiguous. You can start and stop data collection for the same report as many times as you want. However, to generate a report, you must collect data for at least six hours.
   - **Stop and Analyze Data**
     Immediately generates a health report from the collected data. This option appears if data has been collected for at least 6 hours. The report remains displayed until you start data collection for a new report or exit Performance View.

     NOTE
     You cannot resume data collection for the same report once you have clicked this button.
   - **Stop and Delete Data**
     Ends the data collection process and deletes all collected data. No report is generated.

Save Health Reports

You can save the current health report to preserve the data.

Follow these steps:

1. Click the save the current health report icon  .
   The Choose Directory and Filename for HTML Report dialog opens.

   NOTE
   When specifying a filename for the report, do not include the .htm file extension. It is added automatically.
2. Navigate to the folder in which to save the report, enter a filename, and click Save.
   The health report is saved.

Open Health Reports

Open health reports to view or print them.

Follow these steps:

1. Click the Open a previously saved health report  .
   The Choose Report File to Open dialog opens.
2. Navigate to the report, select it, and click Open.
   The report is displayed in a separate, read-only window.

Print Health Reports

You can print the health report that is currently displayed on the Health Report tab.

Follow these steps:

1. Click the print the health report 
   The Print dialog opens.
2. Select settings in the Print dialog, and click OK.
   The health report prints.

You can also print a health report that you have saved.

Follow these steps:

1. Open the report.
   The SpectroSERVER Capacity Report dialog opens, displaying the selected report.
2. Click Print.
   The Print dialog opens.
3. Select settings, and click OK.
   The health report prints.
4. Click Close.
   The SpectroSERVER Capacity Report dialog closes.

Run Health Reports from the Command Line

You can start data collection for a health report at any time by entering the desired parameters from the command line. Or you can use your native scheduling service to execute the command at a specified time or at regular intervals.

The command line executable is named pviewrep and is located in the <*$SPECROOT*>\PView directory.

The syntax for the pviewrep command is as follows:

```
pviewrep vnm -c collectTime -e addrList
```

- **vnm**
  Specifies the name of the SpectroSERVER for which to run a report.
- **-c *collectTime***
  Specifies the number of hours for which to collect data.

  NOTE
  The minimum time period for a report is six hours. If you specify fewer than six hours, the report still collects six hours of data before it is generated.
- **-e *addrList***
  Specifies a comma-separated list of email addresses to which to send the completed report. To specify multiple addresses on Windows systems, enclose the list in quotation marks, for example, “address1,address2,address3”.

  WARNING
  In a Windows environment, you must have the Windows Messaging Subsystem or Messaging Application Programming Interface (MAPI) subsystem installed to be able to send messages using the -e option. If the subsystem is not installed, the executable fails to send the email notification. It looks for a registry entry under HKEY_CURRENT_USER\\Software\\Microsoft\\Windows NT\\CurrentVersion\\Windows Messaging Subsystem\\Profiles and creates an application event if the entry cannot be found. The local email system can require confirmation steps before sending the email.

Reports that pviewrep generates are saved automatically with the name of the SpectroSERVER host. These files have an .htm extension. Sequential numbers are added to keep subsequent reports from overwriting existing ones. For example, the first report that is generated for a server named “ace” is ace.htm. The second report is ace_1.htm.

By default, reports that are generated with pviewrep are saved to the directory that was used for the most recent health report from the Performance View main window. If that directory is not available, the report is saved to the current working directory. If the file cannot be written to the current working directory, it is written to the standard output stream for the program.

NOTE
Health reports that are distributed automatically by email are in plain text format, not HTML.

Interpret Health Reports

A health report includes three major sections, which are described in the following topics:

- SpectroSERVER Capacity
- Subsystem Data
- Analysis

SpectroSERVER Capacity

The SpectroSERVER Capacity section of a health report provides the following information:

- **Capacity Used**
  Reports the average percentage of SpectroSERVER capacity being used during the data collection period. This percentage is also shown graphically in the System Capacity panel on the Health Report tab.
- **Data Collection**
  Reports the total amount of time that data was collected. Also reports the start and end times for the individual data collection intervals that make up the total time.
- **SpectroSERVER Process Size**
  Reports the amounts of RAM that the SpectroSERVER used at the beginning and the end of the data collection period, and the difference (Growth) between the two values.

Subsystem Data of a Health Report

The Subsystem Data section of a health report provides Average, Peak, and % Over Critical Value readings for individual parameters within various performance categories (CPU, LATENCIES, DISK, MEMORY, and NETWORK).

Any Average or % Over Critical Value reading that exceeds the Performance View threshold value for that parameter is flagged (displayed in red). Flagged parameters indicate possible performance problems. These parameters contribute to the message that appears in the Analysis area of the report.

The Average, Peak, and % Over Critical Value columns are defined as follows:

- **Average**
  The sum of all of the values for the parameter during the total data collection period (Total Time) divided by the number of collection points. A collection point occurs every 10 seconds.
- **Peak**
  The highest value for a parameter during the total data collection period.
- **% Over Critical Value**
  The percentage of the data collection period during which a value for a parameter exceeded the predetermined threshold value for that parameter.

Analysis Section of a Health Report

The Analysis section of a health report includes a narrative that describes the results of the analysis that was performed on the collected data. The description that you see depends on whether parameter values exceeded predetermined thresholds and which parameters indicated problems.

Report results fall into three possible categories:

- No parameters are flagged to indicate problems
- No Average reading is flagged, but one or more % Over Critical Value readings are flagged
- One or more Average readings are flagged

If all Average numbers and % Over Critical Value numbers are fine (not flagged), the following narrative is displayed:

The SpectroSERVER appears to be running healthy and should be capable of handling approximately (100 - % Capacity) % more load; assuming that the type of new devices being modeled remains relatively the same, and no additional workloads are introduced (for example: high trap rates or additional Watches).

If all Average numbers are fine, but % Over Critical Value number is flagged, the following narrative is displayed:

On average, the SpectroSERVER is running within an acceptable resource utilization range, however, as indicated by a high “% over threshold” value, there are excessive periods of time where one (or more) of the system resources are overutilized. This could be an indication that the resource could be close to a premature bottleneck. Based on the calculated values from this data collection period, it appears that the following problems might exist.

A list of problems that can create the conditions that were flagged then follows.

Other narratives are also displayed for various threshold conditions. In some cases, you are advised to contact CA Support.

Tune OneClick to Improve Performance

Once you have determined the reasons for degraded DX NetOps Spectrum performance, tune the OneClick to improve performance by taking the following steps:

- Modifying the polling interval and poll-to-log ratio of essential device models and application models. Disabling polling of non-essential models. These changes reduce the network traffic and the resulting latency that affect performance.
  - **Polling interval**: The time interval in seconds at which the OneClick reads all device model attributes that are flagged as POLLED.
  - **Poll-to-log ratio**: The number of OneClick device polls that occur prior to logging the attributes that are flagged as LOGGED. The default value is 0 (logging is disabled).

  Polling and logging create the primary workload for OneClick. Changes to polling and logging can have a significant impact on performance. To see the best performance, poll and log only required data.
- Increasing the capacity of the system by increasing memory, CPU speed, or disk space.
- Reducing the number of traps that are mapped to DX NetOps Spectrum events.
- Reducing the amount of data that is requested by customized watches and displayed attributes. As a result, less data is requested from the OneClick and devices.
- Adjusting usage of features such as Live Pipes, Discovery, and automatic device configuration.
- Adjusting client interactions with OneClick. For example, reports that are generated using DX NetOps Spectrum Report Manager can exert a punctuated or prolonged performance burden on the server. The load depends on what is reported and how often the reports are run. Command Line Interface (CLI) scripts, manual discoveries, and other manually-initiated tasks can also affect OneClick performance.

Note: This chapter provides information about configuring polling for device and application models. For other suggested measures to improve OneClick performance, contact CA Support.

Polling Intervals to Manage Information

DX NetOps Spectrum polls devices to retrieve management information. You can change the polling interval for each device as a tuning measure. However, note the following guidelines:

- If you increase the time between polls, less bandwidth is required for management traffic. However, device status is updated less frequently.
- If you decrease the time between polls, device status is updated more frequently. However, more bandwidth is required for management traffic.

Default Polling Intervals

By default, DX NetOps Spectrum polls some devices every 60 seconds, polls other devices every 300 seconds, and does not log statistics (the poll to log ratio is set to 0). Often, the polling of this frequency is unnecessary and slows performance by creating network traffic and resulting latency.

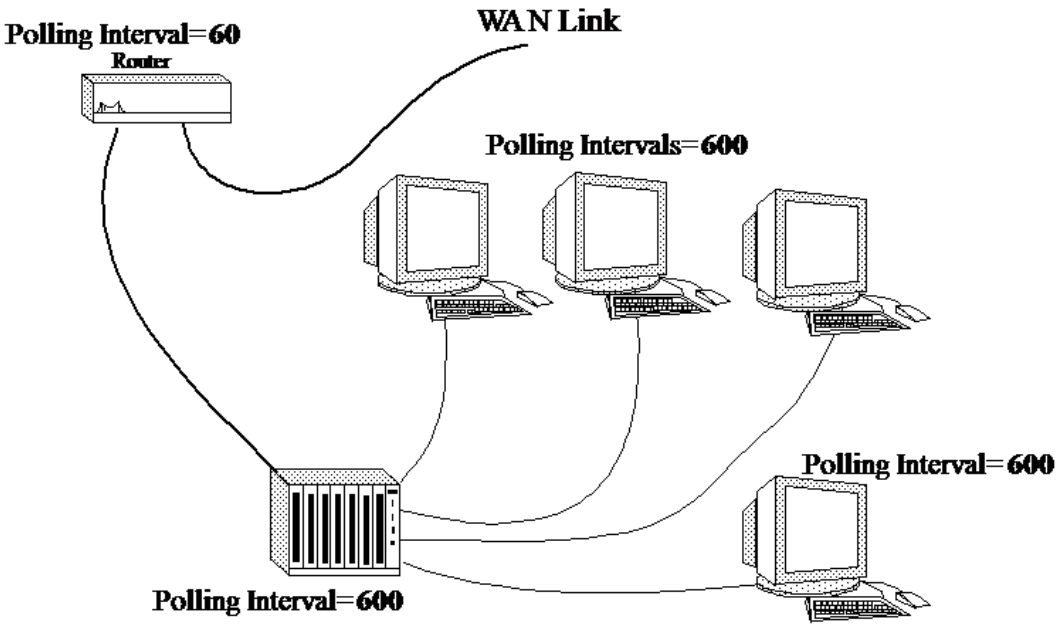
A good rule of thumb is to poll and log critical background devices every 60 seconds and to poll other, less critical network devices every 180 to 300 seconds. Often, polling and logging for end nodes, such as workstations, can be disabled to reduce network traffic and the SpectroSERVER workload.

NOTE
By default, the Poll_Log_Ratio attribute on device models is set to 0, which effectively disables native DX NetOps Spectrum logging. To log device, attribute, and port statistics, we recommend using SSLogger instead of the native method, which writes the information to the Archive Manager database. SSLogger is a DX NetOps Spectrum command-line application that logs statistics directly to ASCII files. This type of logging reduces the load on the Archive Manager database and eliminates the need to export the data. SSLogger also provides increased control over the type and frequency of data that is logged.

NOTE
For more information about SSLogger, see SSLogger .

Staggering Polling Intervals to Reduce SpectroSERVER Workload

You can set staggered polling intervals to reduce network management traffic, spread out the SpectroSERVER workload, and enhance fault management. An example is shown in the following illustration:



If all the devices in the example had a default polling interval of 60 seconds, they would all use SpectroSERVER resources every 60 seconds. SpectroSERVER resource utilization is reduced by setting the polling interval to 60 seconds for the router and to 600 seconds for all the other devices. However, the staggered polling does not impede management capabilities. If any fault occurred on the devices that are downstream from the router, polling would be interrupted, and an alarm would be generated.

Configuring Polling for Multiple Devices

To enhance SpectroSERVER performance, you can modify the polling interval, modify the poll-to-log ratio, or disable polling altogether for multiple devices using the Attribute Editor. The Attribute Editor is an advanced OneClick utility that allows you to change one or more attribute values for multiple models at once.

NOTE
You can also use the command-line interface to change the attribute values for multiple models at once. For more information, see Command Line Interface.

Set the Polling Interval and Poll-to-Log Ratio for Multiple Devices

Polling and logging create the primary workload for OneClick. Configure the polling interval and the poll-to-log ratio to improve SpectroSERVER performance.

The *polling interval* is the time interval in seconds at which the SpectroSERVER reads all device model attributes that are flagged as POLLED. The *poll to log ratio* is the number of SpectroSERVER device polls that occur prior to logging the attributes that are flagged as LOGGED in the database. The default value is 0 (logging is disabled).

NOTE
By default, the Poll_Log_Ratio attribute on device models is set to 0, which effectively disables native DX NetOps Spectrum logging. To log device, attribute, and port statistics, we recommend using SSLogger instead

of the native method, which writes the information to the Archive Manager database. SSLogger is a DX NetOps Spectrum command-line application that logs statistics directly to ASCII files. This type of logging reduces the load on the Archive Manager database and eliminates the need to export the data. SSLogger also provides increased control over the type and frequency of data that is logged.

NOTE
For more information about SSLogger, see SSLogger .

Follow these steps:

1. Search for the device models that you want to modify from the Locater tab in the OneClick Console.

   NOTE
   For more information, see Using OneClick. Results are displayed.
2. Select the models to modify, right-click, and select Utilities, Attribute Editor.
   The Attribute Editor opens.
3. Use the Attribute Editor to modify the following attributes:
   - Poll Interval
   - Poll to Log Ratio

   NOTE
   You can find these attributes under SNMP Communication in the Attributes tree. For more information, see Modeling and Managing Your IT Infrastructure *section*.

Disable Polling for Multiple Devices

You can disable polling for some device models. For example, you want to disable polling for endpoints, such as workstations, to avoid using bandwidth for network polling traffic. Some administrators do not model endpoints at all because of the alarms that can occur each time that the endpoints are powered down.

Follow these steps:

1. Search for the device models that you want to modify from the Locater tab.

   NOTE
   For information about searching using the Locater tab, see Using OneClick . Search results are displayed in the Results tab.
2. Select the models to modify, right-click, and select Utilities, Attribute Editor.
   The Attribute Editor opens.
3. Use the Attribute Editor to set the PollingStatus attribute to no (for false) to disable polling.
   Manually add the attribute to the User Defined folder in the tree.

   NOTE
   For more information, see Modeling and Managing Your IT Infrastructure *section*.

Configuring Polling for Multiple Applications

Application model types have different default polling intervals. The default polling interval for some application models is set to zero. To set the polling interval for these application models, use the Attribute Editor. You can quickly retrieve the application models by performing a search of All Application Models on the Locater tab.

NOTE
In general, we recommend setting the polling interval of application models to 60 seconds.

Polling Interval for a Single Device

The polling interval is the time interval in seconds at which the SpectroSERVER reads all of the attributes of the device model that are flagged as POLLED. You can set polling intervals for individual devices.

Follow these steps:

1. Select the device in the OneClick Console.
   The Component Detail panel displays the information for the selected model in the Information tab.
2. Expand the DX NetOps Spectrum Modeling Information subview.
3. Click set in the Poll Interval (sec) field, type the desired polling interval, and press Enter.
   The polling interval is set for this device.

Disabling Polling for a Single Device

You can disable polling for a single device.

Follow these steps:

1. Select the device in the Explorer tab or the Topology tab in the OneClick Console.
   The Component Details panel displays the information for the selected model in the Information tab.
2. Expand the DX NetOps Spectrum Modeling Information subview.
3. Click set in the Polling field, and select Off.
   Polling is disabled for this device.

Add SpectroServers

If you have not achieved desired performance levels after tuning your existing SpectroSERVER, size the network to determine the appropriate number of SpectroSERVERs to add.

NOTE
For more information, see Distributed SpectroSERVER Administrator .

Sizing the Network

The DX NetOps Spectrum sizing tool determines the number of SpectroSERVERs that are required to efficiently manage your network. The CA Support can run the sizing tool at your request.

Before your network can be sized, you must run a utility called PMCount and must provide the resulting data about your DX NetOps Spectrum environment. The PMCount utility finds the number of pollable models in a database, polling intervals, poll-to-log ratios, number of ports, and more. The sizer uses this raw data to estimate the following:

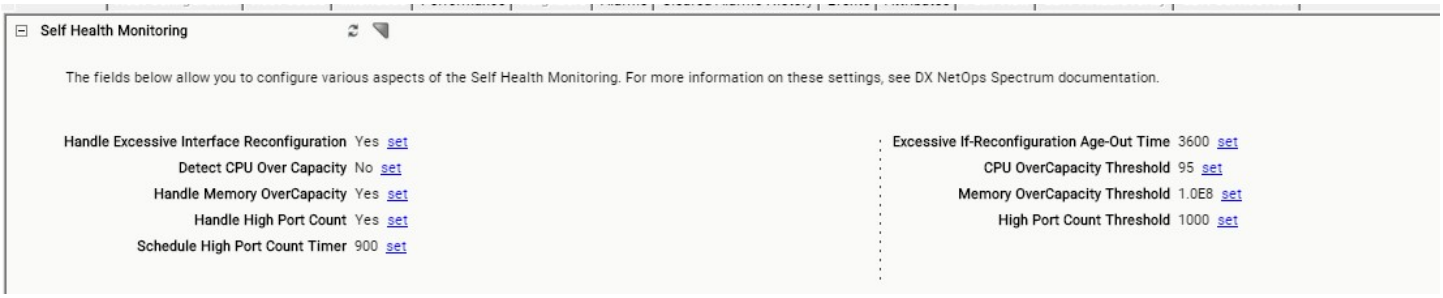
- The additional amount of network management traffic that DX NetOps Spectrum generates
- The number and configuration of additional SpectroSERVERs that are required to efficiently manage the number of models in your environment

Both the PMCount results and the sizing results can also identify places where polling and logging can be further reduced or disabled, thereby improving performance.

CA Support can help you size your deployment. You can also contact CA Support for detailed information about accessing and running PMCount.

Self-Health Monitoring

From 10.3.2, users can benefit from the newly introduced 'Self-Health Monitoring' or the 'Self-Healing' feature that monitors the health of SpectroSERVER as shown in the screenshot below. This feature ensures SpectroSERVER runs smoothly in adverse conditions such as faulty devices, interface issues, CPU, and memory hassles.



The following are the supported capabilities:

1. **Detect CPU's over capacity** - SpectroSERVER (SS) creates perf dmps for continuous high CPU utilization, for a designated time period. The threshold values are configurable. The dmp files are helpful in case the SS runs into performance issues.
2. **Detect memory over capacity** - Kernel crashes SS if the system runs out of available or swap memory. This issue is now addressed and the SS shuts down gracefully, in case the available and swap memory is less than 100 Meg (configurable).
3. **Excessive interface re-configurations handling** - Previously SS raised ‘Excessive Interface Reconfigure' alarm on a device if six interface re-configurations occur within 31 minutes. These frequent interface re-configurations adversely affect the SpectroSERVER's performance. To address this issue, from 10.3.2 onwards, if this option is enabled, SS disables auto-reconfiguration on such devices and waits for an age-out time (configurable to 1 hour by default). After the age-out time expires, SS enables the reconfiguration.
4. **High port count handling** - If this option is enabled, SS restricts the continuous re-configurations on high port count devices by scheduling it for every 15 minutes (configurable). By default, devices with port count of more than 1000 are identified as high port devices and this value is configurable.

   NOTE

   It is recommended that you enable all the capabilities/options that are mentioned above on all the SpectroSERVERS.

   Note that from 10.4.2 onward this functionality is enabled by default. In previous releases, these options were not enabled by default.

Spectrum Performance View (Beta)

Spectrum Performance View provides a real-time health dashboard for the distributed DX NetOps Spectrum environment. The administrator can run this tool to get the performance trends, it is disabled by default. SpectroSERVER tracks the performance statistics like operating system metrics (like CPU, memory, threads, and so on) and other internal metrics like total models, traps, events, alarms, and so on. These metrics are retrieved every minute and stored as an event on the SpectroSERVER Performance model.

From 10.4.2, DX NetOps Spectrum installer configures the Influx server and user **spectrum** is automatically created. The lifecycle of the Influx process is managed by the processd. A new page named **InsideView Configuration(beta)** is added under the OneClick Administration page to configure InsideView, which has options to save, start, and stop InsideView. Also, see Spectrum Performance view improvements after DX NetOps Spectrum upgrade from 10.4.1 to 10.4.2 release.

These metrics are useful in capacity planning and troubleshooting performance problems like:

- Spectrum sending too much traffic
- Frequent disconnection of OneClick Server
- SpectroSERVER hang
- Poll/notifications latencies
- Frequent crashes / getting Out of Memory

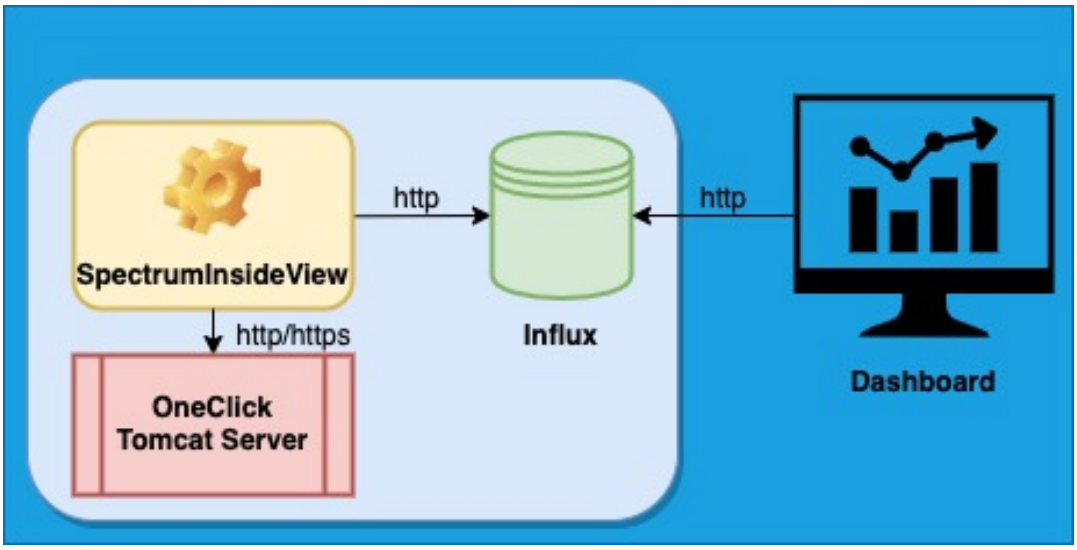
Before the current release, PerfCollector was used for the purpose. PerfCollector is a troubleshooting tool that pulls the performance events from every landscape and creates prf files for performance analysis. It lacks in providing real-time UI to observe the trend.

The Health dashboard provides a real-time DX NetOps Spectrum performance across DSS. You can review the dashboard at regular intervals and take preventive action if you find any dissimilarity in the trend.

DX NetOps Spectrum pulls the performance events from every landscape and stores the data into the Influx database. It uses the REST Interface to pull the events from DDMDB.

Understanding the Spectrum Performance View

The following illustration shows the architecture of the Spectrum Performance View:



The Spectrum Performance View has the following two dashboards:

- **SpectrumInsideView:**
  Application to collect SpectroSERVER performance events and store in influx.
- **Dashboard:**
  - **OneClick Server -DSS View** – High-level summary of all SpectroSERVER metrics attached to OneClick.
  - **Spectrum Health View** – Details of SpectroSERVER metrics.

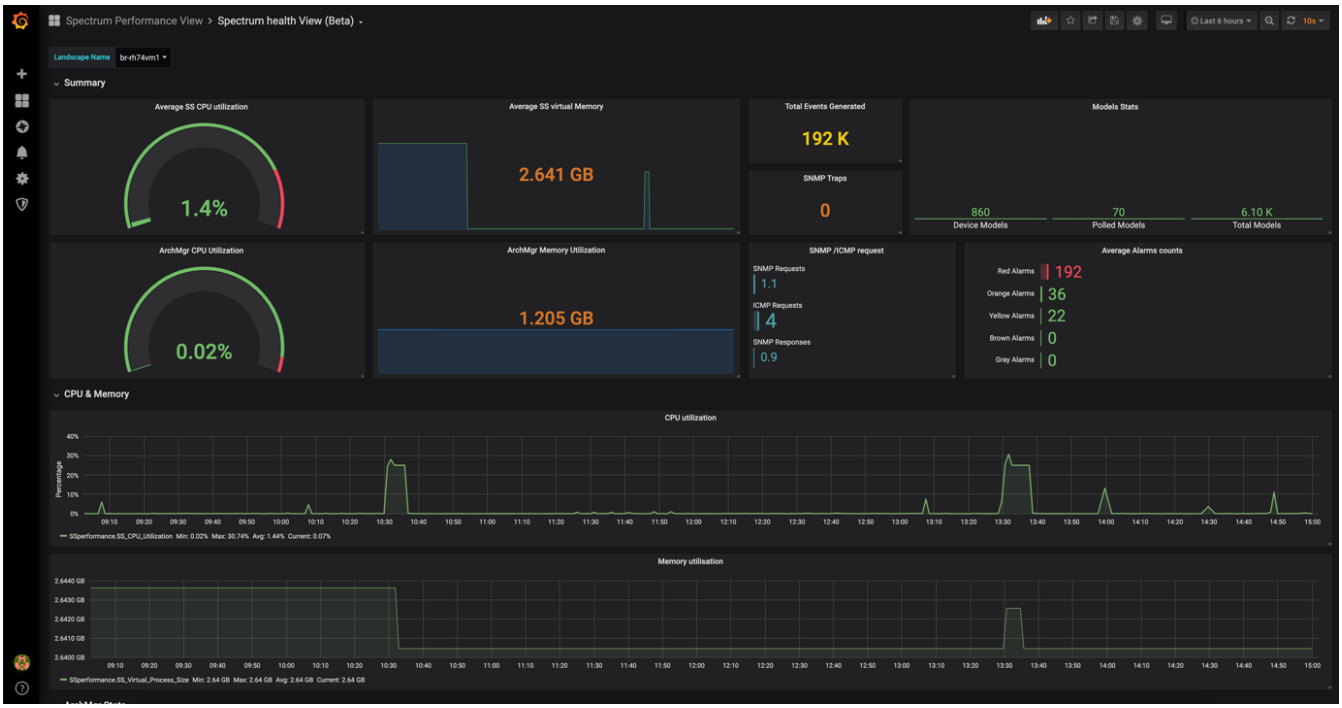
OneClick Server - DSS View

The OneClick Server DSS View (Beta) provides the high-level summary of all SpectroSERVER associated with the OneClick server. The view also provides the following information:

- SpectroSERVER, Archive manager CPU and memory utilization.
- The total number of devices discovered in each SpectroSERVER.

Spectrum Health View

The Spectrum Health View provides a detailed view of each metric of SpectroSERVER like SpectroSERVER CPU, SpectroSERVER Memory, Archive Manager CPU, Archive Manager Memory, SNMP traps and discovered devices, and so on.



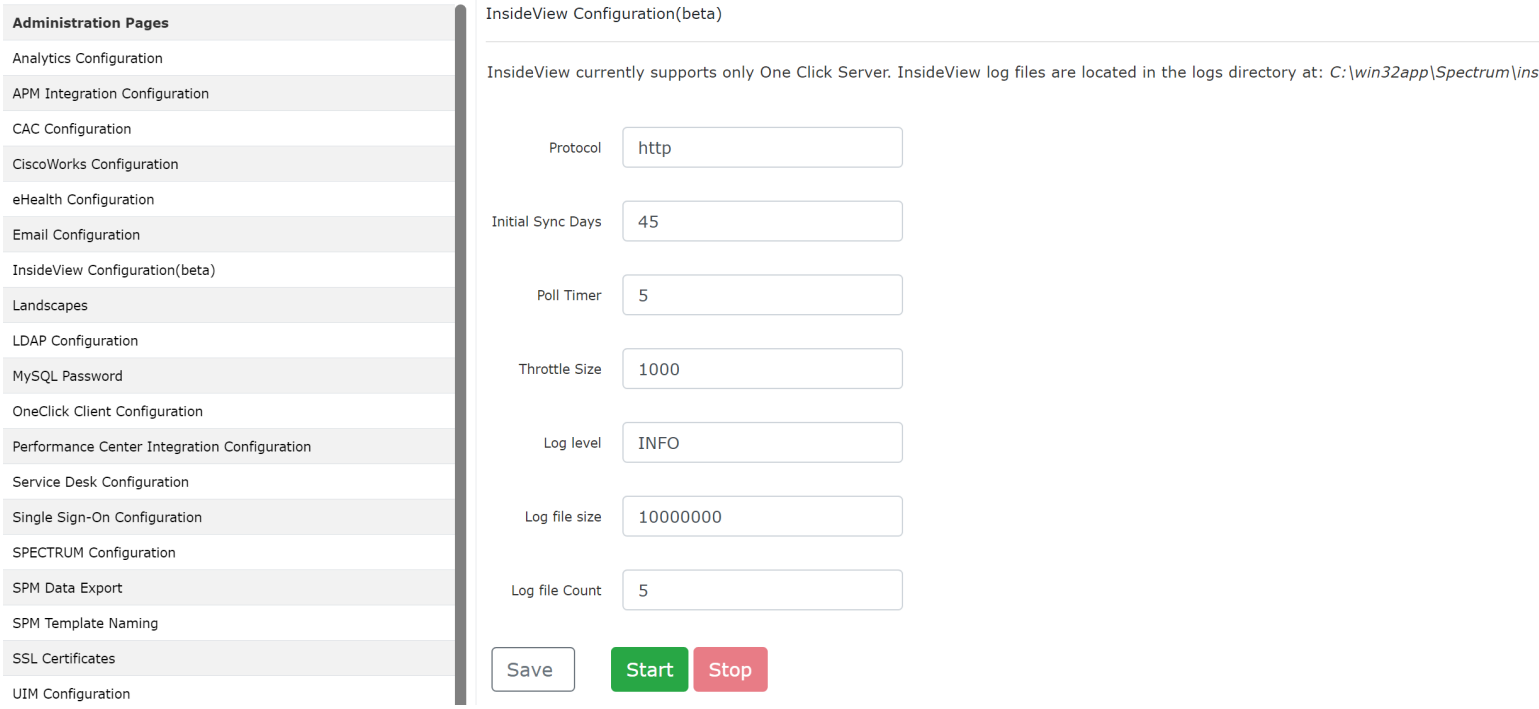
Set Up Spectrum Performance View

This section describes the steps to set up the application.

1. Configure the Spectrum InsideView
2. Install and Import Dashboard in Grafana
3. Customize Influx Database

Configure the SpectrumInsideView

See **InsideView Configuration(beta)** section in the OneClick Administration page to configure SpectrumInsideView,



The **Save** option saves the information but does not start or stop the application. Click the **Start** or **Stop** buttons to start or stop the SpectrumInsideView.

The configuration file for SpectroSERVER Health View, `SpectrumInsideView.conf` is located in the `$SPECROOT\insideView` directory. The `SpectrumInsideView.conf` file contains OneClick Server name, HTTP Port, mode, and initial sync and increment sync duration. The file also contains the Influx DB details like database name, database credentials, HTTP port, and mode.

> **NOTE**
> HTTPS is currently not supported.

OneClickNodes

- **Protocol**
  Specifies the appropriate protocol.
  Default: HTTP
- **Initial Sync Days**
  Specifies the interval at which the OneClick must sync with Spectruminsideview. The minimum value must be one minute.
- **Poll Timer**
  Specifies the interval between two polls.
  Optimal Interval to avoid Overload: Five minutes
- **Throttle Size**
  Specifies the throttleSize
- **LogLevel: INFO**
  Specifies the DEBUG, INFO, ERROR, WARN.
  Default: ERROR
- **Log File Size**

Specifies the size of the log file in bytes.

- **LogFileCount: 5**
  Specifies the maximum number of log back up files.

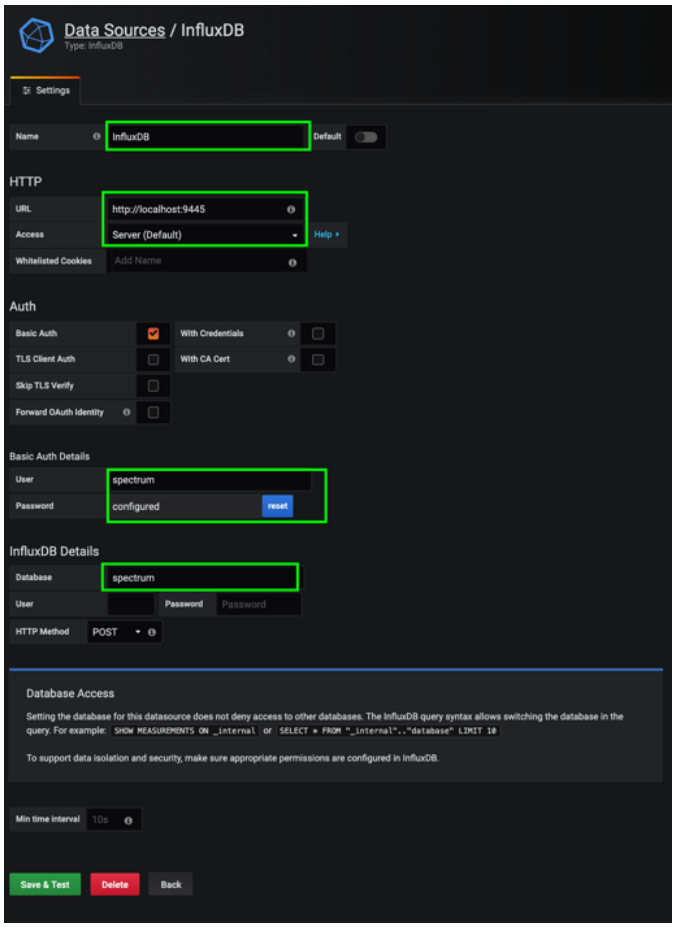
Install and Import Dashboard in Grafana

This section describes the procedure to install and import the SpectroSERVER health view dashboard in Grafana.

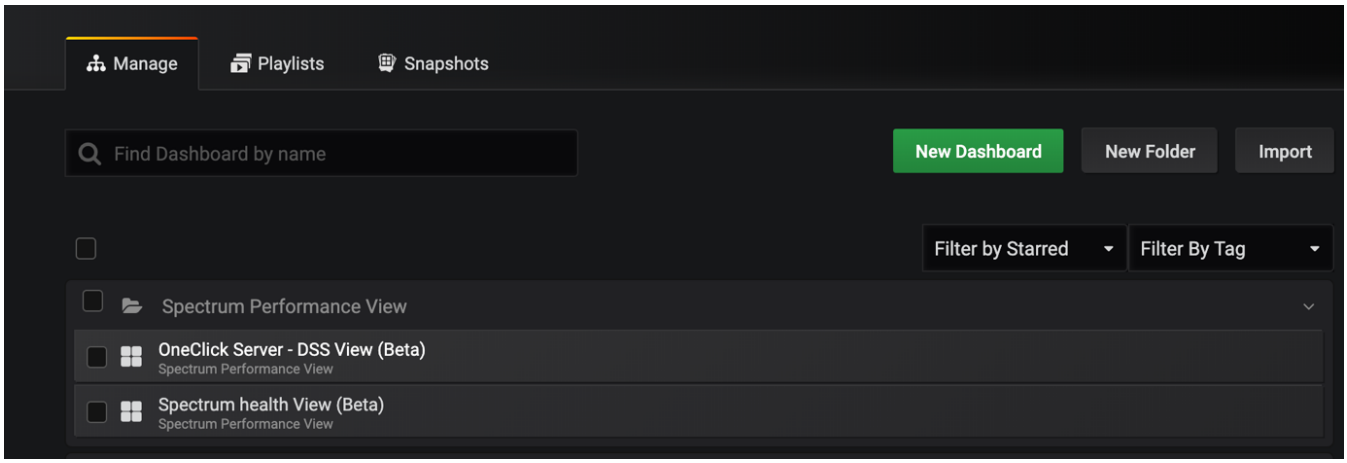
> **NOTE**
> If Grafana is not available in your pod/machine, install it.

Follow these steps:

1. Log in to the Grafana portal.
   Default Credentials: admin/admin
2. Add influx DB data source name as "InfluxDB" (URI : http://<oneClickServername>:9445, username: spectrum / password).



3. Create a dashboard folder with the name **Spectrum Performance View.**
4. Import the Spectrum_health_View.json and OneClick_Server_View.json files from `$SPECROOT/insideView/dashboard` into Grafana and place under the **Spectrum Performance View** directory.



Customize Influx Database

The Influx database is automatically created by the DX NetOps Spectrum installer. The influx database is installed in the `$SPECROOT\influx` folder. It runs on a default port 9445 and uses HTTP communication.

> **NOTE**
> You can change the port by editing the **influxdb_spectrum.conf** file under the HTTP section, port number.

Spectrum Performance View Improvements After DX NetOps Spectrum Upgrade from 10.4.1 to 10.4.2

After you upgrade DX NetOps Spectrum from 10.4.1 to 10.4.2, note the following improvements:

- processd starts the Influx process automatically.
- Database user, password along with database, and data present in 10.4.1 are retained post-upgrade.
- The `SpectrumInsideView.conf` file updated in 10.4.1 is retained. You can update the same file using the **InsideView Configuration(beta)** option on the OneClick Administration page.
- You can **start** or **stop** SpectrumInsideView from **InsideView Configuration(beta)** on the OneClick Administration page.

Troubleshoot the Spectrum Performance View

This section describes the commonly identified issues and workaround to resolve the issue.

- If any error occurs while executing SpectrumInsideView, restart the SpectrumInsideView by changing the sync files to 0, then SpectrumInsideView starts from initial sync. This does not lead to data duplication in influx.
- Ensure that OneClick Rest call throttle size is between 1000 and 2000, to avoid overload on OneClick Server.
- The optimal value for the PollTimerMins parameter is five minutes, a lower value may overload the OneClick server with RestAPI.

Spectrum TrapInsight Dashboard View

From 10.4.2, DX NetOps Spectrum TrapInsight provides a real-time trap trend analysis dashboard for the distributed DX NetOps Spectrum environment. The administrator can run this tool to get the trap analysis trend; it is disabled by default. This feature is part of SDC with TrapX installation. When a new trap is received, TrapX forwards the trap to Logstash. Logstash processes and sends the trap to the Influx database using the logstash-influx-output plugin. DX NetOps Spectrum installer configures the TrapInsight server.

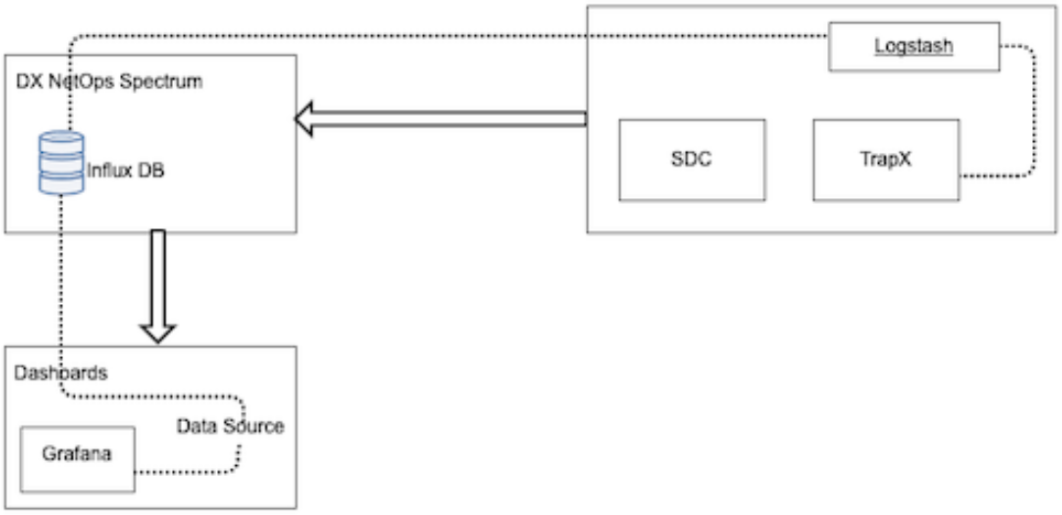
A user named **spectrum** and the TrapInsight database is automatically created. The Influx database is installed only OneClick server in the `$SPECROOT\influx` folder. It runs on a default port 9445 and uses HTTP communication.

You can configure multiple SDCs with Logstash to send the trap information to the same Influx database.

Implementing TrapInsight sends the trap information to the configured Influx database, allowing you to visualize the data and create the dashboards to get the trap trend analysis. You can review the dashboard at regular intervals and take preventive action, if you find any dissimilarity in the trend.

Spectrum TrapInsight Architecture

The following diagram shows the Spectrum TrapInsight architecture:



Deploy Logstash

To deploy Logstash, enable the Trapx option when you install SDC. Once the installation is done, the Logmonitor folder is created as part of the installation.

NOTE
Ensure that the Influx database server is running before executing the setup_logstash script. Login to OneClic server and run the following command from Influx directory to check if Influx is running:

```
influx -port 9445 -username spectrum -password spectrum
```

NOTE
In the Windows environment:

- The bash file is located at `SDC_Installation_Directory\SDMConnector\NT-TOOLS\SDCRE` directory.
- Use the SDC shipped Cygwin for executing the setup_logstash script to avoid the failure.
- A new command prompt is opened while executing the script, do not close the window. You must stop the Logstash process using `CTRL+C` from this window.

Follow these steps:

1. Open a bash shell and execute the following script from the `SDC_Installation_Directory/bin` folder:

```
./setup_logstash <Influx DB IP Address> <SDC IP Address>
```

NOTE
For example, `/c/Program Files/CA/SDMConnector/bin > ./setup_logstash.sh 10.175.90.201 10.175.90.200`

Once the script execution is done, Logstash logs (`logstash-plain.log`) can be found at `\Logmonitor \logstash\Logs\` directory. The SDC service is restarted as part of this execution. The script updates the `TrapX.config` file to forward the traps to Logstash.

NOTE
Port 9162 is used for Logstash. If the port is used by another process, change it.

2. (Optional) Change the Logstash port in the `logmon.conf` file in the `SDC_Installation_directory/bin` folder if the default port is used by another process.

Enable SSL for TrapInsight

When the Influx database is running with SSL enabled, you must enable SSL for TrapInsight.

Follow these steps:

1. Open the `logmon.conf` from `SDC_Installation_directory/bin` folder.
2. Set the value of the parameter `ssl` to `true`
   By default, SSL is set to false.

TrapInsight Dictionary

Trap Insight Dictionary is used to map the trap OID to the corresponding name. The dictionary is created using the Mibtools database and shipped with Logstash deployment.

Update the MIB Name Mapping

After you import any new MIB, to view the newly added trap OID to its corresponding name in the TrapInsight dashboard update the TrapNameList.yaml file.

Follow these steps:

1. In SpectroSERVER, run the `fetch_trap_info.sh` command from `$SPECROOT/mysql/bin` folder.
   The script generates the `TrapNameList.yaml` file at `MYSQL_DIR/data/mibtools` directory.
2. Copy the `TrapNameList.yaml` file to `\Logmonitor\logstash\bin` directory in the SDC Trapx machine to update the new mib name mapping.

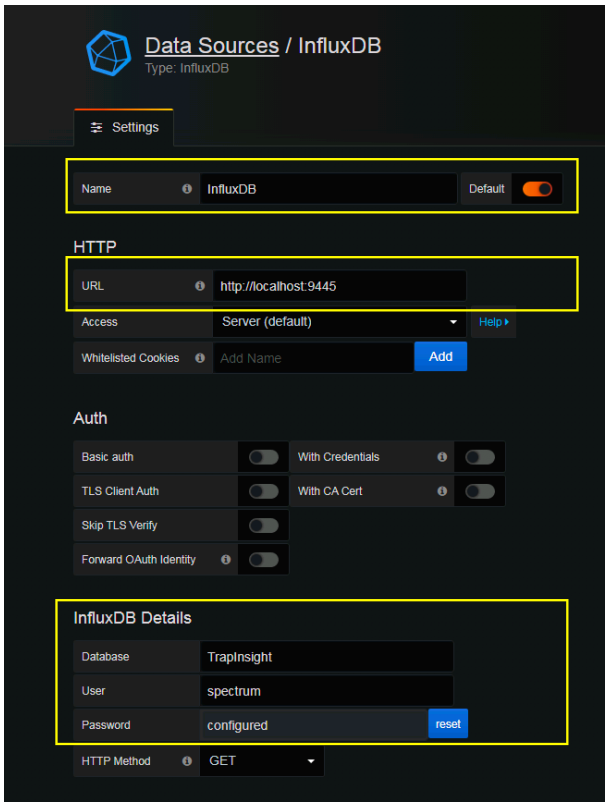
Install and Import Dashboard in Grafana

This section describes the procedure to install and import the Influx dashboard into Grafana.

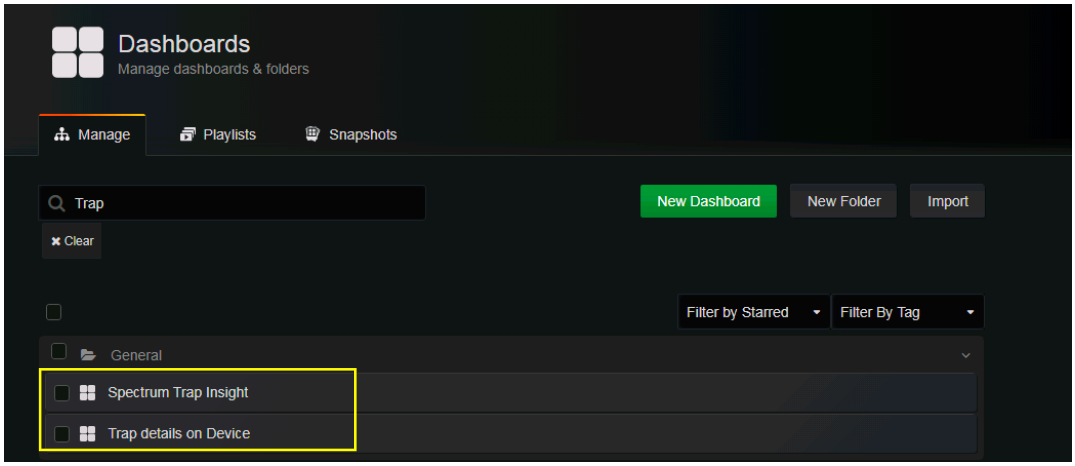
NOTE
If Grafana is not available in your pod/machine, install it.

Follow these steps:

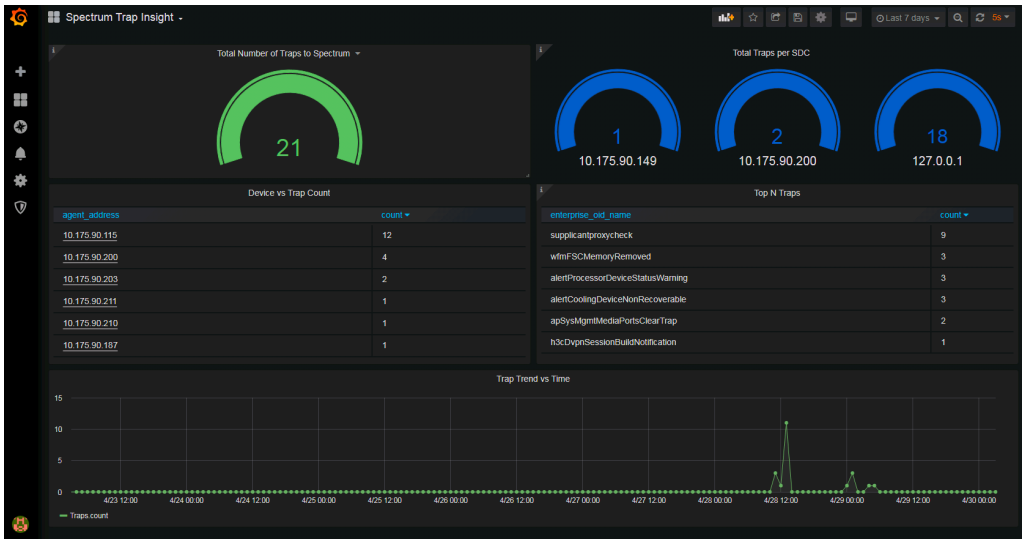
1. Log in to the Grafana portal.
   Default Credentials: admin/admin
2. Add the Influx database data source name as "InfluxDB" (URI : http://<oneClickServername>:9445, username: spectrum /password).



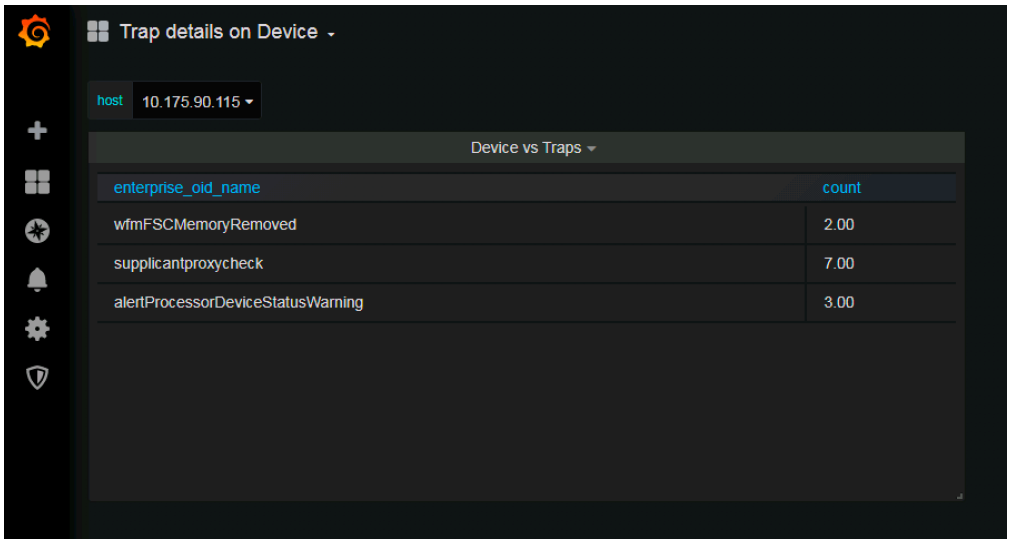
3. Create a dashboard folder with the name Spectrum Trap Insight.
4. Import the following files from the `$SPECROOT/insideView/dashboard` directory into Grafana and place under the Spectrum Trap Insight directory.
   - Spectrum_Trap_Insight.json
   - Trap_details_on_Device.json
   - Spectrum_TrapInsight_Ledger_View.json



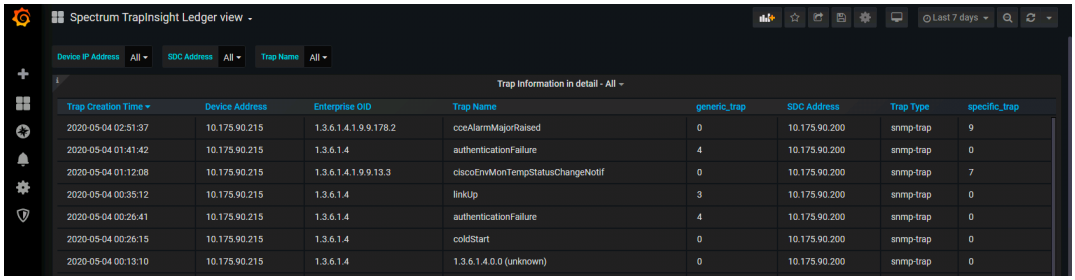
5. You can open the individual dashboard to see the following views:
   Spectrum Trap Insight



Trap Insight Details on Device



Spectrum TrapInsight Ledger View



Stop the Logstash

You must stop the Logstash before the SDC upgrade or changing the Influx server. Stopping Logstash stops sending the trap information to Influx.

Follow these steps:

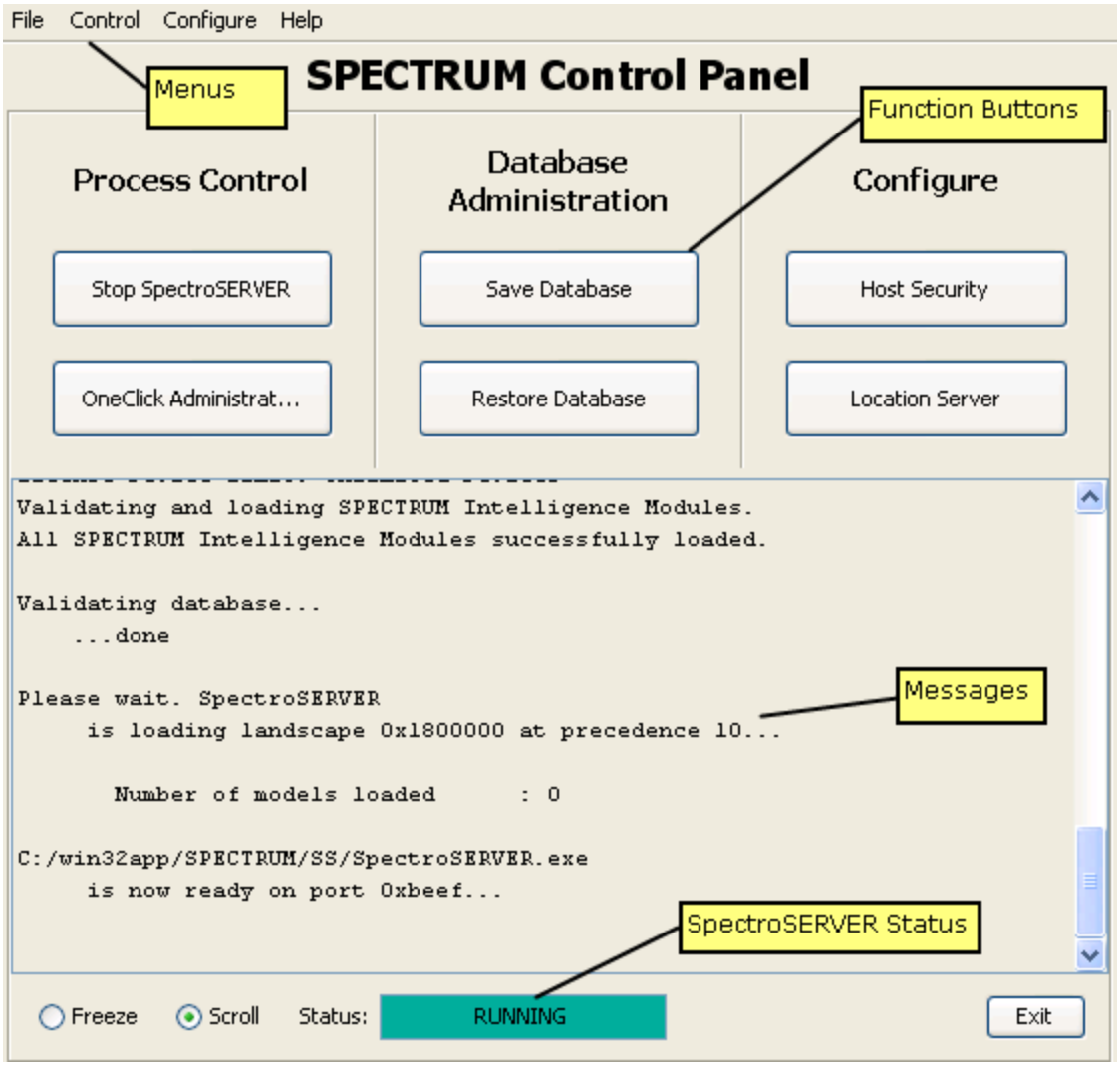
- **On Linux:** Run the `stoplogstash.sh` script in `SDC_Installation_directory/bin` folder.
- **On Windows:** Click on the window opened by Logstash, type CTRL+C, and press Enter.

OneClick Administration

This section explains how administrators can configure the important components like the Spectrum Control panel, SpectroSERVER, OneClick, and other important functionalities of DX NetOps Spectrum.

DX NetOps Spectrum Control Panel Overview

The Spectrum Control Panel lets you configure resources, start and stop the SpectroSERVER, perform database administration, and maintain the DX NetOps Spectrum installation. It provides menus, function buttons, messages, and a status bar.



Access the DX NetOps Spectrum Control Panel

To access the Spectrum Control Panel on the SpectroSERVER host, you can take the following steps:

- **Windows:** Click Start, Programs, CA, Spectrum Control Panel.
- **Linux:** Run the SCP command located in *$SPECROOT*/bin/

You must be logged in as the install user or as someone in the install user's group to launch applications using the Spectrum Control Panel.

File Menu

The File menu provides the following functionality:

- **Select Host Machine**
  Lets you select the SpectroSERVER that you want to manage.
- **Save Database**
  Lets you create an online backup or perform a complete save of the SpectroSERVER database.
  - **Online Backup**
    If the SpectroSERVER is running, selecting Save Database initiates a DX NetOps Spectrum online backup.
  - **Save**
    If the SpectroSERVER is not running, selecting Save Database initiates a complete save of the SpectroSERVER database using SSdbsave (with the -c and -m switches).

    NOTE
    For more information, see the Database Management section.
- **Restore Database**
  Lets you load a previously saved database.
- **Initialize to Legacy Database**
  Initializes your database to the state that existed following your last installation. All models that are specific to your network are removed. The remaining database structure consists of the modeling catalog and a few internal models.

  WARNING
  Do not use this feature without first making a backup copy of your database.

Control Menu

The Control menu provides the following functionality:

- **Start/Stop SpectroSERVER**
  Controls the operation of the SpectroSERVER.
- **Auto Start/Stop Archive Manager**
  Lets you configure the Archive Manager to start or stop with the SpectroSERVER on the workstation you are managing.
- **Start/Stop Archive Manager**
  Starts or stops the Archive Manager.

  NOTE
  The Archive Manager control buttons will be disabled when any other running process has locked the database.

  WARNING
  Do not attempt to change the state of the Archive Manager when you are running an online backup.
- **OneClick Administration**
  Prompts for the host and port of the OneClick web server and opens a browser to the OneClick Administration Pages.

  NOTE
  To determine the browser location on Windows, DX NetOps Spectrum opens the default web browser for the current user. To determine the browser location on UNIX and Linux, DX NetOps Spectrum uses

the PATH variable to first locate Firefox, Mozilla, and then Netscape. To specify a browser, set the SPECTRUM_BROWSER environment variable. For more information, see the Set SPECTRUM_BROWSER Variable section.

- **SpectroSERVER Performance**
  Opens the Performance View application. For more information about Performance View, see the SpectroSERVER Performance Administration section.
- **Users**
  Lets you view user details, set a new password for an existing user, and create an administrative super user account.

Configure Menu

From the Configure menu you can open the Model Type Editor, and you can open dialogs that let you configure the SpectroSERVER, the location server, and host security.

SpectroSERVER Configuration Dialog

The SpectroSERVER Configuration dialog lets you control certain aspects of the SpectroSERVER configuration. When you make changes in this dialog, you are editing the .vnmrc resource file, which controls SpectroSERVER operation and performance.

NOTE
See the Distributed SpectroSERVER Administration section for more information about the .vnmrc file.

- Communications
  - **Communications Port**
    Specifies a TCP port number indicating the port through which the client's user interface communicates with the SpectroSERVER. This parameter can be any valid, unreserved TCP port greater than the port number assigned to the IPPoPORT_USERRESERVED and less than 64,000.
    Default: 0xBEEF
  - **SNMP Comm. Port**
    Specifies a value that can be used to select a port from which SNMP requests can be sent via the SpectroSERVER. It can be set to any unsigned 16-bit integer in the range 0x400 (1,024) to 0xFFFF (65,535). Some implementations of SNMP agents treat the port as a signed number. In these cases, this resource must be set to a value between 0x400 (1,024) to 0x7FFF (32,768).
- **File Paths**
  The SpectroSERVER Configuration dialog provides access to the file paths that are defined within the .vnmrc file.
  - **VNM File Path**
    Specifies the root subdirectory which contains SpectroSERVER external files such as specific device alert mapping.
- **Performance Tuning**
  SpectroSERVER is a multi-threaded process. During normal operation, each subsystem allocates numerous work threads. Each thread consumes memory and computing capacity. As a result, they can affect performance. Max Number of Poll Threads and Work Thread Age are two of the parameters that control the allocation of work threads.

  NOTE
  To better understand the interaction between resources and the parameters that control work threads, see Deployment Capacity and Optimization Best Practices.
- **Event Log**
  Under normal conditions, events are recorded in the DX NetOps Spectrum Distributed Data Manager (DDM) database. However, if communication between the SpectroSERVER and the Archive Manager is lost, event information is stored temporarily in the SpectroSERVER database until communication is re-established.

The growth of this temporary event data in the SpectroSERVER database is regulated by entries in the SpectroSERVER .vnmrc resource file. Use the Event Log fields to edit these settings.

- **Max Event Recs to Save**
  Maximum number of records that can be stored in the database.
  Default: 20,000
- **Event Record Increment**
  Specifies the number of records to be deleted from the database when the number of records exceeds the Max Event Recs to Save value.
  Default: 100
  Note: If you remove the event_record_increment entry from the .vnmrc file, the default is 250 records.

- **Statistics Log**
  Under normal conditions, statistics are recorded in the DX NetOps Spectrum Distributed Data Manager (DDM) database. If communication between the SpectroSERVER and the Archive Manager is lost, however, statistics information is stored temporarily in the SpectroSERVER database until communication is re-established.
  The growth of this temporary statistics data in the SpectroSERVER database is controlled by entries in the SpectroSERVER .vnmrc resource file. Use the Statistics Log fields to edit these settings.
  - **Max Statistics Recs to Save**
    Specifies the maximum number of records that can be stored in the database.
    Default: 5,000
  - **Statistics Record Increment**
    Specifies the number of records to be deleted from the Statistics Log database when the number of records exceeds the Max Statistics Recs to Save value.
    Default: 500

Location Server Configuration Dialog

The DX NetOps Spectrum *Location Server* is used to locate other DX NetOps Spectrum services on the network. The Location Server Configuration dialog lets you define the location server characteristics and your client applications.

> **NOTE**
> For more information about location servers, see the Distributed SpectroSERVER Administration section.

The Location Server Settings section of the Location Server Configuration dialog contains the following settings:

- **Main LS Host**
  Specifies the Main Location Server (MLS) hostname. This host workstation determines which connection services are available on the network. Other Location Servers connect to the MLS to determine the location and availability of services.
- **Main LS Port**
  Specifies the Main Location Server port address.
  Default: 0xdaff
- **Backup Main LS Host**
  Specifies the backup MLS name. If the MLS is not available when another host attempts to connect to it, the host is redirected to the backup MLS.
  When a DX NetOps Spectrum system starts up (or the Process Daemon, processd, is stopped and started), the location server on that system attempts to connect to the MLS to download “map” information. Map information is a listing of each DX NetOps Spectrum service that is available and the location of each server. If the MLS is down at that time, the map information is not available to the DX NetOps Spectrum system. Therefore, clients cannot connect to any DX NetOps Spectrum service.
  The role of the backup Main Location Server is to provide redundancy for the MLS in this scenario. If a backup MLS has been configured, the Location Server attempts to contact it after contact to the MLS has failed. Clients can then access DX NetOps Spectrum services even though the MLS is down.
  Use highly available systems for both the backup MLS system and the Main Location Server.

Each system that is pointing to the same MLS should also point to the same backup main location server.

- **Backup Main LS Port**
  Specifies the backup MLS port address.
- **Max Connections**
  Specifies the maximum number of port connections that can be made to this location server.
  Default: 750

The Client Applications section of the Location Server Configuration dialog contains the following settings:

- **Hostname**
  Specifies the client application main location server hostname. It preserves landscape map integrity for different environments.
- **Port**
  Specifies the client application Main Location Server port.

Host Security Dialog

The Host Security dialog lets you enter a list of servers and users allowed to connect to the host. You can also do this by editing the .hostrc file in the DX NetOps Spectrum directory.

> **NOTE**
> For more information about the .hostrc file, see the Distributed SpectroSERVER Administration section.

Model Type Editor Option

The Model Type Editor option in the Configure menu starts the Model Type Editor application. The Model Type Editor lets you modify the SpectroSERVER modeling catalog and configure relations, object-classes, and their contents. This option is not available unless the SpectroSERVER is in an INACTIVE or STOPPED state.

> **NOTE**
> To learn more about the operation of the Model Type Editor, see the Model Type Editor section.

SpectroSERVER Status

The Status field in the Spectrum Control Panel indicates the status of the SpectroSERVER with text and color.

- **Starting: yellow**
  This field changes to Running (green) after the start-up period expires.
- **Stopping: yellow**
  This field changes to Inactive (blue) after the server has shut down.
- **Running: green**
  This field indicates a normal running state.
- **Terminated: red**
  This condition is abnormal and indicates an error.
- **Inactive: blue**
  This field indicates that server shutdown is complete.

Restore a Database

You can load a previously-saved database using either the File menu or by clicking **Restore Database**.

Follow these steps:

1. In the Spectrum Control Panel, click **Restore Database**.
   A dialog asks whether you want to initialize your database.

2. Click **No** to perform a models-only load.
   The Restore Database dialog opens.
3. Locate and select the appropriate previously-saved database backup.
4. Click **Open**.
5. Click **OK**.
   The database load begins. If SpectroSERVER is running, SpectroSERVER restarts after the database loads.

Initialize to Legacy Database

Initialize to Legacy Database initializes your SpectroSERVER database to its state following your last installation. All models specific to your network are removed. The remaining database structure consists of the modeling catalog and a few internal models.

> **WARNING**
> Do not use this feature without first backing up the database.

Follow these steps:

1. Select File, Initialize to Legacy Database.
   An information dialog displays a warning.
2. Click Yes to continue. Or click Cancel to retain your existing database.
   If the SpectroSERVER is running when you start to initialize your database, a second dialog indicates that the SpectroSERVER is shut down during this process.
3. Click Yes.
   The initialization starts.
4. Restart your OneClick web server and restart any open OneClick Consoles.

> **NOTE**
> This action refreshes the OneClick explorer hierarchy and topology view.The SpectroSERVER database is initialized.

Configure Host Security

The Host Security window lets you enter a list of servers and users allowed to connect to the host.

> **NOTE**
> You can also control access to the host by editing the .hostrc file in the DX NetOps Spectrum directory. For more information, see the Distributed SpectroSERVER Administration section.

- To add a server/user to the list, enter the name in the appropriate box and click Add.
- To delete an item from either the Server List or the User List, select the server or user and click Remove.
- To let all hosts and users have access to the host server, enter a plus sign in Server List Add box and click Add.
- To let only the server where you are logged in to connect to the server, enter a minus sign in the Server List Add box and click Add. The minus sign becomes the name of your computer when is is added to the Server list.

> **NOTE**
>
> To save the host security configuration, at least one entry in the Server List is required.

OneClick Web Server Administration

This section discusses tasks that OneClick administrators can perform to configure and optimize the OneClick web server. It also covers server-related and client-related configuration and maintenance issues.

Start and Stop the OneClick Web Server from the Command Line

You can start or stop the OneClick web server from a command prompt.

On Linux, log in as root. Use the following commands:

- To start the web server:
  ```
  <$SPECROOT>/tomcat/bin/startTomcat.sh
  ```
- To stop the web server:
  ```
  <$SPECROOT>/tomcat/bin/stopTomcat.sh
  ```
- To restart (stop, then start) the web server:
  ```
  <$SPECROOT>/tomcat/webapps/spectrum/restart.sh
  ```

To start or stop the OneClick web server on Windows, enter the following commands at a command prompt:

- To start the web server:
  ```
  C:\> net start spectrumtomcat
  ```
- To stop the web server:
  ```
  C:\> net stop spectrumtomcat
  ```

Start and Stop the OneClick Web Server from an Administration Page

Several of the OneClick Administration Pages include Restart OneClick Server buttons. You can easily restart the OneClick web server to apply a configuration change. These restart buttons use the ‘at’ utility to schedule a restart script to run. You can configure this utility for different platforms.

> **NOTE**
> Confirm that the ‘at’ command utility is installed and it has the necessary server permissions. If the ‘at’ command is not available or it fails to execute, restart or config the tomcat manually using the restart.sh or configtomcat.sh script.

If an error occurs while restarting the OneClick web server, you see an error message on the administration page. In this event, use the troubleshooting tips in one of the following sections to identify the problem. Or restart the web server from the command line.

Troubleshooting the at utility on Windows

By default, DX NetOps Spectrum users have permissions to execute ‘at’ on Windows. However, verify that the current user and the current user group have Read and Execute permissions on the C:\WINDOWS\system32 folder.

Check the status of the ‘at’ operation by typing ‘at’ in a command prompt shell to view the ‘at’ queue. The queue contains all jobs that are scheduled through ‘at’ that are still pending. If earlier attempts on these scheduled jobs have failed, the jobs also have an error status code.

Troubleshooting the at utility on Linux

Users who are listed in the following file are denied permission to use the ‘at' utility:

- (Linux) /etc/at.deny

Verify that the user currently running the OneClick process (the OneClick web server) is not listed in this file. Typically, this user is the DX NetOps Spectrum Installation Owner user. You can identify the user by entering the following command in a command shell:

```
ps -eaf | grep OneClick
```

If the ‘mail’ utility is set up for the operating system and the current OneClick user, ‘at’ automatically emails notifications about scheduled jobs and their output or error messages. Check these email notifications for pertinent information.

Start and Stop the OneClick Web Server from the Windows Control Panel

You can start and stop the OneClick web server from the Windows Control Panel.

Follow these steps:

1. Click Start, Control Panel.
   The Control Panel opens.
2. Double-click Administrative Tools.
   The Administrative Tools window opens.
3. Double-click Services.
   The Services window opens.
4. Select SpectrumTomcat from the services list and determine its status.
5. Do *one* of the following:
   – If the SpectrumTomcat service is running, click Stop to stop the web server.
   Or click Restart to stop and then start the web server.
   – If the SpectrumTomcat service is stopped, click Start to start the web server.

Configure the OneClick Server to Support Over 100 Users

To support a large number of users on a single OneClick server, increase the hard limit on the number of file descriptors. We recommend this step to ensure support for more than 100 OneClick Console users. The /etc/system file sets the limit of file descriptors.

Follow these steps:

1. Make a backup of your /etc/system file.
2. Add the following line to your /etc/system file:

```
set rlim_fd_max=4096
```

Launch OneClick Clients with Context

You can launch OneClick clients within the context of a topology or model. Pass contextual parameters and values with the URL that launches OneClick. You can launch the **OneClick WebApp** in-context to open the supported operations.

> **NOTE**
> To launch OneClick WebApp in-context, always launch a new WebApp instance.

The **thick client** URL can include parameters in the following format:

```
http://<hostname>/spectrum/oneclick.jnlp?<parameter>=<value>
```

The **WebApp** URL can include parameters in the following format:

```
http://<hostname>:<port>/spectrum/webapp?<parameter>=<value>
```

Possible parameters include the following:

topology Parameter

The value of the topology parameter can be a model handle or an IP address. For Global Collection, the topology parameter is Unique Key. Using this parameter in a URL launches an OneClick client or reuses an existing one, selects the Explorer tab if not already selected, expands the tree to show the model, selects the Topology tab if not already selected, and selects in the Topology panel the model specified by the topology parameter in the URL.

Examples:

```
http://<hostname>/spectrum/oneclick.jnlp?topology=0x3780003d
http://<hostname>/spectrum/oneclick.jnlp?topology=10.253.9.7
http://<hostname>:<port>/spectrum/webapp?topology=0x3780003d
http://<hostname>:<port>/spectrum/webapp?topology=10.253.9.7
```

For a Global Collection:

```
http://<hostname>/spectrum/oneclick.jnlp?topology=5416910d-01b7-1000-1f81-8a2a56270000
http://<hostname>:<port>/spectrum/webapp?topology=5416910d-01b7-1000-1f81-8a2a56270000
```

explorer Parameter

The value of the explorer parameter can be a model handle or an IP address. For Global Collection, the explorer parameter is Unique Key. Using this parameter in a URL launches a OneClick client or reuses an existing one, selects the Explorer tab if not already selected, and expands the tree to show the model. The currently selected tab in the Contents panel will reflect the new model.

Examples:

```
http://<hostname>/spectrum/oneclick.jnlp?explorer=0x3780003d
http://<hostname>/spectrum/oneclick.jnlp?explorer=10.253.9.7
http://<hostname>:<port>/spectrum/webapp?explorer=0x3780003d
http://<hostname>:<port>/spectrum/webapp?explorer=10.253.9.7
```

For a Global Collection:

```
http://<hostname>/spectrum/oneclick.jnlp?explorer=5416910d-01b7-1000-1f81-8a2a56270000
http://<hostname>:<port>/spectrum/webapp?explorer=5416910d-01b7-1000-1f81-8a2a56270000
```

alarm Parameter

The value of the alarm parameter can be either the integer alarm ID (to facilitate integration with legacy applications), the complete global alarm ID (in the form 3f983d3d-2045-1000-012b-000bdb5a1c31), or <*model handle*>@<*alarm ID*>. Using this parameter in a URL launches a OneClick client or reuses an existing one, selects the Explorer tab if not already selected, expands the tree to show the model, selects the Alarms tab if not already selected, and selects the alarm.

Examples:

```
http://<hostname>/spectrum/oneclick.jnlp?alarm=0x3780003d@7710
http://<hostname>:<port>/spectrum/webapp?alarm=0x3780003d@7710
```

where 0x3780003d@7710 is <*modelhandle*>@<*alarm ID*> and

```
http://<hostname>/spectrum/oneclick.jnlp?alarm=7710
http://<hostname>:<port>/spectrum/webapp?alarm=0x3780003d@7710
```

where 7710 is the integer <*alarm ID*>.

If you pass the integer alarm ID, pass the model handle also. The integer alarm ID is not guaranteed to be unique across SpectroSERVERs. The full global alarm ID is preferable as it is unique across SpectroSERVERs, but it may not be available to the application launching OneClick.

> **NOTE**
> When launching in context, a new instance of OneClick is not launched if an instance is already running on the host. The context is changed in the current instance of OneClick.

Configure OneClick Client Memory Settings

The default initial memory footprint for OneClick clients is 96 MB, with a maximum of 1024 MB. The initial memory setting lets the Java Virtual Machine (JVM) preallocate memory for potentially faster startup. The maximum setting lets the JVM memory grow into a limited space, to accommodate application use that requires additional memory. For example, large views, searches, and other actions can require more memory.

If clients experience out-of-memory errors, you can increase the maximum memory setting.

If you change the client memory settings, keep in mind that the settings apply to all OneClick clients. Therefore, take into account any client computers that lack sufficient resources. Use a modest adjustment, such as a 25% increase in the maximum memory allocation, to 640 MB. The steps in this procedure are an *optional* method for addressing out-of-memory issues.

> **WARNING**
> Setting either one of these memory values too high can cause the OneClick clients to fail to launch.

To configure OneClick client memory settings:

1. Click Administration in the OneClick home page.
2. The Administration Pages open.
3. Click OneClick Client Configuration in the left panel.
   The OneClick Client Configuration page opens.
4. Complete the fields in the Java Memory Usage section.
5. Click Save.

To change the Web Server memory:

1. Click on the Administration tab on the OneClick home page.
2. Click on the web server memory in the left panel
3. Enter the memory (MB) the server can use.
4. Select the Save and Reconfigure button to save the memory setting.

> **NOTE**
>
> Increasing the Web Server Memory depends on the OneClick Server physical memory(RAM) availability.

Configure the OneClick Web Server URL

The Using OneClick section describes the OneClick home page as a central place where users can launch the OneClick client. By default, all OneClick users must use the following URL to access the OneClick home page:

```
http://<<oc> web server>/spectrum
```

Also by default, the URL http://<OneClick web server> launches a Tomcat web server configuration page. You can configure the OneClick web server to automatically redirect from http://<*OneClick web server*> to http://<*OneClickweb server*>/spectrum.

Follow these steps:

1. Navigate to the <$*SPECROOT*>\tomcat\webapps\ROOT directory.
2. Create a file named index.html using your preferred text editor.
3. Edit the index.html file to contain the following text:

```
<html>
 <head>
  <meta http-equiv="refresh" content="0;url=/spectrum">
 </head>
```

```
  <body>
  </body>
</html>
```

4. Save the index.html file in the ROOT directory referenced in Step 1.
   All OneClick users navigating to http://<*OneClick web server*> are now redirected automatically to http://<*OneClick web server*>/spectrum.

Configure OneClick MySQL Server Passwords

You can change the passwords for both the default OneClick MySQL user (OC_user) and the administrative OneClick MySQL user (OC_admin). Change passwords on the MySQL Password Administration page.

> **WARNING**
> Do not attempt to manually change the MySQL user passwords using a MySQL client connection. Storage of the passwords in OneClick depends on MySQL connectivity. As a result, the only safe way to change the passwords is through the OneClick MySQL Password Administration page.

Follow these steps:

1. Click Administration in the OneClick home page.
   The Administration Pages open.
2. Click MySQL Password in the left panel.
   The Change MySQL Password page opens.
3. Enter the current password and the new password for the user whose credentials you want to modify.
4. Confirm the new password.
5. Click Change Password.
   The password changes immediately. Restarting MySQL or Tomcat is not required.

OneClick Server Communications and Network Configuration

This section discusses OneClick server communications and network configuration.

Name Resolution Requirements

For the OneClick web server system to communicate with a SpectroSERVER, the OneClick web server system must be able to resolve the non-fully-qualified hostname of the SpectroSERVER to an IP that can be used to reach the SpectroSERVER.

We recommend using hosts files for the name resolution of SpectroSERVER hostnames. This practice makes it less likely that name resolution is impacted by a network failure.

Configure OneClick for Secure Sockets Layer

OneClick supports the Secure Sockets Layer (SSL) protocol to encrypt communications between the OneClick web server and OneClick clients. OneClick clients can access information securely across unsecured networks, such as the Internet. In addition to encryption, SSL uses certificates for authentication. Authentication protects users from downloading and running applications from suspicious or "untrusted" sources.

Both Certificate Authority-signed certificates and self-signed certificates provide secure connections using SSL encryption. However, certificates signed by a Certificate Authority provide an additional level of security. These certificates verify the creator of the certificate and certify that the product is truly from that vendor. Certificates that are signed by a Certificate

Authority protect servers by making it difficult to impersonate a trusted entity (the certified vendor). However, self-signed certificates are appropriate if you require the encryption that an SSL certificate provides without requiring proof of the certificate source.

NOTE

After upgrading to DX NetOps Spectrum 10.3, when you configure OneClick for SSL, you will see a warning message to migrate JKS to PKCS12 format. Please ignore this warning and do not migrate to PKCS12 format.

NOTE

The tomcat bundled with DX NetOps Spectrum 10.3, requires at least one trusted cert in a keystore, for it to work.

Follow these steps:

1. On the OneClick web server host, change to the *$SPECROOT*/Java/bin directory.
2. Generate a private self-signed certificate in the custom cacerts file by issuing the following command:

   ```
   ./keytool -genkey -alias tomcatssl -keyalg RSA
   -keystore c:/win32app/Spectrum/custom/keystore/cacerts
   ```

   The keytool prompts with a series of questions and uses the values that you specify to perform the following actions:
   - Create an issuer name for your organization (This name is an X.500 Distinguished Name that is intended to be unique across the Internet. For more information, see the keytool utility at http://java.sun.com).
   - Generate the self-signed certificate using the issuer name.

     NOTE

     In case the keystore is not saved to $SPECROOT/custom/keystore, it is overwritten during an upgrade.
3. Enter your answers to the following questions:

   ```
   Enter keystore password:
   ```

   If you change the default password for the Tomcat web server, specify the custom password in the *$SPECROOT*/tomcat/conf/server.xml configuration file.

   ```
   What is your first and last name?
   ```

   Enter the common name (with the fully qualified domain name) of your website. For example, www.ca.com.

   ```
   What is the name of your organizational unit?
   ```

   Enter a small organization name, such as the name of a division, business unit, or department. For example, Purchasing.

   ```
   What is the name of your organization?
   ```

   Enter a large organization name, such as ABCSystems, Inc.

   ```
   What is the name of your City or Locality?
   ```

   Enter your city name, such as Hyderabad.

   ```
   What is the name of your State or Province?
   ```

   Enter the full name, such as Andhra Pradesh.

   ```
   What is the two-letter country code for this unit?
   ```

   Enter the two-letter country code. For example, IN.

   ```
   Is CN=www.ca.com, OU=Purchasing, O="ABCSystems, Inc.", L=Hyderabad, ST=Andrapradesh, C=IN correct?
   ```

   Enter Yes.

   ```
   Enter key password for <tomcatssl> (RETURN if same as keystore password):
   ```

   Enter key password for <tomcatssl>. Press Enter to use the same password as the keystore password.

   WARNING

   After adding the tomcatssl key, ensure you take a backup of the $SPECROOT/custom/keystore/cacerts file, in case the keystore gets corrupted.

4. (Optional) If you require a certificate that is signed by a Certificate Authority, request the certificate from the Certificate Authority and then import it.

   NOTE
   Before proceeding with this step (Step 4), as a best practice, skip to Step 5 and set up SSL. You can then test to determine whether the information that you provided in the previous step was correct. If HTTPS works, you can continue with this step.

   As part of certificate configuration, generate a Certificate Signing Request (CSR) file from the system that runs the secure OneClick web server. The Java Development Kit (JDK) that is included with OneClick provides a keytool utility that you can use to generate the CSR file. Use the information that you provided in the previous step. Use the same alias name as tomcatssl.
5. Request and import the Certificate Authority-signed certificate as follows:
6. On the OneClick web server host, change to the *$SPECROOT*/Java/bin directory.
   a. Generate the CSR file by entering the following command:

      ```
      ./keytool -certreq -alias tomcatssl
      -keystore $SPECROOT/custom/keystore/cacerts -file filename.csr
      ```

      NOTE
      You are prompted for a password. Use the same password that you provided earlier.The contents of the .csr file that is generated are used to request the secure certificate from the Certificate Authority (the next step).

      Request a secure certificate from a Certificate Authority. Verify the following examples:

      VeriSign: http://www.verisign.com

      TrustCenter: http://www.trustcenter.de

      thawte: http://www.thawte.com

      Instructions are available at these websites.

      Import the Certificate Authority-signed certificate into the keystore that is used by the OneClick web server. For more information, see Import a Certificate Authority-Signed Certificate.
7. Configure the secure socket on the server that hosts the OneClick web server. For more information, see Configure the Secure Socket on the OneClick Web Server Host.
8. If you are running Report Manager, configure OneClick to be launched from Report Manager using SSL. For more information, see Configure OneClick and Report Manager for Secure Sockets Layer.

Import a Certificate Authority-Signed Certificate

If you have obtained a Certificate Authority-signed SSL certificate, import it into the keystore that the OneClick web server uses.

A chain (root) certificate from the Certificate Authority must also exist in the keystore. By default, OneClick includes chain certificates from many popular vendors. Click List on the SSL Certificates administration page to view the aliases for these certificates. This information helps you determine whether to obtain one and import it.

Follow these steps:

1. If necessary, download a chain (root) certificate from the Certificate Authority from which you obtained the signed certificate.
2. If you downloaded a chain certificate in the previous step, import it into the keystore used by the OneClick web server:
   a. On the OneClick web server host, change to the *$SPECROOT*/Java/bin directory.
   b. Enter the following command:

      ```
      ./keytool -import -alias root -keystore $SPECROOT/custom/keystore/cacerts -trustcacerts -file
       root_chain_certificate_filename
      ```

NOTE
You are prompted for a password for the Tomcat web server. The alias name does not have to be 'root'. You can supply a more descriptive name for the type of root certificate that you are importing. The alias name cannot already exist.

3. Import the Certificate Authority-signed SSL certificate into the keystore used by the OneClick web server:
   a. If necessary, on the OneClick web server host, change to the *$SPECROOT*/Java/bin directory.
   b. Enter the following command:

```
./keytool -import -alias tomcatssl -keystore $SPECROOT/custom/keystore/cacerts -trustcacerts -file
 your_certificate_filename
```

NOTE
You are prompted for a password for the Tomcat web server. Use the same alias that you used when you generated the private self-signed certificate. See Name Resolution Requirements for more information.

NOTE

keytool can import X.509 v1, v2, and v3 certificates, and PKCS#7 formatted certificate chains consisting of certificates of that type. Please verify that the certificate from CA is of this type.

(Optional) Log in Using Non-Fully Qualified Domain Name

SSL security forces you to use the fully qualified domain name of your OneClick server for login. For example: `https://oneclick.ca.com/spectrum` . To log into the non-fully qualified domain name (for example: `https://oneclick/spectrum`), or a DNS entry that is different than the local OneClick server name, use a SAN (Subject Alternate Name) with the -ext option:

```
./keytool -genkey -alias tomcatssl -keyalg RSA -keysize 2048 -ext SAN=dns:oneclick -
keystore c:/win32app/Spectrum/custom/keystore/cacerts
```

Modify JVM Arguments when SSL is Enabled

Perform the procedure described in the Post Upgrade Tasks section on the OneClick WebApppage, in case you enabled SSL in OneClick or changed the OneClick ports after you upgraded to DX NetOps Spectrum 10.4 or higher.

NOTE
To disable the non-HTTPS connector port, see the KB Article: Block access to HTTP on port 80 (Windows) or port 8080 (Linux) in Spectrum OneClick.

Configure the Secure Socket on the OneClick Web Server Host

Configure the secure socket on the server that hosts the OneClick web server. Consider this task as the final step in configuring the OneClick web server for SSL.

NOTE
DX NetOps Spectrum supports the use of SSL v3.

Follow these steps:

1. Shut down the OneClick web server.
2. Open $SPECROOT/tomcat/conf/server.xml in your preferred text editor.
3. Locate the following section in the server.xml file:

```
<!-- Define a SSL Coyote HTTP/1.1 Connector on port 8443 -->
<!--
 <Connector
  port="8443"
  enableLookups="true" disableUploadTimeout="true" tcpNoDelay="true"
```

```
acceptCount="100" scheme="https" secure="true" SSLEnabled="true"
clientAuth="false" sslProtocol="TLS"
sslEnabledProtocols="TLSv1.2"
ciphers="TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,
  TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,
  TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA,
  TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA,
  TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256,
  TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384,
  TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,
  TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,
  TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA,
  TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA,
  TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256,
  TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384,
  TLS_DHE_RSA_WITH_AES_128_GCM_SHA256,
  TLS_DHE_RSA_WITH_AES_256_GCM_SHA384,
  TLS_DHE_RSA_WITH_AES_128_CBC_SHA,
  TLS_DHE_RSA_WITH_AES_256_CBC_SHA,
  TLS_DHE_RSA_WITH_AES_128_CBC_SHA256,
  TLS_DHE_RSA_WITH_AES_256_CBC_SHA256"
keystoreFile="custom/keystore/cacerts"
keystorePass="changeit">

</Connector>
-->
```

By default the <Connector> element in the section is commented out.

> **NOTE**
> The preceding XML fragment is Windows-specific, with 443 as the default port where the OneClick web server listens for SSL communications. End users can omit the port from the URL for accessing the OneClick home page:

```
https://<fully_qualified_host_name>/spectrum
```

On a UNIX-based installation, the OneClick web server is not run as root, and the default port is 8443 (because it must be greater than 1024). As a result, end users must specify the port number in the web browser when they enter the URL to access the OneClick home page:

```
https://<fully_qualified_host_name>:8443/spectrum
```

4. Remove the comments around the Connector definition. Perform the following actions:
   a. Remove "<!--" from the line preceding to <Connector.
   b. Remove "-->" from the end of the section (after </Connector>).
5. Replace the <*SPECROOT*> variable in the value for the keystoreFile attribute with the fully qualified path to the directory where DX NetOps Spectrum is installed. You can use the cacerts file for the keytool commands to generate the certificates. Verify the following examples:
   – **Windows**

     ```
     C:/win32app/SPECTRUM/custom/keystore/cacerts
     ```

   – **UNIX**

     ```
     /usr/SPECTRUM/custom/keystore/cacerts
     ```

6. Save and close the server.xml file.
7. If you have DX NetOps Spectrum integrated with CA Performance Center, perform the following steps to enable the communication between SSL enabled OneClick and CA Performance Center:
   a. Open the "axis2.xml" file in an editor from "$SPECROOT/tomcat/webapps/axis2/WEB-INF/conf".

b. Locate the following section in axis2.xml:

```
<transportReceiver name="http"
                        class="org.apache.axis2.transport.http.AxisServletListener">
        <parameter name="port">8080</parameter>
</transportReceiver>
```

c. Change the section as follows:

```
<transportReceiver name="https"
                        class="org.apache.axis2.transport.http.AxisServletListener">
        <parameter name="port">8443</parameter>
</transportReceiver>
```

If you need to configure both HTTP and HTTPS, it is necessary to explicitly configure the port numbers in axis2.xml, such as in the following example:

```
<transportReceiver name="http"
 class="org.apache.axis2.transport.http.AxisServletListener">
<parameter name="port">8080</parameter>
</transportReceiver>

<transportReceiver name="https"
 class="org.apache.axis2.transport.http.AxisServletListener">
<parameter name="port">8443</parameter>
</transportReceiver>
```

8. Start the OneClick web server.
   You can find instructions on configuring SSL and configuration parameters. For more information, see http://tomcat.apache.org/tomcat-9.0-doc/ssl-howto.html.

   NOTE
   To disable the non-HTTPS connector port, see the KB Article: Block access to HTTP on port 80 (Windows) or port 8080 (Linux) in Spectrum OneClick.

Configure OneClick and Report Manager for Secure Sockets Layer

If you are running Report Manager and you have configured OneClick to use the Secure Sockets Layer (SSL) protocol to encrypt communications between OneClick clients and the OneClick web server, you must also configure OneClick to be launched from Report Manager using SSL.

Report Manager allows you to create reports on the inventory, performance, change history, and fault history of the network assets managed by DX NetOps Spectrum. For more information, see Report Manager.

Follow these steps:

1. Enable write permissions on the following file:
   `<$SPECROOT>\tomcat\webapps\spectrum\repmgr\js\repmgr.js`
2. Open the file that you modified in the previous step, and locate the launchOneClick function.
3. Locate the following line in the launchOneClick function:
   `url = "http://"+servername+contextApp+"/oneclick.jnlp";`
4. Change "http" to "https" as follows:
   `url = "https://"+servername+contextApp+"/oneclick.jnlp";`
5. Save and close the file.

   WARNING
   This file is overwritten during an upgrade. Repeat this procedure after an upgrade.

NOTE
You can launch OneClick in the context of a specific report (for example, in the context of a device that is listed in an asset report). However, this type of launch cannot be configured to use SSL.

Unable to connect to OneClick using https after upgrading to 10.3

Symptom:

After upgrading to DX NetOps Spectrum 10.3, unable to connect to OneClick using the https connection.

This problem occurs due to the migration of JKS to PKCS12 format. After generating the SSL key, the following message appears, which recommends to migrate JKS to PKCS12 format:

```
The JKS keystore uses a proprietary format. It is recommended to migrate to PKCS
12 which is an industry standard format using "keytool -importkeystore -srckeyst
ore c:/win32app/SPECTRUM/custom/keystore/cacerts -destkeystore c:/win32app/SPECT
RUM/custom/keystore/cacerts -deststoretype pkcs12".
```

Solution:

We recommended to ignore this warning message and do not migrate to PCKS12 format. In case you have already migrated, then replace the backed up keystore while migrating to PKCS12 format and restart Tomcat.

Follow these steps:

1. While migrating to PKCS12 format, the old keystore gets backed up as /usr/Spectrum/custom/keystore/cacerts.old
2. Remove or backup the file /usr/Spectrum/custom/keystore/cacerts and rename cacerts.old to cacerts at /usr/Spectrum/custom/keystore/
3. Restart tomcat.

Error when logging into OneClick web configured for SSL

Symptom:

After configuring the DX NetOps Spectrum OneClick server for SSL, an error is thrown when attempting to connect from a browser (Chrome, FireFox, IE or Safari).

This error occurs due to the URL being used to point the browser at the OneClick server. The URL contains either an IP address or hostname that does not match that which was used to generate the certificate that was added to the OneClick server keystore. Or alternately, the DNS lookup does not resolve to the correct name/IP.

Solution:

Follow these steps:

1. When generating the private, self-signed certificate, you use the following command:
   ```
   ./keytool -genkey -alias tomcatssl -keyalg RSA -keystore $SPECROOT/custom/keystore/cacerts
   ```
   This command then asks a number of questions, the second of which is: **What is your first and last name?**
   This refers to the common name (singular hostname) or the FQDN of the OneClick server . So when logging in with the browser, you need to refer to this hostname in the URL (not the IP address) for the HTTPS connection to work and the certificate to be validated by the browser.
2. Import the certificate signed by your CA:
   ```
   $SPECROOT/Java/bin> ./keytool -import -alias tomcatssl -keystore $SPECROOT/custom/keystore/cacerts -trustcacerts -file <PATH>/<FILENAME.cer>
   ```
3. Enter the keystore password. The certificate reply was installed in the keystore. If your DNS is not resolving the hostname of the OneClick server, then modify your hosts file (In Windows: ~\win32\drivers\etc\hosts, in Linux/etc/

hosts) file to include both the singular and FQDN hostnames of the OneClick server so as to get around the problems with your DNS. Then in the browser, target the OneClick server URL using:

```
https://<HOSTNAME>:443/spectrum
```

Errors Connecting to the Secure OneClick Web Server from a OneClick Client Using SSL

Symptom:

I am encountering errors when I try to connect to the secure OneClick web server from a OneClick client using SSL.

Solution:

Verify the following:

- The fully qualified domain name of the host on which the OneClick web server is running was specified in the private key you generated for signing the security certificate used for authentication. When you generated the key, you should have entered the fully qualified domain name at the following prompt: “What is your first and last name?”
- Both the Certificate Authority chain (root) certificate *and* the security certificate were imported into the cacerts file in the custom directory on the secure OneClick web server.

Errors Launching OneClick Client from Report Manager Using SSL

Symptom:

I am encountering errors when I launch a OneClick client from Report Manager using SSL.

Solution:

Verify that you have completed the configuration procedure described in Configure OneClick and Report Manager for Secure Sockets Layer.

Enable ModSecurity Web Application Firewall

To prevent malicious remote clients from accessing OneClick server (Tomcat), and for full HTTP traffic logging, you must enable the ModSecurity Web Application Firewall (WAF). ModSecurity is deployed in DX NetOps Spectrum environment using the reverse proxy method. If you edit the "httpd.conf" file and set the configuration shown in this method is achieved by using a dedicated Apache server as a reverse proxy for Tomcat, and by adding the open-source ModSecurity module to it. With this implementation, you get a proper web application firewall. When you install the OneClick server on Windows, the "apache" folder is created in the $SPECROOT directory. This folder includes the following items:

- Apache HTTP server 2.4.35 package is required to install and to start the Apache server.

  NOTE

  In the 10.3.1 release, Apache HTTP is upgraded from 2.4.12 to 2.4.35. After upgrade to 10.3.1, you must re-enable the ModSecurity firewall. For existing configuration, you can refer to the "conf.old" available in the $SPECROOT/apache directory.
- Open source ModSecurity 2.9 package that is required to run the Apache server as a reverse proxy.
- Open source ModSecurity core rule set 2.2.9 package for the firewall capability.

When you install the OneClick server on Linux, a pre-built Apache server 2.4.35 with ModSecurity 2.9 and 2.2.9 core rule set is created in the $SPECROOT directory.

NOTE

- You can enable ModSecurity only with any of the following two Apache setups:
  - The Apache setup of DX NetOps Spectrum environment on your OneClick host.
  - An existing Apache setup on your OneClick host.
- Enabling ModSecurity with an Apache setup that is running outside the OneClick host is not supported.

This page contains the following topics:

How ModSecurity Works

When ModSecurity is enabled in the reverse proxy deployment, the following firewall architecture is enabled:

- Apache server becomes an HTTP router that is designed to stand between the OneClick server and its clients.
- Clients connect only to the Apache server.
- Apache forwards and gets the request from Tomcat for clients.
- Access to the Tomcat server for clients is disabled.

This architecture is configured by setting the following attributes and directives in the "httpd.conf" file that is located at "$SPECROOT\apache\conf" as shown here:

```
Listen 80
<VirtualHost *:80>
  ProxyPreserveHost On
  ProxyPass /spectrum http://localhost:8081/
  ProxyPassReverse /spectrum http://localhost:8081/
</VirtualHost>


For this example, assume that the tomcat server was listening at port 80. To enable
 ModSecurity,
Tomcat server port is assigned to the Apache server. A free port 8081 is assigned to the
 Tomcat
server by adding port 8081 in the server.xml file that is located at "$SPECROOT\tomcat
\conf".
```

If you edit the "httpd.conf" file and set the configuration shown in the example, the Apache server runs as a proxy of the Tomcat server as follows:

- Tomcat port 80 is assigned to the Apache server.
- Apache server becomes the virtual host that is mapped to the OneClick url. As a result, the client request with the url "http://<hostname><:80>/spectrum" connects to the Apache server.
- The directive **ProxyPass** instructs the Apache server to pass all client requests to the Tomcat server which now listens at 8081.
- The directive **ProxyPassReverse** rewrites the HTTP Header in the response of Tomcat to make it look for clients as if it came from Apache.

In DX NetOps Spectrum environment, ModSecurity is enabled using the "configApacheModsec.sh" script. This script is located at "$SPECROOT\apache\bin". It enables ModSecurity by performing the following functions:

1. Lets you assign the port of Tomcat server to the Apache server, and lets you assign another free port to the Tomcat server.
2. Updates the newly assigned port to the Tomcat server in the "server.xml" file.
3. Configures the "httpd.conf" file based on these port assignments.
4. Installs and starts the Apache service with ModSecurity.

If the Tomcat server port is assigned to the Apache server, clients can use the existing OneClick url to connect to the Apache server. Otherwise, you need to assign a free port to the Apache server. In this case, clients can connect to the Apache server using the OneClick url only when they use the newly assigned port to the Apache server in that url. You must provide to clients the updated OneClick url which contains the newly assigned port to the Apache server. Based on

whether you want the clients to use the existing OneClick url or not, you can enable ModSecurity using any one of the following two methods:

- Enable ModSecurity Using the Tomcat port for Apache
- Enable ModSecurity Using a Free Port for Apache

> **NOTE**
> When the ModSecurity is enabled, the OneClick Console default customization options like the branding (logos, images) are disabled. To configure the ModSecurity to load resources from non-default location, see Modify Apache ModSecurity Configuration To Enable OneClick Console Customization.

Enable ModSecurity Using the Tomcat port for Apache

By default, Apache listens on port 8080. When you assign the existing tomcat port to Apache, the clients can use the existing url without changing the port number. In this case, the Tomcat server is assigned another port, which is disabled for external clients. Internal client is the client that accesses from the OneClick server host itself. For the following procedure assume that the existing Tomcat port is 80.

Follow these steps:

1. On Windows, launch the "Services.msc" program, or execute the following command at the command prompt to stop the "SpectrumTomcat" service:
   $SPECROOT\NT-Tools\SRE\bin\bash.exe "$SPECROOT\\tomcat\\bin\\stopTomcat.sh"
   On Linux, execute the following command at the bash prompt (from **$SPECROOT\tomcat\bin**) to stop the SpectrumTomcat service:
   ./stopTomcat.sh
   The "SpectrumTomcat" service stops, and the Tomcat port 80 is now free to be assigned to the Apache server.
2. On Linux, execute the following command at the bash prompt (from **$SPECROOT\apache\bin**) to enable ModSecurity:
   ./configApacheModsec.sh enable On Windows, execute the following command at the command prompt to enable ModSecurity:
   $SPECROOT\NT-Tools\SRE\bin\bash.exe "$SPECROOT\\apache\\bin\\configApacheModsec.sh" "enable"
   You are prompted to confirm whether tomcat is running in SSL mode or not.
   If Tomcat is running in SSL mode, follow the steps in Enable ModSecurity in SSL Mode. The script displays the following message, and does not enable ModSecurity:
   Screenshots for SSL mode
   Windows

```
pangy01-w2k8vm1%/c/win32app/Spectrum/apache/bin > ./configapachescriptfinal.sh enable
Is Tomacat running on SSL(Y/N) :
y
Disable SSL in SpectrumTomcat and run the script
```

   Linux

```
[root@cumulus-rh7vm5 bin]# ./configApacheModsec.sh enable
Is Tomacat running on SSL(Y/N) :
y
Disable SSL in SpectrumTomcat and run the script
[root@cumulus-rh7vm5 bin]#
```

   Screenshots for non-SSL mode
   Windows
   If Tomcat is running in non-SSL mode, the script prompts you to select whether you want to assign the tomcat port to Apache or not.

```
Do u want to assign Tomcat Port  to Apache(Y/N):
y
Please enter some free port for Tomcat:
8081
Apache Service is started.
```

Linux

```
[root@cumulus-rh7vm5 bin]# ./configApacheModsec.sh enable
Is Tomacat running on SSL(Y/N) :
n
Do u want to assign Tomcat Port 8080 to Apache(Y/N):
y
Please enter some free port for Tomcat:
8081
Apache Service is started.
[root@cumulus-rh7vm5 bin]#
```

3. Press the 'y' key, and then press Enter.
   Port 80 is assigned to Apache. The script prompts you to input a free port for Tomcat.
4. Enter a new port for Tomcat, and then press Enter.
   For example, enter 8081.
   The script applies the following configuration in the httpd.conf file:

```
Listen 80

<VirtualHost *:80>

ProxyPreserveHost On

ProxyPass          /spectrum http://localhost:8081/spectrum
ProxyPassReverse   /spectrum http://localhost:8081/spectrum
</VirtualHost>
```

   You get the "Apache Service is started" message at the command prompt.
5. On Linux, execute the following command at the bash prompt to start the "SpectrumTomcat" service:
   ./startTomcat.sh
   On Windows, Launch "Services.mcs" program, or execute the following command at the command prompt to start the SpectrumTomcat service:
   $SPECROOT\NT-Tools\SRE\bin\bash.exe "$SPECROOT\\tomcat\\bin\\startTomcat.sh
6. On Linux, execute the following command at the bash prompt to verify that the Apache service has started:

```
[root@scrh63-vm3 bin]# ps -eaf | grep httpd
root      4356     1  0 02:53 ?        00:00:00 /usr/Spectrum/apache/bin/httpd -d /usr/Spectrum/apache/ -k start
daemon    4357  4356  0 02:53 ?        00:00:01 /usr/Spectrum/apache/bin/httpd -d /usr/Spectrum/apache/ -k start
daemon    4358  4356  0 02:53 ?        00:00:01 /usr/Spectrum/apache/bin/httpd -d /usr/Spectrum/apache/ -k start
daemon    4359  4356  0 02:53 ?        00:00:01 /usr/Spectrum/apache/bin/httpd -d /usr/Spectrum/apache/ -k start
daemon    4489  4356  0 02:56 ?        00:00:01 /usr/Spectrum/apache/bin/httpd -d /usr/Spectrum/apache/ -k start
root      5959 20286 11 04:26 pts/1    00:00:00 grep httpd
```

7. On Windows, launch the "Services.msc" program to verify that the Apache service has started.

When clients use the existing url "http://<hostname:80>/spectrum", they connect to the Apache server and get the response from it. To disable the Tomcat port 8081 for external clients, the Loopback address is added in the server.xml file that is located at "$SPECROOT\tomcat\conf".

Enable ModSecurity Using a Free Port for Apache

When you do not assign the existing tomcat port to Apache, the clients have to use the url with a newly assigned port to the Apache server. In this case also, the existing tomcat port is disabled for external clients. For the following procedure assume that the existing Tomcat port is 80.

Follow these steps:

1. On Linux, execute the following command at the bash prompt (from **$SPECROOT\apache\bin**) to enable ModSecurity:
   ./configApacheModsec.sh enable
   On Windows, execute the following command at the command prompt to enable ModSecurity:
   $SPECROOT\NT-Tools\SRE\bin\bash.exe "$SPECROOT\\apache\\bin\\configApacheModsec.sh" "enable"
   You are prompted to confirm whether tomcat is running in SSL mode or not.
   If Tomcat is running in SSL mode, follow the steps in Enable ModSecurity in SSL Mode. The script displays the following message, and does not enable ModSecurity:
   Screenshots for SSL mode
   Windows

```
pangy01-w2k8vm1%/c/win32app/Spectrum/apache/bin > ./configapachescriptfinal.sh enable
Is Tomacat running on SSL(Y/N) :
y
Disable SSL in SpectrumTomcat and run the script
```

   Linux

```
[root@cumulus-rh7vm5 bin]# ./configApacheModsec.sh enable
Is Tomacat running on SSL(Y/N) :
y
Disable SSL in SpectrumTomcat and run the script
[root@cumulus-rh7vm5 bin]#
```

   Screenshots for non-SSL mode
   If Tomcat is running in non-SSL mode, the script prompts you to select whether you want to assign the tomcat port to Apache or not:

   Windows

```
pangy01-w2k8vm1%/c/win32app/Spectrum/apache/bin > ./configapachescriptfinal.sh enable
Is Tomacat running on SSL(Y/N) :
n
Do u want to assign Tomcat Port 80 to Apache(Y/N):
n
Assign a new port for Apache
8082
Apache service is Registered Successfully.
Apache Service is started.
```

   Linux

```
[root@cumulus-rh7vm5 bin]# ./configApacheModsec.sh enable
Is Tomacat running on SSL(Y/N) :
n
Do u want to assign Tomcat Port 8080 to Apache(Y/N):
n
Assign a new port for Apache
8082
Apache Service is started.
[root@cumulus-rh7vm5 bin]#
```

2. Press the 'n' key, and then press Enter.
   The script prompts you to enter a free port for the Apache server as shown in the earlier image. In this example, 8080 is assigned to the Apache server.
   The script applies the following configuration in the "httpd.conf" file:

```
Listen 8080
<VirtualHost *:8080>

ProxyPreserveHost On

ProxyPass           /spectrum http://localhost:8080/spectrum
ProxyPassReverse    /spectrum http://localhost:8080/spectrum
</VirtualHost>
```

   You get the "Apache Service is started" message at the command prompt.
   Note: If the CA PC integration is enabled, and if apache is running in a non SSL mode, then the following script is applicable in the httpd-ssl.conf or httpd.conf file, for the integration to work. You may replace 8443, with the actual tomcat port that is configured.

```
ProxyPass /axis2 https://localhost:8443/axis2

ProxyPassReverse /axis2 https://localhost:8443/axis2
```

3. On Linux, execute the following command at the bash prompt to verify that the Apache service has started:

```
[root@scrh63-vm3 bin]# ps -eaf | grep httpd
root      4356     1  0 02:53 ?        00:00:00 /usr/Spectrum/apache/bin/httpd -d /usr/Spectrum/apache/ -k start
daemon    4357  4356  0 02:53 ?        00:00:01 /usr/Spectrum/apache/bin/httpd -d /usr/Spectrum/apache/ -k start
daemon    4358  4356  0 02:53 ?        00:00:01 /usr/Spectrum/apache/bin/httpd -d /usr/Spectrum/apache/ -k start
daemon    4359  4356  0 02:53 ?        00:00:01 /usr/Spectrum/apache/bin/httpd -d /usr/Spectrum/apache/ -k start
daemon    4489  4356  0 02:56 ?        00:00:01 /usr/Spectrum/apache/bin/httpd -d /usr/Spectrum/apache/ -k start
root      5959 20286 11 04:26 pts/1    00:00:00 grep httpd
```

4. On Windows, launch the "Services.msc" program to verify that the Apache service has started.

When clients use the existing url "http://<hostname:80>/spectrum", they cannot connect to Apache server. Clients must use the updated url "http://<hostname:8082>/spectrum" to connect to the Apache server and get the response from it.
To disable the Tomcat port 80 for external clients, the Loopback address is added in the server.xml file that is located at "$SPECROOT\tomcat\conf".

Preventing Clickjack attack using ModSecurity

Clickjacking (User Interface redress attack) is a malicious practice of manipulating an activity of a website user by concealing hyperlinks beneath legitimate clickable content, thereby causing the user to perform actions of which they are unaware. It is a browser security issue that is a vulnerability across a variety of browsers and platforms. A clickjack takes the form of embedded code or a script that can execute without the user's knowledge, such as clicking on a button that appears to perform another function.

Defending Clickjacking with X-Frame-Options

You can use the X-Frame-Options HTTP response header to indicate whether or not a browser should be allowed to render a page in a <frame> or <iframe>.

Follow these steps to update the 'httpd.conf' file for protecting DX NetOps Spectrum from Clickjack attack:

1. Open the "$SPECROOT\apache\conf\httpd.conf" file with any text editor
2. Add the following X-Frame Options Response Header and save the file:
   ```
   Header always append X-FRAME-OPTIONS "SAMEORIGIN"
   ```
3. Restart the Apache server.

Cache-Controling using ModSecurity

Using ModSecutiy you can control the cache by defining a mod_headers in 'httpd.conf' file. This allows a server to return control how, and for how long, the browser and other intermediate caches can be cached for an individual response.

Follow these steps:

1. Open the "$SPECROOT\apache\conf\httpd.conf" file with any text editor
2. Add the following configuration and save the file:
   ```
   <IfModule mod_headers.c> Header set Cache-Control "no-cache, no-store, must-revalidate,max-age=0" Header
    set Pragma "no-cache" Header set Expires 0 </IfModule>
   ```
3. Restart the Apache server.

After updating the 'httpd.conf' file with the above cache configuration, you can observe that all the calls going from DX NetOps Spectrum are controlled with the cache. The cache control directives are displayed in the responsive headers.

Disable ModSecurity

On Linux, execute the following command (from **$SPECROOT\apache\bin**) at the bash prompt to disable ModSecurity:

```
[root@scrh63-vm3 bin]# ./configApacheModsec.sh disable
Apache Service is unregistered Successfully.
```

On Windows, use the command prompt with the following syntax to disable ModSecurity:

```
C:\win32app\Spectrum\NT-Tools\SRE\bin\bash.exe "C:\\win32app\\Spectrum\\apache\\bin\
\configApacheModsec.sh" "disable"
```

If you enabled ModSecurity using the Tomcat port for the Apache server, the script disables ModSecurity by making the following changes:

- The Apache service is stopped, uninstalled, and the configuration that is applied to the "httpd.conf" file becomes inapplicable.
- Port 80 is assigned back to the Tomcat server, and port 8080 is assigned back to the Apache server.
- The Loopback address is removed and port 80 is added back in the server.xml file. As a result, external clients can directly access the Tomcat server using port 80.

If you enabled ModSecurity using the default port for the Apache server, the script disables ModSecurity by making the following changes:

- The Apache service is stopped, uninstalled, and the configuration applied to "httpd.conf" file becomes inapplicable.
- The Loopback address is removed from the "server.xml" file.

WARNING

In this case, you must add the existing port number (which is 80 in this example) in the server.xml file.

Finally, you get the following message at the command prompt:

```
Apache Service is unregistered Successfully
```

Enable ModSecurity in SSL Mode

To enable ModSecurity in SSL mode, the Apache server is first configured to run in SSL mode. The following configuration tasks are performed to execute Apache in SSL mode:

- Editing the "$SPECROOT\apache\conf\extra\httpd-ssl.conf" file to configure the virtual host configuration (setting the Apache SSL port, proxypass and proxypassreverse directives) to map the OneClick url with the Apache SSL port.
- Uncommenting the "#Include conf/extra/httpd-ssl.conf" directive in the httpd.conf file so that the Apache server runs in the SSL mode when it is started.
- Configuring the log file paths in the "$SPECROOT\apache\conf\extra\httpd-ssl.conf" file for logging SSL logs.
- Configuring the paths of SSL certificate files (server.crt and server.key), and generating those files using the "openssl" command.

After performing these configurations enable ModSecurity in SSL mode by manually installing and starting the Apache service.

NOTE

- Do not use the "configApacheModsec.sh" script to enable ModSecurity in SSL mode.
- To enable ModSecurity in SSL mode, the Tomcat server must also run in SSL mode. If the Tomcat server is running in non-SSL mode, disable that mode and enable SSL mode. To configure Tomcat in SSL mode, follow the instructions provided in the Configure OneClick for Secure Sockets Layer section.

Follow these steps to Enable ModSecurity in SSL Mode:

1. Change the "httpd-ssl.conf" and the "httpd.conf" file from read-only mode to write mode.

   NOTE

   Before proceeding to step 2, search for the following attribute and comment / hide it in the "httpd.conf" file to avoid creating multiple host configuration.

   Listen 80
   <VirtualHost *:80>

   ProxyPreserveHost On
   ProxyPass /spectrum http://localhost:8081/
   ProxyPassReverse /spectrum http://localhost:8081/
   </VirtualHost>

2. Find the "<VirtualHost _default_:443>" tag then change it to "<VirtualHost *:443>" in the "httpd-ssl.conf". Add the following virtual host configuration in between <VirtualHost *:443> and </VirtualHost> tags as shown in the following example:

```
<VirtualHost *:443>
  ProxyPreserveHost on
  SSLEngine on
  SSLProxyEngine on
  SSLProxyVerify none
  SSLProxyCheckPeerCN off
  SSLProxyCheckPeerName off
  SSLProxyCheckPeerExpire off
  ProxyPass /spectrum https://localhost:8443/spectrum
  ProxyPassReverse /spectrum https://localhost:8443/spectrum
```

```
</VirtualHost>

*This configuration indicates that ModSecurity on httpd (Apache Web Server) is
 running in SSL mode on port TCP/443 and Tomcat is also running in SSL mode on port
 TCP/8443 locally.
```

NOTE
To ensure that the CA NIM customization functionality works in 10.4.1 when OneClick is in the SSL mode and ModSecurity is enabled, you must add the following configuration:

```
ProxyPass /ca-nim-sm https://localhost:8443/ca-nim-sm

ProxyPassReverse /ca-nim-sm https://localhost:8443/ca-nim-sm
```

NOTE
If the CA PC integration is enabled and Apache is running in an SSL mode, then the following script is applicable in the httpd-ssl.conf or httpd.conf file, for the integration to work. You may replace 8443, with the actual tomcat port that is configured.

```
ProxyPass /axis2 https://localhost:8443/axis2

ProxyPassReverse /axis2 https://localhost:8443/axis2
```

3. Edit the "httpd.conf" file to:
   a. Include the "httpd-ssl.conf" file in the "httpd.conf" file by removing the '#' symbol in front of the "Include conf/extra/httpd-ssl.conf" directive in the "httpd.conf" file.
   b. Load the module "mod_ssl.so" in the "httpd.conf" file by removing the '#' symbol in front of the "LoadModule ssl_module modules/mod_ssl.so" directive in the "httpd.conf" file.
   c. Replace the default path (<$SPECROOT>) with the absolute path (C:/win32app/Spectrum) for ServerRoot.
4. Replace the default path (<$SPECROOT>) with the absolute path (C:/win32app/Spectrum) for DocumentRoot, Errorlog, and TransferLog in the "httpd-ssl.conf" file.
5. Execute the following command on Windows to generate the "server.crt" and the "server.key" file:
   $SPECROOT\NT-Tools\SRE\bin\bash.exe "$SPECROOT\\apache\\conf\\openssl req -newkey rsa:1024 -keyout server.key -nodes -x509 -out server.crt"
   Execute the following command on Linux to generate the "server.crt" and the "server.key" file:
   $SPECROOT\Apache\conf>openssl req -newkey rsa:1024 -keyout server.key -nodes -x509 -out server.crt
   You are prompted to enter the following details.
   a. Enter your Country Name in the (2 letter code) [XX] format, and press Enter.
   b. Enter your State or Province Name, and press Enter.
   c. Enter your Locality Name, and press Enter.
   d. Enter your Organization Name, and press Enter
   e. Enter your Organizational Unit Name, and press Enter.
   f. Enter your Common Name in the following format, and press Enter.
      <host_name>@domain.com
   g. Enter your Email address in the following format:
      id@domain.com
      You are prompted to verify whether the information that you provided is correct or not.
   h. Type 'yes', and press Enter.
6. Do the following edits to the httpd-ssl.conf file to update the path of the server.crt and server.key certificate files:
   a. Find the "SSLCertificateFile "c:/Apache24/conf/server.crt" line, and update the path to "$SPECROOT/apache/conf/server.crt".
   b. Find the "SSLCertificateKeyFile "c:/Apache24/conf/server.key" line, and update the path to "$SPECROOT/apache/conf/server.key".

7. On Linux, execute the following command at the bash prompt to enable ModSecurity in SSL mode:
   ```
   #./httpd -d /$SPECROOT/apache -k start
   ```
8. On Windows, execute the following commands as an admin user and not as a DX NetOps Spectrum user, at the bash prompt to enable ModSecurity in SSL mode:
   ```
   C:\$SPECROOT\apache\bin>httpd.exe -k install

   C:\$SPECROOT\apache\bin>httpd.exe -k start
   ```
   ModSecurity is enabled with the Apache server running at the default SSL port 443. Now, clients must use the "https://<hostname><:443>/spectrum" url to connect to the Apache server. Execute the following steps to disable the existing Tomcat port (SSL) for the external clients, so that the tomcat cannot be accessed directly from external clients:
   a. Find the "<!-- Define a SSL Coyote HTTP/1.1 Connector on port 443 -->" segment in the server.xml file.
   b. Add the "address=127.0.0.1" attribute in the next <Connector /> tag segment.

      NOTE

      - On Linux, execute the "**./httpd -d /$SPECROOT/apache -k stop**" command at the bash prompt to disable ModSecurity in SSL mode.
      - On Windows, execute the "**httpd.exe -k stop**" command at the bash prompt to disable ModSecurity in SSL mode.
      - On Windows, execute the "**httpd.exe -k uninstall**" command at the bash prompt to uninstall the Apache server.

      After disabling ModSecurity, change the "httpd-ssl.conf" and the "httpd.conf" file from write mode to read-only mode.

Import Third-Party SSL Certificate

You can also import the SSL certificate of a third-party organization. When you raise a request for a third party SSL certificate, that particular third party organization gives the following files:

- server.crt (server certificate)
- server.key (private key)

NOTE

Restrict the access to these files to only the root user.

Follow these steps:

1. Download the "server.crt" and "server.key" files from the third party, and save it in the filesystem of the Apache server host.
2. Locate the following line in the httpd.conf file, and remove the '#' character in this line to load the "mod_ssl.so" module:
   ```
   #LoadModule ssl_module modules/mod_ssl.so
   ```
   Save the file.
3. Locate the following line in the httpd-ssl.conf file, and remove the existing path in the SSLCertificateFile "c:/Apache24/conf/server.crt" statement:
   ```
   #   Server Certificate:
   #   Point SSLCertificateFile at a PEM encoded certificate.  If
   #   the certificate is encrypted, then you will be prompted for a
   #   pass phrase.  Note that a kill -HUP will prompt again.  Keep
   #   in mind that if you have both an RSA and a DSA certificate you
   #   can configure both in parallel (to also allow the use of DSA
   #   ciphers, etc.)
   #   Some ECC cipher suites (http://www.ietf.org/rfc/rfc4492.txt)
   #   require an ECC certificate which can also be configured in
   ```

```
#   parallel.
SSLCertificateFile "c:/Apache24/conf/server.crt"
#SSLCertificateFile "c:/Apache24/conf/server-dsa.crt"
#SSLCertificateFile "c:/Apache24/conf/server-ecc.crt"
```

Specify the absolute path to the downloaded "server.crt" file using the following syntax:

```
SSLCertificateFile "<absolute path to the downloaded server.crt>"
```

4. Locate the following line in the httpd-ssl.conf file, and remove the existing path in the SSLCertificateKeyFile "c:/Apache24/conf/server.key" statement:

```
#   Server Private Key:
#   If the key is not combined with the certificate, use this
#   directive to point at the key file.  Keep in mind that if
#   you've both a RSA and a DSA private key you can configure
#   both in parallel (to also allow the use of DSA ciphers, etc.)
#   ECC keys, when in use, can also be configured in parallel
SSLCertificateKeyFile "c:/Apache24/conf/server.key"
#SSLCertificateKeyFile "c:/Apache24/conf/server-dsa.key"
#SSLCertificateKeyFile "c:/Apache24/conf/server-ecc.key"
```

   Specify the absolute path to the downloaded "server.key" file using the following syntax:

```
SSLCertificateFile "<absolute path to the downloaded server.key>"
```

   Save the file.
5. Locate the following line in the "httpd.conf" file, and remove the '#' character in this line to include the "httpd-ssl.conf" file:

```
#Include conf/extra/httpd-ssl.conf
```

   Save the file.
6. Restart the Apache server.

How to enable ModSecurity in an Existing Apache Setup

If an Apache service is already running on your OneClick server host, perform the following steps to enable ModSecurity:

1. Stop the Apache service.
2. Access the following folders which are required to enable ModSecurity, and copy them to your existing Apache setup:
   $SPECROOT\apache\modsecurity-crs
   $SPECROOT\apache\modules
3. Find the "Dynamic Shared Object (DSO) Support" segment in the "$SPECROOT\apache\conf\httpd.conf" file.
4. Compare the "LoadModule" directives of your existing httpd.conf file with the "LoadModule" directives present in the "$SPECROOT\apache\conf\httpd.conf" file .
   The "LoadModule" directive includes a module that enables a functionality. To enable ModSecurity, you must include ModSecurity modules in your existing "httpd.conf" file.
5. Copy the "LoadModule" directives that are present in the "httpd.conf" of the DX NetOps Spectrum environment, and paste them in your existing "httpd.conf" file.
6. Read the following topics to understand how to enable ModSecurity:
   - How ModSecurity Works.
   - Enable ModSecurity Using the Tomcat port for Apache.
   - Enable ModSecurity Using a Free Port for Apache.
7. Configure the httpd.conf file as explained in these topics, and start the Apache service.

How ModSecurity Blocks Malicious Clients

The "httpd.conf" file includes ModSecurity core rule set configuration files (base rules) that are located at "$SPECROOT\apache\modsecurity-crs\base_rules". The core rule set thresholds and parameters of these base rules are configured

in "modsecurity_crs_10_setup.conf" file that is located at "$SPECROOT\apache\modsecurity-crs\". The "httpd.conf" file includes this core rule set configuration file and all the base rules. These base rules provide a strong firewall capabilities to the Apache server. The following table lists each base rule and its corresponding firewall capabilities:

ModSecurity Base Rule	Firewall Capability
modsecurity_crs_20_protocol_violations.conf	Some protocol violations are common in HTTP attacks. Validating HTTP requests eliminates a large number of application layer attacks. The purpose of this rules file is to enforce HTTP RFC requirements that state how the client is supposed to interact with the server. Identify Invalid URIs.
modsecurity_crs_21_protocol_anomalies.conf	All HTTP web requests include Host, User-Agent and Accept headers. In legitimate HTTP requests these headers exists, but are not empty. This rule checks if these headers exist, and also if they are empty. If the headers are empty, such HTTP requests without common headers are blocked.
modsecurity_crs_23_request_limits.conf	This rule defines limitations on the number of arguments and argument lengths in HTTP requests. For example, an HTTP request with 400 arguments, can be suspicious. You can define the length in this rule. HTTP requests violating this length are blocked.
modsecurity_crs_30_http_policy.conf	This rule set sets limitations on the use of HTTP by clients. Very few requests require the breadth and depth of the HTTP protocol. Many HTTP attacks abuse such valid but rare HTTP use patterns. You can restrict such patterns and usages with this rule.
modsecurity_crs_35_bad_robots.conf	Bad robots detection is based on checking elements easily controlled by the client. As such a determined attack can bypass those checks. Therefore bad robots detection should not be viewed as a security mechanism against targeted attacks but rather as a nuisance reduction. This rule eliminates most of the random attacks against your website. For example, you can prevent a security scanner from scanning your server.
modsecurity_crs_40_generic_attacks.conf	This rule checks against HTTP requests contaning OS Command Injection Attacks. These rules look for attempts to access OS commands such as "curl", "wget", and "cc". These commands are used in injection attacks to force the victim web application to initiate a connection to a hacker site to download, compile, and install malicious tool kits such as those to participate in Botnets.
modsecurity_crs_41_sql_injection_attacks.conf	This rule blocks HTTP requests that contain sql injection attacks.
modsecurity_crs_41_xss_attacks.conf	This rule blocks cross-site scripting attacks coming from unknown and malicious web requests. If these script attacks are not blocked, the malicious scripts can access cookies, session tokens, or other sensitive information retained by the browser.
modsecurity_crs_42_tight_security.conf	This rule detects Path Traversal Attack in the HTTP requests, and blocks such http requests.

modsecurity_crs_45_trojans.conf	This rule detects access to known Trojans already installed on a server. Uploading of Trojans is part of the Anti-Virus rules and uses external Anti Virus program when uploading files. Detection of Trojans access is especially important in a hosting environment, where the actual Trojan upload may be done through valid methods and not through hacking. Trojans detection is based on checking elements controlled by the client.
modsecurity_crs_47_common_exceptions.conf	This rule is used as an exception mechanism to remove common false positives.
modsecurity_crs_49_inbound_blocking.conf	This rule denies access or redirects the malicious requests based on anomaly score settings specified in the 10 config file.
modsecurity_crs_50_outbound.conf	
modsecurity_crs_59_outbound_blocking.conf	This rule checks the overall anomaly score, and the configured action for those threshold violations, and prevents outbound data leakages.
modsecurity_crs_60_correlation.conf	This rule is used in post processing after the response has been sent to the client (in the logging phase). Its purpose is to provide inbound and outbound correlation of events to provide a more intelligent designation as to the outcome, or result of the transaction, that is to confirm whether it was a successful attack, or not.

NOTE

You cannot disable a specific ModSecurity base rule.

ModSecurity Logs

When ModSecurity is enabled, the following types of log files are generated:

Install Log

The "install.log" is created when you first enable ModSecurity using the script. Install log logs the following type of information:

- – Domain name, ServerName, ServerAdmin, and ServerRoot details of the Apache server.
  – The value of the ServerSslPort.
  – The port number with which Apache is installed.
  – The names and locations of all the configuration files which are loaded for the Apache server.

Error Log

The "error.log" file is generated when an error or any malicious attempt is encountered on Apache. All error logs (Apache error logs + ModSecurity Error logs) are generated in this file. It means all Apache error logs, warnings, fatal errors, and the ModSecurity error logs are found in this log file.

Audit Log

The "audit.log" file contains the detailed information about all of the HTTP client intrusions that are detected by ModSecurity. When ModSecurity detects a malicious event, and finds that the event is logged into the error log file, an audit log entry for the same event is logged in this log file. It is the most useful piece of information the system collects, because it contains the actual client request including the client header and data payload about the attack or event.

Debug Log

The "debug.log" file logs all of the ModSecurity errors and exceptions that are useful for debugging.

NOTE

During DX NetOps Spectrum uninstallation on Windows, the uninstaller removes the "apache" folder only when the Apache service is stopped using the "Services.msc" program.

Modify Apache ModSecurity Configuration To Enable OneClick Console Customization

When you configure the Apache ModSecurity firewall, the OneClick Console does not show branding, custom logos, and custom images.

The following configuration helps you to allow Apache ModSecurity firewall to read the custom files and show branding, custom logos, and custom images in OneClick Console. The configuration should be done in the OneClick web server.

Follow these steps:

1. Stop the Apache Service.
2. Open the "httpd.conf" file that is located at "$SPECROOT\apache\conf" (or when "https/ssl" is enabled, open ./extra/httpd-ssl.conf) ./apache/conf/**httpd.conf** ./apache/conf/extra/**httpd-ssl.conf**
3. Modify the **ProxyPass** and **ProxyPassReverse** entries as shown below:

```
..
<VirtualHost _default_:443>
ProxyPreserveHost On
SSLEngine on
SSLProxyEngine on
SSLProxyVerify none
SSLProxyCheckPeerCN off
SSLProxyCheckPeerName off
SSLProxyCheckPeerExpire off

# disable default
#ProxyPass / https://localhost:8080/
#ProxyPassReverse / https://localhost:8080/
# enable local entries
ProxyPass /spectrum https://localhost:8080/spectrum
ProxyPass /spectrum/common https://localhost:8080/spectrum/common
ProxyPass /customimages https://localhost:8080/customimages
ProxyPass /customimages/maps https://localhost:8080/customimages/maps

ProxyPassReverse /spectrum https://localhost:8080/spectrum
ProxyPassReverse /customimages/ https://localhost:8080/customimages
ProxyPassReverse /spectrum/common https://localhost:8080/spectrum/common

 ProxyPassReverse /customimages https://localhost:8080/customimages

 ProxyPassReverse /customimages/maps https://localhost:8080/customimages/maps
```

4. Restart the Apache Server.

Troubleshooting with ModSecurity

'httpd.exe - System Error' message appears when ModSecurity is configured on Windows 2012

Symptom: After Modsecurity is configured on Windows version 2012, there is a 'httpd.exe-System Error' window that appears.

Resolution: 'The program can't start because api-ms-win-crt-runtime-l1-1-0.dll is missing from your computer' system error message appears when ModSecurity is configured on Windows 2012. Try reinstalling the program to fix thisModsecurity should be installed (httpd.exe -k install) and started (httpd.exe -k start) by logging in as an ‘administrator’ only. Refer to the following article and repair the Visual C++ 2015 Redistributable package.

'Forbidden - You don't have permission to access /spectrum/sdn/do/sdnGatewayConfig on this server' message' appears blocking access to DX NetOps Spectrum specific urls.

Symptom: An error message 'Forbidden - You don't have permission to access/spectrum/sdn/do/sdnGatewayConfig on this server' appears when a user is trying to access DX NetOps Spectrum url.

Resolution: Perform the following steps to resolve this issue:

1. Navigate to the **C:\win32app>Spectrum>apache>logs>error** file, which displays the injection attack information blocking links in DX NetOps Spectrum from being accessed.
2. Select the Id which displays the restricted access information and add it to the whitelist configuration file in this format:
3. Navigate to the following location to find the whitelist.conf file: **C:\win32app>Spectrum>apache>modsecurity-crs>activated_rules>whitelist.conf**, which allow for entries with the Id to permit access to DX NetOps Spectrum specific links.

Configure OneClick to Communicate through a Web Proxy Server

If you use a web proxy server that relays HTTP and HTTPS requests (such as the iPlanet and Microsoft proxy servers), OneClick honors the proxy settings used by Java Web Start. OneClick supports both HTTP and HTTPS proxies and also supports proxy authentication. An administrator must configure the OneClick web server to communicate through a proxy server.

All clients connecting through a proxy must configure the proxy settings in the Java Web Start preference console. See the Fresh Install section for more information about the Java Web Start proxy settings. To connect through an HTTP 1.1 proxy, that console setting might be the only required change.

> **NOTE**
> The following changes are not necessary to connect to a proxy that supports HTTP 1.1.

For HTTP 1.0 proxy support, configure the OneClick web server to communicate through a proxy server.

Follow these steps:

1. Open the <$*SPECROOT*>/tomcat/conf/server.xml file for editing using your preferred text editor.
2. Add the following attribute to any active Connector elements:

   ```
   maxKeepAliveRequests="1".
   ```

   Setting this attribute to 1 disables keepalives.
3. Save the changes to the server.xml file.
4. Stop and restart the OneClick web server.

Troubleshoot Proxy Issues

A failed attempt to launch a OneClick client with a proxy results in the normal conditions described in Step 1 and Step 2 and the failure in Step 3:

1. A web browser can access the OneClick web server and load the OneClick home page at *http://<hostname>:<portnumber>*/spectrum/index.jsp (through the proxy).
2. Java Web Start can access the OneClick web server and download the needed OneClick files.
3. The OneClick client *cannot* access the OneClick web server and fails with a “Can't connect to ...” error.

> **NOTE**
> If the procedures in this article do not enable OneClick to communicate through the proxy server in your environment, consider disabling web proxies. For more information, see the Fresh Install section for information.

Troubleshoot Poor OneClick Client Performance

Platform: Windows

Symptom:

The OneClick clients take a long time to start up. Once the clients have started, users experience long wait times for the user interface to react to mouse clicks and navigation tasks. They are so slow that we can hardly use them.

Solution:

This behavior results from the default setting for the "javaws.reuseConnections" Java System property, which is "false". In previous versions, the default value was "true". A change was made to facilitate out-of-the-box connectivity for users with web proxies or load balancers in their environment. Without reusing connections, SSL certificate verification is performed for every request from client to server. This work is expensive, in terms of round-trip times.

Change the value of the "javaws.reuseConnections" Java Runtime System property to "true".

To change the property setting, edit the oneclick.jnlp file.

Follow these steps:

1. Navigate to the following directory:
   ```
   <$SPECROOT>/tomcat/webapps/spectrum/
   ```
2. Open the oneclick.jnlp file for editing using your preferred text editor.
3. Add the following line, immediately below the "<resources>" line:
   ```
   <property name="javaws.reuseConnections" value="true"/>
   ```
4. Restart all open OneClick clients.

Firewalled Environments

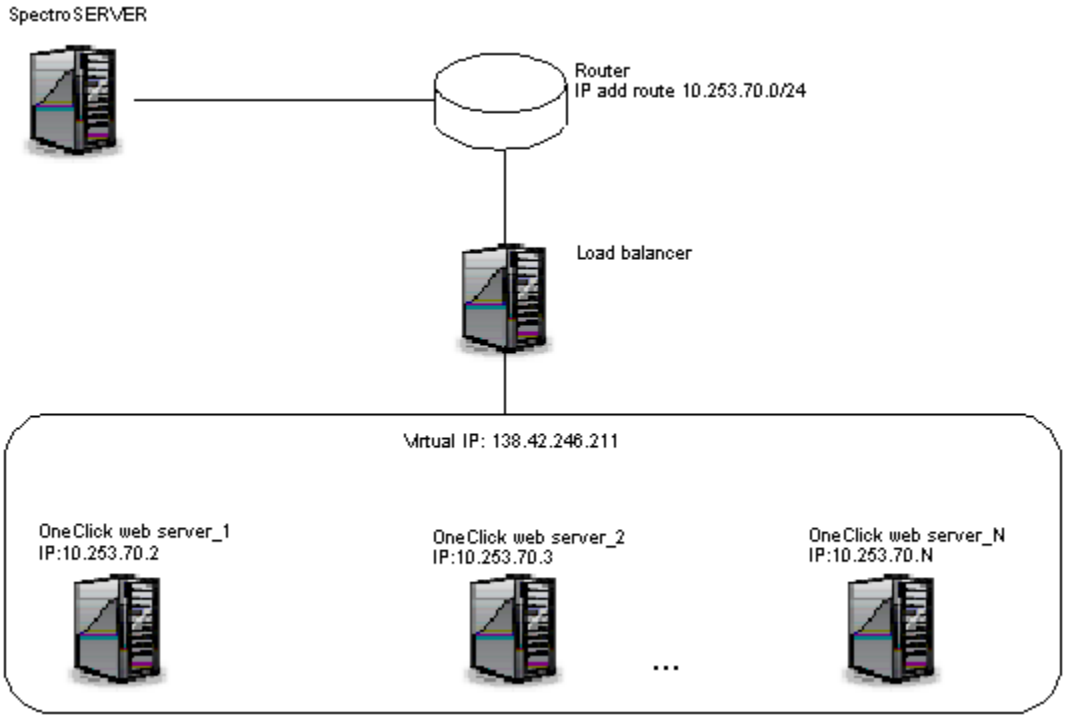
The OneClick web server must communicate with processes on the SpectroSERVER host system to gather data for display to OneClick clients. For the most part, this communication is initiated by the OneClick web server, which establishes connections to specific TCP ports for sending requests and receiving responses. The SpectroSERVER uses bidirectional IIOP (Internet Inter-ORB protocol) to communicate with its CORBA clients. Port 14001 must be open on the firewall to allow the Tomcat web server to receive communications from the SpectroSERVER.

If you use network address translation (NAT) on your network, it is not necessary to perform any additional configuration steps for the OneClick web server to communicate with the SpectroSERVER. However, because SpectroSERVER communication is based on resolving an advertised host name to an IP address, you must configure name resolution on your systems appropriately. Consider a SpectroSERVER host (hostname: “spectrumss”) that is behind a NAT firewall with a private IP address of 192.168.0.2 and a public address of 128.113.0.2. Hosts on the private side of the NAT must resolve “spectrumss” to 192.168.0.2. Hosts on the public side must resolve “spectrumss” to 128.113.0.2.

Load Balancers

To achieve load balancing, identically configured OneClick web servers are accessed through an external load balancing device that employs host/session persistence and any load-balancing mode.

The following figure illustrates a supported load-balancing configuration for multiple OneClick servers.



Check OneClick Web Server Status

You can configure your load balancer to check the status of each OneClick server by using the following HTTP GET statement during periodic server health checks:

```
http://<hostname>:<portnumber>/spectrum/stable
```

A successful GET returns the contents of the "stable" file. The presence of the stable file indicates that the SpectrumTomcat process is in a stable state. Failure to retrieve the file indicates that the SpectrumTomcat process is not running or is unstable.

How to Log the Actual Client IP Address in a Load Balancing Environment

In a load balancing environment, the load balancer performs SNAT (Source Network Address Translation). As a result, you see the IP Address and the host name of the load balancer instead of the actual client in the logs. You can view the logs when you select Client Log in OneClick.

Configure the load balancer to insert the true source IP address into the HTTP request header field "X-Forwarded-For". You can then see the IP Address and hostname of the actual client that logged in to your OneClick console, instead of the load balancer.

More information about load balancer setup is available on the Internet. For example, see Configuring F5 BIG-IP Load Balancer and Configuring Cisco ACE Load Balancer.

OneClick Administration Pages

Access the OneClick Administration Pages

The OneClick Administration Pages are accessible from the OneClick home page. Only OneClick users with OneClick web administration privileges can access these web pages.

To access the OneClick Administration Pages, click **Administration** in the OneClick home page.

About the OneClick Administration Pages

The Administration Pages provide a navigation panel to select specific features to configure and a contents panel on the right that displays the configuration information for each feature. The menu bar contains the following options:

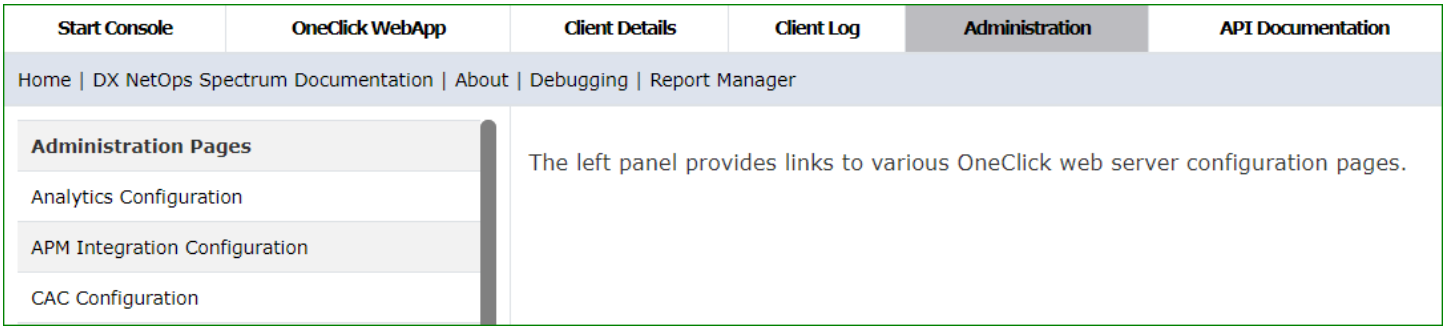
- **Home**
  Opens the OneClick home page.
- **DX NetOps Spectrum Documentation**
  Opens the documentation page.
- **About**
  Opens a window that shows product information like product version, customer support, community link and patent details.
- **Debugging**
  Opens the Debugging page, which contains links to and information about the debugging tools that are included with the product.

  WARNING
  Only use the debugging tools with the help of CA Support.
- **Report Manager**
  Opens the Report Manager Admin Tools pages.

  NOTE
  For more information about administering Report Manager, see Install Report Manager.

The administrative pages that are listed in the navigation panel reflect any OneClick add-on applications that are installed.

The navigation panel and the other OneClick web page links remain available from any OneClick administration page.



Analytics Configuration Page

The Analytics Configuration page lets you configure the integration between DX NetOps Spectrum and Analytics platform/ CA Digital Operational Intelligence.

The Analytics Configuration page contains the following settings:

- **Analytics Server URL**
  Specifies the analytics server URL and port number. This configuration is used to enable launch in context into the Analytics platform from the OneClick page.
  Example**:** http://analyticsserver:port

  NOTE

  Enabling 'HTTPS' is not supported in this release.
- **Analytics Elastic Server URL**
  Specifies the analytics elastic server URL and port number. This configuration is used to fetch the Analytics data to display in the OneClick page for a selected device. For example**:** http://analyticselasticserver:port

  NOTE

  Enabling 'HTTPS' is not supported in this release.

DX NetOps Spectrum Data Publisher

The DX NetOps Spectrum Data Publisher utility allows you to publish inventory and alarms data from DX NetOps Spectrum to Analytics platform/CA Digital Operational Intelligence.

You can download the '*SpectrumDataPublisher.tar*' file and set up the integration between DX NetOps Spectrum and Analytics platform/CA Digital Operational Intelligence.

CAC Configuration Page

Use the CAC Configuration page to configure OneClick to use Common Access Cards (CAC).

NOTE
For more information, see Common Access Card Authentication.

Email Configuration Page

Use the Email Configuration page to configure OneClick to integrate with your existing email system. Operators can then email alarm-related information from OneClick to assigned troubleshooters and other individuals.

The default SMTP Server Host entry for the mail server is "mailhost", which is a common DNS alias for the mail server. If your environment does not use this alias, you can add an entry for “mailhost” to the /etc/hosts file on the OneClick web server.

InsideView Configuration(beta)

Use the InsideView Configuration(beta) section to configure the SpectroSERVER Health View. Spectrum Performance View provides a real-time health dashboard for the distributed DX NetOps Spectrum environment. The administrator can run this tool to get the performance trends; it is disabled by default. SpectroSERVER tracks performance statistics such as Operating System Metrics (for example, CPU, memory, threads, and so on) and other internal metrics such as total models, traps, events, alarms, and so on. These metrics are retrieved every minute and are stored as an event on the SpectroSERVER Performance model.

You can **start** or **stop** InsideView from this page.

NOTE
HTTPS is currently not supported.

Enter the following details and click **Save**.

- **Protocol**
  Specifies the appropriate protocol.
  Default: HTTP
- **Initial Sync Days**

Specifies the interval at which OneClick must sync with Spectruminsideview. The minimum value must be one minute.
Default: 45

- **Poll Timer**
  Specifies the interval between two polls.
  Optimal Interval to avoid Overload: Five minutes
  Default: 5
- **Throttle Size**
  Specifies the throttle size.
  Default: 1000
- **LogLevel: INFO**
  Specifies the DEBUG, INFO, ERROR, WARN.
  Default: ERROR
- **Log File Size**
  Specifies the size of the log file in bytes.
  Default: 10000000
- **LogFileCount: 5**
  Specifies the maximum number of log backup files.
  Default: 5

Reload EvFormat/PCause Configuration

The EvFormat/PCause Configuration page lets you reload modified Event Format or Probable Cause files into the OneClick server.

You can also reload the EvFormat/PCause files from the command line if desired. Use the command line to reload EvFormat/PCause files from any server.

Follow these steps:

1. Obtain and install GNU wget.

   NOTE
   GNU wget is a simple freeware utility.
2. Run the following command:

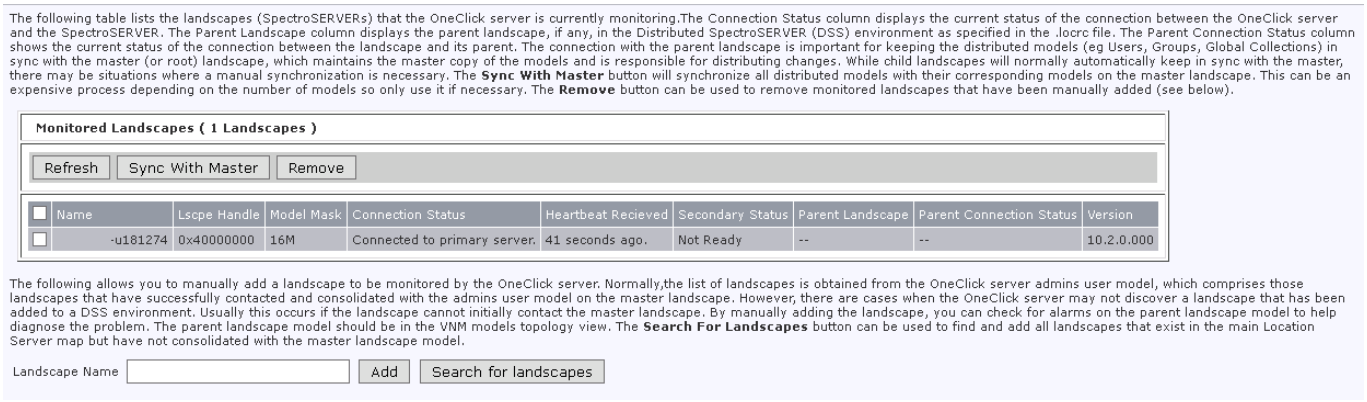
   ```
   wget http://ochost:ocport/spectrum/admin/ecds.jsp?reload=Reload --user username --password password
   ```

   – **ochost**
   Specifies the hostname of the OneClick web server.
   – **ocport**
   Specifies the port number of your OneClick web server.
   – **username**
   Specifies an administrator username for the OneClick web server.
   – **password**
   Specifies an administrator password for the OneClick web server.

Landscapes Page

Use the Landscapes page to view the status for all the landscapes (SpectroSERVERs) that the OneClick server is currently monitoring. You can identify information related to a distributed SpectroSERVER (DSS) setup, including any parent and child landscapes. You can perform a manual synchronization between all distributed models with their corresponding models on the master landscape using the Sync With Master button.

You can manually add or remove landscapes monitored by this OneClick server. You can remove only landscapes that you have manually added.



From 10.2, the value displayed in the Model Mask column is based on the Landscape Handle type you opt for during installation. The value is 16M if you select Huge Landscape handle type, and 1M if you select Legacy Landscape Handle type, during installation.

> **WARNING**
>
> Please ensure that the Model Mask is consistent for all landscapes (SpectroSERVERs) in a Distributed SpectroSERVER (DSS) environment.

LDAP Configuration Page

Use the LDAP Configuration page to configure the OneClick webserver to use an external LDAP server for user authentication.

This administration page includes the following settings and functionality:

- **LDAP Server Settings**
  Server settings to identify a primary and secondary LDAP server by IP address and port number, use SSL, add an SSL certificate, and specify timeout values when attempting to connect to or query the LDAP server.
- **Save LDAP Passwords to DX NetOps Spectrum Database**
  Lets you give access to OneClick users if the LDAP server is down, based on their last known correct LDAP password.
- **User Name Lookup**
  Configure OneClick lookups of usernames, either as a User by Search or a User by Pattern lookup.
- **Test LDAP Configuration**
  Once you have configured the OneClick interface with an external LDAP server, lets you test the configuration.

  > **NOTE**
  > If three consecutive LDAP-based login failures occur, a critical alarm is generated on the VNM that is hosting the Location Server for the OneClick web server.

Primary LDAP Query Authentication Enhancement

10.3.2 now includes a community winning idea to, withdraw from the primary LDAP querying, going forward, for an improved user experience. DX NetOps Spectrum login authentication is now the primary authentication for allowed users who have enabled the LDAP integration. If this initial DX NetOps Spectrum login authentication fails, then the LDAP authentication takes place. DX NetOps Spectrum super users are by default allowed to log in with their DX NetOps Spectrum password, regardless of an LDAP account or the LDAP server availability.

LDAP User Group Authentication

From 10.4.2, you can log in to DX NetOps Spectrum when it is integrated with LDAP even if the user is not present in DX NetOps Spectrum. The user is automatically created in DX NetOps Spectrum during the first login. However, only those

users who are part of the configured LDAP user groups in DX NetOps Spectrum can log in automatically. In DX NetOps Spectrum, the administrator must manually create a user group in all the landscapes with the same group name and required privileges as present in LDAP.

IMPORTANT
Review the following points:

- The user model is created in DX NetOps Spectrum in all the available landscapes in which the user group is present.
- If any landscape is down when the user logs in, then you must manually create the user in the landscape when the landscape is available.
- If the user is removed from the LDAP server, then the user must be manually removed from the DX NetOps Spectrum user group in every landscape.
- If the user is moved from one user group to another in the LDAP server, then you must do it manually in the DX NetOps Spectrum groups. However, login of the user is not affected for the user even if the user is not moved in DX NetOps Spectrum.
- If the user is part of the multiple groups in the LDAP server and matched with the multiple groups configured in DX NetOps Spectrum, then the first matching group is considered for the user authentication. In this case, the order in which the LDAP server returns the user group names is random. Therefore, matching is not always the same.

Follow these steps:

1. Log in to OneClick Console.
2. Create a user group with the same name as present in the LDAP server.
3. Copy the **`ldap-grps-mappings-config.xml`** file from the `$SPECROOT\tomcat\webapps\spectrum\WEB-INF\ldap\config` directory to the `$SPECROOT\tomcat\custom\ldap\config` directory.
4. Edit the **`ldap-grps-mappings-config.xml`** file.
5. Set the property **`LDAPGroups authEnabled`** to **true** as shown in the following example:

   ```
   <LDAPGroups authEnabled="false">
   To
   <LDAPGroups authEnabled="true">
   ```

   NOTE
   If the LDAP groups are configured and the `LDAP groups authEnabled` property is not set to true, the LDAP user cannot be authenticated in DX NetOps Spectrum.
6. Add the group search tag and the search string for each LDAP group.

   ```
   <Group searchTag="memberOf" searchString="CN=group_name,CN=Users,DC=company,DC=local"/>
   ```

   NOTE
   Ensure that DX NetOps Spectrum contains the user group with the same name as in the LDAP server.
7. Save the file.
8. Restart the OneClick server.

MySQL Password Page

OneClick has its own MySQL Server users and passwords: a basic user (OC_user) and an administrative user (OC_admin). Both are used to access the MySQL reporting database on behalf of DX NetOps Spectrum applications such as Report Manager and Service Manager, but the administrative user can also grant privileges to other users for this database. For greater security, DX NetOps Spectrum provides the MySQL Password page which lets you change the passwords of the OneClick MySQL users.

WARNING
Do not attempt to manually change the MySQL user passwords using a MySQL client connection. Storage of the passwords in OneClick depends on MySQL connectivity. As a result, the only safe way to change the passwords is through the OneClick MySQL Password Administration page.

The MySQL Password page contains the following settings:

- **Default User**
  Specifies the credentials for the default OneClick MySQL Server user (OC_user) that DX NetOps Spectrum uses to access the reporting database used in OneClick web applications.
  - **Current Password**
    Specifies the current password for the OC_user MySQL user.
  - **New Password**
    Specifies a new password for the OC_user MySQL user.
  - **Confirm New Password**
    Confirms the new password for the OC_user MySQL user.
- **Admin User**
  Specifies the credentials for the administrative OneClick MySQL Server user (OC_admin) that DX NetOps Spectrum uses to access the reporting database used in OneClick web applications and grant privileges to other users for this database.
  - **Current Password**
    Specifies the current password for the OC_admin MySQL user.
  - **New Password**
    Specifies a new password for the OC_admin MySQL user.
  - **Confirm New Password**
    Confirms the new password for the OC_admin MySQL user.

OneClick maintains the MySQL server user credentials so that it can connect to MySQL. OneClick stores this password in an encrypted form for security purposes.

OneClick Client Configuration Page

OneClick uses the Java Web Start framework developed to launch the OneClick Console from the OneClick home page in a browser. To determine how to launch the OneClick Console (for example, the location of necessary JAR files), the Java Web Start framework relies on several Java Network Launching Protocol (JNLP) configuration files for the launch parameter values.

The OneClick Client Configuration page lets you configure settings associated with the OneClick client. These settings are used by the JNLP files which are located in the directory at the following location:

```
C:\win32app\SPECTRUM\tomcat\webapps\spectrum
```

Any modifications that you make to the JNLP files are saved to custom files in the following directory:

```
$SPECROOT\custom\common\config
```

- **Supported JRE Versions**
  Specifies which versions of JRE can be installed on the client to start OneClick. If none of the specified versions are installed, the user cannot start the OneClick Console from the OneClick home page and, instead, receives an error message. Multiple JRE versions can be specified as needed. When multiple JRE versions are specified, Java Web Start processes precedence from the top of the list to the bottom of the list.
  - **Allow new versions**
    Allows OneClick clients to run any JRE version beyond the specified required JRE, including both major and minor JRE releases. Please be advised, as with any custom JRE configuration, using a JRE version other than the version which is provided to you could result in undesirable application behavior.

    NOTE
    DX NetOps Spectrum supports a documented minimum JRE version level, and supports running the OneClick user interface in that minimum version, or in any later version, unless noted otherwise in the product documentation. We also test with new JRE versions as they become available and update

the product documentation, the online Support knowledge base, or both, if specific JRE versions are incompatible.

- **Java Memory Usage**
  The Java Memory Usage section lets you set the minimum and maximum size of the object heap for the Java Virtual Machine used by OneClick clients. For more information, see Configure OneClick Client Memory Settings.
  - **Minimum Client Memory Usage (megabytes)**
    The minimum amount of memory, in megabytes, that must be available on the client to start OneClick.
    Default: 96
  - **Maximum Client Memory Usage (megabytes)**
    The maximum amount of memory, in megabytes, that OneClick can use on the client.
    Default: 1024
- **OneClick Client Inactivity**
  Lets you configure OneClick to check clients for inactivity and time inactive clients out. This feature is disabled by default. If enabled, when timeout occurs, users can enter their username and password to continue using the OneClick client. Inactivity is determined by the absence of keyboard or mouse activity is detected for the specified amount of time.

Enable Inactive OneClick Client Timeout

You can configure DX NetOps Spectrum to check OneClick clients for inactivity and time out those clients that have been inactive for a specified amount of time. This setting can enhance network security. For example, if a user leaves the OneClick client running unattended on a desktop, it times out.

Follow these steps:

- Click Administration in the OneClick home page.
  The Administration Pages open.
- Click OneClick Client Configuration.
  The OneClick Client Configuration page opens.
- Complete the settings in the OneClick Client Inactivity section as needed:
  - **OneClick Client Inactivity Timeout (minutes)**
    Specifies the number of minutes of inactivity to allow before timing out (logging off) a OneClick client.
    Default: 0 (Disabled)
  - **Applet Inactivity Timeout (minutes)**
    Specifies the number of minutes of inactivity to allow before timing out (logging off) an applet.
    Default: 0 (Disabled)

    NOTE
    If you have OneClick clients that are dedicated to network monitoring and inactivity is likely, you can remove the Inactivity Timeout privilege for a user. The user cannot receive a timeout if the timeout setting has been specified.
- Click Save.
  Timeout of inactive OneClick clients is now enabled.

Performance Center Integration Configuration Page

The Performance Center Integration Configuration page lets you configure event sharing between DX NetOps Spectrum and CA Performance Center.

The Performance Center Integration Configuration page contains the following settings:

- **Event Polling Interval**
  Specifies how frequently DX NetOps Spectrum queries the Performance Center Event Manager component for events. If you modify this value, the new polling interval takes effect at the next polling cycle.

Default: 60 seconds

- **Event Polling**
  Enables or disables event polling.

  NOTE
  For more information, see CA Performance Management and DX NetOps Spectrum.

SDN Gateway Integration Configuration Page

The SDN Gateway Integration page lets you configure CA Virtual Network Assurance to view both virtual network data and traditional physical infrastructure data. The CA VNA collects data from all your SDN environments and delivers that information to DX NetOps Spectrum.

DX NetOps Spectrum and CA VNA integration lets you access SDN inventory via a single point of integration. The integration uses VNA Client API (uses webSocket) to connect to VNA and UDM (Unified Data Model) to access the data.

The SDN Gateway data source contributes the following item types to DX NetOps Spectrum:

- Topology
- SFC View
- Inventory updates (includes SDN entities and their status information)

For more information about SDN, see the CA Virtual Network Assurance documentation.

Service Desk Configuration Page

You can view, configure, test, and save DX NetOps Spectrum and supported Service Desk application integration settings using the Service Desk Configuration administration page.

NOTE
Before configuring OneClick to connect to CA Service Desk Manager and ServiceNow, download and install the integration components on your Service Desk/ MDR server.

You can create and modify CA Service Desk Manager server and admin user parameters, enable and disable the DX NetOps Spectrum and CA Service Desk Manager integration, and add and remove DX NetOps Spectrum alarms that generate CA Service Desk tickets. For more information on integrating DX NetOps Spectrum with supported Service Management applications including CA Service Desk Manager, see How to Install and Configure the Integration.

Single Sign-On Configuration Page

The Single Sign-On Configuration page lets you enable and select a Single Sign-On option for DX NetOps Spectrum. DX NetOps Spectrum supports Single Sign-On using CA Embedded Entitlements Manager (CA EEM) or CA SiteMinder®.

When you save changes to these configuration settings, the OneClick server is automatically restarted in order to apply the changes. If you encounter errors during the restart, see Start and Stop the OneClick Web Server from an Administration Web Page for troubleshooting tips.

DX NetOps Spectrum Configuration Page

You can view and set the following DX NetOps Spectrum configuration parameters:

- Main Location Server Name
- Backup Location Server Name
- Admin User Name
- SpectroSERVER Polling Interval (sec)
- SpectroSERVER Request Timeout (sec)
- The properties of the object request broker (ORB) used by the OneClick web server for CORBA-based communication with the SpectroSERVER
- Use Secure CORBA (SSL) for DX NetOps Spectrum Communication
  The default value is No. Select Yes to enable Secure SSL.
  You will be prompted to restart the OneClick Server. Please restart the OneClick server to enable secure communication. For more information about additional Firewall rule requirements to make this SSL communication work, see the How to Configure CORBA SSL Ports KB Article.

  NOTE
  You can also restart the OneClick server so that any modifications that require a restart can take effect. If you encounter errors during the restart, see Start and Stop the OneClick Web Server from an Administration Web Page for troubleshooting information.

For more information about the implementation of Common Object Request Broker Architecture (CORBA) in DX NetOps Spectrum, see the Development API Reference .

SPM Data Export Page

The SPM Data Export page lets you change the following settings for SPM (Service Performance Manager) Data Export:

- **SPM Data Export Enabled**
  Specifies whether SPM Data Export is enabled or disabled. You must enable it to modify all the other settings.
- **Log File Cycle Time (min)**
  Specifies when (in minutes) the SPM log file is saved and closed, and a new file is opened for logging.
- **Log File Directory**
  Sets the full path to the directory where SPM log files are stored. The directory structure specified must be created prior to saving.
- **Choose Test Result to export**
  Lets you specify whether you want to export the minimum, maximum, or/and average value of the IP SLA tests. Based on the selected option, the corresponding data is exported. You can view the same result in the SPMResult file, which is created in the specified directory. These values are picked up from the SPM result events that are created for these tests. By default, the average value is exported.
- **Landscape Filter**
  Specifies which landscapes SPM data is obtained from.
- **Restart OneClick Server**
  Restarts the OneClick server so that any setting changes that require a server restart can take effect.

SPM Template Naming Page

This page allows you to specify the naming convention for tests created on test hosts that have had a SPM test template applied to them:

- **IP Address**
  The test name consists of the template name and the IP address of the test target, which may be the test host or a particular device.
- **Model Name**
  The test name consists of the template name and the model name of the test target, which may be the test host or a particular device.

NOTE
For information about working with SPM test templates, see Service Performance Manager.

SSL Certificates Page

You can use the SSL Certificates page to view and add SSL certificates used by the OneClick web server.

- **File with Certificate**
  Uploads the new key certificate you want the OneClick webserver to use.
- **Alias Name**
  Specifies a short name for the certificate. This name should be consistent with some of the other commands that may need to be used with it. For example: “ldap” when setting up ldap; “ssl” or “tomcat” when setting up web server SSL.

  NOTE
  For LDAP configuration information, see LDAP Configuration. For OneClick SSL configuration, see Name Resolution Requirements.
- **Overwrite**
  Specifies whether to overwrite an existing certificate with this new certificate. To overwrite an existing certificate, you must load the new certificate with the same alias name of the existing certificate.
  - **Yes**
    Overwrites the existing certificate.
  - **No**
    Does *not* overwrite the existing certificate.
- **List**
  Opens a list of certificates already added to the Keystore. Certificates are listed by alias name.
- **Save**
  Saves your changes and prompts you to restart the OneClick web server.

UIM Configuration

The UIM Configuration page allows you to set up and configure the DX NetOps Spectrum integration with CA UIM. You need to specify details for the following fields to enable the integration.

- **UIM Server Host Name**
  Indicates the IP address/hostname of the UIM Server.
- **UIM Server Port**
  Indicates the server port number of CA UIM.
- **UMP Server Host Name**
  Indicates the IP address/hostname of UMP.
- **UMP Server Port**
  Indicates the server port number of UMP.
- **UMP Server Protocol**
  Indicates the network protocol of the CA UMP server.
- **UIM Group Name**
  Indicates the group name that is available in UIM Server.
- **SpectroSERVER** drop-down list
  Allows you to select the SpectroSERVER where the new UIM Host models are to be created.
- To enable the integration(s), select the required options in the UIM Integrations section:

1. – **VMWare Management** - enables integration of VMWare from CA UIM
   – **Server Management** - enables integration of Servers from CA UIM
   – **DX NetOps Spectrum-UIM Bidirectional Management** - enables the DX NetOps Spectrum-CA UIM bidirectional integration, allowing inventory and alarm synchronization.

Watch Reports

The Watch Reports page lets you generate reports about multiple watches. A *watch* is a mechanism for adding thresholds for model attributes. Watches let you monitor network elements, such as routers, with a high level of detail. They also provide current data that can be used with other DX NetOps Spectrum tools in network analysis.

> **NOTE**
> For more information, see Watches.

Web Server Logs Configuration Page

You can use the Web Server Logs Configuration page to view and set OneClick server log file rotation settings. The OneClick web server log files are located in the following <$SPECROOT>/tomcat/logs directory.
Tomcat Log: C:\win32app\Spectrum\tomcat\logs
Webtomcat Log: C:\win32app\Spectrum\webtomcat\logs

You can set an alarm notification when the log file directory becomes larger than a specified size in megabytes. You can view the current size of the log file directory. You can specify the age in days at which a log file is deleted from the directory.

Purge Tomcat and Webtomcat Log Files

From 10.4.2.1, when the Alarm logs directory exceeds the set threshold, an alarm is displayed with the threshold value. Irrespective of the file size when the threshold value set for `Purge log files older than` is reached, the old log files are deleted.

Web Server Memory Page

You can use the Web Server Memory page to view and set the maximum amount of memory the OneClick server uses. Any changes you make require you to restart the OneClick server. You can also view the percentage of the maximum memory allocation the OneClick server is currently using.

> **NOTE**
> You can also restart the OneClick server so that any modifications that require a restart can take effect. If you encounter errors during the restart, see Start and Stop the OneClick Web Server from an Administration Web Page for troubleshooting information.

Web Server Performance

The **Web Server Performance** page displays the '**Infer Pipes in Global Collections**' option. By default, this option is set to "**Yes**". This option creates pipes between devices that are connected through Wide Area (WA) Links, even when the link is not contained in the Global Collection.

You can disable this option if the GC topology rendering is slow.

Select "**No**" from the drop-down list, and then click the **Save** button.

This will prevent the pipes from being drawn between devices that are connected via WA links when the link is not contained in the Global Collection.

Web Server Performance Graph

The **Web Server Performance Graph** page is designed to view OneClick Server Performance data in a graphical format. This option allows DX NetOps Spectrum Administrators to monitor real-time OneClick Server performance information directly from the OneClick web page.

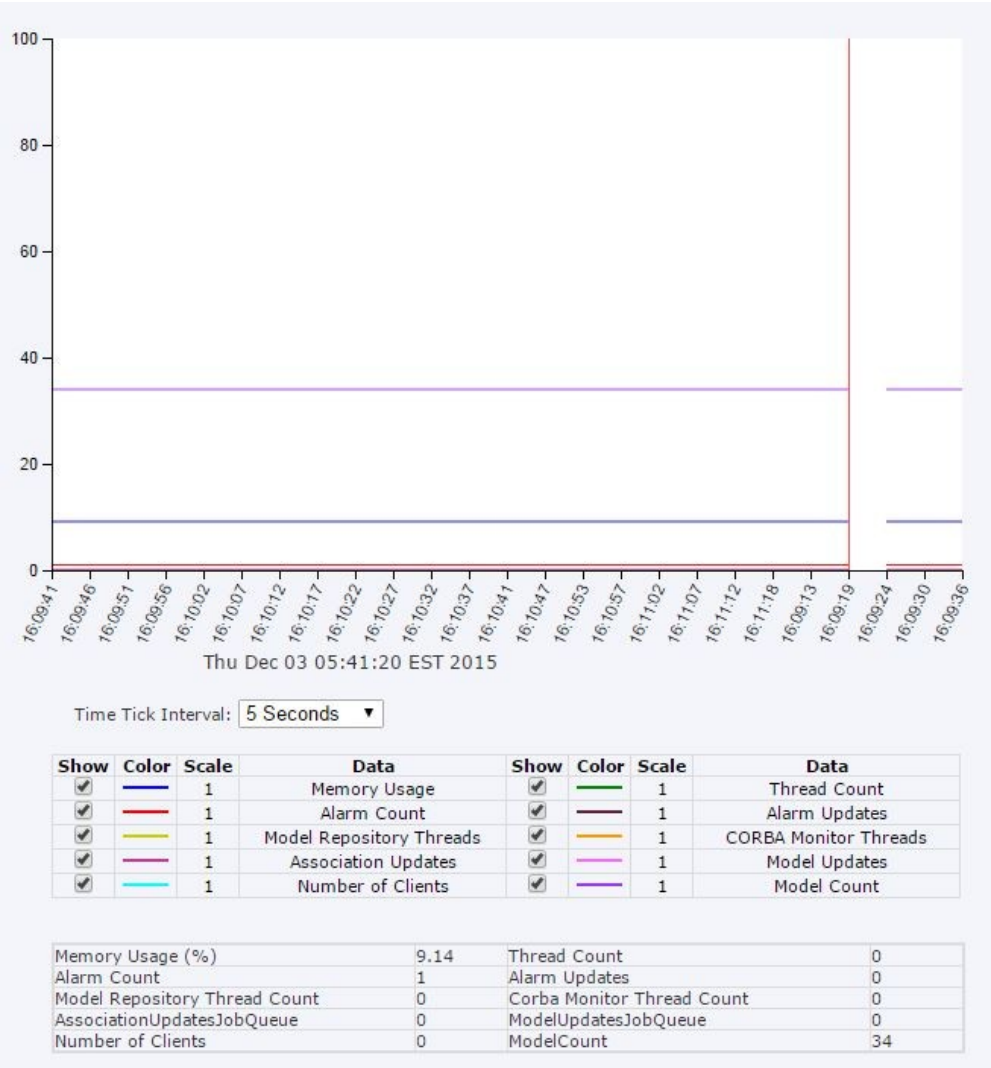
Using this live graph, DX NetOps Spectrum Administrators can diagnose OneClick performance issues. The default refresh time interval for the graph is set to 5 seconds. You can modify the refresh interval time using the 'Time Tick Interval' drop-down list. In this graph, X-axis shows time interval, and Y-axis shows performance data.

NOTE
In 10.1, only real-time monitoring is supported.

Only Administrators can access the Web Server Performance Graph page.

Follow these steps:

1. Log in to OneClick as an Administrator.
2. Click Administration, Debugging.
3. Select the ‘Web Server Performance Graph’ in the left panel.



Metrics:

Following are the available metrics for Web Server Performance Graph. You can select or clear the check box available for each of these metrics to show or hide the performance details of these metrics.

- **Memory Usage (%)** - Percentage of system memory being used by OneClick Server.
- **Thread Count** - Total number of threads spawned by OneClick Server.
- **Alarm Count** - Number of alarms being processed.
- **Alarm Updates** - Number of alarm updates being processed.
- **Model Repository Thread Count** - Number of model repository threads.
- **CORBA Monitor Thread Count** - Number of CORBA monitor threads.
- **Association Updates Job Queue size** - Size of association updates job queue.
- **Model Updates Job Queue size** - Size of model updates job queue.
- **Number of Clients** - Number of OneClick clients.
- **Total Model Count** - Count of total models.

The GIS View

The GIS (Geographical Information System) View displays device models geographically on the Google maps. You can view the model name and model handle details of a device when mouse over on that device in this view. When you click the device in this view, it displays the device details in OneClick. The condition of the device model is also highlighted. For example, a device model with the critical condition is displayed in red color. By default, the map focuses on the USA.

NOTE

The GIS View feature does not work if you integrate DX NetOps Spectrum with CA SiteMinder SSO.

You can configure the default location by entering your favorite location in gisGC.config file. The source of input for GIS View is Global Collection (GC) or list of IP addresses. The default primary source is GC. You can configure the default source to IP address list by modifying the 'UseGisGCconfig=no' in gisGC.config file.

Enter the IP addresses for which you want DX NetOps Spectrum to display GIS view in the following files:

For GC:

```
$SPECROOT/tomcat/webapps/spectrum/console/gisGC.config
```

WARNING
We recommend using a unique name for GC

For IP addresses list:

```
$SPECROOT/tomcat/webapps/spectrum/console/gisIPAddressList.config
```

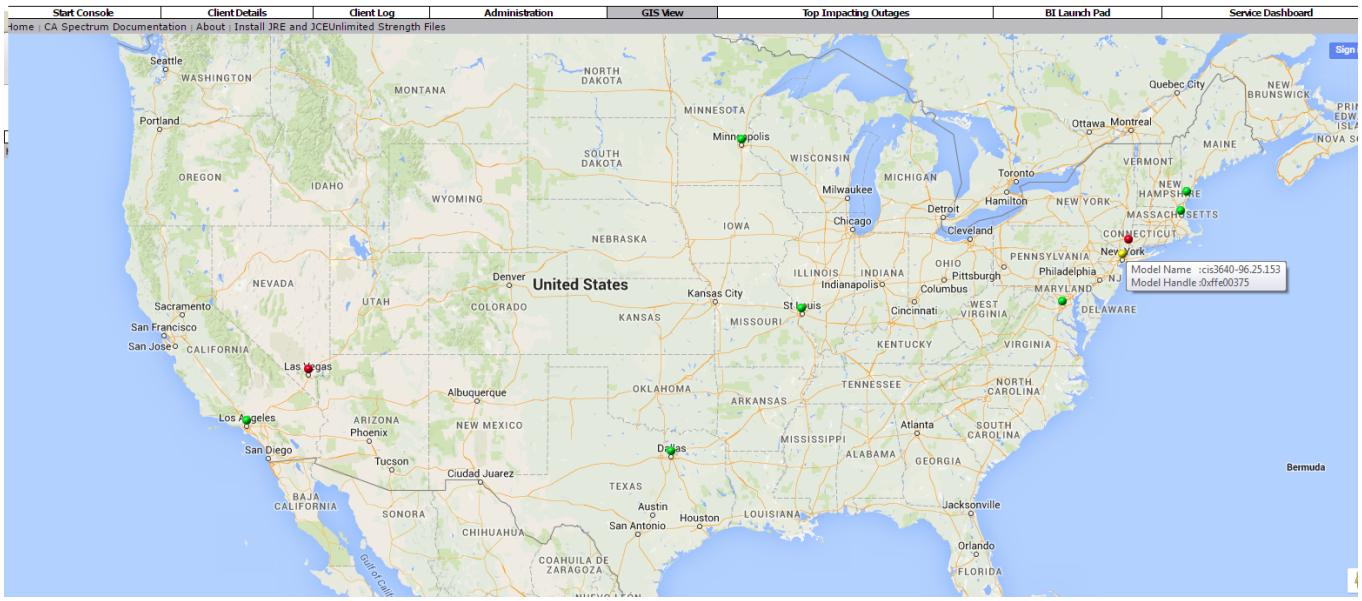
NOTE

- The free version of Google's APIs allows you to view limited number of devices (as per Google API policy) in GIS View.
- Google requires an API key, to enable the Google Maps usage embedded in Websites. You must request for the API key from Google APIs page, and then update the following file with new API key. (applies for both free and enterprise use)

  ```
  $SPECROOT\tomcat\webapps\spectrum\console\googleAPIKEY.config
  ```

  UseLicense=Yes
  API_KEY=<new_api_key>
  Replace the "<new_api_key>" with the API key obtained from Google APIs. Save and close the file.
- We recommend to configure hundred or less devices for GIS View.

The following image shows how the GIS View looks like:



User Administration in OneClick

This section discusses the ways in which user administration is performed in Oneclick.

About OneClick User Administration

User administration involves creating and managing OneClick user accounts. As the OneClick system administrator, you must create a user account for each new user you want to access the system.

As you create new user accounts in OneClick, you can add them within user groups or as stand-alone users. When you have multiple users with similar needs, consider creating user groups to manage their user accounts. When you have users with unique needs, you may want to create user accounts independent of user groups.

Best Practices for Creating and Managing User Accounts

This section describes some best practices for creating and managing user accounts in OneClick.

> **WARNING**
> OneClick includes a default Administrator user with full privileges. This user is the Installation Owner account that you created during SpectroSERVER installation. You cannot delete this user from the Users tab. However, you can delete this account from the Results list that is displayed after a search using the Locater tab. You can also delete this user by removing the landscape of the main location server from the user account. *Do not remove this default Administrator user. Deleting this user produces undesirable results, such as preventing access to OneClick for all other users*.

The benefits of creating and managing individual user accounts include:

- Simplest method
- Best for environments with a small number of users
- Best for users with unique OneClick access requirements
- Individual user accounts can be moved to a user group later if needed

Creating and Managing User Accounts Within a User Group

The benefits of creating and managing user accounts within user groups include:

- Best for environments with a large number of users. Lets you group multiple users by geographic area, function, department, and so on.
- Ability to grant all users within a group the same access and privileges at one time. You can define a minimum set of privileges that all users within the group should have. Then, you can customize the individual privileges of any user in the group.

Consider the following example of creating a user group:

Task:

Within OneClick, you want to grant network operators one set of minimal privileges that enable them to monitor the network. You also want to grant one of these network operators the additional privilege of modeling the network in OneClick.

Solution:

By creating one user group you can easily satisfy this requirement. To configure this requirement in OneClick, you would do the following:

1. Create one user group and place all the user accounts for the network operators in the user group.
2. Grant everyone in the group minimal monitoring privileges.
3. Grant only the one network operator in the group the modeling privilege.

Who Can Perform User Administration?

The OneClick administrator must configure user administration in OneClick. Initially, this configuration must be performed by the user who installed DX NetOps Spectrum (the Installation Owner user). During the installation, DX NetOps Spectrum creates a user account for the Installation Owner. Using this account, this initial user has administrative access to all OneClick features, including user management.

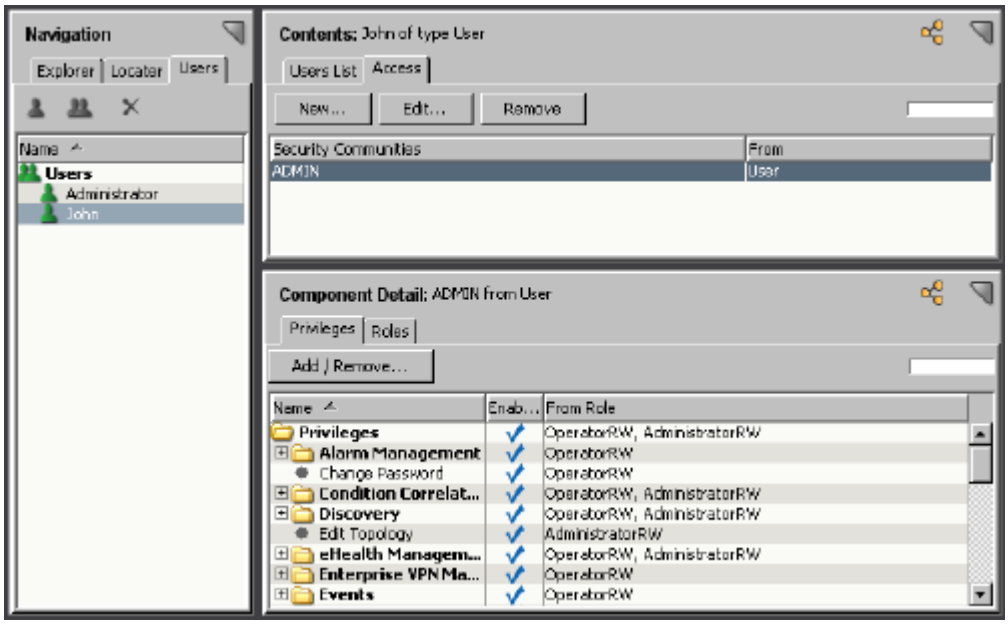
If you are not the initial user but are responsible for user administration, the Installation Owner user must create an administrator account for you. Your account must include an Administrator license and the appropriate user management privileges.

Licenses and Privileges

OneClick includes a set of Administrator licenses and Operator licenses. These licenses determine the privileges that a system administrator can assign to a OneClick user. The privileges that are available with a given license are enabled by default. As the system administrator, you can choose to leave these privileges enabled or you can individually disable them, customizing license privileges.

OneClick User Administration Interface

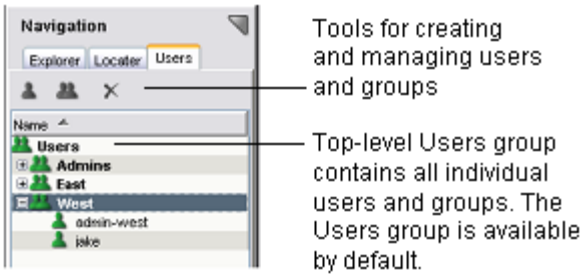
As the OneClick system administrator, you create and manage users within OneClick using options available from the Users tab, as shown in the following image.



Users Tab

The Users tab displays a hierarchical list of users and user groups under the top-level Users group. Initially, after installing DX NetOps Spectrum, the Users tab lists only the top-level Users group and the initial DX NetOps Spectrum user who installed DX NetOps Spectrum (the Installation Owner user) under the Users group.

From the Users tab, you can create and manage user accounts using the tools on the toolbar above the list of users and user groups.



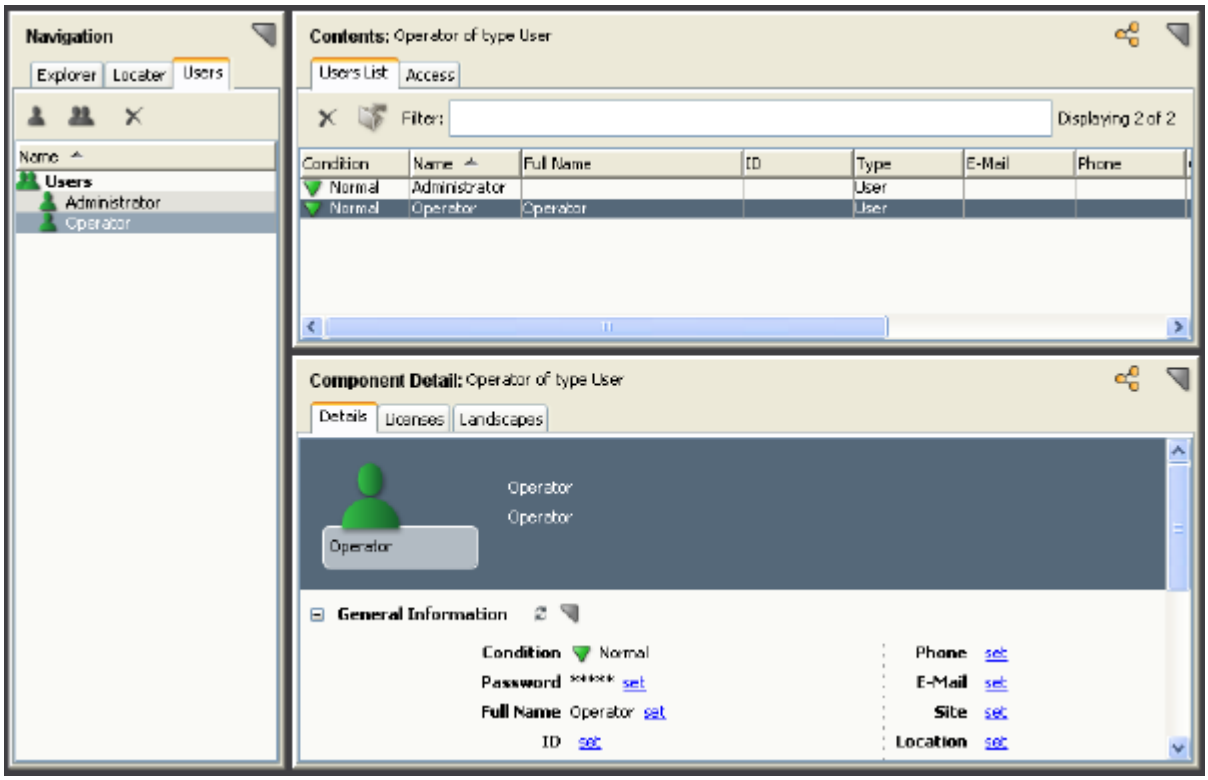
To manage an existing user or user group in OneClick, select it on the Users tab of the Navigation panel. When you select a user or user group on the Users tab, the Users List and Access tabs appear in the Contents panel.

Users List Tab

The Users List tab displays a list of users and user groups for the current landscape along with information about each entry displayed in columns. You can customize this table view by selecting the columns display and by changing the sort order of the table based on the content of a column.

> **NOTE**
> See the Using OneClick section for more information about customizing table views.

When you select the Users List tab, the Details, Licenses, and Landscapes tabs appear in the Component Detail panel, as shown in the following image.



- **Details tab**
  Lets you view information about the selected user or group in the Details tab of the Component Detail panel. The following subviews are available for both users and user groups:
  - **General Information**
    Specifies general information for the selected user or group. Values for certain attributes such as contact information can be set by clicking a ‘set’ link and entering a new value.
  - **Advanced**
    Specifies the following attributes, used for troubleshooting issues with distributed models:
  - **Master Model Handle**
    Specifies the model on the master (or root) landscape that maintains the master copy and is responsible for distributing changes.
  - **Home Model Handle**
    Exists on the landscape that is designated as the home. The master model is typically the home except if the master model was not explicitly created, in which case, a 'hidden' model is implicitly created on the master landscape and one of the other landscapes is designated as the home. The home is not used for distribution but is maintained for legacy purposes.
  - **Duplicate Model Handle List**
    Contains the models from all other landscapes, excluding the home, on which this distributed model exists.
  - **Synchronize Now**
    Synchronizes all the distributed models with the master. In general this will be done automatically but is provided as a button in case it needs to be done manually.

  The following subview is only available when a group is selected:
  - **User Inherited Attributes**
    Specifies the attributes that users will inherit from the selected group.

  The following subview is only available when a user is selected:
  - **LDAP Configuration**
    Specifies whether OneClick users can log in locally if they are not present in the LDAP directory.

NOTE
Super users with passwords set in OneClick can log in locally regardless of this setting.

- **Licenses tab**
  Lets you view and edit licenses for the selected user or group.
- **Landscapes tab**
  Lets you view and edit landscape membership for the selected user or group.

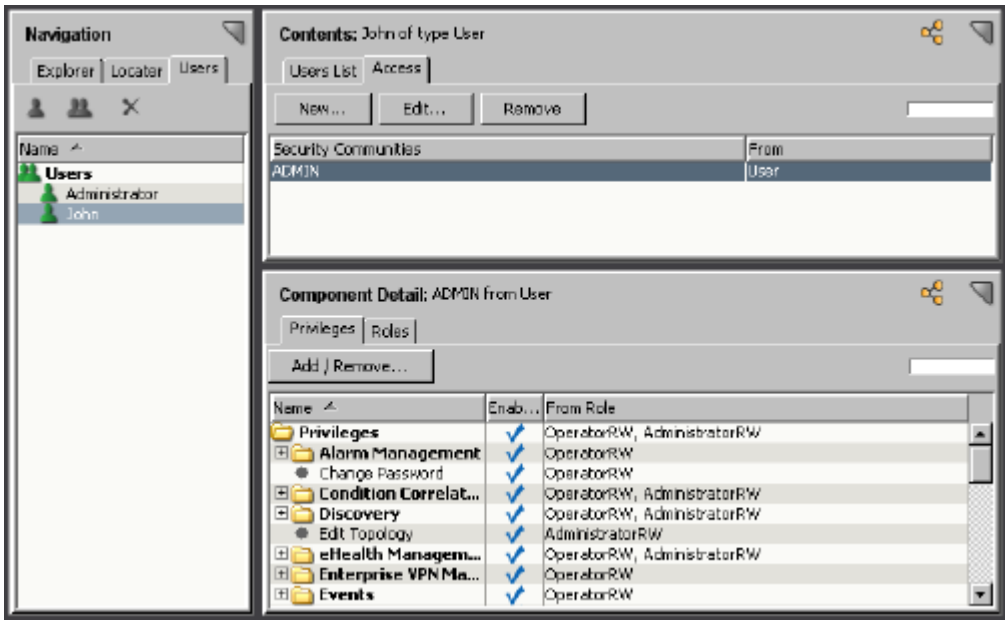
From the Users List tab, you can also do the following:

- Delete the selected user or group.
- Export the Users List tab.

Access Tab

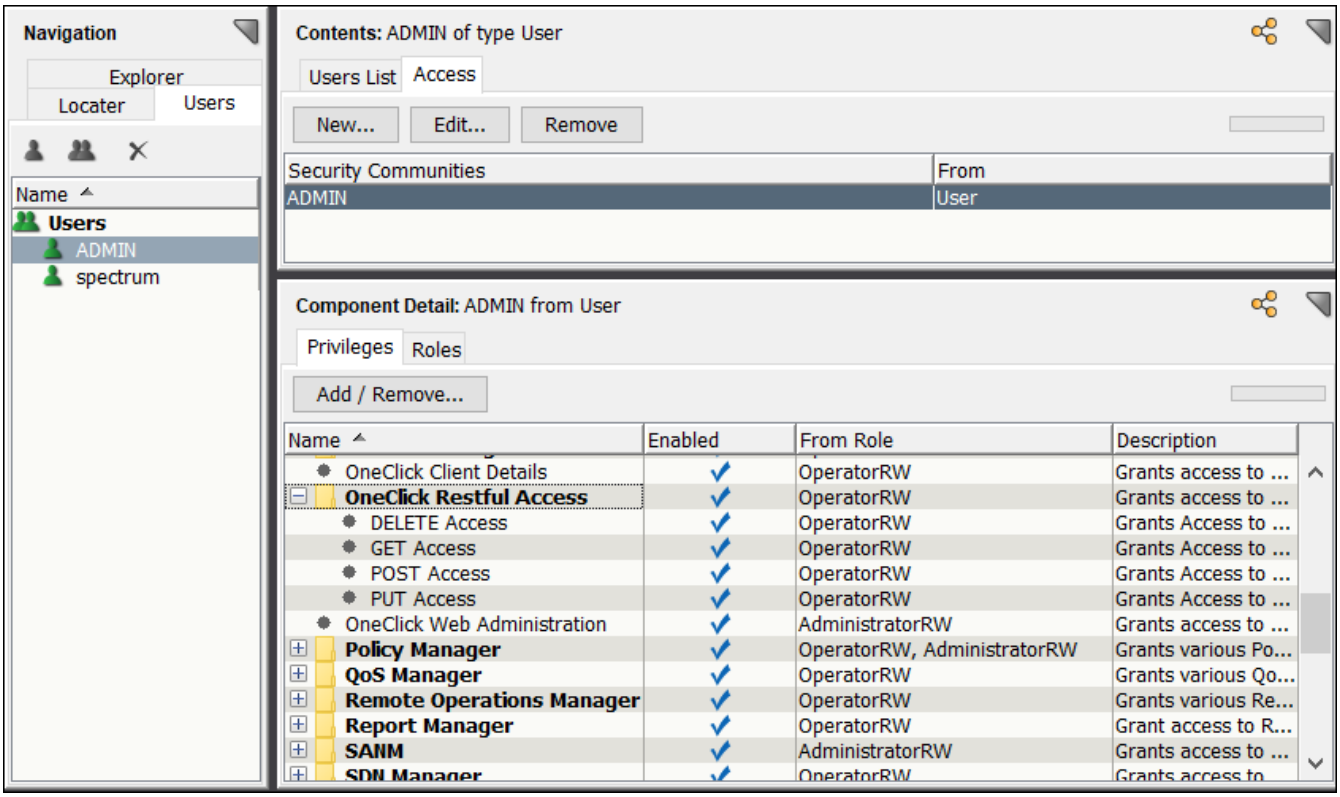
The Access tab in the Contents panel displays the list of security communities assigned to the selected user or user group. Security communities are a tool you can use to limit user access to specific sets of models and views in OneClick. The source of each assigned security community is displayed either from an individual user or a user group.

When you select the Access tab, the Privileges and Roles tabs appear in the Component Detail panel, as shown in the following image.



View and Change Privileges

Access groups appear on the Access tab for a selected user as shown in the following image.



With an access group selected for a user, the available privileges appear in the Privileges tab of the Component Detail panel.

The example Component Detail label shows that these privileges are associated with the selected ADMIN access group at the user level (ADMIN from User). A checkmark is displayed in the Enabled column for each privilege granted to this user. The From Role column shows that all of these enabled privileges are being granted by the OperatorRW role.

Clicking Add/Remove lets you enable and disable privileges for this user at the selected ADMIN access group. You cannot use Add/Remove at the user level to manage privileges for an access group that is inherited from a user group.

Effects of Customizing Privileges

In OneClick you can customize the privileges assigned to an individual user account and/or a user group. When you edit privileges for an individual user account, the changes only affect that user. When you edit the privileges assigned to a user group, the change affects all users within that user group. Users within a group automatically inherit privileges from the group but also retain all assigned individual-level privileges.

You can add privileges to an access group at the user level for users that are members of a group. You can also edit privileges for an access group at the user group level. You cannot remove privileges at the user level that are granted to the user by a user group.

Effects of Removing Privileges Granted by a Role for a User

When you customize the privileges granted by a role for a user, the user is removed from the role. The user retains any privileges granted by the role that is not removed. The role itself does not change. If the privilege is added back for the user, the user does not regain membership in the role.

Customizing the privileges granted by one of the default DX NetOps Spectrum roles (such as AdministratorRW or OperatorRW) for a user -- which also results in the removal of the role from the user and the direct assignment of the remaining, enabled privileges -- has additional consequences:

- If you later create a custom privilege for the default role, which is an assignment specified in the XML file that defines the privilege, the privilege is not automatically granted to the user.
- If you later upgrade DX NetOps Spectrum, any new privileges available in the newer version of DX NetOps Spectrum that are associated with the default role are not automatically granted to the user.

In either situation, to grant the custom or new privileges to the user, you must either explicitly add them to the user or reassign the default role to the user.

Example: For a new privilege role called "SPECTRA", restrict and allow access to some global collections.

Solution: First, create a community name called "SPECTRA". After creating it, select the "global collections" and other required privileges during the privileges selection. And, do the following:

- For each global collections that SPECTRA should see, set the Security String: ADMIN|SPECTRA
- For other global collections that SPECTRA should not see, set the Security String: ADMIN

Manage Users Within User Groups

An excellent way to manage multiple users in OneClick is with user groups. After you have created a user group, you can configure it to provide a minimum set of user privileges for all users within that user group. Each user account you place within this group automatically inherits the group-level privileges.

Inheritance Details for Users in User Groups

Users within a user group inherit the following values from the user group:

- Security community
- Legacy SNMP community string
- Access groups
- Privilege roles
- Attributes specified by the administrator

The following special considerations apply to users contained within user groups:

- Any changes made at the group level are automatically inherited by users within the group.
- Membership in DX NetOps Spectrum landscapes is not inherited from the user group. This must be set at the individual user level.

Specify Inherited Attributes

You can specify the attributes that you want users to inherit from the groups to which they belong.

Follow these steps:

1. Click the Users tab in the Navigation panel.
   The Users List opens in the Contents panel.
2. Select the group containing the users whose inherited attributes you want to specify.
   The Details tab displays the User Inherited Attributes subview.
3. Expand the User Inherited Attributes subview.
   The list of attributes currently applied to users in this group is displayed.
4. Click Edit the Common Attributes
   icon 
   The Common Attributes Editor opens.
5. Double-click the attributes in the Available Attributes list that you want users to inherit.
   The attributes are moved to the Selected Attributes list.

6. Click Save.
   The Common Attributes Editor closes and the selected attributes appear in the User Inherited Attributes list; users in this group will now inherit the values of these attributes.

Create User Accounts and User Groups

When you create a new user or user group, OneClick assigns an Operator license and the OperatorRW privilege role to the new user or user group by default. When you create a new user or group you can choose to assign an Administrator license in addition to the Operator license. When you assign an Administrator license to users or groups, the users automatically inherit all of the privileges associated with both the OperatorRW and the AdministratorRW privilege roles.

To start administering user accounts in OneClick, create users with OneClick default settings. When you create a user or a user group, OneClick assigns an Operator license and the ADMIN security community by default. You can selectively replace the ADMIN security community by modifying users to give them access only to the devices and containers that they manage.

By default, no security is applied to models. To restrict access to a model, add a security string to that model. You can create administrators by adding the ADMIN security string to a universe model and verifying that the appropriate users have access to the ADMIN security community.

Create a new user account or user group using the default privileges that the operator or administrator license provides.

Follow these steps:

1. In the Users tab of the Navigation panel, take *one* of the following steps:
   - **Create a stand-alone user.** Select the top-level Users node and click the Create New User button.
     The Create User dialog opens.
   - **Create a user group.** Select the top-level Users node and click the Create New User Group button.
     The Create Group dialog opens.
   - **Create a user within a group.** Select an existing user group in which you want to create a user and click the Create New User button.
     The Create User dialog opens.
2. Specify the appropriate user information for the user or user group.
   - **Name**
     Specifies the user name for the new user or group. For OneClick users that are present in the configured LDAP directory, this name must match the LDAP user logon name of the user.
   - **Full Name (Create User only)**
     Specifies the full name of the user.
   - **Web Password (Create User only)**
     Specifies a web password for this user. This password is used by OneClick to authenticate this user. For OneClick users that are present in the configured LDAP directory, this password is not used.
   - **Confirm Web Password (Create User only)**
     Confirms the web password you entered when you enter it again in this field.
3. In the Licenses tab, select the licenses that you want to assign to this user or group in the appropriate Member Of check box. By default, new users receive an Operator license and the OperatorRW privilege role.
4. Click the Landscapes tab to configure landscapes for this user or group.

   NOTE
   By default, all available landscapes are selected. In a distributed environment, you can choose additional landscapes in which you want this user to be present. At least one landscape must be selected.
5. Click the Access tab to edit the default model security setting for this user or group.

At least one security community, such as the default ADMIN community, must be specified here. By default, the user or user group receives the read/write ADMIN access group, which gives them access to all models.

6. (Optional) Create additional access groups for the user.

   NOTE
   Models have blank security strings by default. We recommend adding security strings to individual models or containers and using the corresponding security communities to selectively grant user access to models.

7. Click OK in the Create User or Create Group dialog.
   The new user or group is created and displayed in the Users tab of the Navigation panel.

About Creating, Editing, and Assigning Roles and Privileges

You can individually disable and enable privileges for a user or user group. You can also use roles to grant a set of privileges to a user or user group. You can use the default privilege roles in OneClick, or you can create your own custom privilege roles; however, you cannot edit the default privilege roles themselves. After users are assigned a license category, they can have access privileges provided by the predefined roles.

There are six default roles:

- **AdministratorRW**
  (Read/write) Grants privileges required to set up DX NetOps Spectrum and its users, as well as perform all network management tasks. This is the least restrictive role. Some examples include the ability to perform device discovery, model management, topology configuration, eHealth integration management, device certification, and user configuration.
- **AdministratorRO**
  (Read-only) Grants privileges required to access DX NetOps Spectrum modeling and attribute information. Some examples include the ability to view SNMP community strings and SNMPv3 security profiles.
- **OperatorRW**
  (Read/write) Grants privileges required to perform most typical tasks for network management using DX NetOps Spectrum. Some examples include alarm management tasks, Service Performance Manager tasks, and most Network Configuration Manager tasks.
- **OperatorRO**
  (Read-only) Grants privileges that allow the user to monitor network activity and perform limited network management tasks. Some examples include the ability to snooze alarms and to view topology information.
- **Service ManagerRW**
  (Read/write) Grants privileges that allow access to the Service dashboard, as well as the ability to edit Service Outages.
- **Service ManagerRO**
  (Read-only) Grants privileges that allow access to the Service dashboard.

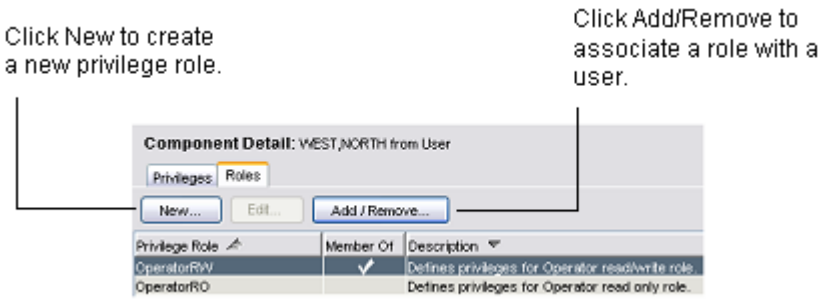
If these predefined roles do not meet your requirements, you can create custom roles. Although you cannot modify the predefined roles, you can modify individual privileges.

Note: When you upgrade to a newer version of DX NetOps Spectrum, any new privileges available in the newer version are automatically added to the appropriate default roles. However, you will need to explicitly add them to any custom roles you may have created, as applicable.

When you edit privileges for an individual user, the changes only affect that user. When you edit the privileges granted by a user group, the changes affect all of the users within that user group. Users within a user group inherit privileges from the group level.

To edit privileges and roles, you modify settings in the Privileges tab and/or the Roles tab for a selected user, as shown in the following image.

In addition to editing individual privileges, you can also grant multiple privileges at one time by assigning a privilege role using the Roles tab, as shown in the following figure.



The default roles included with OneClick and the custom roles that you create are reusable and can be assigned to one or more users. The OperatorRW privilege role automatically grants the privileges provided with the Operator license.

Create and Assign Roles to Users or User Groups

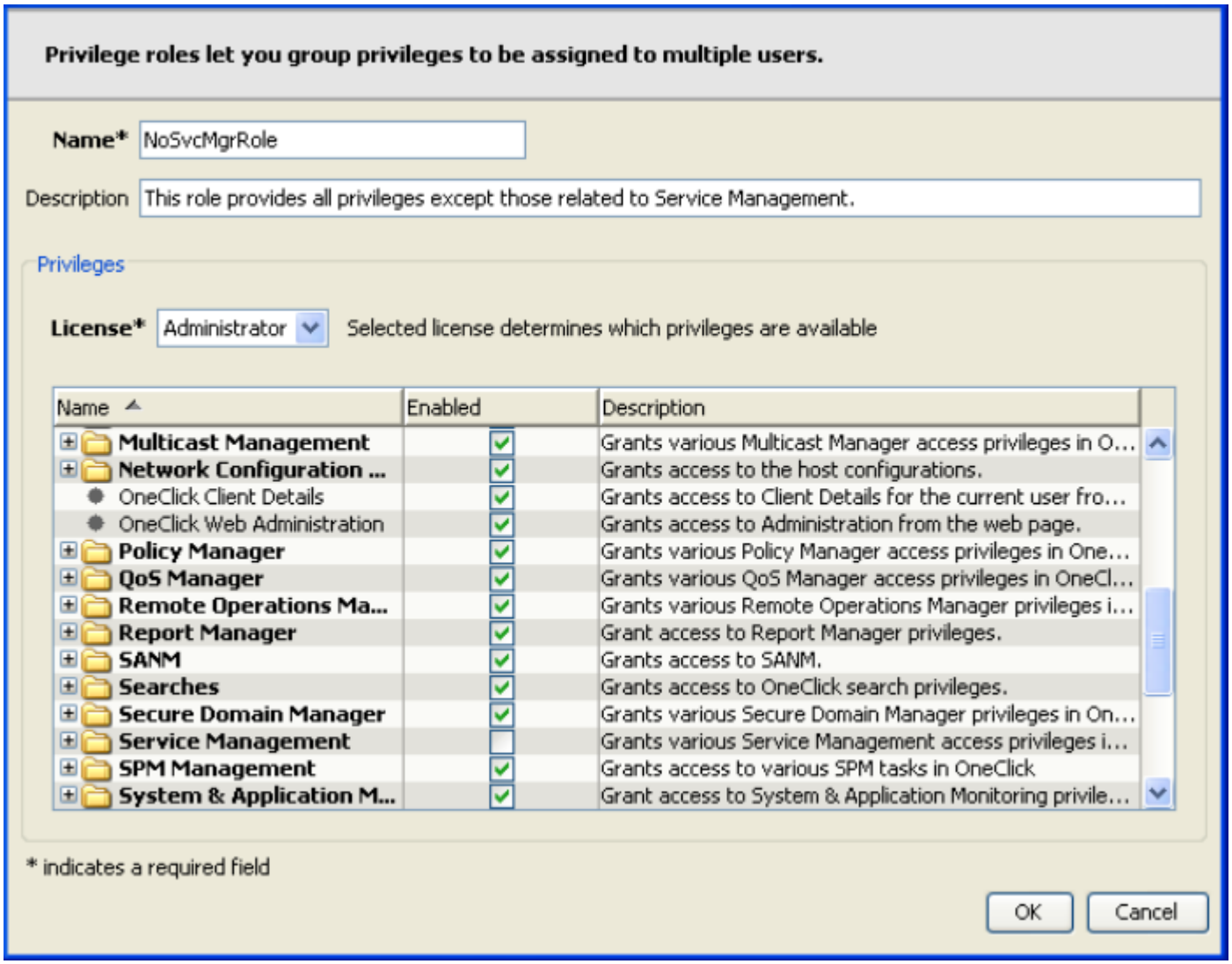
You can create a custom privilege role and then associate it with a user or group. The role has no effect until it is associated with a user account or user group.

You can create a custom privilege role.

Follow these steps:

1. Select a user in the Users tab of the Navigation panel.

   NOTE
   To create an Administrator-licensed privilege role, select a user with the Administrator license. To create a privilege role based on the Operator license, select a user with the Operator license.
2. Click the Access tab in the Contents panel.
   The Privileges and Roles tabs appear in the Component Detail panel.
3. Click the Roles tab, and click New.
   The Add Privilege Role dialog opens.



1. Type a descriptive name for the new role in the Name field.
2. (Optional) Type a full description of this role in the Description field.
3. Select the appropriate license from the License drop-down list.

   NOTE
   The license chosen here determines the privileges that can be enabled with this role.
4. Select the privileges you want this role to grant by selecting or clearing the Enabled check boxes.
5. Click OK.
   The new role appears as an option in the Roles tab of the Component Detail panel. This role is now ready to be used with any user or user group that has the appropriate license.

You can also assign a privilege role. Assigning a privilege role lets you assign an existing role to a user.

Follow these steps:

1. Select the user you want to apply the role to in the Users tab of the Navigation panel.
2. Click the Access tab and select an access group.
   The Privileges and Roles tabs appear in the Component Detail panel.
3. Click the Roles tab and click the Add/Remove button.

The Assign Roles dialog opens.

NOTE
For users in a group, this step must be done at the group level. Assigning a role at the group level affects all users in the group.

4. Move the role you want to assign to the Exists in/Create in column using the arrow buttons.
5. Click OK.
   The role is automatically assigned to the access group selected in Step 2.

Create a Super User

As the OneClick administrator, you can easily grant all possible privileges and access to a user. A *super user* in DX NetOps Spectrum has all available DX NetOps Spectrum license roles, privileges, and access in OneClick. Because access groups and privilege roles do not apply to super users, the Access tab is disabled when a user designated as super user is selected in OneClick.

When you install DX NetOps Spectrum, the initial DX NetOps Spectrum user that is created is a super user. This initial user (also referred to as the Installation Owner user) remains a super user and must always exist in DX NetOps Spectrum. The existence of this account is verified each time the SpectroSERVER starts. The value for the initial_user_model_name setting in the *$SPECROOT*/SS/.vnmrc file stores the setting for the initial DX NetOps Spectrum super user. The default password for the initial user is ‘spectrum’.

NOTE
Consider creating an administrator user with user management privileges to manage users. This user is in addition to the user that installed OneClick (the initial user) and can even manage the initial user account. To ensure that a OneClick administrator has all possible privileges, set the value of 'Is Super User' for that administrator (user) to *true*.

Follow these steps:

1. Select a user from the Users List in the Contents panel.
   The Details tab displays information about the user account.
2. Click set in the 'Is Super User' field, and select Yes from the list.
3. Press Enter.
   The user account is now a super user.

Manage User Access with LDAP Configuration

For environments where LDAP is used for authentication, you can allow or restrict local logins from OneClick users who are not present in the LDAP directory. For example, non-LDAP users, such as non-employees who provide support, training, or troubleshooting with no access to LDAP, require log-in access to OneClick.

NOTE
Super users with passwords set in OneClick can log in locally, regardless of this setting.

Follow these steps:

1. Select the user or user group to edit in the Users tab of the Navigation panel.
2. Navigate to the Details tab of the Component Detail panel for that user or user group.
3. Expand the LDAP Configuration subview.
4. Set the option to 'Allow User to Log In if either the LDAP Password is Invalid or the User does not exist in LDAP' to Yes.

NOTE
For security reasons, we recommend saving the LDAP user password to the DX NetOps Spectrum database. If the option to 'Allow User to Log In if either the LDAP Password is Invalid or the User

does not exist in LDAP' is enabled, you can use the LDAP password for user authentication against the DX NetOps Spectrum database.

Non-LDAP users can log in to OneClick even when they are not present in the designated LDAP directory. Setting this option to No prevents the user from logging in without an LDAP account.

WARNING
If LDAP is configured to search for User by Pattern and no match is found during lookup, your attempt to log in fails. In such cases, verify that LDAP is configured to authenticate User by Search.

Change Details Displayed for a User or User Group

You can modify user or group attributes from the Component Detail panel.

Follow these steps:

1. Select the user or user group to edit in the Users tab of the Navigation panel.
2. Navigate to the Details tab of the Component Detail panel for that user or group.
3. Use the 'set' link to edit attributes such as the password and security string of an existing user or group.

Change the Licenses of a User or Group

The default settings for a new user account include an Operator license that offers operator privileges. To perform administrative tasks such as user management, discovery, and modeling in OneClick, users must have administrator privileges. The default Operator license does not provide any administrative privileges.

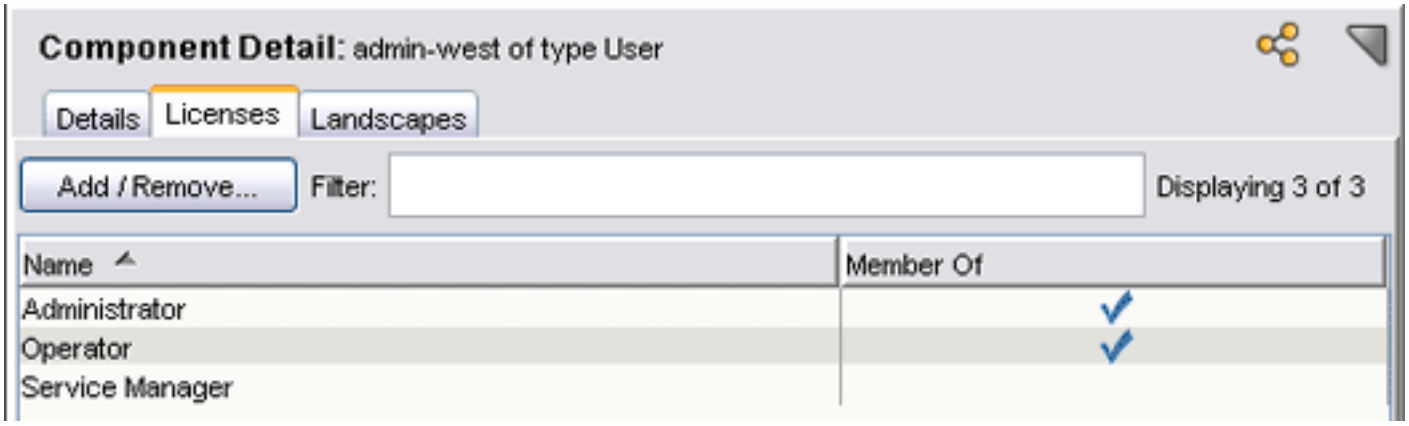
Example: A user group of operators in a Network Operations Center (NOC staff group) with OperatorRW should also see the modeling information of Component detail pane.

Solution: Users in the NOC staff group cannot access the modeling information as this privilege is included in the Administrator license only. To give them access to modeling information, add the Administrator license to this group.

The Administrator license provides the privileges required to perform the following OneClick administrative tasks:

- User Management
- Collection Management
- Discovery
- Topology Editing
- Pipe Management
- Create and Destroy Models
- Search Management

If you are configuring a user account that requires administrator privileges, you must assign the account an Administrator license. You do this by clicking the Add/Remove button in the License tab of the Component Detail panel, shown in the following figure.



When a user logs in, that user consumes assigned licenses from the pool of available licenses. For example, when a user with both Operator and Administrator licenses logs in, one of each license is used.

You can change the licenses that are assigned to a user or to a user group.

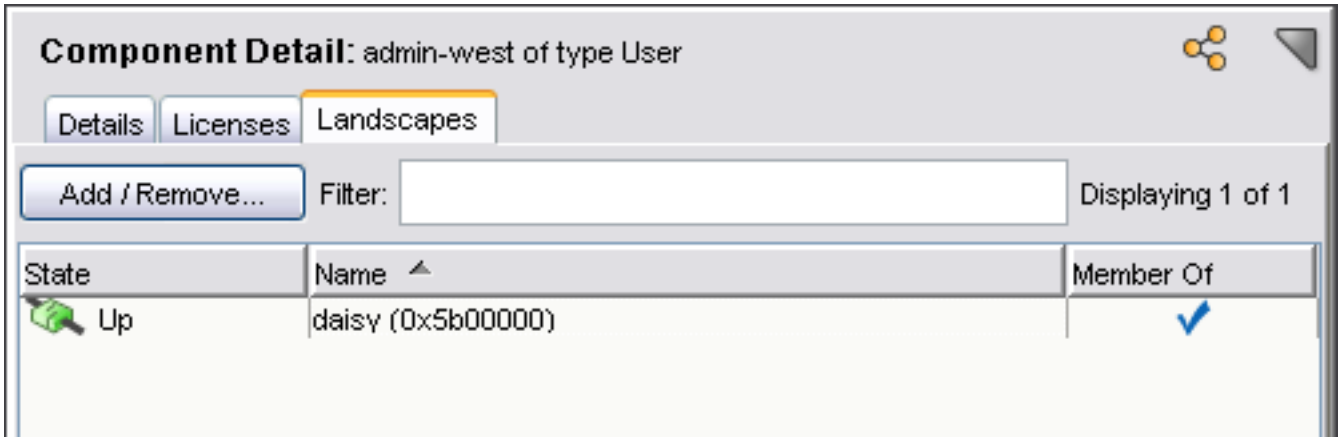
Follow these steps:

1. Select the user or user group in the Users tab of the Navigation panel.
2. Click the Licenses tab of the Component Detail panel.
3. Click the Add/Remove button to select licenses for this user or group.

Change the Landscapes for a User

In a distributed DX NetOps Spectrum environment, you can change the landscape membership of users and groups. Use the Landscapes tab of the Component Detail panel. A distributed environment has multiple SpectroSERVERs, each with its own DX NetOps Spectrum landscape. For a OneClick user to have access to an additional DX NetOps Spectrum landscape, the user must be a member of that landscape.

The following image displays the Landscapes tab for a fictitious admin-west user. This tab displays the state and name of each known DX NetOps Spectrum landscape. The check marks in the Member Of column indicate landscapes in which the user is present.



Tips

- You cannot edit membership in landscapes that are in the “down” state.
- We recommend changing user group landscape membership while the group contains no users. Add users to the group once the empty user group is a member of the desired landscapes.

Follow these steps:

1. Click the Landscapes tab of the Component Detail panel.
2. Click the Add/Remove button.
3. Choose the landscapes where you want this user or group to be present.
4. Click OK.

Change Individual Privileges for a User or User Group

User and user group privileges can be added and removed individually.

Follow these steps:

1. Navigate to the Access tab of the Contents panel for the selected user.
2. Select the access group whose privileges you want to modify.
3. Navigate to the Privileges tab of the Component Detail panel for the selected access group.
4. Click the Add/Remove button.
5. Enable or disable the privileges that you want for this access group by selecting or clearing the Enabled check box.

Locate and Review Role Usage

You can search for user roles, review whether they are in use, and determine the users or user groups that are using them. Reviewing this information is helpful if you are trying to delete a user role. You cannot delete user roles that are in use. However, you can verify whether the user is still valid. You can then remove users and groups that are no longer valid from the role and delete the role itself.

Follow these steps:

1. Click the Locater tab in the Navigation panel.
2. Expand the Roles folder and double-click All Roles.

   NOTE
   Enter landscape information if prompted.

   The search results appear in the Contents panel. The Role In Use column indicates whether the role is currently being used. Roles that are in use have a "Yes" hyperlink.
3. (Optional) Click Yes to review the users or user groups that are using this role.
   The Role in Use dialog displays the users and user groups that are currently using this role.
4. Click Close.
   The Role in Use dialog closes.

Unassign Roles

You can unassign roles from user groups and users as needed.

Follow these steps:

1. Select the user or user group whose role you want to remove in the Users tab of the Navigation panel.
2. Click the Access tab in the Contents panel, and select an access group.
   The Privileges and Roles tabs appear in the Component Detail panel.
3. Click the Roles tab, and click Add/Remove.
   The Assign Roles dialog opens.

   NOTE
   For users in a group, this step must be performed at the group level. Unassigning a role at the group level affects all users in the group.

4. Move the role you want to unassign to the 'Does not exist in/Delete from' column using the arrow buttons.
5. Click OK.
   The role is automatically unassigned from the selected access group.

Delete Unused User Roles

You can delete user roles when they are no longer used by any users or by any user groups.

Follow these steps:

1. Locate the unused user role that you want to delete.
   The search results appear in the Contents panel. Roles that are in use have a "Yes" hyperlink in the "Role in Use" column.
2. Select the unused role to delete.
3. Click Delete.
   The confirm delete dialog opens.
4. Click Yes.
   The user role is deleted.

Move Existing Users to User Groups

You can move existing users to user groups. However, you cannot move the user account under which you are logged in.

Follow these steps:

1. Right-click a user in the Users tab or on the Users List tab.
2. Click Move To Group.
   The Select User Group dialog opens.
3. Select the destination group, and click OK.

Remove Users from User Groups

When you remove a user from a group, the user automatically appears as an individual user. Removing users from groups causes them to lose any privileges that were inherited from the user group level.

Follow these steps:

1. In the Users tab of the Navigation panel, right-click a user.
2. Select Remove From Group.
   The user is removed from the group. This user now appears under the top-level Users group in the Users tab of the Navigation panel.

   NOTE
   After removing a user from a group, verify that the user has the desired access groups and privileges.

Delete Users or User Groups

Users and groups can be deleted from OneClick as necessary. When you delete a user group, any users contained in that group are then organized under the top-level Users node.

WARNING
OneClick includes a default Administrator user with full privileges. This user is the Installation Owner account that you created during SpectroSERVER installation. You cannot delete this user from the Users tab. However, you can delete this account from the Results list that is displayed after a search using the Locater tab. You can also delete this user by removing the landscape of the main location server from the user account. *Do not*

remove this default Administrator user. Deleting this user produces undesirable results, such as preventing access to OneClick for all other users.

To delete a user or user group

1. Select the User or User Group for deletion in the Users tab.
2. Click

(Delete).
The user or group is deleted.

About Using Security Communities to Manage User Access to Models and Devices

Security communities limit user access to specific sets of models and views that use the same security string. Only users with membership in a security community that matches the security string on a model can access the model. You can assign security communities to an individual user or to a user group. All users in a user group inherit the privileges of the security communities assigned to the group.

WARNING
By default, no security is applied to models. Until you apply security to a model, all DX NetOps Spectrum users can see it.

To limit user access to models, create a security community with the desired privileges, and assign it to specific users and user groups. Then selectively apply the security string to the models that those users manage.

Restricted View of Community Names

By default, the Operator Read Only privilege role restricts users from viewing community names. You must enable this privilege for specific Operator Read Only users as needed.

From the Access tab, Privileges tab, you can view, create, edit, and remove security community assignments from a user or user group.

NOTE
You cannot configure security communities for super users. If you select a user who is a super user, the Access tab is disabled.

Use Security Communities to Manage User Access to Models and Devices

You can use security communities to manage user access to data. The Users tab lets you view the currently assigned privileges for community names.

Follow these steps:

1. In the Users tab, select the user or group for which you want to view privileges to a community name.
2. Select the community name in the Security Community list in the Access tab.
3. In the Privileges tab of Component Detail panel, select Model Management, View Attributes, Community Names.

You can add or remove community names from a user’s view.

Follow these steps:

1. In the Users tab, select the user or group for which you want to change community name viewing privileges.
2. In the Security Community list on the Access tab, select the community name that you want to change the user's access to.
3. Click Add/Remove in the Privileges tab in the Component Detail panel.

The Add/Remove Privileges dialog opens.
4. Click Model Management, View Attributes, Community Names.
5. Change the existing privilege by selecting or clearing the Enabled check box.
6. Click OK.
   The change is implemented.

Or you can create a role that adds the community name privilege. You must then assign the new role to a user or group.

Follow these steps:

1. In the Navigation panel, click the Users tab, and select the user or user group to which you want to assign a security community.
2. In the Contents panel, click the Access tab.
3. Click New in the Access tab.
   The New dialog opens.
4. Enter the name of the new security community that you want to create.

   NOTE
   Do not use spaces when naming security communities.
5. Click Add.
6. Enter any additional security communities that you want to share the same privileges.
7. Click OK.
8. (Optional) Click New again to create security communities for the selected user or user group that will not share the same privileges as the security communities you just created.
   The new security communities appear in the Access tab.

Perform the following procedure in conjunction with assigning new security communities to specific models or model types. Security communities do not provide or limit access to data in OneClick until they are assigned. You must also assign privileges or privilege roles to the security communities that you have created.

Follow these steps:

1. In the Navigation panel, click the Users tab, and select the user or user group to which you want to assign a security community.
2. In the Contents panel, click the Access tab.
3. Select the security community you want to edit.
4. Click Edit in the Access tab.
   The Edit dialog opens.
5. Do *one* of the following:
   – To add an entry to the selected security community, enter the name of the new entry in the first field, and click Add.
   – To remove an entry from the selected security community, select the entry from the list, and click Remove.
   – To modify an existing entry for the selected security community, select the entry from the list. Make modifications to the security community entry in the first field, and click Modify.
6. Click OK.
   The modifications to the selected security community appear in the Access list.

   NOTE
   To enable the changes that you made in the preceding procedure, the modified security communities must match the security string attributes that are already applied to models. Or they must match security string attributes that you plan to apply as part of an overall device access and security policy.

You can remove a security community assignment from a user or user group.

Follow these steps:

1. In the Navigation panel, click the Users tab, and select the user or user group whose security community you want to remove.
2. In the Contents panel, click the Access tab, and then select the security community to remove.
3. Click Remove in the Access tab, and then click Yes to confirm your selection.
   The security community is removed from the Access tab for the selected user or user group. This user or user group no longer has access to the relevant models.

Manage Users From the Client Details Page

The OneClick Client Details page lets administrators perform these user management tasks:

- Send messages to logged in clients.
- Manage OneClick licenses by administratively logging off selected users.

> **NOTE**
> This page is not automatically updated with the latest client information. To ensure that you have the latest information, use the Reload function of your web browser to reload the page.

To view the Client Details page:

1. Navigate to http://*<webserver>*/spectrum/index.jsp in a web browser.
   The OneClick home page opens.
2. Click the Client Details link.
   The Client Details page opens, displaying a Client(s) Logged On table.

To send a message to clients using the Client Details page:

1. In the Client(s) Logged On table, select the check boxes next to the user names of the clients to whom you want to send a message, and click Send Message.
   The Enter Message dialog opens.
2. Enter a message and click Send.
   Your message is sent to the clients you selected.

To log clients off using the Client Details page:

1. In the Client(s) Logged On table, select the check boxes next to the user names of the clients that you want to log off, and click Log off Clients.
   A confirmation dialog opens.
2. Click OK.
   The clients are logged off. They receive a message indicating the administrator who logged them off.

> **NOTE**
> Operators can also access the Client Details page, but only to view or log off their own clients.

Manage OneClick Licenses by Limiting Concurrent User Logins

Each time a user launches a OneClick client, that client consumes one instance of each OneClick license assigned to the user. By default, OneClick users can launch unlimited clients with a single set of login credentials. As a result, a single user can consume all of the available OneClick licenses by launching clients repeatedly without exiting from other clients.

The DX NetOps Spectrum administrator can restrict the number of concurrent OneClick licenses that a user or user group can consume at one time. You can distribute the available OneClick licenses among multiple users. In addition, you can set the maximum number of logins to zero to effectively lock out a user without destroying the user account.

You can set maximum login restrictions at both the user and the user group level. The maximum login value is not inherited from the user group. The process that DX NetOps Spectrum uses to manage the different values for users and groups includes verification of the user's maximum login count. If that count has been exceeded, a message appears.

Otherwise, DX NetOps Spectrum then verifies the maximum login count specified for the relevant user group. Therefore, all users in a group can log in multiple times until the maximum values are reached. The total count of all users in the group cannot exceed the group total.

By default, users or user groups can launch as many clients as there are licenses. You can modify the setting for maximum logins in the Details tab of the Component Detail panel for a selected user or user group.

To restrict users to one concurrent login:

1. Navigate to the Details tab of the Component Detail panel for a given user selected on the Users tab.
2. Click set in the Maximum Logins field.
3. Type **1** and click Save.
   This user is now restricted to one concurrent login only.

To restrict users from launching a OneClick client:

1. Navigate to the Details tab of the Component Detail panel for a given user selected on the Users tab.
2. Click set in the Maximum Logins field.
3. Type **0** and click Save.
   This user can no longer launch a OneClick client.

To let users launch an unlimited number of OneClick clients:

1. Navigate to the Details tab of the Component Detail panel for a given user in the Users tab.
2. Click set in the Maximum Logins field.
3. Click Unlimited.

NOTE

The steps mentioned are true while navigating via the explorer tab in OneClick. However, by default in the Topology view, all users can see (but not access) all containers, even if they do not have the required security community access. All containers remain visible by default, so that the ***Neighbors*** tab functions effectively.

To avoid users being able to see other users' containers, place a unique security string in the container that contains the shared containers. The unique security string removes the Topology view of this container, from users who do not have the required security community. Users can still navigate via the explorer view and the topology view outside of the shared containers.

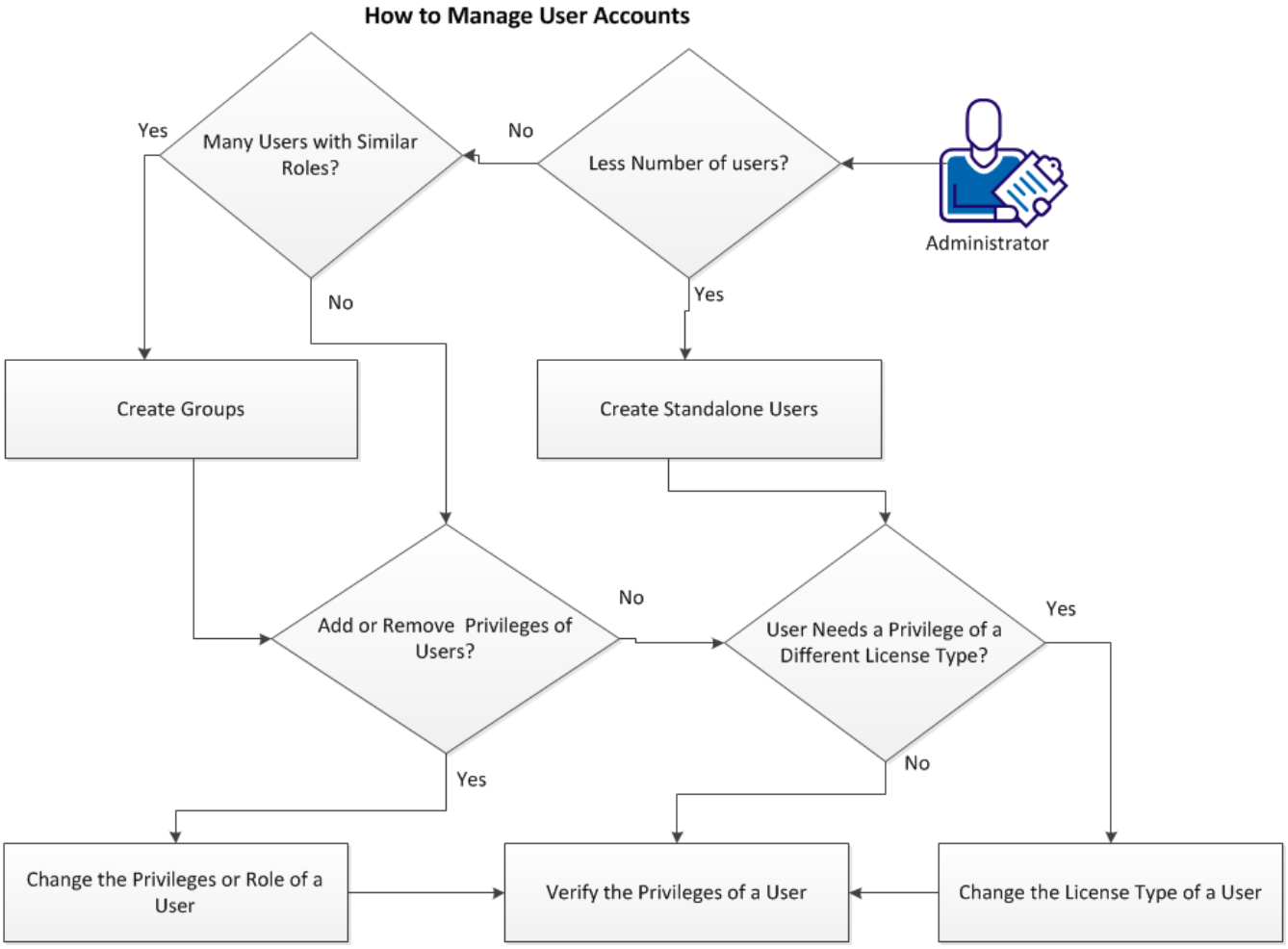
How to Manage User Accounts

This scenario describes how a DX NetOps Spectrum administrator creates and manages user accounts that let different users access DX NetOps Spectrum. Managing user accounts means assigning them the required license, role, and privileges so that each unique product operator has the required level of access to DX NetOps Spectrum.

As a DX NetOps Spectrum administrator you are responsible for managing user accounts of different users. Users log in to OneClick console to access DX NetOps Spectrum. When you install DX NetOps Spectrum with your credentials, you become the installation owner and also the administrator with full privileges. Privileges are components and tools within DX NetOps Spectrum to which a user is granted access based on the license type. Administrator, Operator, and Service Manager are the three types of licenses which you can assign to a new user.

Each license can have Read and Write, or Read only access to DX NetOps Spectrum. These licenses determine which privileges you can assign to a user. The privileges available with a given license are enabled by default. You can keep the privileges enabled or you can customize them. As you create new user accounts in OneClick, you can add them within user groups or as stand-alone users. When you have multiple users with similar needs, consider creating user groups to manage their user accounts. When you have users with unique needs, create user accounts independent of user groups.

The following diagram shows how to manage user accounts of DX NetOps Spectrum users:



Verify the Prerequisites

Before you discover devices in your network, verify the following prerequisites:

- DX NetOps Spectrum is installed on your host.
- You are logged in to the OneClick console as an administrator.
- The OneClick console displays your DX NetOps Spectrum landscape with your host name on the Navigation panel.
- On the Navigation panel, the Users tab exists.
  The Users tab appears when you are logged in as an administrator. You manage user accounts through the Users tab.

Create Standalone Users

When you are not the only user of your DX NetOps Spectrum environment and someone wants access, you create a login user account for that person on the OneClick console. Creating standalone users is best when you have less number of users with unique privileges.

Follow these steps:

1. Log in to the OneClick console.
2. From the Navigation pane, select Users.

3. On the toolbar, Click Creates a new user
   The Create User dialog opens.
4. Enter the Name, Web Password, and confirm the Web Password.
5. Click Licenses, and select Administrator, Operator, or the Service Manager as the license type.
6. Click Landscapes, and select the landscape to which you want the user to belong.
   By default, the user belongs to the administrators landscape.
7. Select Details, and enter the details of the user and click OK.

The new user account is created under Users in the OneClick console.

Create Groups

User groups are best suited for environments with many users. A user group lets you group multiple users by geographic area, role, or department. You can grant all users within a group the same access and privileges at one time. You can define a minimum set of privileges that all users within the group must have. Later, you can customize the individual privileges of any user in the group.

NOTE
Users in a group inherit the license and privileges of the group.

Follow these steps:

1. Log in to the OneClick console.
2. From the navigation pane, select Users.
3. Click Creates a new group.
   The Create Group dialog opens.
4. Enter a Name for the new group.
5. Under Licenses, Select Administrator, Operator, or Service Manager as the license type.
6. Click Landscapes, and select the landscape to which you want the group to belong.
   By default, the group belongs to the administrators landscape.
7. Under Landscapes, select the required landscape, and click OK.

NOTE
By default, your local landscape is selected. In such a case, you can ignore this step.

The new group is created under Users in the OneClick console.

Create Users Within a Group

You create users within that group after creating a group for multiple users. By default, the Users within the group inherit the license and privileges from the group. You can change the privileges and license of any user within the group.

Follow these steps:

1. Log in to the OneClick console.
2. Select Users from the Navigation pane.
3. Right-click the group that you created, and select New User.
   The Create User dialog opens.
4. Enter the Name, Web Password, confirm the Web Password, and click OK.

The new user account is created within the group in the OneClick console.

Change the Privileges or Role of a User

When a stand-alone user or a user within a group wants extra privileges, you can add these privileges to the user. You can also remove some privileges of any user. You can change the Role of a user to Read only, or Read and Write for

any given License type. For example, a user with an operator license can be assigned either the OperatorRW (Read and write) or the OperatorRO (Read Only).

NOTE
By default, for any license type you get the Read and Write Role.

Follow these steps:

1. Log in to the OneClick console.
2. From the Navigation pane, click Users, and select the stand-alone user, or a user within a group by expanding the group.
3. Select Access from the Contents pane.
4. Select the Security community of the user.
5. From the Component Detail pane, select Privileges.
   Under Privileges, you can add or remove a Privilege from the list of privileges. The list also shows the privileges that are enabled for the user.
6. Click Add / Remove.
   The Add / Remove Privileges dialog opens.
7. Select the Privileges that you want to enable for the user, and click OK.
8. To change the Role of the User, Select Roles from the Component Detail pane and click Add / Remove.
   The Assign Role dialog opens.
9. Select the required Role from the right pane and move it in the left pane, and click OK.

The extra Privileges are enabled or disabled for the user. The required role is also assigned to the user.

Change the License Type of a User

Some Privileges like the Discovery are enabled only for the Administrator license. When you want to enable a Privilege that does not belong to the operator license change the license type of the user to Administrator. You can change the license type of a stand-alone user or of a user within a group.

Follow these steps:

1. Log in to the OneClickconsole.
2. Select Users from the Navigation pane.
3. Right-click the stand-alone or a user from a group, and select Component detail.
   The Component Detail dialog opens.
4. Select Licenses, and click Add / Remove.
   The Select Licenses dialog opens.
5. Select the required license from the right pane and move it in the left pane, and click OK.

The required license is granted to the user, and all the privileges belonging the license type are enabled for the user.

Verify the Privileges of a User

You have changed the Privileges of a user. Now, you can verify the overall privileges that are enabled for that user.

Follow these steps:

1. Log in to the OneClick console.
2. Select Users from the Navigation pane.
3. Select a stand-alone user or a user within a group.
4. Select Access from the Contents pane.

NOTE
When you select a user within a group and click Access, select the Security Community of the user from the contents pane.

5. Check the privileges that are enabled for the user under Privileges from the Component Detail pane.

You have verified the Privileges of the user.

Troubleshooting User Administration

Error: "The user xxxxxx already exists" when trying to create a user in DX NetOps Spectrum

Symptom:
When you delete a user and try to recreate the user in DX NetOps Spectrum, you may see the following error:

"SPC-OCC-11916 The user xxxxxx already exists."

However, this user is not seen in the Users tab.

Cause:
The entire user model did not get completely deleted. This is seen most often in a Distributed SpectroSERVER (DSS) environment.

Resolution:
Do the following on each primary SpectroSERVER in the environment:

1. Log into the SpectroSERVER system as the user that owns the DX NetOps Spectrum installation
2. (Windows) start a bash shell by running "bash -login"
3. cd to the $SPECROOT/vnmsh directory
4. Enter the following command to start a Command Line Interface (CLI) session
   ./connect
5. Enter the following command, where <xxxxxx> is the User model deleted, to search for any leftover models related to the User model that was deleted:
   ./show models | grep <xxxxxx>
6. If a User model or AccessGroup model for the <xxxxxx> is found, delete the model using the following CLI command where <MH> is the model handle of the model:
   ./destroy model -n mh=<MH>
7. After checking each SpectroSERVER system using steps 1-6 above, log out of OneClick and then restart tomcat on the OneClick server to clear the OneClick cache.
8. Log into OneClick and recreate the User model

For more information, see Command Line Interface section.

Error: No Privilege for the User to Access OneClick Rest API

Symptom:

During integration of DX NetOps Spectrum with other products, I see an error "No Privilege for the User to Access OneClick Rest API"

Resolution:

As an administrator, you can allow or restrict access to the OneClick RESTful APIs. You can set API type-level access for GET, POST, PUT and DELETE.

Follow these steps:

1. From the OneClick Console, select the **Users** tab in the **Navigation** panel.
2. Select the user to change the access.
3. Select the **Access** tab, in the **Contents** panel.
   By default, the **Security Community** is selected.
4. Select the **Privileges** tab in the **Components Details** panel.
5. Select **Add/Remove**.
6. Expand the **OneClick Restful Access** option.
7. Change the privileges as required and click **OK**.

Configuring Additional OneClick Applications

This chapter discusses DX NetOps Spectrum add-on application administration and configuration in OneClick from the perspective of a OneClick administrator. This includes managing and configuring other DX NetOps Spectrum applications.

Configure Service Performance Manager (SPM) Data Export Parameters in OneClick

By default, SPM Data Export is disabled in OneClick. To configure SPM data export in OneClick, enable SPM data export logging. Set the time period when data is written to each log file. Then create and specify the directory for the output log file. At the end of the interval that you set, the file is saved, and a new file is created for incoming data. By default, when SPM data export is enabled, 60 minutes of data is captured before the SPM log file is saved and a new file is opened.

> **NOTE**
> By default, when SPM data export is enabled, OneClick attempts to save SPM data files to the /tmp directory. You must first create the /tmp directory or create an alternate location for the SPM log files.

Follow these steps:

1. Click Administration in the OneClick home page.
   The Administration Pages open.
2. Click SPM Data Export in the list on the left.
   The SPM Data Export Configuration page opens.
3. For SPM Data Export Enabled, click Yes.
4. For Log File Cycle Time (min), enter the elapsed time in minutes when the current SPM log file will be saved and closed and a new file opened for logging. The default value for this logging interval is 60 minutes.
5. For Log File Directory, enter the fully qualified file path for the directory where OneClick will store SPM log files.

   > **NOTE**
   > Create the directory structure for OneClick to save the data files in. By default OneClick attempts to save the data files in /tmp which you must create first if it does not exist.
6. For Landscape Filter, specify the DX NetOps Spectrum landscapes from which OneClick exports data in a distributed environment. Use the left arrow to move the landscapes from which to export data to the Show Landscapes list. Move any landscapes from which data is not exported to the Hide Landscapes list. By default, all available landscapes are included.
7. Click Save.
   You are prompted to commit your changes and restart the OneClick web server. The OneClick web server must be restarted for the changes to take effect.
8. Click OK.
   Your changes are saved and the OneClick web server is restarted.

Display Topology Tab Contents in a Web Page

You can use a topology applet to make the contents of your Topology tab available from a web page. Specify the container-based model handle whose topology you want to view.

> **NOTE**
> You can determine the model handle of the container to use from the Attributes tab. For more information, see the Modeling and Managing Your IT Infrastructure section.

To display Topology tab contents in a web page, type the following URL into your web browser:

```
http://<hostname>:<portnumber>/spectrum/topology.applet?mh=<model handle>
```

- **<model handle>**
  Specifies the container-based model handle that you want to view the topology of.

The portion of the Topology tab that you specified is now accessible from this web page. From here, you can drill into other containers and return to the starting point.

To display Topology tab contents within an existing web page, take one of the following steps:

- Embed the topology applet into the web page using an iframe:

```
<iframe src="http://<hostname>:<portnumber>/spectrum/topology.applet?mh=<model handle>" width="830"
 height="530"/>
    Your browser does not support embedded objects, <a href="http://<hostname>:<portnumber>/spectrum/
topology.applet?mh=<model handle>">click here</a> to go to included content.
</iframe>
```

  > **NOTE**
  > This method works best for Internet Explorer browsers.

- Embed the topology applet into the web page using the following syntax to avoid using iframes:

```
<div>
<object data="http://<hostname>:<portnumber>/spectrum/topology.applet?mh=<model handle>" type="text/html"
 width="830" height="530">
    Your browser does not support embedded objects, <a href="http://<hostname>:<portnumber>/spectrum/
topology.applet?mh=<model handle>"> click here</a>to go to included content.
</object>
</div>
```

  > **NOTE**
  > This method works best for Firefox browsers.

The portion of the Topology tab that you specified is now accessible in a web portlet. From here, you can drill into other containers and return to the starting point.

Model Security in OneClick

This section describes model security and how to configure it in OneClick.

How Are Models Secured in OneClick

Model security in OneClick lets you control user access to models. Secure a model by setting the security string on that model. By default, users can access all models.

Secure modeled network elements in OneClick using the following process:

1. Apply a security string to a modeled element that you want to secure. For example, set the security string of a LAN container model to lan1. For more information, see Using Security Strings to Secure Modeled Elements.
2. The security string that is set on the model in Step 1 must appear in an entry on the Access tab of a given user account for that user to access that secured model. For more information, see Use Security Communities to Manage User Access to Models and Devices.
3. To prevent a user from accessing secured models, modify the access group in the Access tab for that user. For more information, see Scenarios for Implementing Model Security.

NOTE

The steps mentioned are true while navigating via the explorer tab in OneClick. However, by default in the Topology view, all users can see (but not access) all containers even if they do not have the required security community access. All containers remain visible by default, so that the ***Neighbors*** tab functions effectively.

To avoid users being able to see other users' containers, place a unique security string in the container that contains the shared containers. The unique security string removes the Topology view of this container, from users who do not have the required security community. Users can still navigate via the explorer view and the topology view outside of the shared containers.

Using Security Strings to Secure Modeled Elements

Set the security string on a model to prevent users without a matching entry on their Access tab from accessing the model. By default, the security string is empty.

The procedure that follows provides the basic steps to configure model security. It does the following:

- Secures a model with a security string
- Gives a user access to that secured model
- Prevents unauthorized users from accessing secured models

NOTE
This procedure assumes you have already modeled elements in your OneClick environment. For more information, see the Modeling and Managing Your IT Infrastructure section.

Follow these steps:

1. Select the modeled element, such as a device model, that you want to secure in the Topology tab.
2. Click the Information tab in the Component Detail panel.
3. Expand the DX NetOps Spectrum Modeling Information subview, click set in the Security String field, type a security string, and press Enter.
   This model is now inaccessible for users who lack an Access tab entry with this security string.
4. To give a user access to this secured model:
   a. Select the user on the Users tab of the Navigation panel, and click the Access tab of the Contents panel.
   b. (Optional) Remove any access groups that this user no longer requires by selecting access groups and clicking Remove.

      NOTE
      When you remove an access group from a user account, any privileges that are assigned with that access group are also removed.
   c. Click New in the Access tab of the Contents panel.
      The New access group dialog opens.
   d. Enter the security community from Step 3, and click OK.
5. Verify that this user has adequate privileges. To perform this step, assign the appropriate privileges to the access group that you added:

a. Select the access group.
b. In the Privileges or Roles tab for the selected access group, assign the privileges you want this user to have for this security community. For example, in the case of an operator user, you might assign the OperatorRW privilege role using the Roles tab.

When this user logs on, models that have a security string that matches the access group entry that you added and all unsecured models appear on the Topology tab. This user also sees any container models that contain models that are accessible to this user account.

NOTE

The steps mentioned are true while navigating via the explorer tab in OneClick. However, by default in the Topology view, all users can see (but not access) all containers, even if they do not have the required security community access. All containers remain visible by default, so that the ***Neighbors*** tab functions effectively.

To avoid users being able to see other users' containers, place a unique security string in the container that contains the shared containers. The unique security string removes the Topology view of this container, from users who do not have the required security community. Users can still navigate via the explorer view and the topology view outside of the shared containers.

How to Customize Security String Inheritance

Use the following process to customize security string inheritance:

1. Add relations for security string roll down.
2. Define security string roll down overrides for model types.

Relations for Security String Roll Down

DX NetOps Spectrum will roll security strings down from the left model to the right model, from the following relations:

- Application
- Can_Assign
- CollectsChassis
- Collects
- Contains
- HASPART
- Manages
- Organizes
- Owns
- Provides

NOTE
For specific details about how to use the Model Type Editor to create new model types, see the Model Type Editor section.

To add new relations that security strings will roll down

1. Stop the SpectroSERVER if it is running, and verify that there are no other programs running that can access the SpectroSERVER database.
2. Open the Spectrum Control Panel, and click Configure, Model Type Editor.
   The Model Type Editor opens, and the Root model type is set as the current model type. The Root model type is the model type at the highest point in the model type hierarchy.

3. In the Model Type View, find the Security_Model model type.
4. Create a new model type, whose base model type is Security_Model.
5. In the newly created model type, add new attributes of type Relation Handle.
6. Set the default value of each new attribute to the Relation Handle of the relation that you want security strings to roll down.

Define Security String Roll Down Overrides for Model Types

When joining the security strings of two models for a security string roll down, the AND operator is used by default, unless the model type on the right side of the association has a predefined override.

DX NetOps Spectrum provides an override for the Container model type. When rolling down a security string to a model on the right side of a security relation whose model type is derived from Container, the OR operator is used. The only exception to this override is the WA_Link model type, which is derived from the Container model type. When rolling down a security string to a model on the right side of a security relation whose model type is WA_LINK, the AND operator is used.

Define security string roll down overrides in the Model Type Editor.

> **NOTE**
> For specific details about how to use the Model Type Editor and create new model types, see the Model Type Editor section.

Follow these steps:

1. Stop the SpectroSERVER if it is running, and verify that there are no other programs running that can access the SpectroSERVER database.
2. Open the Spectrum Control Panel, and click Configure, Model Type Editor.
   The Model Type Editor opens, and the Root model type is set as the current model type. The Root model type is the model type at the highest point in the model type hierarchy.
3. In the Model Type View, find the Security_Model model type.
4. Create a new model type, whose base model type is Security_Model.

   > **WARNING**
   > Create a new model type rather than modify the Security_Model model type directly since changes to the Security_Model type could be overwritten when installing future DX NetOps Spectrum upgrades.
5. In the newly created model type, take the following steps:
   a. Modify the default value of the Security_String_Mtypes (0x12967) attribute, adding the model types for which you want to define an override.
   b. Modify the default value of the Security_String_Operators (0x12968) attribute, defining the override operators (0 maps to AND, 1 maps to OR) for the model types that were added to the Security_String_Mtypes attribute. The value of instance x in the Security_String_Operators attribute should be the override operator for the model type identified by the value of instance x of the Security_String_Mtypes attribute.
6. Save your changes and close the Model Type Editor.

   > **NOTE**
   > Overrides that are defined on model types that are derived from the Security_Model model type take precedence over any overrides that are defined directly on the Security_Model model type.

Model Security Scenarios

The following scenarios provide examples of both simple and more complex model security use cases.

Secure a model in a remote office from local users

To help you understand security strings in OneClick, a simple example follows:

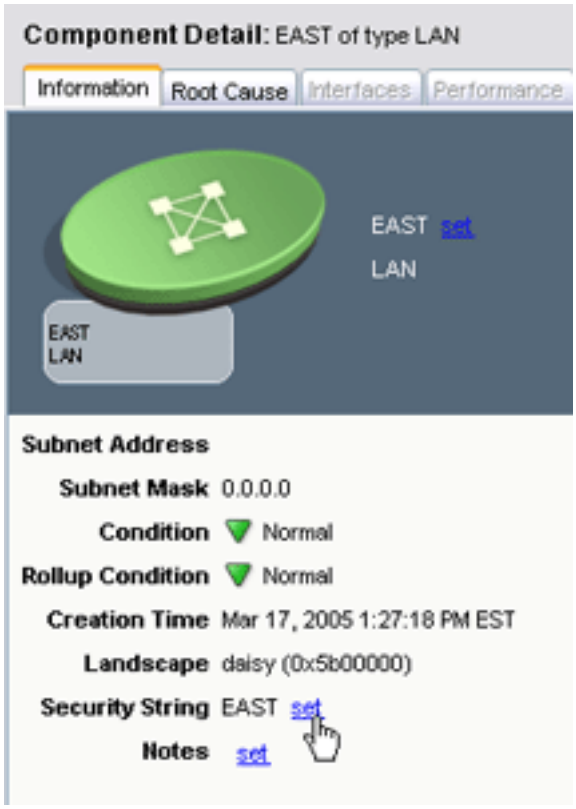
You want to secure a single model in a remote office so that local OneClick users cannot access it. Setting the security string of the model (to “remote,” for example) would secure it. Non-administrator users who lacked an access group with an entry of "remote" would not be able to access that model. Users with only an access group entry of "local", for example, would not have access to this model.

Secure administrative access to a branch office network

As a complete example of a security implementation in OneClick, consider an East Coast office and a West Coast office. Network administrators in the East coast office must have read/write access to the East Coast office’s network in OneClick. They must also have read-only access to the West Coast network. The inverse is true for the West Coast administrators.

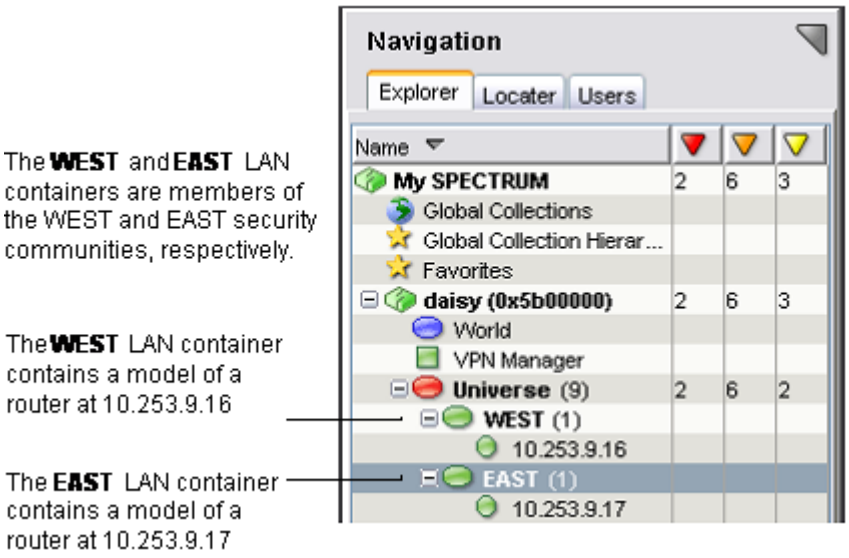
The following procedure can be used to create a solution to this requirement. It can also be modified to suit your needs.

1. Create two LAN containers in OneClick representing the two networks. Name one LAN container WEST, and name the other EAST.
2. Populate each container with different modeled network assets.
3. Set the security string on each LAN container. On the Information tab of the Component Detail panel for a selected LAN container, set the security string:
   a. Set the security string for the EAST LAN container to EAST. This step effectively creates a security community named EAST.

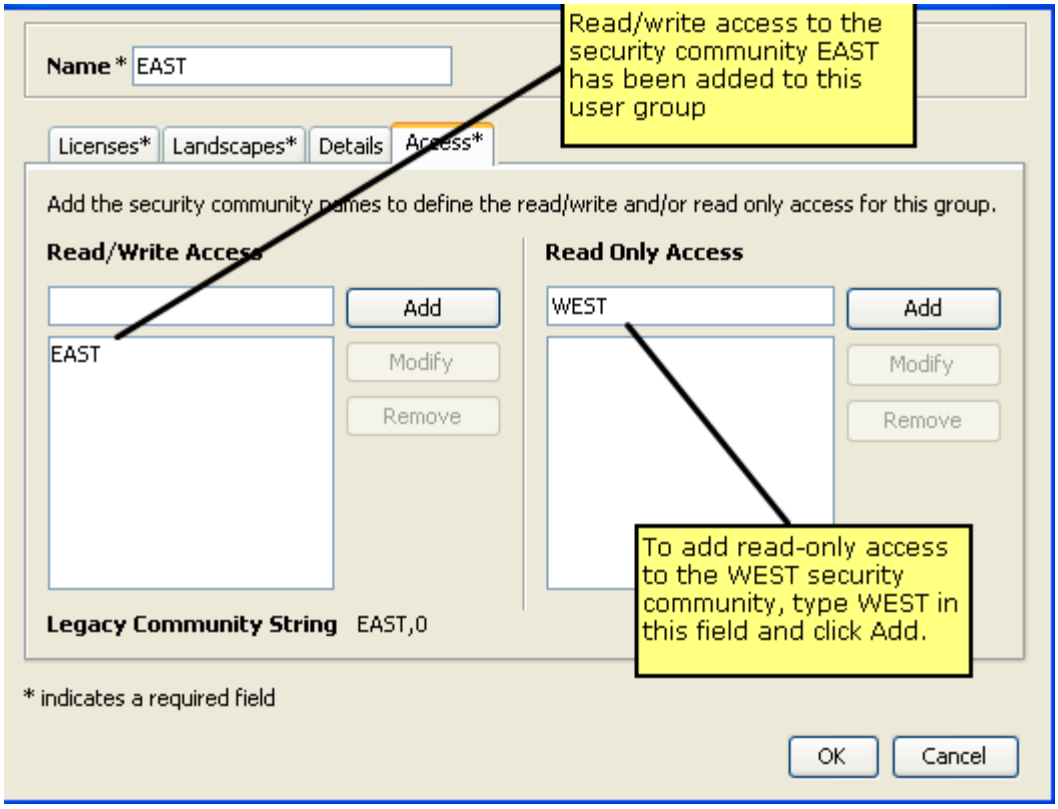


   a. Set the security string for the WEST LAN container to WEST. This step effectively creates a security community named WEST.

These security strings filter down from the LAN container level to its contained models. A security string set at the container level is automatically set for all its contained models.
When this task is complete, the Explorer tab of the Navigation panel resembles the following image to the OneClick Administrator user:



4. Create user groups to correspond with the EAST and WEST network containers:
   a. Create an EAST user group. In the Create Group dialog, create an access group with read/write privileges for the EAST security community:



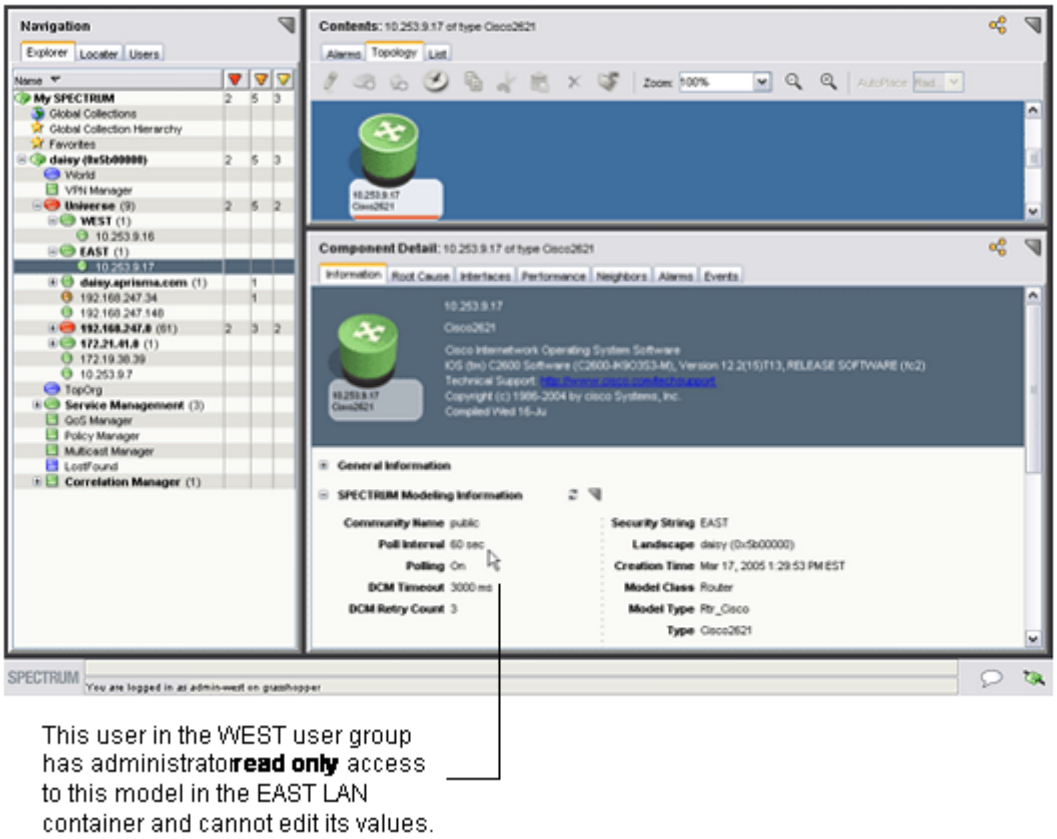
Create a WEST user group. In the Create Group dialog, create an access group with read-only privileges for the EAST security community.

Create a user inside the EAST user group and another user inside the WEST user group.

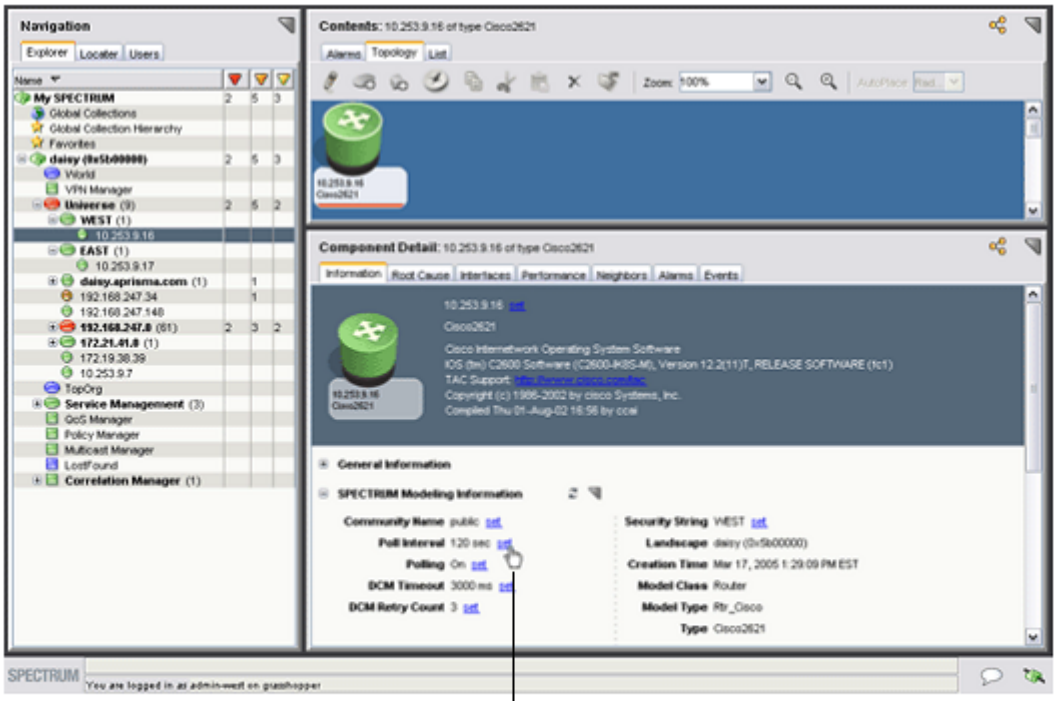
Note on the Access tab that the access groups (security community) are filled in from the User Group level (not editable here at the User level).

To test the changes, log in to OneClick as the user you created inside the WEST user group and navigate to the EAST LAN container.

When viewing models inside the EAST LAN container, users in the WEST user group have Administrator read-only rights as shown in the following image. For example, the image illustrates the fact that this user at this model cannot edit the values in Modeling Information.



1. Navigate to the WEST LAN container. Note that this user in the WEST user group has Administrator read/write privileges for models inside the WEST LAN, as shown in the following image. The image shows that this user at this model can edit the values in Modeling Information.



1. If you log in to OneClick as the user that you created inside the EAST user group and navigate to the WEST LAN container, the inverse situation is true: users in the EAST user group have Administrator read-only rights to the models inside the WEST LAN container. And they have read/write rights to the models in the EAST LAN.

Setting Preferences for Users and Groups

This chapter describes preferences in OneClick and how to use the Set Preferences dialog to set preferences for users and groups.

Set Preferences Dialog

Preferences in OneClick control the appearance of the OneClick console and the behavior of some user interface options. For example, Preferences control the fonts that are used in tables and the sort order of columns in user interface. You can configure the privileges for users and user groups in OneClick and can also set preferences for users and groups. For more information, see User Administration in OneClick. The Set Preferences dialog lets you set, lock, and save preferences for multiple users and groups.

NOTE
Selecting View, Preferences from the main OneClick menu opens user-level preference editing for the current user. The Alarm Filter dialog that is accessed from this menu can be launched from a button on the Alarms toolbar or from within the Set Preferences dialog.

The Set Preferences dialog organizes OneClick preference settings into the following groups of tasks:

- Alarms Tab
- Events Tab
- Explorer Tab
- General
- Interfaces Tab
- List Tab
- Locater Tab
- Topology Tab
- VPN Manager

If you select the top-level preferences group in the navigation panel, all available preferences and the tools to edit them are displayed in the content panel. Selecting a preference or preference group in the navigation panel displays the preference or preference group in the content panel.

The left panel of the Set Preferences dialog also lets you lock preferences for the selected user or user group. When you launch the Set Preferences dialog in the context of setting preferences for users and groups, the Set Preferences dialog displays the name of the user or group that is edited at the base of the navigation panel.

Access the Set Preferences Dialog

You can access the Set Preferences dialog to set preferences for a user or user group or to set preferences globally for all users.

User or User Group

Follow these steps:

1. On the Users tab, right-click a user or user group to set preferences.
2. Select Set Preferences from the menu.
   The Set Preferences dialog opens.
   You can now set the preferences for a user or user group.

All Users (Globally)

Follow these steps:

1. Right-click the top-level user group (Users) on the Users tab.
2. Select Set Preferences from the menu.
   The Set Preferences dialog opens.
   You can now set the preferences for all users.

About Setting or Locking Preferences

The OneClick administrator can set and lock user preferences at the global level (all users) or at the user group level. Users cannot lock their own preferences. If a preference is set and locked for a user or group, the user or members of the user group cannot change the preference.

Note: A locked preference can only be unlocked and edited at the level where it is locked. If a preference is locked at the global or user group level, the preference cannot be unlocked or edited at the user level. If the Set Preferences dialog is launched in the context of a given user and a preference is locked at the global or group level for that user, the administrator cannot change the preference status. The lock check box is disabled.

The following OneClick administrator privileges control the access to set user and group preferences:

- The Set User Preferences privilege grants access to set preferences for particular users and groups. This privilege is controlled by the user/group model security string. For more information, see the Glossary.
- The Set Global Preferences privilege grants access to set preferences at the global level.

The following figure shows preferences for the alarm count columns in the Explorer tab. These preferences are edited to display all alarms for the user group administration. No user in this group can change this preference because it is locked at the user group level. Locked preferences display a small padlock icon.

Set or Lock User Preferences

You set or lock user preferences for users or user groups.

Follow these steps:

1. Right-click the desired user or user group in the Users tab and click Set Preferences.
   The Set Preferences dialog opens.
2. Navigate to the preference you want to set in the hierarchy in the navigation panel.
3. Make changes to the preference in the right panel.
4. Select the check box in the Locked column to lock any corresponding preferences.
   Locking a preference group also locks all preferences that are contained by the preference group. The Locked At column shows the level at which the preference is locked (user, user group, or all users). The Locked By column displays the administrator who locked it.

Alarm Filter Preferences

In addition to being available from a button on the Alarms toolbar, the Alarm Filter dialog can also be launched from the Set Preferences dialog using the Alarms tab, Alarm Filter preference. The right panel displays the Set Alarm Filter button. Access to the alarm filter can be administratively locked. If the alarm filter preference is locked, the filter button in the Alarms toolbar is not available.

You can create multiple alarm filters that are selectable using the Available Filters drop-down in the Alarm Filter dialog. You can configure the available filters for a user or user group and then lock it so the filters cannot be changed but the user can still select from the list of available filters. The Available Filters drop-down is also available on the Alarms tab.

NOTE
For more information about creating alarm filters, see the Using OneClick section.

Export Individual Alarm Filter Preferences

You can also export individual alarm filter preferences to other users and user groups. The exported filters are added to the user's or user group's existing filters; they replace existing filters. You cannot import individual alarm filters. Instead, all filters from the importing user or user group are added to the existing filters; they replace existing filters.

NOTE

When exporting preferences in bulk you can only export all alarm filters. You are not given a choice to select individual alarm filters when exporting preferences in bulk. The exported filters replace existing filters.

How the Filter Preferences Work

You can set up either group level filters or user specific filters but not both. When you create alarm filter preferences at user level, you do not get new or modified alarm filters from the group since user preferences take priority over group preferences.

If user has own alarm filter preferences and want to view only group level preferences at user level, do one of the following actions:

- Lock the group level alarm filters or
- Reset the user specific alarm filter preferences

Reset Preferences

The Reset Defaults button in the Set Preferences dialog lets you reset preference values back to the default. Resetting the preference automatically applies to the selected user or user group. When you reset the preference, the following occurs:

- For a user, it defaults to:
  - The setting on the User Group if the user is in a group and the preference is set for the group
  - Otherwise, the global setting for all users if set
  - Otherwise, the factory default setting
- For a User Group, it defaults to:
  - The global setting for all users if set
  - Otherwise, the factory setting
- For all users (the top-level Users node), it defaults to:
  - The factory setting

You cannot reset preferences that are locked. If you are modifying the preferences for a user and a given preference is locked at the user's group level, you cannot edit, import, or reset that preference.

Import and Export Preferences

Preferences can be imported from a user or user group and exported to other users and user groups.

Follow these steps:

1. Right-click the desired user or group in the Users tab, and click Set Preferences.
   The Set Preferences dialog opens.
2. Select the preferences to import or export.
   Selecting a preference group selects all of the preferences that it contains. If the top-level Preferences folder is selected, all preferences are selected.
3. Take *one* of the following steps:
   - Click Import to import preferences.
   - Click Export to export preferences.
4. Verify that the preferences you want to import or export are selected in the dialog, and click OK.
   The Select User/Group dialog displays the available users and user groups.
   - When exporting preferences, select a user or user group to which to export. Selecting the top-level Users node specifies all users (global). You only see users/groups for which you have the Set User Preferences privilege. If you lack the Set Global Preferences privilege, you do not see the top-level Users node.
   - When importing preferences, select a single user or group from which to import.
   - For both import and export, you only see the users and user groups that you have permission to view.

NOTE
For both import and export, the lock state of each preference is also transferred. For example, importing a locked preference from another user also locks that preference for the target user. If you export a preference that is locked at a higher level for the target user/group, the preference setting is not saved.

Managing Searches

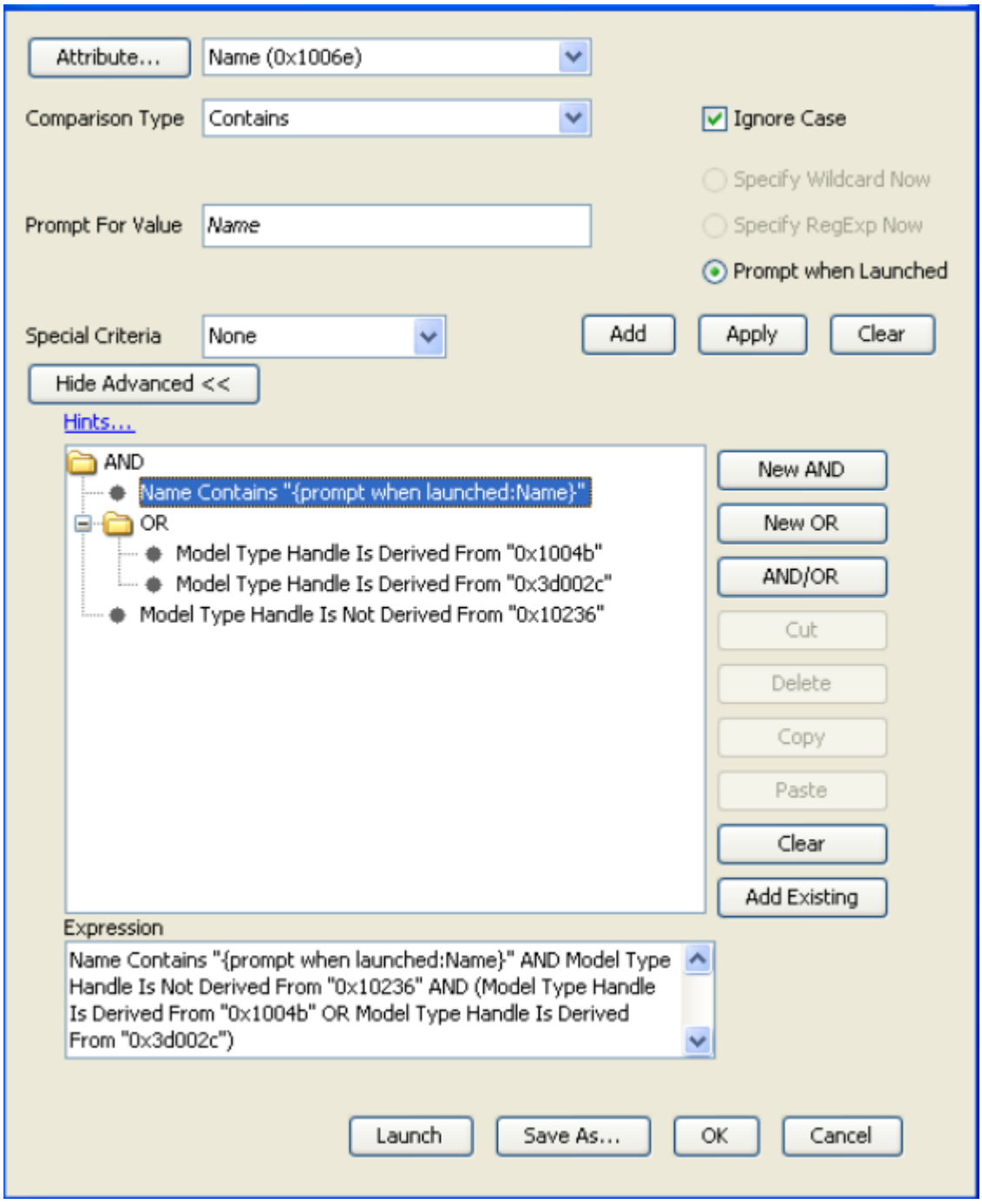
About Searches

You can create custom searches based on attribute values and various comparison criteria. This section describes how to create and manage custom searches. In general, these search management tasks are privileges that are granted only to OneClick administrators, not OneClick operators.

NOTE
While OneClick operators cannot create and manage searches, they can launch them. For information about how operators can use searches, see the Using OneClick section.

Create Search Dialog

The Create Search dialog contains several options and settings for creating simple and complex searches. The following image is an example of the Create Search dialog.



The options and settings available in the Create Search dialog depend on the type of search you are creating.

- **Attribute**
  Specifies an attribute of a device to filter.

  NOTE
  If you choose an alphabetic attribute value, you can either clear (ignore) or select (include) the Ignore Case check box.
- **Comparison Type**

Specifies the type of comparison to be made against the value specified in the Attribute field. Options can include Matches Pattern, Equal To, Not Equal To, Contains, Does Not Contain, Starts With, or Does Not Start With. Only the comparison types appropriate to the attribute's data type are available.

- **Ignore Case**
  Specifies whether the comparison should be case-sensitive. Selecting the Ignore Case check box makes the comparison not case-sensitive. This selection is only available when it is appropriate for the data type of the attribute you selected.
- **Attribute Value**
  Enter or select the desired attribute value you want to use in the comparison.

  NOTE
  Depending on the attribute type you select, you may be able to search for empty attribute values by leaving this field blank.

  Remove the "Allow PCRE searches" privilege for operators if you do not want them to run regular expression searches. Operators without this privilege will only be able to run wildcard searches for applicable searches.
- **Prompt for Value/Prompt When Launched**
  To create a search that prompts users to enter an attribute value when they run the search, select the Prompt when Launched option and then enter the prompt to display in the Prompt for Value field. This feature lets you create searches that are flexible enough to meet the different search requirements of OneClick users.
  Consider the following implementation examples:
  - If you want to create a search that locates any particular device type, you could create a search with a string comparison type (contains, does not contain, begins with, and so on) that prompts users to provide a particular device name when they run the search.
  - If you want to create a search that locates any device type with a particular Condition attribute value, you could create a search that prompts users to provide a particular condition value when they run the search.

Note: You can clear all fields at any time by clicking Clear.

- **Special Criteria**
  Constrains the search criteria in one of the following ways:

  NOTE

  The Special Criteria options cannot be used for 'Interface Attributes' and 'Device Attributes'. To search only devices or interfaces, use the 'None' option in the Special Criteria.

  - **None**
    Specifies that the search criteria will not be restricted to returning only devices or their interfaces.
  - **Interfaces of Devices**
    Specifies that you want the search to return only the interfaces of the devices it finds in the results list.
  - **Devices Only**
    Specifies that you want the search to return only devices in the results list.
- **Show Advanced**
  Opens the Advanced section of the Create Search dialog. The Advanced section in the Create Search dialog lets you create complex search criteria with any combination of nested AND clauses and OR clauses. This is represented in a tree structure grouped by logical operator (AND and OR) nodes. Each logical operator node can contain any number of attribute criteria nodes and other logical nodes. All nodes directly underneath a logical node are combined using the logical operator.
  - **Add**
    Adds a new attribute criteria node to the selected AND node or OR node with the information you entered into the Attribute, Comparison Type, and Attribute Value fields.
  - **Apply**

Applies the information entered in the Attribute, Comparison Type, and Attribute fields to the selected attribute criteria node.

- **New AND**
  Adds a new AND operator node to the selected AND node or OR node.
- **New OR**
  Adds a new OR operator node to the selected AND node or OR node.
- **AND/OR**
  Toggles the selected AND node or OR node. That is, if the logical operation is currently AND, clicking this button changes it to OR and vice versa.
- **Cut**
  Removes the selected node. It can be pasted below another node.
- **Paste**
  Pastes the last removed node below the selected AND node or OR node.
- **Clear**
  Removes all the nodes below the root node.
- **Add Existing**
  (Optional) Adds existing attribute-based, action-based, or relation-based searches to your custom search.

  NOTE
  Adding an existing search to your custom search copies the existing search and embeds it, as it is now, into your custom search. If the existing search is later modified, your custom search will not change because it contains only a copy of that existing search, as it was when you copied it and added it to your custom search.

- **Expression**
  Displays a textual representation of the search criteria as you create it.

Create Simple Searches

You can create searches that use complex criteria, such as a combination of AND clauses and OR clauses. A simple search contains only a single expression. You can also save searches for later use and organize them in folders.

Follow these steps:

1. Select the Locater tab in the Navigation panel.
2. Do one of the following in the Locater tab:
   - If you want to create a new search from a blank template, click the create a new search icon  .
   - If you want to create a new search based on an existing search, select a search and click the copy the selected search  .

     NOTE
     Some searches cannot be copied and used as the basis for another search. For example, Devices > By IP Address cannot be copied. However, you can create a new advanced search and can copy *any* predefined search criteria into that search. For more information, see Add Existing Searches to Custom Searches.

The Create Search dialog opens.

- Complete the fields in the dialog as desired.
- Click Save As.

The Save Search dialog opens.

- Enter a name and a description for the search.
- (Optional) Select the appropriate privilege if you want to limit access to the search to users who have a specific custom privilege. The privilege can be either assigned directly to the user or inherited from a role or user group.

  NOTE
  For more information about custom privileges, see the OneClick Customization section.
- Select a folder for the search.

  NOTE
  The Locater folder is the top-level folder.
- Click OK.
  The search is saved in the selected folder.
- (Optional) Click Launch to run the search.
  The search results appear in the Results tab of the Contents panel.
- Click OK.

Create Advanced Searches

Use the Advanced options in the Create Search dialog to create complex search criteria. You can build a search with many combinations of nested AND clauses and OR clauses.

Follow these steps:

1. Select the Locater tab in the Navigation panel.
2. Take one of the following steps on the Locater tab:
   - To create a search from a blank template, click the create a new search icon  .
   - To create a search from an existing search, select a search and click the copy the selected search icon  .

     NOTE
     Some searches cannot be copied and used as the basis for another search. For example, Devices > By IP Address cannot be copied. However, you can create a new advanced search and can copy *any* predefined search criteria into that search. For more information, see Add Existing Searches to Custom Searches.

The Create Search dialog opens.

- Complete the fields in the dialog as desired.
- Click Show Advanced to create complex search criteria that include a combination of AND clauses and/or OR clauses.
  The compound expression tree, logical operator buttons, and Expression field appear.
- Click Add to move the single expression that you created in Step 3 to the compound expression tree.
  The single expression appears in the compound expression tree.
- Click one of the following logical operator buttons to build a compound expression:
  - New AND
  - New OR
  - AND/OR

  The selected operator is inserted into the compound expression tree.
- Repeat Step 3, Step 5, and Step 6 for each compound expression that you want to build.
- (Optional) Add existing predefined search criteria.
- Click Save As.

The Save Search dialog opens.

- Enter a name and a description for the search.
- (Optional) Select the appropriate privilege from the Privilege drop-down list. Privileges limit access to the search to users with a specific custom privilege. Custom privileges can either be assigned directly or they can be inherited from a role or user group.

  NOTE
  For more information, see the OneClick Customization section.
- Select a folder in which to save the search from the Save In Folder section.

  NOTE
  The Locater folder is the top-level folder.
- Click OK.
  The Save Search dialog closes and you return to the Create Search dialog.
- (Optional) Click Launch to run the search.
  The search results appear in the Results tab of the Contents panel. The applicable entities have been excluded from the results list based on the compound search expressions you specified.
- Click OK.
  The Create Search dialog closes and you have now created an advanced search.

Add Existing Searches to Custom Searches

You can add existing attribute-based, action-based, or relation-based searches to any custom search you create. This lets you include predefined search criteria from existing searches including special searches such as All Devices and Devices By IP Address Range.

NOTE
Adding an existing search to your custom search copies the existing search and embeds it, as it is now, into your custom search. If the existing search is later modified, your custom search will not change because it contains only a copy of that existing search, as it was when you copied it and added it to your custom search.

To add an existing search to your custom search

1. Click the Locater tab in the Navigation panel.
2. Click the create a new search icon 
   The Create Search dialog opens.
3. Complete the fields at the top of the dialog as desired.
4. Click Show Advanced.
   The compound expression tree, logical operator buttons, and Expression field are displayed.
5. Click Add Existing.
   The Add Existing Search dialog opens.
6. Select the existing search that contains the criteria you want to copy and add to the current search and click OK.
   The Add Existing Search dialog closes and the criteria you selected is added to the compound expression.
7. (Optional) Click set next to the criteria you added to modify prompt information as desired.
   The Search dialog opens.
8. Do *one* of the following depending on whether you want to prompt users for a value:
   - Select 'Prompt the user' to configure how you want to prompt users:
   - **Prompt text**
     Specifies the text you want to prompt users with when they run the search.
   - **Default value**

Specifies a default value for this prompt.
  - **Note:** The default value is not shown to users until they run this search.
  - Select 'Specify value now' to enter the prompt value yourself now; users are not prompted to enter anything when they run this search.
9. Click OK.
10. Save the search as described Create Advanced Searches.
    You have now created a custom search that includes the addition of an existing search.

Search Recommendations

The following provides search criteria recommendations when defining advanced searches. The order of the criteria can affect the search performance.

The order of attribute criteria is based on two categories: *storage of information* and *data type.*

- **Storage of information**
  Attributes should be ordered from least CPU (quickest access) to most CPU (slowest access), as follows:
  - Memory flag (least CPU/quickest access)
  - Database flag
  - Calculated
  - External flag (most CPU/slowest access)
- **Data type**
  Attributes should be ordered from quickest comparison to slowest comparison, as follows:
  - Integer, counter, enumeration, model type handle (quickest comparison)
  - IP address, octet string
  - Text string (slowest comparison)

Combining the two categories of criteria, the overall attribute placement for complex searches of AND/OR order from top to bottom is as follows:

1. Memory flag
   a. Integer, counter, enumeration, model type handle
   b. IP address, octet string
   c. Text string
2. Database flag
   a. Integer, counter, enumeration, model type handle
   b. IP address, octet string
   c. Text string
3. Calculated
   a. Integer, counter, enumeration, model type handle
   b. IP address, octet string
   c. Text string
4. External Flag
   a. Integer, counter, enumeration, model type handle
   b. IP address, octet string
   c. Text string

Example

You would like to define a search based on the following search criteria (in no particular order):

- ifDesc
- Topology model name string
- Network address
- Model type handle

How should these attributes be ordered for best performance?

Using the recommended ordering logic, the following is the recommended order:

1. Model type handle (memory flag : model type handle)
2. Network address (memory flag/database flag : IP address)
3. Topology model name string (calculated flag : text string)
4. ifDesc (external flag : text string)

Edit Searches

You can edit a custom search that you have saved. The predefined searches cannot be modified.

Follow these steps:

1. In the Locater tab, select the search from the available searches, and click the edit the selected search  .
2. Edit the search using the controls that are described in Create Search Dialog. Select an attribute criteria node to see its information. You can then modify the attribute criterion.
3. Click Apply to change the selected node. Or click the Add button to create a new attribute criteria node.
4. Click OK.
   The modified search is saved.

Delete Custom Searches

NOTE
You cannot delete preconfigured folders and searches, but you can delete custom searches.

Follow these steps:

1. On the Locater tab, to organize, rename, or delete your searches, click  .
   The Organize Searches dialog opens.
2. Navigate to the custom search, and select it.
3. Click Delete.
4. Click OK.
   The custom search is deleted.

Organize Custom Searches

NOTE
You can organize your custom searches in a folder hierarchy. Predefined folders and searches cannot be edited.

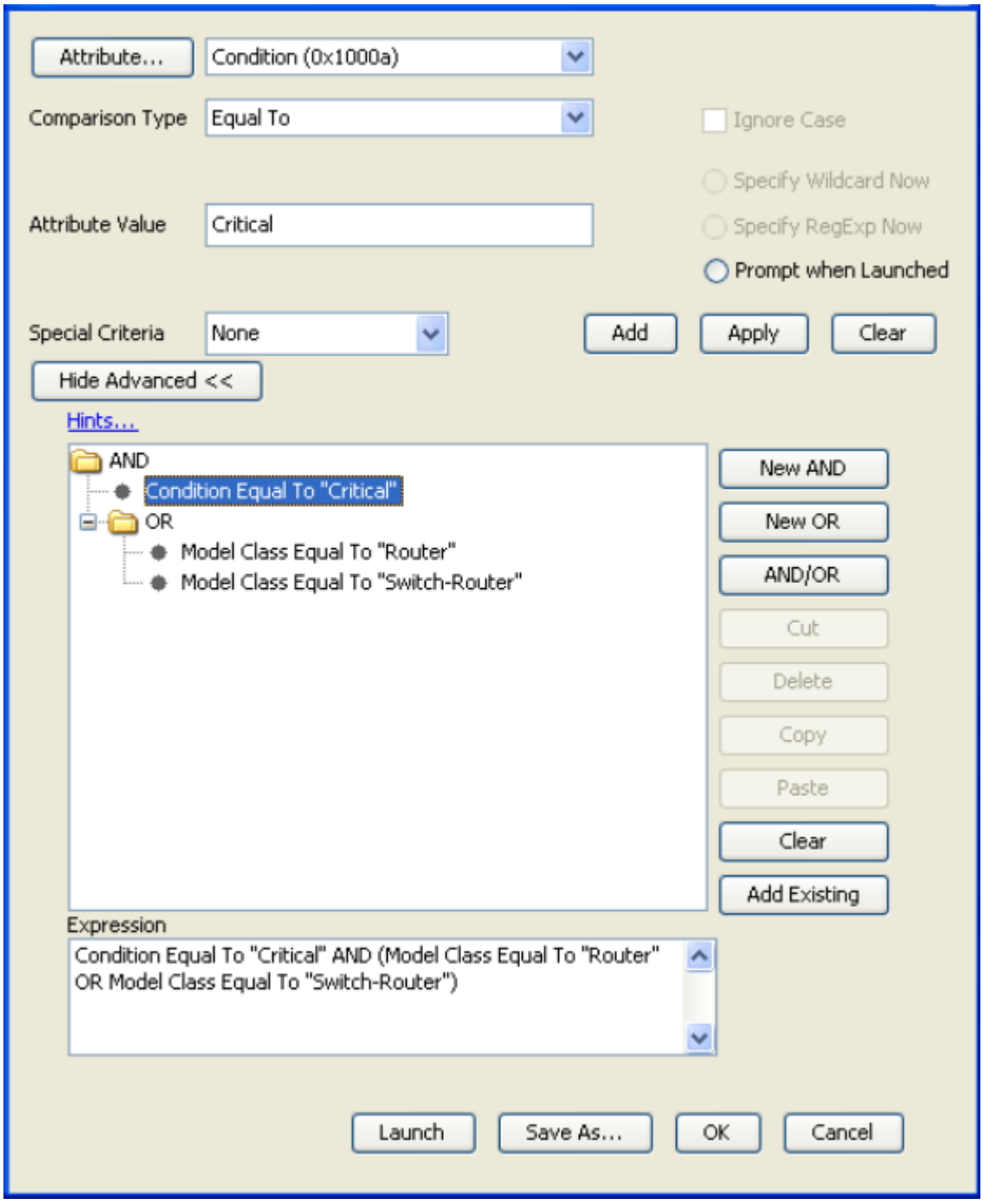
Follow these steps:

1. On the Locater tab, to organize, rename, or delete your searches, click  .
   The Organize Searches dialog opens.

2. Use the dialog to create a hierarchy of folders.
3. Move the searches that you have created into the new folders.
4. Use the Organize Searches dialog to rename or delete your custom folders and searches.
5. Click OK.
   Your custom searches are organized.

Example Search Find Devices In Critical Condition

Create a compound search that finds all routers or switch routers with a status of "Critical." The following image shows an example of the Create Search dialog after the appropriate compound expressions have been added:



The following procedure provides an example of a useful compound search.

Follow these steps:

1. Select the Locater tab in the Navigation panel.
2. On the Locater tab, to create a new search,

   click 
   The Create Search dialog opens.

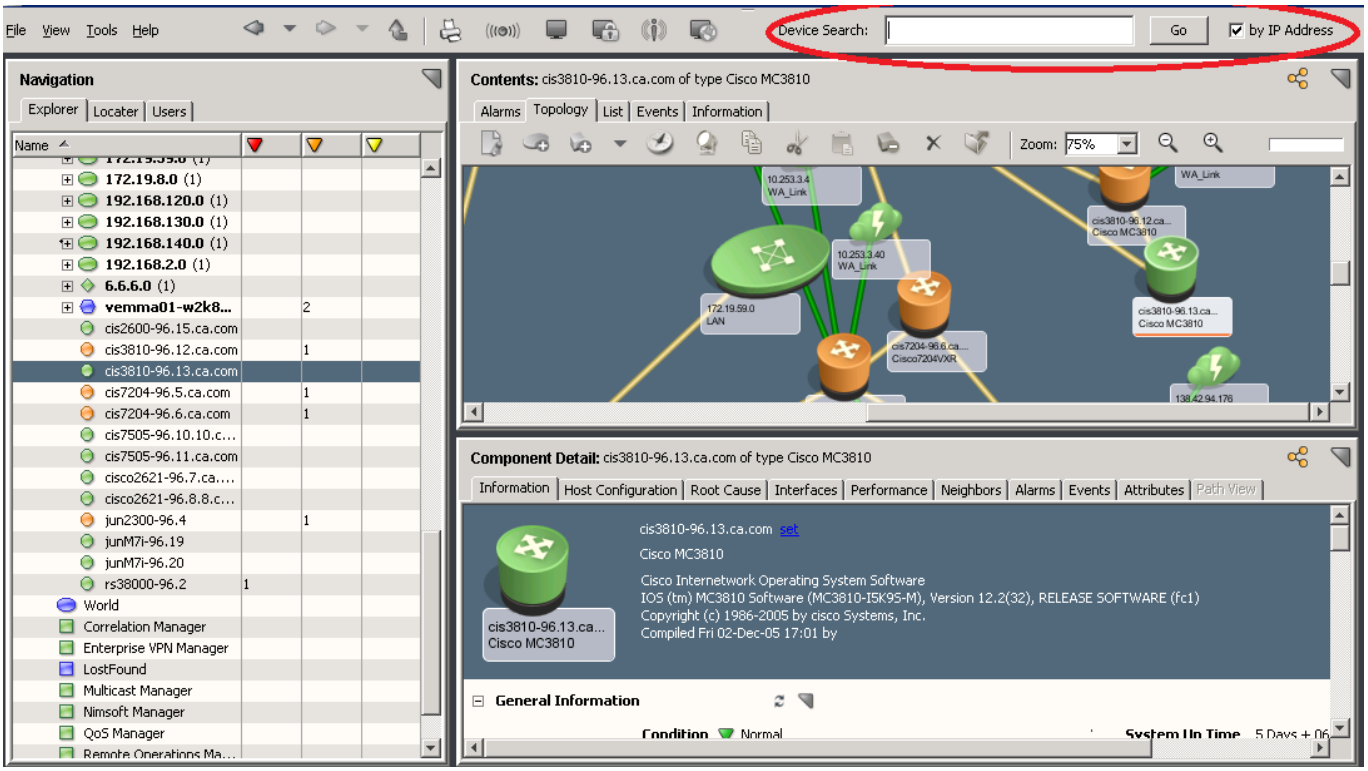
3. Complete the fields as follows:
   - **Attribute**
     Condition (0x1000a)
   - **Comparison Type**
     Equal To
   - **Ignore Case**
     N/A
   - **Attribute Value**
     Critical
4. Click the ‘Show Advanced’ button.
   The compound expression box and logical operator buttons appear.
5. Click Add to move the single expression created in Step 3 to the compound expression tree.
   The single expression appears in the compound expression tree in text string format in the Expression field at the bottom of the Create Search dialog.
6. Click the ‘New OR’ button.
   The OR operator is inserted into the compound expression tree, beneath the expression: Condition Equal To “Critical.”
7. Complete the fields on top of the Create Search dialog, using the following parameters:
   - **Attribute**
     Model Class (0x11ee8)
   - **Comparison Type**
     Equal To
   - **Ignore Case**
     N/A
   - **Attribute Value**
     Router
8. Click Add to move this expression to the compound expression tree.
   This expression (Model Class Equal To “Router”) is inserted into the compound expression tree, beneath the OR operator.
9. Complete the fields on top of the Create Search dialog, using the following parameters:
   - **Attribute**
     Model Class (0x11ee8)
   - **Comparison Type**
     Equal To
   - **Ignore Case**
     N/A
   - **Attribute Value**
     Switch-Router
10. Click Add to move this expression to the compound expression tree.
    This expression (Model Class Equal To “Switch-Router”) is inserted into the compound expression tree, beneath the OR operator.
11. (Optional) Click Save As to save this search in the Locater tab. You can then run it at any time.
12. Click Launch to run the search immediately.

The search results appear in the Results tab of the Contents panel.

About OneClick Quick Device Search

OneClick allows you to directly find a device model without going to the Locater Search. Enter the full IP address of the device in the device search bar of the OneClick console. You can also find a device by its redundant IP addresses, if that device has **Redundancy Preferred Addresses**. To find a device by its full name or a string, uncheck the "by IP Address" check box and enter the full name or a string.

The following image shows how the device search bar looks like:



Searching by a String

A search result that includes a list of all devices containing the matching string in their names is displayed. The search result includes device name, IP address, and the device landscape.
When you double-click the required device from the search result, the device is highlighted in the navigation panel.

Searching by Full Device Name or IP Address

The exact device is highlighted in the navigation panel. In a DSS environment when the device exists in multiple landscapes, you get the search result with the same information.

Select the device belonging to the required landscape.

In both cases, alarms and other information of the device are displayed in the Contents and Component Detail pane.

NOTE

Ensure to enter a valid IP address when you find by IP address. Only a valid IPv4 address enables the search.

Troubleshooting OneClick

This section discusses some of the most common problems while working with OneClick.

Non-LDAP Users Cannot Log In

Error binding: javax.naming.AuthenticationException: [LDAP: error code 49 - 80090308: LdapErr: DSID-0C0 903A9, comment: AcceptSecurityContext error, data 52e, v1db1]; remaining name

Reason:

The 'Allow user to login if no LDAP user is found' option is enabled, but your non-LDAP users cannot log in. This error occurs when LDAP is configured using "User by Pattern."

Action:

Reconfigure LDAP to use "User by Search."

Memory Resources Not Available

The memory resources required to complete the operation were not available.

Reason:

You are attempting to export very large (4000x4000 pixels and greater) Topology view images from OneClick.

Action:

Either reduce the image size by zooming out in the Topology view or increase the OneClick client memory settings as described in Configure OneClick Memory Settings.

OneClick Web Server Shuts Down

OneClick Web Server Shuts Down

Symptom:

I upgraded to VMware 2.0 and it runs an Apache Tomcat server of its own. After I install the OneClick web server, the OneClick web server shuts down when it attempts to bind to port 8005. Then, I receive the following error message:

```
- StandardServer.await: create[8005]:
java.net.BindException: Address already in use: JVM_Bind
```

Solution:

By default, Apache Tomcat uses port 80 on Windows platforms and port 8080 on Linux platform. If SSL is configured, Apache Tomcat uses port 443. Apache Tomcat also uses the default server shutdown port 8005. When installing the OneClick web server, be sure that other applications on the same computer do not use these ports. Or, you can change the ports on the instance of Apache Tomcat that DX NetOps Spectrum uses.

Blank Panels in OneClick Clients

Symptom:

In a fault-tolerant environment, I am seeing OneClick clients display blank panels after failover to the secondary SpectroSERVER has occurred. OneClick clients display three empty (gray) panels but the connection status shows that the failover switch has occurred.

Solution:

The blank panels occur because user privileges are not in sync between the primary and secondary servers, and the privileges are lost during failover. All user models must be created and completely configured on the primary SpectroSERVER before the primary server's database is copied to the secondary SpectroSERVER. If they are not, any actions done to User models (user associations made to license roles, access groups, and so on) will not be in sync with the secondary server until an online backup occurs. For more information, see the discussion about online backups in the Database Management section.

OneClick Web Server Shuts Down

Symptom:

I upgraded to VMware 2.0 and it runs an Apache Tomcat server of its own. After I install the OneClick web server, the OneClick web server shuts down when it attempts to bind to port 8005. Then, I receive the following error message:

```
- StandardServer.await: create[8005]:
java.net.BindException: Address already in use: JVM_Bind
```

Solution:

By default, Apache Tomcat uses port 80 on Windows platforms and port 8080 on Linux platforms. If SSL is configured, Apache Tomcat uses port 443. Apache Tomcat also uses the default server shutdown port 8005. When installing the OneClick web server, be sure that other applications on the same computer do not use these ports. Or, you can change the ports on the instance of Apache Tomcat that DX NetOps Spectrum uses.

> **NOTE**
> We recommend that you do *not* install the OneClick web server on a computer where an instance of Apache Tomcat is already running.

Using the getSpectrumInfo Script

getSpectrumInfo is a script used to gather information about your DX NetOps Spectrum environment. The collected data is written to a file that can conveniently be sent to CA Support. The following lists some of the data that is included:

- host information
- configuration files
- installation logs
- SpectroSERVER logs
- Tomcat logs

To use the getSpectrumInfo script

1. Log in to the system for which you want to collect environment data. You will need write permissions to the DX NetOps Spectrum installation directory to create the output file.
2. Prepare to enter the script, as follows:
   - On Windows:
     - From the Start, Run menu, type **cmd**, and click OK.
       The DOS prompt appears.
     - Enter **bash - login** to start a bash shell.
     - Navigate to the DX NetOps Spectrum installation directory.
   - On UNIX platforms, navigate to the DX NetOps Spectrum installation directory.
3. Enter the following command to run the script:
   ```
   ./bin/support/getSpectrumInfo.sh [full|lite|mini]
   ```
   You can use the following parameters on the command:
   full - The complete set of environment data, including all of the Install-Tools/LOGS directory, is collected. The getSpectrumInfo.sh command without any parameters defaults to this option. The output file created can be large.

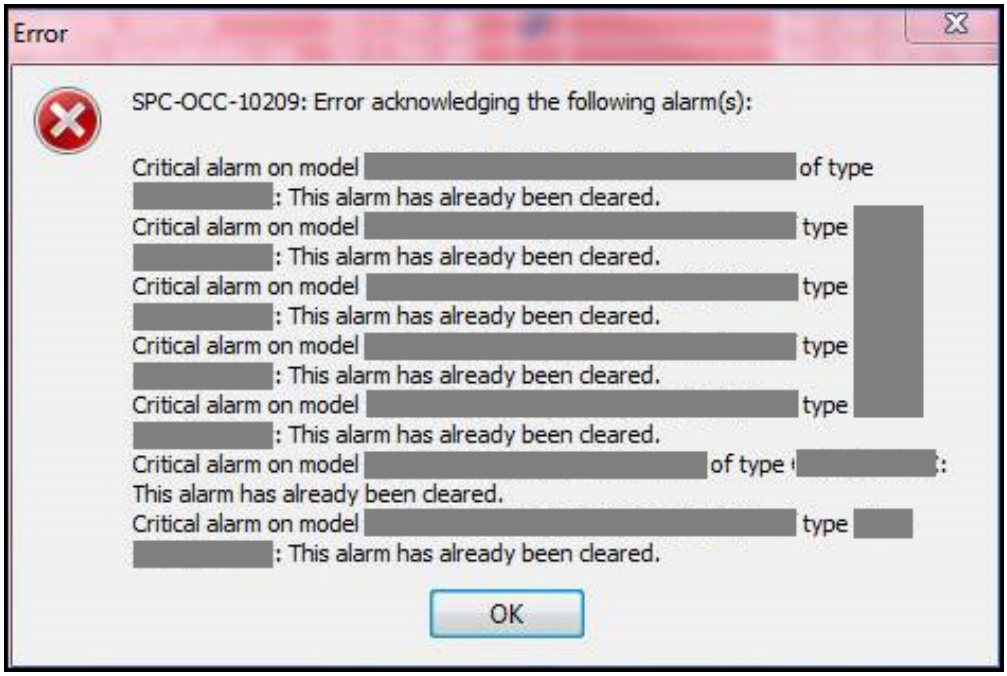
lite - A subset of environment data, including selected files from the Install-Tools/LOGS directory, is collected.
mini - Only the minimum environment data is collected.
The getSpectrumInfo script begins and displays informational messages as it runs. When it completes, a zipped file is created in the DX NetOps Spectrum installation directory in the following format:

```
logs-hostname-YYMMDD-nnnn.tar.gz
```

4. Contact CA Support for where to upload the file.

Unable to acknowledge or clear alarms in OneClick Console

Issue:
In OneClick, when an alarm is manually acknowledged or cleared, you may observe the following errors:



Environment:
All supported platforms.

Probable Cause:
This could happen when the SpectroSERVER and the OneClick Server has synchronization issue.

Resolution:
To resolve the synchronization issue, please schedule restart of the OneClick Web Server.

To know how to restart the OneClick Web Server, see Start and Stop the OneClick Web Server from the Command Line section.

Error Message in the Landscape Configuration Section

Running SpectroSERVER process using '**root**' user instead of '**spectrum**' user on linux machines may result in the following error message "*The user running the parent server does not have a user model in the local landscape*" on the landscape page found in the administration tab on the oneclick console.

Run 'ps -ef | grep -i spectro' to check if the SpectroSERVER process is owned by '**root**' user or other user instead of '**spectrum**' user (You can check the value of '**initial_user_model_name**' parameter in $SPECROOT/SS/.vnmrc file to check for the '**spectrum**' user) to resolve the issue.

Follow the steps mentioned to fix the issue:

1. Launch Spectrum Control Panel using '**root**' or other user who is currently owning the SpectroSERVER process. Stop SpectroSERVER process thoroughly by clicking [**Stop SpectroSERVER**] button.

2. Change the ownership of VNM.OUT, RCPD.OUT etc. under $SPECROOT/SS directory back to '**spectrum**' user using 'chown' command. Make sure when you run 'ls -l' under $SPECROOT/SS directory all files and directories are owned by '**spectrum**' user except the SpectroSERVER file. The SpectroSERVER file should be owned by '**root**' user and have the following attributes:

```
-rwsr-x---   1 root     spectrum     12841 Nov 21 01:16 SpectroSERVER
```

3. Launch Spectrum Control Panel using '**spectrum**' user and start SpectroSERVER process by clicking [**Start SpectroSERVER**] button.

System Customizations for OneClick

This section lists the parameters that can be edited in the context.xml file and the web.xml file to customize the server and client environment.

context.xml Customization Parameters

The context.xml file, located in the <$SPECROOT>/webapps/spectrum/META-INF directory, contains many OneClick customization parameters. You must restart the OneClick web server after making changes to this file.

- **maxProcessors**
  Controls the maximum number of OneClick clients that can be running:
  ```
  <parameter>
      <name>maxProcessors</name>
      <value>75</value>
  </parameter>
  ```
- **locServerName**
  Provides the hostname of the DX NetOps Spectrum location server:
  ```
  <parameter>
      <name>locServerName</name>
      <value>snowball</value>
  </parameter>
  ```
- **orbAgentName**
  ```
  <parameter>
      <name>orbAgentName</name>
      <value>snowball</value>
  ```

```
</parameter>
```

- **orbAgentPort**

```
<parameter>
    <name>orbAgentPort</name>
    <value>14000</value>
</parameter>
```

- **adminUserName**

```
<parameter>
    <name>adminUserName</name>
    <value>admin</value>
</parameter>
```

- **smtpHostName and smtpPort**

  Configure these parameters to set the host name of your mail server and the port that it uses, respectively:

```
<parameter>
    <name>smtpHostName</name>
    <value>mailhost</value>
</parameter>
<parameter>
    <name>smtpPort</name>
    <value>25</value>
</parameter>
```

- **useSecondarySS**

  A value of false prevents failover:

```
<parameter>
    <name>useSecondarySS</name>
    <value>true</value>
</parameter>
```

web.xml Customization Parameters

The web.xml file contains additional customization parameters. It is located in the following directory:

```
<$SPECROOT>/tomcat/webapps/spectrum/WEB-INF directory
```

You must restart the OneClick web server after making changes to this file.

To configure the OneClick web server to use a path to SG-Support other than the default, edit the value of the com.aprisma.spectrum.root.install parameter in the following section of the web.xml file:

```
<context-param>
    <param-name>com.aprisma.spectrum.root.install</param-name>
    <param-value>/usr/SPECTRUM/WebApps/SG-Support</param-value>
    <description>
    This parameter defines the absolute path to the directory where
    SG-Support was installed for the Spectrum core product. This
    directory should be <$SPECROOT>/SG-Support.
    </description>
</context-param>
```

HTTP method vulnerability

HTTP offers several methods that can be used to perform actions on the web server. Many of these methods are designed to aid developers in deploying and testing HTTP applications. These HTTP methods can be used for nefarious purposes by intruders if the web server is misconfigured and can make the server vulnerable.

The following restricted methods are not used by DX NetOps Spectrum. So they can be disabled safely.

Add the following methods to $SPECROOT/tomcat/conf/web.xml, towards the end of the file (that is, above the </web-app> end tag) for restricting/disabling these methods. Restart the OneClick server after this change.

<security-constraint>
<web-resource-collection>
<web-resource-name>restricted methods</web-resource-name>
<url-pattern>/*</url-pattern>
<http-method>TRACE</http-method>
<http-method>PUT</http-method>
<http-method>DELETE</http-method>
<http-method>HEAD</http-method>
<http-method>OPTIONS</http-method>
</web-resource-collection>
<auth-constraint />
</security-constraint>
<filter>
<filter-name>CorsFilter</filter-name>
<filter-class>org.apache.catalina.filters.CorsFilter</filter-class>
<init-param>
<param-name>cors.allowed.methods</param-name>
<param-value>GET,POST,CONNECT</param-value>
</init-param>
</filter>
<filter-mapping>
<filter-name>CorsFilter</filter-name>
<url-pattern>/*</url-pattern>
</filter-mapping>

Dynamic Host Configuration Protocol (DHCP) Support

In a DHCP environment, the IP addresses of certain devices change based on the DHCP lease. Prior to 10.3, DX NetOps Spectrum was not able to monitor these types of devices properly.
Starting From 10.3, you can run DNS Lookup to resolve the original IP address of the devices in DHCP environment.

DNS Lookup

Starting from the 10.3 release, you can trigger DNS lookup to resolve the correct IP address for the following issues:

- Device Contact Lost
- Duplicate IP or model
- Management Agent Lost
- Device IP swap (Schedule DNS Lookup)

To enable DNS lookup in DX NetOps Spectrum, follow these steps:

1. Log in to OneClick console.

2. Click the Explorer tab.
3. Select the device model on which you want to enable the DNS lookup.
4. Click the 'Attributes' tab in the Component Details panel.
5. *Search for the 'do_dns_lookup' attribute.*
6. Change the attribute value to 'Yes'.Default: No
   The DNS lookup feature is enabled for the selected model.

   NOTE

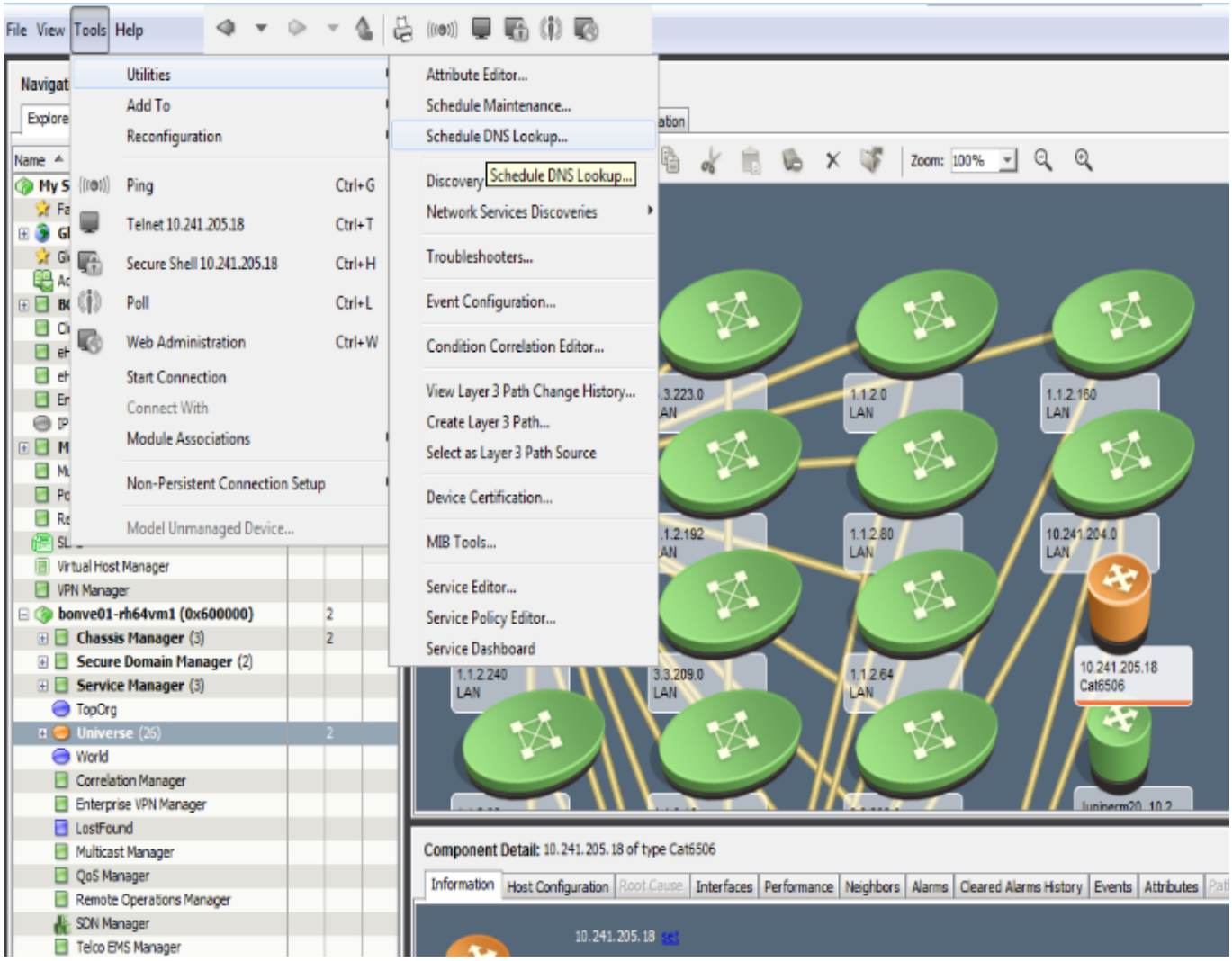
   Though the '*do_dns_lookup'* attribute is set on a device, if the ‘IP redundancy’ is also enabled, it does not trigger the DNS Lookup.

Schedule DNS Lookup

The Schedule DNS lookup is created to perform DNS lookup at a specified date and time. The DNS lookup is triggered only for the models having ' *do_dns_lookup*' attribute value set to yes, as part of the schedule.

Follow these steps to schedule a DNS lookup:

1. Log in to OneClick console.
2. Click the Explorer tab.
3. Select the device model on which you want to create a schedule for the DNS lookup.
4. Navigate to **Tools**, **Utilities**, **Schedule DNS Lookup**.



5. Select from the **Available Schedules** or create **New** schedules.
   Available recurrences are **Daily**, **Weekly**, **Monthly**, **Yearly**.
   (The time that is specified is relative to the 'SpectroSERVER's local time zone).
   After the schedule is created and attached to a model or group of models, the DNS Lookup is performed on those models when the schedule goes active.
6. To see the models that are associated to a DNS schedule, navigate to **Locater, Schedules, All Schedules**, **Information Tab** and expand **Items Scheduled for DNS Lookup** subview.

Events for DNS Lookup

The following events are generated, when a DNS schedule is associated and disassociated:

- 0x11005 - (DNS Schedule is associated)
- 0x11006 - (DNS Schedule is disassociated)

The following event is generated, when the IP changed after the DNS lookup.

- 0x11007

Enable and Disable Event Logging

To enable event logging, from the command-line interface, run the following command:

```
update action=0x10803 mh=<model handle of the model for which logging has to be enabled>
```

To disable event logging, from the command-line interface, run the following command:

```
update action=0x10804 mh=<model handle of the model>
```

The disable command disables the logging irrespective of the model for which the logging is enabled.

You can find the above events that are captured as part of logging into VNM.OUT.

Single Sign-On

Single Sign-On is the authentication scheme for DX NetOps Spectrum and all supported data sources. Once they are authenticated to DX NetOps Spectrum, users can navigate the console and registered data sources without signing in again.

Enabling the navigation of multiple product interfaces ensures a seamless drilldown experience for operators analyzing performance and status data. For example, if a user logs in to DX NetOps Spectrum and follows a drilldown path to the data source interface, that the user does not log in again.

DX NetOps Spectrum uses a distributed architecture. The Single Sign-On website is automatically installed on every server where a supported data source or DX NetOps Spectrum is installed. The distributed architecture lets users log in to data source products by logging in to the servers where these products are running.

SAML2 Authentication in DX NetOps Spectrum

You can authenticate users with SAML 2.0 through your organization's Identity Provider (IdP). DX NetOps Spectrum now supports Security Assertion Markup Language (SAML) 2 authentication as a single sign-on login standard for this purpose. SAML is a standard for logging users into applications based on their sessions in another context.

SAML2 authentication in DX NetOps Spectrum supports the following IdPs:

- Auth0
- Okta
- Onelogin
- Microsoft Azure Active Directory (Azure AD)
- Any Other SAML2.0 IdP

Enable Single Sign-On

The following procedure lists the steps to integrate DX NetOps Spectrum with the identity provider server.

Follow these steps:

1. Enable SSL in DX NetOps Spectrum OneClick Server.
2. Configure the IDP server - Create DX NetOps Spectrum app in Identity Provider (IdP) Server.
3. Create DX NetOps Spectrum users for Single Sign-On.

   NOTE
   The **IdP usernames** and **DX NetOps Spectrum usernames** must match, else the authentication fails.
4. Download the IdP Server x509 certificate and Import the certificate into KeyStore of DX NetOps Spectrum OneClick Server.

5. Configure the fediz_config.xml file.
   a. Open the `fediz_config.xml` file located in the `<SPECROOT>/tomcat/conf` directory.
   b. Update the following fields:
      - **audienceItem**
        Specifies the Audience URI. For example, https://spectum_host/spectrum/

        NOTE
        Audience URI: The application-defined unique identifier that is the intended audience of the SAML assertion. This is most often the SP Entity ID of your application.
      - **Issuer**
        Specifies the IdP Single Sign-On URL. For example, https://oneclickhostname.broadcom.net:8443/spectrum/

        NOTE
        Single Sign-On URL: https://oneclickhostname.broadcom.net:8443/spectrum/ The location where the SAML assertion is sent with an HTTP POST. This is often referred to as the SAML Assertion Consumer Service (ACS) URL for your application.
      - **realm**
        Specifies the IdP Audience URI. For example, https://spectum_host/spectrum/

        NOTE
        The **audienceItem** and **realm** parameters must have the same value.
      - **(Optional) reply**
        Single Sign-On URL: https://oneclickhostname.broadcom.net:8443/spectrum/ The location where the SAML assertion is sent with an HTTP POST. This is often referred to as the SAML Assertion Consumer Service (ACS) URL for your application.
   c. Update the Keystore file location in the following parameters values:
      - certificateStores
      - signingKey
      - tokenDecryptionKey
   d. **Retain the default values for all other parameters.**
   e. Save and close the file.
6. Configure Basic Authentication for other product integrations.
   DX NetOps Spectrum SAML Authentication supports web browser single sign-on. Some integration clients like the ones listed below are not browser-based applications:

   - Spectrumgtw probe - UIM Integration
   - DX NetOps Spectrum - DX OI Connector
   - Other clients which use Rest API's to communicate with DX NetOps Spectrum.

   For such clients, DX NetOps Spectrum allows the user to communicate using the basic authentication.
   To configure basic authentication, follow the below steps:

   1. Open the `non-saml-config.xml` file from the `<SPECROOT>/tomcat/conf` directory.
   2. Set the `allowBasicAuthentication` parameter to true.
      ```
      <allowBasicAuthentication>true</allowBasicAuthentication>
      ```
   3. Specify the spectrum user name which is used for integrations. You can use only the listed users in this configuration.
      ```
      <userName>spectrumUser1</userName>
      <userName>spectrumUser2</userName>
      ```
   4. Save and close the file.
7. Enable SAML Authentication.
   a. In the OneClick Web interface, click the **Administration** tab.
   b. Click the **Single Sign-On Configuration** link.

c. Select SAML as SSO option.
d. In case the tomcat does not restart automatically, manually restart the tomcat.
e. Stop the tomcat and start the tomcat.
For more information, see Start and Stop the OneClick Web Server from the command-line.
The changes are applied after the tomcat server restarts.

Disable Single Sign-On

You can disable the SAML authentication when it is not required. You can use the DX NetOps Spectrum administration UI to disable SAML. In case, you did not create the IdP user in DX NetOps Spectrum and enabled SAML, you can disable SAML using the command line.

Disable SAML using GUI

Follow these steps:

1. In the DX NetOps Spectrum, navigate to Administrator, Single Sign-On Configuration.
2. Select **No Single Sign-On.**
3. Save and confirm save at the prompt.
4. Restart the Tomcat server.

Disable SAML using Command Line

Follow these steps:

1. Open the `context.xml` file from the `<SPECROOT>/tomcat/conf/` directory.
2. Change the value from `com.aprisma.tomcat.authenticator.Saml2FederationAuthenticator` to `org.apache.catalina.authenticator.BasicAuthenticator`.
3. Save and close the file.
4. Open the `web.xml` file from the `<SPECROOT>/tomcat/webapps/spectrum/WEB-INF/` directory.
5. Change the value of the `<auth-method></auth-method>` parameter from WSFED to BASIC.
6. Save and close the file.
7. Restart the Tomcat server.

Integrating with CA Embedded Entitlements Manager

About the DX NetOps Spectrum Integration with CA Embedded Entitlements Manager

Single Sign-On is a separate component that uses the CA Embedded Entitlements Manager (CA EEM) solution. It is not configured as part of the DX NetOps Spectrum installation. Instead, you must activate it after you install DX NetOps Spectrum by modifying configuration settings from the Administration pages on the OneClick server.

Intended Audience

The How to Configure DX NetOps Spectrum/CA EEM Integration section is intended for administrators who want to set up authorization access for DX NetOps Spectrum with CA EEM. The DX NetOps Spectrum integration with CA EEM addresses the need for fine-grained access control to DX NetOps Spectrum enterprise applications and other applications integrated with DX NetOps Spectrum.

Before using this section to integrate DX NetOps Spectrum with CA EEM, you should have knowledge about DX NetOps Spectrum user management and you should be familiar with the OneClick Administration pages. No special knowledge of CA EEM is required to specify integration parameters from the DX NetOps Spectrum environment. However, some

knowledge of user management such as user creation and integration with LDAP (for registering DX NetOps Spectrum users in CA EEM) is required in the CA EEM environment.

How to Configure the Integration

Follow this process to configure the integration of DX NetOps Spectrum and CA EEM:

1. Install CA EEM.

   NOTE
   For more information, see the *CA EEM* documentation. We recommend that you install CA EEM on a separate machine.
2. Register DX NetOps Spectrum users with CA EEM.
3. Configure the CA EEM server for single sign-on in OneClick:
   a. Configure CA EEM server connection parameters.
   b. Test connectivity to the CA EEM server.
   c. Save connection parameters settings.

Register DX NetOps Spectrum Users with CA EEM

After installing CA EEM, register DX NetOps Spectrum users in CA EEM before you configure Single Sign-On (SSO) settings for them.

Follow these steps:

1. Open a browser and navigate to the CA EEM home page.
2. Log in as the CA EEM administrator.
   Default: EiamAdmin.
3. Click the Manage Identities tab.
4. Click the New User icon in the Users panel.
5. Add the DX NetOps Spectrum user to the CA EEM system by supplying values in the fields provided.

   NOTE
   You can point CA EEM to an LDAP or Active Directory server from the Configure tab. CA EEM 12.0 and later can point to more than one LDAP or Active Directory server. For more information, see Support for Multiple Active Directory Domains.

Create DX NetOps Spectrum Users for Single Sign-On

Creating user accounts in DX NetOps Spectrum requires entering a password manually. This step may not be possible if DX NetOps Spectrum is integrated with CA Embedded Entitlements Manager for Single Sign-On. But if DX NetOps Spectrum is integrated with an LDAP server, you can create users with blank passwords. The LDAP server then handles user authentication.

NOTE

Integrating DX NetOps Spectrum with LDAP is not mandatory. However, the integration simplifies user creation with blank passwords in DX NetOps Spectrum.

If your DX NetOps Spectrum deployment requires you to use SSO through CA EEM, and if EEM is integrated with an LDAP server, configure the LDAP server in DX NetOps Spectrum, create the users in DX NetOps Spectrum and then configure SSO settings to integrate with CA EEM.

NOTE

- Configure EEM with the LDAP server first, then configure DX NetOps Spectrum, create user accounts and integrate with CA EEM.
- When CA EEM points to an AD for SSO, DX NetOps Spectrum supports only the user accounts in that AD. Do not use group names for SSO.

Follow these steps:

1. Log in to the OneClick Web Console.
2. Configure the Web Console with your LDAP server as described in the OneClick Administration Pages section.
3. Launch the OneClick Console.
4. In the Users tab of the Navigation panel, select the top-level Users node and click Creates a New User.
   The Create User dialog opens.
5. Create a user account, but leave the password blank. Save the user account.
6. Repeat the previous steps to create all required DX NetOps Spectrum users.
7. Close the OneClick Console.
8. In the OneClick Web interface, click the Administration tab.
9. Click the Single Sign-On Configuration link.
10. Complete DX NetOps Spectrum configuration with EEM.
11. Restart the Tomcat server on the OneClick Web server host for the changes to take effect.

Now LDAP users can log in to DX NetOps Spectrum and can be authenticated using the EEM Server.

Configure OneClick to Connect to the CA EEM Server

Follow these steps:

1. Log in to the OneClick web server.
2. Click Administration in the menu bar on the OneClick home page.
   The system verifies your administrator credentials.
3. Click Single Sign-On Configuration in the Administration Pages panel on the left.
   The Single Sign-On Configuration page opens.
4. Select CA EEM in the Choose SSO Option section.
   The CA EEM Single Sign-On Configuration section opens.
5. Specify the following parameters for connecting with the CA EEM server in the CA EEM Server Connectivity section:
   - **CA EEM Server Hostname**
     Specifies the host name of the CA EEM server you want to connect to.
   - **OneClick Server Domain Name**
     Specifies the domain where the OneClick server resides (for example, ca.com).

     NOTE
     If you are trying to inter-operate between eHealth and DX NetOps Spectrum using CA EEM or CA SiteMinder®, a second-level domain or greater is required for the cookie domain.

     Cookies are restricted to a certain domain level for security reasons. According to "RFC 2901" and "RFC 2965", cookies cannot be set to a top-level domain (such as .com, .org, .gov). A minimum of second-level domain is required. For more information, consult the RFC documentation.
     If a domain name ends with a two letter country code, a minimum of a third-level domain is required. A cookie that is set to a second-level domain is visible at all of its third-level domains. However, a cookie that is set to a third-level domain is not visible at its parent second-level domain or at other sub domains. If no domain name is specified when a cookie is written, the cookie domain attribute defaults to the domain name where the application resides.
   - **Spectrum Application Name in CA EEM**
     Specifies the name of the DX NetOps Spectrum application in CA EEM, enabling you to set up rules in CA EEM. Enter *spectrum* in this field. This is not a mandatory field.
   - **Proxy URL**

Specifies the URL to be used for proxy connectivity.

6. Select Yes in the Authentication Logging section to enable logging to either the Tomcat log or to a specified log location for debugging connectivity issues.
7. (Optional) Perform the following steps to test the configuration:
   a. Complete the Test Username field and the Test Password field with appropriate credentials for testing the connection to the CA EEM server.
   b. Click Test.

   The OneClick Console notifies you when proper authentication occurs.
8. Click Save.
   DX NetOps Spectrum verifies whether the CA EEM single sign-on conflicts with any other SiteMinder single sign-on option. If a conflict is detected, you see an error. Otherwise, a dialog asks you to restart the web server.
9. Click OK.
10. The system saves the information to an eem-sso.conf configuration file in the Tomcat directory. You can find the file at the following location: *$SPECROOT*/custom/sso/eem-sso.conf
11. Restart Tomcat to let the changes take effect.
    OneClick is configured to connect to CA EEM server.

Support for Multiple Active Directory Domains

DX NetOps Spectrum leverages the multiple Active Directory domains feature of CA EEM to authorize users from multiple domains. CA EEM 12.0 and later versions support this feature.

For example, User1 and User2 are two users belonging to different domains, "Domain1.com" and "Domain2.com" respectively. With this feature, user1 and user2 can connect to OneClick. When you create these users in OneClick, supply user names that are identical to the principal names of these users in CA EEM.

- If the principal name of "User1" of "Domain1.com" in CA EEM is "Domain1.com\User1", in DX NetOps Spectrum create this user as "Domain1.com\User1".
- If the principal name of "User2" of "Domain2.com" in CA EEM is "Domain2.com\User2", in DX NetOps Spectrum create this user as "Domain2.com\User2".

Two users with the same user name can exist in more than one domain. Include the domain prefix for such identical users during the OneClick authentication. For example, supply "Domain1.com\User1" and "Domain2.com\User1". When a user with the same user name is not present across domains, during the OneClick authentication domain prefix before the user name is not mandatory.

> **NOTE**
> Do not use the "User@Domain" format to configure principal name in CA EEM, DX NetOps Spectrum supports only the "Domain\User" format.

As a result, when User1 and User2 of two different domains try to access the OneClick, DX NetOps Spectrum sends an authentication request to CA EEM. When CA EEM successfully authenticates these two users by resolving its actual domain, DX NetOps Spectrum authorizes them to access the SpectroSERVER.

CA EEM 12.0 supports the following two configuration types to enable the multiple Active Directory domains feature:

- Active Directory Domain
- Active Directory Forest

For more information about configuring multiple domains in CA EEM, see the CA Embedded Entitlements Manager - Implementation Release 12.51.

How to Disable the CA EEM Integration

To disable the DX NetOps Spectrum and CA EEM Integration, follow this process:

1. Close all the OneClick client consoles.
2. Stop the One Click Tomcat Web Server.
   Windows: Open the Services view and Stop the Spectrum Tomcat service.
   Linux: Run the stopTomcat.sh command from the $SPECROOT/tomcat/bin directory.
3. In the One Click web server, navigate to the $SPECROOT/custom/ directory and rename the sso directory to sso.bak.
4. Edit the web.xml in the $SPECROOT/tomcat/webapps/spectrum/WEB-INF directory:
   a. Change the following entry:
   ```
   <login-config>
   <auth-method> EXTERNALSSO </auth-method>
   <realm-name>SPECTRUM</realm-name>
   </login-config>
   ```
   To become:
   ```
   <login-config>
   <auth-method> BASIC </auth-method>
   <realm-name>SPECTRUM</realm-name>
   </login-config>
   ```
   b. Comment out the following entry that should show at the top of the file:
   ```
   <listener>
   <listener-class>com.aprisma.tomcat.authenticator.ExternalSSOAuth</listener -class>
   </listener>
   ```
   To make it:
   ```
   <!--
   <listener>
   <listener-class>com.aprisma.tomcat.authenticator.ExternalSSOAuth</listener-class>
   </listener>
   -->
   ```
   c. Save the changes made to the $SPECROOT/tomcat/webapps/spectrum/WEB-INF/web.xml file.
5. Go to the $SPECROOT/tomcat/conf/context.xml file.
   a. Add the following line:
   <Valve className="org.apache.catalina.authenticator.BasicAuthenticator" changeSessionIdOnAuthentication="false" /
   ></Context>
   b. Comment out the following line:
   <Valve className="com.aprisma.tomcat.authenticator.ExternalSSOAuth" changeSessionIdOnAuthentication="false" /
   ></Context>
   So that it looks like this:
   <!--
   <Valve className="com.aprisma.tomcat.authenticator.ExternalSSOAuth" changeSessionIdOnAuthentication="false" /
   ></Context>
   -->
6. Save the file changes to the $SPECROOT/tomcat/conf/context.xml file.
7. Start the One Click Tomcat Web Server and attempt to log in with a non LDAP account.

Troubleshooting Integration with CA EEM

This chapter describes how to respond to potential authentication problems with the DX NetOps Spectrum CA EEM integration.

Cannot log in to OneClick web page after CA EEM 12.51 integration

Symptom:

After upgrading to 10.2.1 and enabling the CA EEM 12.51 integration, unable to log in to OneClick web page, keep asking for username and password.

The following error was found in the $SPECROOT/tomcat/logs/sdtout.log file:

<date and time> - SPC-OCA-10486: Error: DX NetOps Spectrum user model not found for user <domain>\<user>.

Example: SPC-OCA-10486: Error: DX NetOps Spectrum user model not found for user ad1.ca.com\user1

The following error is found in the CA EEM debug log:

[2017-07-28 10:14:55.877] [EEMSSOContext::authenticateWithPassword] EEM password authentication successful. [/spectrum/][<user>]

[2017-07-28 10:14:55.877] [EEMSSOContext::authenticateWithPassword] [/spectrum/][<user>] [UserSession;Version-1.0;dd71c8d592f2115a37d1f7826a619a86-597b1913-cc82b00-4] returned.

[2017-07-28 10:14:55.877] [ExternalSSOAuth::authenticate] Successfully authenticated user/pass with SSO Server ["GET /spectrum/"]["0:0:0:0:0:0:0:1"]["<user>"][authenticated="true"][authorized="false"]

[2017-07-28 10:14:55.877] [ExternalSSOAuth::authenticate] usernameFromToken <domain>\<user>

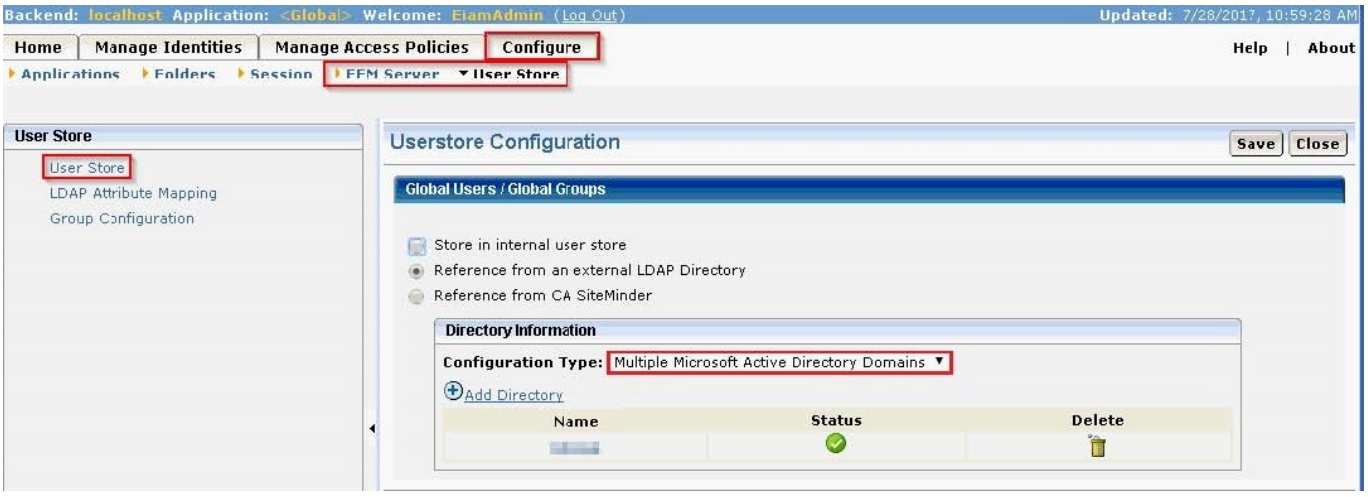
[2017-07-28 10:14:55.892] [ExternalSSOAuth::authenticate] Could not authorize request for resource. ["GET /spectrum/"] ["0:0:0:0:0:0:0:1"]["<domain>\<user>"][authenticated="true"][authorized="false"]

Environment:

CA Spectrum 10.2.x, 10.1.x and CA EEM 12.51

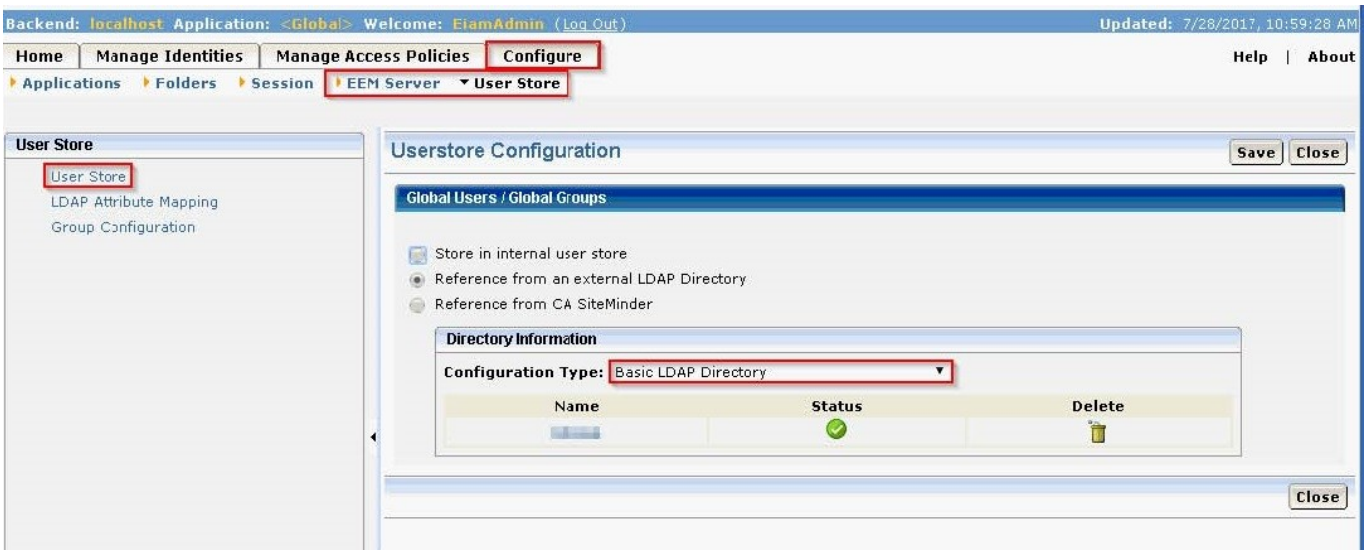
Probable Cause:

If you have configured the CA EEM for Multiple Microsoft Active Directory Domains, but you have only one Active Directory. The CA EEM is appending the domain in the account name, like: ad1.ca.com\user1, instead of only user1.



Resolution:

To resolve this issue, delete the setting for Multiple Microsoft Active Directory Domains and configure for Basic LDAP Directory.



The login is success.

Cannot Log In to DX NetOps Spectrum

Symptom:

User unable to authenticate to DX NetOps Spectrum.

Solution:

Verify that the user name and password have been entered correctly in DX NetOps Spectrum.

Configuration Test Fails

Symptom:

Unable to authenticate with CA EEM though the DX NetOps Spectrum configuration test.

Solution:

Verify that the proper user name, password, server name, and port have been entered.

Managing Client Applications

This section describes how to use OneClick, Alarm Notifier, SANM, Report Manager, and so on.

AlarmNotifier

This section discusses the AlarmNotifier Application, and how it is used for alarm notifications. This section also discusses how alarm notifications can be customized for various requirements of the administrators and troubleshooters.

About AlarmNotifier

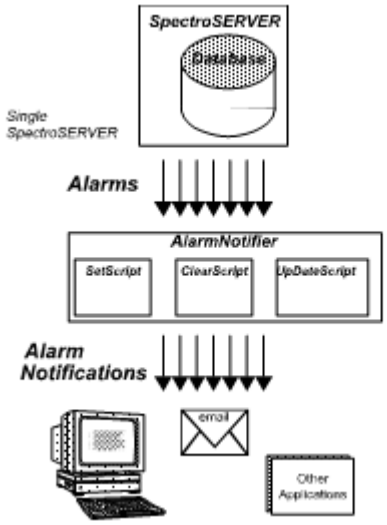
AlarmNotifier is a SpectroSERVER-client application that installs with core DX NetOps Spectrum components. The AlarmNotifier application connects to a single SpectroSERVER and invokes scripts that provide notifications about DX NetOps Spectrum alarm status.

Start AlarmNotifier from a terminal shell command prompt. Once started, it continuously displays output from scripts that are invoked whenever alarms are either set, cleared, or updated. AlarmNotifier provides the following features for DX NetOps Spectrum:

- Single SpectroSERVER alarm monitoring.
- Three scripts that generate alarm information: SetScript, ClearScript, and UpdateScript. These scripts contain settings that can be customized for your environment.
- Resource file parameters that can be configured to modify AlarmNotifier operational features.

Alarm Monitoring Process

AlarmNotifier supplements DX NetOps Spectrum alarm monitoring and notification features. The following diagram illustrates the relationship between AlarmNotifier and DX NetOps Spectrum:



DX NetOps Spectrum performs some alarm functions, while AlarmNotifier performs others. DX NetOps Spectrum polls the modeled network elements and updates the status information about each element that is stored in the SpectroSERVER database.

DX NetOps Spectrum generates an alarm when it receives a trap from the network or when it detects a critical status change in a network-element model. In the OneClick Topology view, the condition of the model icon changes from green to another color to indicate alarm severity. DX NetOps Spectrum posts information about the alarm in the Alarms tab. Event information for the alarm appears in the Events tab in the OneClick Contents panel.

When AlarmNotifier is started, it registers with DX NetOps Spectrum. Then a model named AlarmNotifier of type ClientApp is created. This model is not visible in any of the DX NetOps Spectrum Topology views. However, you can see it in the Events tab. The Events tab displays information such as the application start and stop time for this model.

AlarmNotifier queries the SpectroSERVER and requests information about existing alarms. AlarmNotifier runs scripts and generates notifications about existing alarms.

Each time an alarm is set, cleared, or updated, AlarmNotifier receives information from the SpectroSERVER and invokes the relevant script. AlarmNotifier scripts can initiate email notifications of alarms that are sent to network personnel. They can also transmit alarm information to third-party applications.

Spectrum Alarm Notification Manager (SANM)

Spectrum Alarm Notification Manager (SANM) is an add-on component for DX NetOps Spectrum that can enhance AlarmNotifier features. The SANM Policy Administrator lets you create multiple alarm-filtering policies, which you can associate with uniquely named instances of AlarmNotifier applications. Use these policies to instruct AlarmNotifier to generate notifications only for the alarms that you consider relevant.
The SANM Policy Administrator lets you associate policies with AlarmNotifier applications as required. You can also automate this association process using the Scheduler utility.
With SANM installed, AlarmNotifier offers the following capabilities:

- Distributed SpectroSERVER alarm monitoring
- Additional script parameters that provide more alarm information
- Commands for acknowledging and clearing alarms
- Additional startup options that let you log AlarmNotifier activities and concurrently run multiple instances of AlarmNotifier
- SANM alarm-filtering tools

NOTE

For more information about SANM, see the Spectrum Alarm Notification Manager (SANM) section.

Operating AlarmNotifier

This section discusses how you can operate AlarmNotifier. Operating AlarmNotifier involves starting, viewing alarm notifications, update script parameters, and stopping AlarmNotifier.

Start AlarmNotifier Application

AlarmNotifier is located in the <*$SPECROOT*>/Notifier directory. This directory contains the following files by default:

- .alarmrc
- AlarmNotifier
- ClearScript
- README
- SetScript
- UpdateScript

AlarmNotifier includes the following additional files and directory:

- **AlarmAck**
  Acknowledges an alarm.
- **AlarmClear**
  Clears an alarm.
- **Trace**
  Displays trace files.

To start AlarmNotifier, use the following AlarmNotifier command in the <*$SPECROOT*>/Notifier directory:

```
AlarmNotifier [-r resourcefile] [-n application][-tl summary|details [-tn tracefile] [-ts size]]
```

- **-r *resourcefile***

Lets you specify a resource file other than the default resource file .alarmrc.

- **-n *application***
  Lets you override the application name value that is specified by the APPLICATION parameter in the resource file. You can specify a different name for an AlarmNotifier application instance. This option lets you start multiple instances of AlarmNotifier and associate each of them with a different SANM alarm-filtering policy. If a name is not assigned to the APPLICATION parameter in the resource file, use the -n option at start-up to specify an application name.
- **-tl summary | details**
  Lets you activate tracing at a specified level, summary or detailed. The default format for an AlarmNotifier trace file is the application name together with the date when the trace file was created.
- **-tn *tracefile***
  Lets you specify a trace file name other than the default name, which is provided when only the -tl option is used. Use this option with the -tl option.
  When using the trace file option, the output file is written by default to the <*$SPECROOT*>/Notifier/trace directory. To explicitly name an output file and path, use the [-tn filename] option. If <*filename*> is a relative path, trace output is written to a file that is relative to the current directory. If <*filename*> is an absolute path, trace output is written to the absolute path.
- **-ts *size***
  Lets you specify the number of lines in the trace file. Use this option with the -tl option. The application writes this number of lines to the file and then wraps around to the beginning of the file. Entries are numbered sequentially, and an END OF TRACE line follows the last entry. The default number of lines in a trace file is 10000.

NOTE

If you migrate DX NetOps Spectrum to a higher release, verify the following settings in the ".alamrc" file:

- The paths to the scripts are correct.
- You have all the permissions to the "ClearScript" script.

AlarmNotifier does not start if these settings are incorrect.

Stop AlarmNotifier

Once started, the AlarmNotifier runs continuously. To stop the AlarmNotifier, use the following commands:

- To stop the AlarmNotifier running in the foreground for Windows, run the command:
  ```
  Control-C
  ```
- To stop the AlarmNotifier running in the background for Windows, run the command:
  ```
  $SPECROOT/lib/SDPM/kill - TERM <pid>
  ```

Start AlarmNotifier with the Process Daemon

You can automate the startup of AlarmNotifier processes using processd, the DX NetOps Spectrum process management daemon.

The processd automatically starts during DX NetOps Spectrum installation and whenever the system restarts. Once processd is started, it automatically starts and manages other processes via *.idb ticket files that are located in the **$SPECROOT/lib/SDPM/partslist** directory. For the AlarmNotifier process, processd uses the AlarmNotifier.idb file.

To enable processd to launch and track the AlarmNotifier application, edit the **$SPECROOT/lib/SDPM/partslist/AlarmNotifier.idb** file and change AUTOBOOTSTART to Y.

Follow these steps:

1. Using a text editor, edit the **$SPECROOT/lib/SDPM/partslist/AlarmNotifier.idb** file and change AUTOBOOTSTART to Y:

```
# Processd Install Ticket for Alarm Notifier
PARTNAME;ALARMNOTIFIER;
APPNAME;Alarm Notifier;
WORKPATH;$SPECROOT/Notifier;
LOGNAMEPATH;$WORKPATH/ALARMNOTIFY.OUT;
ADMINPRIVS;y;
AUTORESTART;y;
AUTOBOOTSTART;y;
#STATEBASED;N;
NUMPROCS;1;
RETRYTIMEOUT;6000;
TICKETUSER;<USERNAME>;
RETRYMAX;20;
STARTPRIORITY;30;
#ENV;<var>=<value>;
ARGV;$WORKPATH/AlarmNotifier<CSEXE>; //
```

2. The LOGNAMEPATH parameter specifies the name and path of the log file for the AlarmNotifier application.

   NOTE
   Each time AlarmNotifier starts, a new log file is generated, and a backup of the previous log file is created. However, the SpectroSERVER only stores the two most recent AlarmNotifier log files. To keep more Notifier log files, add this line after the LOGNAMEPATH entry, **LOGBACKUPMAX;10; This will save 10 copies of your Notifier Log file.**

   The value of the TICKETUSER parameter, <USERNAME>, must be the username of a valid DX NetOps Spectrum User Model.
   The SpectroSERVER must be running before AlarmNotifier starts. Therefore, the STARTPRIORITY parameter can be set to 30, indicating that AlarmNotifier is dependent on the SpectroSERVER. For more information, see the Distributed SpectroSERVER Administration section.

Run Multiple AlarmNotifiers

Multiple AlarmNotifiers with different notification policies can be started when DX NetOps Spectrum processd starts. You can enable this setup by creating install ticket files for each new AlarmNotifiers. Perform the following tasks to run multiple AlarmNotifiers:

- Make a copy of the default **$SPECROOT/lib/SDPM/partslist/ALARMNOTIFIER.idb** file and rename the files accordingly. For more information, see Start AlarmNotifier with the Process Daemon. For example: ALARMNOTIFIER_OPERATORS.idb
- Make a copy of the **$SPECROOT/Notifier/.alarmrc** and keep the naming consistent: .alarmrc_operators
- After the install ticket files are created, you can change the values of the install ticket file to start an AlarmNotifier with a preferred notification policy. You can provide the following values:
  - **PARTNAME**

    Identifies a particular process/application with a multi-character string with no spaces. The install ticket for this application has a filename in the form *<PARTNAME>*.idb. *<PARTNAME>* is the process name. For example, if you named the idb file ALARMNOTIFIER_OPERATORS.idb, set this to: PARTNAME;ALARMNOTIFIER_OPERATORS;
  - **APPNAME**

Specifies the name of the application that is created to implement the notification policy. To run the new instance of AlarmNotifier, edit your copied idb file and change this value to the name you would like to see in a process listing. For example: AlarmNotifier_Operators. For more information, see the Spectrum Alarm Notification Manager (SANM) section.

– **AUTOBOOTSTART**

Notifies the process daemon to start a process when processd starts. Set the value to Y.

– **WORKPATH**

Specifies the working path where the application can be found. Leave this set to **$SPECROOT/Notifier;**

– **LOGNAMEPATH**

Specifies the name and path of the log file for the AlarmNotifier application. Change the name of the log to match the APPNAME. For example: **LOGNAMEPATH;$WORKPATH/AlarmNotifier_Operators.OUT;**

NOTE
LOGNAMEPATH must be unique for each instance.

– **ARGV**

Specify the following value using your custom APPNAME:

```
$SPECROOT/Notifier/AlarmNotifier<CSEXE> -r .alarmrc_operators -n <name of the application that you
 specified for APPNAME>
```

For example:

```
$SPECROOT/Notifier/AlarmNotifier<CSEXE> -r .alarmrc_operators -n
 AlarmNotifier_Operators
```

NOTE
Do not copy the AlarmNotifier application and rename it as the copied executable will not be upgraded when you upgrade Spectrum. The best practice is to use the default AlarmNotifier application and provide the new name in both the APPNAME variable and the ARGV variable in your .idb file. This ensures that the AlarmNotifier version will be consistent with the version of Spectrum that you are running.

AlarmNotifier Output

AlarmNotifier invokes the SetScript, ClearScript, or UpdateScript whenever AlarmNotifier detects an alarm that is set, cleared, or updated in DX NetOps Spectrum. Each script generates a notification containing information about alarm status (set, cleared, or updated) and displays it. Each notification contains the parameters that are defined in Script Parameter Definitions.

- **SetScript**
  Invoked for an alarm in the following situations:
  – AlarmNotifier is started and detects an existing alarm. AlarmNotifier invokes SetScript unless the value of GET_EXISTING_ALARMS is set to 'false' in the .alarmrc resource file.
  Default: true.
  – DX NetOps Spectrum generates an alarm while AlarmNotifier is running.
- **ClearScript**
  Invoked when an alarm is cleared.
- **UpdateScript**
  Invoked when an alarm is updated. An alarm is defined as updated in these situations:

- A troubleshooter has been assigned to an alarm, or the troubleshooter name has been changed. The RepairPerson parameter in the scripts represents this troubleshooter name.
- The status of an alarm has changed. Status information for an alarm is entered in the Alarms tab. The AlarmStatus parameter represents status in the scripts.
- An alarm is acknowledged or unacknowledged in DX NetOps Spectrum.
- A new event, or a change to an existing event, occurs on a device that is in an alarm state.

Script Parameter Definitions

You can update the parameters of the AlarmNotifier scripts to customize their functionality. The following list describes the parameters that are available for SetScript, ClearScript, and UpdateScript.

> **NOTE**
> Additional script parameters are available when you use SANM with AlarmNotifier. For more information, see the Spectrum Alarm Notification Manager (SANM) section.

- **Date**
  Specifies the date when AlarmNotifier detects that the alarm is set, updated, or cleared.
  Format: mm/dd/yyyy
- **Time**
  Specifies the time when AlarmNotifier detects that the alarm is set, updated, or cleared.
  Format: hh:mm:ss
- **Mtype**
  Specifies the type of model for which the alarm is set, updated, or cleared.
- **ModelName**
  Specifies the name of the model whose alarm is set, updated, or cleared. If the ModelName contains special characters, pass it to the script as an environment variable to avoid errors. A special character is a character that the command shell interprets as having special meaning such as '$' or '*'.
  To pass the model name as an environment variable, add the attribute ID for ModelName (0x1006e) to the values for the EXTRA_ATTRS_AS_ENVVARS parameter in the .alarmrc file. For more information, see Passing DX NetOps Spectrum Attributes to Scripts and .alarmrc Parameters.
- **AlarmID**
  Specifies the numeric identifier that DX NetOps Spectrum assigned to the alarm.
- **Global AlarmID**
  Specifies a unique numeric identifier that DX NetOps Spectrum assigns to the alarm. Unlike the AlarmID, the global alarm ID is not only unique within the DX NetOps Spectrum environment, but also can be passed as a unique identifier to other environments. Use this value to pass a unique identifier to third-party software. By default, Global AlarmID is commented out in each of the AlarmNotifier scripts. Remove the comment mark (#) to pass this parameter.
- **Severity**
  Specifies the DX NetOps Spectrum severity-level code for the alarm: Critical (Red), Major (Orange), Minor (Yellow), Maintenance (Brown), Suppressed (Gray), or Initial (Blue).
- **ProbableCauseID**
  Specifies the hexadecimal identifier that is associated with the probable cause for the alarm.
- **RepairPerson**
  Specifies the troubleshooter who is assigned to the alarm in the Alarms tab. AlarmNotifier invokes the UpdateScript whenever a troubleshooter is first assigned and each time thereafter.
  The following circumstances determine whether the name of a troubleshooter (a repair person) appears in the notifications that SetScript and ClearScript generate:

  - If a troubleshooter is assigned after AlarmNotifier detects that an alarm has been set, SetScript does not display a name. The UpdateScript and the ClearScript do display a name.
  - If a troubleshooter is assigned before AlarmNotifier detects the set (for an alarm that exists before AlarmNotifier is started), all three scripts display the name of the troubleshooter.

    NOTE
    For more information, see the Using OneClick section.
- **AlarmStatus**
  Indicates the status information for the alarm in OneClick. AlarmNotifier invokes UpdateScript whenever status information is first entered and each time thereafter. Status information typically appears in the notifications that the SetScript and ClearScript generate. The following circumstances are exceptions:
  - If status information is entered after AlarmNotifier detects that an alarm has been set, SetScript does not display the information. The UpdateScript and the ClearScript do display the information.
  - If status information is entered before AlarmNotifier detects the set (for an alarm that exists before AlarmNotifier is started), all three scripts display the status information.

    NOTE
    For more information, see the Using OneClick section.
- **SpectroSERVER**
  Specifies the name of the host for the SpectroSERVER where the alarm has been set, updated, or cleared.
- **Landscape**
  Specifies the handle for the landscape from which the alarm has been set, updated, or cleared.
- **ModelHandle**
  Specifies the handle of the model for which the alarm has been generated.
- **ModelTypeHandle**
  Specifies the handle of the model type for which the alarm has been set, updated, or cleared.
- **IPAddress**
  Specifies the IP address of the network element for which the alarm has been set, updated, or cleared.
- **SecurityString**
  Specifies the security string of the model for which the alarm has been set, updated, or cleared.
- **AlarmState**
  Specifies whether the alarm state is "Existing" or "New."
  The state of an alarm is "Existing" if the alarm is set before AlarmNotifier is started. AlarmNotifier invokes SetScript for an existing alarm if the GET_EXISTING_ALARMS parameter in the .alarmrc resource file is set to true.
  The state of an alarm is "New" if the alarm is generated after AlarmNotifier is started. The alarm state is also "New" when the SpectroSERVER for the AlarmNotifier restores connections to a previously connected landscape where the alarm occurred.
- **Acknowledged**
  Specifies whether the alarm has been acknowledged.
- **UserClearable**
  Specifies whether a user can clear the alarm.
- **DeviceType**
  Specifies the value of the DeviceType attribute on the model for which the alarm has been set, updated, or cleared. For more information, see the Certifications section.
- **Raw Alarm Time**
  The Alarm date and time is available as $DATE and $TIME. To get the unformatted alarm time, you can reference $RAW_ALARM_TIME.

    NOTE
    The unformatted alarm time is the number of seconds that have elapsed since midnight UTC of January 1, 1970.

Persistent and Stale Alarms

When the SpectroSERVER stops and restarts, alarms that were already present continue to exist. These alarms are "persistent" alarms. The persistent alarm feature lets DX NetOps Spectrum retain alarm-related information such as troubleshooter assignments and status when the SpectroSERVER shuts down.

In some cases, the underlying cause of an alarm is resolved between the time that the SpectroSERVER shuts down and restarts. The alarm still appears in the Alarms list in OneClick but is considered to be stale. You can clear all stale alarms (which are also known as "residual alarms") manually.

However, stale alarm information is not forwarded to AlarmNotifier by the SpectroSERVER. Instead, a single *new* alarm that indicates that stale or residual alarms exist on the landscape is generated and sent to AlarmNotifier. When you manually clear a stale alarm, that alarm is also cleared in AlarmNotifier. When the final stale alarm is cleared, a "clear" is issued for the stale alarm notification.

Customizing AlarmNotifier

This section discusses how you can customize AlarmNotifier. Customizing AlarmNotifier involves the following things:

- Modifying scripts
- Limiting script outputs
- Passing DX NetOps Spectrum attributes to scripts
- Sending data to third-party applications
- Customizing the ".alarmrc" resource file

Modifying Scripts

You can modify AlarmNotifier scripts to customize AlarmNotifier actions and output. You can configure the scripts to initiate email notifications to specified recipients. You can also customize scripts to limit the range of information that alarm transition notifications provide, or to integrate with a third-party application.

Enable Email Notifications in a Script

Each AlarmNotifier script includes two parameters (SENDMAIL and VARFORMAIL) that you can configure to enable email notifications to the troubleshooter for an alarm. You can enable email notifications for one or more scripts.

Note: To preserve default script configuration settings in case of accidental loss, make a backup copy of the default script that you plan to edit.

Follow these steps:

1. Navigate to the <*$SPECROOT*>/Notifier directory, or to the directory where the script is saved.
2. Open the script with a text editor.

   NOTE
   All of the scripts are executed serially. Therefore, you can edit a script without stopping the AlarmNotifier.
3. Set the SENDMAIL parameter in the script to True.
4. Set the VARFORMAIL parameter to RepairPerson.
5. Save and close the script.
   Email is sent to the troubleshooter who is assigned to the alarm in the Alarms tab. This person must be an authorized user of (or user model in) the landscape where the alarm originates.

   NOTE
   The value for RepairPerson is established after the alarm has occurred. As a result, mail cannot be sent in response to a set action (using the SetScript). However, if you are also using SANM, you can configure mail

to be sent as a result of a set action. For more information, see the Spectrum Alarm Notification Manager (SANM) section.

Mail Service on Windows Platform

On Windows, use the mail command to enable the mail service so that AlarmNotifier scripts can send email notifications.

Take this step before AlarmNotifier starts. Otherwise, an error message prompts you to configure the mail service and then start AlarmNotifier.

Mail Command Parameters

Several parameters are required when running the mail command from a terminal window. These parameters are as follows:

- **-m**
  Is the Return host name. The Return host is the computer where incoming mail is received.
- **-h**
  Is the Simple Mail Transfer Protocol (SMTP) host, the computer where outgoing email is sent to be processed.
- **-u**
  Is the username.

  NOTE
  AlarmNotifier can send notifications to a pager. We recommend first configuring AlarmNotifier to send notifications to a valid local mail account to test this configuration. You can then reconfigure the Mail Service to send the notifications to the pager.

Configure Mail Service

You can set up the mail service on Windows so that AlarmNotifier can send email notifications.

Follow these steps:

1. Consult with your mail server administrator to verify the correct values of the mail command parameters.
2. Open the bash shell.
3. Enter the following command:
   ```
   mail -m your-company.com -h smtp.your-company.com -u username
   ```
   The command usage list appears once the command has completed successfully.
4. To verify that the configuration is complete, view the registry entries for HKEY_CURRENT_USER/Software/SMail. The hostname, smtphost, and username keys now contain the information that you included in the command string.

Limiting Script Output

You can comment out AlarmNotifier script parameters to reduce the amount of output from a script.

NOTE
Before you modify a script, review the parameter descriptions in Script Parameter Definitions. We recommend a thorough understanding of the information that you are suppressing.

Follow these steps:

1. Close the instance of the AlarmNotifier application that you want to configure.
2. Navigate to the default script directory, <$SPECROOT>/Notifier, or the directory where the script that you want to edit is saved.
3. Open the script with a text editor.

4. Comment out the echo command lines that you want to suppress by typing a pound sign (#) at the beginning of each line. In the following example, the UserClearable parameter is commented out:

```
echo "SecurityString: " $SECSTR
echo "AlarmState: " $ALARMSTATE
echo "Acknowledged: " $ACKD
#echo "UserClearable: "  $CLEARABLE
```

   When this script generates notifications, the information from the parameter that has been commented out does not appear on the screen.
5. Save and close the script.

> **NOTE**
> Do *not* comment out or modify the assignments of the variables themselves or the shift commands. The script does not display alarm information properly if you change these lines.

Passing Product Attributes to Scripts

Attributes of a model with an alarm can be passed to AlarmNotifier. Model attributes can be used as parameters in SetScript, ClearScript, or UpdateScript. Use the .alarmrc parameters EXTRA_ATTRS_AS_ENVVARS or EXTRA_ATTRS_AS_ARGS to pass in attributes. To enable these parameters, set the USE_NEW_INTERFACE .alarmrc parameter to TRUE.

EXTRA_ATTRS_AS_ENVVARS passes attributes to AlarmNotifier as environment variables. EXTRA_ATTRS_AS_ARGS passes attributes as command-line arguments. For most attributes, either of these mechanisms can be used. However, EXTRA_ATTRS_AS_ENVVARS is required in cases where new lines or special characters can cause problems for the script that parses the extra data. When USE_NEW_INTERFACE=TRUE, the environment variable mechanism is used to pass $STATUS, $EVENTMSG, and $PCAUSE to avoid this problem.

In the .alarmrc file, set the appropriate parameter equal to the DX NetOps Spectrum attribute IDs that you want to pass. You can reference the attribute ID either in hexadecimal or decimal notation.

If you pass an attribute as an environment variable using EXTRA_ATTRS_AS_ENVVARS, you reference this variable in a script using the following syntax:

```
$SANM_<attribute_ID>
```

- **<attribute_ID>**
  Specifies the attribute ID of the attribute you are referencing. If you have used hexadecimal notation to call this attribute in the .alarmrc file, hexadecimal notation is also required in the script. If you have used decimal notation to call this attribute in the .alarmrc file, use decimal notation in the script.

  > **NOTE**
  > Windows automatically sets environment variables to uppercase. Therefore, when you reference these variables, use the uppercase format, such as $SANM_0X100C5.

If you use EXTRA_ATTRS_AS_ARGS to pass an attribute as an argument, you can reference this variable in a script by assigning the value to a variable within the script:

```
<variable>=$<x>
```

- **<variable>**
  Specifies the variable that holds the value of the attribute.
- **<x>**
  Specifies the appropriate variable number for the order and number of arguments that you have passed.

Example

The following example shows four sample DX NetOps Spectrum attributes that are passed to AlarmNotifier in the .alarmrc file. The attributes are then referenced in a script.

.alarmrc File Reference

```
USE_NEW_INTERFACE=TRUE
EXTRA_ATTRS_AS_ENVVARS=0x100c5,0x11f84
EXTRA_ATTRS_AS_ARGS=0x110df,0x117dc
```

Script Reference

```
#These lines read 0x110df and 0x117dc into the variables MAC_ADDRESS
#and FIRMWARE_VERSION respectively.
shift 9
MAC_ADDRESS=$1
FIRMWARE_VERSION=$2
#These lines print out the value of each attribute.
echo "The value of attribute 0x100c5 is: " $SANM_0x100c5
echo "The value of attribute 0x11f84 is: " $SANM_0x11f84
echo "The value of attribute 0x110df is: " $MAC_ADDRESS
echo "The value of attribute 0x117dc is: " $FIRMWARE_VERSION
#These lines print out the value of each attribute.(Windows Platform)
#references to environmental variables are in uppercase
echo "The value of attribute 0x100c5 is: " $SANM_0X100C5
echo "The value of attribute 0x11f84 is: " $SANM_0X11F84
echo "The value of attribute 0x110df is: " $MAC_ADDRESS
echo "The value of attribute 0x117dc is: " $FIRMWARE_VERSION
```

Global Alarm Attributes

This section lists the DX NetOps Spectrum Global Alarm attributes and their corresponding attribute IDs. Pass any of these attributes to AlarmNotifier using the method that is described in this article.

- **Acknowledged**
  0x11f4d
- **Alarm_Source**
  0x11fc4
- **Alarm_Status**
  0x11f4f
- **Cause_Code**
  0x11f50
- **Cleared_By_User_Name**
  0x11f51
- **Creation_Date**
  0x11f4e
- **ImpactScope**
  0x1290e
- **ImpactSeverity**
  0x1290d
- **Last_Occurrence_Date**
  0x1321a
- **Occurrences**
  0x11fc5
- **Originating_Event**

0x1296e
- **Persistent**
  0x12942
- **Primary_Alarm**
  0x11f54
- **Severity**
  0x11f55
- **Trouble_Shooter_Email**
  0x12a6c
- **Trouble_Shooter_mh**
  0x11fc6
- **Trouble_Ticket_ID**
  0x12022
- **TroubleShooter**
  0x11f57
- **User_Clearable**
  0x11f9b
- **Customer_Impact**
  0x12bf6
- **Service_Impact**
  0x12bf7

NOTE
Service_Impact does not always reflect the current health of the service. Service_Impact represents the health status of the service at the time when the device alarm was generated. If only the service health changes, Service_Impact is not affected.

Sending Data to a Third-Party Application

You can customize or replace SetScript, ClearScript, or UpdateScript to create an integration with a third-party application.

You can supplement and extend the functionality of SetScript, ClearScript, or UpdateScript. You can also include DX NetOps Spectrum CLI commands in these scripts to retrieve more information from the SpectroSERVER. You can also add code of your own to the script that sends data to a third-party application.

If you do not use the functionality of the existing script, you can direct AlarmNotifier to run your own script or executable. In the AlarmNotifier resource file (.alarmrc), the Set_Script parameter controls the script that runs when an alarm is set. The Clear_Script parameter controls the script that runs when an alarm is cleared, and the Update_Script parameter controls the script that runs when an alarm is updated.

By default, the Set_Script parameter has a value of SetScript, the Clear_Script parameter has a value of ClearScript, and the Update_Script parameter has a value of UpdateScript. You can modify the values of these parameters to launch a different script when an alarm is set, cleared, or updated.

Custom script parameters depend on the data to extract from DX NetOps Spectrum and on the third-party application to which the data is sent. To create your own script or executable, first, understand the arguments that are passed from DX NetOps Spectrum to the receiving script or executable. Your script or executable must receive all of the arguments from DX NetOps Spectrum in the correct order.

Arguments -- USE_NEW_INTERFACE Set to True

The following table shows the number, name, and format of each argument that is passed to each script when the `USE_NEW_INTERFACE` parameter in the `.alarmrc` file is set to TRUE.

NOTE
When USE_NEW_INTERFACE is set to TRUE, the Status argument is sent as an environment variable. The order of arguments is therefore affected.

Argument	Name	Format
1	Date	mm/dd/yyyy
2	Time	hh:mm:ss
3	Model Type	Text
4	Model Name	Text
5	Alarm ID	Integer
6	Severity	Text
7	Cause	Text
8	Repair Person	Text
9	Server	Text
10	Landscape	Hexadecimal
11	Model Handle	Hexadecimal
12	Model Type Handle	Hexadecimal
13	IP Address	xxx.xxx.xxx.xxx
14	Security String	Text
15	Alarm State	Text
16	Acknowledged	Text
17	Clearable	Text
18	Device Type	Text

Arguments -- USE_NEW_INTERFACE Set to False

The following table shows the number, name, and format of each argument that is passed to each script when the `USE_NEW_INTERFACE` parameter in the `.alarmrc` file is set to FALSE.

NOTE
If you are working with a SANM-enabled AlarmNotifier, additional arguments are passed. For more information, see the Spectrum Alarm Notification Manager (SANM) section.

Argument	Name	Format
1	Date	mm/dd/yyyy
2	Time	hh:mm:ss
3	Model Type	Text
4	Model Name	Text
5	Alarm ID	Integer
6	Severity	Text
7	Cause	Text
8	Repair Person	Text
9	Status	Text
10	Server	Text

11	Landscape	Hexadecimal
12	Model Handle	Hexadecimal
13	Model Type Handle	Hexadecimal
14	IP Address	xxx.xxx.xxx.xxx
15	Security String	Text
16	Alarm State	Text
17	Acknowledged	Text
18	Clearable	Text
19	Device Type	Text

Date and Time

The following conditions apply to the Date and Time arguments for the Set, Clear, and Update Scripts:

- **For SETs:** Date and Time are derived from the CREATION_DATE (0x11f4e) attribute. SpectroSERVER sets this attribute when the alarm is created.
- **For CLEARs:** Date and Time are derived from the CLEAR_DATE (0x129af) attribute. SpectroSERVER sets this attribute when the alarm is cleared.
- **For UPDATEs:** Date and Time reflect when the AlarmNotifier received notification that the alarm has been updated. SpectroSERVER does not set this value. Do not rely on this value to determine the exact time that the update occurred.

Customizing the .alarmrc Resource File

The .alarmrc resource file, which is saved in the Notifier directory, includes AlarmNotifier operational parameters. You can modify the resource file in the following ways:

- Specify whether AlarmNotifier processes alarms that DX NetOps Spectrum generated before AlarmNotifier started.
- Specify optional DX NetOps Spectrum attributes to pass to AlarmNotifier.
- Replace SetScript, ClearScript, or UpdateScript with custom scripts.
- Specify the SpectroSERVER to which AlarmNotifier connects.
- Disable the parameters that specify AlarmNotifier actions that you do not plan to deploy, reducing network traffic.

NOTE
Commenting out parameters does not disable them. Instead, their default value is used.

Follow these steps:

1. Navigate to the <*$SPECROOT*>/Notifier directory and make a backup copy of the .alarmrc file.
2. Open the file with your preferred text editor.
3. Edit the file by turning off optional parameters or by entering new parameter values. You can disable a parameter by giving it a value of False or by leaving the value blank.

   NOTE
   Do not disable required parameters or delete parameters.
4. Save and close the file, and then restart AlarmNotifier.
   Your changes go into effect when AlarmNotifier is restarted.

.alarmrc Parameters

The following list describes the resource file parameters that are provided with AlarmNotifier. For more information, see the Spectrum Alarm Notification Manager (SANM) section.

- **LANDSCAPE**
  Identifies the initial SpectroSERVER host to which AlarmNotifier connects. Enter only one name here. If LANDSCAPE is not defined, AlarmNotifier defaults to using the first landscape in the VNM landscape map. An informational window shows the default landscape handle.
- **VNM_MAIL_TIMEOUT**
  Specifies the minimum time that the Mail Service waits for a response from the SpectroSERVER before the request is canceled.
  Default: 60,000 milliseconds (one minute)
- **VNM_CONNECT_TIME_LIMIT**
  Specifies the minimum delay before an initial TCP connect request between AlarmNotifier and a SpectroSERVER times out.
  Default: 60,000 milliseconds (one minute)
- **KEEP_ALIVE_TIMEOUT**
  Specifies the amount of time before a keep-alive request times out.
  Default: 30,000 milliseconds (30 seconds)
- **KEEP_ALIVE_INTERVAL**
  Specifies the amount of time between keep-alive requests that are sent to the SpectroSERVER. A keep-alive request checks to see if the SpectroSERVER is still connected to the AlarmNotifier. If AlarmNotifier does not receive a response to the request, it disconnects from the SpectroSERVER. If your SpectroSERVER is slow to respond to these requests, you can increase this value to prevent the AlarmNotifier from disconnecting from the SpectroSERVER.
  Default: 60,000 milliseconds (1 minute)
- **SEND_ALARM_DELAY**
  Specifies the minimum delay between successive alarm notifications.
  Default: 1,000 milliseconds (1 second)
- **GET_GRAY_INITIAL_ALARMS**
  Specifies whether you want to receive Gray and Initial alarms. If you do not want Gray or Initial alarms, set this parameter to FALSE. This setting reduces the network traffic that AlarmNotifier generates and improves its performance.
- **GET_EXISTING_ALARMS**
  Specifies whether you want to receive reports about the alarms that exist when AlarmNotifier is invoked. Otherwise, you only receive reports of alarms that occur after AlarmNotifier is invoked. The "Waiting for more alarms from the SpectroSERVERs" message appears during any interval between alarm notifications.

Reinstalling or Upgrading the Product

When you reinstall DX NetOps Spectrum or you upgrade the version of DX NetOps Spectrum, the install process automatically saves the SetScript, UpdateScript, and ClearScript to a backup directory. Versions of the default scripts that you have saved under another name, for example SetScript_version1 or UpdateScript_modified, are retained in the <*$SPECROOT*>/Notifier directory. That directory also contains the default scripts that are included with the reinstallation or upgrade.

In addition, the install process saves your .alarmrc file to a backup directory. Versions of the .alarmrc resource file that you have saved under another name, .alarmrc1 or .alarmrc2 for example, are retained in the <*$SPECROOT*>/Notifier directory.

The backup scripts and the backup .alarmrc are saved to the following directory:

```
<$SPECROOT>/Install-Tools/SAVES_<date>/<time>/Notifier
```

WARNING

When using a custom AlarmNotifier, created by using a non default-directory, the AlarmNotifier does not upgrade by the DX NetOps Spectrum migration task and it results in having AlarmNotifier executable files (and modules/libraries) not in sync with the upgraded DX NetOps Spectrum install. This may cause severe issues such as, having the "old_release Notifier" opening a connection to the "new_release SpectroSERVER". In this case you have to manually update the custom AlarmNotifier directory to cover the current release executables/modules.

WARNING

During the DX NetOps Spectrum upgrade process, customization that were made to the default files(shipped with the product) in the $SPECROOT/lib/SDPM/partslist/ location will be lost. It is recommended to preserve these files before starting the upgrade process, and merge them back after the upgrade.

Spectrum Alarm Notification Manager (SANM)

The Alarm Notification Manager (*SANM*) is a DX NetOps Spectrum component that enhances the functionality of DX NetOps Spectrum alarm-processing applications. Multiple alarm-processing applications are available for DX NetOps Spectrum, including AlarmNotifier and Attention! These applications respond to DX NetOps Spectrum alarms by sending email notifications, creating trouble tickets, and more. SANM lets you create and associate alarm notification policies with applications.

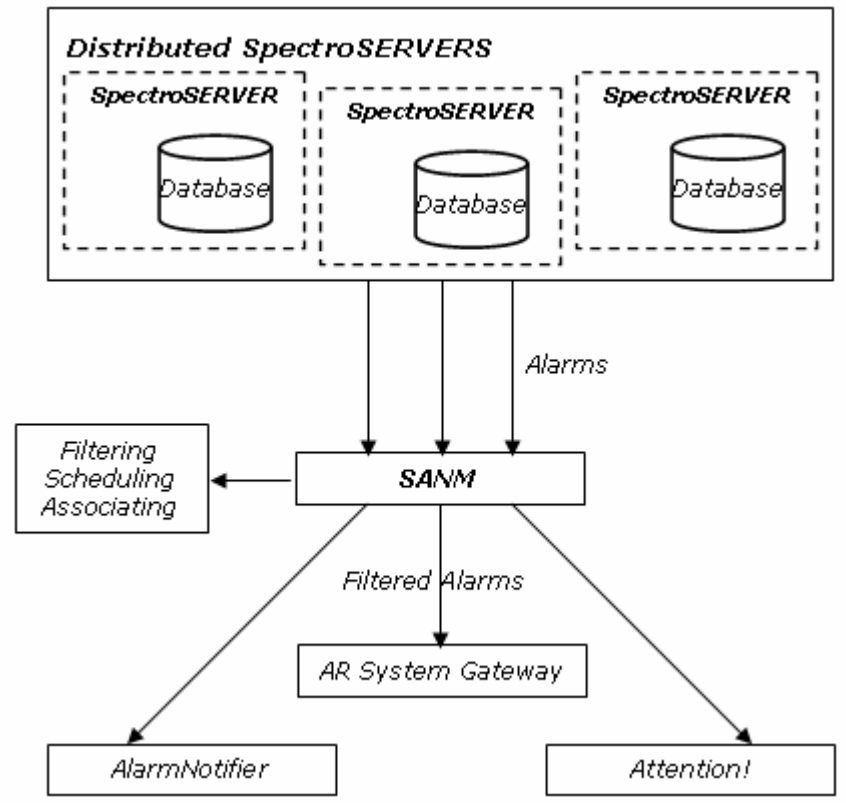
NOTE

From 10.3.1 onwards, SD Notifier (SANM) supports HTTPs.

How the Product Monitors Alarms

DX NetOps Spectrum, alarm-processing applications, and SANM work together in the alarm monitoring process.

The following diagram shows the alarm monitoring process:



The following workflow describes how DX NetOps Spectrum monitors alarms:

1. DX NetOps Spectrum polls the modeled network elements and updates the status of each element in the SpectroSERVER database.
2. DX NetOps Spectrum generates an alarm when it receives a trap from the network, or when it detects a critical status change in a network model. In the OneClick Console, the model icon changes from green to another color that indicates the alarm severity level.
   - DX NetOps Spectrum posts specific information for each alarm on the Alarm Details tab of the Component Detail pane.
   - DX NetOps Spectrum posts alarm event information to the Events tab of the Component Detail panel.
3. Data about alarms that DX NetOps Spectrum has generated is passed to SANM. SANM lets you create and associate alarm notification policies with alarm processing applications. In addition, the SANM Schedule subview lets you schedule application and policy associations and automates the association process.
4. SANM passes the alarm information to alarm processing applications only when the alarm types specified in the policies occur.

AlarmNotifier

Alarms that SANM filters are sent to AlarmNotifier. When both SANM and AlarmNotifier are installed, AlarmNotifier gains some capabilities:

- You can apply the SANM alarm-filtering policies to individual instances of AlarmNotifier.
- AlarmNotifier can generate alarm notifications from all landscapes of a distributed SpectroSERVER environment.
- Additional commands are available to acknowledge and clear alarms from AlarmNotifier.
- A new startup command lets you start multiple instances of AlarmNotifier. You can associate each instance with a different SANM alarm notification policy.
- Other new startup commands let you create a summary or detailed trace files.
- AlarmNotifier scripts include new parameters that contain information about troubleshooting alarms.
- The AlarmNotifier resource file includes new parameters to obtain more information about alarms.
- SANM lets you automatically associate a different policy with AlarmNotifier at a specified time.

Attention!

Attention! is a client-server network monitoring and notification system. The Attention! application alerts system managers to critical system and network events. Supported alert formats include alphanumeric paging, telephone calls, email, PA announcements, electronic message boards, and custom notifications. You can use SANM as a foundation for integration between DX NetOps Spectrum and Attention!.

The Alarm Resource File

The alarm resource file, .alarmrc, contains operating parameters that define SANM defaults. You can modify these parameters to customize SANM alarm management.

You can find the alarm resource file in the <*$SPECROOT*>/Notifier directory. For more information, see the Alarm Notifier section.

If you reinstall DX NetOps Spectrum or upgrade the version, the installation saves your resource file, .alarmrc, to a backup directory. Versions of the resource file that you saved with another name are preserved in the <*$SPECROOT*>/SANM directory. That directory also contains the default resource file that is included with the reinstallation or upgrade.

> **NOTE**
> We recommend creating a backup copy of the file before you modify it.

Create an Alarm Notification Policy

An alarm notification policy specifies the alarm types that an alarm-processing application receives and filters the unwanted alarms. You can create alarm notification policies to determine which applications receive alarms of the types you select.

Follow these steps:

1. Click the Locater tab in the Navigation panel of the OneClick Console.
2. Select All Applications under SANM, and click the search icon 
   The Select Landscapes to Search dialog opens.
3. Select the landscapes to include in your search, and click OK.
   The available applications and the policy that they are using appear in the Contents panel on the right. The policy details appear in the Component Detail panel below the Contents panel.

   > **NOTE**
   > Run AlarmNotifier at least once. Otherwise, the search returns no models. The AlarmNotifier file is located in the <*$SPECROOT*>/Notifier directory.

4. In the Component Detail panel, click the link to create or set policy under General Information.
   The Select Policy dialog opens.
5. Click Create.
   The Create SANM Policy dialog opens.
6. Enter the policy name in the Name text box.
7. (Optional) Create one or more filters to associate with the new policy.

   NOTE
   We recommend assigning policy names that indicate when the policy is used. For example, use a name like 'ciscoRtrPM' so that you can identify Cisco router policies in a collection.
8. Click OK.
   The new policy is created.

Define a Filter For a Policy

You can define alarm notification policy filters that refine notification policies. A filter must be associated with a policy. Filters include parameters to include or exclude alarms by severity and by device type. You can set filters for alarms of specified types, on specified landscapes, or in specified topologies.

Follow these steps:

1. Click the Locater tab in the Navigation panel of the OneClick Console.
2. Select All Applications under SANM and click the search icon 
   The Select Landscape to Search dialog opens.
3. Select the landscapes you want to include in your search and click OK.
   The available applications and the policy they are using appear in the Contents and Component Detail panels on the right.
4. In the Component Detail panel, click the create/set policy link in the General Information subview.
   The Select Policy dialog opens.
5. Click Create.
   The Create SANM Policy dialog opens.
6. Click the Add button.
   The Add Filter dialog opens.
7. Enter the following information:
   - **Name**
     Defines the new filter name.
   - **Notes**
     (Optional) Describes the filter.
   - **Age Time**
     (Optional) Indicates the time for which the filter holds the alarm. The alarm passes to the alarm processing application after the age time.

     NOTE

     If the alarm must pass multiple filters with different ages, SANM uses the shortest, non-zero alarm age interval.
   - **Notification Data**
     (Optional) Defines the data that is sent with the alarm notification.

     NOTE
     DX NetOps Spectrum can differentiate the filter based on Alarm Severity, Landscape, Model Type e.t.c, but can not be differentiated with the Age out time.

For example, If an alarm comes for the same device (modeled on same landscape) with same severity, and if same policy is applied with two different filters, then the alarm passes through the filter with minimum age out and will not pass through the higher age out filters. Since the minimum age out criteria is already met, the higher age out filters are ignored. But DX NetOps Spectrum still collects the notifications from other higher age out filters and sends them to all.

8. Define parameters for your filter:
   a. (Optional) Select the Landscapes tab to define the landscapes for the filter. To define the landscape, select servers in the Include and Exclude lists. You can move servers between the Include and Exclude lists by using the arrow buttons provided.

      NOTE
      OneClick combines the Landscapes and Servers parameters of the legacy SANM UI into a single parameter, Landscapes.
   b. (Optional) Select the Severity tab to define the alarm severity to include or exclude. To define the severity, select alarm severity levels from the Include and Exclude lists.
   c. (Optional) Select the Device Type tab to specify the device types for the filter, as follows:
      - Select an option to see the lists of device types to Include or Exclude.
      - Enter a device type and click Add to add it to the included or excluded list.
      - **Note:** Enter the name of an existing device type, or the name of a device type that you plan to create.
      - Click Browse to select from a list of existing device types.
      - Select a device type and click Remove to remove it from the list.
      - Select a device type and click Modify to edit that device type.
   d. (Optional) Select the Collections tab to specify the collection of policies for the filter. Alarms on devices that are in these collections are filtered. The steps to include, exclude, add, remove, modify, and browse for containers are the same as for the previous tab.
   e. (Optional) Select the Topology tab to specify the topology containers for the filter. Alarms on devices that are in these topologies are filtered. The steps are the same as for the previous tabs.
   f. (Optional) Select the Alarm Type tab to include or exclude alarms of specific types.
   g. (Optional) Select the Model Type tab to include or exclude models of specific types.
   h. (Optional) Select the Location tab to specify location containers for the filter.
      Alarms on devices that are in these locations are filtered.
   i. (Optional) Select the Organization tab to specify the organization containers for the filter.
      Alarms on devices that are in these organizations are filtered.
   j. (Optional) Select the IP Address/Range tab to specify the Internet Protocol (IP) addresses for the filter.
      SANM only passes alarms that are generated within the specified network, subnet, or IP address range to the alarm processing application.
   k. (Optional) Select the Model Name tab to specify the model names for the filter.
9. Click OK.
   The new filter is defined.

   NOTE
   If you create a filter with multiple parameters, you create an AND condition. As a result, all of the parameters must return TRUE for the filter to return any results. To create an OR condition, create two filters, each with a different filter parameter.
10. Enter a name for the new policy in the Name field of the Create SANM Policy dialog, and click OK.

    NOTE
    We recommend assigning policy names that indicate when the policy is used. For example, use a name like 'ciscoRtrPM' so that you can identify Cisco router policies in a collection.

The new policy is created.

Add a Filter to an Existing Policy

You can add a filter to an existing policy.

Follow these steps:

1. Click the Locater tab in the Navigation panel of the OneClick Console.
2. Select SANM, All Policies and click the search icon 
   The Select Landscapes to Search dialog opens.
3. Select the landscapes that you want to include in your search and click OK.
   The existing polices display in the Contents panel on the right.
4. Select the policy for which you want to add a filter.
   The policy details display in the Component Detail panel.
5. Expand the Filters menu under the Information tab in the Component Detail panel.
6. Click the icon 
   Opens a dialog (Add a Filter) to add a filter to this policy.
7. Enter the filter information as explained in Define a New Filter and save the information.
   The filter is added to the policy.

Change the Filter Order

You can change the order in which the filters that are associated with a policy are processed.

This feature applies only to Notification Data. For instance, if Notification Data on filter 1 has jack@xyz.com, and filter 2 has jill@xyz.com, the alarm notifier returns jack@xyz.com:jill@xyz.com. If you change the order, the output is jill@xyz.com:jack@xyz.com.

Follow these steps:

1. Click the Locater tab in the Navigation panel of the OneClick Console.
2. Select SANM, All Policies, and click the search icon 
   The Select Landscape to Search dialog opens.
3. Select the landscapes to include in your search, and click OK.
   The existing policies appear in the Contents panel on the right.
4. Select a policy whose filter order you want to change.
   The policy details appear in the Component Detail panel below the Contents panel.
5. Expand the Filters menu under the Information tab.
6. Click the icon 
   Opens a dialog (Set Order) to set the Notification data order.
7. Select a filter.
8. Use the arrow buttons to move the filter up or down in the order, and click OK.
   The filter is processed according to the new order.

Edit a Filter

Edit a filter to change the values of filter parameters. You can add, edit, and delete the filter parameters.

Follow these steps:

1. Click the Locater tab in the Navigation panel of the OneClick Console.
2. Select SANM, All Policies, and click the search icon  .
   The Select Landscape to Search dialog opens.
3. Select the landscapes that you want to include in your search, and click OK.
   The existing policies appear in the Contents panel.
4. Select the filter in the filter table.
5. Click the icon  .

   Opens a dialog to enable editing of the selected filter.
6. Edit the Name, Notes, Age Time, Notification Data fields as required.
7. Click each parameter tab to add, edit and delete the corresponding parameter values, as explained in Define a New Filter.

   NOTE
   If you delete all values of a parameter, the filter no longer includes that parameter.
8. Select the Show only filtered by parameters check box if you want to view only the parameters included in the filter.
9. Click OK.
   The filter is edited.

Add Filter Parameters

You can add parameters to a filter to increase the level of filtering.

NOTE
You can add a parameter to a filter by adding a value to that parameter. That is, if a parameter was not included when you created the filter, defining a value for that parameter adds that parameter to the filter.

Follow these steps:

1. Open the filter for editing.
2. Click the tab of the parameter that you want to add to the filter.
3. Add one or more values to the parameter, as explained in Define a New Filter.
4. Click OK.
   The parameter is added to the filter.

Delete a Filter

You can delete a filter that is no longer required.

To delete a filter, select the filter in the filter table, and click the delete icon to permanently delete the selected item  .

The filter is deleted.

Add a Model or Alarm to a Policy

You can add a model or an alarm to a policy.

Follow these steps:

1. Click the Explorer tab in the Navigation panel of the OneClick Console.
   The model or alarm details display in the Contents panel on the right.
2. Right-click the model or alarm and select Add to, SANM Policy, Add.
   The Select Policy dialog opens.
3. Select a policy and click OK.
   The Select Write Option dialog opens.

   NOTE
   To remove the item, select Remove.
4. Select an option.
   The model or alarm is added to the selected policy.

Editing an Alarm Notification Policy

You can edit a policy before or after you save it, regardless of whether it is associated with an application. If the policy is associated with an application, SANM begins enforcing the new policy as soon as you save your changes.

WARNING
The Archive Manager must be running and connected to the SpectroSERVER for modified policies to take immediate effect.

The Association Process

After you create an alarm notification policy, you associate the policy with one or more alarm processing applications. An association between a policy and an application remains in effect until you associate another policy with that application or delete the associated policy.

SANM enforces a rule that an application can have only one associated policy at a time. To let an application to process different alarms at different times, associate the policies with the applications manually at run time. Or use the Schedule subview to schedule the associations automatically at a specified date and time. To run the same application with different policies, start multiple instances of the application, each with a unique name. Then associate the different policies with the application instances.

To change the policy that is associated with an application, associate a policy, such as the default policy, with that application. If you instead delete the associated policy, SANM associates the default policy with the application. Editing a policy that is associated with multiple alarm processing applications changes the policy for all of the applications. Reassociating the policy with each application is not required.

The SANM Default Policy

SANM associates a default policy with each application when you start the application for the first time, or when you delete a policy associated with that application. You can also explicitly associate the default policy with an application.

The default policy is a null policy; it does not filter alarms. That is, applications that are associated with the default policy receive all alarm notifications that occur in every landscape in the landscape map of the SpectroSERVER to which SANM is connected.

You can modify the default policy to add filters, but SANM continues to associate it with applications by default.

If you delete a policy that is associated with an application, SANM associates the default policy with that application. Therefore, before you delete a policy, check whether the default policy has been modified. If you delete a policy that is associated with an application, or if you modify the default policy, SANM displays a warning.

You can avoid associating the default policy associated with an application when you delete the associated policy. First associate a different policy with the application. The current policy is automatically deleted.

Associate a Policy with an Application

You can associate a policy with an application in OneClick.

Follow these steps:

1. Click the Locater tab in the Navigation panel of the OneClick Console.
2. Select SANM, All Applications and click
   
   (Launch the selected search).
   The Select Landscape to Search dialog opens.
3. Select the landscapes to include in your search, and click OK..
   The existing applications appear in the Contents panel on the right.
4. Click the Create/Set Policy link.
   The Select Policy dialog opens.
5. Select a policy and click OK.
   The policy is associated with the application.

The Schedule Subview

The Schedule subview automates the association process and lets you implement alarm notification policies according to a schedule. For example, if you want an alarm application to take action in response to an alarm during the evening, you can create a special evening policy and can schedule the association of this policy with the application for 6 PM every day. You can then schedule the association of a different daytime policy with the same application for 7 AM every day. The Schedule subview lets you perform scheduled associations. You can avoid manually associating a new policy each time you want a change in alarm filtering.

You can verify the results of operations that were performed by the Schedule subview on the Events tab in OneClick.

Schedule an Association

You can schedule a policy association with an application in OneClick.

Follow these steps:

1. Click the Locater tab in the Navigation panel of the OneClick Console.
2. Select SANM, All Applications and click
   
   (Launch the selected search).
   The Select Landscape to Search dialog opens.
3. Select the landscapes to include in your search, and click OK..
   The existing applications display in the Contents panel on the right.
4. Select the SANM application whose policy you want to schedule.

5. In the Component Detail panel, expand the Scheduled Policies menu under the Information tab and click 
(Opens a dialog to schedule a policy to the current policy).
The Select Policy And Schedule dialog opens.
6. Select a policy, select a schedule, and click OK.

NOTE
You can create custom policies and schedules by clicking the Create buttons.

The scheduled policy displays in the Scheduled Policies table.

Additional Utilities

AlarmNotifier includes three utilities that you can use to manage existing alarms:

- assignticket
- clearticket
- updatealarm

assignticket Utility

The assignticket utility is used to populate the Trouble Ticket ID field of an alarm with the name of the person to whom the ticket is assigned.

Run this utility using the following syntax:

```
assignticket modelhandle alarmid assignee [username]
```

- ***modelhandle***
  Indicates the handle of the model where the alarm was raised.
- ***alarmid***
  Indicates the ID of the alarm to which to write.
- ***assignee***
  Indicates the name of the user to whom the ticket is assigned.
- ***username***
  (Optional) Specifies the name of the DX NetOps Spectrum user account to use to connect to the SpectroSERVER.

clearticket Utility

Use the clearticket utility to clear an alarm.

Run this utility using the following syntax:

```
clearticket -mh model_handle -ai alarm_ID -su username
```

- **-mh *model handle***
  Indicates the handle of the model where the alarm exists.
- **-ai *alarm_ID***
  Indicates the ID of the alarm to clear.
- **-su *username***
  Specifies the name of the user account to use to connect to the SpectroSERVER.

updatealarm Utility

Use the updatealarm utility to set the value of any attribute on any alarm.

Run this utility using the following syntax:

```
updatealarm modelhandle alarmid attrid attrvalue [username]
```

- ***modelhandle***
  Indicates the handle of the model where the alarm was raised.
- ***alarmid***
  Indicates the ID of the alarm to which to write.
- ***attrid***
  Indicates the ID of the attribute to which to write.
- ***attrvalue***
  Indicates the value to write to the attribute.
- ***username***
  (Optional) Specifies the name of the user account to use to connect to the SpectroSERVER.

Monitoring SANM Processes

Monitoring SANM processes involves viewing SANM events and tracing policies. This section discusses how you can view the SANM events and the type of trace files available to trace policies.

SANM Events

The Events tab in OneClick lists events that occur on a SpectroSERVER. When a user performs a SANM operation, the results of the operation appear on the Events tab with other DX NetOps Spectrum events. The following information about an event is listed:

- Date and time of the operation
- Application name and policy name
- User's host and user name
- Explanation of the event
- Event code

> **NOTE**
> For more information about using the Events tab, see the Using OneClick section.

SANM Event Codes

Each SANM event code corresponds to a SANM operation. Use the following SANM event codes to locate SANM operation entries or to filter out all entries that are not specific to SANM operation.

- **00d70000**
  Application registered with SANM
- **00d70001**
  Application unregistered with SANM
- **00d70002**
  Association created
- **00d70004**
  Scheduled association created
- **00d70006**
  Policy created
- **00d70008**

Policy modified
- **00d7000a**
  Application created
- **00d7000b**
  Application creation failed

Tracing Policies

To collect information about how a policy is working for a SANM-enabled application, at application startup you can enable the creation of a detailed or summary trace file for that application.

- **Detailed trace file**: Indicates the filters in a policy alarm that did not match when they were evaluated against that policy.
- **Summary trace file**: Indicates the time when an alarm notification is passed to the associated application when that application is started. A summary trace file does not include information about alarms that do not meet the criteria that are specified in a policy.

Use a record of policy-based actions by SANM as a decision-making tool. The results may confirm that you have the correct policy in place for an application, or they may compel you to refine your policy. For example, you can discover that you are inadvertently excluding alarms that should be passed to an application.

The Summary Trace File

The summary trace file includes a summary of all alarm notifications (set, cleared, updated) sent to the application, as follows:

```
05/24/2000 15:48:44 SANM Trace Entry 1
Notification sent to AlarmNotifier for Alarm 52 set on landscape 0x540000
05/24/2000 15:48:44 SANM Trace Entry 2
Notification sent to AlarmNotifier for Alarm 21 updated on landscape 0x540000
05/24/2000 15:48:44 SANM Trace Entry 3
Notification sent to AlarmNotifier for Alarm 26 cleared on landscape 0x540000
```

The summary trace file does not indicate the alarms that failed the policy.

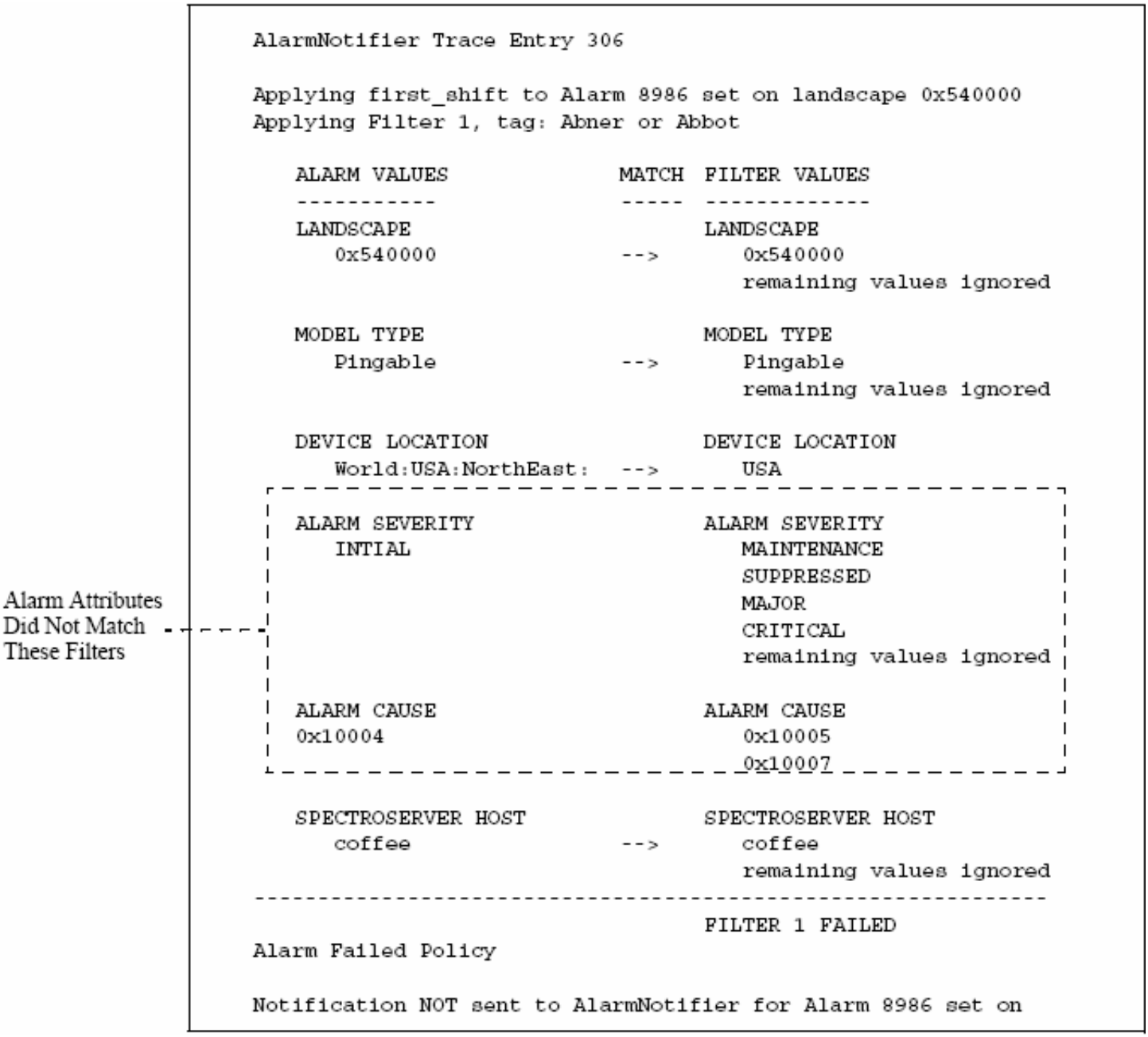
The Detailed Trace File

A detailed trace file includes entries for alarms that meet and that do not meet the criteria of a policy. An alarm entry includes the alarm attribute values, which that are compared to the filter parameter values. An arrow symbol under MATCH between ALARM VALUES and FILTER VALUES indicates a match. The arrow is absent if the values do not match.

The following is an example of a trace file that indicates that an alarm passed a policy:

```
AlarmNotifier Trace Entry 305

Applying first_shift to Alarm 8982 set on landscape 0x540000
Applying Filter 1, tag: Abner or Abbott

   ALARM VALUES              MATCH  FILTER VALUES
   -----------               -----  -------------
   LANDSCAPE                        LANDSCAPE
      0x540000               -->       0x540000
                                       remaining values ignored

   MODEL TYPE                       MODEL TYPE
      Pingable               -->       Pingable
                                       remaining values ignored

   DEVICE LOCATION                  DEVICE LOCATION
      World:USA:NorthEast:   -->       USA

   ALARM SEVERITY                   ALARM SEVERITY
      CRITICAL                         MAINTENANCE
                                       SUPPRESSED
                                       MAJOR
                              -->      CRITICAL
                                       remaining values ignored

   ALARM CAUSE                      ALARM CAUSE
   0x10007                             0x10005
                              -->      0x10007

   SPECTROSERVER HOST               SPECTROSERVER HOST
      coffee                 -->       coffee
                                       remaining values ignored
-------------------------------------------------------------
                                    FILTER 1 PASSED
Alarm Passed Policy

Notification sent to AlarmNotifier for Alarm 8982 set on landscape
0x540000
```

The following is an example of a trace file that indicates that an alarm failed a policy:



SANM and AlarmNotifier

This section discusses how SANM enhances the capabilities of AlarmNotifier.

AlarmNotifier Enhancements

AlarmNotifier gains capabilities when you install SANM on your system. These capabilities include additional startup options for specifying application names and for creating trace files, alarm acknowledge and alarm clear commands, and script and resource file parameters. SANM also lets AlarmNotifier operate in a distributed environment.

Start AlarmNotifier

AlarmNotifier is located in the <*$SPECROOT*>/Notifier directory. This directory contains the following files by default:

- .alarmrc
- AlarmNotifier
- ClearScript
- README
- SetScript
- UpdateScript

AlarmNotifier includes the following additional files and directory:

- **AlarmAck**
  Acknowledges an alarm.
- **AlarmClear**
  Clears an alarm.
- **Trace**
  Displays trace files.

To start AlarmNotifier, use the following AlarmNotifier command in the <*$SPECROOT*>/Notifier directory:

```
AlarmNotifier [-r resourcefile] [-n application][-tl summary|details [-tn tracefile] [-ts size]]
```

- **-r *resourcefile***
  Lets you specify a resource file other than the default resource file .alarmrc.
- **-n *application***
  Lets you override the application name value that is specified by the APPLICATION parameter in the resource file. You can specify a different name for an AlarmNotifier application instance. This option lets you start multiple instances of AlarmNotifier and associate each of them with a different SANM alarm-filtering policy. If a name is not assigned to the APPLICATION parameter in the resource file, use the -n option at start-up to specify an application name.
- **-tl summary | details**
  Lets you activate tracing at a specified level, summary or detailed. The default format for an AlarmNotifier trace file is the application name together with the date when the trace file was created.
- **-tn *tracefile***
  Lets you specify a trace file name other than the default name, which is provided when only the -tl option is used. Use this option with the -tl option.
  When using the trace file option, the output file is written by default to the <*$SPECROOT*>/Notifier/trace directory. To explicitly name an output file and path, use the [-tn filename] option. If <*filename*> is a relative path, trace output is written to a file that is relative to the current directory. If <*filename*> is an absolute path, trace output is written to the absolute path.
- **-ts *size***
  Lets you specify the number of lines in the trace file. Use this option with the -tl option. The application writes this number of lines to the file and then wraps around to the beginning of the file. Entries are numbered sequentially, and an END OF TRACE line follows the last entry. The default number of lines in a trace file is 10000.

  NOTE

  If you migrate DX NetOps Spectrum to a higher release, verify the following settings in the ".alamrc" file:

  - The paths to the scripts are correct.
  - You have all the permissions to the "ClearScript" script.

  AlarmNotifier does not start if these settings are incorrect.

Access Alarm Management Parameters

The Alarm Management view lets you control some aspects of alarm management. Two parameters in this view, Generate Alarm Events and Add Events to Alarms, determine how the SpectroSERVER reacts to alarm updates.

You can view and modify alarm management parameters in OneClick to control some aspects of alarm management.

Follow these steps:

1. Open OneClick.
2. In the Navigation panel, select a VNM model in the Universe view.
   The corresponding details appear in the Contents panel and Component Detail panel on the right.
3. In the Component Detail panel, select the Information tab and open the Alarm Management menu.
   The following alarm management parameters affect alarm event updates:
   - **Generate Alarm Events**
     Enables the generation of alarm change events (which indicate that alarms are generated, updated, or cleared).
     Default: Enabled (Yes).

     WARNING
     When **Generate Alarm Events** is enabled, always select the **Store Event in Historical Database** option for the event in the Event Configuration window. Otherwise, event information does not appear in the Events tab of the Component Detail pane as that event is not logged in the Historical database. In addition, always enable **Generate Alarm Events** when the **Store Event in Historical Database** option is selected. Otherwise, the **Severity**, **Cleared On** and the **Cleared By** fields are not updated in the Events tab.
   - **Add Events to Alarms**
     Controls whether alarm change events are added to each alarm. If disabled, alarm change events are not displayed in the Events tab of the Component Detail panel for the alarm. When enabled, adds any event that affects the alarm, thus incrementing the Occurrence counts. For example, events with different event types that generate the same alarm, such as alarm management or alarm clearing events, are also added.
     Default: Disabled (No).

     NOTE
     For more information, see the Distributed SpectroSERVER Administration section.

Verify that Originating Event Data Is Saved

High traffic levels can prevent the Archive Manager from consistently providing the events that are associated with reported alarms. In such a case, you can still retrieve some basic information about the events that are associated with alarms. The SpectroSERVER stores this information by default.

NOTE
Only information about the first event that is associated with an alarm (the originating event) can be retrieved.

The Store_Originating_Event attribute (0x1296f) of the Alarm Management application model determines whether originating event information is available to the AlarmNotifier. Verify that the default setting, Yes (Enabled), is in force so that event information is available in failover situations.

Follow these steps:

1. Click the Locater tab in the Navigation panel of the OneClick Console.
2. Expand Application Models.
3. Double click By Name.
4. The Search box opens.
5. Type "AlarmMgmt" in the Model Name Contains field, and click OK.
   The Alarm Management model appears in the Results panel.

6. Select the AlarmMgmt model.
   The corresponding details appear in the Component Detail panel.
7. In the Component Detail panel, click the Attributes tab.
8. Type "Store" in the Search box to locate the Store_Originating_Event attribute.
9. Double-click it to verify the value in the right pane.

Alarm Acknowledgement

The AlarmAck command allows you to acknowledge alarms. This command can be used at any shell command prompt to acknowledge specific alarms, or it can be incorporated into a script. AlarmAck returns a value of 0 if the operation succeeds. Otherwise, it returns a non-zero value.

To acknowledge an alarm, run the AlarmAck command with the following syntax:

```
AlarmAck -a alarm -l landscape
```

- **-a *alarm***
  Defines the alarm ID.
- **-l *landscape***
  Defines the landscape handle for the landscape where the alarm was raised.

  NOTE
  Available only for distributed SpectroSERVER environments.

To acknowledge all alarms for a model, run the AlarmAck command with the following syntax:

```
AlarmAck -m modelhandle
```

- **-m *modelhandle***
  Specifies the model handle for the model with the alarm conditions.

User-Clearable Alarms

The AlarmClear command clears user-clearable alarms. To determine whether an alarm is user-clearable, check the value of the UserClearable parameter in alarm notifications. AlarmClear can be launched from any shell command prompt to clear specific alarms, or you can incorporate it into a script. AlarmClear returns a value of 0 if the operation succeeds. Otherwise, it returns a non-zero value.

You can run the AlarmClear command to clear alarms using the following syntax:

```
AlarmClear -a alarm -l landscape
```

- **-a *alarm***
  Defines the alarm ID.
- **-l *landscape***
  Defines the landscape handle of the landscape where the alarm was raised.

SANM-Enabled Script Parameters

The SetScript, UpdateScript, and ClearScript scripts have additional parameters when they run on a computer where SANM is installed.

The following list describes the SANM-enabled script parameters:

- **FlashGreen**

Displays in ClearScript notifications but not in SetScript or UpdateScript notifications.
When enabled, the cleared alarm exhibits the *flash green* condition: the flash green option for the model is enabled, and the GET_FLASH_GREEN parameter in the .alarmrc resource file is set to True. Even though SetScript and UpdateScript notifications do not display this field, the parameter is passed to these scripts, but it is invalid and has a default value of False.

- **Location**
  Identifies the location model that contains the network element whose alarm is set, updated, or cleared. The element must be modeled in the OneClick World topology view. You can find the location model that contains the model for the problematic network element in a colon-separated, hierarchical list of location models. For example, an alarm for a model that is contained in Room 222 on the first floor of the Boston building in the northeast region of the United States appears as follows:
  USA:Northeast:BostonBldg:FirstFloor:Room222.
- **AlarmAge**
  Specifies the length of time that SANM retains an alarm from an instance of AlarmNotifier that is associated with that policy. The AlarmAge is set in the filters in an SANM policy. If the alarm must pass multiple filters with different ages, SANM uses the shortest, non-zero alarm age interval.
- **NotificationData**
  Lists notification data entries (names of persons) that SANM passes to an instance of AlarmNotifier that is associated with that policy. The entries are specified in the filters in an SANM policy. AlarmNotifier scripts can be configured to initiate email notifications to those persons in the notification data entries.
- **ProbableCause**
  Is the probable cause text associated with the alarm.
- **EventMessage**
  Is the message about the events that are associated with the alarm. This field is blank if the DX NetOps Spectrum alarm has no associated events, or if the event does not include alarm information.

Email Notifications

If you use an AlarmNotifier script to send an email notification, set the value for the VARFORMAIL parameter in the script. This parameter specifies to whom the email message is sent.

If you are using SANM-enabled AlarmNotifier, use the NotificationData parameter to set the value for VARFORMAIL. If you use NotificationData as the value for VARFORMAIL, email is sent to the persons who are specified in the NotificationData parameter in the SANM policy that is associated with the instance of AlarmNotifier that invokes the script. For example, if the Notification Data entry is formatted as "John: Mary or Sue: Lynn, Jeff", email is sent to John, Mary, Lynn, and Jeff, but not to Sue, because AlarmNotifier interprets the colon as an AND operator and does not act on the OR operator.

Other possible values for the VARFORMAIL parameter are RepairPerson or both. The RepairPerson option is the only option that is available for AlarmNotifier when it is not running with SANM. Both options indicate that the email notification is sent to the designated RepairPerson and to the person who is specified by the NotificationData parameter.

NOTE
For more information about configuring an AlarmNotifier script to send an email notification, see the AlarmNotifier section.

Third-Party Applications

You can customize or replace the SetScript, ClearScript, or UpdateScript for integration with a third-party application. If you create your own script or executable, understand which arguments are passed from DX NetOps Spectrum to the receiving script or executable. The script or executable must receive all of the arguments that DX NetOps Spectrum passes to it in the correct order.

NOTE
Any DX NetOps Spectrum attribute of the model with the alarm can be passed to AlarmNotifier and can be used in a script. For more information, see the AlarmNotifier section.

The following table shows the argument number, name, and format for each argument that is passed to each script when the USE_NEW_INTERFACE .alarmrc parameter is set to TRUE:

Argument	Name	Format
1	Date	mm/dd/yy
2	Time	hh:mm:ss
3	Model Type	Text
4	Model Name	Text
5	Alarm ID	Integer
6	Severity	Text
7	Cause	Text
8	Repair Screen	Text
9	Server	Text
10	Landscape	Hexadecimal
11	Model Handle	Hexadecimal
12	Model Type Handle	Hexadecimal
13	IP Address	xxx.xxx.xxx.xxx
14	Security String	Text
15	Alarm State	Text
16	Acknowledged	Text
17	Clearable	Text
18	Flash_Green	Text
19	Location	Text
20	Age	Integer
21	Notifdata	Text

The following table shows the argument number, name, and format for each argument that is passed to each script when the USE_NEW_INTERFACE .alarmrc parameter is set to FALSE:

Argument	Name	Format
1	Date	mm/dd/yy
2	Time	hh:mm:ss
3	Model Type	Text
4	Model Name	Text
5	Alarm ID	Integer
6	Severity	Text
7	Cause	Text
8	Repair Screen	Text
9	Status	Text
10	Server	Text

11	Landscape	Hexadecimal
12	Model Handle	Hexadecimal
13	Model Type Handle	Hexadecimal
14	IP Address	xxx.xxx.xxx.xxx
15	Security String	Text
16	Alarm State	Text
17	Acknowledged	Text
18	Clearable	Text
19	Flash_Green	Text
20	PCause	Text
21	Location	Text
22	Age	Integer
23	Notifdata	Text
24	EventMsg	Text

If USE_NEW_INTERFACE is set to TRUE, the Status, PCause, and EventMsg arguments are sent as environmental variables. The argument order is therefore affected. If USE_NEW_INTERFACE is set to FALSE, use the following syntax in your script to read data from the PCause and the EventMsg argument into a variable as follows:

```
<variablename>=`echo "$2" | tr '\350' '\012' | tr '\351' '"'`
```

This syntax is required to avoid problems when the script parses the extra data from new lines or other special characters.

Note: For more information about the USE_NEW_INTERFACE .alarmrc parameter, see AlarmNotifier.

SANM-Enabled .alarmrc Parameters

The AlarmNotifier resource file, .alarmrc, has several additional parameters when you run AlarmNotifier on a computer that has SANM installed.

The following list describes the SANM-enabled parameters:

- **APPLICATION**
  Defines the application name that identifies this AlarmNotifier application. If you use multiple AlarmNotifier applications on your network, distinguish them with unique application names, such as AlarmNotifier1 or AlarmNotifier2. You can then use unique SANM alarm-notification policies with each application. If you use the n option when invoking AlarmNotifier, the APPLICATION parameter value is ignored.
  Default: AlarmNotifier
- **GET_LOCATIONS**
  Lets you specify whether to notify you of the location of the device with the alarm. If you are not interested in location information, set this parameter to False. A False setting overrides any location that is specified as a filter parameter in an alarm-notification policy, reducing network traffic.
- **GET_PROBABLE_CAUSES**
  Lets you specify whether you want to receive the Probable Cause text that is associated with each alarm. If you are not interested in Probable Cause information, set this parameter to False, improving AlarmNotifier performance.
  Default: True.
- **GET_EVENTS**
  Lets you specify whether to receive the Event message that is associated with an alarm. If you are not interested in event information, set this parameter to False. Excluding events reduces network traffic that AlarmNotifier generates and improves performance.

Default: True.

- **GET_FLASH_GREEN**
  Lets you specify whether to receive the Flash Green status for a model. ClearScript is the only script that displays the Flash Green status. When Flash Green is enabled for a model, the model continues to flash green after alarms are cleared. The flashing status signals that alarms have occurred even though they no longer exist. If the value of GET_FLASH_GREEN is set to False, the Flash Green status is always passed to the ClearScript as false. If set to True, the Flash Green status is correctly passed as either False or True.
  Default: True.
- **MSG_TIMESTAMP_FORMAT**
  Sets the format for the timestamp on all SANM messages. The maximum length of the output string is 127 characters. Any characters other than the conversion strings are output as text in the timestamp. The default setting is %X %x:. The colon (:) is appended to the end of the timestamp. For example, to output the date/time for the current locale and the time zone name, the string %x %X %Z is entered as the value. If left blank, no timestamp is output on the messages. If an incorrect string is entered, that string displays as text in the output.
- **POLICY_LANDSCAPE**
  Lets you specify the landscape that AlarmNotifier uses for all SANM policy definitions. This parameter works with the POLICY_LANDSCAPE setting in the SANM .sanmrc file.
- **SHOW_ALL_EVENTS**
  Lets you specify whether to receive the most recent event or all events that were generated for an alarm. If set to False, AlarmNotifier only forwards the most recent event. For example, assume that an alarm was created based on an event, and then someone updated the status of that alarm. When the alarm status changed, another event that was related to that alarm was generated. In such a situation, AlarmNotifier only receives the status of that second event. The purpose of this type of filtering is to eliminate events that have already been forwarded. Filtering is especially important if the size of the message is relevant, for example, if the event message is sent as a page.
  Default: False.
- **GET_EXISTING_ALARMS**
  Specifies whether to receive all existing alarms.
  Default: True
- **UPDATE_EXISTING_ALARMS**
  Specifies whether to update the existing alarms. This is different than GET_EXISTING_ALARMS because setting this option to true does not cause the Set Script to get called for existing alarms.
  Default: True.
- **ENABLE_CORRELATION**
  Lets you to enable the alarm correlation awareness feature for all alarms. The value of the SHOW_SYMPTOM_ALARMS parameter determines whether a symptom alarm associated with a cause alarm to display.
  Default: False.
- **SEND_EVENT_UPDATES**
  Lets you to specify whether you want an update action to be triggered when an event is appended to an alarm.
  Default: True.

Using SANM in a Distributed SpectroSERVER Environment

This section discusses how you can use SANM in a DSS environment.

Landscapes and Alarm Monitoring

A Distributed SpectroSERVER (DSS) environment lets you divide network management tasks among several SpectroSERVERs. When you create a network model with multiple SpectroSERVERs, it is possible for SANM to access information from more than one SpectroSERVER simultaneously.

A landscape is the DX NetOps Spectrum term for a network domain that a single SpectroSERVER manages. When SANM operates in a distributed environment, it monitors alarms from all landscapes. Even though different landscapes can model each other in a DSS environment, SANM-enabled applications do not receive duplicate alarm information.

Because SANM evaluates alarms across VNMs in a DSS environment, you may want to limit the type of alarm notifications that you receive. In a DSS environment, limit the number of alarm notifications by carefully defining the parameters, Landscape, Subnet IP Address, and Device Location in the alarm notification policy.

SANM Policy Management Across Multiple Landscapes

Choose between two options for configuring SANM in a distributed environment. You can create SANM policies on any landscape and let SANM read all policies from all landscapes. Or you can create all SANM policies on one landscape and only let SANM read policies from that landscape. In either case, you can associate alarm-processing applications from any landscape with the SANM policies.

How to Create SANM Policies in a Single Landscape

If you set up a distributed environment so that all policies for all landscapes are defined and managed from a single SpectroSERVER, you can install alarm-processing applications on any of the SpectroSERVERs in the distributed environment. If the values in the application resource file are appropriate, the application finds the server that contains the SANM policy definitions and associates it with the appropriate policy. This configuration reduces the initial traffic on the network to associate alarm processing applications and SANM policies, and it also facilitates ongoing SANM policy management.

> **NOTE**
> You cannot migrate or move an SANM policy from one landscape to another. If you want to institute this configuration and already have policies defined on various landscapes, you must recreate these policies on the new landscape from which you will manage SANM policies.

To configure all SANM policies in one landscape, take the following steps:

1. Change the POLICY_LANDSCAPE parameter in the .alarmrc file to the landscape handle of the SpectroSERVER where SANM is installed, and where policies are created and managed.
2. Change the POLICY_LANDSCAPE parameter in the alarm-processing application resource file (.alarmrc) to the landscape handle of the SpectroSERVER where SANM is installed. This parameter instructs the application where to look for defined policies.
3. Restart the SpectroSERVER where SANM is installed and restart the alarm-processing applications so that the changes to the resource file parameters are read.
4. Open SANM, All Policies on the Locater tab of OneClick, and click the search icon to launch the selected search 
   The only available policies are the policies that are created on this landscape. All alarm-processing applications whose POLICY_LANDSCAPE parameter is set to the landscape handle of this landscape are seen in the applications list.

For example, if you have three landscapes (Landscape1, Landscape2, and Landscape3) in your distributed environment, and you have set the value of the 'POLICY_LANDSCAPE' parameter value to 'Landscape2'.

POLICY_LANDSCAPE=<landscape2>

- All the alarm-processing applications point to 'Lancscape2'.
- All the policies are saved to Landscape2 only.
- In case, you decommission the Landscape1 from the environment, you do not lose the policies associated with Landscape1 because all these policies are saved to Landscape2.
- If you decommission the Landscape2 from the environment, all the policies are deleted because all the policies in this environment are saved to Landscape2.

How to Create SANM Policies on Multiple Landscapes

You can set up a distributed environment so that SANM policies can be defined and managed on any SpectroSERVER. Alarm-processing applications on any SpectroSERVER in the distributed environment have access to all of these policies.

To configure SANM policies in multiple landscapes, verify the following requirements:

- Make sure the alarm-processing applications are associated with their respective landscapes.
- Makes sure the POLICY_LANDSCAPE parameter in the different alarm-processing application resource files (.alarmrc) has associated value for different landscapes.

Then take the following steps:

1. Restart the SpectroSERVER where SANM is installed.
2. Restart the alarm-processing applications so that the changes to the resource file parameters are read.
3. Open OneClick. Verify that all policies that have been created within the distributed environment are available.
4. Verify that all alarm-processing applications in the distributed environment are available for association.

For example, if you have three landscapes (Landscape1, Landscape2, and Landscape3) in your distributed environment, and you have set the “POLICY_LANDSCAPE =” value different for each alarm-processing application.

POLICY_LANDSCAPE=<respective landscape>

- All the alarm-processing applications are associated with their respective landscapes.
- All the policies are saved to different landscapes depending on the 'POLICY_LANDSCAPE' parameter setting in the different alarm-processing application resource files.
- In case, you decommission a landscape from the environment, you will lose the policies associated with that landscape only because all these policies are saved to its respective landscape.

NOTE

If the POLICY_LANDSCAPE =” value is empty (no landscape is set) then all the policies are saved to MLS.

- If the AlarmNotifier applications are running and points to MLS only, then all the policies can be created on MLS only.
- If you decommission any of the SpectroServers except MLS, then policies will not be lost because all the policies are saved to MLS only.

Methods for Determining Monitored Landscapes

You can use the following methods to determine which landscapes are monitored by SANM:

- Use the DX NetOps Spectrum Command Line Interface (CLI) application to connect to the SpectroSERVER to which SANM is connected. Then enter **show landscapes** on the command line.
  The CLI application displays a list of all landscapes that are modeled in that server.

  NOTE
  For more information, see the Command Line Interface section.
- Open any one of the detailed trace files that you specified for SANM-enabled applications. A trace file indicates the connection status of each landscape in the landscape map for the SpectroSERVER to which SANM is connected. Trace files are stored by default in a trace directory in the home directory of an SANM-enabled application.

Command Line Interface

This section explains how you can use the command-line interface to perform DX NetOps Spectrum operations from the UNIX, DOS, and Bash command prompts.

Introduction to Command Line Interface (CLI)

This section discusses the following Concepts:

- Use of CLI to perform DX NetOps Spectrum operations from the command prompt.
- DX NetOps Spectrum CLI commands
- How CLI commands can be incorporated into shell scripts or menu systems to give you a more powerful and versatile method of accessing DX NetOps Spectrum data.
- Components of the DX NetOps Spectrum Command Line Interface.
- Environment variables for the CLI.
- How the DX NetOps Spectrum CLI works by using the CLI local server and the startup file.

Command Line Interface Overview

The DX NetOps Spectrum Command Line Interface (CLI) is a core DX NetOps Spectrum component and is installed with the core DX NetOps Spectrum product.

You can access DX NetOps Spectrum data and can execute DX NetOps Spectrum operations from the OneClick user interface. However, if you prefer to execute DX NetOps Spectrum operations from the command line, you can use the CLI. For those tasks that you cannot execute in OneClick, CLI is the only DX NetOps Spectrum resource available to you.

CLI is a powerful tool, but it does not provide the safeguards that OneClick does, especially related to modeling. CLI must be used by DX NetOps Spectrum administrators who understand the potentially harmful effects of haphazardly creating and destroying models and modifying model attributes on a network modeling scheme.

CLI is a flexible option. You can open a CLI session and issue commands from any of the command prompts that are available on your system, such as UNIX, DOS, and Bash.

CLI Commands

CLI commands are similar to UNIX commands, and they can be used with UNIX or DOS commands, especially grep (find), pipes, and redirect symbols. Some CLI commands, however, can conflict with UNIX commands of the same name. For example, the CLI update command can conflict with the UNIX update command.

To avoid conflicts, use **./update** from within the vnmsh directory. When using a script, use the full pathname for the CLI command, for example, <*$SPECROOT*> **/vnmsh/update**.

> **NOTE**
> The CLI update command always provides a response, either a confirmation that the update was successful or a message that the update failed. If you receive no response from CLI when using the update command, type "**which update**". The system likely responds with:

```
/etc/update
```

For more information on commands that are used in the DX NetOps Spectrum Database tools, refer to the following utility programs:

- db_remove
- dbtool
- HostUpdate
- MapUpdate
- reports
- SSdbload
- SSdbsave
- Database Model Conversion Tool (DBconv)

CLI in Shell Scripts

CLI commands can be incorporated into shell scripts or menu systems to give you a more powerful and versatile method of accessing DX NetOps Spectrum data.

Each CLI command sends output reporting the success or failure of the command to standard error. Normal output that is expected as the result of the success of a command, however, is sent to standard output. Each of the commands also generates a return code of zero on success and a non-zero error code on failure. Return codes enable shell scripts using the CLI commands to proceed according to the success or failure of each command.

CLI Components

CLI components are described in this section as follows:

- Executable commands
- Four environment variables
- The daemon that maintains communication with a SpectroSERVER
- The set of sample shell scripts that incorporate CLI commands

CLI Environment Variables

You can set the following four environment variables for CLI:

- **CLIMNAMEWIDTH**
  Displays a model name. By default, the create, seek, and show commands display a maximum of 16 characters for a model name. However, with the environment variable CLIMNAMEWIDTH, you can specify up to 1024 characters to display for model names. For example, using the C shell:
  ```
  setenv CLIMNAMEWIDTH 32
  ```
  You can set this variable in your .login file, in a script, or simply before you issue a command. You can set or change any number of times in a CLI session, depending on the length of the model names.
- **CLISESSID**
  Represents the ID for use in scripts. Set the CLISESSID variable to <$$>, which represents the process ID of the running shell script. This variable is necessary when using cron to run CLI scripts concurrently. For example, using the bash shell:
  ```
  CLISESSID=<$$>; export CLISESSID
  ```
  Also, setting the CLISESSID environment variable to a unique value for each CLI session is required if you run CLI on Windows using the bash shell instead of DOS. You can give a unique timestamp to each bash shell, for example:
  ```
  export CLISESSID='date +%s'
  ```
- **SPECROOT**
  Displays the alarm or event description. The SPECROOT environment variable is required for the show alarms or show events commands when the -x option is specified. This variable gets the description of the alarm or event from the SG-Support directory tree if it can be located so that the output from the commands is expanded.

On UNIX, you can specify the SPECROOT variable in your login shell and can set this variable to the DX NetOps Spectrum home directory. For example:

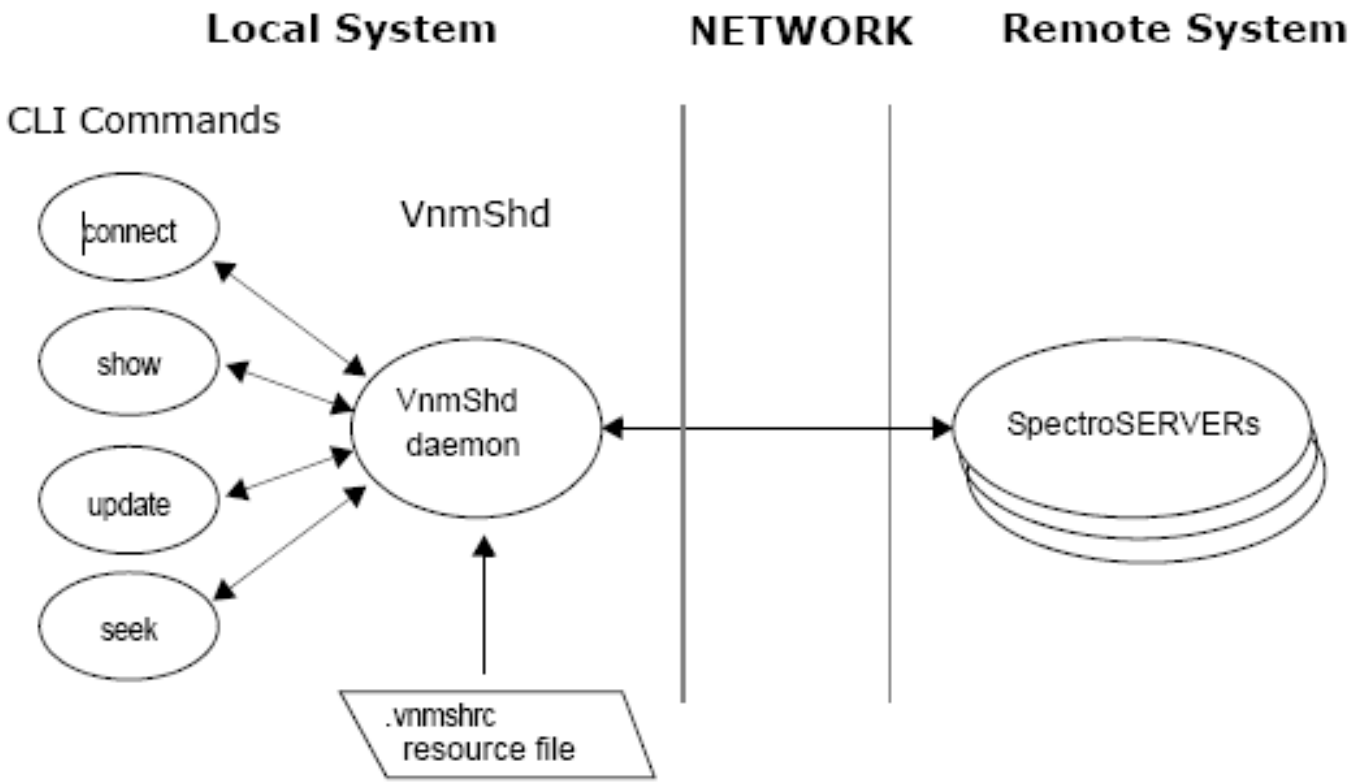
```
SPECROOT=/home/<sp>; export SPECROOT
```

On Windows, see your system documentation for details about setting environment variables.

- **CLIPATH**
  Displays the path of the <*$SPECROOT*>/vnmsh directory and the scripts require to use CLI commands.

CLI Architecture

The following image depicts the CLI architecture:



The CLI Local Server, which uses .vnmshrc at startup, performs the following major functions:

- Maintaining a constant network connection with SpectroSERVER. The CLI Local Server prevents disconnection when a command is executed each time. This server maintains a *single* connection to SpectroSERVER regardless of the number of CLI users that are connected to the daemon. The socket connects and disconnects are expensive as far as time and resource usage are concerned.
- Maintaining state information for each CLI user. The 'current' and setjump commands, for example, require the CLI Local Server to store state information. The current command stores a model handle and a landscape handle for use in future commands. The setjump command stores a text string to identify the current position of users in a DX NetOps Spectrum landscape.

The Startup File

The .vnmshrc file, the CLI Local Server startup file, is located in the <*$SPECROOT*>/vnmsh directory. This file contains several parameters that control how vnmsh communicates with the SpectroSERVER. These parameters are described in the following list:

- **vnm_hostname**
  Specifies the host name of the SpectroSERVER to connect to.
- **client_handshake_timeout**
  Specifies the number of milliseconds the client waits for server ID information, when setting up a connection.
  Default: 900
- **server_handshake_timeout**
  Specifies the number of milliseconds the server waits for client ID information, when setting up a connection.
  Default: 900
- **connect_time_limit**
  Specifies the maximum number of milliseconds to wait for a connection to the SpectroSERVER.
  Default: 1000
- **listen_backlog**
  Specifies the number of client requests to the SpectroSERVER held in queue while waiting for prior ones to complete.
  Default: 10
- **vnm_tcp_port**
  Specifies the TCP port that the vnmsh is using to communicate with the SpectroSERVER when the vnmsh is a SpectroSERVER client.
- **vsh_tcp_port**
  Specifies the TCP port where the vnmsh listens for TCP messages when the vnmsh is acting as a server to the client requests such as show, update.
- **debug_file**
  Specifies the file to which CLI writes error messages.
- **max_show_event_length**
  Specifies the maximum number of characters that are shown when the **show events -x** command is used to display an event message.
  Default: 512

The CLI Local Server

The first user to issue the connect command automatically starts the CLI Local Server (VnmShd daemon) on that workstation and establishes a connection to a SpectroSERVER. Only one CLI Local Server per workstation can be running, and that daemon makes only one connection to a SpectroSERVER.

After the CLI Local Server has been started on a workstation, all subsequent users who connect to CLI on that workstation use the same CLI Local Server.

Error Checking

DX NetOps Spectrum enforces certain rules when you perform tasks in OneClick. For example, rules control the allowable actions when you create or move device models in different views.

CLI does not enforce these rules and cannot perform any error checking. As a result, CLI lets users create models and place them wherever they want without performing error checking. You see an error if you attempt to use a CLI command in a manner that does not conform to its format.

Working with Command Line Interface

This section describes how to start CLI sessions on UNIX, DOS, and Bash prompts. This section also includes examples that demonstrate how to use CLI commands for common tasks in DX NetOps Spectrum. Also how to generate event reports and switch models using the CLI.

Start a CLI Session on UNIX

On a UNIX platform, you can start a CLI session from the shell prompt.

> **NOTE**
> You can use a script to pack up CLI so that it can be sent to another server. For more information, see the Distributed SpectroSERVER Administration section.

Follow these steps:

1. Start the SpectroSERVER to which you want to connect.
2. Navigate to the vnmsh directory in the DX NetOps Spectrum installation directory:
   ```
   $ cd <$SPECROOT>/vnmsh
   ```
3. Open the connection:
   ```
   $ connect
   ```
   You are connected to the CLI session.

Start a CLI Session on Windows using DOS Prompt

On the Windows platform, you can start a CLI session from the DOS prompt.

> **NOTE**
> You can use a script to pack up CLI so that it can be sent to another server. For more information, see the Distributed SpectroSERVER Administration section.

For all instances of UNIX (as opposed to CLI) commands in this section, substitute the equivalent DOS command when necessary. For example, use *find* instead of *grep*.

Follow these steps:

1. For DOS prompt, select Start, Programs, Command Prompt.
   The DOS prompt appears, ready to accept CLI commands.
2. Start the SpectroSERVER to which you want to connect.
3. Navigate to the vnmsh directory in the DX NetOps Spectrum installation directory:
   ```
   $ cd <$SPECROOT>/vnmsh
   ```
4. Open the connection:
   ```
   $ connect
   ```
   You are connected to the CLI session.

Start a CLI Session on Windows Using Bash Prompt

On the Windows platform, you can also start a CLI session from a bash shell prompt.

> **NOTE**
> You can use a script to pack up the CLI program so that it can be sent to another server. For more information, see the Distributed SpectroSERVER Administration section.

Follow these steps:

1. Click Start, Programs, and Command Prompt.
   The DOS prompt appears.
2. From the DOS prompt, type **bash**.
3. Click Start, Run, and type **bash -login**.
   You can start a CLI session from a bash shell prompt.
4. Start the SpectroSERVER to which you want to connect.

5. Navigate to the vnmsh directory in the DX NetOps Spectrum installation directory:
   ```
   $ cd <$SPECROOT>/vnmsh
   ```
6. Open the connection:
   ```
   $ connect
   ```
   You are connected to the CLI session.

Example Usage

The examples in this section demonstrate how to use CLI commands for common tasks in DX NetOps Spectrum.

Create a User Model

A User model gives a user access to DX NetOps Spectrum. Users are identified by login IDs.

> **NOTE**
> Before you start a CLI session, verify that the User Model is created and the SpectroSERVER to connect is started.

Follow these steps:

1. Connect to the SpectroSERVER.
   ```
   $ cd <$SPECROOT>/vnmsh
   $ ./connect
   ```
   You are connected to the SpectroSERVER.

   > **NOTE**
   > If you have trouble connecting, verify for error messages. For more information, see Error Messages.
2. Determine the model type handle for the model type you want to create using the show command. In this case, it is a model of type User. Enter this command:
   ```
   $ ./show types | grep User
   ```

   > **NOTE**
   > The “./” is important. Some UNIX systems use the show command for reading email. If the “.” is not the first path in the user's environment, then "./" are required.

   A list of model types that include the string 'User' appears with the User model type listed first.
   ```
   Handle     Name          Flags
   0x10004    User          V,I,D
   0x1040a    UserGroup     V,I,D
   0x1040f    DefUserGroup  V,I,N,U,R
   0xaa000d   GenSwUserPort V,I,D
   0xf000d    ForeUserAgen  V,I,D
   0xaf000c   ForeUserApp   V,I,D
   ```
3. List the attributes using the show command for the User model type and determine the attribute ID for the model name attribute. You need this attribute ID to create the model. Enter this command:
   ```
   $ ./show attributes mth=0x10004 | grep -i name
   ```
   A list of User model type attributes including the model name attribute appears.
   ```
   Id        Name                  Type          Flags
   0x10000   Modeltype_Name        Text String   R,S,M
   0x1006e   Model_Name            Text String   R,W,G,O,M,D
   0x10074   User_Full_Name        Text String   R,W,O,D
   0x1155f   gib_mtype_nameText    String        R,W,S,D
   0x11560   gib_mtype_name_menu   Text String   R,W,S,D
   0x11561   gib_model_name        Text String   R,W,D
   ```

```
0x11563   gib_model_name_menu   Text String   R,W,D
0x1197d   WatchNames            Tagged Octet  R,W,D
```

4. Create the model using the create command with model type handle, the attribute ID for the model name, and the value (the login ID name) for the user. In this example, the user login ID is j_doe. Enter this command:

   ```
   $ ./create model mth=0x10004 attr=0x1006e,val=j_doe
   ```

   A system message resembling the following command confirms that the model is created:

   ```
   created model handle = 0xbe0001b
   ```

   NOTE
   All handles and IDs used in these examples are fictitious. The model handle for the model that you created is different; model handles are created by the system.

Modify a Model Attribute

This section provides an example to change the value of a model attribute using CLI commands. In particular, this example demonstrates how to change the community string attribute value for the model (j_doe) created in a User Model. For more information, see Create a User Model.

Follow these steps:

1. Determine the j_doe model handle, and then set j_doe as the current model:
   a. $ ./show models | grep j_doe
      The following information about the j_doe model appears:

      ```
      0xbe0001b        j_doe(Active)        0x10004        User
      ```

   b. $ ./current mh=0xbe0001b
      The system confirms that j_doe is the current model:

      ```
      current model is 0xbe0001b
      current landscape is 0xbe00000
      ```

2. Determine the ID for the community string attribute.

   NOTE
   For the sake of brevity, this step shows a known portion (community string) of the attribute name as an argument to the grep command. If you do not know the name of the attribute, you can show and scan all attributes for the model to determine the correct attribute name and its attribute ID.

3. Enter this command:

   ```
   $ ./show attributes | grep -i community_string
   ```

   The attribute ID, the attribute name, and the community string value appear:

   ```
   0x1007a        User_Community_String        ADMIN,0
   ```

   DX NetOps Spectrum assigns a default value of ADMIN,0 to all user models when they are created. ADMIN,0 confers full administrative privileges in DX NetOps Spectrum to user models.

4. Change the administrative permission level from ADMIN,0 to example permission level ADMIN,5 (read-only) for the j_doe model using the update command. Enter this command:

   ```
   $ ./update attr=0x1007a,val=Subnet3,5
   ```

   An entry showing the change in the attribute value is returned:

   ```
   Id        Name                   Iid    Value
   0x1007a   User_Community_String         ADMIN,5
   ```

   The Iid attribute has no value here because it applies only to list attributes. For more information, see the *DX NetOps Spectrum* Administration section.

Create and Modify a Model in One Step

This section provides an example of how to create a model and replace a default attribute value with another value in a single command string. You can only execute a complex command of the type shown in this section if you know the values of the relevant model identifiers that are provided for the command before you attempt to execute.

The following example uses the parameter values introduced in Creating a User Model and Modifying a Model Attribute:

```
$ ./create model mth=0x10004 attr=0x1006e,val=j_doe attr=0x1007a,val=ADMIN,5
```

Sample CLI Script File - Create a New User

> **NOTE**
> You can execute shell scripts that incorporate CLI commands from the bash prompt in the Windows platform simply as you execute the command from the shell command prompt on UNIX.

This following example demonstrates how a script can be used to create a DX NetOps Spectrum user model.

```
#
#  Check to see if CLIPATH is set.  If it is not then we will have to create it.
#
#  Setup a variable to point to the /install_area/vnmsh directory so we can
#  find the commands we need.
#
if [ -z "$CLIPATH" ]
then
    CLIPATH=/usr/data/spectrum/7.0/vnmsh
    export CLIPATH
fi
#
#  Test to make sure the CLIPATH points to a valid directory
#
if [ ! -d $CLIPATH ]
then
       echo "ERROR: could not find $CLIPATH"
       echo "Please find the correct path to the vnmsh directory and set"
       echo "the CLIPATH environment variable to it."
       exit 0
fi
#
#  Now check to see how many command line arguments there are.  If there are
#  none, then echo a usage message.  If there is one, that is all we really
#  need to create a new user...  If there is a second argument then we can
#  set the Community_String at the same time.
#
#  This setup is only for creating a user on the local system or what the
#  .vnmshrc file points to for the vnm_hostname.  A third field could be
#  added that accepts the vnm_hostname to connect to.
#
#  Optionally, the getopts shell command can be used to parse "switches" to
#  the script:  -n for name, -c for community string and -v for vnm_hostname.
#
#  (NOTE:  getopts should be located in /usr/bin/getopts if the script is
#   done in bourne shell (sh).  k-shell has a built in getopts function)
#
```

```
if [ $# -eq 0 ]
then
       echo "Usage:  $0 username [Community_String]"
       exit 1
elif [ $# -eq 1 ]
then
       command="attr=0x1006e,val=$1"
       flag=0
elif [ $# -eq 2 ]
then
       command="attr=0x1006e,val=$1 attr=0x1007a,val=$2"
       flag=1
fi

#
#  Okay, we should be all set now to go ahead and create the new user.
#  The first thing we have to do is connect.
#
$CLIPATH/connect
#
#  Now let's check the exit status of the connection to see if we got in...
#
if [ $? -ne 0 ]
then
       echo "ERROR:  could not connect to <ss>.  $0 exiting"
       exit 0
fi
#
#  Okay if we made it this far then we have a connection.  Let's try the
#  create command.
#
$CLIPATH/create model mth=0x10004 $command
#
#  Now we check the exit status again and see if we actually created a model.
#
if [ $? -ne 0 ]
then
       echo "ERROR:  could not create a new user.  $0 exiting"
       exit 0

else
       echo -n "New user $1 created"
       if [ $flag -eq 1 ]
then
       echo " Community_String was set to $2"
fi
       echo "Successfully created new model... exiting."
fi
$CLIPATH/disconnect
exit 1
```

Event Report Generation

The CLI keeps a list of the 2000 most current events that occur on a landscape. However, if many events occur on a landscape, the most recent events are approximately one hour old.

You can set the SPECROOT environment variable when using the -x option with the show events command. The following command is an example of to run event reports using CLI:

```
$ ./show events | more
$ SPECROOT=/home/spectrum; export SPECROOT
$ ./show events -x > event_rpt
```

Model Switch

The jump and setjump commands are useful in scripts where you can move back and forth between different models. The setjump command lets you assign a text string to represent a model handle and its corresponding landscape handle. Then you can use the jump command with that text string to retrieve that information as the current model handle. For example:

- $ ./current mh=0xb6000f8
  ```
  current model is 0xb6000f8
  current landscape is 0xb600000
  ```
- $ ./setjump emme
  ```
  model 0xb6000f8 and landscape 0xb600000 stored under emme
  ```
- $ ./jump emme
  ```
  current model is 0xb6000f8
  current landscape is 0xb600000
  ```

Create a Troubleshooter Model

You can create and associate Troubleshooter models with User models using the CLI. Once created and associated, these troubleshooters can be assigned alarms and receive email notification that they must investigate and resolve the alarms.

The following procedure describes how to create a TroubleShooter model and associate it with a User model. The User model that is created in Create a User Model is used as an example.

Follow these steps:

1. Navigate to the <*$SPECROOT*>/vnmsh directory.
2. Connect to the SpectroSERVER by typing the following command at a command prompt (example from a bash shell with a $ prompt):
   ```
   $ ./connect
   ```
3. Determine the TroubleShooter model type. Enter this command:
   ```
   $ ./show types | grep -i trouble
   ```
   The TroubleShooter model type entry is returned.
   ```
   0x10372        TroubleShooter       V,I
   ```
4. Determine the TroubleShooter model type EmailAddress attribute ID. Enter this command:
   ```
   $ ./show attributes mth=0x10372 | grep -i email
   ```
   The EmailAddress entry appears:
   ```
   0x11d24        EmailAddress         TextString    R,W,D
   ```
5. Create a TroubleShooter model using the CLI create command:
   ```
   $ ./create model mth=0x10372
   ```

```
attr=0x1006e,val=j_doe_fixit
attr=0x11d24,val=j_doe@aprisma.com
```

A system message resembling the following command confirms that the model is created:

```
$ created model handle = 0xbe0001c
```

NOTE
All handles and IDs used in examples are fictitious. The model handle for the model you created are different. It is whatever your system creates for it.

6. Create the association between the j_doe User model (mh=0xbe0001b) and the j_doe_fixit TroubleShooter model (mh=0xbe0001c) using the CLI create command:

```
$ ./create association rel=Is_Assigned lmh=0xbe0001b rmh=0xbe0001c
```

A system message similar to the following command confirms that the association was created:

```
$ create association successful
```

Assign an Alarm to a Troubleshooter

This section describes how to list alarms and assign an alarm to a troubleshooter using CLI commands.

Follow these steps:

1. List alarms using the show command.
   This step shows how to find only those alarms with an alarm_severity of MAJOR.

```
$ ./show alarms | grep MAJOR
```

A list of MAJOR alarms appears. For example:

```
7509   09/27/2000   14:46:44   0xd80008   0xa6000df   duncan    9E133_36   MAJOR   No
7645   09/27/2000   14:47:16   0xd80008   0xa60025e   infinity  9H422_12   MAJOR   No
7518   09/27/2000   14:47:01   0xd80008   0xa6000eb   rugone    9E132_15   MAJOR   No
7979   09/27/2000   14:53:12   0xf40002   0xa600161   FDDI2     FddiMAC    MAJOR   No
8018   09/27/2000   14:53:13   0xf40002   0xa6003da   FDDI FNB  FddiMAC    MAJOR   No
7512   09/27/2000   14:46:47   0xd80008   0xa6000af   ruthere   9A426_02   MAJOR   No
```

1. Select an alarm to which you want to assign a troubleshooter. In this example, alarm ID 7512 for the 9A426-02 device is selected.
2. Select a troubleshooter to assign to the alarm. In this example, the j_doe_fixit TroubleShooter model that was created in Create a Troubleshooter Model is selected.

NOTE
Use the TroubleShooter model handle, 0xa600722, rather than the TroubleShooter model name, j_doe_fixit, to specify the troubleshooter in the update command in the next step.

3. Assign the alarm to the troubleshooter using the update command:

```
$./update alarm aid=7512 assign=0xa600722
```

A system message similar to the following example confirms that the troubleshooter was assigned to the alarm:

```
$ update: successful
```

The person who is represented by the j_doe_fixit model has now been assigned to the alarm. This person receives an email notification of the alarm assignment.

Create a Global Collection

You can create a global collection and can set the search criteria in a CLI session using GUID. A unique identifier (GUID) is a key attribute to create global collection. The GUID is required for global collection to function properly. You can obtain a GUID through an action to the VNM model.

NOTE
A global collection without a GUID in the CLI is invalid. When a global collection is created using OneClick, a GUID is automatically created. For more information, see the Modeling and Managing Your IT Infrastructure section.

Follow these steps:

1. Enter the following command:
   ```
   update action=(Action to get a unique identifier) 0x10474 mh= (VNM Model Handle)
   ```
   A GUID is created.

   NOTE
   The action to get a new GUID is 0x10474, which is the same as the global collection model type.
2. Enter the following command to create the global collection using GUID:
   ```
   create model mth=(Model Type for global collection) 0x10474 attr=(GUID) 0x12e56, val=(Previous value you
    received) 4a85b9af-0d52-1000-017f-0013727f8c0a
   ```
   The Global Collection is created and appears in the Navigation pane under global collections.

Example: Create a global collection with or without XML string

Enter the following command to create a global collection with or without XML string:

```
update action=(Action to get GUID)0x10474 mh=(Landscape)
Printed:
update action: successful
Response has 1 attributes:
 0) Attribute 0x0 text: EXAMPLE GUID(4a85b9af-0d52-1000-017f-0013727f8c0a)
create model mth=(Model Type for global collection)0x10474
attr=(GUID)0x12e56,val=(Previous value you received)4a85b9af-0d52-1000-017f-0013727f8c0a
attr=(dynamicCriteriaXML)0x12a6a,val=(XMLString)'<search-criteria><filtered models><equals-ignore-case><model-
name>sometext</model-name></equals-ignore-case></filtered-models></search-criteria>'
Printed:
created model handle = New model handle(0x78101069)
```

NOTE
You can specify the dynamicCriteriaXML (0x12a6a) attribute with the create command or you can update the model later.

Suppress Headers in CLI Output

To suppress the headers in CLI output, you can create a file that includes the functions provided in the following section and can refer this file at the top of each script.

The functions in the following procedure call CLI commands and strip the header information from the output of the commands.

Follow these steps:

1. Create a file named StripHeaders in your scripts directory.
2. Include the following functions in the StripHeaders file:
   ```
   tcreate() # only needed for the createalarm
       { # and create event commands
       $CLIPATH/create $@ | tail +2
       }
   tseek()
       {
   ```

```
        $CLIPATH/seek $@ | tail +2
    }
tshow()
    {
    $CLIPATH/show $@ | tail +2
    }
tupdate()
    {
    $CLIPATH/update $@ | tail +2
    }
```

3. Include the name StripHeaders at the top of your CLI script as follows:
   ```
   . StripHeaders
   ```
4. Call the tcreate(), tseek(), tshow(), and tupdate() functions instead of the corresponding CLI command whenever you want to strip the headers from CLI output.
   For example, the following line generates the output of the show models command without the CLI header information:
   ```
   tshow models
   ```

Command Descriptions

This section provides descriptions of CLI commands and output:

Command Descriptions Overview

You can make changes to the DX NetOps Spectrum knowledge base without the safeguards that are available in DX NetOps Spectrum using the CLI. A system crash or database corruption can result if you specify incorrect information. Therefore, proceed with caution when you use the create, destroy, or update commands.

> **NOTE**
> Use the CLI command parameters for creating and managing response time tests with CLI.

For more information, see the Service Performance Manager section.

ack alarm - Acknowledges Alarm

The ack alarm command acknowledges the alarm specified by alarm_id in the landscape that is specified by landscape_handle. If landscape_handle is not specified, the command acknowledges the alarm that is specified by alarm_id in the current landscape.

Acknowledging one alarm for a model means that you acknowledge only that alarm and no other alarm for that model.

The command has the following format:

```
ack alarm aid=<alarm_id> [lh=<landscape_handle>]
```

If ack alarm is entered with a valid alarm_id and a valid landscape_handle, the following message is displayed:

```
ack alarm: successful
```

Example: ack alarm

```
$ ack alarm aid=42 lh=0x400000
ack alarm: successful
```

Start, Re-start or Connect - Connects to SpectroSERVER

The start, restart or connect command connects the user of the DX NetOps Spectrum Command Line Interface to the SpectroSERVER running on host system, *hostname.* This command also sets the landscape that is specified by landscape_handle to be the current landscape. If the CLI Local Server is not already running, the connect command starts it.

The command has the following format:

```
connect [<hostname>] [lh=<landscape_handle>][vnmsocket=<vnmsocket>]
```

- **hostname**
  (Optional) If hostname is not specified, the command connects the user to the host specified in the CLI resource file .vnmshrc.

  NOTE
  DX NetOps Spectrum Command Line Interface does not support localhost or the 127.0.0.1 option. To connect to localhost, you can specify the actual hostname or do not specify any parameter.
- **landscape_handle**
  (Optional) If landscape_handle is not specified, the command sets the current landscape to the landscape of the host name specified.
- **vnmsocket**
  (Optional) If vnmsocket is not specified, the command connects to the SpectroSERVER using the socket specified in the .vnmshrc file. You can use vnmsocket to connect to another SpectroSERVER on a different port connection that is defined by vnmsocket.

On Unix, error messages that are reported by the CLI Local Server are displayed in the console window. On Windows, these errors are displayed in the user bash shell window.

Example: connect

```
#! /usr/bin/sh
# A sample script to get alarms of a specific
# severity and set the CLISESSID

if [$# !=1]
then
   echo "Usage: $0 <alarm severity>"
   exit 0
fi

CLISESSID=$$

$SPECROOT/vnmsh/connect
$SPECROOT/vnmsh/show alarms | grep -i $1
$SPECROOT/vnmsh/disconnect

exit 0
```

If the command is successful, the following message is displayed:

```
connect: successful hostname
current landscape is <landscape_handle>
```

Hostname is the user-entered SpectroSERVER host or the host that is specified in the .vnmshrc file. landscape_handle is the user-entered landscape or the landscape for the host.

Considerations when Using connect Command

The following are important considerations when using the connect command:

- The user on a terminal must use the connect command to initiate communications. The same user must use the disconnect command to terminate communication with the SpectroSERVER.
- Once the first user has entered the connect command, the CLI Local Server is connected to the SpectroSERVER.
- Other CLI users using the same CLI Local Server can connect only to those SpectroSERVERs that are in the landscape map of the initial SpectroSERVER. Once all users have disconnected, use the connect command to connect to SpectroSERVERs in a different landscape map.
- To successfully connect to the SpectroSERVER, the first user of the connect command must be defined as a user in the DX NetOps Spectrum database of the original SpectroSERVER.
- Windows users running CLI in the bash shell must also define CLISESSID.
- The terminal device for a particular user is determined using the ttyslot(3V) function.
- Cron scripts are not attached to a ttyslot. As a result, the ttyslot function returns 0 for all cron scripts. That is, two CLI scripts running as cron scripts at the same time appear as one CLI user to the CLI Local Server, which leads to unpredictable results. Therefore, you must insert a line at the top of the script to export the environment variable CLISESSID. Set CLISESSID to a unique numeric value. CLI can now distinguish between the different cron scripts. The following example defines a unique CLI session ID within a script:
  ```
  CLISESSID=$$; export CLISESSID
  ```
- This example sets the CLISESSID as the process ID of the shell running the script. CLI uses CLISESSID to identify a user when the ttyslot function returns zero. Set CLISESSID once for each CLI session. If a CLI script that is running as a cron script calls other CLI scripts, only the top-level script will set the CLISESSID environment variable. The other CLI scripts run under the same process ID unless you invoke a new shell (#!/bin/sh) at the top of the script. To invoke a new shell in the other scripts, export the CLISESSID and then connect and disconnect again.
- In some environments or configurations, even when the command is entered from a command line, the ttyslot function may return zero. In such a case, the connect command returns the following error:
  ```
  connect: variable CLISESSID not set
  ```
  In this situation, set CLISESSID either from the command line or from .cshrc or another startup file.
- The CLI uses the user name and terminal device to identify each CLI user. A user running multiple scripts from a terminal device at one time will appear to CLI as the same user. The CLI can give unpredictable results if a script is running in the background and another script is running in the foreground, or multiple scripts are running in the background.
  For example, Script A1 sets the current model to be Model A. Script B1, which is run by the same user from the same terminal device, sets the current model to be Model B. If Script A1 performs an update command on Model A, the update command is also performed on Model B of Script B1.
  Run only one CLI session from a particular terminal device at one time. To run multiple CLI sessions at once, run them from a separate terminal device, or run them using the at(1) or batch(1) commands with the CLISESSID environment variable set to a unique value for each.

create - Create Object

Use the create command to create an object.

> **NOTE**
> For information about how to create a model in a secure domain, run ./create to display a usage statement.

The command has the following format:

```
create  model ip=<IP Address | Low_IP-High_IP> [sec_dom=Secure_Domain_Address][comm=Community_Name]
 [to=Time_Out] [tc=Try_Count] [lh=landscape_handle] |
create  model mth=model_type_handle [attr=attribute_id,val=value ...] [lh=landscape_handle] |
```

```
create  association rel=relation lmh=left_model_handle rmh=right_model_handle
create  alarm [-nr] sev=alarm_severity cause=probable_cause_id [mh=model_handle] |
create  event type=event_type text=event_text [mh=model_handle|lh=landscape_handle]
```

create alarm

The command *create alarm* creates an alarm with severity alarm_severity and cause probable_cause_id for the model with model_handle. Valid alarm severity options are: CRITICAL, MAJOR, MINOR, OK, MAINTENANCE, SUPPRESSED, or INITIAL. By default, the new alarm replaces an existing alarm.

If create alarm is entered with a valid alarm_severity, a valid probable_cause_id, and a valid model_handle, the created entry in the alarm table is displayed. The create time is displayed in hh:mm:ss format.

Example: create

```
$ create alarm sev=CRITICAL cause=0x10308 mh=0x400134
ID     Date          Time        PCauseID    MHandle     MName    MTypeName     Severity    Ack
984    05/11/2000    12:33:27    0x10308     0x400134    12.84    Bdg_CSI_CN    CRITICAL    No
```

create association

The command *create association* creates an instance of the relation (an association) between the model with left_model_handle and the model with right_model_handle.

If create association is entered with a valid relation between a valid left_model_handle and right_model_handle, the following message is displayed:

```
create association: successful
```

Example: create association

```
$ create association rel=Collects lmh=0x400009 rmh=0x400134
create association: successful
```

create event

The create event command, creates an event with type event_type and text event_text for the model that is specified by model_handle. If a landscape_handle is specified, the event is created for the user model that created the event.

> **NOTE**
> In previous versions of the CLI, the event was created for the landscape model.

If model_handle or landscape_handle is not specified, the event is created for the user model that created the event. Or the event is created for the current model if one has been specified. Some events in DX NetOps Spectrum lack an associated model. For example, when an application connects to the SpectroSERVER, no model is associated with the event.

If create event is entered with a valid event_type, valid event_text, and a valid model_handle or landscape_handle (if present), the entry appears in the event table. The create time is displayed in hh:mm:ss format.

The event_type command (also named an event code in DX NetOps Spectrum) is a 4-byte hexadecimal number. The two most significant bytes specify the developer ID for the event (0001 for DX NetOps Spectrum-generated event codes), and the two least significant bytes are a unique event identifier. Not all event types include user-entered text. Examples of such event types are those that include the variable {S 0} in their event format files. For those event types that do not include user-entered text, the event_text parameter is ignored but must still be present on the command line.

For more information, see the Certifications section.

Example: create event

```
$ create event type=0x1061a text="fan down" mh=0x40013
Date          Time        Type       MHandle     MName    MTypeName
05/11/2000    12:39:42    0x1061a    0x400134    12.84    Bdg_CSI_CNB20
```

create model

You can specify create model with an IP address or with a model type handle. In either case, the system creates the model in the landscape that is specified by landscape_handle. If landscape_handle is not specified, the command creates the model in the current landscape.

NOTE
The model_name attribute is required only when creating a User model.

- If you specify create model with an IP address, the system finds the object at the specified ip_address and creates a model for it. The model has all of the properties of that object including any associated children. For example, if the object is a hub, the create model command creates a model of a hub with all of its ports.
- You can specify an IPv4 address or an IPv6 address. IPv6 ranges are not supported and this command does not support the setting of attribute IDs.
- To create several models at once, you can define a range of IP addresses with the create model command. Specify the Low_IP and High_IP parameters, separated by “-”. If the Community_Name is not specified, the newly created model is of type "Pingable". If the Community_Name is specified, the device is modeled to the appropriate model type. The Try_Count and Time_Out options are similar to options in the OneClick 'Create Model by IP' dialog.
- If you specify the create model command with a model type handle, the system creates a model of type model_type_handle. You can then set the value of one or more attributes for the created model.
- When you specify the create model command with a model type handle, you can also specify multiple attributes in that one command. Specify multiple 'attribute_id, value' pairs, separating each pair from adjacent pairs by a space.
- The attribute values that the user specifies when creating a model of a particular model type while using OneClick should be specified in the create model command. Otherwise, Inference Handler errors can occur within a SpectroSERVER when the model is created. For example, when creating a Hub_CSI_IRM3 model using OneClick, a window is displayed in which you can enter values for Model Name, Network Address, Community String. Specify values for these attributes when using the create model command to create a model of the same type using the CLI.
- If create model is specified with a valid model_type_handle and valid attribute_id, value pairs (if present), the created model handle is displayed.
- If create model is specified with a valid ip_address, the created model handle is displayed.

Example: create model

```
$ create model mth=0x102d attr=0x12d7f,val=132.177.12.84 attr=0x1006e,val=12.84lh=0x400000
created model handle = 0x400134
$ create model ip=206.61.231.1-206.61.231.5
Creating model for IP=206.61.231.1
created model handle =  0x9a00259
Creating model for IP=206.61.231.2
create model: DCM device unreachable
Creating model for IP=206.61.231.3
create model: DCM device unreachable
Creating model for IP=206.61.231.4
create model: DCM device unreachable
Creating model for IP=206.61.231.5
created model handle =  0x9a0025a
```

NOTE
By default, the create command displays a maximum of 16 characters for the model name. However, with the environment variable, CLIMNAMEWIDTH, you can specify a different number of characters (up to 1024) to be displayed for model names.

current - Sets Model or Landscape

The current command sets the model that is specified by model_handle to be the current model. Or this sommand sets the landscape that is specified by landscape_handle to be the current landscape. If the model_handle and the landscape_handle are not specified, current displays the current model handle and the current landscape handle.

When the user sets a current model, the CLI sets the current landscape to the landscape that contains the model. When a user sets the current landscape, the CLI sets the current model as undefined.

Separate current model and current landscape values are maintained for each session that is connected to the CLI Local Server.

The current command retains state information, the current model and the current landscape, for example, only for the session that named it.

This command has the following format:

```
current [mh=<model_handle>|lh=<landscape_handle>]
```

- If a valid model_handle is specified as input, the following message is displayed:
  ```
  current model is <model_handle>
  current landscape is <current_landscape_handle>
  ```
- If a valid landscape_handle is specified as input, the following message is displayed:
  ```
  current model is undefined
  current landscape is <landscape_handle>
  ```
- If model_handle and landscape_handle are not specified, the following message is displayed:
  ```
  current model is <current_model_handle>
  current landscape is <current_landscape_handle>
  ```
- If model_handle and landscape_handle are not specified and current model is not defined, the following message is displayed:
  ```
  current model is undefined
  current landscape is <current_landscape_handle>
  ```

Examples: current

```
$ current mh=0x400142
current model is 0x400142
current landscape is 0x400000
$ current lh=0x500000
current model is undefined
current landscape is 0x500000
$ current
current model is undefined
current landscape is 0x500000
```

NOTE
The current landscape always contains a value because it is set by the connect command.

destroy - Destroys Object

Use the destroy command to destroy an object. This command has the following format:

```
destroy model [-n] mh=model_handle |
destroy association [-n] rel=relation lmh=left_model_handle rmh=right_model_handle|
destroy alarm [-n] aid=alarm_id [lh=landscape_handle]
```

- **-n**
  If the -n (no prompt) option is specified with the destroy command, the system does not prompt for confirmation. This option is useful in CLI scripts.
  Unless the -n option is specified, one of the following messages is always displayed:
  ```
  destroy model: are you sure?
  destroy association: are you sure?
  destroy alarm: are you sure?
  ```
  Valid responses are y, yes, Y, Yes, n, no, N, and No.

destroy alarm

Destroys the alarm specified by alarm_id in the landscape that is specified by landscape_handle. Unless the -n option is specified, destroy alarm prompts you for confirmation before destroying the alarm. If the landscape_handle is not specified, the command destroys the alarm that is specified by alarm_id in the current landscape. Use the show alarms command to determine the alarm_ids for a model.

If the destroy alarm command is entered with a valid alarm_id and a valid landscape_handle, the following message is displayed:

```
destroy alarm: successful
```

Examples: destroy alarm

```
$ destroy alarm aid=300
destroy alarm: are you sure? y
destroy alarm: successful
```

destroy association

Destroys the association (instance of the relation) between the model with left_model_handle and the model with right_model_handle. Unless the -n option is specified, destroy association prompts you for confirmation before destroying the association.

If destroy association is entered with a valid relation between a valid left_model_handle and right_model_handle, the following message is displayed:

```
destroy association: successful
```

Example: destroy association

```
$ destroy association rel=Lost_and_Found lmh=0x400001 rmh=0x40h0142
destroy association: are you sure? y
destroy association: successful
```

destroy model

Destroys the model with the specified model_handle. Unless the n option is specified, destroy model prompts you for confirmation before destroying the model.

If destroy model is entered with a valid model_handle, the following message is displayed:

```
destroy model: successful
```

Example: destroy model

```
$ destroy model mh=0xa600715
Following model will be destroyed:
Model_Handle      ->     0xa600715
Model_Type_Handle ->     0x10004
Model_Name        ->     garciaparra
Model_Type_Name   ->     User
destroy model: are you sure? y
destroy model: successful
```

disconnect - Disconnects from SpectroSERVER

Use the disconnect command to disconnect the CLI user from the currently connected SpectroSERVER.

This command has the following format:

```
disconnect
```

If the command is successful, the following message is displayed, where host name is the name of the SpectroSERVER host to which the user was connected:

```
disconnect: successful from <hostname> or <IP address> - connected for xx hours, yy minutes
```

jump - Jumps to Saved Model or Landscape

The jump command jumps to the previously saved model and landscape. The jump command sets the current model and the current landscape to be the model and landscape that were saved under the label text_string by the setjump command. If text_string is not specified, a list of text_strings that were given in previous setjump commands is displayed.

The command has the following format:

```
jump [<text_string>]
```

- If jump is entered with a valid text_string that has been previously defined, the new current model and the current landscape are displayed:
  ```
  current model is <current_model_handle>
  current landscape is <current_landscape_handle>
  ```
- If jump is entered without a text_string, a list of the currently defined text_strings is displayed. For example:
  ```
  text_string1
  text_string2
  --
  ```
- If jump is entered and the new current model is undefined, the following message is displayed:
  ```
  current model is undefined
  current landscape is <current_landscape_handle>
  ```

Example: jump

```
$ jump tutorial
current model is 0x400142
current landscape is 0x400000
```

seek - Locates a Model

Use the seek command to locate a model. The seek command finds the model(s) in the landscape that is specified by landscape_handle that possess the specified value for the attribute that is specified by attribute_id. If landscape_handle is not specified, the command finds the model(s) in the current landscape that possess a value for the attribute with the specified attribute_id. You can also use a wildcard (*) with seek to find instances of models that contain a specified substring. If you enter a null value, you can find all models that have no name (for example, attr=0x1006e).

You cannot search for a one-character attribute value using the seek command. Attempting such a search returns an error.

The command has the following format:

```
seek [-i] [-s]  attr=attribute_id,val=value [lh=landscape_handle]
```

> **NOTE**
> The options can be used in any order, for example, -i -s, or -s -i.

- **-i**
  If the -i (ignore case sensitivity) option is specified with the seek command, then the model information that is specified with the val parameter is returned without regard to case.
- **-s**
  If the -s (substrings allowed) option is specified with the seek command, the model information that is specified with the val parameter is returned with substrings, if applicable.
  If seek is entered with a valid attribute_id and a valid value, all matching models are displayed in the following format:

  ```
  MHandle      MName   MTypeHnd         MTypeName
  modelhandle  name    modeltypehandle  name
  ```

  If no matching models are found, the following message is displayed:

  ```
  seek: no models found
  ```

  > **NOTE**
  > By default, the seek command displays a maximum of 16 characters for the model name. However, with the environment variable CLIMNAMEWIDTH, you can specify a different number of characters (up to 1024) to be displayed for model names.

Examples: seek

```
$ seek attr=0x1006e,val=<sp>
MHandle       MName      MTypeHnd   MTypeName
0xb100018     spectrum    0x1004     User
0xb10008d     spectrum    0x820000   ScmConfig
$ seek attr=0x1006e,val=<sp>
MHandle       MName      MTypeHnd   MTypeName
0xb100018     spectrum    0x820000   ScmConfig
$ seek attr=0x1006e,val=SPE
seek: no models found
$ seek attr=0x1006e,val=spe lh=0xb100000
seek: no models found
$ seek -i attr=0x1006e,val=<sp>
MHandle       MName      MTypeHnd   MTypeName
0xb10018      spectrum    0x10004    User
0xb1008c      spectrum    0x820000   ScmConfig
0xb1008d      spectrum    0x820000   ScmConfig
$ seek -i -s attr=0x1006e,val=<sp>
MHandle       MName      MTypeHnd   MTypeName
0xb10018      spectrum    0x10004    User
```

```
0xb10089      spectrum    0x820000   ScmConfig
0xb1008c      spectrum    0x820000   ScmConfig
0xb1008d      spectrum    0x820000   ScmConfig
$ seek -i -s attr=0x1006e,val=<sp> lh=0xb100000
MHandle       MName      MTypeHnd   MTypeName
0xb10018      spectrum    0x10004    User
0xb10089      spectrum    0x820000   ScmConfig
0xb1008c      spectrum    0x820000   ScmConfig
0xb1008d      spectrum    0x820000   ScmConfig
$ seek -s attr=0x1006e,val=<sp>
MHandle       MName      MTypeHnd   MTypeName
0xb10008c     spectrum    0x820000   ScmConfig
$ seek attr=0x110df,val=0.0.C.18
seek: no models found
$ seek -s attr=0x110df,val=0.0.C.18
MHandle       MName        MTypeHnd      MTypeName
0xb100070     frog10       0x210022      Rtr_CiscoIGS
0xb100072     frog10_1     0x220011      Gen_IF_Port
0xb10005b     cisco rtr    0x210022      Rtr_CiscoIGS
0xb100070     frog10_2     0x220011      Gen_IF_Port
0xb100070     cisco rtr_1  0x220011      Gen_IF_Port
0xb100070     cisco rtr_2  0x220011      Gen_IF_Port
$ seek attr=0x1006e,val=spe*
MHandle       MName           MTypeHnd    MTypeName
0xb10018      spectrum         0x10004     User
0xb10089      spectrum         0x820000    ScmConfig
0xb1008d      spectrum         0x820000    ScmConfig
$ seek attr=0x1006e,val=
MHandle       MName        MTypeHnd      MTypeName
0xd00258      0x102c8                    Physical_Addr
0xd002f8      0x102c8                    Physical_Addr
0xd00368      0x820000                   ScmConfig
0xd00259      0x102c8                    Physical_Addr
0xd002f9      0x102c8                    Physical_Addr
0xd00301      0x102c8                    Physical_Addr
$ seek attr=0x12d7f,val=192.168.93.*
MHandle       MName               MTypeHnd      MTypeName
0x28000190    192.168.93.14       0xd0004       HubCSIEMME
0x28000190    192.168.93.14       0xd0004       HubCSIEMME
0x280001a0    192.168.93.14_Sy    0x23001c      System2_App
0x28000198    192.168.93.14_St    0x590006      RMONApp
0x28000191    192.168.93.14_A     0xd000a       CSIIfPort
0x280001a1    192.168.93.14_IC    0x230012      ICMP_App
0x28000199    192.168.93.14_E     0x590013      RMONEthProbe
0x280001a2    192.168.93.14_UD    0x230019      UDP2_App
0x280001c2    DLM App             0x830001      DLM_Agent
0x2800019a    192.168.93.14_E     0x590013      RMONEthProbe
0x28000192    192.168.93.14_B     0xd000a       CSIIfPort
0x2800019b    192.168.93.14_E     0x590013      RMONEthProbe
```

setjump - Saves Model and Landscape

The setjump command saves the current model handle and current landscape handle under the label text_string. The user can later use the jump command with text_string to set the current model handle and the current landscape handle back to the one stored under text_string. The user is prompted for verification if the same text_string is used in two setjump commands.

Separate setjump values are maintained for each session that is connected to the CLI Local Server. The setjump command retains information, that is, the session-assigned setjump text strings, only for the session that named it.

The command has the following format:

```
setjump [-n] <text_string>
```

- **-n**
  If the -n (no prompt) option is specified with the setjump command, then the system does not prompt if text_string has been used before.
- If setjump is entered with a new <text_string> and a current model exists, the following message is displayed:
  ```
  model <current_model_handle> and landscape
  <current_landscape_handle> stored under <text_string>
  ```
  where <current_model_handle> is the handle of the current model and <current_landscape_handle> is the handle of the current landscape.
- If setjump is entered with a new <text_string> and a current model does not exist, the following message is displayed:
  ```
  model undefined and landscape <current_landscape_handle>
  stored under <text_string>
  ```
- If setjump is entered with a text_string that has already been defined in a previous setjump command, the following message is displayed:
  ```
  setjump model: <text_string> already used. Overwrite?
  ```
  Valid responses are y, yes, Y, Yes, n, no, N, and No.

Example: setjump

```
$ current mh=0x400142
current model is 0x400142
current landscape is 0x400000
$ setjump -n tutorial
model 0x400142 and landscape 0x400000 stored under tutorial
```

show - Displays Object

To display objects, use the show command.

The command has the following format:

```
show models [mhr=low_model_handle-high_model_handle]
    [mth=model_type_handle][mname=model_name][lh=landscape_handle] |
devices [lh=landscape_handle]|
landscapes |
types [mthr=low_mth-high_mth] [mtname=mt_name]
         [flags=V|I|D|N|U|R] [lh=landscape_handle] |
relations [lh=landscape_handle] |
associations [mh=model_handle] |
parents [rel=relation] [mh=model_handle] |
```

```
children [rel=relation] [mh=model_handle] | attributes [-e]
         [attr=attribute_id[,iid=instance_id][,next]...|
         [attrr=low_attr-high_attr] [attrname=attr_name]]
         [mh=model_handle] |
attributes [-c] [-e]
         [attr=attribute_id[,iid=instance_id][,next]...|
         [attrr=low_attr-high_attr] [attrname=attr_name]]
         [mh=model_handle]|
attributes mth=model_type_handle [attrr=low_attr-high_attr]
         [attrname=attr_name]        [flags=E|R|W|S|T|G|O|M|D|P|L|V]
         [lh=landscape_handle] |
alarms [-a] [-x] [-t] [-s] [-d]
        [mh=model_handle|lh=landscape_handle] |
events [-x] [ -a | -n no_events ]
         [mh=model_handle|lh=landscape_handle] |
inheritance mth=model_type_handle [lh=landscape_handle] |
rules rel=relation [lh=landscape_handle] |
enumerations [attr=attribute_id] [mth=model_type_handle]
         [lh=landscape_handle] |
watch [mh=model_handle]
```

- **-a**
  If the -a (all) option is specified, show alarms do not perform any masking and displays all CRITICAL, MAJOR, MINOR, MAINTENANCE, SUPPRESSED, and INITIAL alarms.
- **-x**
  If the -x (expand) option is specified (and the variable $SPECROOT is set), the output of the show alarms command displays the text for the probable causes at the end of the output. The output of the show events command displays event formats. The number of characters displayed by the show events -x command is controlled by the .vnmshrc resource file parameter.
- **-d**
  If the -d option is specified, the output of the show alarms command displays the "Title" for every alarm listed. This option also displays all that the option -x displays.
- **max_show_event_length**
  If you are using a SpectroSERVER-only workstation to run CLI, the -x option does not provide the normal alarm cause or event format information because the CsPCause and CsEvFormat files do not exist in the SG-Support directory. Possible errors messages are:
  - No cause information available (associated with show alarms)
  - No event format information available (associated with show events)

  To remedy this problem, copy the SG-Support/CsPCause and SG-Support/CsEvFormat directories and files to the <*$SPECROOT*>/SG-Support directory on the SpectroSERVER workstation.
  Default: 512
- **-e**
  If the -e (enumerations) option is specified, the output of the show attributes command displays database enumeration strings.
- **-c**
  If the -c (Read Most Current) option is specified, the attribute Read Mode is set to Read Most Current. This mode uses the attribute value from the latest user interface poll, which is updated every 5 seconds. If this flag is not set, the Read Most Available mode is used. This mode uses the latest value stored in the database by the last DX NetOps Spectrum poll. Polling frequency is a user-defined interval.
- **-n**

If the -n (number of events) option is specified, the output of the show events command displays the specified number of events.

- **-t**
  If the -t (trouble ticket id) option is specified, the output of the show alarms command displays the trouble ticket id field.
- **-s**
  If the -s (impact severity) option is specified, the output of the show alarms command displays the impact severity field.

NOTE
For the show alarms and show events commands to work with the -x option, which displays probable cause messages for alarms and expanded event messages, OneClick must be installed on the local server, and the SPECROOT environment variable must be set to the path of the spectrum support root directory. For example, if the SG-Support files are in /usr/spectrum/SG-Support, set SPECROOT to /usr/spectrum.

show alarm

The show alarms command shows all alarms for the model that is specified by model_handle. Or it shows only the most severe alarm (if the alarm is CRITICAL, MAJOR, or MINOR) for each model in the landscape that is specified by landscape_handle. If landscape_handle is specified, show alarms masks any models that have INITIAL, SUPPRESSED, or MAINTENANCE alarms. As a result, only models with CRITICAL, MAJOR, or MINOR alarms are displayed. If neither model_handle nor landscape_handle is specified, show alarms also performs masking and shows only the most severe alarm (if the alarm is CRITICAL, MAJOR, or MINOR) for each model in the current landscape.

The Ack field indicates whether the alarm has been acknowledged. The possible values for this field are Yes and No. The Stale field indicates whether an alarm is stale. The possible values for this field are Yes and No. The Assignment and Status fields show the alarm troubleshooter information and the alarm status, respectively. The alarm creation time is displayed in hh:mm:ss format.

NOTE

To synchronize event and alarm support files with other OneClick servers, in a distributed environment, copy the folders containing the event format files and the probable cause files to the other SpectroSERVERs. $SPECROOT/custom/Events/CsEvFormat $SPECROOT/custom/Events/CsPCause Also copy the contents of these same directories to a custom directory on all of the SpectroSERVERs in your environment.

The show alarms command displays information in the following format:

```
Id  Date        Time      PCauseId  MHandle  MName  MTypeName  Severity  Ack  Stale  Assignment  Status
id  mm/dd/yyyy  hh:mm:ss  cause_id  handle   name   name       severity  ack  stale  assignment  status
```

If show alarms is used with the -x option, a table of cause codes and probable cause text messages is displayed after the last alarm. For example:

0x10402 DUPLICATE PHYSICAL ADDRESS0x10302 SpectroSERVER has lost contact with this device.

NOTE
The show command displays a maximum of 16 characters for the model name. However, with the environment variable CLIMNAMEWIDTH, you can specify a different number of characters (up to 1024) to be displayed for model names.

Example: show alarms

The show alarms command displays information in the following format:

```
$ show alarms lh=0x110000
ID   Date        Time      PCauseId   MHandle     MName     MTypeName  Severity    Ack  Stale   Assignment
 Status
928  05/11/2000  02:33:22  0x10c04    0x110000c   infinity  VNM        CRITICAL    No   No      McDonald
 Working on it
```

show association

The show associations command shows all instantiated relations (associations) that are defined for the model with model_handle. If model_handle is not specified, show associations shows all instantiated relations for the current model.

The show associations command displays information in the following format:

```
LMHandle   LMName   Relation   RMHandle   RMName
handle     name     relation   handle     name
```

Example: show associations

The show associations command displays information in the following format:

```
$ show associations mh=0x400141
LMHandle   LMName     Relation        RMHandle   RMName
0x400001   LostFound  Lost_and_Found  0x400141   12.77-bridge
```

show attributes

The show attributes command shows the attributes specified by attr=attribute_id for the model with model_handle. If no attribute_id is specified, show attributes lists all attributes and their values for the model with model_handle.

If model_handle is not specified, show attributes shows all applicable attributes for the current model. You can specify a range of attributes using attr=low_attr-high_attr. The instance ID for an attribute can be specified in instance_id when displaying a single attribute or a list of attributes for a particular model. The instance_id must be a sequence of positive integers separated by periods. Instance IDs can only be specified for list attributes. List attributes are attributes that have the list flag set.

The following rules apply to list attributes:

- To display all attribute values and instance IDs for a list attribute, do not enter an instance_id with the attribute_id. Enter an attribute_id only.
- To display the first attribute value and instance ID for a list attribute, enter the following command after the attribute_id:
  ```
  ,next
  ```
- To display a specific attribute value and instance ID for a list attribute, enter an instance_id with the attribute_id.
- To display the next attribute value and instance ID after a specific instance ID of a list attribute, enter the following command after the instance_id:
  ```
  ,next
  ```
- An instance ID cannot be specified when displaying all the attributes of a model, for the following reasons:
  - An instance ID only applies to list attributes (for example, board and port attributes of a hub)
  - The instance ID for certain attributes of a model may differ from the instance ID of other attributes within the same model.
- The show attributes command shows all attributes (by ID, name, type, and flags) for model_type_handle in the landscape specified by landscape_handle. If landscape_handle is not specified, this command shows all model types that are defined in the current landscape. The Flags field lists the abbreviations of each attribute flags (separated by commas) that is currently set. If a flag is not set, its abbreviation is not included in the list.
  The following list includes the attribute flags and their abbreviations:

- External = E
- Readable = R
- Writable = W
- Shared = S
- List = T
- Guaranteed = G
- Global = O
- Memory = M
- Database = D
- Polled = P
- Logged = L
- Preserve Value = V

NOTE
For a more detailed description of the attribute flags, see the Model Type Editor section.

The show attributes command displays information in the following format:

```
Id      Name      Iid      Value
id      name      iid      value
```

The show attributes mth command displays information in the following format:

```
Id      Name      Type      Flags
id      name      type      flags
```

Example: show attributes

```
$ show attributes mh=0xcd00011
Id          Name                        Iid      Value
0xd0000     Modeltype_Name                       User
0x10000     Modeltype_Handle                     0x10004
0x10004     Contact_Status                       1
0x10009     Security_String                      ADMIN
0x1000a     Condition                            0
$ show attributes -e mh=0xcd00011
Id          Name                        Iid      Value
0xd0000     Modeltype_Name                       User
0x10000     Modeltype_Handle                     0x10004
0x10004     Contact_Status                       Established
0x10009     Security_String                      ADMIN
0x1000a     Condition                            Normal
$ show attributes -e attr=0x1000-0x11fff attrname=status mh=0xcd00011
Id          Name                        Iid      Value
0x10004     Contact_Status                       Established
0x110ed     Dev_Contact_Status                   2
0x111a56    ContactStatusEventSwitc              FALSE
$ show attributes attr=0x1006e mh=0x400165
Id          Name                        Iid      Value
0x1006e     Model_Name                           142.77
$ show attributes attr=0x100d4 mh=0x400165
Id          Name                        Iid      Value
0x100d4     If_Out_Ucast_Pkts            1       1169585
0x100d4     If_Out_Ucast_Pkts            2       1227557
```

```
0x100d4    If_Out_Ucast_Pkts           3      1227557
0x100d4    If_Out_Ucast_Pkts           4      8624873
$ show attributes attr=0x100d4,next mh=0x400165
Id         Name                        Iid     Value
0x100d4    If_out_Ucast_Pkts           1      1169589
$ show attributes attr=0x100d4,iid=2 mh=0x400165
Id         Name                        Iid     Value
0x100d4    If_Out_Ucast_Pkts           2      1227569
$ show attributes attr=0x100d4,iid=2,next mh=0x400165
Id         Name                        Iid     Value
0x100d4    If_out_Ucast_Pkts           3      1227573
$ show attributes mth=0x10004 lh=0xd00000
Id         Name                        Type             Flags
0xd0000    namingTree                  Group ID         S,D
0x10000    Modeltype_Name              Text String      R,S,M,K
0xd0200    upsBatteryCapacityInteger                    E,R
$ show attributes mth=0x3d0002 attrname=port
Id         Name                        Type             Flags
0x10023    Agent_Port                  Integer          R,W,M,D
0x112e3    IF_Port_Types               Octet String     R,W,S,D
0x11554    Create_IF_Port              Boolean          R,S,D
0x11d28    PortLinkDownEventCode       Counter          R,S,D
0x11d29    PortLinkUpEventCode         Counter          R,S,D
0x11d3d    support_ICMP                Boolean          R,W,D
0x11d41    Poll_Linked_Ports           Boolean          R,W,M,D
0x11e24    TelnetPortNum               Integer          R,W,G,D
$ show attributes mth=0x3d0002 attrname=port flags=rwmd
Id         Name                        Type             Flags
0x10023    Agent_Port                  Integer          R,W,M,D
0x11d41    Poll_Linked_Ports           Boolean          R,W,M,D
$ show attributes -e attrname=port mh=0xcd00023
Id         Name                        Iid      Value
0x10023    Agent_Port                           161
0x112e3    IF_Port_Types                        11.0.22.0
0x11554    Create_IF_Port                       TRUE
0x11d28    PortLinkDownEventCode                66312
0x11d29    PortLinkUpEventCode                  66313
0x11d3d    support_ICMP                         TRUE
0x11d41    Poll_Linked_Ports                    TRUE
0x11e24    TelnetPortNum                        0
```

show children

The show children command shows the children in relation to the model with model_handle. If relation is not specified, show children shows the children in all relations. If model_handle is not specified, the command shows the children for the current model.

The show children command displays information in the following format:

```
MHandle    MName    MTypeHn   MTypeName     Relation
handle        name     handle    name        relation
```

Example: show children

```
$ show children mh=0x400009
```

```
MHandle    Name    MTypeHnd   MTypeName      Relation
0x40000d   12.84   0x100d6    Bdg_CSI_CNB2   Collects
```

show devices

The show devices command shows a listing of all device models in the landscape that is specified by the landscape_handle.

The show devices command displays information in the following format:

```
MHandle   MName   MTypeHnd   MTypeName
Handle    Name    Handle     Name
```

Example: show devices

```
$ show devices lh=0x400000
MHandle     MName           MTypeHnd    MTypeName
0x1005c0    10.253.32.101   0x3d002     GnSNMPDev
0x100030    10.253.2.10     0x2c60021   RstonesSwRtr
```

show enumerations

The show enumerations command shows enumerated string value mapping for the corresponding enumerated value specified.

The show enumerations command displays information in the following format:

```
Id   String   Value
id   string   value
```

The show enumerations mth command displays information in the following format:

```
MHandle   String   Value
HandLE    string   value
```

Example: show enumerations

```
$ show enumerations attr=0x10004
ID        String        Value
0x10004   Lost          0
0x10004   Established   1
0x10004   INITIAL       2
$ show enumerations mth=0x10004
ID        String        Value
0x10004   Lost          0
0x10004   Established   1
0x10004   INITIAL       2
```

show events

The show events command shows the events for the model with model_handle or the events for all models in the landscape that is specified by landscape_handle. By default, the show events command shows the 2,000 most recent events for the model that is specified by model_handle or landscape_handle. If the -a option is specified, this command shows a maximum of 10,000 events for the model which is specified by model_handle or landscape_handle.

If the -n option is specified with an explicit no_events statement, the specified number of events is displayed for the model which is specified by model_handle or landscape_handle. If neither model_handle nor landscape_handle is specified,

this command shows events for all models in the current landscape. If the -x option is specified, the CLI displays text messages explaining the event types. The event time is displayed in hh:mm:ss format.

NOTE

To synchronize event and alarm support files with other OneClick servers, in a distributed environment, copy the folders containing the event format files and the probable cause files to the other SpectroSERVERs. $SPECROOT/custom/Events/CsEvFormat $SPECROOT/custom/Events/CsPCause Also copy the contents of these same directories to a custom directory on all of the SpectroSERVERs in your environment.

The show events command displays information in the following format:

```
Date          Time        Type     MHandlE    MName    MTypeName
mm/dd/yyyy    hh:mm:ss    type     handle     name     name
```

If show events is used with the -x option, the events displayed do not have a fixed format. The following is an example of typical output:

```
Thur 11 May, 2000 - 8:04:01 - Alarm number 10 generated for device AntLAN of type LAN_802_3.
Current condition is INITIAL(DEFAULT).
(event [00010701])
```

Example: show events

```
$ show events lh=0x400000
Date         Time       Type      MHandle    MName    MTypeName
04/25/1999   13:27:38   0x10302   0x4000f9   1.3      Host_IBM
04/25/1999   13:27:38   0x10202   0x400131   qa1sgi   Host_SGI
$ show events -n
Date         Time       Type      MHandle             MName             MTypeName
08/21/1999   11:30:02             0x100090xcd00067    els100-01.india   RMONApp
08/21/1999   11:25:33             0x100090xcd00067    els100-01.india   RMONApp
08/21/1999   11:20:17             0x100090xcd00067    els100-01.india   RMONApp
08/21/1999   11:15:52             0x100090xcd00067    els100-01.india   RMONApp
08/21/1999   11:10:27             0x100090xcd00067    els100-01.india   RMONApp
```

show inheritance

The show inheritance command shows the model type inheritance for the model type that is specified by model_type_handle in the landscape that is specified by landscape_handle. If the landscape_handle is not specified, the current landscape is used. The possible values for this field are Base or Derived.

The show inheritance command displays information in the following format:

```
MHandle   MName   Flags   Inheritance
handle    name    flags   inheritance
```

Example: show inheritance

```
$ show inheritance mth=0x1037b lh=0x400000
Handle    Name         Flags    Inheritance
0x10000   Root         V,D      Base
0x103ad   BanVinesFS   V,I,U    Derived
```

show landscapes

The show landscapes command shows all landscapes that are defined for each SpectroSERVER. The landscape map that is displayed is the map of the initial SpectroSERVER.

The show landscapes displays information in the following format:

```
SSName    Precedence    Port    Service    LHandle
ssname    precedence    port    service    handle
```

Example: show landscapes

```
$ show landscapes
SSName    Precedence    Port    Service   LHandle
devsgi    10            0xbeef  0x10101   0x28000000
devibm    10            0xbeef  0x10101   0x11f00000
```

show models

The show models command shows all models that are defined in the landscape, which is specified by landscape_handle. If landscape_handle is not specified, this command shows all models that are defined in the current landscape. A range of model handles can be specified by the following command:

```
mhr=low_model_handle-high_model_handle
```

Specific models can be searched for by specifying mname=model_name.

User models are identified by the show models command as either (Active) or (Not Active). If the user model status is (Not Active), the user cannot yet connect to the server. Once the user model status is (Active), the user can connect to the server.

The show models command displays information in the following format:

```
MHandle    MName    MTypeHnd    MTypeName
handle     name     handle      name
```

Example: show models

```
$ show models lh=0x400000
MHandle    MName    MTypeHnd    MTypeName
0x400004   World    0x10040     World
0x4000d9            0x10020     AUI
$ show models mname=
MHandle     MName        MTypeHnd        MTypeName
0xcd00016   0x1120002    AppDataServer
0xcd00022   0x1006b      SnmpPif
0xcd00030   0x1028f      IcmpPif
$ show models mhr=0xcd00000-0xcd000ff mth=0x230018 mname=india lh=0xcd00000
MHandle     MName                        MTypeHnd    MTypeName
0xcd000a3   hplaser.zeitnet.India.com    0x230018    TCP2_App
0xcd0002b   desire.zeitnet.India.com     0x230018    TCP2_App
```

show parents

The show parents command shows the parents in relation to the model with model_handle. If relation is not specified, it shows the parents in all relations. If model_handle is not specified, show parents shows the parents for the current model.

The show parents command displays information in the following format:

```
MHandle    MName    MTypeHnd    MTypeName    Relation
handle     name     handle      name         relation
```

Example: show parents

```
$ show parents mh=0x40000d
MHandle    MName              MTypeHnd    MTypeName    Relation
0x400009   auto-lan-30x1003c              LAN_802_3    Collects
```

show relations

The show relations command shows all relations that are currently defined in the landscape specified by landscape_handle. If landscape_handle is not specified, this command shows all relations that are defined in the current landscape.

The show relations command displays information in the following format:

```
Name             Type
relation_name    relation_type
```

Example: show relations

```
$ show relations
Name Type
Passes_Through MANY_TO_MANY
Lost_and_Found ONE_TO_MANY
Owns ONE_TO_MANY
Contains ONE_TO_MANY
```

show rules

The show rules command shows the rules for a relation. The relation is specified in the landscape that is specified by landscape_handle. If landscape_handle is not specified, the current landscape is used.

The show rules command displays information in the following format:

```
LMTHandle    LMTName    RMTHandle    RMTName
handle       name       handle       name
```

Example: show rules

```
$ show rules rel=Owns lh=0x400000
LMTHandle    LMTName    RMTHandle    RMTName
0x102da      Org_Owns   0x10043      Site
0x102da      Org_Owns   0x210023     Rtr_CiscoMGSshow
```

show types

The show types command shows all model types that are currently defined in the landscape that is specified by landscape_handle. If landscape_handle is not specified, this command shows all model types defined in the current landscape. The Flags field lists the abbreviations for each of the six attribute flags that are currently set. If a flag is not set, its abbreviation is not included in the list.

The following list includes the model type flags and their abbreviations:

- Visible = V
- Instantiable = I
- Derivable = D
- No Destroy = N
- Unique = U
- Required = R

The show types command [mth=low_mth-high_mth] shows all model types within the range between low_mth and high_mth.

NOTE
For more information about model type flags, see the Model Type Editor section.

The show types command displays information in the following format:

```
Handle    Name    Flags
handle    name    flags
```

Example: show types

```
$ show types lh=0x400000
Handle    Name          Flags
0x10000   Root          V,D
0x10080   Gen_Rptr_Prt  V,D
$ show types mthr=0x10002-0x10008
Handle    Name             Flags
0x10002   Network_Entity
0x10003   VNM              V,I,D,N,U,R
0x10004   User             V,I,D
0x10005   VIB
0x10007   DataRelay        V,D
$ show types mthr=0x210020-0x21002f mtname=Rtr_Cisco lh=0xcd00000
Handle      Name              Flags
0x210020    Rtr_CiscoAGS      V,I,D
0x210021    Rtr_CiscoCGS      V,I,D
0x210022    Rtr_CiscoIGS      V,I,D
0x210023    Rtr_CiscoMGS      V,I,D
0x210024    Rtr_CiscoMIM      V,I,D
0x21002b    Rtr_Cisco2500     V,I,D
0x21002c    Rtr_CiscoMIM3T    V,I,D
0x21002d    Rtr_Cisco3000     V,I,D
0x21002e    Rtr_Cisco4000     V,I,D
0x21002f    Rtr_Cisco7000     V,I,D
$ show types flags=VIDNUR lh=0xcd00000
Handle        Name              Flags
0x25e0000     MgmtInventory     V,I,D,N,U,R
0x10040       World             V,I,D,N,U,R
0x102cf       Top_Org           V,I,D,N,U,R
0x10003       VNM               V,I,D,N,U,R
0x102be       LostFound         V,I,D,N,U,R
0x25e0001     TopologyWrkSpc    V,I,D,N,U,R
0x10091       Universe          V,I,D,N,U,R
```

show watch

The show watch command lists applicable watch data for a model that is specified by model_handle.

The show commands use the following defaults when landscape_handle and model_handle are not specified:

Command	Default
show alarms	current landscape

show associations	current model
show attributes	current model
show attributes mth	current landscape
show children	current model
show devices	current landscape
show enumerations	current landscape
show enumerations mth	current landscape
show events	current landscape
show inheritance	current landscape
show models	current landscape
show parents	current model
show relations	current landscape
show rules	current landscape
show types	current landscape
show watch	current model

NOTE
The ‘show alarms’ and ‘show events’ commands can work with the x option to display probable cause messages for alarms and expanded event messages.

Verify the following prerequisites:

- OneClick is installed on the local server.
- The environment variable SPECROOT is set to the path of the root directory (SG-Support).
  For example, if the SG-Support files are in/usr/spectrum/SG-Support, set SPECROOT to /usr/spectrum.

The show watch command displays information in the following format:

```
Watch_Id   Watch_Name   Watch_Type   Watch_Status
watch_id   watch_name   watch_type   watch_status
```

Example: show watch

```
$ show watch mh=0xc600015
Watch_Id     Watch_Name   Watch_Type   Watch_Status
0xffff0001   watch798     Calc         Active
```

stopShd - Terminates CLI Local Server

Use the stopShd command to terminate the CLI Local Server (VnmShd daemon). The stopShd command disconnects all DX NetOps Spectrum CLI users from the currently connected SpectroSERVER and terminates the CLI Local Server. This command prompts the user for confirmation before disconnecting users and shutting down the server. (You can also shut down the daemon by using the kill -2 command.)

The command has the following format:

```
stopShd [-n]
```

- **-n**
  Specifies 'no prompt'. Include this option with the stopShd command to disable confirmation prompts. Otherwise, the following message is always displayed:
  ```
  stopShd: n users are connected, are you sure?
  ```

The 'n' represents the number of connected users, including yourself.
Valid responses are y, yes, Y, Yes, n, no, N, No.
If the command is successful, the following message is displayed:

```
stopShd: successful
```

When stopShd terminates the CLI Local Server, the following message is displayed on the system console:

```
VnmShd: stopShd executed. Exiting...
```

Example: stopShd

```
$ stopShd
stopShd: 2 users are connected, are you sure? y
stopShd: successful
```

update - Updates Model and Model Attributes

Use the update command to update model and model type attributes.

The command has the following format:

```
update  [mh=modelhandle]
attr=attribute_id[,iid=instance_id],val=value
  [attr=attribute_id[,iid=instance_id],val=value...] |
  [mh=modelhandle]
  attr=attribute_id,iid=instance_id,remove
  [attr=attribute_id,iid=instance_id,remove...] |
  [-n] mth=model_type_handle |
attr=attribute_id,val=value [attr=attribute_id,val=value ... ]
  [lh=landscape_handle] |
alarm [-r] aid=alarm_id <assign=troubleshooter |
  status=status_text | ticket=troubleticketID |
  ack=(true|false)> [lh=landscape_handle] |
action=action_code [watch=watch_id] [mh=modelhandle]
```

- **-n**
  If the -n (no prompt) option is specified with the update command, then the system does not prompt for confirmation. This option is useful in CLI scripts.
- **-r**
  The -r (replace status text/replace trouble ticket ID) option can be specified with the update alarm command when using the status or the ticket arguments. When the -r option is used, the existing alarm status text or alarm trouble ticket ID is replaced with the text specified by the status argument or the ticket argument. When the -r option is not used, the new values are appended to the existing values.
- action_code
  - reconfig, 0x1000e, or 65550 to reconfigure a model
  - activate, 0x00480003, or 4718595 to activate a watch
  - deactivate, 0x00480004, or 4718596 to deactivate a watch
  - reconfigure_apps, 0x210008, or 2162696 to reconfigure application models on Cisco and Wellfleet devices
  - reload_event_disp, 0x000100a2, or 65698 to update the SpectroSERVER with changes to EventDisp and AlertMap files

    NOTE
    The watch = <*watch_id*> parameter is applicable only for the following actions: activate (or the hexadecimal equivalent 0x00480003) and deactivate (or the hexadecimal equivalent 0x00480004).

NOTE
The following points describe the features of update command:

- The update command updates the attribute specified by attribute_id value either for model with model_handle or for all models with the model type model_type_handle in the landscape specified by landscape_handle.
- Multiple attributes can be updated with one update command by specifying multiple attribute_id, value pairs, each pair that is separated from adjacent pairs by a space.
- The remove option removes instances that are specified from a list attribute.
- If landscape_handle is not specified when updating model type attributes, the current landscape is used. If model_handle is not specified, then the specified attribute of the current model is updated.
- When you are updating model type attributes, remember that only shared attributes can be updated. Shared attributes are attributes that have the shared flag set. Use the show attributes command to see if an attribute is shared.
- Security-sensitive attributes, such as User_Community_String and Model_Security_String, can be updated through CLI. However, the current user model cannot update its own User_Security_String or Security_String, but it can update those of other models.
- The update command also lets the user specify an instance ID when changing a single attribute value. When updating a list of attribute values, an instance ID can be specified for each attribute on the list. instance_id is the instance ID for the corresponding attribute. The instance_id must be a positive integer, or sequence of dot-separated positive integers.
- If an instance ID is not specified, the update command uses the first valid instance that it finds for the attribute. If no valid instances are found, an error message is displayed:
  ```
  update: no valid instance for list attribute <attr_id>
  ```
- The update alarm command updates the alarm specified by alarm_id with the value specified by the Troubleshooter (Troubleshooter model handle or Troubleshooter name), status_text, troubleticketID, or ack parameter. To clear the existing alarm values for Troubleshooter, Status text, or Trouble Ticket ID, you can set the appropriate parameter to have no value (status=, ticket=, or assign=). The landscape_handle parameter specifies the landscape in which the alarm will be found.
- The update action command performs an action specified by action_code on a device specified by model_handle. With action_code reconfig, any device of model type GnSNMPDev, or of any model type that inherits from GnSNMPDev, can be reconfigured. The activate or deactivate action_code update a watch status on a device of a specified model_handle. When the activate action object is sent, there may be a short delay between the time the watch status changes from INITIAL to ACTIVE, depending upon the intelligence that is built into the selected model. The watch_id of the watch slated to have its status updated can be obtained by using the show watch command. The reconfigure_apps action_code update application model types for Cisco and Wellfleet device models. The reload_event_disp action_code update the SpectroSERVER with changes made to EventDisp or AlertMap files.
- Use caution when using the update action command. As with any CLI command, you can corrupt the SpectroSERVER database if you use this command incorrectly. For example, inadvertently reconfiguring a critical router can cause unpredictable results on your network.
- If update is entered with a valid model_handle or valid model_type_handle, valid attribute_id(s), and valid values, the modified attributes and their values are displayed in the following format:
  ```
  Id Name Value
  Id Name Value
  ```
- If you do not use the -n option when updating models of a specified model type, the following confirmation message is displayed:
  ```
  update: all models of this type will be updated, are you sure?
  Valid responses are y, yes, Y, Yes, n, no, N, No.
  ```
- If the update alarm command is successful, the following message is displayed:
  ```
  update:successful
  ```
- If the update action command is successful, the following message is displayed:
  ```
  update action: successful
  ```

Examples: Update

- In the following example, the update command with an instance_id is used to disable port 7 on board 5 of the Hub represented by model handle 0x4001f6:

```
$ update mh=0x4001f6 attr=0x10ee0,iid=5.7,val=1
Id        Name              Iid       Value
0x10ee0   CsPortAdminState            1
```

- In the following example, the update command is used with the remove option to remove an IP address (iid) from the deviceIPAddressList (attr) for a particular model (mh).

```
$ update mh=0xc600018 attr=0x12a53,iid=10.253.8.65,remove
update: successful
```

- In the following example, the update command is used to update the attribute named AutoPlaceStartX on all models of the model type represented by model type handle 0x10059.

```
$ update mth=0x10059 attr=0x118f2,val=100 lh=0x400000
update: all models of this type will be updated, are you sure? y
Id    Name              Value
Id    AutoPlaceStartX   100
```

- In the following example, the update alarm command is used to update an alarm troubleshooter assignment.

```
$ update alarm aid=928 assign=0xa600722
update: successful
```

- In the following example, the update alarm command is used to update alarm status. The -r option is used to overwrite the existing status.

```
$ update alarm -r aid=928 status='Working on it'
update: successful
```

- In the following example, the update alarm command is used to update the alarm Trouble Ticket ID. The -r option is used to overwrite the existing value for Trouble Ticket ID.

```
$ update alarm -r aid=928 ticket='Ax1009'
update: successful
```

- In the following example, the update alarm command is used to clear the existing value for Trouble Ticket ID.

```
$ update alarm aid=928 ticket=
update: successful
```

- In the following example, the update alarm command is used to acknowledge the alarm.

```
$ update alarm aid=928 ack=TRUE
update: successful
```

- In the following example, the update command is used to restrict updating of the User_Community_String.

```
$ update mh=0x9a000ff attr=0x1007a,val=AA,11
update: successful
```

- In the following example, the update action command is used to reconfigure a Cisco router.

```
$ update action=reconfig mh=0xc600030
update action: successful
$ update action=activate watch=0xffff0001 mh=0xc600015
Watch_Id      MHandle      Watch_Status
0xffff0001    0xc600015    INITIAL
$ update action=0x480004 watch=0xffff0001 mh=0xc600015
Watch_Id      MHandle      Watch_Status
0xffff0001    0xc600015    INACTIVE
```

Sample Scripts

This section explains how to use the sample scripts that are included with CLI. The functionality and prerequisites for each script are explained in the section.

The sample scripts included with CLI demonstrate how CLI commands can be incorporated into UNIX shell scripts so that you can automate your CLI sessions. You can find some of these scripts, or some of the functions within them, useful for your own work.

CLI includes the following scripts, and a readme file that describes the scripts in the *<$SPECROOT>*/vnmsh/sample_scripts directory:

- active_ports
- app_if_security
- cli_script
- database_tally
- update_mtype
- octet_to_ascii.pl

Review the following prerequisites when you work with CLI scripts:

- Each script has an internal variable named CLIPATH. To use a script, set the CLIPATH variable to the pathname of the directory where CLI executables are located.
- The CLIPATH variable and the other environment variables that are pathnames can be *full* or *relative* pathnames depending on how the script is run. Use *full* pathnames for the CLIPATH and other environment variables when you run a sample script as a cron script. Otherwise, you can use *relative* pathnames for these variables.
- Except for update_mtype, you can run all the CLI scripts as cron scripts.

  NOTE
  Do not run update_mtype as a cron script because it prompts the user for input.
- When you run CLI scripts, specify the correct names for the vnm_hostname variable in the .vnmshrc file.

active_ports Script

Use the active_ports script to identify all ports for each board of an IRM2 hub and to identify the active ports on each board.

The active_ports script places a report for the hub with the name hub_name in a file with the name output_file. This report lists all the ports for each board. An asterisk (*) in the ON column of the report shows you which ports are active.

This script has the following format:

```
active_ports <hub_name> <output_file>]
```

app_if_security Script

Use the app_if_security script to update the Security_String attribute value in all the interface and application models in the DX NetOps Spectrum database. The app_if_security script does update by copying the attribute value from the parent model. The script does not update any models if the recipient model (child) already has a value for the Security_String attribute or if the parent does *not* have a Security_String attribute value. After updating a model's security string, administrators can use this script to update the security string of the model's children.

This script has the following format:

```
app_if_security
```

cli_script Script

Use cli_script to execute most of the CLI commands in batch mode when you provide a data file as input. The CLI sample data file, named datafile, contains switches that indicate the command to execute and also the necessary parameters to pass to the command. The script verifies that each command is executed successfully and also maintains a runtime log.

One advantage of this script is that you can create batch files using names instead of handles. For example, you can use a model type name, rather than the hexadecimal model type handle. While this makes the files easier to create and read, the real advantage comes when you want to perform subsequent actions on a model that you have created. Instead of assigning hexadecimal model handles to the model, you can refer to the model by name.

This script has the following format:

```
cli_script datafile
```

The cli_script uses two files, datafile and clean.awk, that are also located in the sample_scripts directory.

- **datafile**
  Contains the input for cli_script. It contains each CLI command currently implemented in cli_script. See the cli_script header information for instructions about the format and syntax of this file.
- **clean.awk**
  Contains the input used in execution. The .awk files are used for formatting what data appears to the console.

Consider the following points when working with cli_script:

- Remember to change the “dummy” Network_Address (255.255.255.255) in the sample datafile to a real address.
- If you move the cli_script to another directory, you must update the environment variable SPECROOT to the support root directory (SG-SUPPORT).
  For example, if the SG-SUPPORT files are in /usr/spectrum/SG-Support, set SPECROOT to /usr/spectrum.

database_tally Script

Use this script to determine how many models of each type are currently in the database. Administrators may find this script useful when evaluating system performance. The script displays a list of all the model types and the number of models of each model type in the database.

This script has the following format:

```
database_tally <vnm-name>
```

update_mtype Script

Use this script to update a specific attribute for all models of a model type. If the attribute is a shared attribute of the model type, the script does not update the model's attribute. One advantage of this script is that you can use the model and attribute names at the prompt rather than their hexadecimal ID handles.

Note: Set the CLIMNAMEWIDTH environment variable to 256. The high value prevents truncation of model names that can cause false matches when running the script.

This script has the following format:

```
update_mtype <model_name> \ <model_typename>[<attribute_name> | <attribute_id> <value>]
```

- **model_name | model_type_name**
  Specifies a model name, or part of a model name, for a model of the model type for which the attribute update is done. You can specify any model of the model type in the command.

The script then displays a listing of all model types that have models with names containing the model name argument that you entered. The script asks you to select a model type from the list.
If you use the model name instead of the model type name, the script updates all models whose names include the string entered on the command line or at the prompt. In this case, all models of a given model type are not updated as described above.

NOTE
We recommend using the model name or model type name at the prompt and not handle.

- **attribute_name | attribute_id**
  If you do not specify these arguments initially, the script prompts for attribute name or attribute id, when it runs. At this point, you must specify either the attribute name or part of the attribute name. The script then asks you to select from a list of attributes containing the text that you entered. You can run the entire script, therefore, without prior knowledge of the hexadecimal model type handles or attribute handles.

Error Messages

This section describes the most common error messages that the CLI throws in the response to an invalid or wrongly executed command. Each error message includes reason and action to be performed.

ack alarm <alarm_id> invalid alarm id

Reason:

The alarm ID that you entered is invalid.

Action:

Enter the ack alarm command again, using a valid alarm_id.

ack alarm <landscape_handle> invalid landscape handle

Reason:

The landscape handle that you entered for the alarm is invalid.

Action:

Enter the ack alarm command again, using a valid landscape_handle.

command failed to connect with VnmShd, please run connect first

Reason:

You attempted to run other commands before running the connect command.

Action:

Begin a CLI session with the connect command.

connect already connected to <hostname> since <date/time>

Reason:

The attempt to connect is unnecessary. You are already connected to a SpectroSERVER host.

Action:

None.

connect cannot open resource file <pathname>/.vnmshrc

Reason:

The connect command cannot find the CLI resource file .vnmshrc.

Action:

The .vnmshrc resource file must be in the same directory as the connect command itself.

connect can only connect to SpectroSERVERs in <hostname> landscape map - other user(s) already connected

Reason:

The connect command has already been used to connect to a particular SpectroSERVER. You can connect only to a SpectroSERVER that is in the landscape map of the original SpectroSERVER.

Action:

None

connect ERROR No such DX NetOps Spectrum user as <username>

Reason:

The first user of the connect command is not defined as a DX NetOps Spectrum user.

Action:

Reconnect to the SpectroSERVER as a DX NetOps Spectrum user.

connect <hostname> not responding or not permitting access

Reason:

The connect command cannot connect to SpectroSERVER because the hostname is incorrect, the SpectroSERVER is not running, or the user has no user model.

Action:

Verify that the hostname is correct, that SpectroSERVER is running, and that the user has a user model.

connect <landscape_handle> invalid landscape handle

Reason:

The landscape_handle specified by the user is not valid for the specified hostname, or the handle cannot be accessed by your VNM.

Action:

Verify that the landscape_handle is valid for the specified hostname, and verify that the handle can be accessed by your VNM.

connect incompatible SpectroSERVER <version>

Reason:

The user is attempting to connect to a SpectroSERVER host whose version is incompatible with the CLI version.

Action:

Update the version of CLI that you are using.

connect invalid <value> for CLISESSID

Reason:

The connect command is used within a cron script or the windowing system returns 0 for ttyslot and the environment variable CLISESSID is set to a non-numeric value.

Action:

Use the connect command outside of a cron script and set CLISESSID to a numeric value.

connect variable <CLISESSID> not set

Reason:

You have attempted to use the connect command within a cron script without first setting the CLISESSID environment variable.

Action:

When using connect within a cron script, set the environment variable CLISESSID.

create user not permitted to create alarm

Reason:

You are not permitted to create an alarm.

Action:

Verify your user permissions.

create user not permitted to create association

Reason:

You are not permitted to create an association.

Action:

Verify your user permissions.

create user not permitted to create event

Reason:

You are not permitted to create an event.

Action:

Verify your user permissions.

create user not permitted to create model

Reason:

You are not permitted to create a model.

Action:

Verify your user permissions.

create alarm <probable_cause_id> invalid alarm cause

Reason:

The create alarm command was entered with an invalid probable_cause_id.

Action:

Re-enter the create alarm command with a valid probable_cause_id.

create alarm <alarm_severity> invalid alarm severity

Reason:

The create alarm command was entered with an invalid alarm_severity.

Action:

Re-enter the create alarm command with a valid alarm_severity.

create alarm <model_handle> invalid model handle

Reason:

The create alarm command was entered with an invalid model_handle.

Action:

Re-enter the create alarm command with a valid model_handle.

create association <left_model_handle> invalid model handle

Reason:

The create association command was entered with an invalid left_model_handle.

Action:

Re-enter the create association command with a valid left_model_handle.

create association models belong to different landscapes

Reason:

The create association command was entered with a left_model_handle and a right_model_handle in different landscapes.

Action:

Use the same landscape for both handles.

create association rel=<relation> invalid relation

Reason:

The create association command was entered with an invalid relation.

Action:

Re-enter the create association command with a valid relation.

create association <right_model_handle> invalid model handle

Reason:

The create association command was entered with an invalid right_model_handle.

Action:

Re-enter the create association command with a valid right_model_handle.

create event <event_type> invalid event type

Reason:

The create event command was entered with an invalid event_type.

Action:

Re-enter the create event command with a valid event_type.

create event <landscape_handle> unknown landscape

Reason:

The create event command was entered with an invalid landscape_handle.

Action:

Re-enter the create event command with a valid landscape_handle.

create event <model_handle> invalid model handle

Reason:

The create event command was entered with an invalid model_handle.

Action:

Re-enter the create event command with a valid model_handle.

create model <attribute_id> invalid attribute id

Reason:

No model is created because the create model command was entered with an invalid attribute_id.

Action:

Re-enter the create model command with a valid attribute_id.

create model DCM device unreachable

Reason:

No model was created because the create model command was entered with an invalid ip_address. The DCM (Device Communication Manager) issues this error message.

Action:

Re-enter the create model command with a valid ip_address.

create model Device limit exceeded

Reason:

No model was created because the Branch Manager SpectroSERVER (50 device model limit) or the Site Manager SpectroSERVER (250 device model limit) contains the maximum number of device models it can contain.

Action:

Verify that the number of device models on the SpectroSERVER have not met the prescribed limits. If they have, you may need to delete some and then re-enter the create model command.

create model <landscape_handle> invalid landscape handle

Reason:

No model was created because the create model command was entered with an invalid landscape_handle.

Action:

Re-enter the create model command with a valid landscape_handle.

create model <model_type_handle> invalid model type handle

Reason:

No model is created because the create model command was entered with an invalid model_type_handle.

Action:

Re-enter the create model command with a valid model_type_handle.

create model <value> invalid value

Reason:

No model is created because the create model command was entered with an invalid value.

Action:

Re-enter the create model command with a valid value.

create model <value> No community name

Reason:

The community name provided in the create request was incorrect.

Action:

Re-enter the create command with a valid community name.

current <model_handle> invalid model handle current model is <current_model_handle>

Reason:

An invalid model_handle was specified so the current model and the current landscape are unchanged.

Action:

Re-enter a valid model_handle if you want to modify the current model and current landscape.

current <landscape_handle> invalid landscape handle current landscape is <current_landscape_handle>

Reason:

Since an invalid landscape_handle was specified, the current model and the current landscape are unchanged.

Action:

Re-enter a valid landscape_handle if you want to modify the current model and current landscape.

current <landscape_handle> not responding or not permitting access current model is <current_model_handle>

Reason:

A landscape_handle was specified and the OneClick for the landscape was down or the user did not have a user model on that landscape.

Action:

Verify that the OneClick for the landscape in question is running; verify that you have a user model on the landscape in question; and then re-enter the landscape_handle.

current <landscape_handle> not responding or not permitting access current landscape is <current_landscape_handle>

Reason:

A model_handle was specified and the SpectroSERVER for the landscape containing that model was down or the user did not have a user model on that landscape.

Action:

Verify that the SpectroSERVER for the landscape in question is running; verify that you have a user model on the landscape in question; and then re-enter the model_handle.

destroy user not permitted to destroy alarm

Reason:

You are not permitted to destroy an alarm.

Action:

Verify your user permissions.

destroy user not permitted to destroy association

Reason:

You are not permitted to destroy an association.

Action:

Verify your user permissions.

destroy user not permitted to destroy model

Reason:

You are not permitted to destroy a model.

Action:

Verify your user permissions.

destroy alarm aid=<alarm_id> invalid alarm id

Reason:

The destroy alarm command was entered with an invalid alarm_id.

Action:

Re-enter the destroy alarm command with a valid alarm_id.

destroy alarm <landscape_handle> invalid landscape handle

Reason:

The destroy alarm command was entered with an invalid landscape_handle.

Action:

Re-enter the destroy alarm command with a valid landscape_handle.

destroy association rel=<relation> invalid relation

Reason:

The destroy association command was entered with an invalid relation.

Action:

Re-enter the destroy association command with a valid relation.

destroy association <left_model_handle> invalid model handle

Reason:

The destroy association command was entered with an invalid left_model_handle.

Action:

Re-enter the destroy association command with a valid left_model_handle.

destroy association <right_model_handle> invalid model handle

Reason:

The destroy association command was entered with an invalid right_model_handle.

Action:

Re-enter the destroy association command with a valid right_model_handle.

destroy association association does not exist between given models

Reason:

An attempt was made to destroy an association between two models that do not exist.

Action:

Verify the existence of the two models belonging to the association you are attempting to destroy.

destroy association models belong to different landscapes

Reason:

The destroy association command was entered with a left_model_handle and a right_model_handle in different landscapes.

Action:

Re-enter the destroy association command using a left_model_handle and a right_model_handle from the same landscape.

destroy model <model_handle> invalid model handle

Reason:

The destroy model command was entered with an invalid model_handle.

Action:

Re-enter destroy model command with a valid model_handle.

disconnect failed

Reason:

The disconnect command failed.

Action:

Re-try the disconnect command.

disconnect failed to connect with VnmShd, please run connect first

Reason:

An attempt was made to run disconnect when the CLI Local Server was not running.

Action:

None. You are already disconnected.

disconnect not connected

Reason:

The disconnect command failed since the user was not connected to the SpectroSERVER.

Action:

None. You are already disconnected.

jump <text_string> text string not defined

jump:<text_string>: text string not defined

```
where text_string1, text_string2... are the currently defined text strings.
```

Reason:

The jump command was entered with an undefined text_string.

Action:

Re-enter the jump command using any of the defined text strings.

<pathname>/VnmShd not found

<pathname>/VnmShd: not found

```
connect: failed
where pathname represents the path to the directory in which CLI attempted to execute VnmShd.
```

Reason:

The connect command cannot find the CLI Local Server.

Action:

Make sure VnmShd and connect are in the same directory and then re-enter the connect command.

Please connect first

Reason:

After connect executed, you ran disconnect or stopShd and then attempted to run another command.

Action:

Reissue the connect command first.

seek <attribute_id> invalid attribute id

Reason:

The seek command was entered with an invalid attribute_id.

Action:

Re-enter the seek command with a valid attribute_id.

seek <error> attribute not keyed

Reason:

The seek command was entered with the attribute_id of an attribute that was not keyed.

Action:

Re-enter the seek command with an attribute_id of a keyed attribute.

seek <value> invalid value

Reason:

The seek command was entered with an invalid value.

Action:

Re-enter the seek command with a valid value.

show attributes <attribute_id> non list attribute

Reason:

The show attributes command was entered with an instance_id for a non-list attribute_id.

Action:

Re-enter the show attributes command with an instance_id for a list attribute_id.

show attributes <attribute_id> invalid attribute id

Reason:

The show attributes command was entered with an invalid attribute_id.

Action:

Re-enter the show attributes command with a valid attribute_id.

show attributes <instance_id> invalid instance id

Reason:

The show attributes command was entered with an invalid instance_id. An instance_id is invalid if it does not consist of a sequence of non-negative integers or if it does not exist for the specified attribute.

Action:

Re-enter the show attributes command with a valid instance_id.

show attributes <model_type_handle> invalid model type handle

Reason:

The show attributes command was entered with an invalid model_type_handle.

Action:

Re-enter the show attributes command with a valid model_type_handle.

show <landscape_handle> invalid landscape handle

Reason:

A show command that uses an optional landscape_handle was entered with an invalid landscape_handle.

Action:

Re-enter the show command with a valid landscape_handle.

show <model_handle> invalid model handle

Reason:

A show command that uses an optional model_handle was entered with an invalid model_handle.

Action:

Re-enter the show command with a valid model_handle.

show no current model defined

Reason:

A show associations command that uses an optional model_handle was entered but no model_handle was specified and no current model was defined.

Action:

Re-enter the show associations command, including both a model_handle and current model.

show alarms no cause information available

Reason:

The show alarms command was used with the -x option, and the DX NetOps Spectrum alarm files containing the probable cause text messages are not available.

Action:

For the show alarms command to work with the -x option, which displays probable cause messages for alarms and expanded event messages, OneClick must be installed on the local server, and the environment variable SPECROOT

must be set to the path of the Support root directory (SG-Support). For example, if the SG-Support files are in the following directory:

/usr/spectrum/SG-Support, set SPECROOT to /usr/spectrum.

show children <relation> invalid relation

Reason:

The show children command was entered with an invalid relation.

Action:

Re-enter the show children command with a valid relation.

show events no event format information available

Reason:

The show events command was entered with the -x option, and the DX NetOps Spectrum event files containing the event format text messages are not available.

Action:

For the show events command to work with the -x option, which displays probable cause messages for alarms and expanded event messages, OneClick must be installed on the local server, and the environment variable SPECROOT must be set to the path of the Support root directory (SG-Support). For example, if the SG-Support files are in the following directory:

/usr/spectrum/SG-Support, set SPECROOT to /usr/spectrum

show parents <relation> invalid relation

Reason:

The show parents command was entered with an invalid relation.

Action:

Re-enter the show parents command with a valid relation.

show rules <relation> invalid relation

Reason:

The show rules command was entered with an invalid relation.

Action:

Re-enter the show rules command with a valid relation.

show inheritance <model_type_handle> invalid model type handle

Reason:

The show inheritance command was entered with an invalid model_type_handle.

Action:

Re-enter the show inheritance command with a valid model_type_handle.

stopShd VnmShd not running

Reason:

An attempt was made to run stopShd when the CLI Local Server was not running.

Action:

Start the CLI Local Server.

stopShd failed

Reason:

The stopShd command failed.

Action:

Try connecting again, and then execute stopShd. If this does not work, kill the VnmShd process manually.

update <attribute_id> Attribute not writable

Reason:

No update occurred because an attempt was made to update model attributes that are non-writable.

Action:

Verify that the model attributes you want to update are writable and then re-enter the update command.

update <attribute_id> invalid attribute id

Reason:

No update occurred because the update command was entered with an invalid attribute_id.

Action:

Re-enter the update command with a valid attribute_id.

update <attribute_id> non shared attribute

Reason:

The update command was used for a model type and an attribute_id of a non-shared attribute was entered.

Action:

Re-enter the update command for the model type, this time using an attribute_id for a shared attribute.

update <instance_id> invalid instance id

Reason:

No update occurred because the update command was entered with an invalid instance_id.

Action:

Re-enter the update command with a valid instance_id.

update <landscape_handle> invalid landscape handle

Reason:

No update occurred because the update command was entered with an invalid landscape_handle.

Action:

Re-enter the update command with a valid landscape_handle.

update <model_handle> invalid model handle

Reason:

No update occurred since the update command was entered with an invalid model_handle.

Action:

Re-enter the update command with a valid model_handle.

update <model_type_handle> invalid model type handle

Reason:

No update occurred because the update command was entered with an invalid model_type_handle.

Action:

Re-enter the update command with a valid model_type_handle.

update <value> invalid value

Reason:

No update occurred because the update command was entered with an invalid value or values.

Action:

Re-enter the update command with valid values.

update <action_code> invalid action code

Reason:

No update occurred because the update command was entered with an invalid action_code.

Action:

Re-enter the update with a valid action_code.

VnmShd Error Failed to connect to SpectroSERVER

Reason:

The CLI Local Server failed to connect to the SpectroSERVER.

Action:

Verify that the SpectroSERVER is running.

VnmShd Error Lost connection with SpectroSERVER

Reason:

The CLI Local Server, detecting that the SpectroSERVER to which it was connected has terminated, terminates.

Action:

Restart the SpectroSERVER and then run the CLI Local Server.

UNIX to DOS Conversion

On the UNIX platform, CLI commands are typically used with UNIX commands in a terminal window. On the Windows platform, you can use CLI commands with DOS commands in a native DOS window. This appendix lists commonly used UNIX commands and their DOS equivalents.

NOTE
The appendix is intended to be a quick reference of UNIX and DOS commands rather than an exhaustive list. See your UNIX, DOS, or Windows documentation for more information about commands and their functions.

UNIX	DOS
#	rem
cat	type
cd	cd
chdir	chdir
clear	cls
cmp, diff	comp, fc
cp	copy
cp -r	xcopy
cpio, dump, tar, ufsdump	cpio, dump, tar, ufsdump
cpio, restore, tar, ufsrestore	restore
csh, sh	command
date	date, time
echo	echo
ed	edlin
exit	exit
exportfs, share	share
fdformat, format	format
format	fdisk
format->analyze	scandisk
fsck	chkdsk
goto (csh)	goto
grep	find
if	if
ln -s	subst
lp, lpr	print
ls	dir
ls -l	attrib
man	help
mkdir	md, mkdir
more	more
mv	move, ren, rename

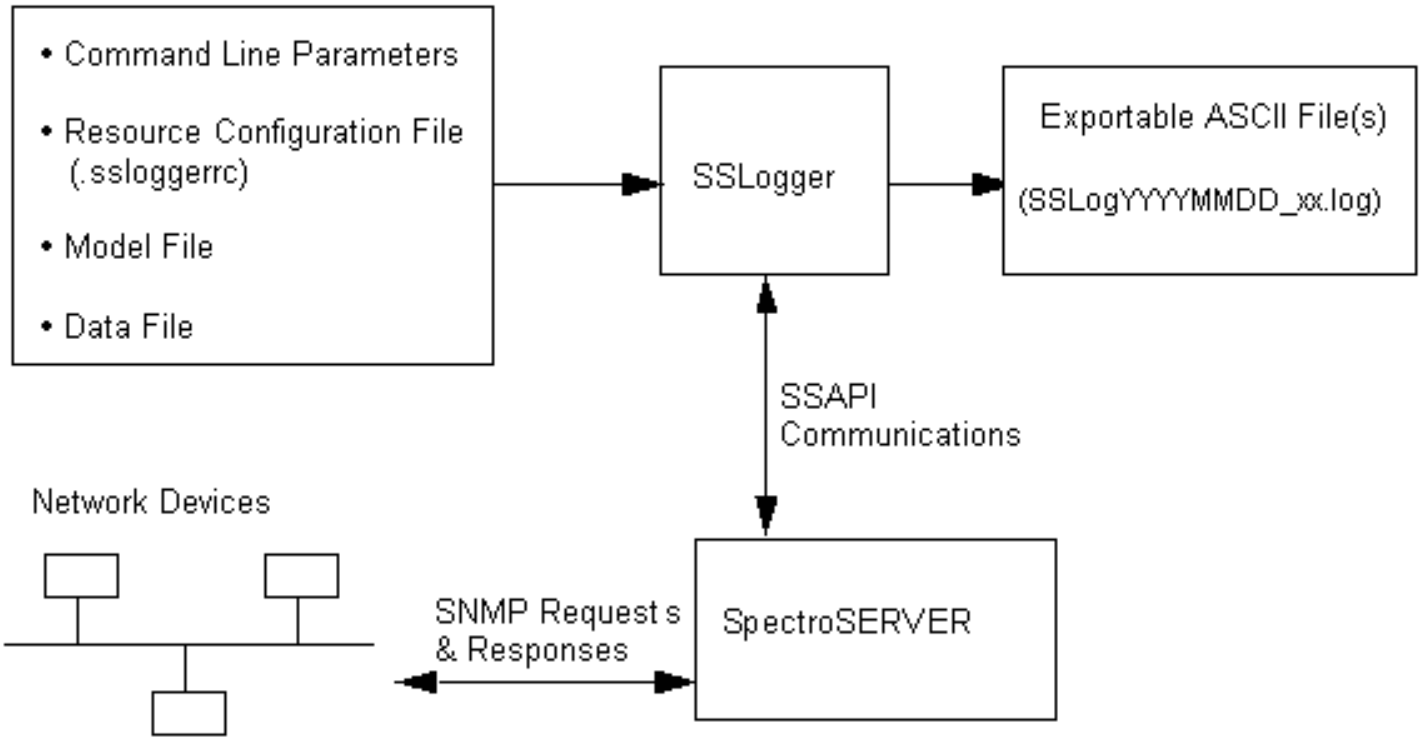
print (sh)	echo
rm	del, erase
rm -r	deltree
rmdir	rd, rmdir
set path= (csh), PATH= (sh)	path
set prompt= (csh), PS1= (sh)	prompt
set var= (csh), var= (sh)	set
shift	shift
showrev	ver
sort	sort
stty	mode
textedit, vi	edit
uncompress, unpack	expand

SSLogger

This section describes the SSLogger application and its benefits when compared with native DX NetOps Spectrum logging.

SSLogger is a DX NetOps Spectrum command-line application that works with SpectroSERVER to poll network devices and log the data to a simple ASCII file suitable for import into databases and reporting systems. As shown in the following diagram, you control SSLogger activity using command-line options and three input files (a configuration file, a model file, and a data file) which specify the target SpectroSERVER, the target devices and ports, the target attributes, the logging frequency per attribute, and the output file rotation frequency. SSLogger uses the SSAPI to communicate with SpectroSERVER, which obtains data from the target devices using SNMP requests. SSLogger then logs the specified data to an ASCII file, closing the current output file and creating a new one depending on your needs.

The following image illustrates the process that SSLogger uses to communicate with SpectroServer:



Benefits of SSLogger

SSLogger offers several advantages over DX NetOps Spectrum's native method of logging statistics:

- SSLogger lets you specify particular devices for which you want to log statistics. The DX NetOps Spectrum native method only allows you to specify the *types* of devices for which you want to log data, which means that you often are forced to poll and log more data than you want.
- SSLogger lets you set the polling and logging frequency for each attribute. The DX NetOps Spectrum native method only allows you to specify the frequency per model type.
- SSLogger writes data to a simple, readily exportable ASCII file. The v native method writes data to the Archive Manager's database, which requires additional steps to export the data.

Disabling DX NetOps Spectrum Native Logging

If you decide to log DX NetOps Spectrum statistics using SSLogger, you can turn off DX NetOps Spectrum native logging by setting the stat_logging_disabled flag to TRUE in DX NetOps Spectrum's /SS/.vnmrc file and then restarting SpectroSERVER. Setting this value to TRUE also turns off the polling of these logged attributes. It prevents DX NetOps Spectrum from using SNMP get actions to check attribute values and prevents DX NetOps Spectrum from writing the values to the Archive Manager database. This results in reduced device traffic as well as a reduced load on the Archive Manager database.

SSLogger Functionality When Contact Status is Lost

SSLogger functions differently when the contact status of a device (or group of devices) it is monitoring is lost. When SSLogger is scheduled to collect statistical data from a device and that device goes down, SSLogger will not write any data from the downed device to the .log file, effectively ignoring the device. SSLogger will not retry the downed device until the next scheduled poll cycle. SSLogger will continue to log data from devices that are up on the scheduled interval

with no interruption. Once DX NetOps Spectrum regains contact with the downed device, SSLogger will log its data on the configured scheduled interval.

SSLogger Input and Output

This section describes SSLogger input and output. It includes information about its command-line parameters as well as the content and syntax of the associated input files.

The SSLogger application uses the following four sources for input:

- Command-line parameters
- .ssloggerrc configuration file
- Model file
- Data file

Each of these input sources is described individually in this section.

Command-Line Parameters

You can use one or more of the following parameter flags with the SSLogger command. If you use these flags to specify a vnm, modelfile, or datafile that is different from the ones specified in your .ssloggerrc file, the values entered at the command line take precedence.

- **-help**
  Causes SSLogger to show a help message.
- **-vnm <*machine name*>**
  Specifies the target SpectroSERVER.
- **-modelfile <*model file name*>**
  Specifies the model file that SSLogger should use.
- **-datafile <*data file name*>**
  Specifies the data file that SSLogger should use.
- **-debug**
  Causes SSLogger to output debug information during operation.
- **-ctrace**
  Causes SSLogger to output SSLogger/SpectroSERVER communication information.
- **-strace**
  Causes SSLogger to output security information.

.ssloggerrc Configuration File

The SSLogger resource configuration file is named .ssloggerrc and contains the following keywords:

- **listen_port=**
  Specifies the port on which SSLogger communicates with the SpectroSERVER.
  Default: 0xd00f
- **vnm=**
  Specifies the target SpectroSERVER.
- **vnm_port=**
  Specifies the target SpectroSERVER communication port.
  Default: 0xbeef
- **modelfile=**
  Specifies the name of the model file that SSLogger should use.
- **datafile=**

Specifies the name of the data file that SSLogger should use.

- **max_threads=**
  Specifies the maximum number of worker threads that SSLogger should allocate for the work of logging attributes.
  Default: 30
- **thread_priority**
  Specifies the thread priority of each worker thread.
  Default: 80
- **debug_interval**
  Specifies how often (in seconds) SSLogger should output debug information if the -debug command line flag is used.
  Default: 600 seconds
- **max_oreq=**
  Specifies the maximum number of polling requests that SSLogger can have outstanding at one time. A zero (0) indicates that SSLogger can have an infinite number of outstanding requests. You can use this keyword to limit SSLogger's effect on SpectroSERVER performance.
  Default: 0

Model File

The model file specifies a list of devices that SSLogger will target. It can include devices of many different model types. Information is divided into four columns, as shown below.

```
#************************************************************
# modelfile
#
# This file specifies devices to monitor.
#************************************************************

0x40001a;   172.19.57.220;    0x1c80018;  2M46_04
0x40013c;   172.19.57.221;    0x1c80018;  2M46_04
0x40032c;   172.19.57.222;    0x1c80018;  2M46_04
0x4004ad;   172.19.57.223;    0x1c80018;  2M46_04
0x400063;   10.253.2.26;      0x2c60021;  RstoneSwRtr
0x400063;   10.253.2.27;      0x2c60021;  RstoneSwRtr
0x400063;   10.253.2.28;      0x2c60021;  RstoneSwRtr
0x400063;   10.253.2.29;      0x2c60021;  RstoneSwRtr
```

The entry for each device contains the model handle, model name, model type handle, and model type name. Semicolons are required. You can log statistics from a new device simply by adding a new line to the model file without changing the data file at all.

Data File

The data file tells SSLogger which attribute values to read. The file can include the keywords listed in this section.

group

The group keyword specifies a group of attributes. The name of the group is up to you. Information that defines each attribute in the group falls into the following five columns:

- The first column specifies the attribute name. You can use any name you want in this column, but it is recommended that you use the DX NetOps Spectrum attribute name as displayed in the CLI output in SSLogger Output.
- The second column specifies how often (in seconds) SSLogger should poll and log the attribute.
- The third column specifies the model type where the attribute is found. A value of 0x0 indicates the device model type.

NOTE
Under the example's port information group, there is a value of 0xd000a, which is the model type handle for port models.

- The fourth column is the DX NetOps Spectrum attribute ID.
- The fifth column specifies whether the attribute is a list/table attribute. The value.0 means the attribute is *not* a list/table attribute. Any other number or text means the attribute *is* a list/table attribute. You may want to put the OID in this field. The example below uses .1 to indicate a list/table attribute.

Example:

```
group: device_information
 sysUpTime            ; 10 ; 0x0 ; 0x10245 ; .0
 ipInReceives         ; 10 ; 0x0 ; 0x10098 ; .0
 ipOutRequests        ; 10 ; 0x0 ; 0x100a0 ; .0
 ifInOctets           ; 10 ; 0x0 ; 0x100cd ; .1
 ifOutOctets          ; 10 ; 0x0 ; 0x100d3 ; .1
group: port_information
 ifInOctets           ; 10 ; 0xd000a ; 0x10e41 ; .0
 ifOutOctets          ; 10 ; 0xd000a ; 0x10e42 ; .0
```

SSlogger_relation

When you want SSLogger to log attributes associated with a port that is associated with a device specified in your model file, you do not actually specify a port model handle. Instead, you specify the port's model type handle in the third column of an attribute line under a group keyword in the data file. You also specify how the port model is associated with the device model. This is done by using the SSlogger_relation keyword, one relation per keyword. If you do not specify any SSlogger_relation keywords, SSLogger will, by default, follow the two shown below. Otherwise, it will only follow the ones you specify.

```
SSlogger_relation: HASPART
SSlogger_relation: PossPrimApp
```

child_mtype_handles

The child_mtype_handles data file keyword affects *only* the display of the fourth column of the SSLogger output file. By default, only a device model handle appears in this column. By specifying the port model type handle under the child_mtype_handles keyword in the example data file below, you instruct SSLogger to display both the device model handle and the port model handle in the fourth column of the output file.

```
child_model_handles:
   0xd000a
```

mtype

The mtype keyword specifies a model type name and not a model name. This is how SSLogger interprets this keyword. For every device in the model file of this model type, poll and log the groups of attributes listed.

```
mtype: 2M46_04
   device_information
   port_information
```

Rotate_log_interval

The Rotate_log_interval keyword specifies how often (in hours or a fraction of an hour) SSLogger should close the current output file and open another one. If you do not specify this keyword, SSLogger will close the current output file and open a

new one at midnight every 24 hours. A value of 1 tells SSLogger to close the current output file at the top of the next hour and open a new output file. If the Rotate_log_interval value specified is .15, SSLogger will close the current output file and open a new output file every 15 minutes. If a value of 1.xx is entered, the value will be rounded to the nearest whole number. If you specify 2 and the current time is 2:25 p.m., SSLogger will close the current output file at 4:00 p.m., open a new file, close that file at 6:00 p.m., and so on.

on_rotate_execute

The on_rotate_execute keyword specifies a path to an executable to be run when the SSLogger log file is rotated. The path must be the full path to the executable. You must have read and executed privileges to run the executable.

```
on_rotate_execute: <full path to executable>
```

SSLogger Output

This section describes the content of the log file that is generated when you run SSLogger. SSLogger writes information to an ASCII file. The name of this output file is similar to the following: SSLog20010727_01.log.

> **NOTE**
> The filename contains the date. If you set the Rotate_log_interval keyword in the data file so that more than one SSLog file will be created in a single day, SSLogger increments the last two numbers in the output file name accordingly.

The output file is organized into the following seven columns, as shown below:

```
20010724; 10:17:26; 10:17:26; 0x40001a; -; 0x10245; 291399900
20010724; 10:17:26; 10:17:26; 0x40001a; 2; 0x100cd; 2041440626
20010724; 14:40:56; 14:40:56; 0x40001a:0x400040; -; 0x10e41; 1582426060
DATE ; START TIME ; END TIME ; MODEL HANDLE ; INTERFACE ; ATTRID ; VALUE
```

Each line in the file represents one reading of the attribute value:

- The first column indicates the date in YYYYMMDD format.
- The second column specifies the time just before SSLogger reads the attribute.
- The third column specifies the time just after SSLogger reads the attribute.
- The fourth column specifies the model handle of the model for which the attribute value is read. If you enter the port model type handle (0xd000a) under the child_mtype_handles keyword in the data file, this column will display both the device model handle and the port model handle.
- The fifth column displays a dash (-) if the attribute is *not* a list/table attribute. Otherwise, it displays the index number of a list/table attribute.
- The sixth column displays the attribute ID.
- The seventh column shows the attribute value itself.

Examples to Use SSLogger

This section provides detailed examples of how to use SSLogger in three different scenarios of increasing complexity. These examples will help you understand how to use SSLogger to log different types of DX NetOps Spectrum statistics. The examples are designed to let you follow along on your own system, creating your own input files and substituting your own and model ID values for the ones used in the examples. Sample output files are also provided. Although it is likely that most users will want to use SSLogger to log statistics associated with individual ports (interfaces) on network devices, the examples start with a simpler scenario and work their way up to the logging of ports. It is strongly suggested that you work through each of these three examples in succession before setting up your own production SSLogger scenario.

Set Up SSLogger

Before using the three examples, you should complete the setup procedure described here.

To set up SSLogger

1. Launch SpectroSERVER and OneClick.
2. Model a device (or choose one that you have already modeled) and write down the name of the device for future reference.
3. Create a text file called “modelfile” in the DX NetOps Spectrum SSLogger directory.
4. Type the following into the file and then save and close the file:

```
#*********************************************************************
# modelfile
#
# This file specifies devices to monitor.
#*********************************************************************
Create another text file called "datafile" in the SSLogger directory.
Type the following into the file and then save and close the file.
#*********************************************************************
# datafile
#
# This file specifies device information to monitor.
#*********************************************************************
```

   Open /SSLogger/.ssloggerrc with a text editor and modify it to reflect the name of your SpectroSERVER (vnm name) and the names of your model file and data file name. This file provides default parameters for running SSLogger. You can override the vnm, modelfile, and datafile settings by entering alternate values at the command line as needed.

```
listen_port=
vnm=<your SpectroSERVER name>
vnm_port=
modelfile=modelfile
datafile=datafile
max_threads=
thread_priority=
debug_interval=
max_oreq=
```

5. Start the Command Line Interface (CLI) from a terminal/Command Prompt window on your machine by navigating to the vnmsh directory and entering the following command:

```
./connect
```

   You are now ready to perform the steps involved in the following three examples.

Example 1 Logging Device Statistics

To complete the first example, you must modify your model and data files, run SSLogger, and analyze the output file.

Modify the Model File

The model file specifies the device models that you want SSLogger to target, or log statistics from. In this example, you will specify only one device model. However, you can list hundreds or thousands of devices in this file if desired.

To modify the model file

1. From the DX NetOps Spectrum vnmsh directory, enter the following CLI command:

```
./show models mname=<name of your model>
```

A list of all models whose name begins with the specified model name appears, identifying each model by model handle, model name, model type handle, and model type name. DX NetOps Spectrum often uses IP addresses to name models. The following sample output shows what the list might look like if the IP address 172.19.57.220 is used as the model name.
The entry with the model handle value 0x40001a represents the device, while the other models represent applications or ports of that device.

```
0x400059     172.19.57.220_IC   0x230012    ICMP_App
0x400056     172.19.57.220_Do   0x230052    CtDownLoadApp
0x400022     172.19.57.220_FD   0xd80004    FddiMAC
0x40004b     172.19.57.220_St   0x590006    RMONApp
0x40005b     172.19.57.220_IP   0x230016    IP2_App
0x400057     172.19.57.220_RS   0x230046    RFC1317App
0x40005d     172.19.57.220_DS   0xc40006    DS1App1406
0x40001a       172.19.57.220     0x1c80018   2M46_04
0x40001f     172.19.57.220_Tr   0xd0031     CT_Tp_Appl
0x40004a     172.19.57.220_21   0xd000a     CSIIfPort
```

2. Find your device in the resulting list, then add all four of the device information values to the model file you set up. For the device in this example, the model file would appear as follows:

```
#*************************************************************
# modelfile
#
# This file specifies devices to monitor.
#*************************************************************
0x40001a    172.19.57.220     0x1c80018   2M46_04
```

3. Separate the four device information values with semicolons, as shown below.

```
#*************************************************************
# modelfile
#
# This file specifies devices to monitor.
#*************************************************************

0x40001a;   172.19.57.220;    0x1c80018;  2M46_04
```

Modify the Data File

The data file specifies the attributes for which you want to log values. In this example you will use the following three attributes:

- ipInReceives
- ipOutRequests
- sysUpTime

To build the data file

1. From the vnmsh directory, enter the following CLI command using the model type handle for your device model from the third column in your model file:

```
./show attributes mth=<your model type handle> flags=E
```

All of the specified model type's attributes for which the external (E) flag is set are listed. For each attribute, the list shows the attribute ID, attribute name, attribute type, and attribute flag or flags.
The following sample output shows what the list might look like if the 172.19.57.220 device's model type handle of 0x1c80018 is used.

```
0x10098     ipInReceives         Counter      E,R
0x10099     ipInHdrErrors        Counter      E,R
```

```
0x1009a     ipInAddrErrors       Counter      E,R
0x1009b     ipForwDatagrams      Counter      E,R
0x1009c     ipInUnknownProtos    Counter      E,R
0x1009d     ipInDiscards         Counter      E,R
0x1009f     pInDelivers          Counter      E,R
0x100a0     ipOutRequests        Counter      E,R
0x100a1     ipOutDiscards        Counter      E,R
0x100a2     ipOutNoRoutes        Counter      E,R
0x101c1     icmpOutAddrMaskReps  Integer      E,R
0x10245     sysUpTime            Time Ticks   E,R,M,P
0x10b5a     sysContact           Text String  E,R,W,M
0x10b5b     sysName              Text String  E,R,W
```

2. In the output that you generate, find the three attributes targeted in this example (ipInReceives, ipOutRequests, and sysUpTime), and then add their names and IDs to your data file along with the model type name of your model so that it looks like the following example:

```
#*************************************************************
# datafile
#
# This file specifies device information to monitor.
#*************************************************************
group: device_information
 sysUpTime            ; 10 ; 0x0 ; 0x10245 ; .0
 ipInReceives         ; 10 ; 0x0 ; 0x10098 ; .0
 ipOutRequests        ; 10 ; 0x0 ; 0x100a0 ; .0
mtype: <model type name of your model>
 device_information
```

   In this data file example, group is an SSLogger data file keyword referring to a group of attributes. You can name a group anything you want. In this case, the device_information group includes the three target attributes, and there are five columns of information about them. The first column specifies the attribute name. You can use any name, but it is suggested that you use the DX NetOps Spectrum names as they appear in the output from Step 1. The second column tells SSLogger how often (in seconds) to poll and log the attribute value associated with the attribute. A polling interval of once every ten seconds would normally be too frequent, but it is used in this example to give you a large amount of output data in a short time. It is recommended that you set the polling interval to the same value as the Rotate_log_interval. The third column specifies the model type where the attribute is found. 0x0 indicates the *device* model type. In a later example, you will set this field to a *port* model type handle for certain attributes. The fourth column specifies the DX NetOps Spectrum attribute ID. The fifth column indicates whether the attribute is a list attribute; in this case the .0 means the attribute is *not* a list attribute.

Run SSLogger

To run SSLogger, enter the following command from the SSLOGGER directory:

```
./SSlogger
```

A message similar to the following appears:

```
sslogger version 3.0rev0 -- built on Jul 3 2001 at 09:33:13
SSlogger started at: Fri Jul 27 10:36:55 2001
```

Tail the SSLogger Output File

While SSLogger is running, you can look at the output file by using the “tail” command and specifying the name of the file, which contains the current date in YYYYMMDD format followed by a two-digit sequence number and a .log extension. For example, the first log file created by SSLogger on September 12, 2001 would be named SSLog20010912_01.log.

NOTE
See Rotate_log_interval for more information on how log files are sequenced and closed.

Assuming this is the currently active log file, you could view statistics being added to the end of the file by entering the following command from the SSLOGGER directory:

```
tail -f SSLog20010912_01.log
```

Replace the filename in the tail command above with the name of your own SSLogger output file, execute the command, and you will see a display similar to the following example.

```
1 20010718; 10:51:43; 10:51:43; 0x40001a; -; 0x10245; 239765127
2 20010718; 10:51:43; 10:51:43; 0x40001a; -; 0x10098; 1152773
3 20010718; 10:51:43; 10:51:43; 0x40001a; -; 0x100a0; 1142885
4 20010718; 10:51:53; 10:51:53; 0x40001a; -; 0x10245; 239766128
5 20010718; 10:51:53; 10:51:53; 0x40001a; -; 0x10098; 1152774
6 20010718; 10:51:53; 10:51:53; 0x40001a; -; 0x100a0; 1142886
7 ...
```

The output file contains seven columns of information:

- The first column shows the date.
- The second column shows the time just before SSLogger reads the attribute value.
- The third column shows the time just after SSLogger reads the attribute value.
- The fourth column shows the device model handle.
- The fifth column shows “-” if the attribute is *not* a list attribute. Otherwise, as seen in a later example, it shows the table index.
- The sixth column shows the attribute ID.
- The seventh column shows the attribute value.

NOTE
For more information about the output file contents, see SSLogger Output.

Example 2 Logging List Attribute Statistics

In Example 1, you logged single-value device attributes. Example 2 shows you how to log *list* attributes or tables of attribute data. As with the first example, this means modifying your data file, running SSLogger, and examining the resulting output file data.

Log List Attribute Statistics

In this example you will use the following two attributes:

- ifInOctets
- ifOutOctets

To log list attribute statistics

1. From the DX NetOps Spectrum vnmsh directory, enter the following CLI command using the model type handle for your device model from the third column in your model file:

   ```
   ./show attributes mth=<your model type handle> flags=ET
   ```

   Because you use the “E” and “T” flags, this command will list all of the specified model type's attributes for which the external (E) and table (T) flags are set. For each attribute, the list will show the attribute ID, attribute name, attribute type, and attribute flag or flags. The following sample output shows what the list might look like if the 172.19.57.220 device's model type handle of 0x1c80018 is used.

   ```
   0x100cd     ifInOctets            Counter           E,R,[]
   0x100ce     ifInUcastPkts         Counter           E,R,[]
   ```

```
0x100cf     ifInNUcastPkts         Counter           E,R,[]
0x100d0     ifInDiscards           Counter           E,R,[]
0x100d1     ifInErrors             Counter           E,R,[]
0x100d2     ifInUnknownProtos      Counter           E,R,[]
0x100d3     ifOutOctets            Counter           E,R,[]
0x100d4     ifOutUcastPkts         Counter           E,R,[]
```

2. In the output that you generate, find the two attributes targeted for this example (ifInOctets and ifOutOctets) and add their names and IDs to your data file so that it looks like the following example:

```
#*************************************************************
# datafile
#
# This file specifies device information to monitor.
#*************************************************************
group: device_information
 sysUpTime            ; 10 ; 0x0 ; 0x10245 ; .0
 ipInReceives         ; 10 ; 0x0 ; 0x10098 ; .0
 ipOutRequests        ; 10 ; 0x0 ; 0x100a0 ; .0
 ifInOctets           ; 10 ; 0x0 ; 0x100cd ; .1
 ifOutOctets          ; 10 ; 0x0 ; 0x100d3 ; .1
mtype: <model type name of your model>
 device_information
```

NOTE
The new lines have ".1" at the end which indicates that these attributes represent tables of information. Any value except ".0" indicates that the attribute is a table attribute. So you may want to put a complete OID in place of ".1" for documentation purposes.

3. Run SSLogger.
4. Tail the SSLogger output file. Again, you can look at the output file by using the "tail" command and specifying the name of the file, which should be the same as the name you used in the first example (see SSLogger Output for information on how and when new output files are opened). To do this, enter the following command from the SSLOGGER directory:

```
tail -f SSLog<current date in YYYYMMDD format>_01.log
```

This time your output should appear as follows:

```
20010724; 10:17:26; 10:17:26; 0x40001a; 1; 0x100cd; 0
20010724; 10:17:26; 10:17:26; 0x40001a; 1; 0x100d3; 0
20010724; 10:17:26; 10:17:26; 0x40001a; -; 0x10245; 291399900
20010724; 10:17:26; 10:17:26; 0x40001a; -; 0x10098; 1307231
20010724; 10:17:26; 10:17:26; 0x40001a; -; 0x100a0; 1297366
20010724; 10:17:26; 10:17:26; 0x40001a; 2; 0x100cd; 2041440626
20010724; 10:17:26; 10:17:26; 0x40001a; 2; 0x100d3; 2308773051
20010724; 10:17:26; 10:17:26; 0x40001a; 3; 0x100cd; 2794679535
20010724; 10:17:26; 10:17:26; 0x40001a; 3; 0x100d3; 2497597136
20010724; 10:17:26; 10:17:26; 0x40001a; 4; 0x100cd; 0
20010724; 10:17:26; 10:17:26; 0x40001a; 4; 0x100d3; 0
20010724; 10:17:26; 10:17:26; 0x40001a; 5; 0x100cd; 0
20010724; 10:17:26; 10:17:26; 0x40001a; 5; 0x100d3; 0
20010724; 10:17:26; 10:17:26; 0x40001a; 6; 0x100cd; 0
20010724; 10:17:26; 10:17:26; 0x40001a; 6; 0x100d3; 0
20010724; 10:17:26; 10:17:26; 0x40001a; 7; 0x100cd; 183630755
20010724; 10:17:26; 10:17:26; 0x40001a; 7; 0x100d3; 176510228
20010724; 10:17:26; 10:17:26; 0x40001a; 14; 0x100cd; 0
20010724; 10:17:26; 10:17:26; 0x40001a; 14; 0x100d3; 0
```

```
20010724; 10:17:26; 10:17:26; 0x40001a; 15; 0x100cd; 0
20010724; 10:17:26; 10:17:26; 0x40001a; 15; 0x100d3; 0
20010724; 10:17:26; 10:17:26; 0x40001a; 16; 0x100cd; 0
20010724; 10:17:26; 10:17:26; 0x40001a; 16; 0x100d3; 0
20010724; 10:17:26; 10:17:26; 0x40001a; 17; 0x100cd; 0
20010724; 10:17:26; 10:17:26; 0x40001a; 17; 0x100d3; 0
20010724; 10:17:26; 10:17:26; 0x40001a; 18; 0x100cd; 0
20010724; 10:17:26; 10:17:26; 0x40001a; 18; 0x100d3; 0
20010724; 10:17:26; 10:17:26; 0x40001a; 19; 0x100cd; 0
20010724; 10:17:26; 10:17:26; 0x40001a; 19; 0x100d3; 0
20010724; 10:17:26; 10:17:26; 0x40001a; 20; 0x100cd; 0
20010724; 10:17:26; 10:17:26; 0x40001a; 20; 0x100d3; 0
20010724; 10:17:26; 10:17:26; 0x40001a; 21; 0x100cd; 0
20010724; 10:17:26; 10:17:26; 0x40001a; 21; 0x100d3; 0
```

NOTE
The fifth column shows the table index on lines that represent table attribute data.

Example 3 Logging Port Statistics

This example shows how to log statistical information associated with a port. Once again, you will modify your data file, run SSLogger, and look at the output file. Note that you do *not* have to modify the model file, nor do you have to specify the model handles of the port models at all. Instead of stipulating port model handles, you will specify port model *type* handles and DX NetOps Spectrum relations. In this example you will also use the ifInOctets and ifOutOctets attributes.

Log Port Statistics

To log port statistics

1. From the vnmsh directory, enter the following CLI command using the model handle for your device model from the first column in your model file:

   ```
   ./show children rel=HASPART mh=<your model handle>
   ```

   This CLI command shows you the port models associated with your target device. You need to execute this command to determine the model type handles of the ports you want to target. In this case, all ports are the same model type. The sample output below shows what the list might look like using the 0x40001a model handle from the first example.

   ```
   MHandle     MName             MTypeHnd    MTypeName        Relation
   0x40003f    172.19.57.220_1   0xd000a     CSIIfPort        HASPART
   0x400040    172.19.57.220_2   0xd000a     CSIIfPort        HASPART
   0x400041    172.19.57.220_3   0xd000a     CSIIfPort        HASPART
   0x400042    172.19.57.220_5   0xd000a     CSIIfPort        HASPART
   ```

2. If your output shows several different kinds of ports, select one port model type and use its model type handle in the following command:

   ```
   ./show attributes mth=<your port model type handle> flags=E
   ```

   This returns a list of external attributes associated with the specified port model type as shown here:

   ```
   0x10e3f     ifAdminStatus             Integer       E,R,W,[]
   0x10e40     ifOperStatus              Integer       E,R
   0x10e41     ifInOctets                Counter       E,R
   0x10e42     ifOutOctets               Counter       E,R
   0x11315     ifInUcastPkts             Counter       E,R
   0x11317     ifInNUcastPkts            Counter       E,R
   0x11318     ifInDiscards              Counter       E,R
   0x11319     ifInErrors                Counter       E,R
   ```

```
0x1131a     ifInUnknownProtos         Counter       E,R
```

Your command output may or may not show the attributes ifInOctets and ifOutOctets. If not, you will have to choose a few of your own to use as you follow along with this example.

3. Add a new group called port_information to your data file, and then enter the attribute name, model type handle (not 0x0, which indicates device models), and attribute ID for each of your attributes. Finally, add the information for the SSlogger_relation and child_mtype_handles so that your data file looks like the example shown below.

```
#************************************************************
# datafile
#
# This is a sample data input file for sslogger.
#************************************************************
group: device_information
 sysUpTime            ; 10 ; 0x0 ; 0x10245 ; .0
 ipInReceives         ; 10 ; 0x0 ; 0x10098 ; .0
 ipOutRequests        ; 10 ; 0x0 ; 0x100a0 ; .0
 ifInOctets           ; 10 ; 0x0 ; 0x100cd ; .1
 ifOutOctets          ; 10 ; 0x0 ; 0x100d3 ; .1
group: port_information
 ifInOctets           ; 10 ; 0xd000a ; 0x10e41 ; .0
 ifOutOctets          ; 10 ; 0xd000a ; 0x10e42 ; .0
SSlogger_relation: HASPART
child_mtype_handles:
 0xd000a
mtype: <model type name of your model>
 device_information
 port_information
```

The SSlogger_relation keyword specifies which relations SSLogger should follow between any devices specified in your model file and any models of model types specified in the attribute lines of your data file. In this example, the value for this keyword is HASPART because that is the relation a DX NetOps Spectrum device model has to its port models.

NOTE
You can see a list of all DX NetOps Spectrum relations using the CLI “show relations” command.

The child_mtype_handles keyword affects ONLY the display of the fourth column of the SSLogger output file (see example in following section). By default, only a device model handle appears in this column. By specifying the port model type handle under the child_mtype_handles keyword, you instruct SSLogger to display *both* the device model handle and the port model handle in the fourth column of the output file.

4. Run SSLogger.
5. Once again, look at the output file by using the “tail” command and specifying the name of the SSLog file, which should be the same as the name you used in the previous examples (see SSLogger Output for information on how and when new output files are opened). To do this, enter the following command from the SSLOGGER directory:

```
tail -f SSLog<current date in YYYYMMDD format>_01.log
```

Your output should appear similar to the following sample:

```
20010718; 14:40:56; 14:40:56; 0x40001a; -; 0x10245; 241140428
20010718; 14:40:56; 14:40:56; 0x40001a; -; 0x10098; 1159008
20010718; 14:40:56; 14:40:56; 0x40001a; -; 0x100a0; 1149142
20010718; 14:40:56; 14:40:56; 0x40001a:0x400040; -; 0x10e41; 1582426060
20010718; 14:40:56; 14:40:56; 0x40001a:0x400040; -; 0x10e42; 1808004560
20010718; 14:40:56; 14:40:56; 0x40001a:0x400041; -; 0x10e41; 2251080181
20010718; 14:40:56; 14:40:56; 0x40001a:0x400041; -; 0x10e42; 1999870610
20010718; 14:40:56; 14:40:56; 0x40001a:0x400042; -; 0x10e41; 0
20010718; 14:40:56; 14:40:56; 0x40001a:0x400042; -; 0x10e42; 0
```

```
20010718; 14:40:56; 14:40:56; 0x40001a:0x40003f; -; 0x10e41; 0
20010718; 14:40:56; 14:40:56; 0x40001a:0x40003f; -; 0x10e42; 0
```

NOTE
Both the device and port model handles appear in the fourth column as described previously.

CA Business Intelligence (CABI)

CA Business Intelligence (JasperReports® Server) Documentation

Report Manager

DX NetOps Spectrum Reporting provides an analysis of the inventory, availability, changes, performance, and fault history of the network assets that are managed in DX NetOps Spectrum. You can share reports throughout the enterprise. DX NetOps Spectrum Reporting compiles the required data and presents it in a specified format.

The DX NetOps Spectrum Reporting data server extracts data from the DX NetOps Spectrum knowledge base and stores it in the reporting database. You can generate reports that provide information on various aspects of network assets that are relevant to an organization. DX NetOps Spectrum Reporting addresses the information requirements of not only the Information Technology (IT) group, but also of other groups in your organization.

DX NetOps Spectrum Reporting helps you make informed decisions on IT assets, and provides the following information:

- Assets that have the most issues.
- Events and Alarms that recur frequently.
- Number of routers or other gateway devices from a specific vendor that are deployed in the network.
- Devices that are the most and least frequently offline.
- Modified and deleted assets.

Report Customization

DX NetOps Spectrum Reporting lets you customize and specify the type of content to include in reports. You can specify how the data is organized and represented using text and graphics. Customization features include the following options:

- Detailed or summary versions of the asset information. Detailed information includes subviews for report items such as devices, ports, or vendors.
- Information that is organized by asset type, landscape, vendor, or global collections.
- Historical period -- asset information from an earlier day, week, month, year, a specific date range, or business day hours.
- Report layout -- title, subtitle, header and footer text, and sort order.
- Charts and graphs.
- Do-it-yourself Ad Hoc reports.
  For more information, see Generating Ad Hoc Reports.

Report Scheduling

DX NetOps Spectrum Reporting lets you set up, save, and schedule reports to run on a one-time or periodic basis. When you schedule a report, DX NetOps Spectrum Reporting automatically generates it at the time you specify and saves the report results. You can schedule reports for other users and also configure email reports to any number of recipients.

Reports On Demand

Running reports interactively, or on demand, provides the flexibility to generate the most recent and required information. The report on demand feature lets you perform the following actions:

- Experiment with various report types and configurations. Testing can be performed to select the reports that you want to schedule.
- Investigate acute problems or trends that occur in an IT infrastructure by generating reports over various intervals and settings.

Ad Hoc Reports

The DX NetOps Spectrum Ad Hoc Reporting feature enables you to define your own reports. DX NetOps Spectrum Ad Hoc Reporting lets you drag-and-drop data objects to construct reports that are specific to your environment. The data objects that are available in the Ad Hoc environment represent a significant subset of DX NetOps Spectrum attributes that support a wide range of custom reporting requirements.

DX NetOps Spectrum Ad Hoc Reporting lets you define all aspects of a report. You can select the data objects, parameters, and layout, using WEBI features.

For more information, see Generating Ad Hoc Reports.

Report Publishing

DX NetOps Spectrum Reporting lets you print and save reports in the following formats that accommodate the publishing, presentation, and recordkeeping requirements of your organization:

- Microsoft® Excel®
- Microsoft® Word®
- PDF
- RTF
- XML
- Crystal Reports® (RPT)
- Comma-separated values (CSV)

Report Types

To accommodate the diverse information requirements in your organization, you can generate various reports. DX NetOps Spectrum reports are grouped into report packs. Each report pack includes predefined reports that provide a specific type of information about your network assets.

Standard report packs include the following types of reports:

- **Alarm**
  Alarm reports generate historical information about alarm events for assets in the IT infrastructure. Alarm reports can assess the network health, identify alarm trends, find recurring or cyclical problems, and locate assets with past alarms.
- **Asset**
  Asset reports generate information about the inventory of assets in the IT infrastructure, including information about asset port availability and asset firmware versions. Asset reports can be used to determine how vendor products are distributed throughout the infrastructure. You can assess whether they are being used effectively and can identify opportunities for improvement.
- **Availability**

Availability reports provide historical information about uptime and downtime for assets in the IT infrastructure. A Projected-Availability report lets you determine the downtime that assets can sustain before they violate a threshold or SLA.
Note: Consult a DX NetOps Spectrum Reporting administrator for information about exemptions for planned outages or outages that occur on holidays.

- **Event**
  Event reports provide information about DX NetOps Spectrum events that are generated for DX NetOps Spectrum models. You can generate event reports for all models or for selected models. You can also generate reports that contain ranked lists of the most frequently occurring events during specific time periods. Event filtering options let you supply event codes to include or exclude from all event reports.

Upgrade reports include the following reports:

- **Legacy Reports**
  Legacy reports are scheduled reports that are migrated from previous DX NetOps Spectrum Reporting versions during an upgrade.

DX NetOps Spectrum Reporting also supports the following optional report packs:

- **Network Configuration Management (NCM)**
  Network Configuration Management (NCM) reports provide information about network configuration activities that NCM recorded. For more information, see the Network Configuration Manager section.

  NOTE

  NCM must be installed with OneClick to enable the NCM reports. There are no out of the box reports for NCM, a Jasper customer, can use the NCM domain to create their reports (ad hoc reports).

- **Response Time**
  Response Time reports provide information about response time test results and analysis that DX NetOps Spectrum Service Performance Manager compiled. The Response Time report pack includes trend and exception reports which help you troubleshoot response time issues before they become problems for end-users. Reports graphically depict past performance and trends in response times. Exception or Top N reports detail the areas where proactive action can be taken to avoid critical issues. For more information, see the Service Performance Manager section.

  NOTE
  DX NetOps Spectrum Service Performance Manager must be installed with OneClick to enable the Response Time reports.

- **Service and SLA**
  Service and SLA reports provide summary and historical information about service and Service Level Agreements (SLAs). The reports focus on service customer models that are created and managed with DX NetOps Spectrum Service Manager. Service and SLA reports enable you to track service assets, gauge service health, and analyze results to determine how to improve service performance.

  NOTE
  Service Manager must be installed with OneClick to enable the Service and SLA reports.

- **Virtual Private LAN Service (VPLS) Reports** (available if the DX NetOps Spectrum VPLS Manager application is installed with OneClick)
  DX NetOps Spectrum VPLS Manager is a DX NetOps Spectrum add-on management application for service providers who are deploying VPLS technology. By integrating DX NetOps Spectrum VPLS Manager with DX NetOps Spectrum Reporting, you can view reports for monitoring the health of your VPLS environment. For more information, see the VPLS Manager Solution section.

CA Business Intelligence - JasperReports Server

Refer to the CA Business Intelligence JasperReports® Server Documentation

Using Reports

This section describes the procedure to use, generate and schedule Reports.

Use BI Launch Pad to Generate Reports

You can access BI Launch Pad from the OneClick home page to generate and manage reports.

Consult your OneClick administrator for the following information:

- Supported Web browsers.
- CABI login credentials to access BI Launch Pad.

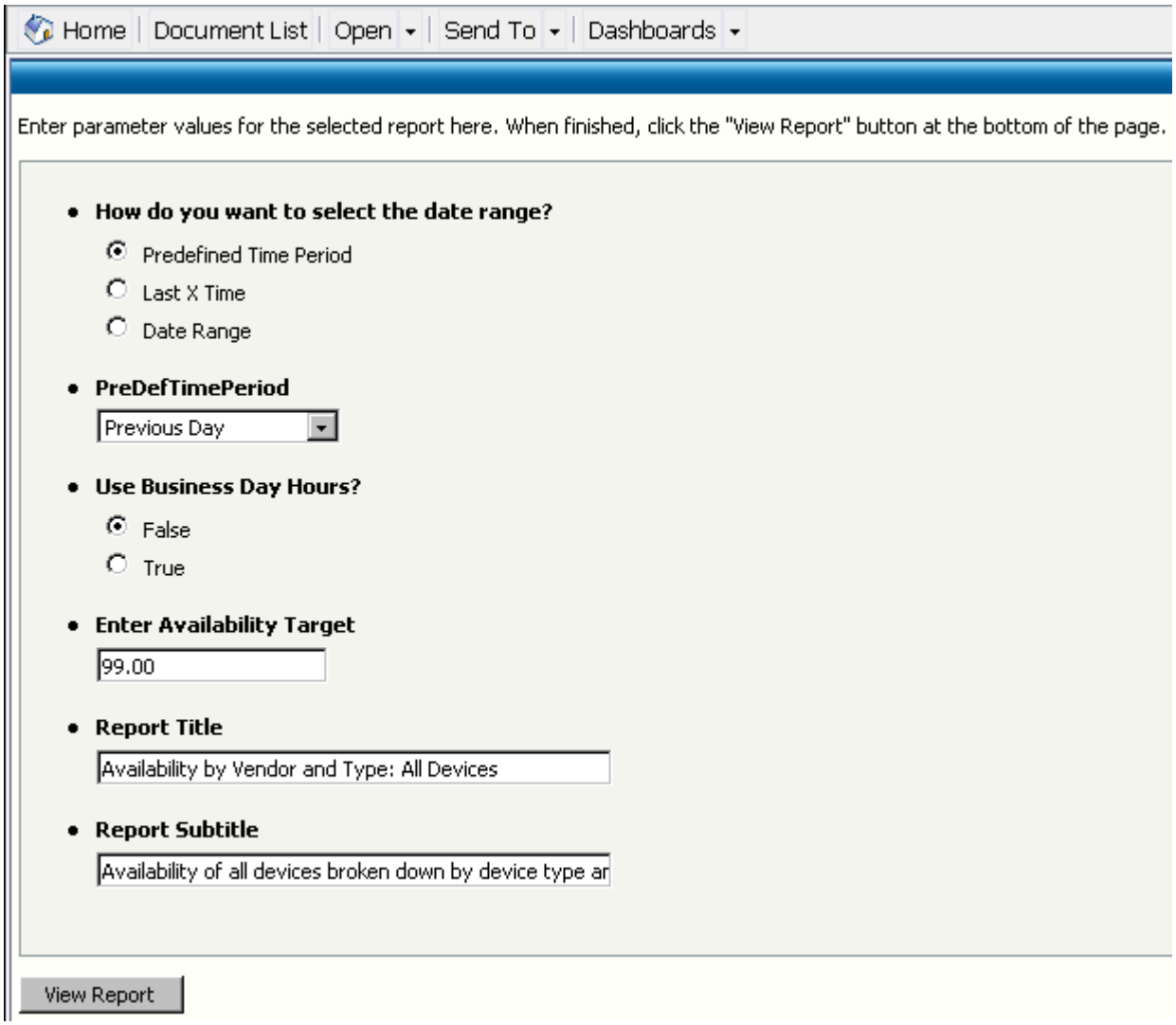
Follow these steps:

1. Log in to OneClick.
2. Click the BI Launch Pad tab in the OneClick Console.
   The BI Launch Pad window opens.

   NOTE
   You can also open BI Launch Pad directly from a Web browser to access DX NetOps Spectrum reports. The typical URL format is as follows:

   ```
   http://<hostname>/BILaunchPadApp
   ```
3. Click the Document List folder.
   The Document List folder lets you view all BI Launch Pad reports, including DX NetOps Spectrum reports.
4. Select the Public Folders, and then CA Reports.

   NOTE
   You can set the CA Reports folder as the BI Launch Pad start page through the Preferences section. For more information, see the CA Business Intelligence Implementation section.
5. Select DX NetOps Spectrum Reports.
   You can access report packs to which you have been granted rights. Consult your administrator for assistance if you cannot access a report pack.
6. Specify the parameter values for the selected report.
   The following image shows the parameter settings for an 'Availability by Vendor and Type: All Devices' report.



7. Click View Report.
   The report displays.
8. To Export/Save a report, click the Export icon and select the report type to save.
   The report is saved.

Understanding Report Parameters

When you generate a DX NetOps Spectrum report, a Parameter dialog lets you specify parameters applicable to the report.

All reports include a report title and subtitle that you can customize. Date range parameters let you generate reports that provide historical information on the availability of assets and their associated changes and problems.

Reports also include parameters for specifying graphical or tabular representations of information.

Usernames and Passwords

OneClick users are automatically added to CABI. The default CABI password is the username. Change the default password at your initial login.

If a default CABI password is set, contact the Spectrum Report Manager administrator to reset the password. For more information, see the Install Report Manager section.

The administrator can add users directly to CABI using the Central Management Console (CMC). CMC is a web-based tool that offers a single interface to perform administrative tasks, including user, content, and server management. CMC lets you publish, organize, and set security levels for your BusinessObjects Enterprise content. For more information, see the CA Business Intelligence Integration section.

Generate a Report in a Secure Environment

In DX NetOps Spectrum Spectrum Report Manager, model-based security is introduced to ensure that you can only report on models to which you have access. DX NetOps Spectrum Reporting honors the model-based security implementation that you have established in OneClick.

For DX NetOps Spectrum Reporting users that exist as OneClick users, user access resolution includes the following components:

- User security communities
- Landscape membership
- DX NetOps Spectrum Reporting View Data permission with a model landscape
- Security String to determine the model accessibility

The following picture illustrates the new access resolution process:

NOTE
For DX NetOps Spectrum Reporting users who lack corresponding OneClick accounts, unlimited reporting access to all models is initially provided.

If you have access to a model during the resolution process, you can effectively report on the model. If a model is inaccessible, any information pertaining to that model is absent from the reports (such as details associated with the model or model values in aggregations).

For more information, see the Install Report Manager section.

Displaying Graphical Elements in Reports

DX NetOps Spectrum Reporting lets you use graphical elements in many reports to represent various types of information on network assets. For example, frequency, proportion, and trends. You can use the graphical elements to generate reports that you plan to include in briefings and presentations for diverse (for example, technical and non-technical) audiences.

The following graphical elements are displayed in DX NetOps Spectrum reports.

Bar Graphs:

DX NetOps Spectrum Reports use bar graphs to display report results that indicate the volume and the relative frequency with which something occurred on an asset (such as, alarms) or a comparative ranking of various assets (such as, availability).

A bar graph from an alarm report illustrates the number of alarms that occurred for a particular group of assets over a particular time period. The graph indicates the number and proportion of alarms for each asset in the group.

Pie Charts:

DX NetOps Spectrum Reports use pie charts to display the percentage breakdown of report results in terms of proportion.

For example, a pie chart displays the following assets:

- An equivalent proportion of switch-routers and switches.
- A greater proportion of routers.
- A small proportion of non-specific pingable devices in the network segment.

Line Charts:

DX NetOps Spectrum Reports use line charts in Response Time reports to graph continuous data over time. Line charts indicate fluctuations in latency measured in milliseconds over a specific time period.

Color Indicators:

DX NetOps Spectrum Reports use the condition color indicators for alarm severity and alarm entries in Alarm reports. A color scheme specific to DX NetOps Spectrum is used to indicate threshold compliance levels in Availability reports.

Schedule a Report

You can schedule a report from the Schedule window. To run scheduled reports correctly, select options on the Parameters page.

Follow these steps:

1. Log in to OneClick.
2. Click the BI Launch Pad tab in the OneClick Console.
   The BI Launch Pad window opens.
3. Select Documents List, Folders, Public Folders, CA Reports, and DX NetOps Spectrum.
   A list of DX NetOps Spectrum reports display.
4. Select a report.
5. Right-click and select Schedule.
   The Schedule window appears.
6. Click Prompts.
   The Prompts page displays.
   The Prompts page is dynamic and provides various fields depending on the report values required. The Prompts page also displays when you run a report on demand.
7. Specify the values in the Prompts page.

8. Click the Schedule button.
   The report is scheduled.

View Scheduled Reports

After you schedule a report, you can verify the schedule by checking scheduled instances of that report.

The history report can be set to show all instances, instances that you own, and instances that are completed. In addition, you can filter instances by time.

If multiple scheduled reports are created from the same on-demand report, all instances for the scheduled reports are listed. You can organize the instance list by sorting on the Title column.

Follow these steps:

1. Navigate to the BI Launch Pad window.
2. Select Document List, Public Folders, CA Reports, and DX NetOps Spectrum.
   A list of DX NetOps Spectrum reports display.
3. Select the report.
4. Right-click and select History.
   The History window shows a history of scheduled instances for that report.

Print a Report

You can print any report from the report view, from a duplicate report view, and from a report subview. During the printing setup process, DX NetOps Spectrum Reporting saves a report to a PDF file. You can print or you can save the pdf file.

NOTE
You must have Adobe Acrobat Reader installed to print a report.

Follow these steps:

1. Navigate to the BI Launch Pad window.
2. Select Document List, Public Folders, CA Reports, and then DX NetOps Spectrum.
   A list of DX NetOps Spectrum reports display.
3. Select the report to print.
4. Click Print this Report (Do *not* use your browser print icon).
   The Print to PDF window opens.
5. Select All Pages to print the entire report, or specify the page range in the From and To fields.
6. Click Export.
   The File Download dialog opens.
7. Perform one of the following steps:
   - Print the report now:
     a. Click Open With in the File Download dialog. The report opens as a PDF file in Adobe Acrobat Reader.
     b. Use the Adobe Acrobat Reader print options to print the report. (You can also save the report to your computer from the Adobe window).
   - Save the report, and print it later:
     a. Click Save File in the File Download dialog to open the Save As window.
     b. Enter a filename (in place of the default filename, ReportViewer) and save it as an Adobe Acrobat document.

   The report is printed.

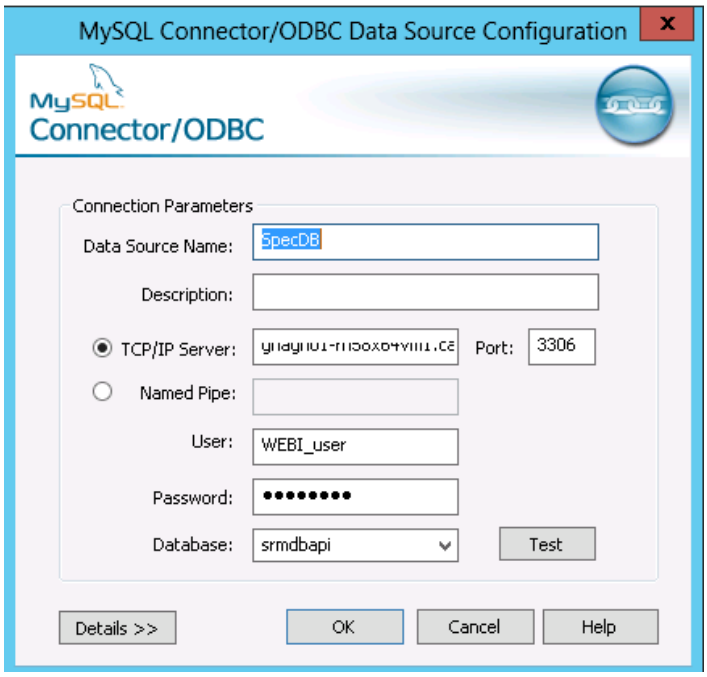
Generating Ad Hoc Reports

Starting from the 10.0 release, Ad Hoc reports use ODBC for accessing reporting database. You must create a data source name (DSN) on the CABI host, which connects to a SRM Host in OneClick.

The following section describe the procedures to configure DSN in Windows and Linux operating systems.

For Windows:

1. Go to Windows Administrative tools
2. Open ODBC Data Sources (64-bit) and add a System DSN with name “SpecDB” as shown below:



3. Similarly, create another DSN in ODBC Data Sources (32-bit) as well.
   If you do not see the MySQL ODBC driver, please download the connectors from https://dev.mysql.com/downloads/connector/odbc/5.1.html
4. Restart CABI Servers.

For Linux:

I. Installing UNIX ODBC Driver

1. Copy unixODBC-2.3.2.tar.gz package
   Download lin:http://www.unixodbc.org/unixODBC-2.3.2.tar.gz
2. Unzip and untar unixODBC-2.3.2.tar.gz
3. From unixODBC-2.3.2, run the following command to install ODBC manager:
   'yum install unixODBC unixODBC-devel'

II. Installing Mysql ODBC Driver

1. Download MySQL ODBC Driver mysql-connector-odbc-5.1.13-linux-el6-x86-64bit.tar
   Refer site link: https://dev.mysql.com/downloads/connector/odbc/5.1.html
2. Untar mysql-connector-odbc-5.1.13-linux-el6-x86-64bit.tar
3. Copy the lib and bin folder content to /usr/local/lib64 and /usr/local/bin respectively

III. Setting up ODBC configuration Execute the following steps as a CABI Install user

1. Create an environment variable with ODBCINI

ODBCINI=<CABI install directory>/sap_bobj/enterprise_xi40/odbc.ini

2. Add or update the LD_LIBRARY_PATH environment variable in
   <CABI install directory>/sap_bobj/setup/env.sh
   export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/lib64:/usr/lib64

   NOTE

   The path added above must be to the location of the libodbc.so and libodbcinst.so files which are installed with the UNIX ODBC manager.
3. Create or update the odbc.ini file in the above location to ensure it contains the DSN entry for MySQL ODBC DSN.

   NOTE

   Before updating, you must take backup of odbc.ini.

   [ODBC Data Sources]
   SpecDB=Spectrum DSN for WEBI reports
   [SpecDB]
   Description=Spectrum DB for Adhoc Reporting
   Driver=/usr/local/lib64/libmyodbc5.so
   SERVER=<*SRM OneClick Server name*> PORT=3306
   USER=WEBI_user
   PASSWORD=0n3cl1Ck
   Database=srmdbapi
   OPTION=3

4. Create Symbolic link in <CABI install directory>/sap_bobj/enterprise_xi40/dataAccess/connectionServer/drivers/lib64 folder
   ln -s /usr/lib64/libodbc.so ./libodbc.so.1
   ln -s /usr/lib64/libodbcinst.so ./libodbcinst.so.1
5. Modify the ODBC driver to 64 bit in mysql.sbo file under folder <CABI install directory>/sap_bobj/enterprise_xi40/dataAccess/connectionServer/odbc
   Change unix64 parameter under database tag as mentioned below:
   <Library Platform="Unix64">dbd_ux32odbc3</Library>
   to
   <Library Platform="Unix64">dbd_uxodbc3</Library>
6. Restart all servers from SIA/ using stopservers and startservers command.

Generate Ad Hoc Reports

This section describes the procedure to generate Ad Hoc Reports.

DX NetOps Spectrum Ad Hoc Reporting is a feature which provides flexibility to define your own reports. You can generate the following Ad Hoc reports using Ad Hoc Reporting:

- Simple Ad Hoc Reports
- Complex Ad Hoc Reports

The DX NetOps Spectrum Ad Hoc Reporting feature provides the following capabilities that enable you to define your own reports:

- Custom selection of data objects
- Custom selection of chart options
- Custom layout of report components

To enable custom reporting, DX NetOps Spectrum Reporting relies on the use of an internally designed BusinessObjects metadata layer referred to as a *universe*. A universe is a data abstraction mechanism that is provided by BusinessObjects

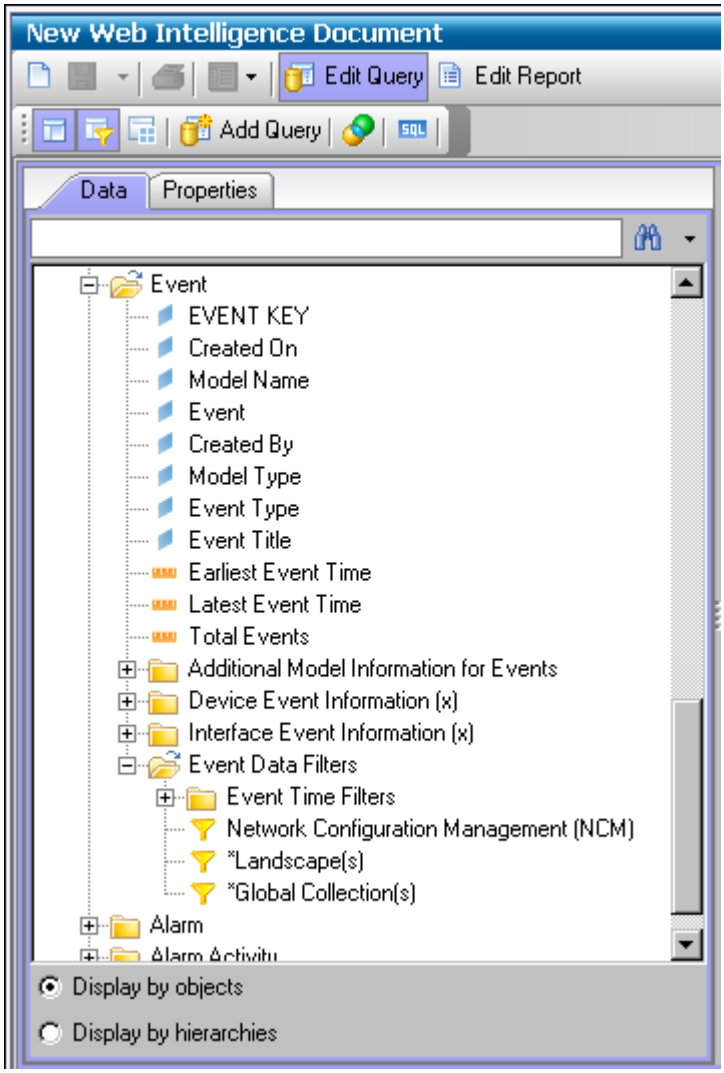
to allow data retrieval from a database without a deep understanding of the underlying data structures. A DX NetOps Spectrum-specific universe provided by CA is deployed to the BusinessObjects Enterprise software to support your Ad Hoc reporting needs.

Ad Hoc Reporting Folder Structure

Within each major folder (Asset, Event, Alarm, Alarm Activity, Availability/Outage), the primary or core objects/fields are presented. Subfolders provide supporting objects.

> **WARNING**
> Consider each major content folder as a self-contained source for report development. Do not mix the objects from one major content area (for example, Alarm) with another (Asset) when you design the report. Mixing objects from disparate content areas can lead to long-running queries against the database.

The following picture illustrates the folder structure of Ad Hoc Reporting.



Fields in capital letters designate the key or object that uniquely identifies the logical reporting entity that is associated with the major containing folder. For example, the inclusion of the EVENT KEY object in an event report query ensures that unique events are returned in the report result set.

The description of the icons is as follows:

- 
  Indicates dimension objects.
-
  Indicates measurement objects.
- 
  Indicates a condition or filter. The ready-made conditions assist you in efficient report development.
- * 
  Indicates a condition or filter. When a report is executed, a dialog appears for further information.
- (x)
  Indicates the objects contained within this folder. Do not mix with objects in a different folder that also has an (x) designation.
  For example, including objects from both the 'Device Event Information (x)' and 'Interface Event Information (x)' folders leads to no results. A model cannot be both a device and an interface simultaneously.

Generate a Simple Ad Hoc Report

You can generate a simple Ad Hoc Alarm Report using Web Intelligence. You can access Web Intelligence from BI Launch Pad.

Review the following considerations before using DX NetOps Spectrum Ad Hoc Reporting:

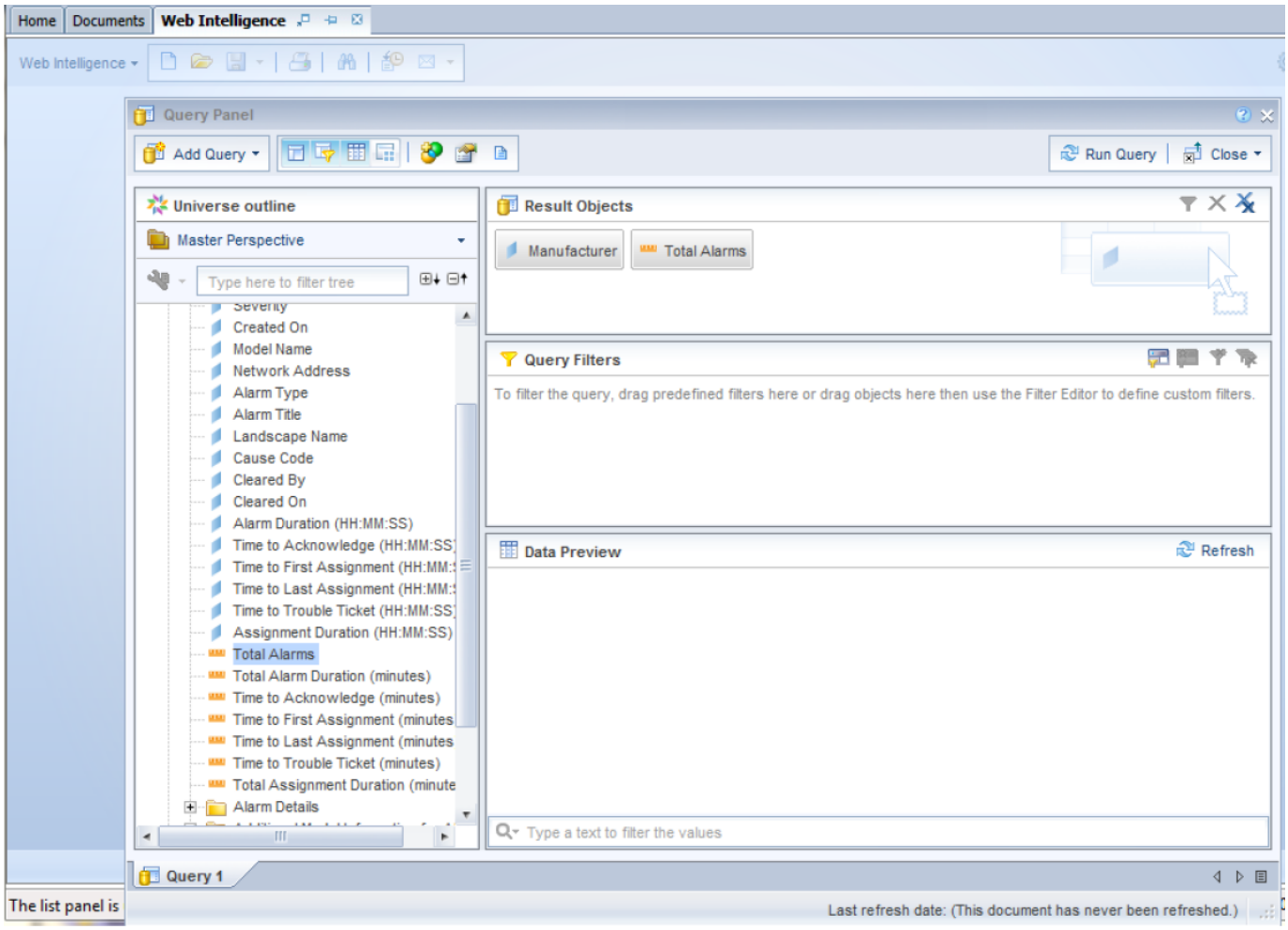
- Use time filters whenever possible to produce efficient running reports.
- Consider the amount of data your report is running against. For example, if you are generating a million events a day, running a year to date report on events likely negatively affect your performance. Specify the shortest time frame possible that produces the data that you require.

Follow these steps:

1. On the OneClick administration page, click BI Launch Pad.
2. Log in to the BI Launch pad.

   NOTE
   The process of initiating a new Web Intelligence session can take a few minutes while a Java applet loads.
3. On the top of the page, click Applications, Web Intelligence.
4. Click the icon in the Web Intelligence toolbar to open the Create a Document window ().
5. Select Universe from the data source list, click OK.
6. Select the 'Spectrum Ad Hoc - MySQL - EN' Universe type to launch a new DX NetOps Spectrum Ad Hoc Reporting Web Intelligence session.
   The Query Panel window appears. The Data panel structure parallels the DX NetOps Spectrum Reporting structure and displays the following nodes:
   - Asset
   - Event
   - Alarm
   - Alarm Activity
   - Availability/Outage
7. Click the + sign to expand folders and view data objects and filters that are related to the folder content.
8. Expand the Alarm folder to view objects.

Drag-and-drop fields from the Data panel to the Results Objects panel.
For example, from the Alarm panel, drag and drop the Manufacturer and Total Alarms objects into the Result Objects panel. The Manufacturer object is located within the 'Device Alarm Information' subfolder.
The following picture illustrates the process to drag and drop the Manufacturer and Total Alarm objects into the Result Objects panel:



9. Drag-and-drop Filter options.
   - Expand the 'Alarm Data Filters' subfolder to view only alarms that have occurred during the current calendar year.
   - Expand the 'Alarm Time Filters' subfolder. Then drag-and-drop the 'Year to Date (YTD)' filter into the 'Query Filters' panel.
10. Select Run Query to execute the query with the objects specified.
    A Report View perspective which contains the default report (simple, and un-formatted report) results display.
    The following picture shows the number of alarms that occurred during the calendar year by Device Manufacture.

Manufacturer	Total Alarms
Alcatel	1
Cisco	145
Enterasys	3
F5 Networks	2
HP	1
Microsoft	2
net-snmp	79
Network Appliance	1
Panthera Networks	3
Reserved/SNMP	5

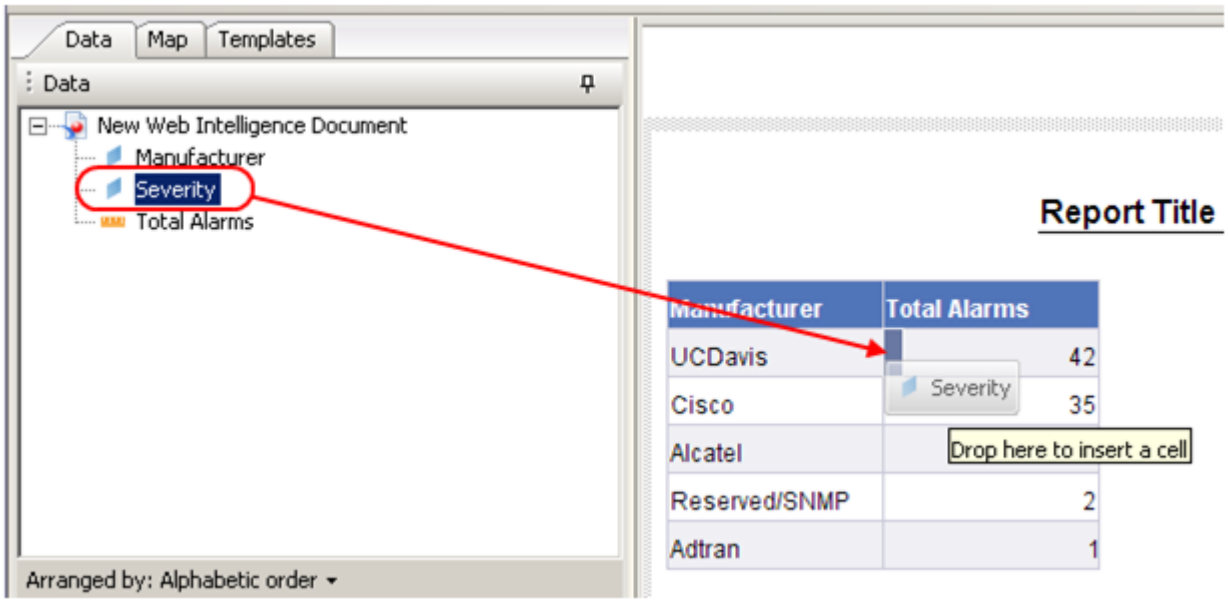
11. (Optional) Use the available formatting options to format the report (such as Title, Templates).
12. To save the document to the My Favorites folder, click Save As. (Optional) You can save the report to your local desktop or can publish the report to the CABI enterprise.
    A Simple Ad Hoc Report is generated.

Generate a Complex Ad Hoc Report with Advanced Formatting

You can generate a Complex Ad Hoc report by modifying a Simple Ad Hoc Report. You can further format your reports to make them more useful.

Follow these steps:

1. Access a Simple Ad Hoc report that you have generated.
   The baseline report captures the number of alarms that have occurred this year, sorted by the device manufacturer.
2. To sort the data, select the column header to sort, and then click Sort.
3. Select Edit Query.
   For example, you can perform the following steps to format your reports:
   a. Drag the Severity object located in the Alarm folder to the Result Objects panel.
   b. Select Run Query.
   c. Run the modified query.
      The Report View perspective displays.
      Report is updated to include the new Severity field that is added to the report query.
4. Drag-and-drop the Severity object into the report table.
   The updated report table now contains three columns.



5. To convert the existing table into a crosstab table, where Severity is on the Y-axis and Manufacturer is on the X-axis, drag the Manufacturer column heading above the table.

   NOTE
   Do not release the column heading until a tooltip appears indicating that a crosstab is created.

   The two-dimensional array enables you to see the total number of alarm counts breakdown by severity within each manufacturer. In addition, the crosstab is helpful for comparing alarm counts across device manufacturers.
6. To add subtotals to the table, perform the following steps:
   a. Highlight all the alarm count cells in the table (do not select the row/column headers).
   b. Click Sigma.
   c. Select Sum.
      The crosstab contains subtotals by manufacturer and severity.
7. To create a graphical representation of this data, such as a bar chart, perform the following steps:
   a. To copy the existing crosstab, select the entire crosstab, right-click, and select Copy.
   b. Move the cursor directly underneath the existing crosstab, right-click, and select Paste.
      You can now see a second identical crosstab below the first crosstab.
   c. To convert the bottom crosstab to a horizontal bar chart, select the entire crosstab, right-click, and select Turn To.
      The Turn To window appears. The Turn To window contains multiple tabs, each tab corresponds to the different types of charts available.
   d. Click the Bar tab, select the Horizontal Stacked chart option, and click OK.
      An unformatted horizontal stacked bar chart appears directly underneath the crosstab.
8. Perform the following steps to add legend and data values to improve the format of your report.
   a. To add a legend, select the entire chart, and click the Data tab.
   b. Expand the Appearance section in the Properties box.
   c. Click the Legend checkbox.
   d. To add data values, select the entire chart in the Appearance section and the Values subsection.
   e. Click the Show data checkbox.
      The report displays with both legend and data values.
9. To change the chart to reflect the standard DX NetOps Spectrum Reporting coloring conventions for alarm severity (Red=Critical, Orange=Major, Yellow=Minor), perform the following tasks:
   a. Within the Appearance and subsequent Data section, select the Palette option.

b. From the Select Palette dialog, select Edit Palette.
c. Select colors (Red, Orange, and Yellow).
d. Click OK to change the color palette.
e. Click OK to apply the modified palette to the chart.

10. Update the report title by double-clicking the default title box and entering a new title.
    A Complex Ad Hoc report is generated with advanced formatting.
11. Click Drill to drill down on data within the report to view individual alarms.
    Continue to select fields in the table report or bars in the graphic report until you have the data that you require.

Use WEBI Sample Reports

DX NetOps Spectrum Reporting includes WEBI Sample Reports, which are based on Crystal Reports. The WEBI sample reports also showcase some of the WEBI features:

- Graphing
- Aggregate Data Functions
- Varied Report Layouts
- Crosstabs and Pivot Tables.

Access WEBI Sample Reports

You can access and run WEBI Sample reports from BI Launch Pad. Using the WEBI sample reports as templates, you can create your own WEBI-based reports and select relevant DX NetOps Spectrum data.

Follow these steps:

1. From the BI Launch Pad window, select Document List, Public Folders, CA Reports, and then DX NetOps Spectrum.
2. Select Sample WEBI Documents.
   The available Sample WEBI documents/reports appear.
3. Double-click a report.
   The report displays.
4. Select report parameter values, where required.
5. Click Run Query.
   The report displays.

Copying and Editing WEBI Sample Reports

The WEBI Sample Reports can be used as a base for copying and editing to create your own WEBI reports.

Important! We recommended *not* editing the WEBI Sample Reports directly. Copy the report to another folder, and edit the report.

Follow these steps:

1. Create a folder outside the CA Reports folder hierarchy, and under the Public Folders level to contain your customized WEBI reports.

   NOTE
   Create folders outside the CA Reports folder hierarchy, and under the Public Folders level is important as product upgrades may overwrite contents inside the CA Reports folder. For more information, see the CA Business Intelligence Integration section.
2. Access the WEBI Sample Reports.
3. Select the report that you want to copy, right-click, and select Organize Copy.

4. Select the report folder that you created, right-click, and select Organize Paste.
   The report appears in the right-hand side panel.
5. Highlight the report that you copied, right-click, and select Modify.
   The Ad Hoc Reporting panel appears.
   a. To edit the report display, click Edit Report.
   b. To edit the query, click Edit Query.
   The Ad Hoc Reporting Panel provides multiple features. For more information, see Generating Ad Hoc Reports .

User Resolving Java Error in Report Manager Sample (WEBI) Reports

Symptom:

A Java error appears in the Spectrum Report Manager Sample (WEBI) reports. When I open sample report, I see the following error message:

```
Java has discovered application components that could indicate a security concern -- Block potentially unsafe
 components from being run? (recommended). (Yes/No)
```

If I select Yes, the report results do not display. If I select No, the report results are displayed.

Solution:

The issue occurs when running your browser on a Windows system with versions higher than Java 6 update 17. To resolve this issue, perform the following steps:

1. Open Java from the Control Panel.
2. Select the Advanced Tab
3. Expand the Security option.
4. Expand the Mixed Code option.
5. Select 'Enable - hide warning and run with protections'.
   The report runs successfully on Windows.

Report Manager DB Schema

Introduction

Spectrum Report Manager uses a MySQL database named "reporting" to store data. Another database "srmdbapi" provides views that are based on the reporting DB to query for data. This section provides an overview of all the database tables and views in 'reporting' and 'srmdbapi' databases so that the users can run their own queries and can use reporting tools of their own.

At startup, Spectrum Report Manager retrieves the data from the primary Archive Manager for each SpectroSERVER through OneClick and stores the data in the SRM databases.

Connecting to the database

Users can connect to the SRM database(which is on the OneClick server machine) using a ‘terminal window’ or a dos prompt (windows -> start -> Run -> cmd).

Use the below instructions to connect to the MySQL SRM DB instance running on the OneClick host

1. change to your $SPECROOT/mysql/bin directory
2. do a “mysql -uroot –proot” to connect to the mysql DB system
3. do a “use reporting;” (don't forget the ; at the end!)
4. Then user can run any of the select statements to fetch and see the data (don't forget the ; at the end of the statement)
5. Once done, run “quit” to disconnect.

Users are also encouraged to use utilities like ‘MySQL Workbench’ to connect to the SRM database and run the required queries.

If user wants to connect to the SRM database remotely from another box through utilities then they have to run a grant command so that access to MySQL DB is allowed from that host.

1. Go in to the $SPECROOT/mysql/bin directory Eg: /usr/Spectrum/mysql/bin
2. Login to mysql with the command “mysql -uroot –proot”
3. Once we logged in, use the command “grant all privileges on *.* to 'root'@'<ip address of the machine from where we are accessing the DB>' identified by 'root';”
4. Example:- grant all privileges on *.* to 'root'@'10.132.15.160' identified by 'root'; (now user can access the reporting DB from MySQL workbench running on host 10.132.15.160)

Once the access is granted then users can connect to the database and can run queries using the workbench.

'reporting' Database

This section contains information about 'reporting' database Tables and Views.

Click here to see how to write sample queries using reporting DB tables.

srmdbapi Database

'reporting' Database

DX NetOps Spectrum Report Manager (SRM) uses a MySQL database named '**reporting**' to store data. This database contains all the tables that are required to store the data that is used by SRM application to generate reports.

At startup, the DX NetOps Spectrum Report Manager retrieves the data from the primary Archive Manager for each SpectroSERVER through OneClick and stores the data in the SRM databases.

Following is the list of tables and views in the '**reporting**' database.

The relations shown in the following diagrams merely indicate table relations. They do not indicate 1:1 or 1: many relations.

Interpret the column icons as indicated below:

- Represents a Primary Key

- Simple NOT NULL column

- Simple column which can be NULL

Tables

alarm_user

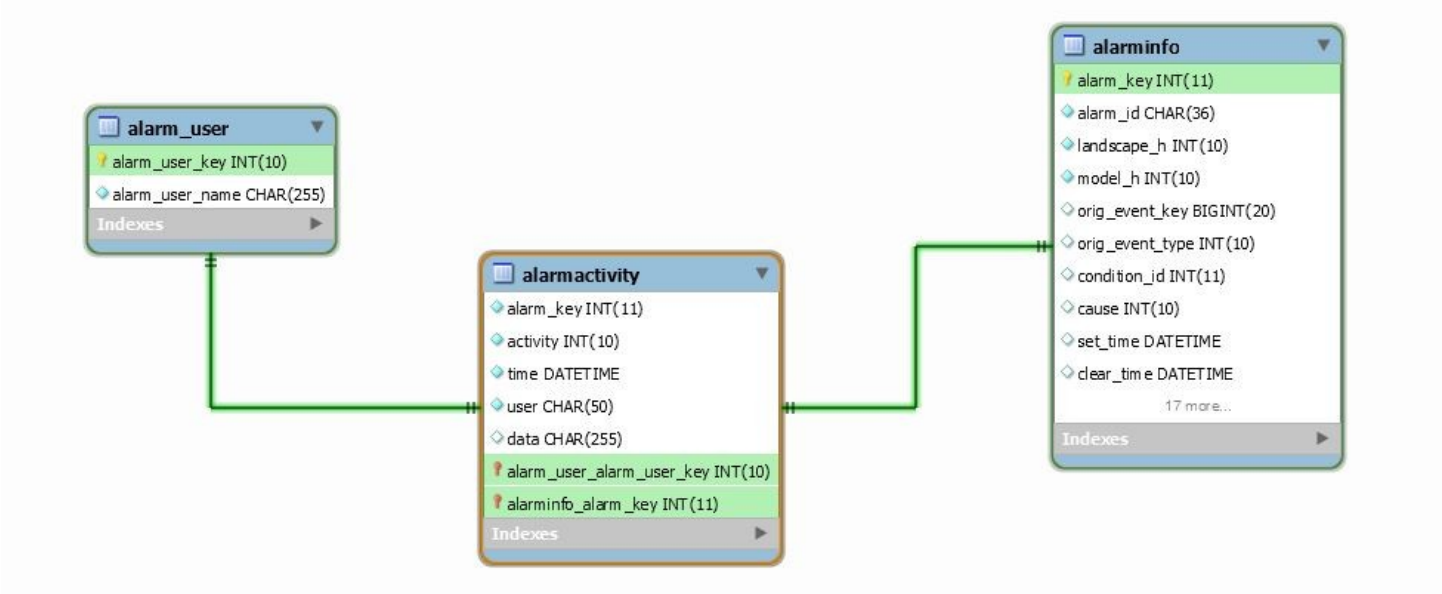
Description

This table lists the users who can manage the alarms in DX NetOps Spectrum.

Columns

Field	Type	Null	Key	Default	Extra	Comment
alarm_user_name	char(255)	NO	UNI			Name of the DX NetOps Spectrum user, who can manage the alarms.
alarm_user_key	int(10) unsigned	NO	PRI		auto_increment	Unique key for each user, who manages the alarm in DX NetOps Spectrum

Relations



alarmactivity

Description

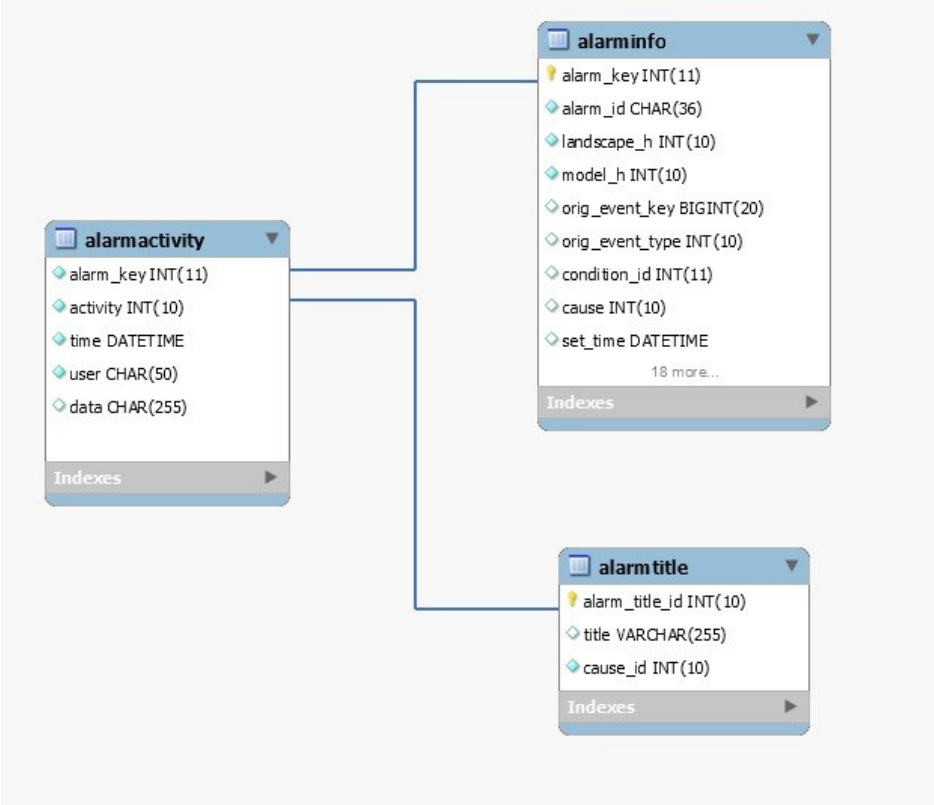
The alarmactivity table stores all the alarm activity that is monitored by SRM. The activity field denotes the type of alarm event generated. This field can be one of the following:

- Set alarm event
- Acknowledge alarm event
- Assign troubleshooter alarm event
- Clear alarm event
- User cleared alarm event
- Assign trouble ticket alarm event

Columns

Field	Type	Null	Key	Default	Comment
user	char(50)	NO	MUL		Name of the user, handling the alarm
time	datetime	NO	MUL	0000-00-00 00:00:00	Time of the activity performed in the alarm.
data	char(255)	YES			
alarm_key	int(11) unsigned	NO	MUL		Key value for specific type of Alarms.
activity	int(10) unsigned	NO	MUL		Activity that is performed by the user on the specific alarm.

Relations



alarmcondition

Description

This table represents the alarm criticality and condition.

Columns

Field	Type	Null	Key	Default	Comment
criticality	tinyint(2)	NO	UNI		Criticality of the alarm like Minor, Major, Critical
condition_name	varchar(11)	NO			Condition name of the alarm, which is mapped to the criticality.
condition_id	int(10) unsigned	NO	PRI		Unique ID for the alarm condition.

Relations

alarminfo

Description

This table contains all the alarm information including the assignee and trouble ticket IDs.

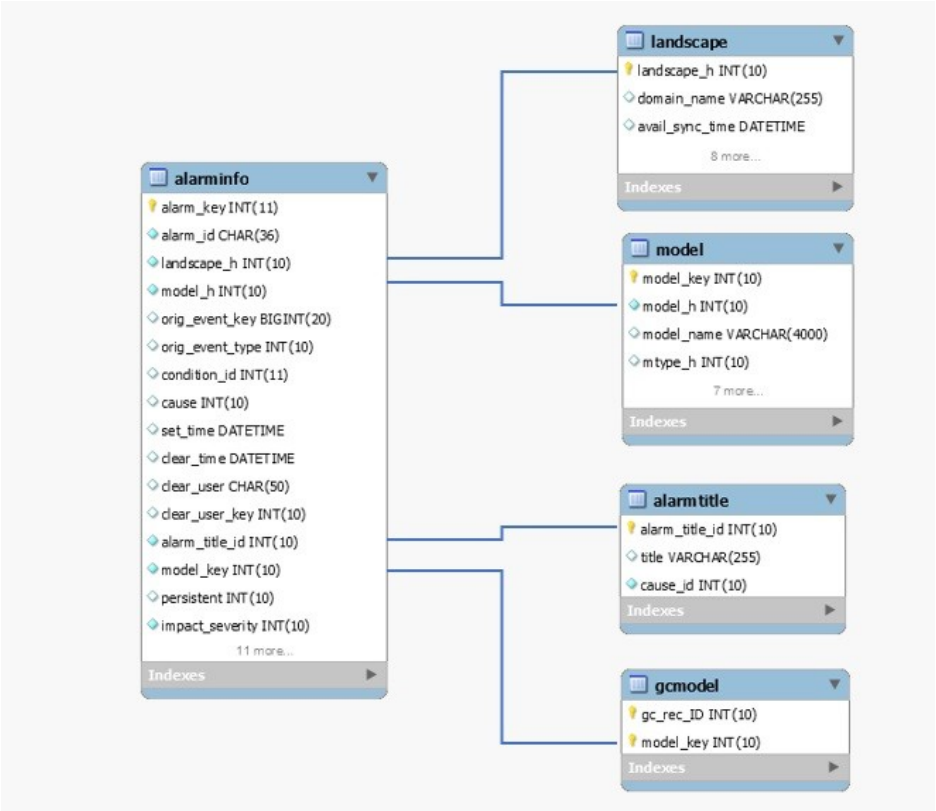
The alarminfo table stores relevant information for an alarm. There is one entry per unique alarm id, as opposed to the alarmactivity table which can have multiple entries for a single Alarm ID. An entry in this table is created when a “set alarm event” is received by SRM. The table is updated through the life of the alarm as each of the other alarm events are received by SRM.

Columns

Field	Type	Null	Key	Default	Comment
set_troubleticket_user_key	int(10) unsigned	YES	MUL		Trouble ticket user key for the specific alarm.
set_troubleticket_time	datetime	YES			Time of the trouble ticket assigned.
set_troubleticket_id	char(255)	YES			Trouble ticket ID of the alarm.
set_time	datetime	YES	MUL		Originated time of the alarm.
persistent	int(10) unsigned	YES	MUL	2	
orig_event_type	int(10) unsigned	YES			Originating event type for the alarm.
orig_event_key	bigint(20) unsigned	YES			Originating event key for the Alarm.
model_key	int(10) unsigned	NO	MUL	0	Unique model key for each alarm.
model_h	int(10) unsigned	NO	MUL		Model handle of the device, which has the alarm on it.
last_assigning_user_key	int(10) unsigned	YES	MUL		
last_assigned_user_key	int(10) unsigned	YES	MUL		
last_assigned_time	datetime	YES			

landscape_h	int(10) unsigned	NO			Landscape handle of the server, where the alarm present.
impact_severity	int(10) unsigned	NO		0	Severity of the alarm.
first_assigning_user_key	int(10) unsigned	YES	MUL		
first_assigned_user_key	int(10) unsigned	YES	MUL		
first_assigned_time	datetime	YES			First assigned time of the alarm to specific user.
condition_id	int(11)	YES			Condition of the alarm, which is mapped to the criticality.
clear_user_key	int(10) unsigned	YES	MUL		
clear_user	char(50)	YES			User who acknowledge the alarm.
clear_time	datetime	YES			When the alarm is cleared from DX NetOps Spectrum.
cause	int(10) unsigned	YES	MUL		Cause/Event ID of the specific alarm.
alarm_title_id	int(10) unsigned	NO	MUL	1	Unique ID for alarm title.
alarm_key	int(11) unsigned	NO	PRI		Unique ID for each alarm present in DX NetOps Spectrum
alarm_id	char(36)	NO	UNI		Alarm ID for specific type of alarms.
ack_user_key	int(10) unsigned	YES	MUL		Specific key for the user who acknowledges the alarm.
ack_time	datetime	YES			When the user acknowledge the alarm.

Relations



alarmtitle

Description

This table represents the title information of DX NetOps Spectrum alarms.

Columns

Field	Type	Null	Key	Default	Extra	Comment
title	varchar(255)	YES				Title of the Alarm.
cause_id	int(10) unsigned	NO	MUL			Cause or Event ID of specific type of alarm in DX NetOps Spectrum
alarm_title_id	int(10) unsigned	NO	PRI		auto_increment	Unique ID for each alarm present in DX NetOps Spectrum.

Relations

backups

Description

This table contains list of database backups.

bo_only_user

Description

This table includes the Business object user information.

boxi_user_sync

Description

This table includes the Business object user sync information.

bucketactivitylog

Description

This table includes the event information from the landscape to process in SRM database.

ca_reportstrings

Description

This table represents the information of localized string for the identifier.

configchangelog

Description

This table contains the information about NCM configuration file and logs.

Columns

Field	Type	Null	Key	Default	Extra	Comment
last_modified	bigint(20)	NO				Last_Modified time of the configuration file.
id	int(10) unsigned	NO	PRI		auto_increment	Unique ID of Configuration file.
filename	varchar(255)	NO				Name of the Configuration file.
content	blob	NO				Content of the configuration file

contentpkg

Description

This table represents the installation content package information.

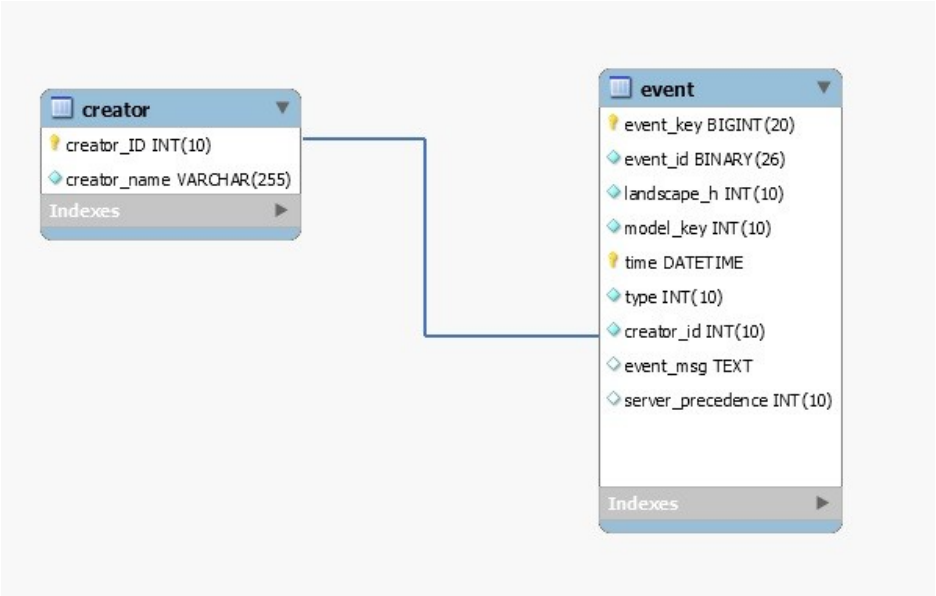
The contentpkg table associates content packages with Crystal Enterprise folder IDs. This table is not meant to be reported against, but instead is actually used by the Report Manager to help identify installation and security issues. A content package might only be installed once and this table helps identify if that is the case.

creator

Description

This table includes the model creator information.

Relations



data_retention_policy_changelog

Description

This table includes the information about the data retention policy.

devicemodel

Description

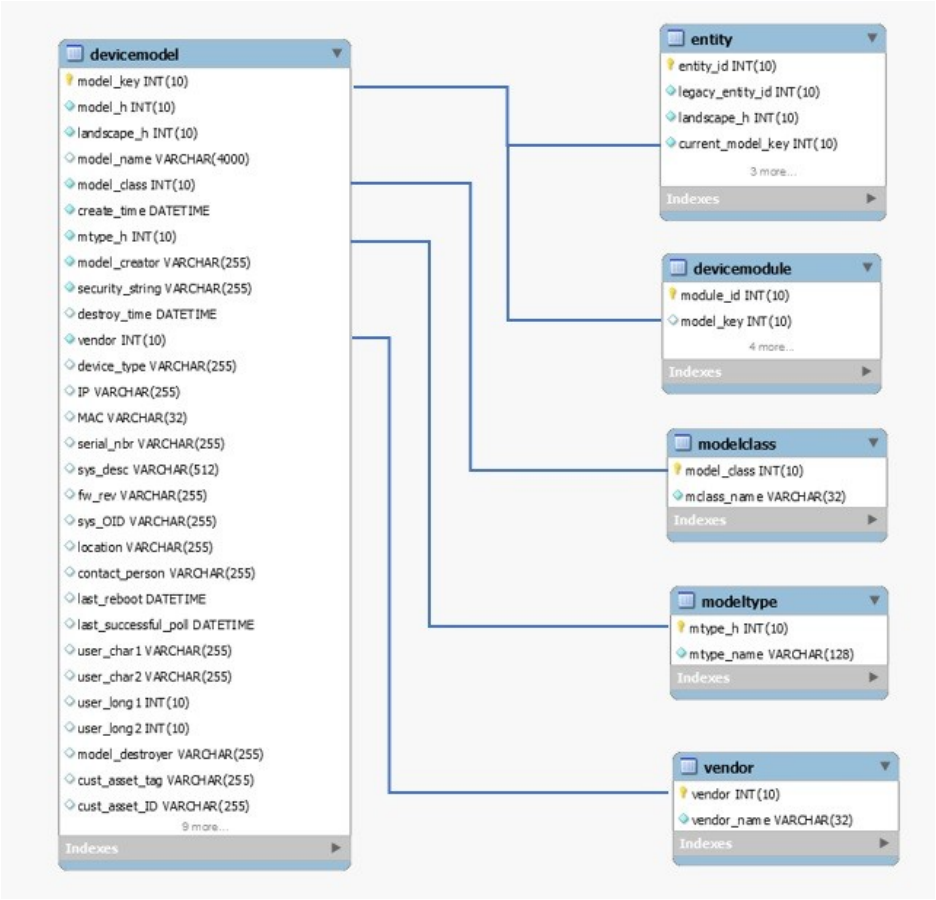
This table represents the complete device model information.

The devicemodel table is filled in initially as the Report Manager extracts model information from the respective SpectroSERVERs. New records are added by the Report Manager as it responds to model creation events for device models.

Certain attributes of this table can change and as such the Report Manager needs a way to keep up with these changes. To keep up with these changes, the Report Manager periodically requests current values for these attributes. These requests are made to the appropriate models through the OneClick architecture. Initially this period for updating device data is set to once every 24 hours.

The user-defined fields are to be left blank, but provide the administrator an opportunity to extend the Report Manager database to include data that is applicable to their assets.

Relations



devicemodel_uda

Description

The devicemodel_uda table contains user-defined polling attribute information.

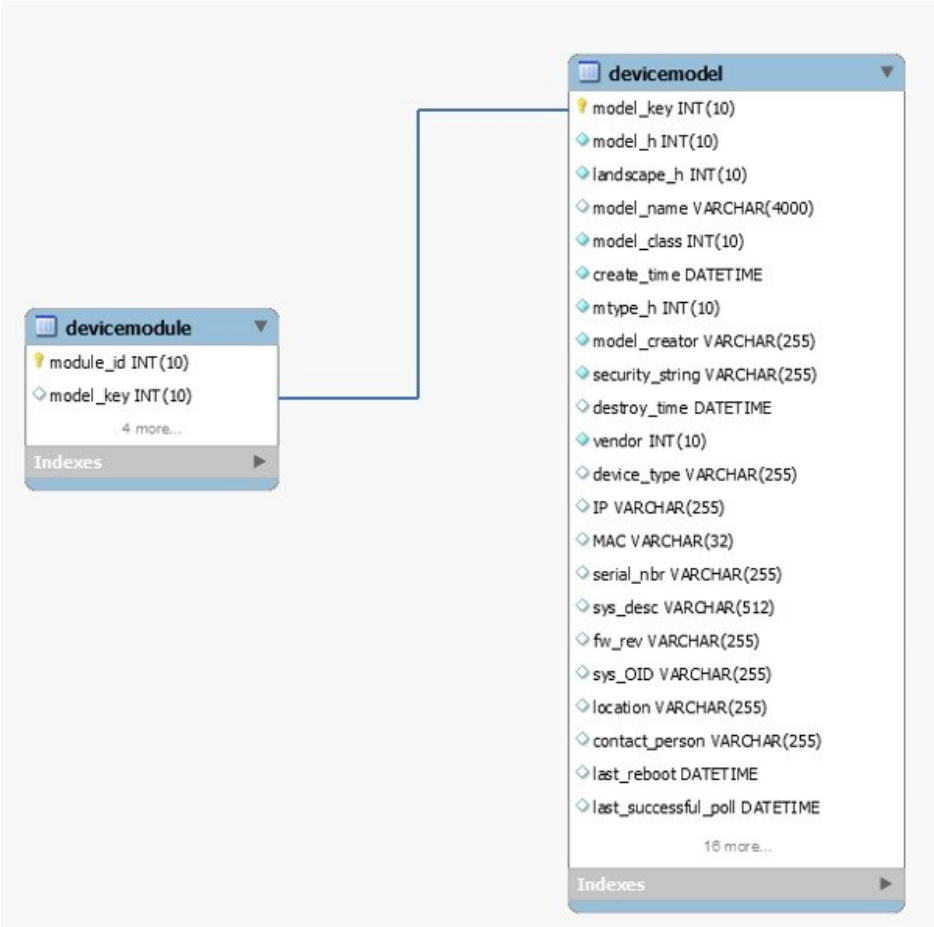
devicemodule

Description

This table includes the chassis information like module name, serial number etc.

This table captures the relationship between Chassis device models and the board modules contained within. This table is dynamically kept up to date.

Relations



Columns (till 10.2.1 release)

Field	Type	Null	Key	Default	Extra
module_id	int(10) unsigned	NO	PRI	NULL	auto_increment
model_key	int(10) unsigned	YES	MUL	NULL	
module_index	int(10)	YES		NULL	
module_name	varchar(255)	YES		NULL	
serial_nbr	varchar(255)	YES		NULL	
software_rev	varchar(255)	YES		NULL	

Columns
(from 10.2.2 release)

Field	Type	Null	Key	Default	Extra
module_id	int(10) unsigned	NO	PRI	NULL	auto_increment
model_key	int(10) unsigned	YES	MUL	NULL	
physical_index	int(10)	YES		NULL	
physical_modelname	varchar(255)	YES		NULL	
physical_name	varchar(255)	YES		NULL	

physical_class	int(10)	YES		NULL	
physical_contained_in	int(10)	YES		NULL	
serial_nbr	varchar(255)	YES		NULL	
software_rev	varchar(255)	YES		NULL	

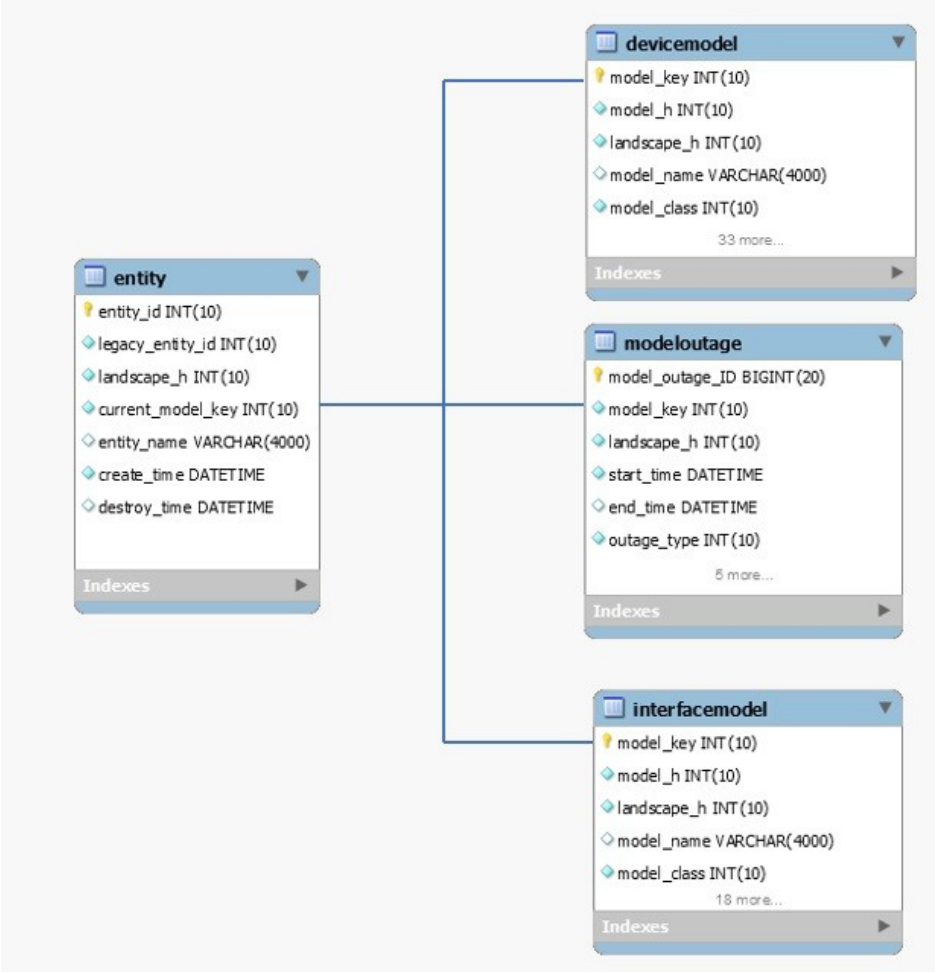
entity

Description

This table maintains distinct devices/interfaces across ALL landscapes.

The entity table is used to identify all entities uniquely that can be reported on. As new unique entities are added to the database, new entity records are created. Entity table record creation is closely tied to devicemodel and interfacemodel table record creation. The current_model, create_time, and destroy_time columns always correspond to the most recently created model.

Relations



entitygroup

Description

This table is about taking the device assets and grouping them into predefined groups that are based on Vendor, Model Class, Landscape.

The EntityGroup table is initially filled in during the startup of the Report Manager application. Queries are made to the individual SpectroSERVERs to learn of the existing model collections (which are actually models themselves).

The EntityGroup table is then kept up to date by having the Report Manager watch for the creation (and destruction) events of the collection models. When a new event occurs indicating the creation of one of these collection models, the name for that collection model is immediately obtained. A search of the EntityGroup table for a record with that name is performed. If no such record exists, one is immediately added. If a record does exist, no further processing is necessary.

entitygroupentity

Description

The EntityGroupEntity table is initially filled in during the startup of the Report Manager application. As EntityGroups are added to the system, queries are made back to each of the servers to determine the membership of those groups. In determining membership, the SpectroSERVER identifies a set of models. Each model can then be referenced in either the devicemodel or interfacemodel table. From there, an entity ID can be obtained and an appropriate entry can be made into this table.

The EntityGroupEntity table can then be kept up to date by monitoring the relationship changes associated with those collection models.

entitygrouptype

Description

The EntityGroupType table is filled in at the time of table creation. EntityGroupTypes are pre-defined before any EntityGroups have been defined. Table records include:

entity_group_type	eg_type_name
101	Vendor group
102	Model Class group
103	Landscape group
1000	User-Defined group
1001	User-Defined group

entitymodel

Description

The entitymodel is used to identify all model handles that an entity has had. This table gets filled in as part of the Entity table updating. When a record gets added to either the devicemodel or interfacemodel table, a process is kicked off to identify if this “new” model is either an existing/known entity, or a new (previously unknown/un-modeled) entity.

A new record gets added to the entitymodel table every time a “new” model record gets added to either the devicemodel or devicemodel table. When a record is added to the entitymodel table, the record is recorded with a timestamp. This timestamp enables the Report Manager to identify the most current model that represents an entity.

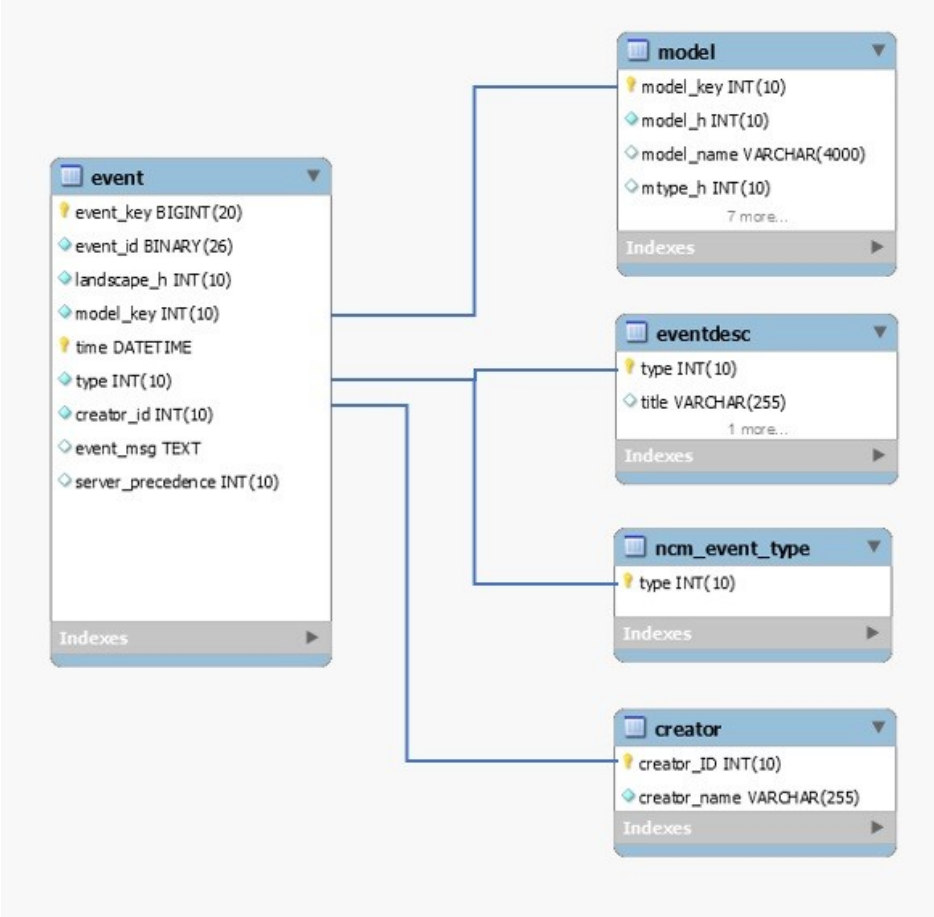
Columns

event

Description

This table described the complete event information like event ID, message, time etc.

Relations



eventactivitylog

Description

Contains number of events processed in each polling cycle for every landscape.

Columns

Field	Type	Null	Key	Default	Extra	Comment
log_id	int(10) unsigned	NO	PRI		auto_increment	Unique log id for each event processing polling cycle
landscape_h	int(10) unsigned	NO	MUL			landscape handle of SS

poll_start_time	datetime	NO				The time when polling cycle started
poll_end_time	datetime	YES				The time when polling cycle end
nbr_of_events	int(10) unsigned	YES				number of events processed in the current polling cycle

eventdesc

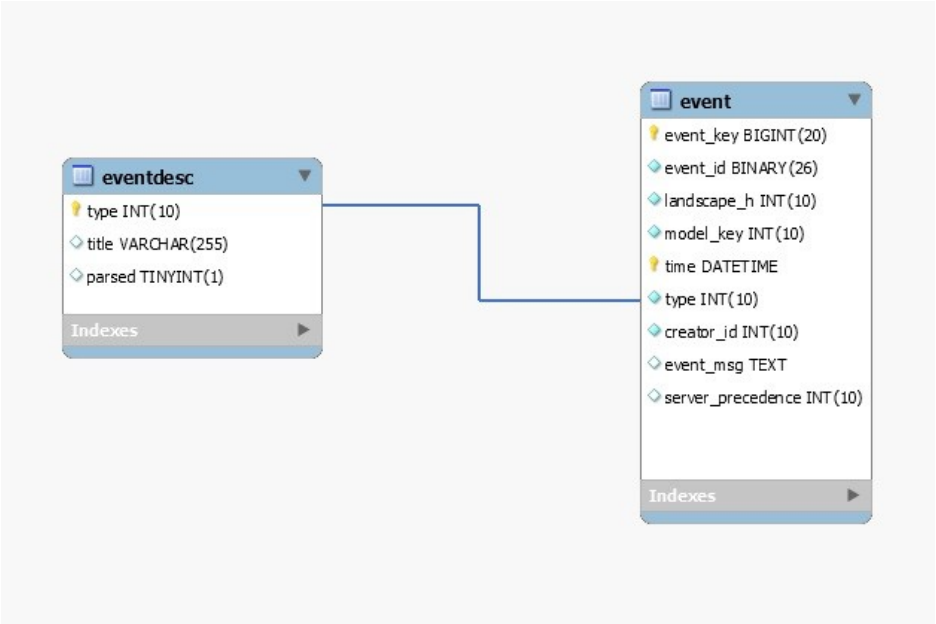
Description

Contains static list of events.

Columns

Field	Type	Null	Key	Default	Comment
type	int(10) unsigned	NO	PRI	0	Event Type
title	varchar(255)	YES			Event title
parsed	tinyint(1)	YES		0	

Relations



folderhierarchy

Description

Contains folder hierarchy present in OneClick explorer view.

Columns

folderidmap

Description

This table maps the SRM entitygroupid of the folder to the CsUniqueID that identifies the folder in One Click and DX NetOps Spectrum. This table is used for custom collection hierarchies.

Columns

Field	Type	Null	Key	Default	Comment
cs_unique_id	char(36)	NO	PRI		
folder_id	int(10) unsigned	NO	MUL		

gcmodel

Description

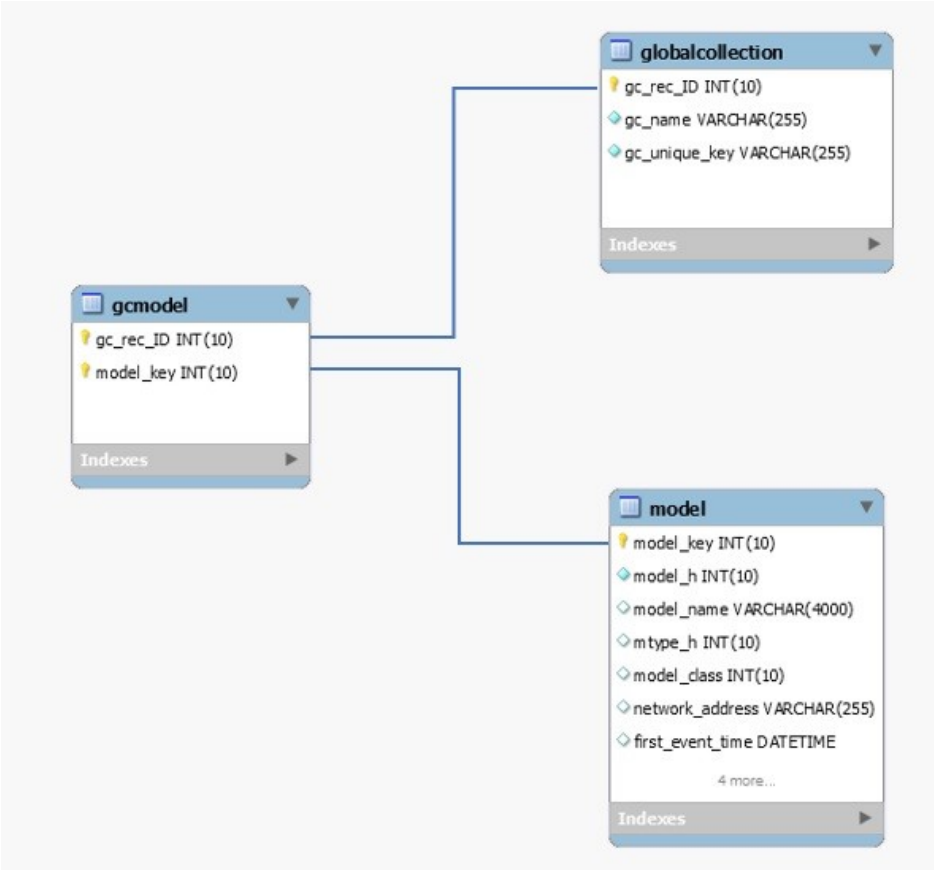
Contains mapping between global collections and models present in it.

Please note that this table does not include a flag like 'destroyed' or 'startdate/enddate' for the relation between the model and the global collection. It only represents the current assignment.

Columns

Field	Type	Null	Key	Default	Comment
gc_rec_ID	int(10) unsigned	NO	PRI		global collection record ID
model_key	int(10) unsigned	NO	PRI		model key of the model present in the global collection

Relations



globalcollection

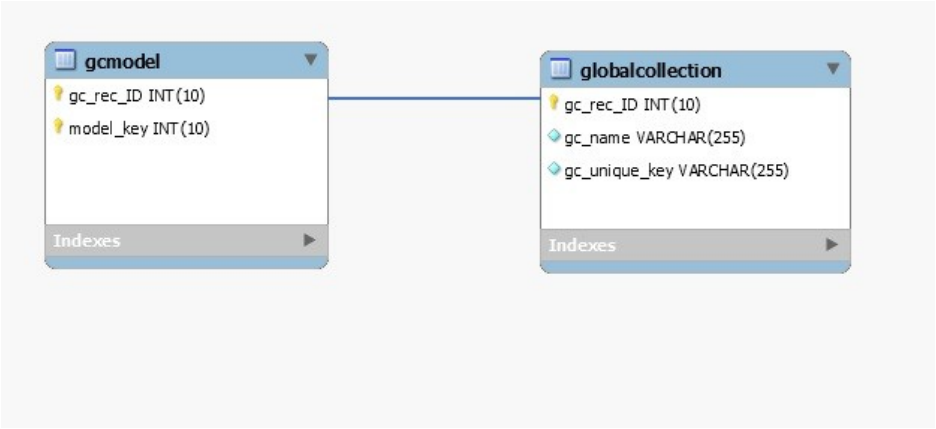
Description

Contains list of global collections.

Columns

Field	Type	Null	Key	Default	Extra	Comment
gc_rec_ID	int(10) unsigned	NO	PRI		auto_increment	unique record ID for each global collection
gc_name	varchar(255)	NO				Name of global collection
gc_unique_key	varchar(255)	NO	UNI			unique key for each global collection

Relations



groupentitygroups

Description

This table stores mapping of group_id to entity _group_id .

Since entity groups of type 'folder' can be made up of multiple entity groups (of type folder, collection or both), this table allows you to find all the entity groups that make up the specified entity group.

> **NOTE**
> When finding all entities of a specified entity group, use this table, not the entitygroup table.

Columns

Field	Type	Null	Key	Default	Comment
group_id	int(10) unsigned	NO	PRI		
entity_group_id	int(10) unsigned	NO	PRI		

handleractivitylog

Description

Containing the log of bucket processing done by all handlers for each landscape.

Columns

Field	Type	Null	Key	Default	Extra	Comment
log_id	int(10) unsigned	NO	PRI		auto_increment	Unique log id for each processing done by the handler
landscape_h	int(10) unsigned	NO	MUL			Landsacpe handle
staging_table	varchar(255)	NO	MUL			Name of the bucket which is getting processed
process_start_time	datetime	NO				Processing start time

process_end_time	datetime	YES				Processing end time
last_event_seq	int(10) unsigned	YES				
event_log_id	int(10) unsigned	NO	MUL			

handlerrollback

Description

This table stores model outage rollback data.

Columns

Field	Type	Null	Key	Default	Comment
landscape_h	int(10) unsigned	NO	PRI		
model_outage_update_ongoing	int(10) unsigned	NO		0	
last_model_outage_id	bigint(20) unsigned	YES			
last_model_outage_event_key	bigint(20) unsigned	YES			
dev_outage_update_ongoing	int(10) unsigned	NO		0	
last_dev_outage_id	int(10) unsigned	YES			
last_dev_event_time	datetime	YES			
int_outage_update_ongoing	int(10) unsigned	NO		0	
last_int_outage_id	int(10) unsigned	YES			
last_int_event_time	datetime	YES			

installedreports

Description

This table contains the file names of the reports that are loaded into the database by SRM. The report ID of the report that is stored in the database with its parent folder ID are also listed. This table is consulted when add or updating reports within SRM.

Columns

Field	Type	Null	Key	Default	Comment
report_ID	int(10) unsigned	NO	PRI		
parent_folder_ID	int(10) unsigned	NO			
file_name	varchar(255)	NO			

interfacemodel

Description

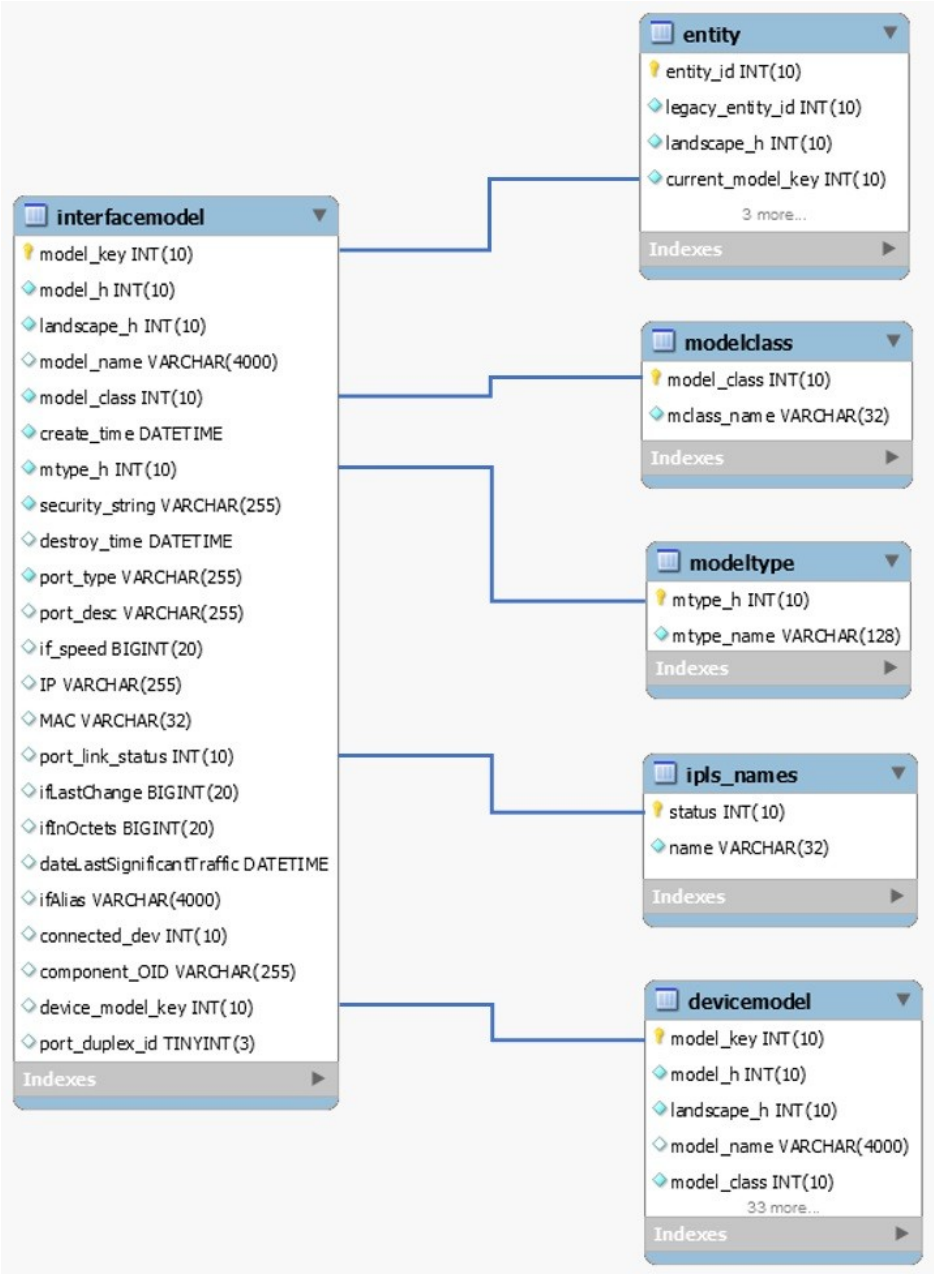
Contains interface information for all device models.

The interfacemodel table is filled in initially as the Report Manager extracts model information from the respective SpectroSERVERs. New records are added by the Report Manager as it responds to model creation events for interface models. The models that are reported on within the interfacemodel table are initially limited to those models that represent physical interfaces. Logical interfaces are not recognized or reported on with this first phase.

Columns

Field	Type	Null	Key	Default	Comment
model_key	int(10) unsigned	NO	PRI	0	model key of the interface
model_h	int(10) unsigned	NO	MUL		model handle of interface
landscape_h	int(10) unsigned	NO	MUL		Landscape handle of SS where the model is present
model_name	varchar(4000)	YES	MUL		Model name of the interface
model_class	int(10) unsigned	NO	MUL		model class of the interface
create_time	datetime	NO			Interface model creation time
mtype_h	int(10) unsigned	NO	MUL		Model Type Handle of the interface
security_string	varchar(255)	NO			security string of interface
destroy_time	datetime	YES			Destroy time of the interface model
port_type	varchar(255)	NO	MUL		Interface / port type
port_desc	varchar(255)	YES			Interface or port description
if_speed	bigint(20) unsigned	YES			Speed of the interface
IP	varchar(255)	YES			IP address of the interface
MAC	varchar(32)	YES			MAC address of the interface
port_link_status	int(10) unsigned	YES	MUL		Port or Interface link status
ifLastChange	bigint(20) unsigned	YES			value of ifLastChange mib attribute
ifInOctets	bigint(20) unsigned	YES			Value of ifInOctets mib attribute
dateLastSignificantTraffic	datetime	YES			The time when the last traffic is seen on the interface or port
ifAlias	varchar(4000)	YES			Contains the value of ifAlias mib attribute
connected_dev	int(10) unsigned	YES			Contains the interface to which this is connected
component_OID	varchar(255)	YES			
device_model_key	int(10) unsigned	YES	MUL		Model key of the device
port_duplex_id	tinyint(3) unsigned	YES			

Relations



ipls_names

Description

This table contains the different values for the port_link_status on an interface model found in the interfacemodel table. This table is filled when it is created with the following values:

Status	Name
0	Good
1	Bad
2	Unknown

3	Disabled
4	Unreachable
5	Init
6	L inked Port Bad
7	Linked Device Bad
8	Dormant
9	Port In Maintenance
10	Bad Suppressed
11	WA Link Bad
12	LL In Maintenance
13	Always Down

Columns

Field	Type	Null	Key	Default	Comment
status	int(10) unsigned	NO	PRI		Status number
name	varchar(32)	NO			Status name

Relations

landscape

Description

Contains the last poll /sync times for each landscape

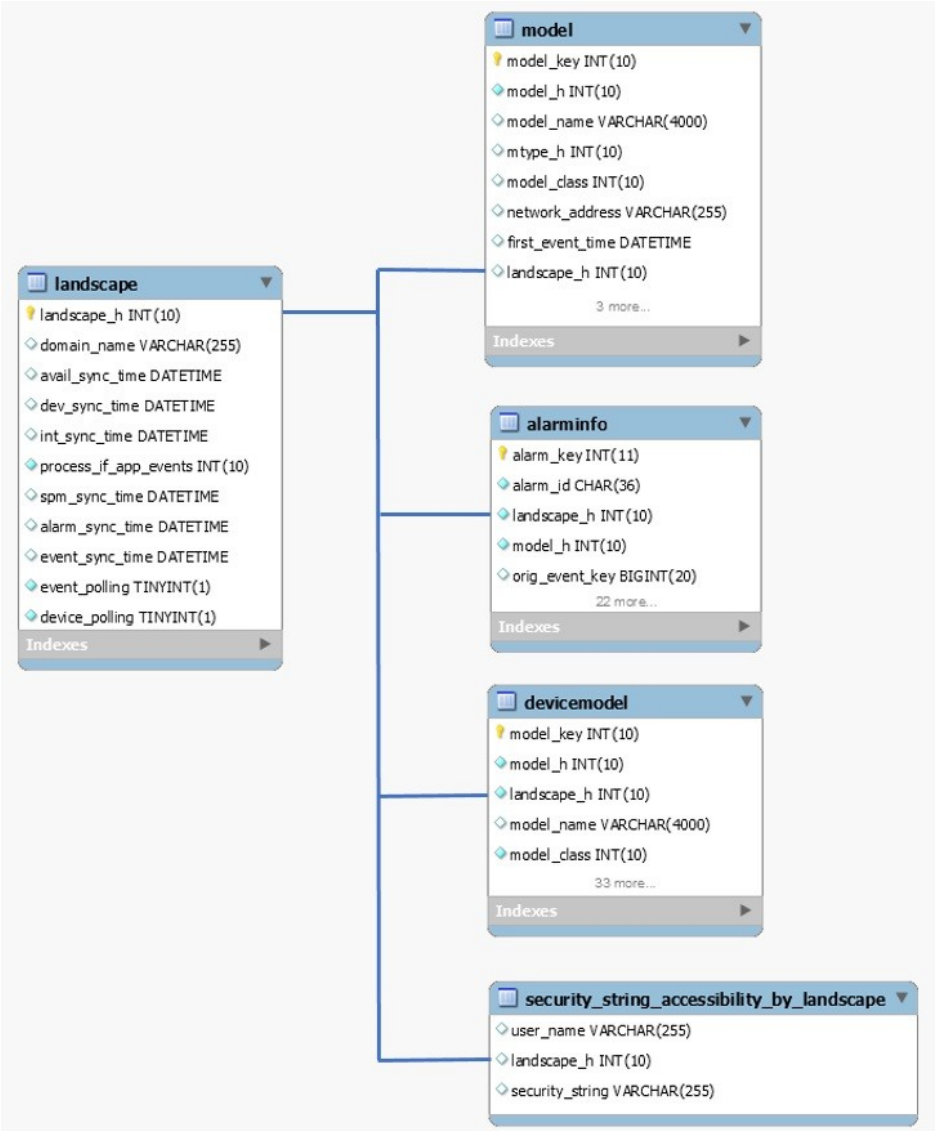
The landscape table lists those landscapes that report manager has seen. The dev_sync_time is the time the last known device event was recorded for the given landscape. The int_sync_time time is the time the last known interface event was recorded.

Columns

Field	Type	Null	Key	Default	Comment
landscape_h	int(10) unsigned	NO	PRI		Landscape handle which is polled by report manager
domain_name	varchar(255)	YES			Domain name of the landscape
avail_sync_time	datetime	YES			The last sync time for availability
dev_sync_time	datetime	YES			The last sync time for device availability
int_sync_time	datetime	YES			The last sync time for interface information

process_if_app_events	int(10)	NO		0	Indicates if it is ok to process application lost /react events for the interfaces
spm_sync_time	datetime	YES			The last sync time for SPM information
alarm_sync_time	datetime	YES			The last sync time for alarms information
event_sync_time	datetime	YES			The last sync time for events information
event_polling	tinyint(1)	NO		1	contains 1 if the event polling is enabled for this landscape else 0
device_polling	tinyint(1)	NO		1	Contains 1 if Asset polling is enabled else 0

Relations



managementoutage

Description

This table stores the management outages for the monitored landscapes.

Columns

Field	Type	Null	Key	Default	Extra	Comment
outage_ID	int(10) unsigned	NO	PRI		auto_increment	
landscape_h	int(10) unsigned	NO	MUL			
start_time	datetime	NO				
end_time	datetime	YES				
outage_type	int(10) unsigned	NO	MUL			

managementoutagetype

Description

This table lists the different types of management outages. This table is filled at table creation time with the following values:

- Expected
- Unexpected
- History

Columns

Field	Type	Null	Key	Default	Comment
outage_type	int(10) unsigned	NO	PRI		
outage_desc	varchar(32)	NO			

model

Description

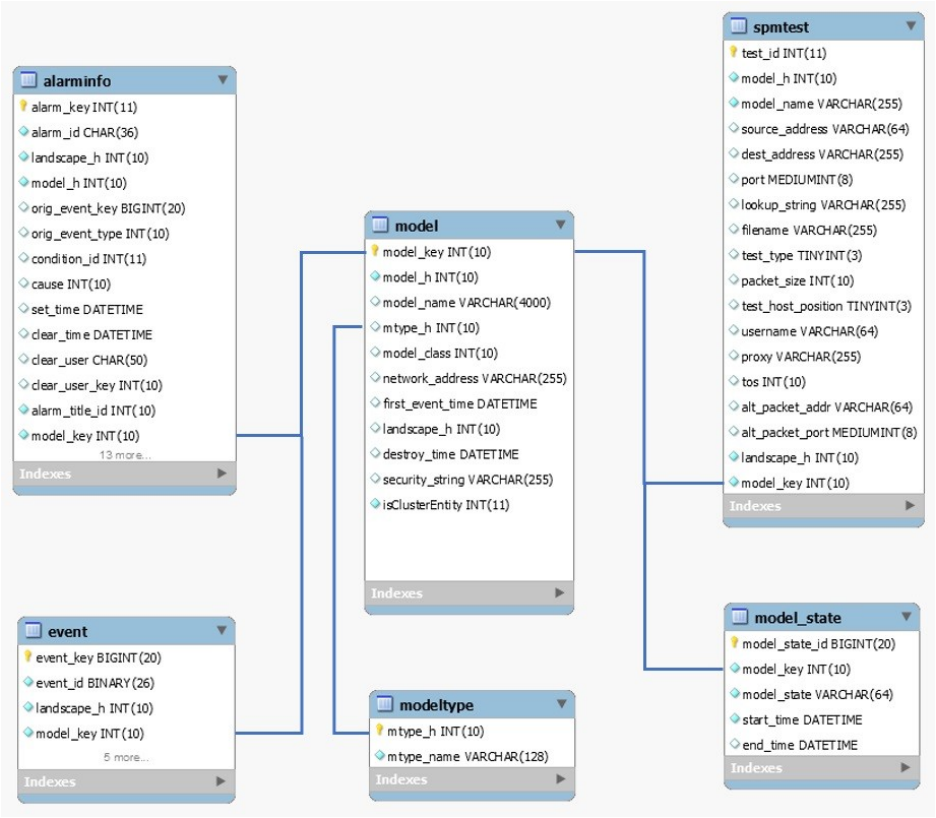
Contains information of all the DX NetOps Spectrum models. This table stores general model information.

Columns

Field	Type	Null	Key	Default	Extra	Comment
model_key	int(10) unsigned	NO	PRI		auto_increment	Unique Model Key for model
model_h	int(10) unsigned	NO	MUL	0		Model Handle of the model
model_name	varchar(4000)	YES	MUL			Name of the Model
mtype_h	int(10) unsigned	YES	MUL			Model Type Handle of the model
model_class	int(10) unsigned	YES	MUL			model class of the model

network_address	varchar(255)	YES				IP or Network address of the model
first_event_time	datetime	YES		2000-01-01 00:00:00		The time when first event got generated on the model
landscape_h	int(10) unsigned	YES	MUL			Landscape handle of SS where the model is present
destroy_time	datetime	YES	MUL			The time when model is destroyed
security_string	varchar(255)	YES	MUL	*UNKNOWN*		security string of the model
isClusterEntity	int(11)	NO		0		Flag that indicated if the model is cluster or not

Relations



model_state

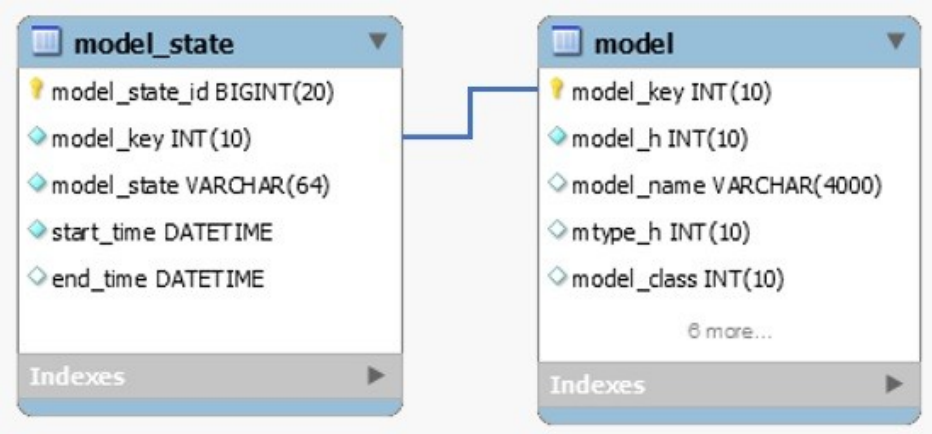
Description

Contains information about model state.

Columns

Field	Type	Null	Key	Default	Extra	Comment
model_state_id	bigint(20) unsigned	NO	PRI		auto_increment	Unique ID for each model state
model_key	int(10) unsigned	NO	MUL			Model key of the model
model_state	varchar(64)	NO	MUL			Model state
start_time	datetime	NO				The time when the model is in this state
end_time	datetime	YES				The time when the model is no longer in this state

Relations



modelclass

Description

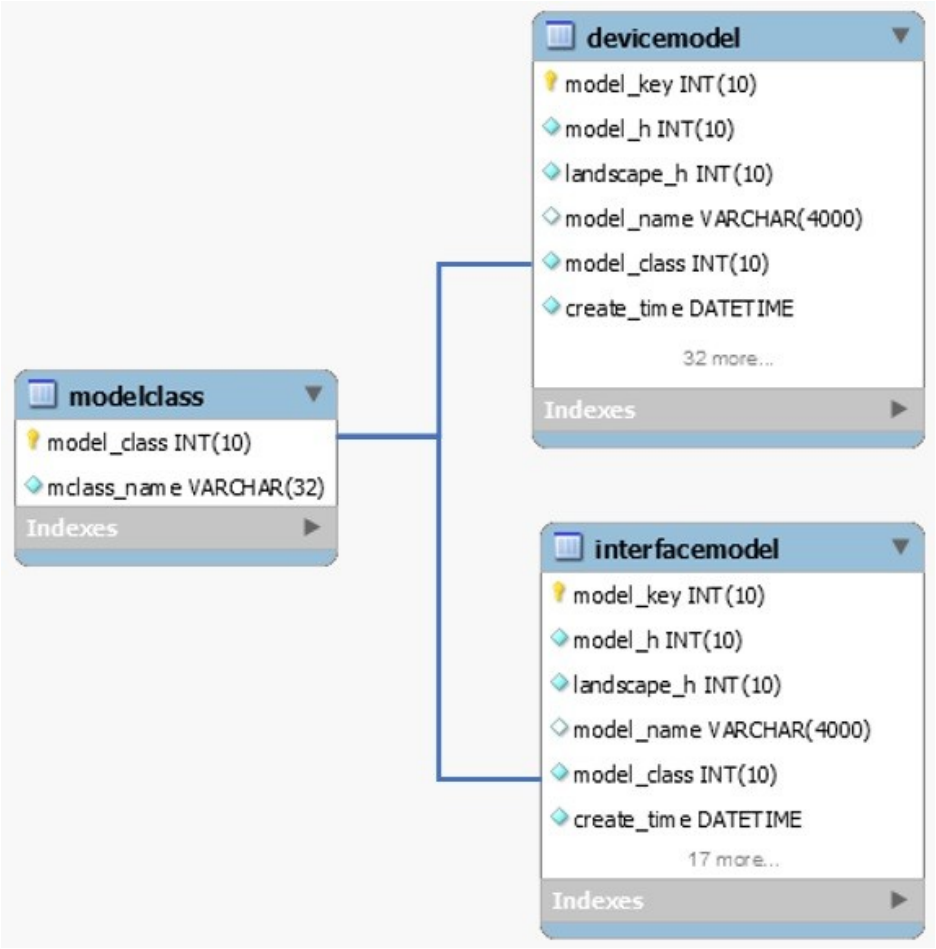
Contains static list containing list of all model class names.

The modelclass table is filled in at the time of table creation.

Columns

Field	Type	Null	Key	Default	Extra	Comment
model_class	int(10) unsigned	NO	PRI			unique number for model class
mclass_name	varchar(32)	NO	MUL			Name of the model class

Relations



modeloutage

Description

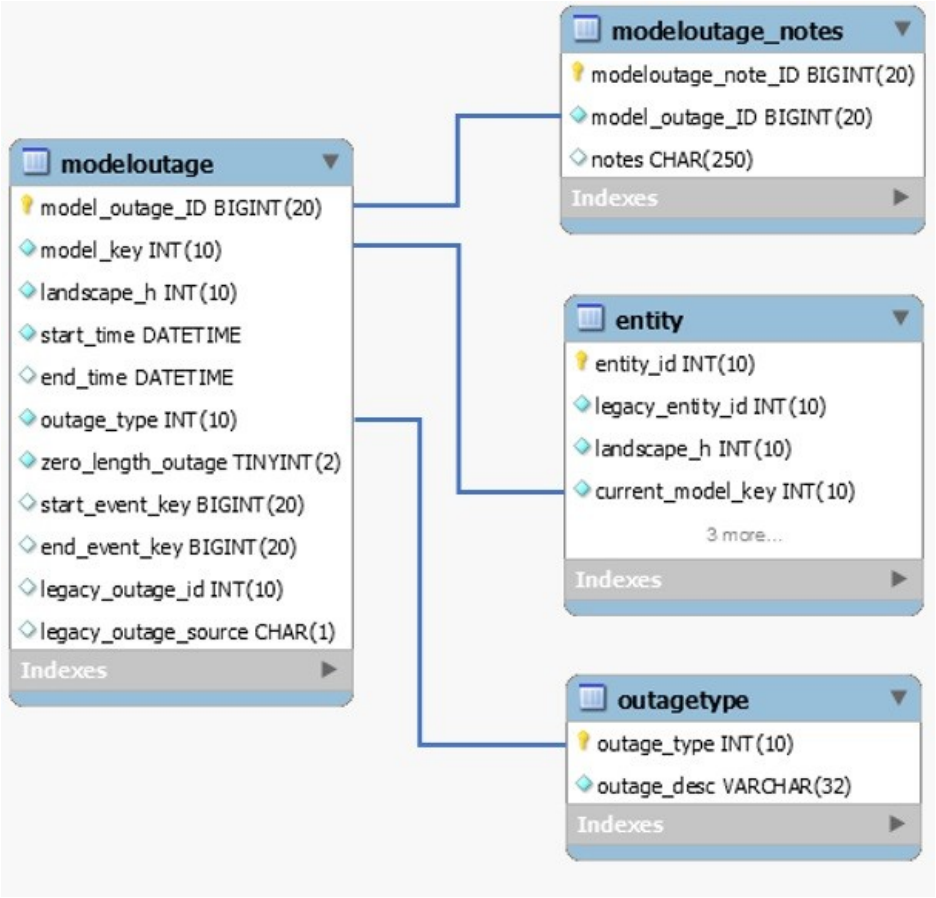
Contains all the outages for the model.

Columns

Field	Type	Null	Key	Default	Extra	Comment
model_outage_ID	bigint(20) unsigned	NO	PRI		auto_increment	Unique ID for each outage of the model
model_key	int(10) unsigned	NO	MUL			Model Key of the model
landscape_h	int(10) unsigned	NO	MUL			landscape handle of SS where the model is present
start_time	datetime	NO	MUL			The time when the outage started for the model
end_time	datetime	YES	MUL			The time when the outage got end for the model
outage_type	int(10) unsigned	NO	MUL			Type of outage on the model
zero_length_outage	tinyint(2)	NO	MUL	0		
start_event_key	bigint(20) unsigned	YES	MUL			

end_event_key	bigint(20) unsigned	YES	MUL			
legacy_outage_id	int(10) unsigned	YES				
legacy_outage_source	char(1)	YES				

Relations



modeloutage_notes

Description

Contains notes for each outage for the model.

Columns

Field	Type	Null	Key	Default	Extra	Comment
modeloutage_note_ID	bigint(20) unsigned	NO	PRI		auto_increment	unique ID for each model outage note
model_outage_ID	bigint(20) unsigned	NO	MUL			Outage ID corresponding to the outage ID in model table
notes	char(250)	YES				model outage notes

Relations

modeltype

Description

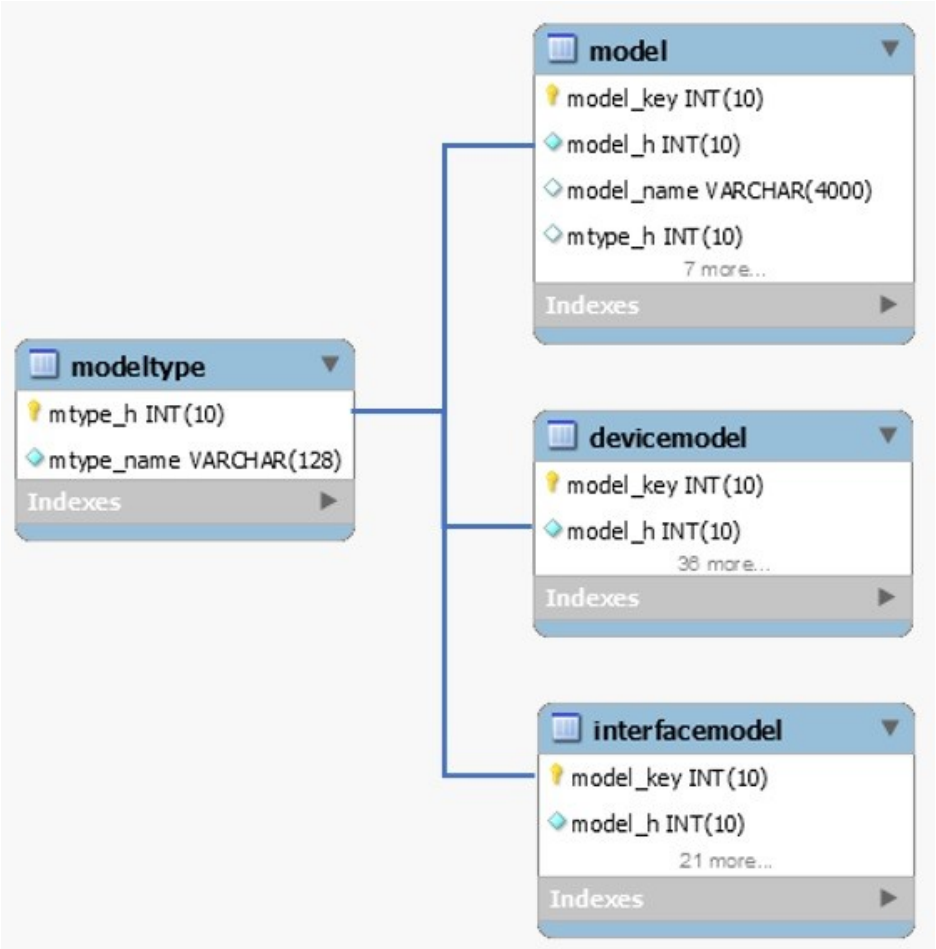
Contains Static list of all model type handles and their names.

The modeltype table is filled in as the Report Manager is starting up. The Report Manager contacts one of the SpectroSERVERs and sends a query requesting the handle and name for all device model types. Once this query is returned, the modeltype table is updated.

Columns

Field	Type	Null	Key	Default	Comment
mtype_h	int(10) unsigned	NO	PRI		Model type handle
mtype_name	varchar(128)	NO			Model Type Name

Relations



ncm_config

Description

Columns

Field	Type	Null	Key	Default	Extra	Comment
config_id	int(10) unsigned	NO	PRI		auto_increment	
config_text_id	int(10) unsigned	NO	MUL			
device_model_key	int(10) unsigned	NO				
change_time	datetime	YES				
trap_from	varchar(255)	NO				
trap_user	varchar(255)	NO				
trap_on	varchar(128)	YES				
lines_changed	int(11)	YES				
rel_lines_changed	int(11)	YES				
violated_policies	varchar(3000)	YES				

compliant_policies	varchar(3000)	YES				
spec_user_name	varchar(255)	YES				
comm_mode	varchar(128)	YES				
landscape	varchar(128)	YES				

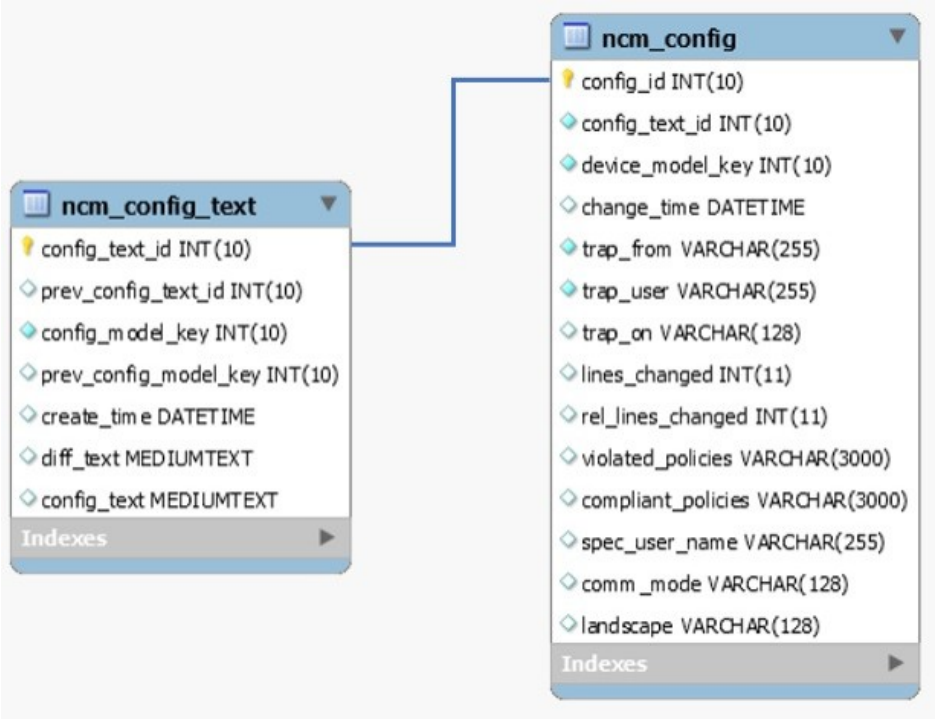
ncm_config_text

Description

Columns

Field	Type	Null	Key	Default	Extra	Comment
config_text_id	int(10) unsigned	NO	PRI		auto_increment	
prev_config_text_id	int(10) unsigned	YES	MUL			
config_model_key	int(10) unsigned	NO	MUL			
prev_config_model_key	int(10) unsigned	YES				
create_time	datetime	YES				
diff_text	mediumtext	YES				
config_text	mediumtext	YES				

Relations



ncm_event_type

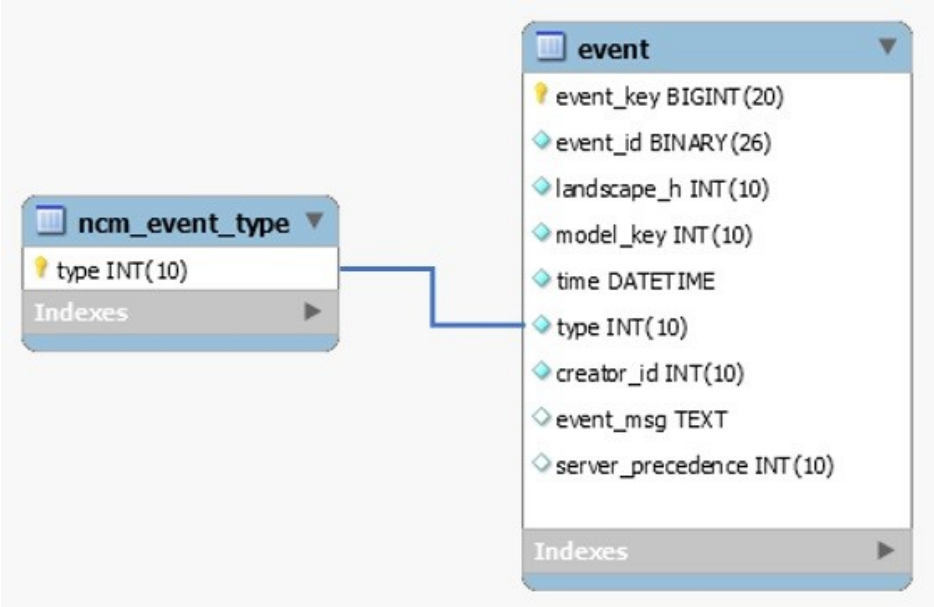
Description

Contains static list of all NCM event types

Columns

Field	Type	Null	Key	Default	Extra	Comment
type	int(10) unsigned	NO	PRI			NCM Event type

Relations



oc_user

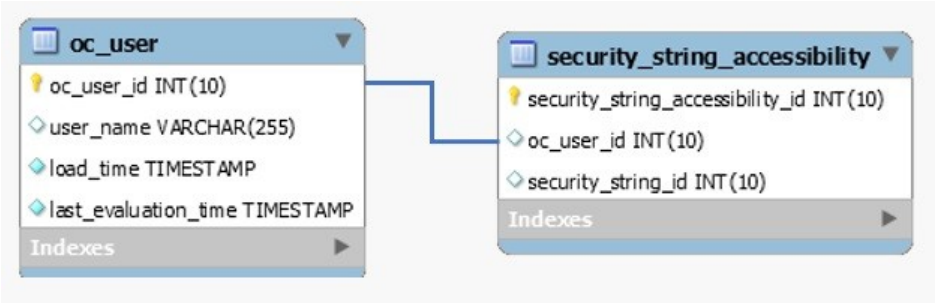
Description

This table contains list of oneclick users

Columns

Field	Type	Null	Key	Default	Extra	Comment
oc_user_id	int(10) unsigned	NO	PRI		auto_increment	
user_name	varchar(255)	YES	UNI			
load_time	timestamp	NO		CURRENT_TIMESTAMP		
last_evaluation_time	timestamp	NO		0000-00-00 00:00:00		

Relations



outagetype

Description

This table contains list outage types and descriptions

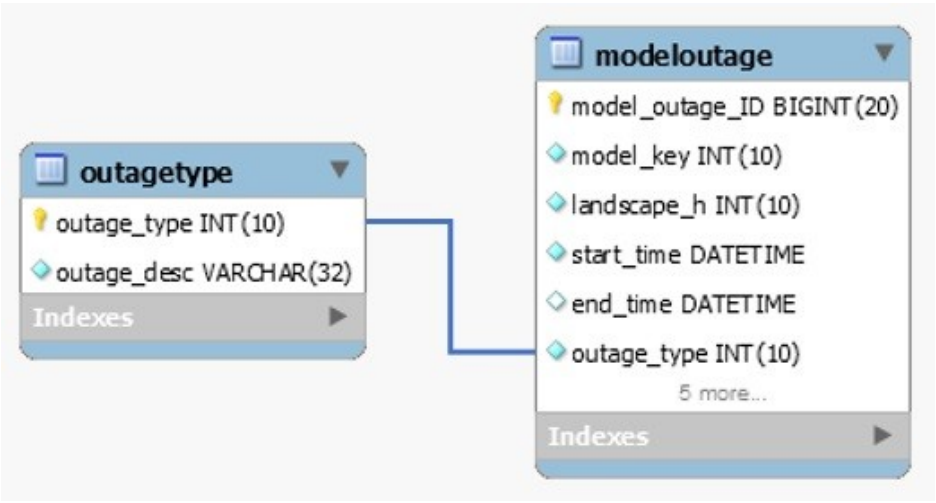
This table is filled in at the time of table creation. Outage types are pre-defined before any outages occur. Table records include:

outage_type	outage_desc
0	Initial
1	Unplanned
2	Planned
3	Exempt

Columns

Field	Type	Null	Key	Default	Extra	Comment
outage_type	int(10) unsigned	NO	PRI			
outage_desc	varchar(32)	NO				

Relations



pcause

Description

This table provides a mapping of cause codes to their titles. It is populated as each new cause is encountered by the alarm handler.

Columns

Field	Type	Null	Key	Default	Extra	Comment
cause_id	int(10) unsigned	NO	PRI			
title	varchar(100)	YES				

performance

Description

This table contains MySQL configuration settings for SRM DB, mostly static data

Columns

Field	Type	Null	Key	Default	Extra	Comment
id	int(10) unsigned	NO	PRI		auto_increment	
runcode	char(255)	YES				
runtime	datetime	YES				
memused	int(10) unsigned	NO				
memfree	int(10) unsigned	NO				
Aborted_clients	int(10) unsigned	NO		0		
Aborted_connects	int(10) unsigned	NO		0		
Binlog_cache_disk_use	int(10) unsigned	NO		0		
Binlog_cache_use	int(10) unsigned	NO		0		
Bytes_received	int(10) unsigned	NO		0		
Bytes_sent	int(10) unsigned	NO		0		
Com_admin_commands	int(10) unsigned	NO		0		
Com_alter_db	int(10) unsigned	NO		0		
Com_alter_table	int(10) unsigned	NO		0		
Com_analyze	int(10) unsigned	NO		0		
Com_backup_table	int(10) unsigned	NO		0		
Com_begin	int(10) unsigned	NO		0		
Com_change_db	int(10) unsigned	NO		0		
Com_change_master	int(10) unsigned	NO		0		
Com_check	int(10) unsigned	NO		0		
Com_checksum	int(10) unsigned	NO		0		
Com_commit	int(10) unsigned	NO		0		
Com_create_db	int(10) unsigned	NO		0		
Com_create_function	int(10) unsigned	NO		0		
Com_create_index	int(10) unsigned	NO		0		

Com_create_table	int(10) unsigned	NO		0		
Com_dealloc_sql	int(10) unsigned	NO		0		
Com_delete	int(10) unsigned	NO		0		
Com_delete_multi	int(10) unsigned	NO		0		
Com_do	int(10) unsigned	NO		0		
Com_drop_db	int(10) unsigned	NO		0		
Com_drop_function	int(10) unsigned	NO		0		
Com_drop_index	int(10) unsigned	NO		0		
Com_drop_table	int(10) unsigned	NO		0		
Com_drop_user	int(10) unsigned	NO		0		
Com_execute_sql	int(10) unsigned	NO		0		
Com_flush	int(10) unsigned	NO		0		
Com_grant	int(10) unsigned	NO		0		
Com_ha_close	int(10) unsigned	NO		0		
Com_ha_open	int(10) unsigned	NO		0		
Com_ha_read	int(10) unsigned	NO		0		
Com_help	int(10) unsigned	NO		0		
Com_insert	int(10) unsigned	NO		0		
Com_insert_select	int(10) unsigned	NO		0		
Com_kill	int(10) unsigned	NO		0		
Com_load	int(10) unsigned	NO		0		
Com_load_master_data	int(10) unsigned	NO		0		
Com_load_master_table	int(10) unsigned	NO		0		
Com_lock_tables	int(10) unsigned	NO		0		
Com_optimize	int(10) unsigned	NO		0		
Com_preload_keys	int(10) unsigned	NO		0		
Com_prepare_sql	int(10) unsigned	NO		0		
Com_purge	int(10) unsigned	NO		0		
Com_purge_before_date	int(10) unsigned	NO		0		
Com_rename_table	int(10) unsigned	NO		0		
Com_repair	int(10) unsigned	NO		0		
Com_replace	int(10) unsigned	NO		0		
Com_replace_select	int(10) unsigned	NO		0		
Com_reset	int(10) unsigned	NO		0		
Com_restore_table	int(10) unsigned	NO		0		
Com_revoke	int(10) unsigned	NO		0		
Com_revoke_all	int(10) unsigned	NO		0		
Com_rollback	int(10) unsigned	NO		0		
Com_savepoint	int(10) unsigned	NO		0		
Com_select	int(10) unsigned	NO		0		
Com_set_option	int(10) unsigned	NO		0		

Com_show_binlog_events	int(10) unsigned	NO		0		
Com_show_binlogs	int(10) unsigned	NO		0		
Com_show_charsets	int(10) unsigned	NO		0		
Com_show_collations	int(10) unsigned	NO		0		
Com_show_column_types	int(10) unsigned	NO		0		
Com_show_create_db	int(10) unsigned	NO		0		
Com_show_create_table	int(10) unsigned	NO		0		
Com_show_databases	int(10) unsigned	NO		0		
Com_show_errors	int(10) unsigned	NO		0		
Com_show_fields	int(10) unsigned	NO		0		
Com_show_grants	int(10) unsigned	NO		0		
Com_show_innodb_status	int(10) unsigned	NO		0		
Com_show_keys	int(10) unsigned	NO		0		
Com_show_logs	int(10) unsigned	NO		0		
Com_show_master_status	int(10) unsigned	NO		0		
Com_show_new_master	int(10) unsigned	NO		0		
Com_show_open_tables	int(10) unsigned	NO		0		
Com_show_privileges	int(10) unsigned	NO		0		
Com_show_processlist	int(10) unsigned	NO		0		
Com_show_slave_hosts	int(10) unsigned	NO		0		
Com_show_slave_status	int(10) unsigned	NO		0		
Com_show_status	int(10) unsigned	NO		0		
Com_show_storage_engines	int(10) unsigned	NO		0		
Com_show_tables	int(10) unsigned	NO		0		
Com_show_variables	int(10) unsigned	NO		0		
Com_show_warnings	int(10) unsigned	NO		0		
Com_slave_start	int(10) unsigned	NO		0		
Com_slave_stop	int(10) unsigned	NO		0		
Com_truncate	int(10) unsigned	NO		0		
Com_unlock_tables	int(10) unsigned	NO		0		
Com_update	int(10) unsigned	NO		0		
Com_update_multi	int(10) unsigned	NO		0		
Connections	int(10) unsigned	NO		0		
Created_tmp_disk_tables	int(10) unsigned	NO		0		
Created_tmp_files	int(10) unsigned	NO		0		
Created_tmp_tables	int(10) unsigned	NO		0		
Delayed_errors	int(10) unsigned	NO		0		
Delayed_insert_threads	int(10) unsigned	NO		0		
Delayed_writes	int(10) unsigned	NO		0		
Flush_commands	int(10) unsigned	NO		0		
Handler_commit	int(10) unsigned	NO		0		

Handler_delete	int(10) unsigned	NO		0		
Handler_discover	int(10) unsigned	NO		0		
Handler_read_first	int(10) unsigned	NO		0		
Handler_read_key	int(10) unsigned	NO		0		
Handler_read_next	int(10) unsigned	NO		0		
Handler_read_prev	int(10) unsigned	NO		0		
Handler_read_rnd	int(10) unsigned	NO		0		
Handler_read_rnd_next	int(10) unsigned	NO		0		
Handler_rollback	int(10) unsigned	NO		0		
Handler_update	int(10) unsigned	NO		0		
Handler_write	int(10) unsigned	NO		0		
Key_blocks_not_flushed	int(10) unsigned	NO		0		
Key_blocks_unused	int(10) unsigned	NO		0		
Key_blocks_used	int(10) unsigned	NO		0		
Key_read_requests	int(10) unsigned	NO		0		
Key_reads	int(10) unsigned	NO		0		
Key_write_requests	int(10) unsigned	NO		0		
Key_writes	int(10) unsigned	NO		0		
Max_used_connections	int(10) unsigned	NO		0		
Not_flushed_delayed_rows	int(10) unsigned	NO		0		
Open_files	int(10) unsigned	NO		0		
Open_streams	int(10) unsigned	NO		0		
Open_tables	int(10) unsigned	NO		0		
Opened_tables	int(10) unsigned	NO		0		
Qcache_free_blocks	int(10) unsigned	NO		0		
Qcache_free_memory	int(10) unsigned	NO		0		
Qcache_hits	int(10) unsigned	NO		0		
Qcache_inserts	int(10) unsigned	NO		0		
Qcache_lowmem_prunes	int(10) unsigned	NO		0		
Qcache_not_cached	int(10) unsigned	NO		0		
Qcache_queries_in_cache	int(10) unsigned	NO		0		
Qcache_total_blocks	int(10) unsigned	NO		0		
Questions	int(10) unsigned	NO		0		
Select_full_join	int(10) unsigned	NO		0		
Select_full_range_join	int(10) unsigned	NO		0		
Select_range	int(10) unsigned	NO		0		
Select_range_check	int(10) unsigned	NO		0		
Select_scan	int(10) unsigned	NO		0		
Slave_open_temp_tables	int(10) unsigned	NO		0		
Slave_retried_transactions	int(10) unsigned	NO		0		
Slow_launch_threads	int(10) unsigned	NO		0		

Slow_queries	int(10) unsigned	NO		0		
Sort_merge_passes	int(10) unsigned	NO		0		
Sort_range	int(10) unsigned	NO		0		
Sort_rows	int(10) unsigned	NO		0		
Sort_scan	int(10) unsigned	NO		0		
Table_locks_immediate	int(10) unsigned	NO		0		
Table_locks_waited	int(10) unsigned	NO		0		
Threads_cached	int(10) unsigned	NO		0		
Threads_connected	int(10) unsigned	NO		0		
Threads_created	int(10) unsigned	NO		0		
Threads_running	int(10) unsigned	NO		0		
Uptime	int(10) unsigned	NO		0		
Com_show_ndb_status	int(10) unsigned	NO		0		
Com_show_triggers	int(10) unsigned	NO		0		
Com_stmt_close	int(10) unsigned	NO		0		
Com_stmt_execute	int(10) unsigned	NO		0		
Com_stmt_fetch	int(10) unsigned	NO		0		
Com_stmt_prepare	int(10) unsigned	NO		0		
Com_stmt_reset	int(10) unsigned	NO		0		
Com_stmt_send_long_data	int(10) unsigned	NO		0		
Com_xa_commit	int(10) unsigned	NO		0		
Com_xa_end	int(10) unsigned	NO		0		
Com_xa_prepare	int(10) unsigned	NO		0		
Com_xa_recover	int(10) unsigned	NO		0		
Com_xa_rollback	int(10) unsigned	NO		0		
Com_xa_start	int(10) unsigned	NO		0		
Handler_prepare	int(10) unsigned	NO		0		
Handler_savepoint	int(10) unsigned	NO		0		
Handler_savepoint_rollback	int(10) unsigned	NO		0		
Innodb_buffer_pool_pages_data	int(10) unsigned	NO		0		
Innodb_buffer_pool_pages_dirty	int(10) unsigned	NO		0		
Innodb_buffer_pool_pages_flushed	int(10) unsigned	NO		0		
Innodb_buffer_pool_pages_free	int(10) unsigned	NO		0		
Innodb_buffer_pool_pages_latched	int(10) unsigned	NO		0		
Innodb_buffer_pool_pages_misc	int(10) unsigned	NO		0		
Innodb_buffer_pool_pages_total	int(10) unsigned	NO		0		
Innodb_buffer_pool_read_ahead_rnd	int(10) unsigned	NO		0		
Innodb_buffer_pool_read_ahead_seq	int(10) unsigned	NO		0		
Innodb_buffer_pool_read_requests	int(10) unsigned	NO		0		
Innodb_buffer_pool_reads	int(10) unsigned	NO		0		
Innodb_buffer_pool_wait_free	int(10) unsigned	NO		0		

Innodb_buffer_pool_write_requests	int(10) unsigned	NO		0		
Innodb_data_fsyncs	int(10) unsigned	NO		0		
Innodb_data_pending_fsyncs	int(10) unsigned	NO		0		
Innodb_data_pending_reads	int(10) unsigned	NO		0		
Innodb_data_pending_writes	int(10) unsigned	NO		0		
Innodb_data_read	int(10) unsigned	NO		0		
Innodb_data_reads	int(10) unsigned	NO		0		
Innodb_data_writes	int(10) unsigned	NO		0		
Innodb_data_written	int(10) unsigned	NO		0		
Innodb_dblwr_pages_written	int(10) unsigned	NO		0		
Innodb_dblwr_writes	int(10) unsigned	NO		0		
Innodb_log_waits	int(10) unsigned	NO		0		
Innodb_log_write_requests	int(10) unsigned	NO		0		
Innodb_log_writes	int(10) unsigned	NO		0		
Innodb_os_log_fsyncs	int(10) unsigned	NO		0		
Innodb_os_log_pending_fsyncs	int(10) unsigned	NO		0		
Innodb_os_log_pending_writes	int(10) unsigned	NO		0		
Innodb_os_log_written	int(10) unsigned	NO		0		
Innodb_page_size	int(10) unsigned	NO		0		
Innodb_pages_created	int(10) unsigned	NO		0		
Innodb_pages_read	int(10) unsigned	NO		0		
Innodb_pages_written	int(10) unsigned	NO		0		
Innodb_row_lock_current_waits	int(10) unsigned	NO		0		
Innodb_row_lock_time	int(10) unsigned	NO		0		
Innodb_row_lock_time_avg	int(10) unsigned	NO		0		
Innodb_row_lock_time_max	int(10) unsigned	NO		0		
Innodb_row_lock_waits	int(10) unsigned	NO		0		
Innodb_rows_deleted	int(10) unsigned	NO		0		
Innodb_rows_inserted	int(10) unsigned	NO		0		
Innodb_rows_read	int(10) unsigned	NO		0		
Innodb_rows_updated	int(10) unsigned	NO		0		
Last_query_cost	int(10) unsigned	NO		0		
Ssl_accept_renegotiates	int(10) unsigned	NO		0		
Ssl_accepts	int(10) unsigned	NO		0		
Ssl_callback_cache_hits	int(10) unsigned	NO		0		
Ssl_client_connects	int(10) unsigned	NO		0		
Ssl_connect_renegotiates	int(10) unsigned	NO		0		
Ssl_ctx_verify_depth	int(10) unsigned	NO		0		
Ssl_ctx_verify_mode	int(10) unsigned	NO		0		
Ssl_default_timeout	int(10) unsigned	NO		0		
Ssl_finished_accepts	int(10) unsigned	NO		0		

Ssl_finished_connects	int(10) unsigned	NO		0		
Ssl_session_cache_hits	int(10) unsigned	NO		0		
Ssl_session_cache_misses	int(10) unsigned	NO		0		
Ssl_session_cache_overflows	int(10) unsigned	NO		0		
Ssl_session_cache_size	int(10) unsigned	NO		0		
Ssl_session_cache_timeouts	int(10) unsigned	NO		0		
Ssl_sessions_reused	int(10) unsigned	NO		0		
Ssl_used_session_cache_entries	int(10) unsigned	NO		0		
Ssl_verify_depth	int(10) unsigned	NO		0		
Ssl_verify_mode	int(10) unsigned	NO		0		
Tc_log_max_pages_used	int(10) unsigned	NO		0		
Tc_log_page_size	int(10) unsigned	NO		0		
Tc_log_page_waits	int(10) unsigned	NO		0		
Com_call_procedure	int(10) unsigned	YES				
Com_create_user	int(10) unsigned	YES				
prepared_stmt_count	int(10) unsigned	YES				
Com_assign_to_keycache	int(10) unsigned	NO		0		
Com_alter_db_upgrade	int(10) unsigned	NO		0		
Com_alter_event	int(10) unsigned	NO		0		
Com_alter_function	int(10) unsigned	NO		0		
Com_alter_procedure	int(10) unsigned	NO		0		
Com_alter_server	int(10) unsigned	NO		0		
Com_alter_tablespace	int(10) unsigned	NO		0		
Com_binlog	int(10) unsigned	NO		0		
Com_create_event	int(10) unsigned	NO		0		
Com_create_procedure	int(10) unsigned	NO		0		
Com_create_server	int(10) unsigned	NO		0		
Com_create_trigger	int(10) unsigned	NO		0		
Com_create_udf	int(10) unsigned	NO		0		
Com_create_view	int(10) unsigned	NO		0		
Com_drop_event	int(10) unsigned	NO		0		
Com_drop_procedure	int(10) unsigned	NO		0		
Com_drop_server	int(10) unsigned	NO		0		
Com_drop_trigger	int(10) unsigned	NO		0		
Com_drop_view	int(10) unsigned	NO		0		
Com_empty_query	int(10) unsigned	NO		0		
Com_install_plugin	int(10) unsigned	NO		0		
Com_release_savepoint	int(10) unsigned	NO		0		
Com_rename_user	int(10) unsigned	NO		0		
Com_rollback_to_savepoint	int(10) unsigned	NO		0		
Com_show_authors	int(10) unsigned	NO		0		

Com_show_contributors	int(10) unsigned	NO		0		
Com_show_create_event	int(10) unsigned	NO		0		
Com_show_create_func	int(10) unsigned	NO		0		
Com_show_create_proc	int(10) unsigned	NO		0		
Com_show_create_trigger	int(10) unsigned	NO		0		
Com_show_engine_logs	int(10) unsigned	NO		0		
Com_show_engine_mutex	int(10) unsigned	NO		0		
Com_show_engine_status	int(10) unsigned	NO		0		
Com_show_events	int(10) unsigned	NO		0		
Com_show_function_status	int(10) unsigned	NO		0		
Com_show_plugins	int(10) unsigned	NO		0		
Com_show_procedure_status	int(10) unsigned	NO		0		
Com_show_profile	int(10) unsigned	NO		0		
Com_show_profiles	int(10) unsigned	NO		0		
Com_show_table_status	int(10) unsigned	NO		0		
Com_stmt_reprepare	int(10) unsigned	NO		0		
Com_uninstall_plugin	int(10) unsigned	NO		0		
Open_table_definitions	int(10) unsigned	NO		0		
Opened_files	int(10) unsigned	NO		0		
Opened_table_definitions	int(10) unsigned	NO		0		
Queries	int(10) unsigned	NO		0		
Uptime_since_flush_status	int(10) unsigned	NO		0		

portduplex

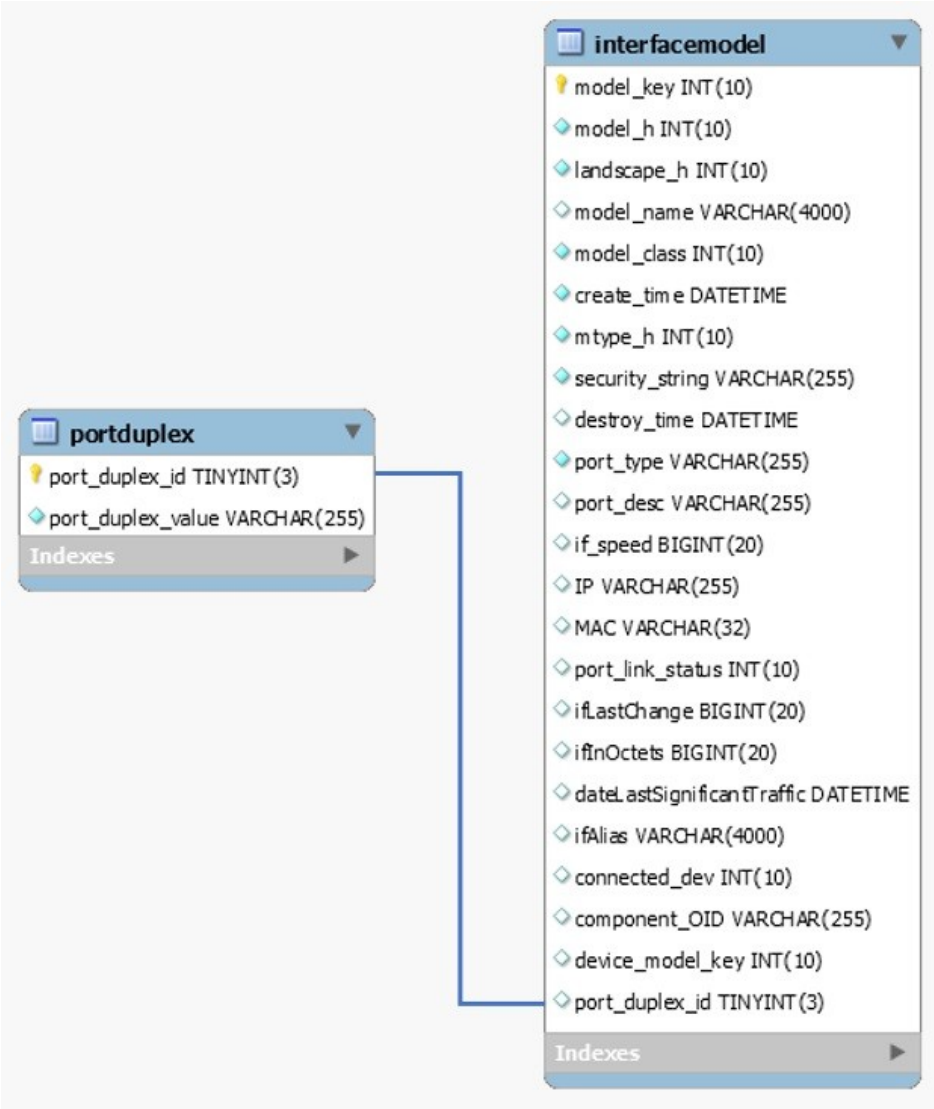
Description

This table contains port duplex IDs and corresponding values

Columns

Field	Type	Null	Key	Default	Extra	Comment
port_duplex_id	tinyint(3) unsigned	NO	PRI			
port_duplex_value	varchar(255)	NO				

Relations



registry

Description

This table provides a storage area for SRM to maintain different properties and attributes of the SRM application. The table stores generic mappings using key/value pairs. Registry entries can have one of the following types:

type id	type name
0	Boolean
1	String
2	Hidden
3	List
4	List Entry

Columns

Field	Type	Null	Key	Default	Extra	Comment
reg_user	varchar(255)	NO	PRI			
OneClickServerEntry	varchar(255)	NO				
SRMPollPeriod	tinyint(4)	NO				
SRMPollStartTime	tinyint(4)	NO				
SRMPollEndTime	tinyint(4)	NO				
IFIdleThreshold	int(11)	NO		0		
ServersList	text	NO				
CrystalInstallRoot	varchar(255)	NO				
CrystalCommonFiles	varchar(255)	NO				
CrystalHome	varchar(255)	NO				
CrystalPassword	varbinary(255)	NO				
ReportSuppressedAlarms	enum('false','true')	NO		false		
BOPassword	varbinary(255)	NO				
BOCommonFiles	varchar(255)	NO				
BOHome	varchar(255)	NO				
BOInstallRoot	varchar(255)	NO				
ConvertedReportsToBOXI	enum('false','true')	NO				
ArchivalRetentionDays	int(11) unsigned	YES		90		
event_archival_retention_days	int(11) unsigned	NO		90		
DataRetentionPolicy	enum('all data','archive','purge')	NO				
customlogopath	text	YES				
isReportingReady	enum('false','true')	NO		false		
handler_batch_size	int(10) unsigned	NO		1000		
event_poller_processing_interval	int(10) unsigned	NO		60		
MonitorSRM	enum('false','true')	YES		true		
SRM_Model	int(11) unsigned	YES				
BOUser	varchar(255)	NO				
BOHost	varchar(255)	NO				
BOPort	mediumint(9)	NO				
BOAuthType	varchar(255)	NO				
BOTomcatPort	mediumint(9)	NO				
CrystalReportsUser	varchar(255)	NO				
CrystalReportsPassword	varbinary(255)	NO				
CrystalReportsHost	varchar(255)	NO				
DBHost	varchar(255)	NO				
BOInfoView	varchar(255)	NO				
BOCmc	varchar(255)	NO				
BOInfoViewCredentials	varchar(255)	NO				

BOInfoViewAuthType	varchar(255)	NO				
UniverseUser	varchar(255)	NO				
UniversePassword	varbinary(255)	NO				
is_security_enabled	enum('false','true')	YES		false		
DefaultBOXIUserPassword	varbinary(255)	NO				
isPerformanceMonitorEnabled	enum('false','true')	NO		false		
install_version	varchar(128)	NO		9.4.2.1.1		
PollTaskMaxTardy	smallint(5) unsigned	YES				
BOSharedSecret	varbinary(255)	YES				

schemaversion

Description

This table stores data related to SRM schema versioning, change log etc. for one-time changes. Table 'schemaversion_recurring' table stores recurring changes.

Internal table, do not edit

Columns

Field	Type	Null	Key	Default	Extra	Comment
change_id	int(10) unsigned	NO	PRI		auto_increment	
session_id	int(10) unsigned	YES				
name	varchar(255)	NO	UNI			
category	varchar(255)	NO				
schema_comments	varchar(255)	NO				
major	tinyint(4)	NO				
minor	tinyint(4)	NO				
service_pack	tinyint(4)	NO				
start_time	datetime	YES				
end_time	datetime	YES				
duration_secs	int(11)	YES				
state	enum('A','N','F')	YES		A		
state_details	text	YES				

schemaversion_recurring

Description

This table stores data related to SRM schema versioning, change log etc. of recurring changes.

Internal table, do not edit

Columns

Field	Type	Null	Key	Default	Extra	Comment
change_id	int(10) unsigned	NO	PRI		auto_increment	

session_id	int(10) unsigned	YES				
name	varchar(255)	NO	MUL			
category	varchar(255)	NO				
schema_comments	varchar(255)	NO				
major	tinyint(4)	NO				
minor	tinyint(4)	NO				
service_pack	tinyint(4)	NO				
start_time	datetime	YES				
end_time	datetime	YES				
duration_secs	int(11)	YES				
State	enum('A','N','F')	YES		A		
state_details	text	YES				

schemaversion_session

Description

Table stores data related to SRM schema versioning session information. Internal table

Columns

Field	Type	Null	Key	Default	Extra	Comment
session_id	int(10) unsigned	NO	PRI		auto_increment	
session_name	varchar(255)	NO				
start_time	datetime	NO				
end_time	datetime	YES				
duration_secs	int(11)	YES				
state	enum('A','F')	YES				

security_string

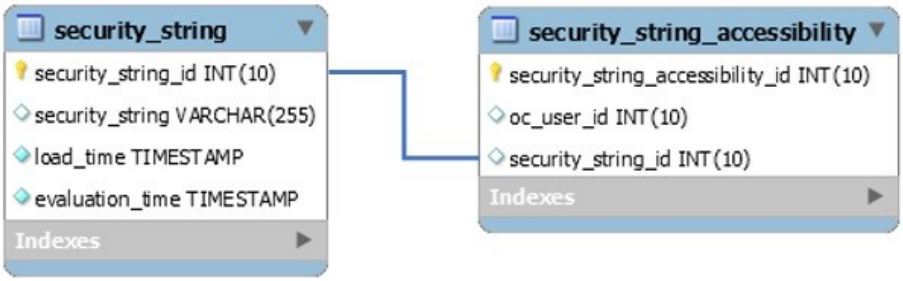
Description

This table stores list of unique security string IDs and security strings for models in reporting DB.

Columns

Field	Type	Null	Key	Default	Extra	Comment
security_string_id	int(10) unsigned	NO	PRI		auto_increment	
security_string	varchar(255)	YES	UNI			
load_time	timestamp	NO		CURRENT_TIMESTAMP		
evaluation_time	timestamp	NO		0000-00-00 00:00:00		

Relations



security_string_accessibility

Description

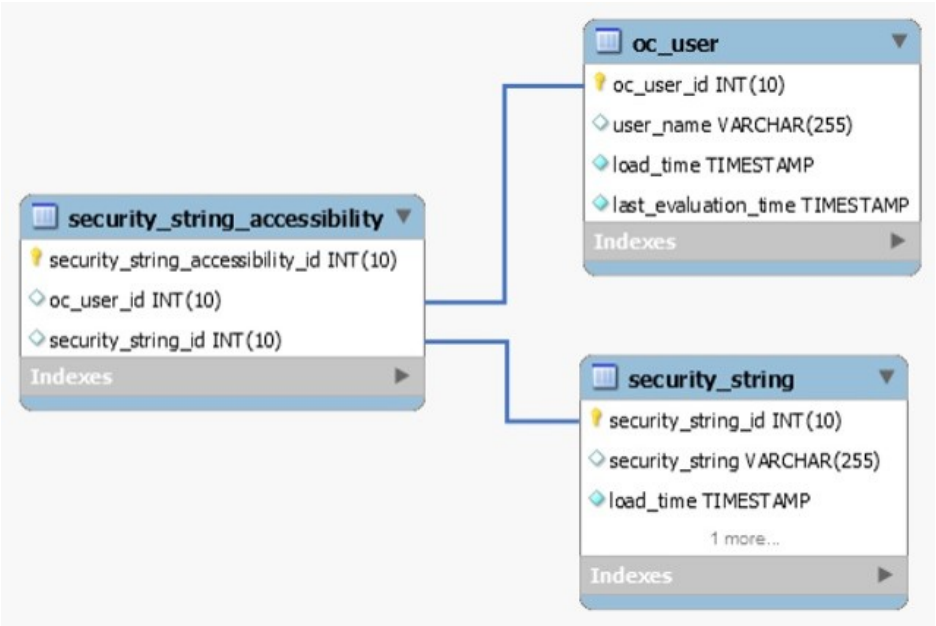
This table stores security string accessibility information. Maps Oneclick user ID to security string ID.

Internal table, do not edit

Columns

Field	Type	Null	Key	Default	Extra	Comment
security_string_accessibility_id	int(10) unsigned	NO	PRI		auto_increment	
oc_user_id	int(10) unsigned	YES	MUL			
security_string_id	int(10) unsigned	YES				

Relations



sm_attributes

Description

Service Manager attributes table - static data

Columns

Column name	Type	Null	Key	Default	Comment
attrID	int(10) unsigned	NO	PRI		Attribute ID of service manager model class
attrName	varchar(255)	NO			Attribute name of service manager model class

Relations

sm_customermhs

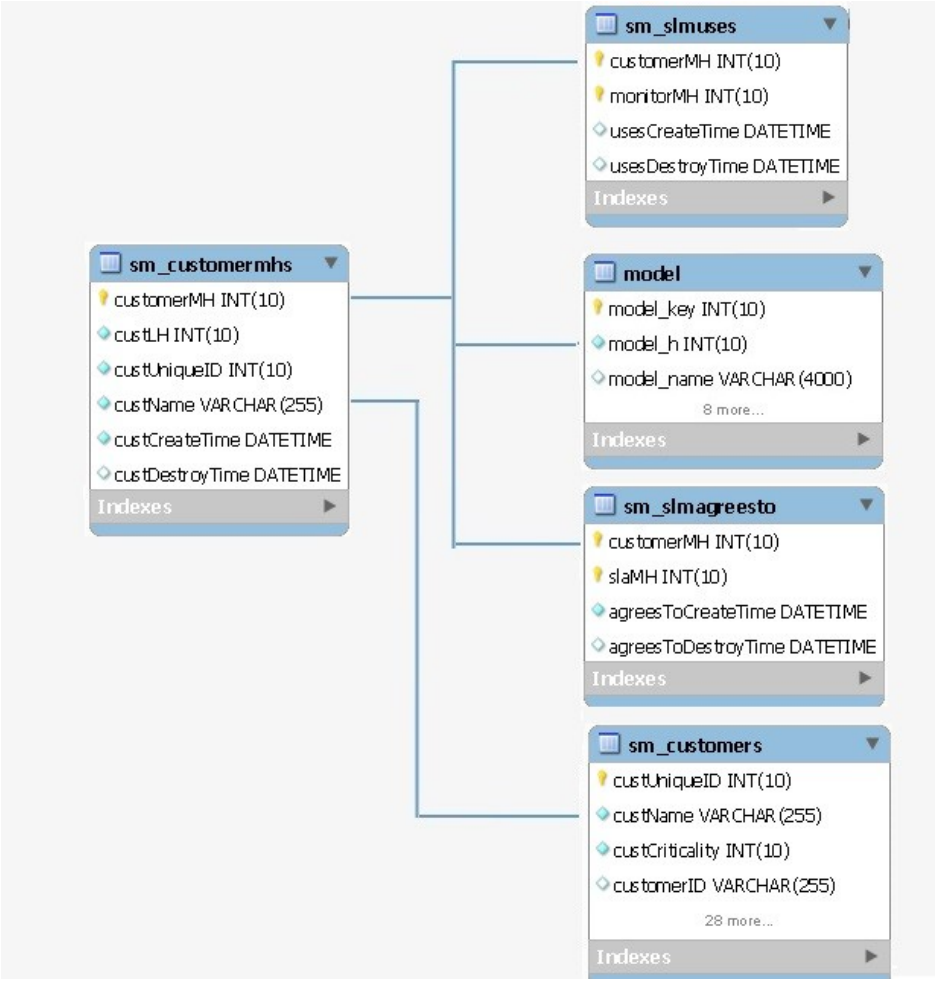
Description

This table stores service manager - customer model handles

Columns

Field	Type	Null	Key	Default	Comment
customerMH	int(10) unsigned	NO	PRI		Customer model handle
custLH	int(10) unsigned	NO			
custUniqueID	int(10) unsigned	NO	MUL		UniqueID of the customer
custName	varchar(255)	NO			customer name
custCreateTime	datetime	NO			customer create time
custDestroyTime	datetime	YES			customer destroy time

Relations



sm_customers

Description

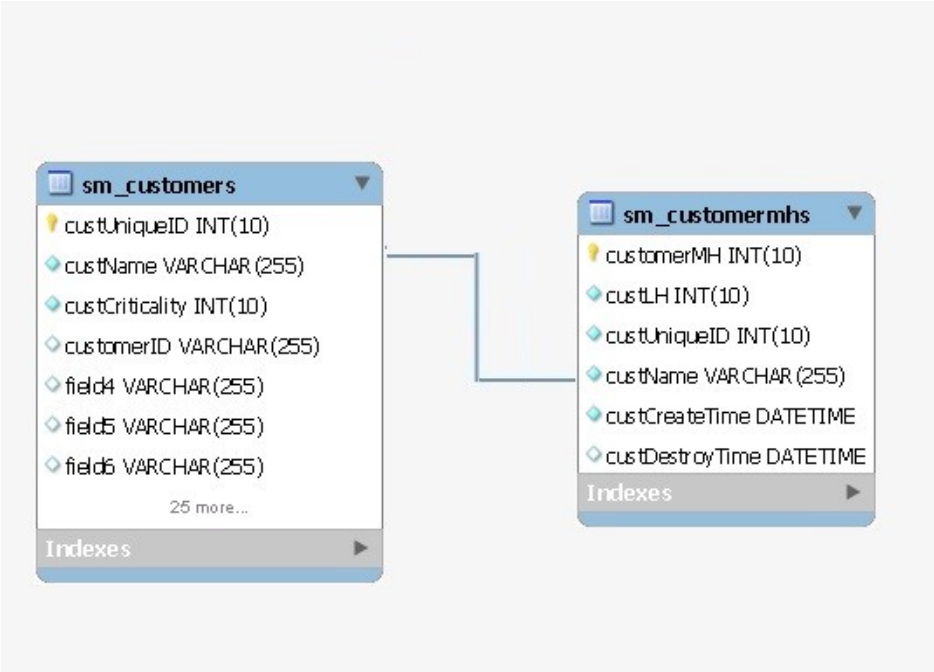
This table stores Service manager – customer details

Columns

Field	Type	Null	Key	Default	Extra	Comment
custUniqueID	int(10) unsigned	NO	PRI		auto_increment	Customer unique ID
custName	varchar(255)	NO	UNI			Customer Name
custCriticality	int(10) unsigned	NO				Customer criticality
customerID	varchar(255)	YES				Customer ID
field4	varchar(255)	YES				
field5	varchar(255)	YES				
field6	varchar(255)	YES				
field7	varchar(255)	YES				

primContName	varchar(255)	YES				Primary contact name
primContTitle	varchar(255)	YES				Primary contact title
primContLocation	varchar(255)	YES				primary contact location
primEmail	varchar(255)	YES				Primary Email
primPhone	varchar(255)	YES				Primary Phone
primMobile	varchar(255)	YES				Primary Mobile
primPager	varchar(255)	YES				Primary Pager
primFax	varchar(255)	YES				Primary FAX
primUserDef1	varchar(255)	YES				
primUserDef2	varchar(255)	YES				
primUserDef3	varchar(255)	YES				
primUserDef4	varchar(255)	YES				
secContName	varchar(255)	YES				Secondary Contact Name
secContTitle	varchar(255)	YES				
secContLocation	varchar(255)	YES				
secEmail	varchar(255)	YES				
secPhone	varchar(255)	YES				
secMobile	varchar(255)	YES				
secPager	varchar(255)	YES				
secFax	varchar(255)	YES				
secUserDef1	varchar(255)	YES				
secUserDef2	varchar(255)	YES				
secUserDef3	varchar(255)	YES				
secUserDef4	varchar(255)	YES				

Relations



sm_guaranteeoutages

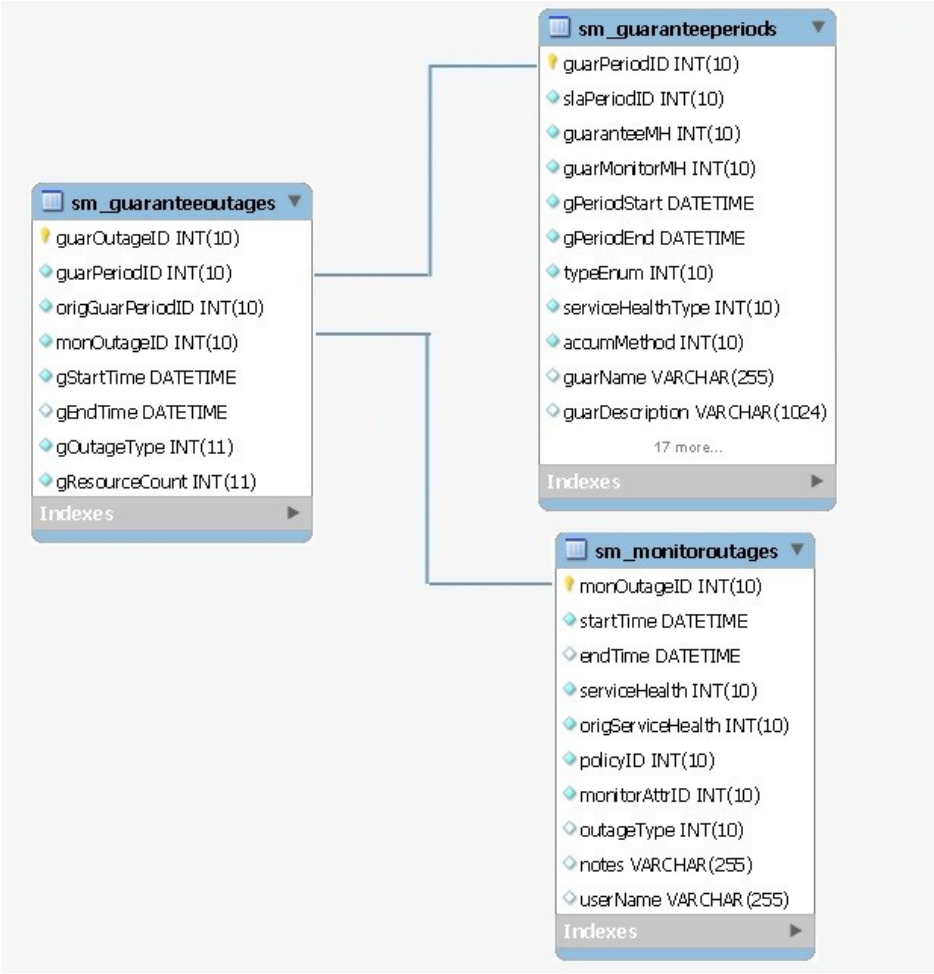
Description

This table stores Service manager - guaranteed outages

Columns

Field	Type	Null	Key	Default	Extra	Comment
guarOutageID	int(10) unsigned	NO	PRI		auto_increment	Guarantee outage ID
guarPeriodID	int(10) unsigned	NO				Guarantee period ID
origGuarPeriodID	int(10) unsigned	NO				Original guarantee period ID
monOutageID	int(10) unsigned	NO				Monitoring Outage ID
gStartTime	datetime	NO				Outage start time
gEndTime	datetime	YES				Outable end time
gOutageType	int(11)	NO				Type of Outage
gResourceCount	int(11)	NO				Resource count affected by outage

Relations



sm_guaranteeperiods

Description

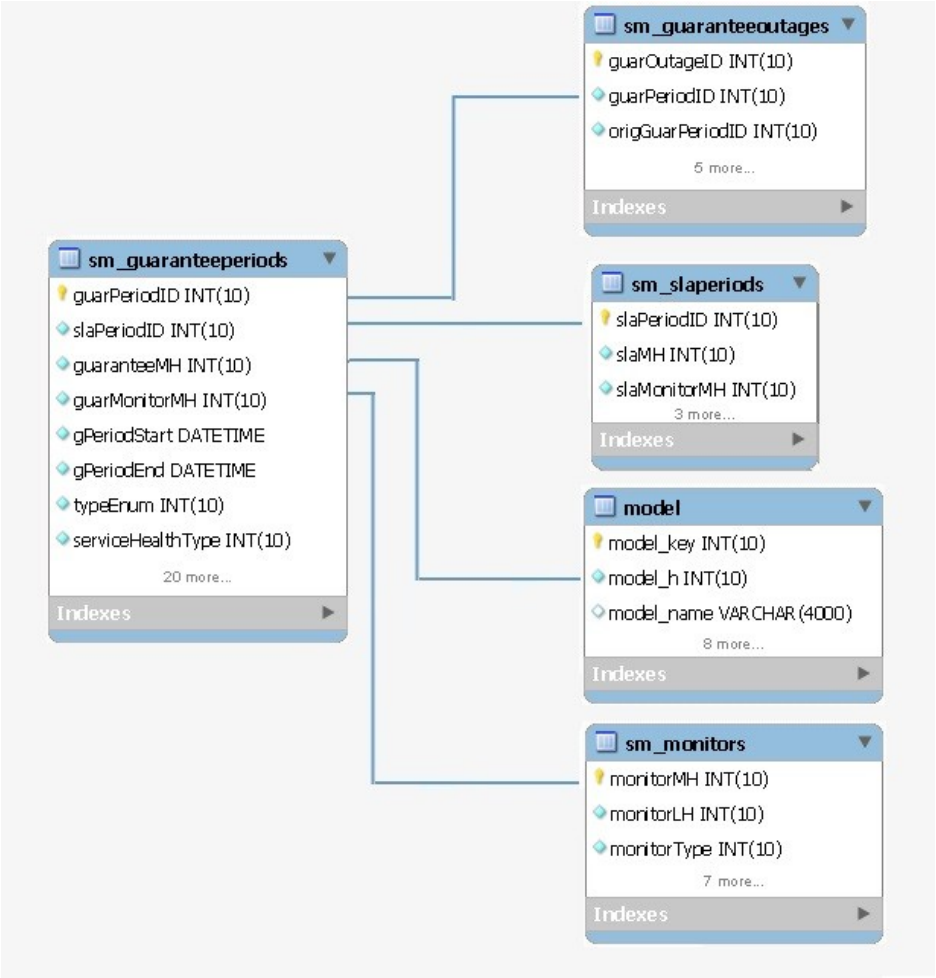
This table stores Service manager guarantee periods

Columns

Field	Type	Null	Key	Default	Extra	Comment
guarPeriodID	int(10) unsigned	NO	PRI		auto_increment	Guarantee period ID
slaPeriodID	int(10) unsigned	NO	MUL			SLA period ID
guaranteeMH	int(10) unsigned	NO				Guarantee model handle
guarMonitorMH	int(10) unsigned	NO				Guarantee monitor model handle
gPeriodStart	datetime	NO				Guarantee period start time
gPeriodEnd	datetime	NO				Guarantee period end time
typeEnum	int(10) unsigned	NO				Type of guarantee period

serviceHealthType	int(10) unsigned	NO				Type of service health
accumMethod	int(10) unsigned	NO				Guarantee period accumulation method
guarName	varchar(255)	YES				Guarantee name
guarDescription	varchar(1024)	YES				
outageTime	int(10) unsigned	NO				Outage time
outageCount	int(10) unsigned	NO				Outage count
activeTime	int(10) unsigned	NO				Time when the outage is active
majorThreshold	int(10) unsigned	NO				
criticalThreshold	int(10) unsigned	NO				
majorThresholdPercent	double	YES				
criticalThresholdPercent	double	YES				
guarStatus	int(10) unsigned	NO				
motValue	int(10) unsigned	NO				
motThreshold	int(10) unsigned	NO				
motStatus	int(10) unsigned	NO				
mttrValue	int(10) unsigned	NO				
mttrThreshold	int(10) unsigned	NO				
mttrStatus	int(10) unsigned	NO				
mtbfValue	int(10) unsigned	NO				
mtbfThreshold	int(10) unsigned	NO				
mtbfStatus	int(10) unsigned	NO				

Relations



sm_monitormaps

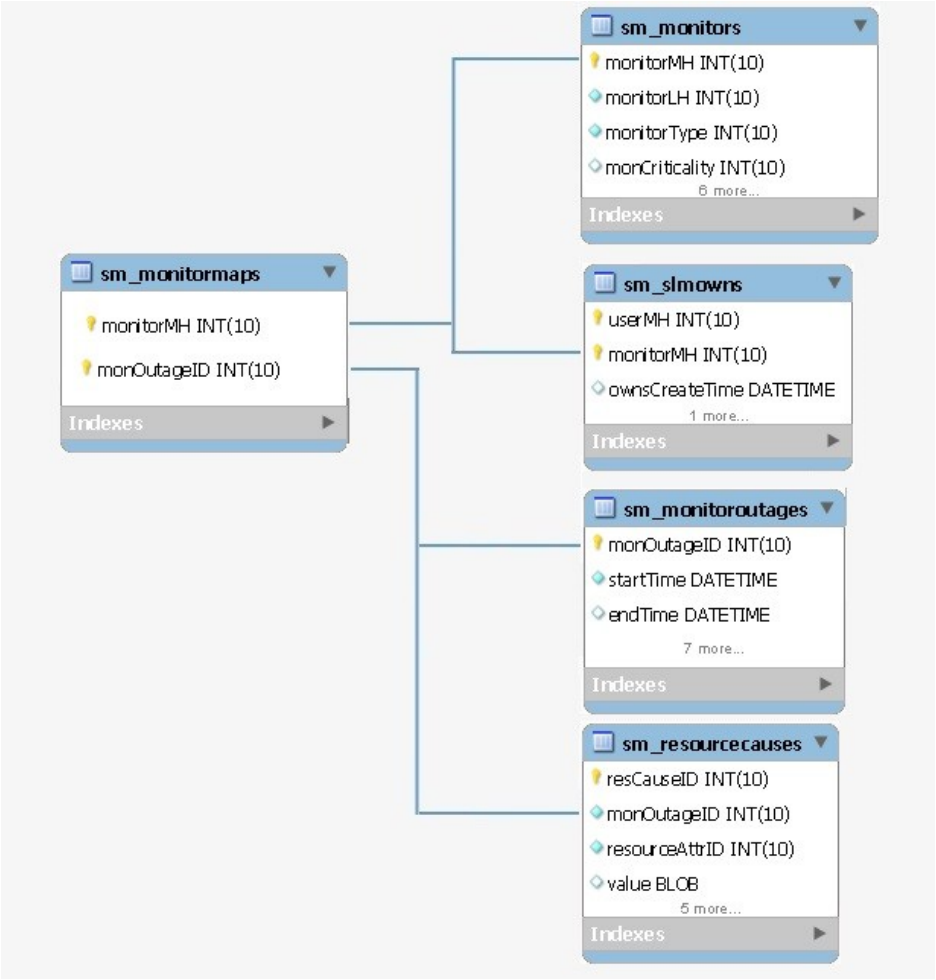
Description

This table stores the mapping of monitor model handle to outage ID

Columns

Field	Type	Null	Key	Default	Extra	Comment
monitorMH	int(10) unsigned	NO	PRI			Model handle of the monitor
monOutageID	int(10) unsigned	NO	PRI			Monitoring outage ID

Relations



sm_monitoroutages

Description

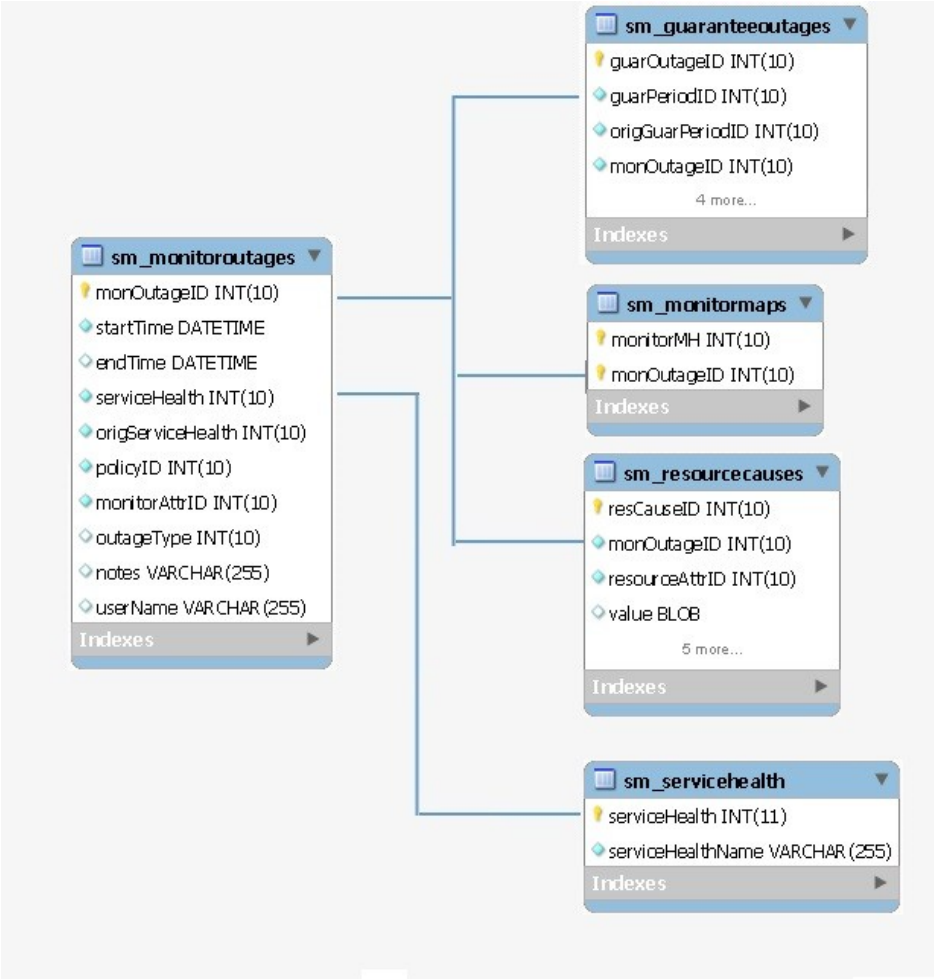
This table stores Service manager - Outages

Columns

Field	Type	Null	Key	Default	Extra	Comment
monOutageID	int(10) unsigned	NO	PRI		auto_increment	Monitoring outage ID
startTime	datetime	NO				Outage starting time
endTime	datetime	YES				Outage ending time
serviceHealth	int(10) unsigned	NO	MUL			Service health
origServiceHealth	int(10) unsigned	NO				Default service health
policyID	int(10) unsigned	NO				policy ID

monitorAttrID	int(10) unsigned	NO				Monitor attribute ID
outageType	int(10) unsigned	YES				Type of outage
notes	varchar(255)	YES				'notes' for the outage
userName	varchar(255)	YES				User name

Relations



sm_monitors

Description

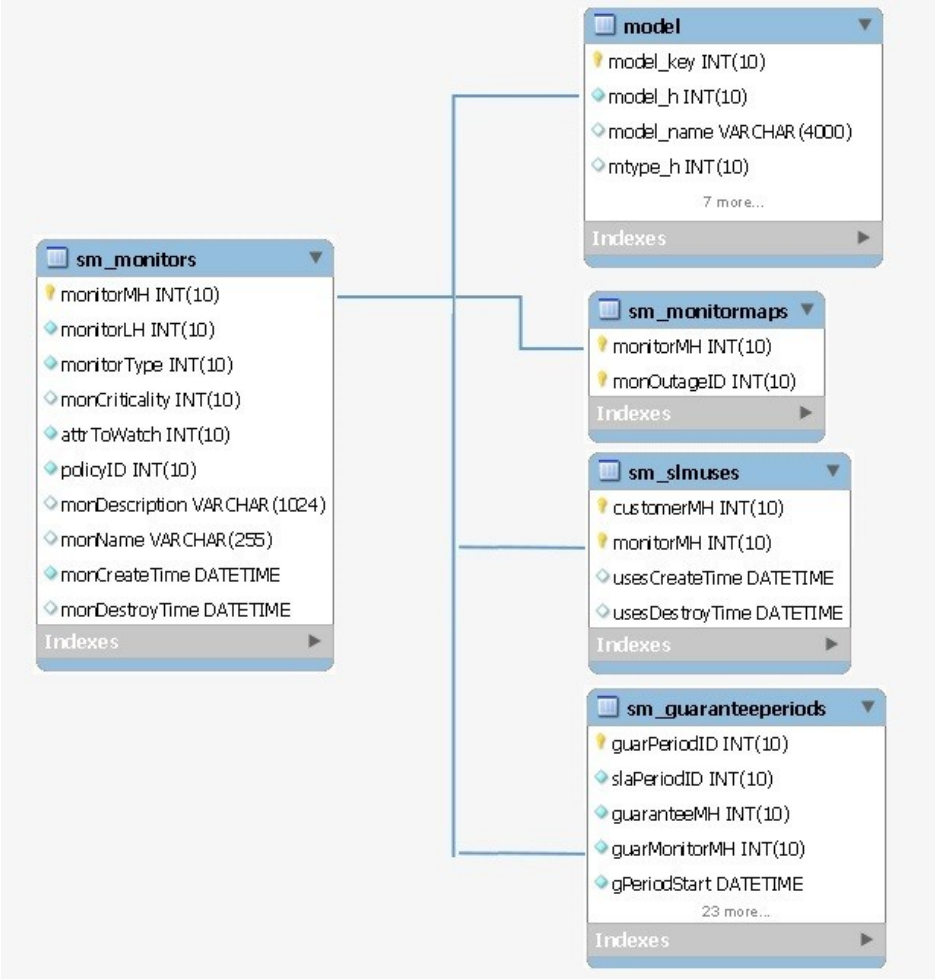
This table stores data of monitor model

Columns

Field	Type	Null	Key	Default	Comment
monitorMH	int(10) unsigned	NO	PRI		Model handle of the monitor

monitorLH	int(10) unsigned	NO			Landscape handle of the monitor
monitorType	int(10) unsigned	NO			type of monitor
monCriticality	int(10) unsigned	YES			Monitor criticality
attrToWatch	int(10) unsigned	NO			Attribute to watch
policyID	int(10) unsigned	NO			Policy ID
monDescription	varchar(1024)	YES			Description of the monitor
monName	varchar(255)	YES			Name of the monitor
monCreateTime	datetime	NO			Monitor creation time
monDestroyTime	datetime	YES			Monitor destroy time

Relations



sm_periods

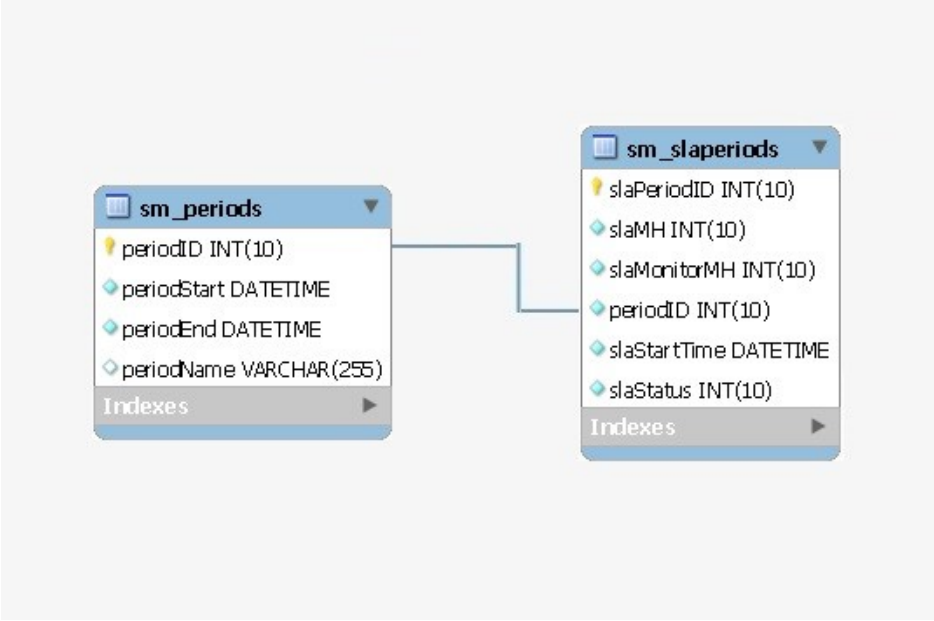
Description

Service Manager Gurantee period Information.

Columns

Field	Type	Null	Key	Default	Extra	Comment
periodID	int(10) unsigned	NO	PRI		auto_increment	Unique ID for each Guarantee period.
periodStart	datetime	NO				Guarantee Start period.
periodEnd	datetime	NO				Guarantee End period.
periodName	varchar(255)	YES				Guarantee period Name.

Relations



sm_policies

Description

Contains information about Service Manager policies.

Column

Field	Type	Null	Key	Default	Comment
policyName	varchar(255)	NO			Name of the Policy
policyID	int(11)	NO	PRI		Unique Policy ID for each service.
policyDescription	text	NO			Description of the Policy

Relations

sm_resourcecauses

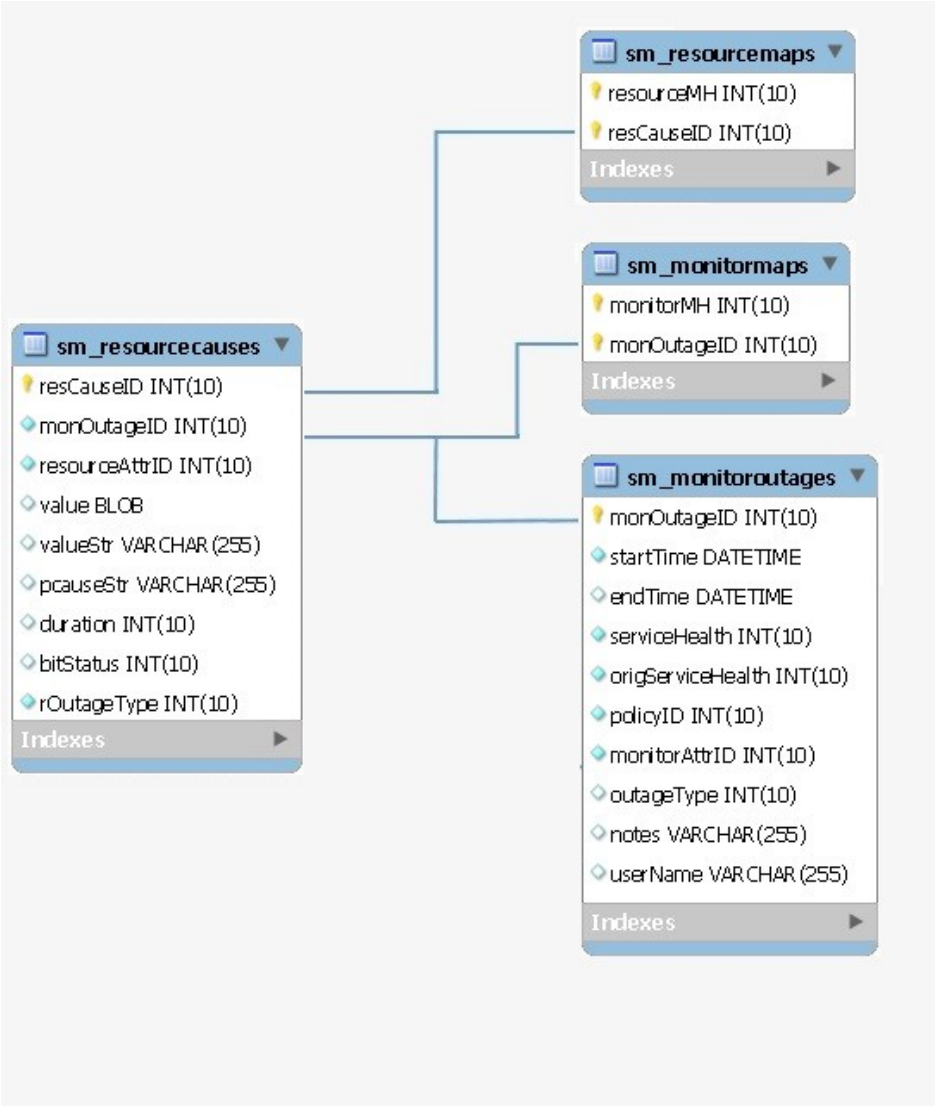
Description

The name of the table used to store the resource outages that make up each monitor outage. Each resource outage contains the model handle of the resource model that the outage was on, and the value that caused the monitor outage.

Column

Field	Type	Null	Key	Default	Extra	Comment
valueStr	varchar(255)	YES				Alarm status of the Resource.
value	blob	YES				Value that caused the monitor outage
rOutageType	int(10) unsigned	NO				Type of the Outage happened to the Resource.
resourceAttrID	int(10) unsigned	NO				Attribute ID of the Resource Monitor
resCauseID	int(10) unsigned	NO	PRI		auto_increment	Unique Cause ID of the Resource.
pcauseStr	varchar(255)	YES				Alarm Title of the Resource.
monOutageID	int(10) unsigned	NO	MUL			Unique Outage ID of the Resource Monitor
duration	int(10) unsigned	YES				Duration of the Event.
bitStatus	int(10) unsigned	YES				bitStatus

Relations



sm_resourcemaps

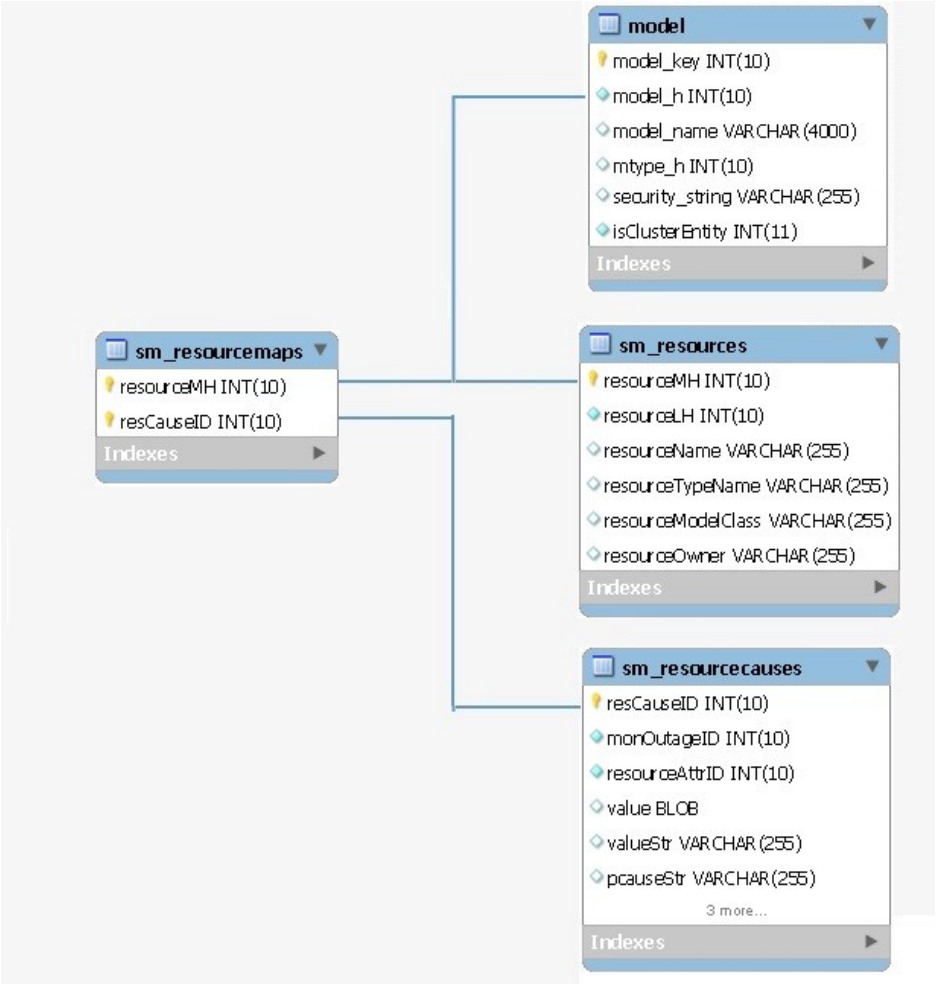
Description

The name of the table used to store the mapping between a resource cause and the model that caused it.

Column

Field	Type	Null	Key	Default	Comment
resourceMH	int(10) unsigned	NO	PRI		Unique Model_Handle of the Resource.
resCauseID	int(10) unsigned	NO	PRI		Unique Cause ID of the Resource.

Relations



sm_resources

Description

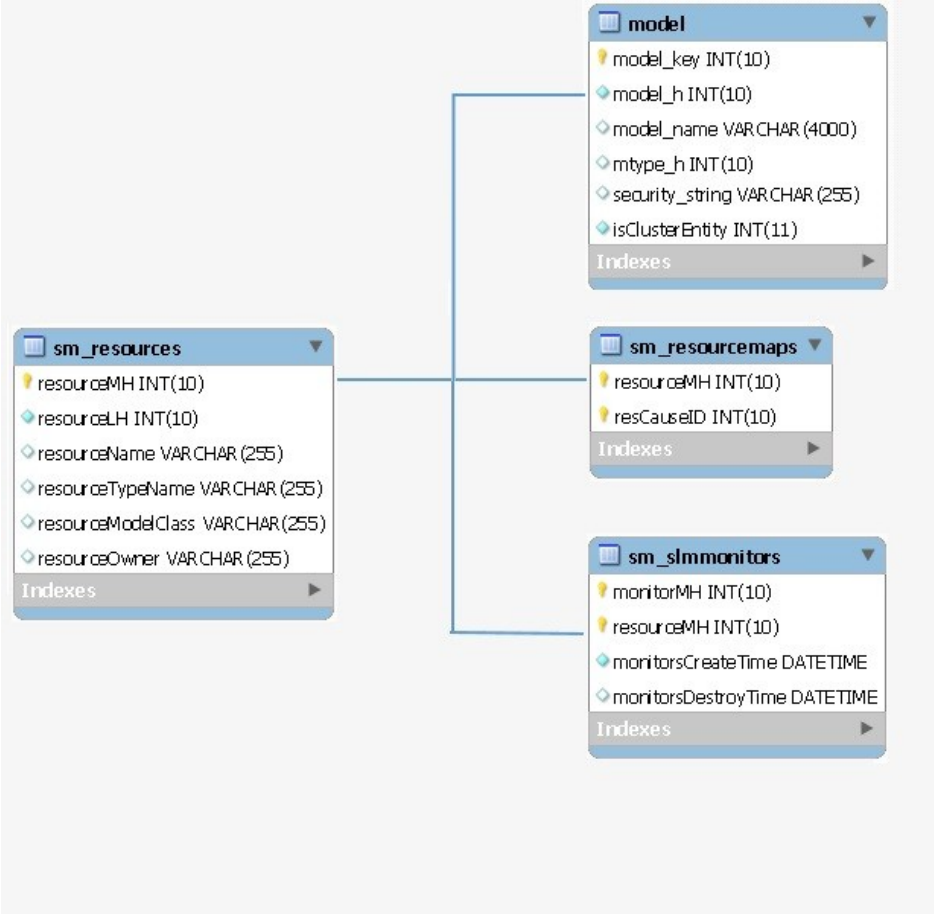
The name of the table that contains all the resources used by a service or attribute monitor.

Column

Field	Type	Null	Key	Default	Comment
resourceTypeName	varchar(255)	YES			Model_Type of the Resource.
resourceOwner	varchar(255)	YES			Owner of the resource
resourceName	varchar(255)	YES			Name of the Resource.
resourceModelClass	varchar(255)	YES			Model_Class of the Resource.
resourceMH	int(10) unsigned	NO	PRI		Unique Model_Handle of the Resource.

resourceLH	int(10) unsigned	NO			Landscape Handle of the Resource.

Relations



sm_schemaversion

Description

The name of the table that contains the current schema version of the SLM DB.

Column

Field	Type	Null	Key	Default	Comment
versionID	varchar(255)	NO	PRI		Unique version ID of the Schema.
comments	varchar(255)	NO			Brief description on the installed version.

Relations

sm_servicehealth

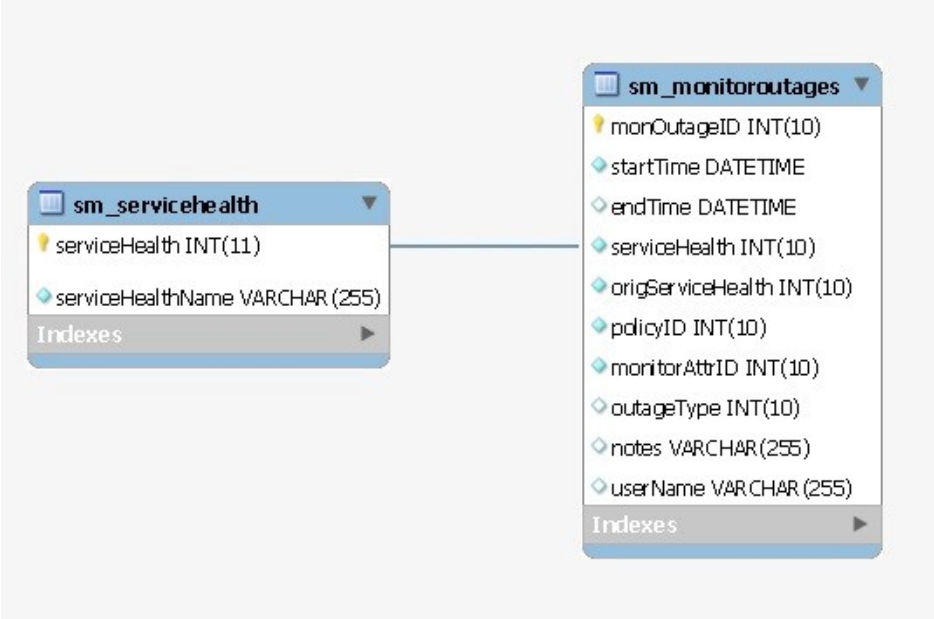
Description

The name of the table used to store attribute type data.

Column

Field	Type	Null	Key	Default	Extra	Comment
serviceHealthName	varchar(255)	NO				Name of the Service.
serviceHealth	int(11)	NO	PRI			Health of the Service.

Relations



sm_slaperiods

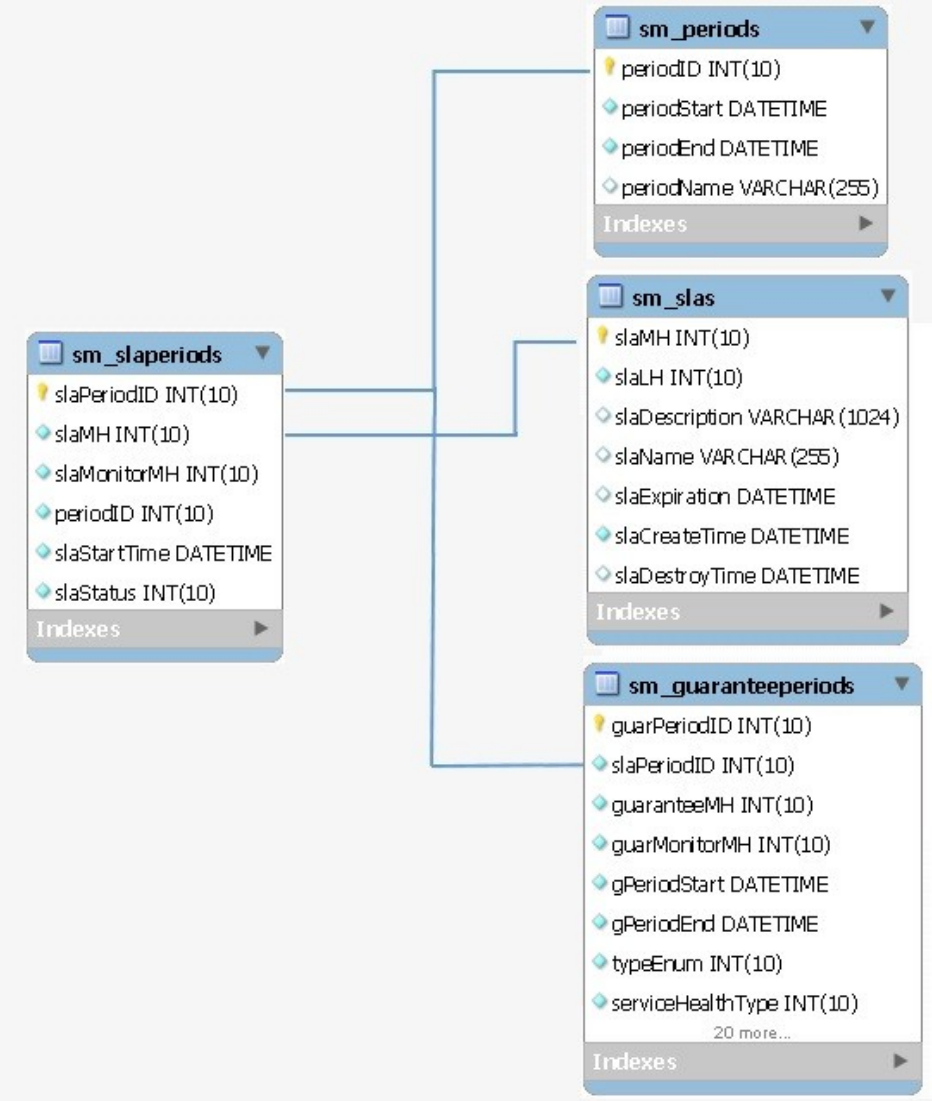
Description

Contains Information of Service Level Agreement.

Columns

Field	Type	Null	Key	Default	Extra	Comment
slaPeriodID	int(10) unsigned	NO	PRI		auto_increment	Unique Period ID for each SLA.
slaMH	int(10) unsigned	NO	MUL			Model_Handle of SLA.
slaMonitorMH	int(10) unsigned	NO				Model_Handle of slaMonitor.
periodID	int(10) unsigned	NO				Period ID of each SLA.
slaStartTime	datetime	NO				Start time of SLA.
slaStatus	int(10) unsigned	NO				Status of SLA

Relations



sm_slas

Description

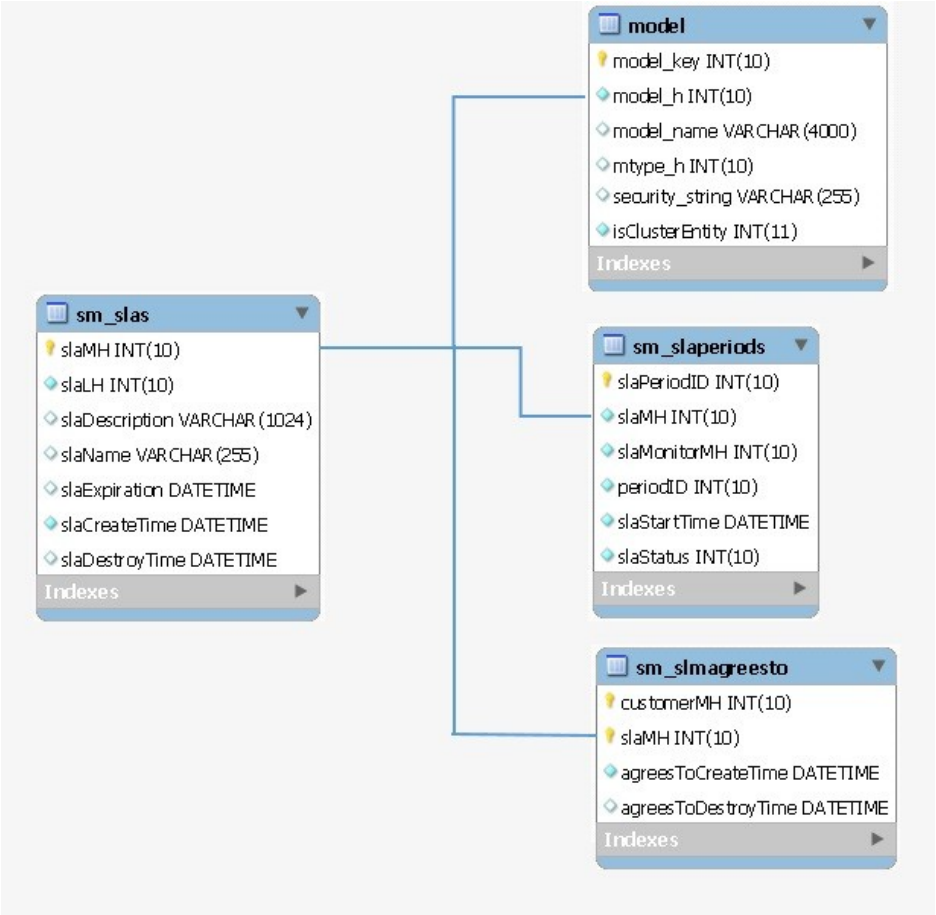
Contains information of all SLA models.

Columns

Column Name	Type	Null	Key	Default	Comment
slaMH	int(10) unsigned	NO	PRI		Model handle of SLA
slaLH	int(10) unsigned	NO			Landscape Handle of SpectroServer where the SLA is present
slaDescription	varchar(1024)	YES			Description of SLA
slaName	varchar(255)	YES			Name of the SLA

slaExpiration	datetime	YES			SLA expiration time
slaCreateTime	datetime	NO			Time when SLA is created
slaDestroyTime	datetime	YES			Time when SLA is destroyed

Relations



sm_slmagreesto

Description

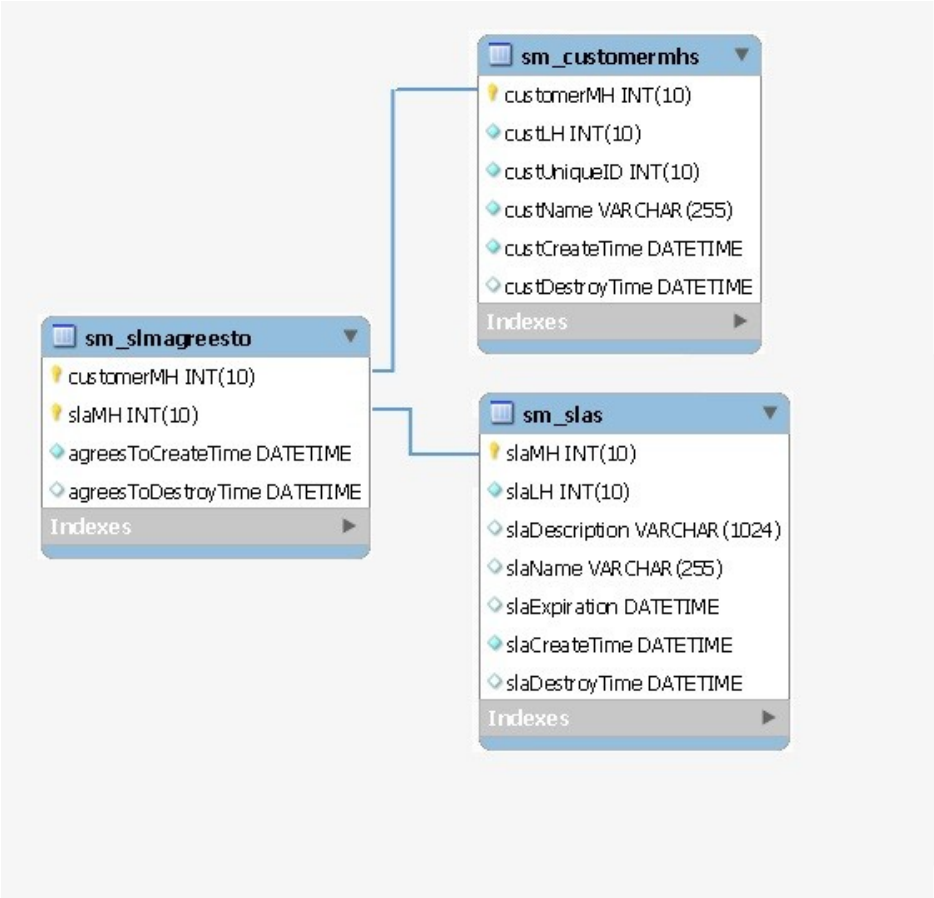
Contains associations between customers and their SLAs.

Columns

Field	Type	Null	Key	Default	Comment
customerMH	int(10) unsigned	NO	PRI		Model handle of customer in decimal
slaMH	int(10) unsigned	NO	PRI		model handle of SLA in decimal

agreesToCreateTime	datetime	NO			The time when customer is associated with SLA
agreesToDestroyTime	datetime	YES			The time when association between customer and SLA is removed

Relations



sm_slmlandscapes

Description

Contains last event retrieved time for each landscape.

Columns

Field	Type	Null	Key	Default	Comment
landscapeHandle	int(10) unsigned	NO	PRI		Landscape handle of SpectroServer
domainName	varchar(255)	NO			Domain Name of SpectroServer

servSyncTime	datetime	YES			Time the last event retrieved from the landscape

Relations

sm_slmmonitors

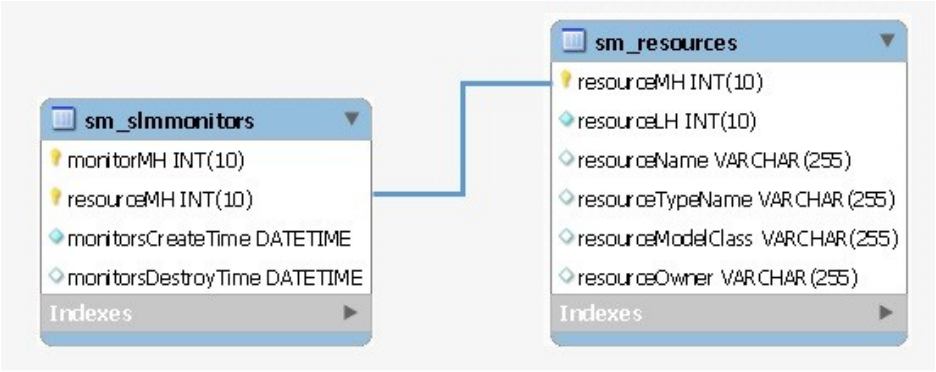
Description

Contains the association between services, resource monitors, and resources.

Columns

Field	Type	Null	Key	Default	Comment
monitorMH	int(10) unsigned	NO	PRI		contains Model handles of services and resource monitors
resourceMH	int(10) unsigned	NO	PRI		Contains model handles of resource monitors and resources (devices)
monitorsCreateTime	datetime	NO			Creation time of monitors (resource monitor or resources/ devices)
monitorsDestroyTime	datetime	YES			Destory time of monitors (resource monitor or resources/ devices)

Relations



sm_slmowns

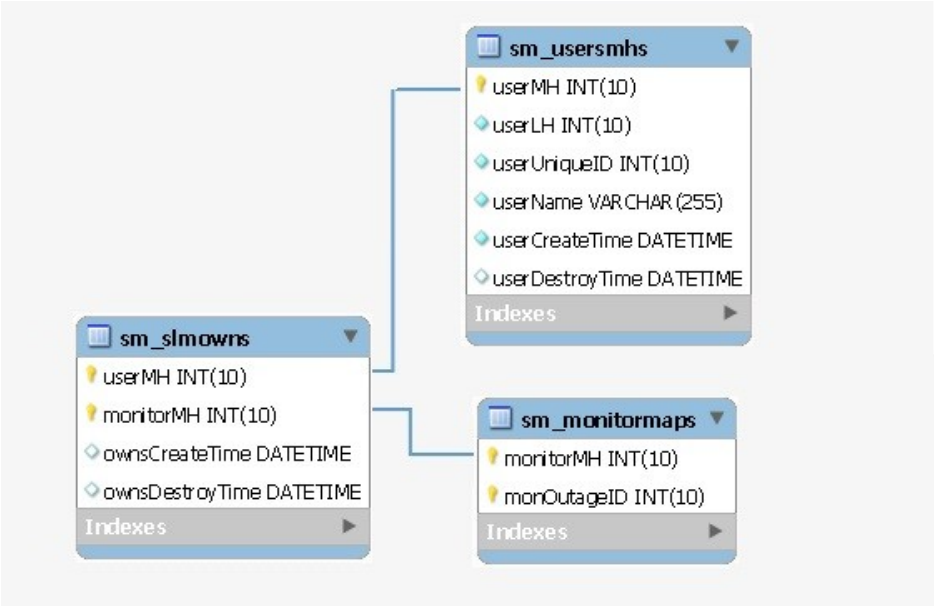
Description

Contains information about the services used by user (owner).

Columns

Field	Type	Null	Key	Default	Comment
userMH	int(10) unsigned	NO	PRI		User model handle
monitorMH	int(10) unsigned	NO	PRI		Model handle of the service that user (owner) uses
ownsCreateTime	datetime	YES			user (owner) creation time
ownsDestroyTime	datetime	YES			user (owner) destroy time

Relations



sm_slmuses

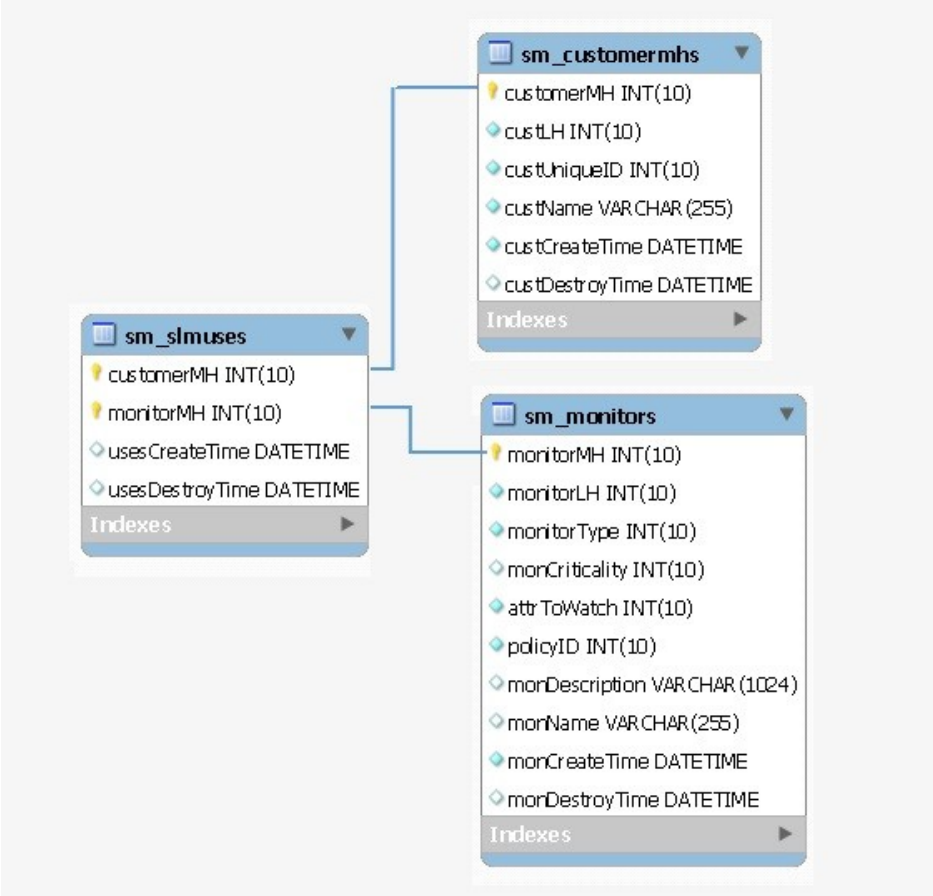
Description

Contains the information about the services used by customer.

Columns

Field	Type	Null	Key	Default	Comment
customerMH	int(10) unsigned	NO	PRI		Customer Model Handle
monitorMH	int(10) unsigned	NO	PRI		Model handle of the service that the customer use
usesCreateTime	datetime	YES			customer creation time
usesDestroyTime	datetime	YES			customer destroy time

Relations



sm_users

Description

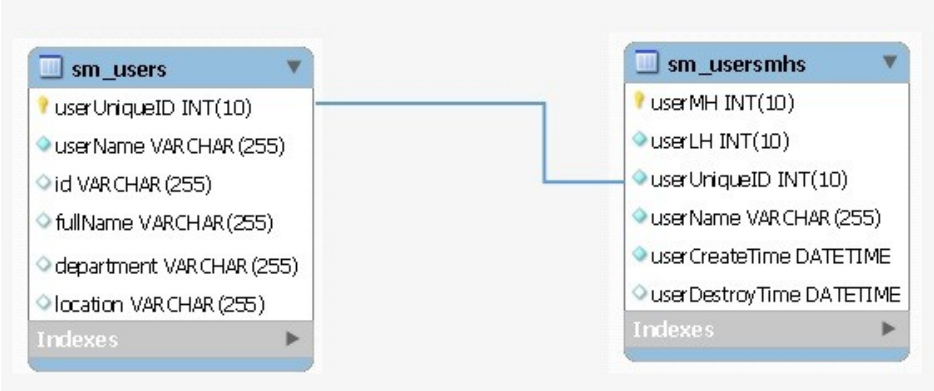
Contains contact information of users.

Columns

Field	Type	Null	Key	Default	Extra	Comment
userUniqueID	int(10) unsigned	NO	PRI		auto_increment	Auto incremented unique user IDs
userName	varchar(255)	NO	UNI			Name of the User
id	varchar(255)	YES				User ID
fullName	varchar(255)	YES				full name of the user
phone	varchar(255)	YES				Phone number of the user
email	varchar(255)	YES				email id of the user
street	varchar(255)	YES				street name of the user

city	varchar(255)	YES				city of the user
state	varchar(255)	YES				state of the user
country	varchar(255)	YES				country of the user
organization	varchar(255)	YES				organization of the user
site	varchar(255)	YES				site of the user loacated
department	varchar(255)	YES				Department of the user
location	varchar(255)	YES				Location of the user

Relations



sm_usersmhs

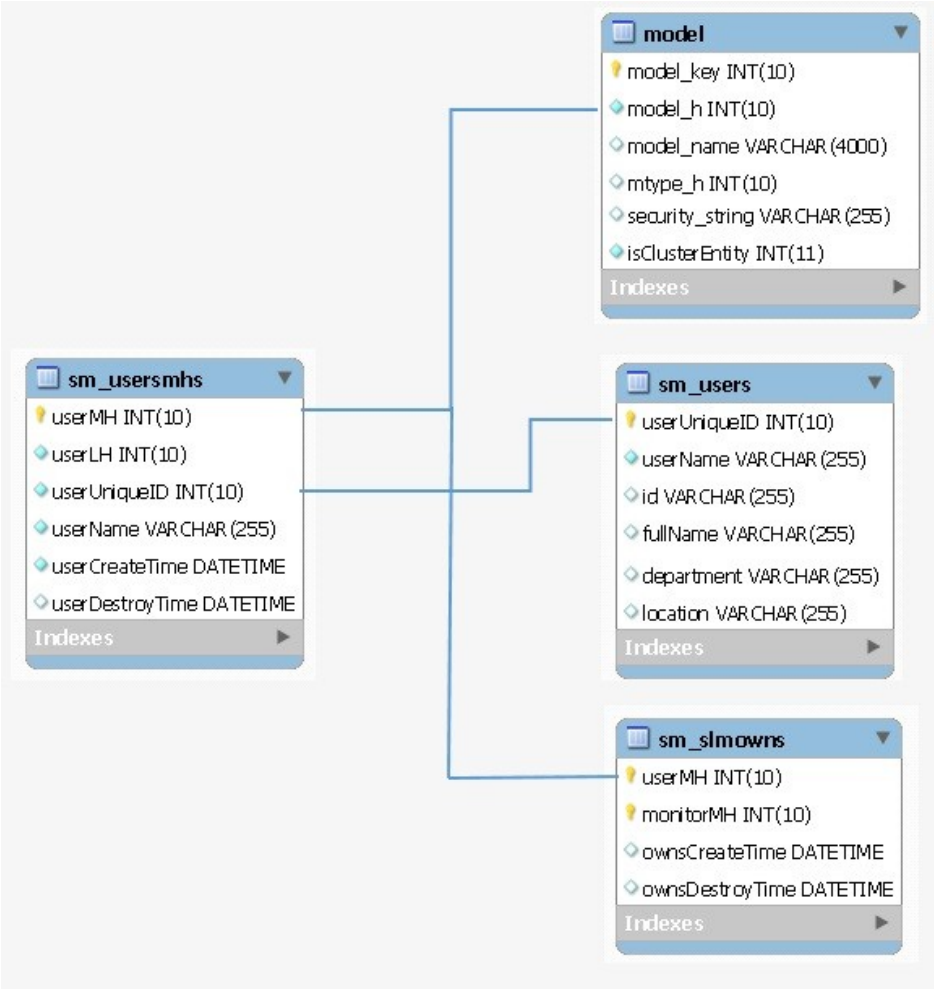
Description

Contain information about service owners or service users.

Columns

Field	Type	Null	Key	Default	Comment
userMH	int(10) unsigned	NO	PRI		service owner model handle
userLH	int(10) unsigned	NO			Landscape handles of the SS where the user is present
userUniqueID	int(10) unsigned	NO	MUL		Auto incremented unique user ids
userName	varchar(255)	NO			Name of service user / owner
userCreateTime	datetime	NO			The time when user created
userDestroyTime	datetime	YES			The time when user Destroyed

Relations



spmbasictestresults

Description

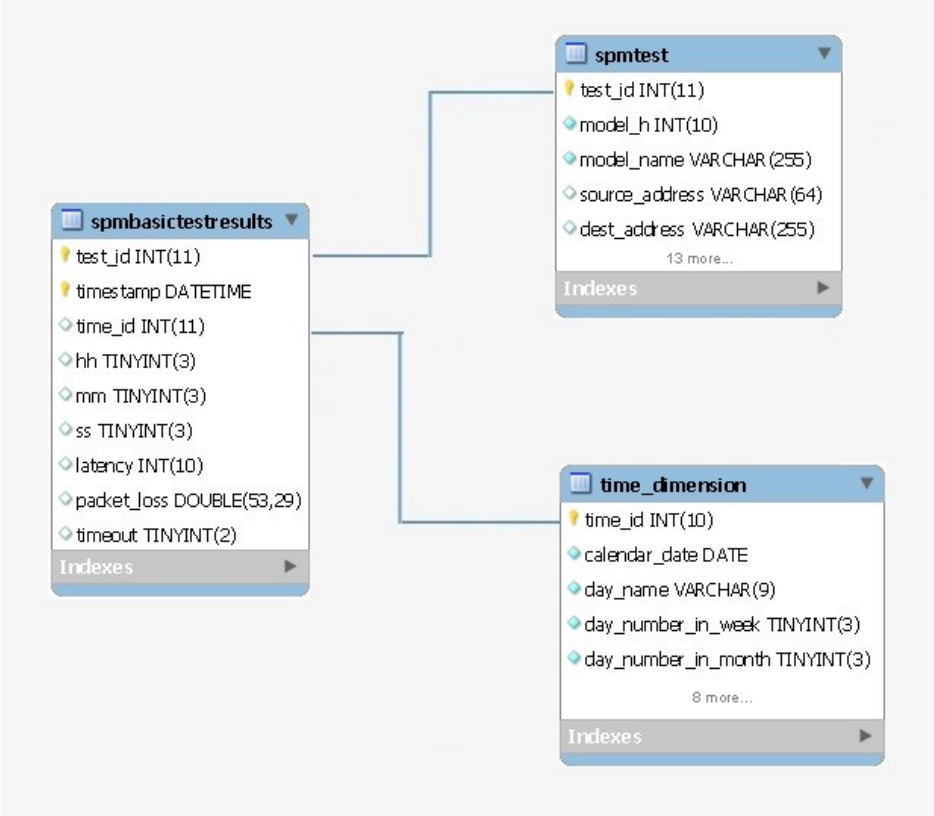
This table stores Service Performance Manager(SPM) basic test results data

Columns

Field	Type	Null	Key	Default	Comment
test_id	int(11) unsigned	NO	PRI	0	Test ID that uniquely identifies a SPM test
timestamp	datetime	NO	PRI	0000-00-00 00:00:00	Timestamp corresponds with time at which result occurred.
time_id	int(11) unsigned	YES	MUL		Time ID from time_dimension table
hh	tinyint(3) unsigned	YES			Hours field of "timestamp" field.

mm	tinyint(3) unsigned	YES			Minutes field of "timestamp" field.
ss	tinyint(3) unsigned	YES			Seconds field of "timestamp" field.
latency	int(10) unsigned	YES			latency for the test
packet_loss	double(53,29)	YES			pocket loss during test
timeout	tinyint(2)	YES			1=timeout occurred, 0=no timeout occurred

Relations



spmhttpfulltestresults

Description

This table stores Service Performance Manager(SPM) http full test results data

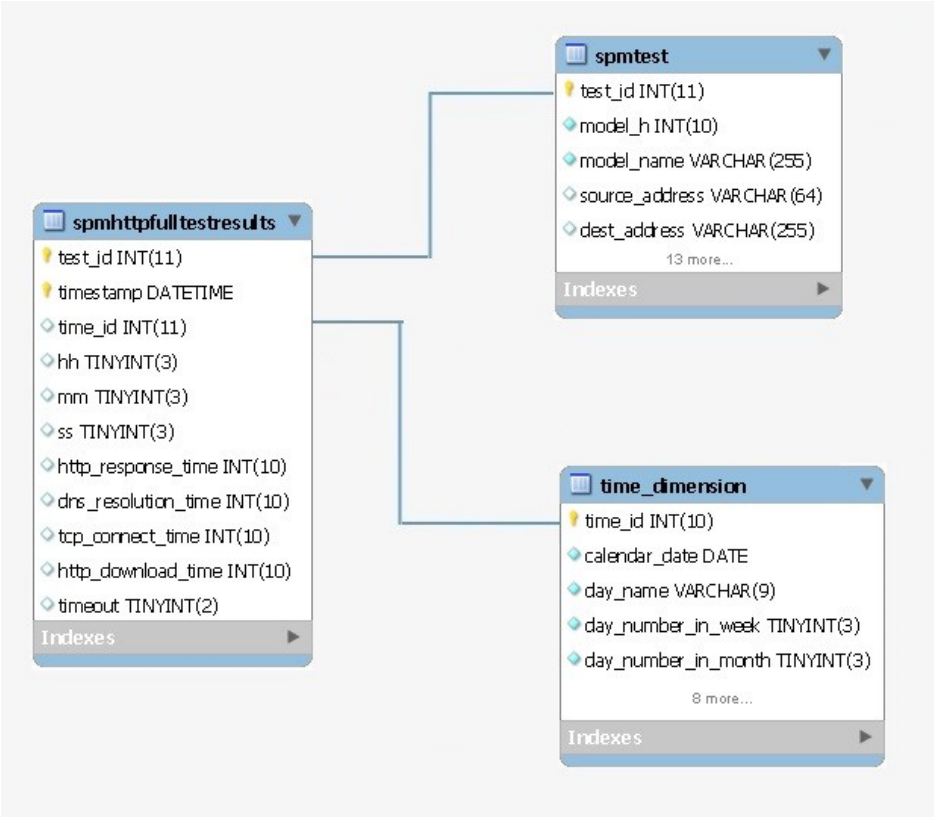
(HTTP tests measure the round-trip time to get a web page.)

Columns

Field	Type	Null	Key	Default	Comment
test_id	int(11) unsigned	NO	PRI	0	Test ID that uniquely identifies a SPM test

timestamp	datetime	NO	PRI	0000-00-00 00:00:00	Timestamp corresponds with time at which result occurred.
time_id	int(11) unsigned	YES	MUL		Time ID from time_dimension table
hh	tinyint(3) unsigned	YES			Hours field of "timestamp" field.
mm	tinyint(3) unsigned	YES			Minutes field of "timestamp" field.
ss	tinyint(3) unsigned	YES			Seconds field of "timestamp" field.
http_response_time	int(10) unsigned	YES			http test response time
dns_resolution_time	int(10) unsigned	YES			time took for dns resolution for this test
tcp_connect_time	int(10) unsigned	YES			tcp connect time
http_download_time	int(10) unsigned	YES			http download time
timeout	tinyint(2)	YES			1=timeout occurred, 0=no timeout occurred

Relations



spmjittertestresults

Description

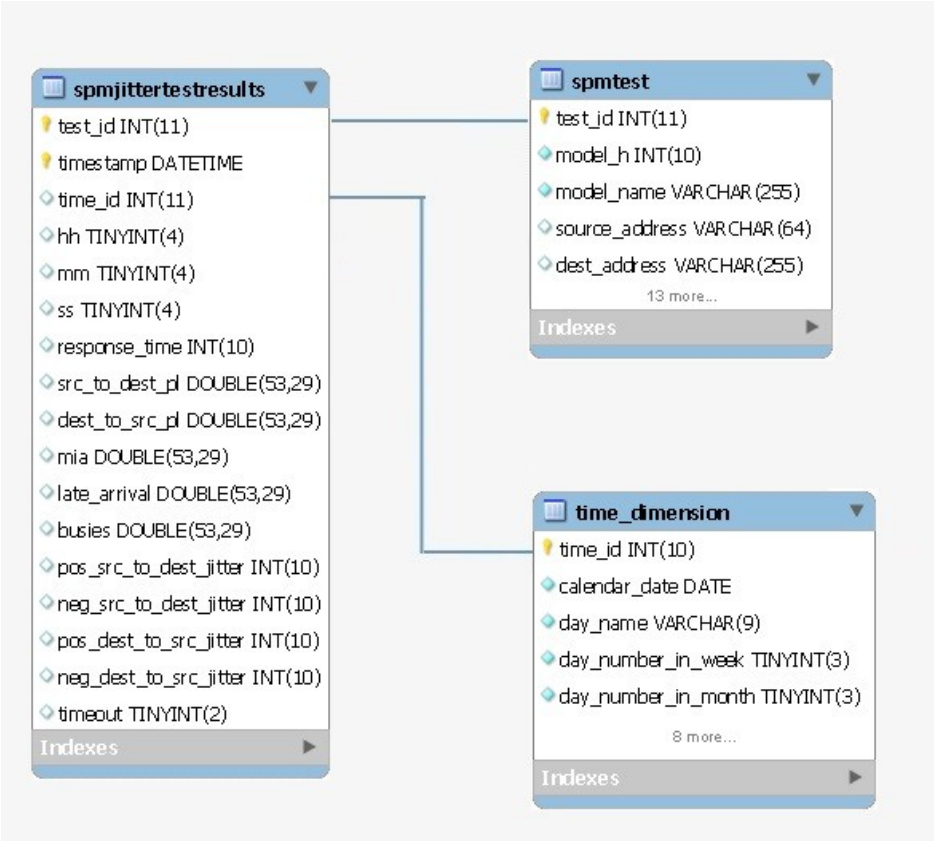
This table stores Service Performance Manager(SPM) jitter test results data

(Jitter tests measure both latency and loss between a test host and a voice-enabled endpoint.)

Columns

Field	Type	Null	Key	Default	Comment
test_id	int(11) unsigned	NO	PRI	0	Test ID that uniquely identifies a SPM test
timestamp	datetime	NO	PRI	0000-00-00 00:00:00	Timestamp corresponds with time at which result occurred.
time_id	int(11) unsigned	YES	MUL		Time ID from time_dimension table
hh	tinyint(4) unsigned	YES			Hours field of "timestamp" field.
mm	tinyint(4) unsigned	YES			Minutes field of "timestamp" field.
ss	tinyint(4) unsigned	YES			Seconds field of "timestamp" field.
response_time	int(10) unsigned	YES			response time for this jitter test
src_to_dest_pl	double(53,29)	YES			Source to Destination Packet Loss
dest_to_src_pl	double(53,29)	YES			Destination to Source Packet Loss
mia	double(53,29)	YES			Missing in Action – Packet Loss with Unknown Direction
late_arrival	double(53,29)	YES			Late Arrival
busies	double(53,29)	YES			Busies
pos_src_to_dest_jitter	int(10) unsigned	YES			Positive source to destination jitter
neg_src_to_dest_jitter	int(10) unsigned	YES			Negative source to destination jitter
pos_dest_to_src_jitter	int(10) unsigned	YES			Positive destination to source jitter
neg_dest_to_src_jitter	int(10) unsigned	YES			Negative destination to source jitter
timeout	tinyint(2)	YES			1=timeout occurred, 0=no timeout occurred

Relations



spmtest

Description

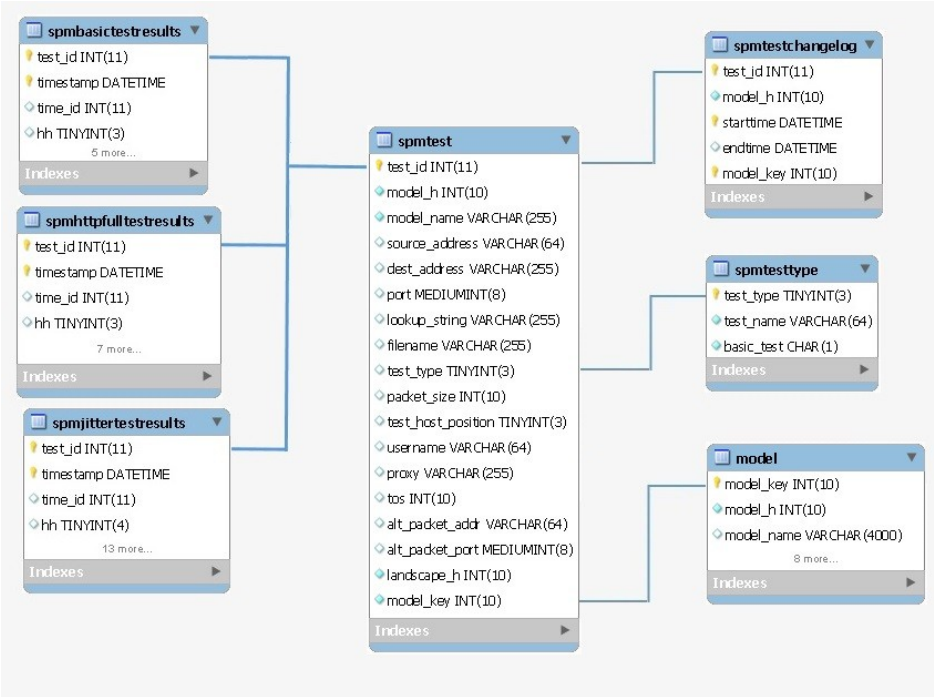
This table stores Service Performance Manager(SPM) test data

Columns

Field	Type	Null	Key	Default	Extra	Comment
test_id	int(11) unsigned	NO	PRI		auto_increment	Test ID that uniquely identifies a SPM test
model_h	int(10) unsigned	NO	MUL	0		Model Handle of the SPM Test Model (decimal form)
model_name	varchar(255)	NO				Model Name of the SPM Test model
source_address	varchar(64)	YES				Source address
dest_address	varchar(255)	YES				Destination address
port	mediumint(8) unsigned	YES				port
lookup_string	varchar(255)	YES				Lookup String

filename	varchar(255)	YES				Filename
test_type	tinyint(3) unsigned	YES	MUL			Type of test
packet_size	int(10)	YES				Packet Size
test_host_position	tinyint(3) unsigned	YES				Test Host Position
username	varchar(64)	YES				user name
proxy	varchar(255)	YES				proxy
tos	int(10) unsigned	YES				Type of service
alt_packet_addr	varchar(64)	YES				Alternate Packet Address
alt_packet_port	mediumint(8) unsigned	YES				Alternate Packet Port
landscape_h	int(10) unsigned	NO		0		Landscape Handle (hexadecimal form)
model_key	int(10) unsigned	NO	MUL	0		Internal Key that uniquely identifies the SPM Test Model

Relations



spmtestchangelog

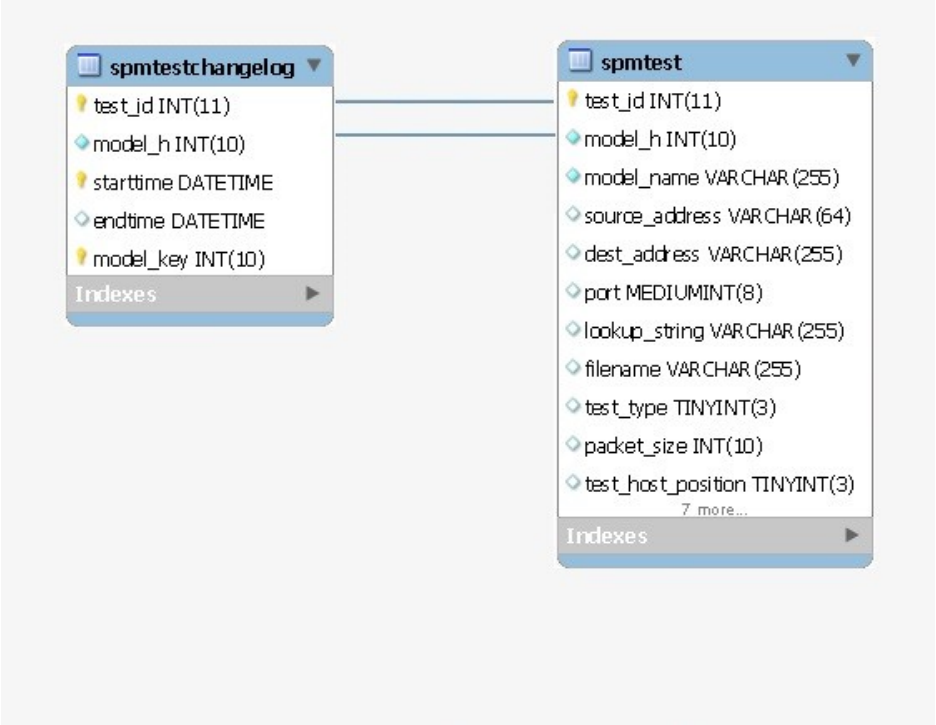
Description

This table stores Service Performance Manager(SPM) test change log data

Columns

Field	Type	Null	Key	Default	Comment
test_id	int(11) unsigned	NO	PRI	0	Test ID that uniquely identifies a SPM test
model_h	int(10) unsigned	NO	MUL	0	Model Handle of the SPM Test Model (decimal form)
starttime	datetime	NO	PRI	0000-00-00 00:00:00	Test start time
endtime	datetime	YES			Test end time
model_key	int(10) unsigned	NO	PRI	0	Internal Key that uniquely identifies the SPM Test Model

Relations



spmtesttype

Description

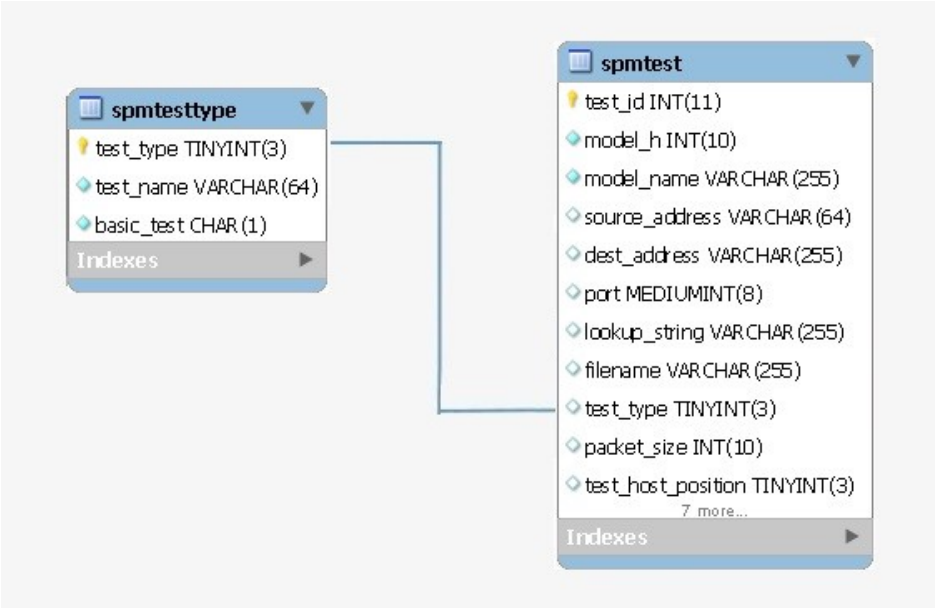
This table stores Service Performance Manager(SPM) test type data like test name and test type etc.

Columns

Field	Type	Null	Key	Default	Comment
test_type	tinyint(3) unsigned	NO	PRI	0	Type of SPM test
test_name	varchar(64)	NO			Name of the SPM test

basic_test	char(1)	NO		Y	Whether this test is basic test or not (Y/N)

Relations



time_dimension

Description

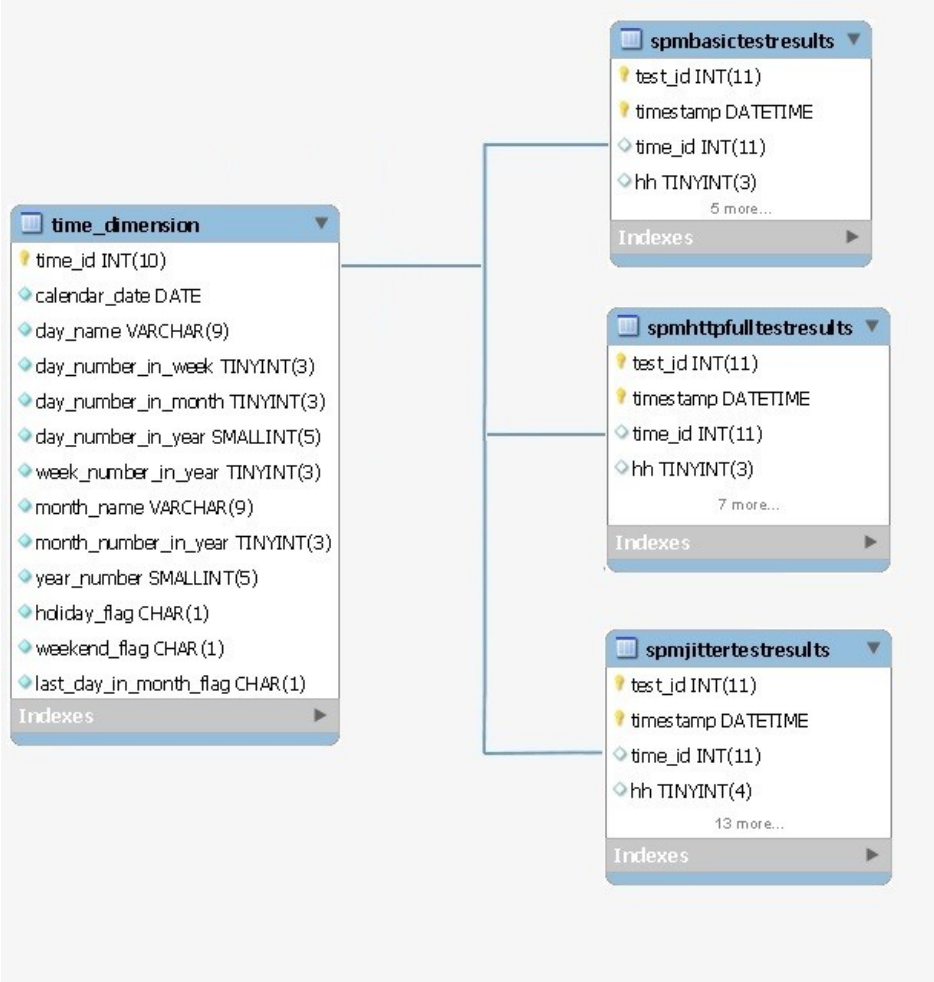
This table stores time dimension data for SPM test result tables

Columns

Field	Type	Null	Key	Default	Extra	Comment
time_id	int(10) unsigned	NO	PRI		auto_increment	Internal ID that uniquely identifies a time in the calendar
calendar_date	date	NO	UNI			Calendar date
day_name	varchar(9)	NO				Day Name (for example, Wednesday)
day_number_in_week	tinyint(3) unsigned	NO				Day Number in Week (Sunday=1, Saturday=7)
day_number_in_month	tinyint(3) unsigned	NO				Day Number in Month
day_number_in_year	smallint(5) unsigned	NO				Day Number in Year
week_number_in_year	tinyint(3) unsigned	NO				Week Number in Year
month_name	varchar(9)	NO				Month Name (for example, January)
month_number_in_year	tinyint(3) unsigned	NO				Month Number in Year (January = 1, December = 12)

year_number	smallint(5) unsigned	NO				Year Number
holiday_flag	char(1)	NO		N		Holiday flag (Y if holiday, N otherwise)
weekend_flag	char(1)	NO		N		Weekend Flag (Y, if Saturday or Sunday)
last_day_in_month_flag	char(1)	NO		N		Last Day in Month Flag (Y, if last day of month)

Relations



translated_string

Description

This table stores string translation data with basic language and target language information

Columns

Field	Type	Null	Key	Default	Extra	Comment
translated_string_id	bigint(20) unsigned	NO	PRI		auto_increment	Translated string ID
base_language	char(5)	YES				Base language of the string
base_language_string	char(255)	YES	MUL			Actual string in the base language
target_language	char(5)	YES				Target language of the string
target_language_string	char(255)	YES				Actual string in the target language

Relations

vendor

Description

This table stores device model vendor data

Columns

Field	Type	Null	Key	Default	Comment
vendor	int(10) unsigned	NO	PRI		Device model vendor ID
vendor_name	varchar(32)	NO			Device model vendor name

Relations

wirelessaps

Description

The table wirelessaps list the following details for event(s) raised on Wireless Controller / Access Points.

Field	Type	Null	Key	Default	Extra	Comment
ap_id	bigint(20) unsigned	NO	PRI	NULL	auto_increment	Unique Identifier
event_key	bigint(20) unsigned	NO		NULL		Event key for the Alarm.
timestamp	datetime	NO		NULL		Timestamp corresponds with time at which event occurred.
controller_mh	int(10) unsigned	NO	MUL	NULL		Model Handle of the WLC Controller.
ap_mh	int(10) unsigned	NO	MUL	NULL		Model Handle of the Access Point.
ap_grpName	varchar(255)	NO	MUL	NULL		Access Point Group Name.
clients_connected	int(10) unsigned	YES		0		Number of Clients connected to Access Point.

data_sent	bigint(20) unsigned	YES		0		The number of bytes sent on the Access Point.
data_received	bigint(20) unsigned	YES		0		The number of bytes received on the Access Point.

wkpeventfilemap

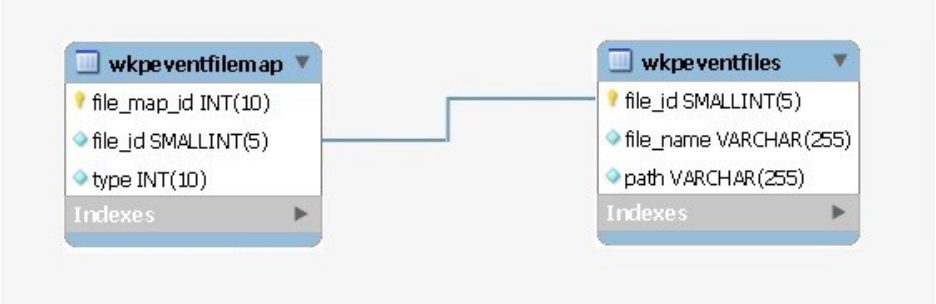
Description

The table wkpeventfilemap maps the event types found in the XML file and maps them to the file from which they were read.

Columns

Field	Type	Null	Key	Default	Extra	Comment
file_map_id	int(10) unsigned	NO	PRI		auto_increment	Event file map ID
file_id	smallint(5) unsigned	NO	MUL			Event file ID
type	int(10) unsigned	NO				Event file type

Relations



wkpeventfiles

Description

This table lists all the event filter XML files that were read from custom directory in OneClick.

Event filters are uniquely named sets of any number of predefined event codes. When users configure an event report, they can elect to specify in the Select the Event Types to include or exclude a field whether to include or exclude data in the report from events in a particular event filter.

An event filter is defined by an XML file that specifies the event codes. Users can create new event filter files, and they can copy and modify the event filter files included with Report Manager. This file resides in custom directory of OneClick.

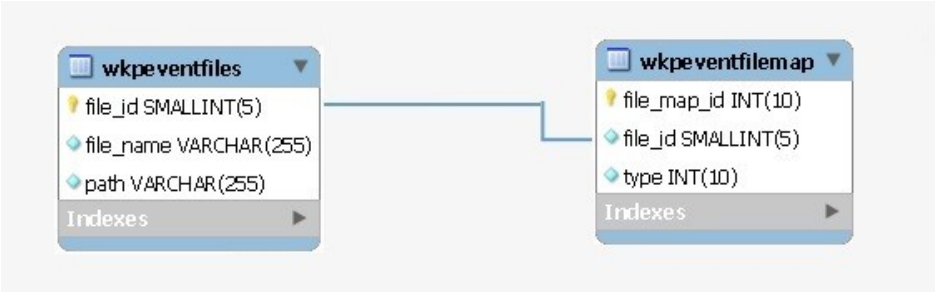
Report manager reads the event filter XML files from custom directory of OneClick. Once read, the data from each XML file is added to two tables: wkpeventfiles and wkpeventfilemap.

Table wkpeventfiles lists all the XML files that were read from the custom directory.

Columns

Field	Type	Null	Key	Default	Extra	Comment
file_id	smallint(5) unsigned	NO	PRI		auto_increment	Event file ID
file_name	varchar(255)	NO				Event file name
path	varchar(255)	NO				Event file path

Relations



z_entity

This is a legacy table, not used in reports

z_entitymodel

This is a legacy table, not used in reports

z_interfaceoutage

This is a legacy table, not used in reports

z_outage

This is a legacy table, not used in reports

Views

v_active_user_model

Description

View for a list of all active oneClick users: All Landscapes

Remarks

View: All Landscapes

v_alarm_activity

Description

List of all alarm activities that are done by oneclick users. Contains information related to assigned user, assigned time, trouble ticket ID etc.

Remarks

View: All Landscapes

v_bi_alarm_activity_by_user

Description

View Definition for report Top N Assets with Most Alarms

Remarks

View: All Landscapes

v_bi_topnalarmtypesmain

Description

View Definition for the report Top N Most Common Alarm Types

Remarks

View: All Landscapes

v_bi_topnassetswithmostalarmsmain

Description

View Definition for the report Alarm Activity by User

Remarks

View: All Landscapes

v_ncm_config_diff

Description

View definition to list NCM configuration changes. Will have data related to configID, change time, number of lines that are changed etc.

Remarks

View: All Landscapes

v_security_string_accessibility_by_landscape

Description

View definition to list the security string access data for users of all landscapes

Remarks

View: All Landscapes

v_user_report_security

Description

View definition as a union of all OC users and BO users

Remarks

View: All Landscapes

srmdbapi Database

The DX NetOps Spectrum Report Manager Database API (SRMDBAPI) provides a fully documented set of read-only database objects to support custom data analysis requirements of DX NetOps Spectrum user. Specifically, the SRMDBAPI consists of a set of database views that are contained within a dedicated multidimensional schema in the MySQL instance that is used by Report Manager.

These views contain the following basic content areas:

- Asset
- Alarm
- Outage/Availability
- Event
- SPM
- NCM

The SRMDBAPI allows users to access mission critical Spectrum Report Manager data. Users can point their Business Intelligence (BI) tools to valuable DX NetOps Spectrum data. Some of the possible use cases are as follows:

- Query this critical data using the BI tool in which customers have already invested.
- Extract the Spectrum Report Manager data and place it within another data repository.
- Incorporate Spectrum Report Manager data into a separate CMDB or financial database.

Click here to see how to write sample queries using srmdbapi views.

List of views in 'srmdbapi' database.

v_dim_alarm_condition

Description

This view enumerates the various alarm conditions (for example, Minor, Major) and associated criticality values.

Columns

Field	Key	Type	Length	Description
condition_id	PK	int(unsigned)	10	Internal ID/Key that uniquely identifies a record in this view
condition_name		varchar	11	Condition Name
criticality	UNQ	tinyint	2	Criticality (1=Maintenance, 2=Minor, 3=Major, 4=Critical)

v_dim_alarm_title

Description

This view enumerates the various alarm titles and associated probable causes that have occurred in the reporting database.

Columns

Field	Key	Type	Length	Description
alarm_title_id	PK	int(unsigned)	10	Internal ID/Key that uniquely identifies a record in this view
alarm_title		varchar	255	Alarm Title
pcause_id_hex		varchar	24	Probable Cause Code (hexadecimal form)
pcause_id_dec		int(unsigned)	10	Probable Cause Code (decimal form)
pcause_title		varchar	100	Probable Cause Title

v_dim_alarm_user

Description

This view enumerates the various usernames that are associated with alarm activity captured in the reporting database.

Columns

Field	Key	Type	Length	Description
alarm_user_key	PK	int(unsigned)	10	Internal ID/Key that uniquely identifies a record in this view
alarm_user_name	UNQ	char	255	Username

v_dim_device_model

Description

This view enumerates all devices (active and destroyed) that are captured historically in the reporting database.

Columns

Field	Key	Type	Length	Description
model_key	PK	int(unsigned)	10	Internal ID/Key that uniquely identifies a record in this view
model_h_dec		int(unsigned)	10	Model handle (decimal form)
model_h_hex		varchar	24	Model handle (hexadecimal form)

landscape_h_dec	FK	int(unsigned)	10	Landscape handle (decimal form); join to v_dim_landscape for additional landscape information.
landscape_h_hex		varchar	24	Landscape handle (hexadecimal form)
model_name		varchar	4000	Device Name
create_time		datetime		Creation Time
model_creator		varchar	255	Model Creator
security_string		varchar	255	Security String
destroy_time		datetime		Destroy Time
device_type		varchar	255	Device Type
ip		varchar	255	Network Address
mac		varchar	32	MAC Address
serial_nbr		varchar	255	Serial Number
sys_desc		varchar	255	System Descriptor
fw_rev		varchar	255	Firmware Version
sys_oid		varchar	255	System Object ID
location		varchar	255	Location
contact_person		varchar	255	Contact Person
last_reboot		datetime		Last reboot time
last_reboot_text		varchar	19	Last reboot time (text form)
last_successful_poll		datetime		Last successful poll time
model_destroyer		varchar	255	Model Destroyer
cust_asset_tag		varchar	255	Asset Tag
cust_asset_id		varchar	255	Asset ID
cust_asset_owner		varchar	255	Asset Owner
cust_asset_organization		varchar	255	Asset Organization
cust_asset_office		varchar	255	Asset Office
cust_asset_contract number		varchar	255	Asset Contract Number
cust_asset_contract startdate		varchar	255	Asset Contract Start Date
cust_asset_contract enddate		varchar	255	Asset Contract End Date
cust_asset_description		varchar	255	Asset Description
sdm_host_address		varchar	255	SDM Host Address
mclass_name		varchar	32	Model Class Name

mtype_h_dec		int(unsigned)	10	Model Type Handle (decimal form)
mtype_h_hex		varchar	24	Model Type Handle (hexadecimal form)
mtype_name		varchar	128	Model Type
vendor_name		varchar	32	Vendor Name
topology_model_name_string		varchar	4000	Topology Model Name String; this field can be used to support container-based reporting capabilities.
varchar_1_attrid_hex		varchar	24	Custom Attribute ID (Varchar1)
varchar_2_attrid_hex		varchar	24	Custom Attribute ID (Varchar2)
varchar_3_attrid_hex		varchar	24	Custom Attribute ID (Varchar3)
varchar_4_attrid_hex		varchar	24	Custom Attribute ID (Varchar4)
integer_1_attrid_hex		varchar	24	Custom Attribute ID (Integer1)
integer_2_attrid_hex		varchar	24	Custom Attribute ID (Integer2)
integer_3_attrid_hex		varchar	24	Custom Attribute ID (Integer3)
integer_4_attrid_hex		varchar	24	Custom Attribute ID (Integer4)
datetime_1_attrid_hex		varchar	24	Custom Attribute ID (Datetime1)
datetime_2_attrid_hex		varchar	24	Custom Attribute ID (Datetime2)
varchar_1_label		varchar	255	Custom Attribute Label (Varchar1)
varchar_2_label		varchar	255	Custom Attribute Label (Varchar2)
varchar_3_label		varchar	255	Custom Attribute Label (Varchar3)
varchar_4_label		varchar	255	Custom Attribute Label (Varchar4)
integer_1_label		varchar	255	Custom Attribute Label (Integer1)
integer_2_label		varchar	255	Custom Attribute Label (Integer2)
integer_3_label		varchar	255	Custom Attribute Label (Integer3)
integer_4_label		varchar	255	Custom Attribute Label (Integer4)
datetime_1_label		varchar	255	Custom Attribute Label (Datetime1)

datetime_2_label		varchar	255	Custom Attribute Label (Datetime2)
varchar_1_value		varchar	4000	Custom Attribute Value (Varchar1)
varchar_2_value		varchar	4000	Custom Attribute Value (Varchar2)
varchar_3_value		varchar	4000	Custom Attribute Value (Varchar3)
varchar_4_value		varchar	4000	Custom Attribute Value (Varchar4)
integer_1_value		bigint	20	Custom Attribute Value (Integer1)
integer_2_value		bigint	20	Custom Attribute Value (Integer2)
integer_3_value		bigint	20	Custom Attribute Value (Integer3)
integer_4_value		bigint	20	Custom Attribute Value (Integer4)
datetime_1_value		datetime		Custom Attribute Value (Datetime1)
datetime_2_value		datetime		Custom Attribute Value (Datetime2)

v_dim_device_module

Description

This view enumerates more information at the slot level for chassis-based devices.

Columns

Field	Key	Type	Length	Description
module_id	PK	int(unsigned)	10	Internal ID/Key that uniquely identifies a record in this view
model_key	FK	int(unsigned)	10	Model Key that is associated with the parent device recorded in v_dim_device_model view; use this field to join to v_dim_device_model for additional device information.
module_index		int	10	Module Index (Slot)
module_name		varchar	255	Module Name (Description)
serial_nbr		varchar	255	Serial Number
software_rev		varchar	255	Software Version

v_dim_event

Description

This view enumerates all the Event Types encountered while processing events for reporting purposes.

Columns

Field	Key	Type	Length	Description
type_dec	PK	int(unsigned)	10	Event Type (decimal form); this field also uniquely identifies a record in this view.
type_hex		varchar	24	Event Type (hexadecimal form)
title		varchar	255	Event Title

v_dim_event_creator

Description

This view enumerates all the event creators that are encountered while processing events for reporting purposes.

Columns

Field	Key	Type	Length	Description
creator_id	PK	int(unsigned)	10	Internal ID/Key that uniquely identifies a record in this view
creator_name		varchar	255	Creator Name

v_dim_global_collection_member

Description

This view enumerates all global collection members in the reporting database. You have a separate record for every global collection/model pairing.

Columns

Field	Key	Type	Length	Description
gc_rec_ID	PK	int(unsigned)	10	Internal ID/Key that uniquely identifies a record in this view
gc_name		varchar	255	Global Collection Name

model_key	FK	int(unsigned)	10	Foreign key that uniquely identifies a member model. This field can be used to join to v_dim_model, v_dim_device_model, or v_dim_interface_model for additional member model information.

v_dim_interface_model

Description

This view enumerates all interfaces that are captured in the reporting database.

Columns

Field	Key	Type	Length	Description
model_key	PK	int(unsigned)	10	Internal ID/Key that uniquely identifies a record in this view
model_h_dec		int(unsigned)	10	Model handle (decimal form)
model_h_hex		varchar	24	Model handle (hexadecimal form)
landscape_h_dec	FK	int(unsigned)	10	Landscape handle (decimal form) associated with this model; this field can be used to join to v_dim_landscape for additional landscape information.
landscape_h_hex		varchar	24	Landscape handle (hexadecimal form)
model_name		varchar	4000	Interface Name
create_time		datetime		Creation Time
security_string		varchar	255	Security String
destroy_time		datetime		Destroy Time
port_type		varchar	255	Port Type
port_desc		varchar	255	Port Description (raw value)
port_desc_text		longtext		Port Description (transformed value)
if_speed		Bigint (unsigned)	20	If Speed (Bytes/Sec)
ip		varchar	255	Network Address
mac		varchar	32	MAC Address

port_link_status		int(unsigned)	10	Port Link Status (raw value)
port_link_status_text		varchar	32	Port Link Status (transformed value)
iflastchange		bigint (unsigned)	20	Last Change
ifinoctets		bigint (unsigned)	20	If In Octets
Datelastsignificant traffic		datetime		Date Last Significant Traffic
hours_idle		bigint	21	Hours Idle
days_idle		bigint	21	Days Idle
ifalias		varchar	4000	If Alias
component_oid		varchar	255	Component OID
device_model_key	FK	int(unsigned)	10	Foreign Key that uniquely identifies the parent device for this interface. Use this field to join to v_dim_device_model.model_key for additional parent device information.
device_model_name		varchar	4000	Parent Device Name
port_status		varchar	32	Port Status
mclass_name		varchar	32	Model Class
mtype_h_dec		int(unsigned)	10	Model Type Handle (decimal form)
mtype_h_hex		varchar	24	Model Type Handle (hexadecimal form)
mtype_name		varchar	128	Model Type
connected_model_h_dec	FK	int(unsigned)	10	Model Handle of Connected Device (decimal form); The value is NULL if no device is connected. Use this field to join to v_dim_device_model.model_h for additional connected device information.
connected_model_h_hex		varchar	24	Model Handle of Connected Device (hexadecimal form); This is NULL if no device is connected.
is_connected		int	1	1 indicates that there is a connected device, 0 indicates no connected device.
duplex_status		varchar	255	Duplex Status

v_dim_landscape

Description

This view enumerates all the landscapes that have been encountered during processing reporting data.

Columns

Field	Key	Type	Length	Description
landscape_h_dec	PK	int(unsigned)	10	Landscape handle (decimal form)
landscape_h_hex		varchar	24	Landscape handle (hexadecimal form)
landscape_name		varchar	255	Landscape(Domain) Name

v_dim_model

Description

This view enumerates all the models that are encountered while processing reporting data.

Columns

Field	Key	Type	Length	Description
model_key	PK	int(unsigned)	10	Internal ID/Key that uniquely identifies a record in this view
model_h_dec		int(unsigned)	10	Model handle (decimal form)
model_h_hex		varchar	24	Model handle (hexadecimal form)
model_name		varchar	4000	Model Name
network_address		varchar	255	Network Address
landscape_h_dec	FK	int(unsigned)	10	Landscape Handle (decimal form); join to v_dim_landscape for additional landscape information.
landscape_h_hex		varchar	24	Landscape Handle (hexadecimal form)
mclass_name		varchar	32	Model Class
mtype_h_dec		int(unsigned)	10	Model Type Handle (decimal form)
mtype_h_hex		varchar	24	Model Type Handle (hexadecimal form)
mtype_name		varchar	128	Model Type Name
security_string		varchar	255	Security String

destroy_time		datetime		Destroy Time (if applicable)

v_dim_ncm_event

Description

This view enumerates all event codes that are associated with Network Configuration Management (NCM).

Columns

Field	Key	Type	Length	Description
type_dec	PK	int(unsigned)	10	Event Type (decimal form); this field also uniquely identifies a record in this view.
type_hex		varchar	22	Event Type (hexadecimal form)
title		varchar	255	Event Title

v_dim_secure_device_model

v_dim_secure_device_model_nofx

v_dim_secure_interface_model

v_dim_secure_interface_model_nofx

v_dim_secure_model

v_dim_secure_model_nofx

v_dim_spm_test

Description

This view enumerates all the SPM Tests that are created while processing.

Columns

Field	Key	Type	Length	Description
test_id	PK	int(unsigned)	11	Internal ID/Key that uniquely identifies a record in this view
test_name		varchar	64	SPM Test Name
model_key	FK	int(unsigned)	10	Internal Key that uniquely identifies the SPM Test Model in v_dim_model.

model_h_dec		int(unsigned)	10	Model Handle of the SPM Test Model (decimal form)
model_h_hex		varchar	24	Model Handle of the SPM Test Model (hexadecimal form)
model_name		varchar	255	Model Name of the SPM Test model
source_address		varchar	64	Source Address
dest_address		varchar	255	Destination Address
port		mediumint (unsigned)	8	Port
lookup_string		varchar	255	Lookup String
filename		varchar	255	Filename
packet_size		int	10	Packet Size
test_host_position		tinyint (unsigned)	3	Test Host Position
username		varchar	64	Username
proxy		varchar	255	Proxy
tos		int(unsigned)	10	Type of Service
alt_packet_addr		varchar	64	Alternate Packet Address
alt_packet_port		mediumint (unsigned)	8	Alternate Packet Port
landscape_h_dec	FK	int(unsigned)	10	Landscape Handle (decimal form); join to v_dim_landscape for additional landscape information.
landscape_h_hex		varchar	24	Landscape Handle (hexadecimal form)
effective_start		datetime		Effective start time of the test
effective_end		datetime		Effective end time of the test (if applicable)

v_dim_time

Description

This view enumerates a separate record for every day in the calendar.

Columns

Field	Key	Type	Length	Description
time_id	PK	int(unsigned)	10	Internal ID/Key that uniquely identifies a record in this view
calendar_date	UNQ	date		Calendar Date

day_name		varchar	9	Day Name (for example, Wednesday)
day_number_in_week		tinyint (unsigned)	3	Day Number in Week (Sunday=1, Saturday=7)
day_number_in_month		tinyint (unsigned)	3	Day Number in Month
day_number_in_year		smallint (unsigned)	5	Day Number in Year
week_number_in_year		tinyint (unsigned)	3	Week Number in Year
month_name		varchar	9	Month Name (for example, January)
month_number_in_year		tinyint (unsigned)	3	Month Number in Year (January = 1, December = 12)
year_number		smallint (unsigned)	5	Year Number
weekend_flag		char	1	Weekend Flag (Y, if Saturday or Sunday)
last_day_in_month_flag		char	1	Last Day in Month Flag (Y, if last day of month)

v_fact_alarm_activity

Description

This view enumerates alarm activities (for example, sets, clears, acknowledgements) that are processed in reporting database.

Columns

Field	Key	Type	Length	Description
alarm_key		int(unsigned)	10	Internal ID/Key that uniquely identifies a record in this view
activity		int(unsigned)	10	Internal Code that is used to identify various activities (1=Set, 2=Ack, 3=Assigned By, 33=Assigned To, 4 or 5=Clear, 6=Ticketed)
activity_title		varchar	17	Activity Title (for example, Set, Acknowledged, and so on)
time		datetime		Time at which activity occurred
username_text		varchar	50	Username that is associated with activity
set_count		int	1	Set Count

ack_count		int	1	Acknowledgment Count
assign_by_count		int	1	Assign By Count
assign_to_count		int	1	Assign To Count
clear_count		int	1	Clear Count
ticketed_count		int	1	Ticketed Count
data		char	255	Additional details

v_fact_alarm_info

Description

This view enumerates a separate record for every alarm that is processed in reporting database.

Columns

Field	Key	Type	Length	Description
alarm_key	PK	int(unsigned)	11	Internal ID/Key that uniquely identifies a record in this view
landscape_h_dec	FK	int(unsigned)	10	Landscape Handle (Decimal Form); join to v_dim_landscape for additional landscape information.
landscape_h_hex		varchar	24	Landscape Handle (Hexadecimal Form)
orig_event_key	FK	bigint (unsigned)	20	Originating Event Key; join to v_fact_event.event_key to capture additional event details.
condition_id	FK	int	11	Condition ID; join to v_dim_alarm_condition for additional condition information.
cause_id		int(unsigned)	10	Cause ID
set_time		datetime		Set Time
clear_time		datetime		Clear Time (if applicable)
duration_seconds		bigint	21	Duration in Seconds
duration_label		varchar	24	Duration Label (HH:MM:SS)
clear_user_key	FK	int(unsigned)	10	Uniquely identifies user who cleared this alarm; join to v_dim_alarm_user.alarm_user_key for more information.
alarm_title_id	FK	int(unsigned)	10	Uniquely identifies an alarm title; join to v_dim_alarm_title for more information.
model_key	FK	int(unsigned)	10	Uniquely identifies the model that is associated with this alarm; join to v_dim_model for more information.
ack_time		datetime		Acknowledgment Time
time_to_ack_seconds		bigint	21	Time to Acknowledge (Seconds)
time_to_ack_duration_label		varchar	23	Time to Acknowledge (HH:MM:SS)

ack_user_key	FK	int(unsigned)	10	Uniquely identifies the acknowledging user in v_dim_alarm_user; join to v_dim_alarm_user.alarm_user_key for more information.
first_assigned_time		datetime		Time at which alarm was first assigned.
time_to_first_assign_seconds		bigint	21	Difference in time between set time and time to first assignment (Seconds)
time_to_first_assign_ duration_label		varchar	23	Difference in time between set time and time to first assignment (HH:MM:SS)
first_assigned_user_key	FK	int(unsigned)	10	Uniquely identifies the first assigned user in v_dim_alarm_user; join to v_dim_alarm_user.alarm_user_key for more information.
first_assigning_user_ key	FK	int(unsigned)	10	Uniquely identifies the first assigning user in v_dim_alarm_user; join to v_dim_alarm_user.alarm_user_key for more information.
last_assigned_time		datetime		Time at which alarm was last assigned.
time_to_last_assign_seconds		bigint	21	Difference in time between set time and time to last assignment (Seconds).
time_to_last_assign_ duration_label		varchar	23	Difference in time between set time and time to last assignment (HH:MM:SS).
last_assigned_user_key	FK	int(unsigned)	10	Uniquely identifies the last assigned user in v_dim_alarm_user; join to v_dim_alarm_user.alarm_user_key for more information.
last_assigning_user_key	FK	int(unsigned)	10	Uniquely identifies the last assigning user in v_dim_alarm_user; join to v_dim_alarm_user.alarm_user_key for more information.
set_troubleticket_time		datetime		Trouble ticket Time
time_to_trouble_ticket_seconds		bigint	21	Difference in time between set time and trouble ticket time (Seconds).
time_to_trouble_ticket_ duration_label		varchar	23	Difference in time between set time and trouble ticket time (HH:MM:SS).

set_troubleticket_user_key	FK	int(unsigned)	10	Uniquely identifies the trouble ticket user in v_dim_alarm_user; join to v_dim_alarm_user.alarm_user_key for more information.
set_troubleticket_id		char	255	Trouble Ticket ID
assignment_duration_seconds		bigint	21	Difference in time between lastassigned time and clear time (Seconds)
assignment_duration_label		varchar	24	Difference in time between last assigned time and clear time (HH:MM:SS)

v_fact_event

Description

This view enumerates every event record that is processed in the reporting database.

Columns

Field	Key	Type	Length	Description
event_key	PK	bigint (unsigned)	20	Internal ID/Key that uniquely identifies a record in this view
landscape_h_dec	FK	int(unsigned)	10	Uniquely identifies a landscape that is associated with the model on which this event occurred (decimal form); join to v_dim_landscape for additional landscape information.
landscape_h_hex		varchar	24	Uniquely identifies a landscape that is associated with the model on which this event occurred (hexadecimal form).
model_key	FK	int(unsigned)	10	Uniquely identifies the model that is associated with this event; join to v_dim_model for more information.
time		datetime		Time at which event occurred.
type_dec	FK	int(unsigned)	10	Event Type (decimal form); join to v_dim_event for more information.

type_hex		varchar	24	Event Type (hexadecimal form)
creator_id	FK	int(unsigned)	10	Uniquely identifies the creator for this event; join to v_dim_creator for more information.
event_msg		text		Fully constituted event message that is associated with this event.

v_fact_model_outage

Description

This view enumerates all outages that are processed in reporting database.

Columns

Field	Key	Type	Length	Description
model_outage_id	PK	bigint (unsigned)	20	Internal ID/Key that uniquely identifies a record in this view
model_key	FK	int(unsigned)	10	Uniquely identifies the model that is associated with this outage; join to v_dim_model for more information.
landscape_h_dec	FK	int(unsigned)	10	Uniquely identifies a landscape that is associated with the model on which this event occurred (decimal form).
landscape_h_hex		varchar	24	Uniquely identifies a landscape that is associated with the model on which this event occurred (hexadecimal form).
start_time		datetime		Start Time of Outage
end_time		datetime		End Time of Outage (if applicable)
duration_seconds		bigint	21	Outage Duration (seconds)
duration_label		varchar	24	Outage Duration (HH:MM:SS)

start_event_key	FK	bigint (unsigned)	20	Uniquely identifies the event that started this outage; join to v_fact_event on event_key for more information.
end_event_key	FK	bigint (unsigned)	20	Uniquely identifies the event that ended this outage; join to v_fact_event on event_key for more information.
notes		char	250	Outage Notes
outage_type		int(unsigned)	10	Outage Type (0=Initial, 1=Unplanned, 2=Planned, 3=Exempt)
outage_desc		varchar	NO	Outage Description

v_fact_spm_basic_test_results

Description

This view enumerates test results for the following Service Performance Manager (SPM) test types: ICMP Ping, UDP, Path Echo, TCP, DNS Lookup, POP3, DHCP, FTP, SMTP, and HTTP (total time only).

Columns

Field	Key	Type	Length	Description
test_id	PK	int(unsigned)	11	Combination of Test ID and Timestamp uniquely identifies a record in this view. Timestamp corresponds with time at which result occurred.
timestamp	PK	datetime	11	Combination of Test ID and Timestamp uniquely identifies a record in this view. Timestamp corresponds with time at which result occurred.
time_id	FK	int(unsigned)	11	Uniquely identifies a day in v_dim_time; join to v_dim_time for more information.
hh		tinyint (unsigned)	3	Hours field of "timestamp" field.
mm		tinyint (unsigned)	3	Minutes field of "timestamp" field.

ss		tinyint (unsigned)	3	Seconds field of "timestamp" field.
latency		int(unsigned)	10	Latency in milliseconds
packet_loss		double	53,29	Packet Loss
timeout		tinyint	2	1=timeout occurred, 0=no timeout occurred

v_fact_spm_http_full_test_results

Description

This view enumerates historical results that are associated with Service Performance Manager (SPM) HTTP tests.

Columns

Field	Key	Type	Length	Description
test_id	PK	int(unsigned)	11	Combination of Test ID and Timestamp uniquely identifies a record in this view. Timestamp corresponds with time at which result occurred.
timestamp	PK	datetime	11	Combination of Test ID and Timestamp uniquely identifies a record in this view. Timestamp corresponds with time at which result occurred.
time_id	FK	int(unsigned)	11	Uniquely identifies a day in v_dim_time; join to v_dim_time for more information.
hh		tinyint (unsigned)	3	Hours field of "timestamp" field.
mm		tinyint (unsigned)	3	Minutes field of "timestamp" field.
ss		tinyint (unsigned)	3	Seconds field of "timestamp" field.
http_response_time		int(unsigned)	10	Overall HTTP response time
dns_resolution_time		int(unsigned)	10	Portion of HTTP response time for DNS resolution
tcp_connect_time		int(unsigned)	10	Portion of HTTP response time for TCP connection

http_download_time		int(unsigned)	10	Portion of HTTP response time for HTTP Download
timeout		tinyint(2)	2	1=timeout occurred, 0=no timeout occurred

v_fact_spm_jitter_test_results

Description

This view enumerates historical results that are associated with Service Performance Manager (SPM) Jitter tests.

Columns

Field	Key	Type	Length	Description
test_id	PK	int(unsigned)	11	Combination of Test ID and
timestamp		datetime		Timestamp uniquely identifies a record in this view. Timestamp corresponds with time at which result occurred.
time_id	FK	int(unsigned)	11	Uniquely identifies a day in v_dim_time; join to v_dim_time for more information.
hh		tinyint (unsigned)	4	Hours field of "timestamp" field.
mm		tinyint (unsigned)	4	Minutes field of "timestamp" field.
ss		tinyint (unsigned)	4	Seconds field of "timestamp" field.
response_time		int(unsigned)	10	Latency
src_to_dest_pl		double	53,29	Source to Destination Packet Loss
dest_to_src_pl		double	53,29	Destination to Source Packet Loss
mia		double	53,29	Missing in Action – Packet Loss with Unknown Direction
late_arrival		double	53,29	Late Arrival
busies		double	53,29	Busies
pos_src_to_dest_jitter		int(unsigned)	10	Positive Source to Destination Jitter
neg_src_to_dest_jitter		int(unsigned)	10	Negative Source to Destination Jitter
pos_dest_to_src_jitter		int(unsigned)	10	Positive Destination to Source Jitter

neg_dest_to_src_jitter		int(unsigned)	10	Negative Destination to Source Jitter
timeout		tinyint	2	1=timeout occurred, 0=no timeout occurred

Sample reporting DB queries

The following sample SQL queries are presented to both demonstrate the potential of the 'reporting DB' in DX NetOps Spectrum and serve as a training aid. These queries merely represent a subset of what is possible from a functional perspective.

Log in to MySQL Server

You can log in to the MySQL server using the 'root' user before executing the following sample queries:

On Linux:

```
$SPECROOT/mysql/bin/mysql --defaults-file=../my-spectrum.cnf -uroot -proot reporting
```

On Windows:

```
$SPECROOT/mysql/bin -uroot -proot reporting
```

NOTE
Timestamps used in the below queries are just to fetch the detail belongs to that specific duration. Change the timestamps according to your requirements.

Get the alarm information for a specified time period

You can perform a query to obtain an alarm for a specified time. The result set contains alarm time, alarm name, alarm title, and the alarm message.

```
SELECT `landscape`.`domain_name`,
       `alarminfo`.`alarm_key`,
       `alarminfo`.`landscape_h`,
       `alarminfo`.`condition_id`,
       `alarminfo`.`set_time`
  FROM ((`reporting`.`alarminfo` `alarminfo`
         INNER JOIN `reporting`.`landscape` `landscape`
            ON `alarminfo`.`landscape_h` = `landscape`.`landscape_h`)
        INNER JOIN `reporting`.`model` `model`
           ON `alarminfo`.`model_key` = `model`.`model_key`)
WHERE (`alarminfo`.`set_time` >={ts '2015-01-01 00:00:00'} AND `alarminfo`.`set_time`<{ts '2015-11-16
 14:30:53'})
```

List all the Events for a specified time period

You can perform a query to obtain the list of all events that occurred on a specified time.

```
SELECT `event`.`time`,
       `model`.`model_name`,
```

```
       `model`.`network_address`,
       `creator`.`creator_name`,
       `event`.`type`,
       `event`.`event_key`,
       `event`.`event_msg`
  FROM ((`reporting`.`event` `event`
         INNER JOIN `reporting`.`model` `model`
            ON `event`.`model_key` = `model`.`model_key`)
        INNER JOIN `reporting`.`creator` `creator`
           ON `event`.`creator_id` = `creator`.`creator_ID`)
WHERE (`event`.`time` >={ts '2015-11-19 00:00:00'} AND `event`.`time`<{ts '2015-11-20 00:00:01'})
ORDER BY `event`.`time` DESC
```

Get the alarm activity grouped by a user for a specified time period

You can perform a query to obtain the details of an alarm activity that is grouped by a user for a specified time.

```
SELECT `alarmactivity`.`user`,
       `alarmactivity`.`activity`,
       `alarmactivity`.`time`
  FROM `reporting`.`alarmactivity` `alarmactivity`
WHERE (`alarmactivity`.`time` >={ts '2015-11-18 19:03:40'} AND `alarmactivity`.`time`<{ts '2015-11-19
 19:03:47'}) AND `alarmactivity`.`activity`<>1
```

Get the list of all active device models

You can perform a query to obtain the list of all active device models.

```
SELECT `devicemodel`.`model_name`,
       `devicemodel`.`IP`,
       `modelclass`.`mclass_name`,
       `devicemodel`.`device_type`,
       `devicemodel`.`fw_rev`,
       `devicemodel`.`last_reboot`,
       `devicemodel`.`destroy_time`,
       `devicemodel`.`model_h`,
       `vendor`.`vendor_name`,
       `entity`.`entity_name`,
       `entity`.`entity_id`,
       `devicemodel`.`model_key`
  FROM (((`reporting`.`entity` `entity`
          INNER JOIN `reporting`.`devicemodel` `devicemodel`
             ON `entity`.`current_model_key` = `devicemodel`.`model_key`)
         INNER JOIN `reporting`.`modelclass` `modelclass`
            ON `devicemodel`.`model_class` = `modelclass`.`model_class`)
        INNER JOIN `reporting`.`vendor` `vendor`
           ON `devicemodel`.`vendor` = `vendor`.`vendor`)
WHERE `devicemodel`.`destroy_time` IS NULL
```

Get SLA Summary by SLA name:

You can perform a query to obtain SLA summary by SLA name.

```
SELECT `sm_slaperiods`.`slaStartTime`,
       `sm_slas`.`slaDestroyTime`,
```

```
       `sm_periods`.`periodEnd`,
       `sm_customermhs`.`customerMH`,
       `sm_customermhs`.`custName`,
       `sm_slaperiods`.`slaStatus`,
       `sm_slaperiods`.`slaPeriodID`,
       `sm_slas`.`slaMH`,
       `sm_slas`.`slaName`,
       `sm_slmagreesto`.`customerMH`,
       `sm_slmagreesto`.`agreesToDestroyTime`
  FROM (((((`reporting`.`sm_slas` `sm_slas`
            INNER JOIN `reporting`.`sm_slmagreesto` `sm_slmagreesto`
               ON `sm_slas`.`slaMH` = `sm_slmagreesto`.`slaMH`)
           INNER JOIN `reporting`.`sm_slaperiods` `sm_slaperiods`
              ON `sm_slas`.`slaMH` = `sm_slaperiods`.`slaMH`)
          INNER JOIN `reporting`.`model` `model`
             ON `sm_slas`.`slaMH` = `model`.`model_h`)
         INNER JOIN `reporting`.`sm_periods` `sm_periods`
            ON `sm_slaperiods`.`periodID` = `sm_periods`.`periodID`)
        INNER JOIN `reporting`.`sm_customermhs` `sm_customermhs`
           ON `sm_slmagreesto`.`customerMH` = `sm_customermhs`.`customerMH`)
WHERE     `sm_slas`.`slaDestroyTime` IS NULL
       AND `sm_slaperiods`.`slaStatus` <= 3
       AND `sm_slas`.`slaMH` = 1
       AND `sm_slas`.`slaMH` <> 1
ORDER BY `sm_slaperiods`.`slaStatus` DESC,
         `sm_slaperiods`.`slaPeriodID`,
         `sm_slaperiods`.`slaStartTime` DESC
```

Get VPLS Site health history for a specified time period

You can perform a query to obtain the history of a VPLS Site for a specified time period.

```
SELECT `model`.`model_key`,
       `model_state`.`model_state`,
       `model`.`model_name`,
       `model_state`.`end_time`,
       `model_state`.`start_time`
  FROM (`reporting`.`model` `model`
        INNER JOIN `reporting`.`model_state` `model_state`
           ON `model`.`model_key` = `model_state`.`model_key`)
WHERE ((`model_state`.`start_time` <{ts '2015-11-23 15:53:31'} AND `model_state`.`end_time` IS
 NULL ) OR (`model_state`.`start_time`<{ts '2015-11-23 15:53:31'} AND `model_state`.`end_time`>={ts
 '2015-11-23 08:53:30'})) AND (`model_state`.`model_state`='Down' OR `model_state`.`model_state`='Good' OR
 `model_state`.`model_state`='Maintenance') AND `model`.`model_key`=1
```

Get SLA inventory by SLA name

You can perform a query to obtain SLA inventory by SLA name.

```
SELECT `sm_slas`.`slaName`,
       `sm_guaranteeperiods`.`guarName`,
       `sm_slas`.`slaMH`,
       `sm_guaranteeperiods`.`gPeriodEnd`,
       `sm_guaranteeperiods`.`gPeriodStart`,
       `sm_guaranteeperiods`.`criticalThresholdPercent`,
```

```
        `sm_guaranteeperiods`.`criticalThreshold`,
        `sm_guaranteeperiods`.`typeEnum`,
        `sm_guaranteeperiods`.`motThreshold`,
        `sm_guaranteeperiods`.`mttrThreshold`,
        `sm_guaranteeperiods`.`mtbfThreshold`,
        `sm_slas`.`slaDescription`,
        `sm_guaranteeperiods`.`guarDescription`,
        `sm_guaranteeperiods`.`guarPeriodID`,
        `sm_slas`.`slaDestroyTime`
   FROM (((`reporting`.`sm_slas` `sm_slas`
           INNER JOIN `reporting`.`sm_slaperiods` `sm_slaperiods`
              ON `sm_slas`.`slaMH` = `sm_slaperiods`.`slaMH`)
          INNER JOIN `reporting`.`model` `model`
             ON `sm_slas`.`slaMH` = `model`.`model_h`)
         INNER JOIN `reporting`.`sm_guaranteeperiods` `sm_guaranteeperiods`
            ON `sm_slaperiods`.`slaPeriodID` =
                  `sm_guaranteeperiods`.`slaPeriodID`)
 WHERE     `sm_slas`.`slaDestroyTime` IS NULL
        AND `sm_slas`.`slaMH` = 1
        AND `sm_slas`.`slaMH` <> 1
 ORDER BY `sm_slas`.`slaMH`
```

Service Manager

DX NetOps Spectrum Service Manager is a tool that provides the capability to monitor and manage IT infrastructure that is based on the business services that it provides. Rather than managing a collection of network devices, servers, and applications; you can organize and manage these elements based on how they provide or support specific services. DX NetOps Spectrum service models offer visibility on how the infrastructure elements affect the availability of the business services. This visibility aids in prioritizing infrastructure faults that are based on their impact on business services, and highlighting weaknesses in the environment.

The Service Manager application includes a comprehensive set of tools for creating, managing, and monitoring business services, Service Level Agreements (SLAs), and service customer models in DX NetOps Spectrum. Leveraging DX NetOps Spectrum fault-management capabilities, Service Manager provides real-time and historical insight into the status of your service management components. It also provides a suite of reports for all service management components which can be generated with the DX NetOps Spectrum Report Manager application.

You can manage and monitor Service Manager components in the following interfaces:

- The OneClick Console provides administrative personnel complete access to Service Manager configuration editors and service management models.
- The Service Dashboard provides service providers and customers access to status views of service management models and service outage management tools.
- The Service Level Manager portlet, which can be incorporated into the Unicenter Management Portal (UMP), provides summary status information about services, SLAs, and customers to Unicenter users with secure access to Service Manager models.

Service Manager lets you extend your infrastructure management capabilities beyond the per-device and per-application level. It provides you with the tools to build mechanisms that let IT-service providers and customers validate service availability and performance.

Services

A DX NetOps Spectrum service is a model that represents some logical business service. For example, a router model represents the status of a physical device similarly, a service model represents the status of the business service.

A service model contains a set of resources and a policy indicating the behavior of resources. Service resources are other DX NetOps Spectrum models that collectively provide or support the availability of a business service. A service is only available when its resources are available. In turn, the service model monitors the availability of its resources to determine its health or availability. By applying the collective resource availability to its service policy, the service model can depict the real-time health of the business service it represents.

A service model differs from other types of models in DX NetOps Spectrum. To confirm this information, perform the following actions:

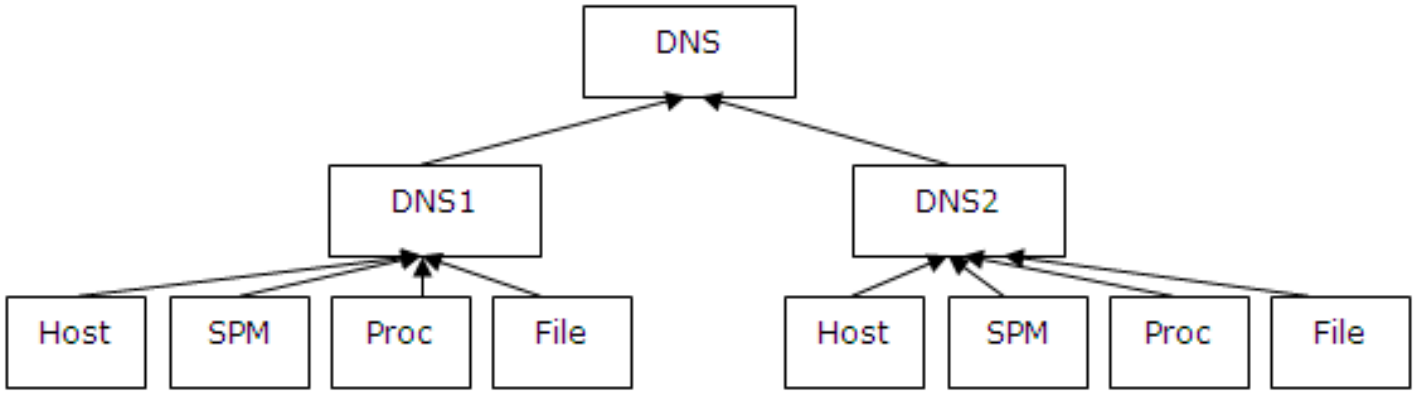
- Consider some of the patterns that are common in IT management. For example, redundant devices, hot swap back ups, load balanced servers, clustered or pooled resources.
- Verify how these patterns are used to support specific business services. One of the oldest and most common business services is DNS. The critical nature of DNS lets you deploy multiple DNS servers. DNS improves performance and also mitigates the risk of DNS failure.
- Represent the real-time status of the DNS service in an environment with two servers that are dedicated to DNS. Monitor the servers themselves as host models in DX NetOps Spectrum. The host model lets you understand the situation if contact is lost to the server and provides information about a number of other potential system faults. You can create SPM tests to validate that each server is responsive. You can also monitor critical processes or file systems on the servers.

Availability of DNS

DNS is available when both servers are functioning. If one of the servers is down, DNS is still available. However, DNS is not available when both the servers are down. The host models, SPM tests, monitored processes, and file systems are all individual models, any of which can fail and produce an alarm. You can view alarms to determine the availability of DNS which can become complex and error-prone with this small set of models.

You can manage this complexity using DX NetOps Spectrum Service Manager. Using service models, you can organize the host models, SPM tests, processes, and file systems to represent the DNS service and express the availability of that service.

The following diagram illustrates the process to implement DNS in DX NetOps Spectrum Service Manager.



Service Policy

In addition to the resources that constitute a service, each service specifies a policy that infers service viability (its service health) based on the collective status of its resources. The role of the service policy is to create an accurate representation of the service health that is based on the health of the resources that comprise the service. The objective is to ensure that the health indicated by the service model in DX NetOps Spectrum reflects the performance of service. Selecting or creating the appropriate service policy is essential to ensure that service health is accurately represented in DX NetOps Spectrum.

A policy specifies a single attribute to monitor for the collective set of service resources. The policy is comprised of two basic components, the Attribute Map and the Rule Set. The Attribute Map associates resource attribute values to service health values. The Rule Set defines the health of the service by giving a set of resource health values.

A service can also include one or more resource monitors instead of a single policy. Each resource monitor can use its own policy extending the monitoring capability of the service.

A number of policies are shipped with DX NetOps Spectrum Service Manager which represent common resource monitoring patterns. If applicable, you can use these policies to familiarize yourself with the creation of custom policies. The use of custom policies can be necessary to verify the accuracy of health that is represented by the service model.

Customers

A DX NetOps Spectrum Customer model or Customer represents any person or an organization that uses services or is a party to a Service Level Agreement (SLA). DX NetOps Spectrum Service Manager lets you associate customers with the services and SLAs to monitor the service management components on a per-customer basis.

SLAs

A DX NetOps Spectrum SLA model, or *SLA*, comprises one or more service *guarantees* that specify the service obligations stipulated in an SLA contract for a particular time period (for example, week, month). Service Manager lets you specify the following two types of guarantees:

- Availability
- Response time.

Both types of guarantees record service outage time and compare it against a user-specified threshold for a period. Availability guarantees also support supplemental thresholds (Mean Time to Repair, Mean Time Between Failure, and Maximum Outage Time).

Create SLAs from scratch or from SLA templates and associate multiple SLAs with a single service and multiple customers with an SLA.

Service Health Values

A *service health* value indicates the viability of a service, whether it is operating at an acceptable or less than acceptable level or is inoperable. Service Health also indicates whether a service is in maintenance mode or in an initial state, if the service has no resources.

The following table lists and describes service health values and corresponding DX NetOps Spectrum alarm states.

Service Health	Description	Icon Color
Up	The service is operating normally.	Green

Down	The service is unavailable. The service is experiencing a critical outage.	Red
Degraded	The service is available but operating at a limited capacity. The service is experiencing a major outage.	Orange
Slightly Degraded	The service is available but operating at a slightly diminished capacity. The service is experiencing a minor outage.	Yellow
Maintenance	The service has been put into maintenance mode and is not actively monitoring resources. The service is experiencing a maintenance outage.	Brown
Loss Of Management	The SpectroSERVER has been shut down. The service is experiencing a loss of management outage.	Gray
Defunct	The service has a configuration error. The service stays in the nonfunctional state until the error has been corrected.	Blue
Initial	The service has no resources associated with it. The service stays in an initial state until resources have been associated with the service.	Blue

Service Management Features

DX NetOps Spectrum Service Manager includes the following features:

- Views that let you monitor the health of services in real time and relate the services to customers affected by IT infrastructure faults.
- Service health records that provide the basis for reports that you can generate on a scheduled or on an on-demand basis using Report Manager.
- SLA violation alarms that notify you when an agreement has been violated or is in danger of being violated.
- Root cause analysis of any service degradation (in terms of infrastructure alarms).
- The capability to designate maintenance periods for services.
- The capability to exempt any service outage from impacting an SLA.
- Extensions to Modeling Gateway that let you create service management components and schedule service maintenance through an XML feed.
- A web page that is added to UMP for viewing service management-related information.

OneClick Licenses and Service Manager Privileges

To access Service Manager from the OneClick Console, OneClick administrator privileges or operator license is required. To access the Service Dashboard, Service Manager license is required. For more information, see the OneClick Administration section.

The following table compares license types and the default roles and privileges that are associated with them:

OneClick License	Default Role	Default Privileges
Operator	OperatorRW	Ability to update service descriptions and view service information and the Service Management hierarchy in the OneClick Console.
Administrator	AdministratorRW	Ability to create and edit services, customers, and SLAs with Service Editor, and create and edit policies, attribute maps, and rule sets with Service Policy Editor.
Service Manager	ServiceManagerRW	Access Service Dashboard.

Service Manager Installation Considerations

WARNING
Service Manager must be installed on both the SpectroSERVER and OneClick servers.

Review the following Service Manager installation considerations:

- The modeling catalog and all modeling intelligence exist within SpectroSERVER.
- The historical database and event handling code exist on the OneClick web server which is installed with OneClick.
- All the client UI components and dashboard are installed with OneClick.
- If you have a separate OneClick installation for Report Manager, install Service Manager on that server to populate the Service and SLA reporting tables.

Plan Service Management Implementation

To benefit fully from the Service Manager capabilities, plan your service management implementation. You can contemplate the following questions and considerations:

- What business services do you want to monitor?
- What particular resources -- processes, software applications, and IT devices -- support those services?
- How can conditions and faults that affect services be detected? Which resource attribute(s) can be monitored to determine the health of a service?
- Who can be notified if a service fails?
- What are the service performance obligations, and how can they be quantified?
- What is the criticality of a given service relative to other services?
- Consider implementing a Service Manager solution as an iterative process. In the initial phase, concentrate on obvious resources and obvious faults. Answer the question: the service does not work, if _____ is down. This information can give you a good foundation for enhancing the service later.
- Identify common sub services or foundation services. If all the multiple services rely on a common set of resources group (resources in their own service), then you can make that service a resource of higher-level services that depend on it.
- Create empty services if you know that a service exists, but you are not familiar with the resources that comprise it. These services left initial are important. These services represent areas of infrastructure that are not familiar, and need to be explicitly monitored.
- Build your services such that they can be easily enhanced. Rather than services which directly monitor resource models such as devices and applications, build services using resource monitors. It is easy to enhance these services in the future by adding new resource monitors without having to change the overall structure of the service.

Service Manager Utilities

You can work with the following utilities to create and manage the Service Manager components:

- **Service Editor**
  Lets you create and manage services, SLAs, SLA guarantees, SLA templates, SLA guarantee templates, customers, and customer groups.
- **Service Policy Editor**
  Lets you create and manage service monitoring policies and the constitute components, attribute maps, and rule sets.
- **Condition Correlation Editor**
  Lets you build correlation domains to monitor resources for specific resource events or combinations of events. These correlation domains can then be used as service resources, extending the monitoring capability of the service from attribute monitoring to event monitoring.

Open the Service Editor

The Service Editor is the primary administrative tool for configuring service models, SLAs, and Customers. The Service Editor displays service models in a flat list or in a hierarchy view. From the Service Editor, you can create new service models or can edit existing service models.

Follow these steps:

1. Select the OneClick Console or the Service Dashboard.
2. Click Tools, Utilities, Service Editor, from the main menu.

   NOTE
   The Service Editor is not available if you do not have access privileges or the OneClick installation does not include the Service Manager product.

   The Service Editor opens.

Open the Service Policy Editor

The Service Policy Editor lists service monitoring policies and policy specifications and includes commands for creating and managing policies. It lets you create and manage policies and their constituent components, attribute maps, and rule sets.

Follow these steps:

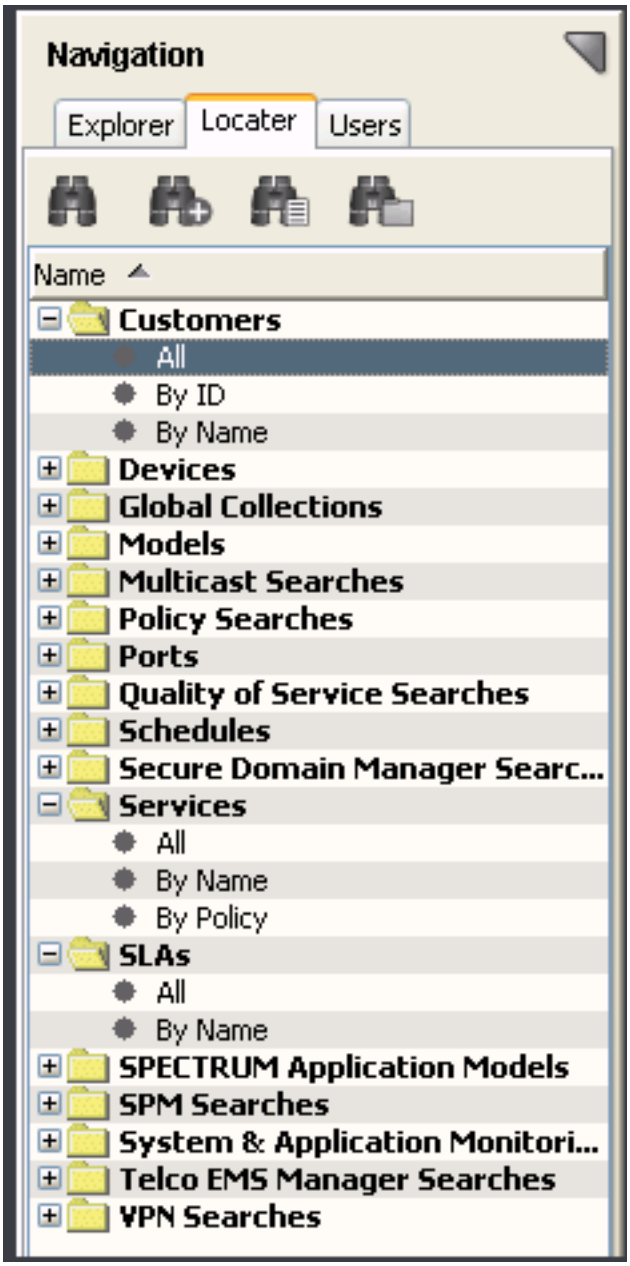
1. Select the OneClick Console or the Service Dashboard.
2. From the main menu, click Tools, Utilities, and Service Policy Editor.

   NOTE
   The Service Policy Editor is not available if you do not have access privileges or the OneClick installation does not include the Service Manager product.

   The Service Policy Editor appears.

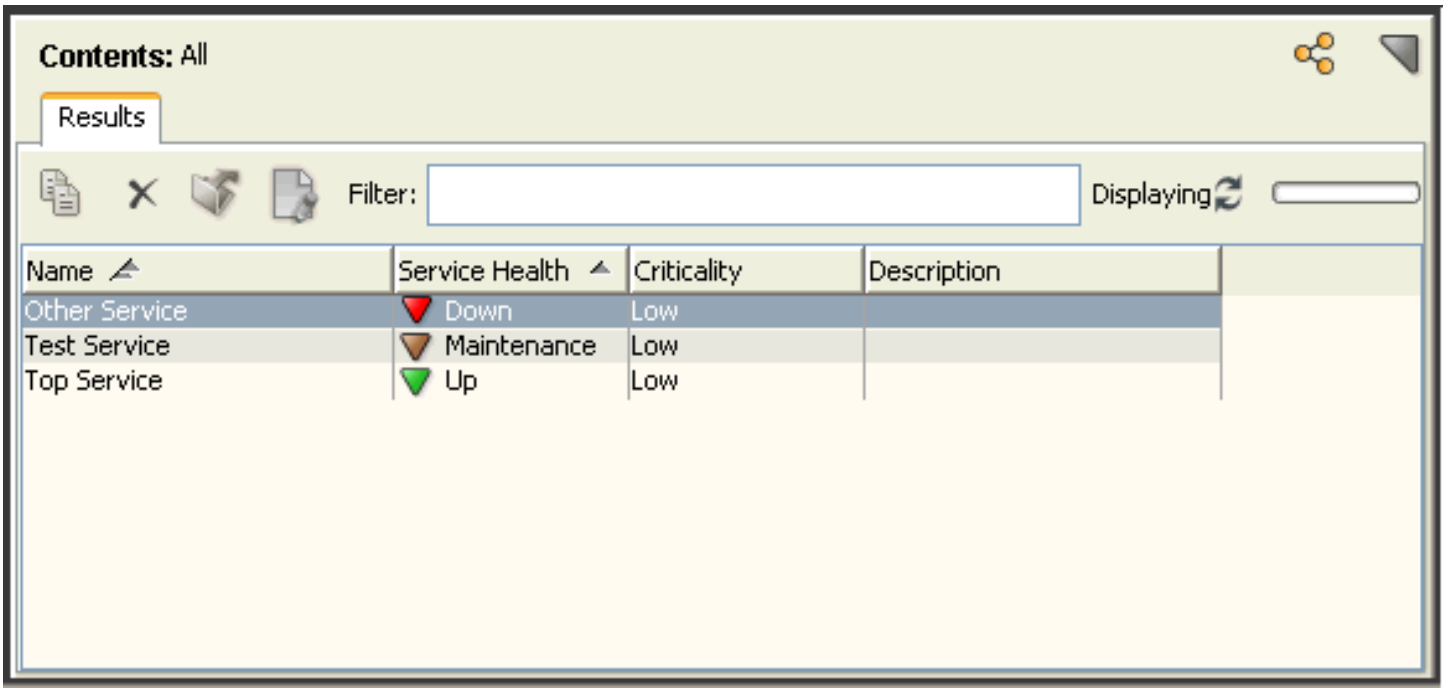
Locating Service Manager Components

You can locate and display services, service customers, and SLAs in the Locater tab of OneClick Console Navigation panel. Search can be initiated from Customers, Services, and SLAs folders, as shown in the following image:

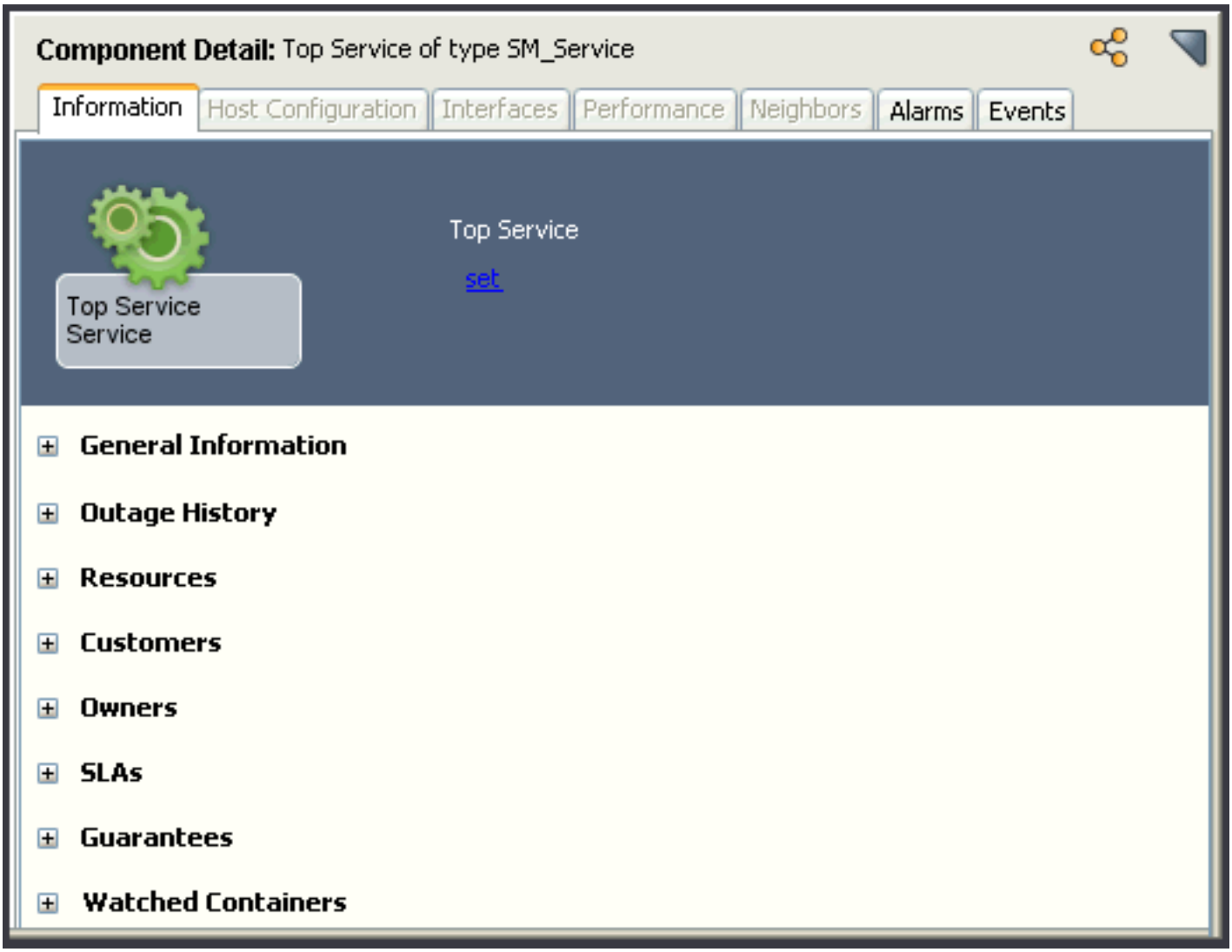


You can track the Service Manager components in a distributed SpectroSERVER environment. For more information, see the Using OneClick section.

The following image displays the search results that appear in the Contents panel of OneClick Console:



The following image displays the details about all aspects of the selected entry in the Component Details panel:



Roll-Up Indications for Service Manager Models

The Service Management tree consists of a set of four models. The root model is the Service Management model, which contains three top-level manager models, such as Service Manager, Customer Manager, and SLA Manager models. These top-level managers organize all of the user created service management components. All service models can be organized under the Service Manager model which is displayed in OneClick with the name Services. Similarly all SLA models and customer models appear under SLAs and Customers.

The Condition of each of these manager models is equivalent to the worst status of all contained models. For example, if the worst service within the landscape has a service health of Degraded, the Service Manager model on that landscape has a Condition of Major. If the worst SLA within the landscape has a SLA status of Violated, the SLA Manager has a Condition of Critical. You can view the Condition of a manager model by the icon color. No alarms are associated to the Condition of the manager models.

Within the Service Dashboard, all landscapes are merged into a single tree in the Explorer panel. Within the service dashboard, a set of folders represent the collective set of manager models from each landscape. Regardless of how many landscapes occur within a DSS the Service Dashboard displays only one folder for each category of manager model.

These folders have the same names as the set of models within the SpectroSERVER: Services, Customers SLAs. The icon for each folder displays the status of the worst top-level model within the DSS. If the worst Service Manager status within the DSS is Minor, the Service Folder in the Service Dashboard displays a yellow icon. Similarly, if a Customer Manager within the DSS has a Condition of critical, the Customers folder in the Service Dashboard displays a red icon.

Service Manager Component Views Outside of the OneClick Console

You can view the Service Manager components outside of the OneClick Console from the following interfaces:

Service Dashboard

The Service Dashboard is a service management only view. It provides visibility to the real-time status for all Services, Customers, and SLAs. The Service Dashboard provides OneClick like console with an Explorer tree, component detail view, and topology view. In addition, the Service Dashboard provides some historical service management views including service outage history, and SLA/Guarantee trending. The dashboard is designed for anyone who is interested in monitoring Service Manager components, and enforces DX NetOps Spectrum model security.

Unicenter Management Portal

CA Unicenter users can view summary information about Service Manager components by adding content from the Service Manager portlet. The Service Manager portlet displays flat lists of service models, customer models, and SLAs models. The Service Manager portlet also provides context launching to the OneClick Console or Service Dashboard.

Service Management Solution Design Guidelines

The key to a successful service management deployment is accuracy in service modeling. The goal when designing services can be to create the most accurate representation possible. Given that objective, often you do not understand how each service functions, and all of the resources on which the service relies. One of the most challenging aspects of building a service management environment is collecting all of the information that is required to model each service. For this task, you require a significant amount of collaboration to create accurate services.

It is rare that a single individual has all the information that is required to build a large service management environment. In most businesses, different teams specialize in different areas of the business each focused purely on their own domain. Ironically it is this typical business structure that creates the need for a service management solution. The more difficult it is to build the service management environment, the greater the need for doing so.

The following section outlines some guidelines for planning and building a service management solution. It takes time to plan and to model a large and complex service management environment. Investing the time to collect information and develop a plan for building the service management solution helps to confirm your success. Part of your plan can include incremental goal, identifying a timeline and milestones for the project. A representation of services for each organization can be included, perhaps a demonstration of how specific infrastructure faults are now processed and prioritized based on service impact. Establishing these milestones lets you realize value throughout the process, and motivates you to expand your solution continually.

One final note on planning is to recognize that you are building a service management solution that must continue to evolve. Your plan may identify a time and point of completion, but that does not mean you will never again have to create or edit a service model. Business infrastructure is continually changing, and you may find that services change too. Your goal is to build a service management solution that is flexible and easy to maintain. On performing this task, you can confirm that your solution can accommodate the dynamic nature of your business.

Service Model Identification and Creation Guidelines

Determining where to start creating your service management environment is the most daunting task. Various techniques such as top down, bottom up, most critical first are available. Regardless of which technique you select, follow the available guidelines.

The first step is to define a phased approach. Develop a multi-phase plan, before you create any service models. The goal of this plan to two folds. A multi-phase plan helps you manage the complexity of creating a large environment and confirms that you receive incremental value throughout the deployment process. It is important that you are able to recognize value from each service that you create and so from each phase of your deployment.

Each environment is different, but the following three-phase deployment plans can work for you, or can be adapted to fit your environment.

Phase I Build a Service Hierarchy Reflecting Your Business Environment

Phase I focuses on building all of the major services, and establishing the service hierarchy which reflects your business environment. This phase produces all of the high-level business-related services, and many of the lower layer services on which they rely. Phase I implementation does not provide a comprehensive solution that encompasses all possible service faults. But creates a hierarchy which shows the services, each organization relies upon and the impact those services have on the overall business.

Throughout this process, identify services with which you are not familiar and for which you cannot determine resources. You can create an empty service. Defining empty services can shed light on areas of the business or infrastructure which are not familiar.

Generally you can start identifying high-level services by listing different organizations within your business environment. If your business is a service provider, this may include different customers and core service components that are shared by those customers. If you are working in an enterprise environment often, you can simply list the different departments within the business. These departments can frequently be considered high-level services. You can also identify dependencies within these high-level services.

Once you have determined the high-level business-specific services, think about what services they rely on. This next tier of services is commonly associated to specific job functions within an organization. These services may be related to specific applications, specific data, and specific security access. Identify various specialized and shared services.

Once you have identified the services upon which various organizations rely, consider the requirements for providing those services. At this level, identify servers for applications or databases. Depending of the nature of these services, recognize specialized service hierarchies and application interdependencies.

You have now defined a broad set of services which are critical to various job functions. This lets you look for shared component services that these job function services require. Regardless of the variety of applications or access areas, you can find many common services that provide the foundation for the specialize services. At this level, you see common networking or domain management services like DNS and DHCP.

Finally, determine the shared network resources upon which all higher-level services rely. Locate core areas of networks that manage traffic and control access. These core services ultimately impact virtually all other services which you have identified.

You have probably collected enough information to start creating service models. For this initial phase of service creation focus on the most obvious resources and obvious faults. The goal of this phase is not to create comprehensive services, but to establish the broad service hierarchy which depicts how the business functions.

It is easy to fall into the trap of trying to define a perfect service that covers all possible resource faults. The problem is that you quickly get mired by complexity, and require assistance from domain experts who understand the intricacies of each service. This process consumes much time, and can stall the process of building the overall service hierarchy.

You can revisit each service in subsequent phases and can refine the resource monitoring capacity to handle the more complex fault scenarios.

As a guideline for determining resources for the first phase, complete the following statement: "The service won't work if _____ is down". The answers to this question are the resources that you can monitor for the first phase. Take care not to dwell on all of the potential resources. Create the service with the obvious resources and move on. Most of your services for phase I have other services as their resources. You can determine that one service relies on another service. Although you do not understand how the other service works, you can create the empty service and move on.

Phase I is complete when you have built a service hierarchy which reflects your business environment. Consult with others and validate your overall design. The hierarchy that you have created serves as the baseline for subsequent phases.

Phase II Add Significant Resource Monitoring Capacity

The goal of phase II is to increase the accuracy of your service models. You can expect the following two things on increasing accuracy:

1. Adding resource monitoring capacity to encompass more potential faults, and eliminating resource monitoring configuration which may produce false service outages.
2. Enhancing your resource monitoring capacity such that you can identify various levels of service degradation instead of simply showing that a service is Up or Down.

Typically in phase II you can revisit the lower layer services which you created in phase I and any empty services that are identified in phase I.

For phase II, you can expand many of the lower layer services by adding new resource monitoring capacity. Refine the resource statement to include the following subtle faults:

- The service won't work if _____ is not available
- The service won't work if _____ is not responding
- The service won't work if _____ is not running
- The service won't work if _____ is not found
- The service won't work if _____ is slow
- The service won't work if _____ is at maximum capacity

Consider the following statements to identify resource faults which degrade service health:

- The service will be slow if _____ is down
- The service may not work if _____ is not available
- The service cannot perform these functions if _____is not accessible
- The service will not be able to handle all requests if _____ is at maximum capacity

In phase II, you can start adding monitoring capability for system resources, performance statistics, and response times. You can refine the existing service resources by identifying more discrete resources, such as the specific interfaces that a service relies instead of devices.

In addition to monitoring, the Condition or status of resource models you can add new resource monitors to evaluate the value of other attributes which express the performance of a model.

Throughout phase II, identify many new service models which encompass subtle, but critical pieces of functionality upon which other services rely.

Look for dependencies between services particularly where an application server relies on another application. Look for areas where network configuration is critical to data processing for security access or quality of service.

Once again take care not to go too deeply into the intricacies of each service. Limit your phase II enhancements to the type of refinements common to many services. For example, regardless of the application, system resources affect availability. Additionally, poor response time between various parts of network impacts many higher-level services.

In particular, do not yet expand resource monitoring to include application-specific detail. This type of monitoring can be addressed in Phase III.

Phase II is complete, when you have added significant resource monitoring capacity. Except for specific application faults, your solution should now be able to process a much wider set of enterprise faults. Similarly, your solution should also be able to report various levels of service degradation.

Phase III Ongoing Refinement

Phase III can be considered the ongoing phase. Regardless of how complete the solution seems to be, there is most likely some potential to refine it further. During phase III, you can extend your resource monitoring capacity to the most complex resource faults. It is difficult to define an end point for phase III, but one guideline is to consider the resolution process. If the resolution process is the same regardless of the fault your monitoring capacity is granular enough. It is tempting to add new resource monitors continually for specific types of faults, but this task is only useful if the course of action differs for each fault.

Depending on your level of expertise, it is the phase where you are most likely to require the collaboration of other domain experts for specific services.

When dealing with application-based services, collect information about the application-specific resources that can be monitored. For example, specific processes, files systems, and log files. Add new services and resource monitors for these various aspects of the application. You can also add monitoring for specific connection ports, and where possible specific transactions.

As part of phase III, add monitoring capacity for configuration changes. It includes monitoring of network configuration policies, and server configuration utilities. Look for configuration changes that can degrade or otherwise impact services.

Whenever possible include monitoring of user access and security mechanism for different applications or networks.

Look for scenarios where a service is impacted even if no specific fault occurs on the service resources. Also look for instances where a service is impacted differently depending upon a combination of resource faults.

You can again refine the following service resource statements to look for additional types of faults:

- The service won't work if the _____ process is not running.
- The service won't work if _____ is restricting access.
- The service won't work if the _____ queue is full.
- The service wont' work if the _____ is archiving.
- The service won't work if _____ doesn't authenticate the user.

Each time that a new resource monitoring capacity to a service you are expanding the set of resource faults the service covers, increases the accuracy of the service.

It is difficult to define when phase III is complete. New resource faults are discovered which were not previously identified. Ideally your design can accommodate these new faults. Each time a new fault is discovered, adapt your resource monitoring such that the next time the fault occurs you can correctly understand its service impact.

Basic Service Definition

Once you have determined what services to model, you can identify several key properties of each service and can characterize the behavior of the service under certain circumstances. To complete this process, perform the following tasks:

Define the Role of the Service

Primarily, identify the role of the service model. The service role encompasses the capability and purpose of the service.

The capability identifies what the service is monitoring. Each service model must represent a capability that can be monitored. This capability can be a tangible process, such as an email service. Alternately, the monitored capability can be more abstract or can represent human resources service. Regardless of the service role, you can state that the service measures the health of some capability.

Once you have identified the capability that is being monitored, define the purpose for monitoring that capability. The purpose is a statement of value that you expect to get from monitoring the capability. For example, many services are intended to provide real-time fault analysis. You can create a service to know the health of the email system. Or a service that shows the impact of resource faults within your infrastructure. Other services are designed to monitor responsibility or compliance. For example, a service that monitors server availability to confirm that it is in compliance with some specified guarantee. In addition, some service models are used to represent organizational dependencies, such as regional offices, depending on resources from a centralized location.

Determining the role of the service helps you select service resources. You can specify service criticality and can define service relationships with customers and SLAs.

Identify Service Resources

Once the service role is defined, you can identify service resources. Service resources provide the capability that the service is monitoring.

For example, consider a service with the role of monitoring the real-time availability of a web-based application. Most likely the application is hosted by one or more servers. These servers can be considered as service resources.

It is important to understand that identifying service resources is an iterative process. It is recommended first to identify the most obvious resources that provide the services capability. In the previous example, the servers hosting the web application are the most obvious resources.

Later, determine the process to monitor the services resources. In many cases, determining the process to monitor one resource reveals other previously unidentified resources. You can manage the service modeling process and can implement your solution in phases. The first phase encompasses the most obvious resources, and higher-level faults. Subsequent phases add additional or more discrete monitoring capability.

Select Resource Models

Once you have identified a service most obvious resources, determine what DX NetOps Spectrum models best represent those resources.

Using the example of a service with the role of monitoring, the real-time availability of a web-based application; you can start with device models that represent the servers. These are likely to be Host models of some sort in DX NetOps Spectrum, or can simply be Pingable models.

Specify the Service Policy

Having located the set of DX NetOps Spectrum models that represent the service resources, determine the best way to monitor these models. This is the process of specifying the service policy that the service applies to its monitored resources.

The first part of specifying the service policy is identifying the model attribute that can be monitored to determine the status of the resource.

Selecting the attribute to monitor has a lot to do with the type of resource that is being monitored. If the model is a Pingable, monitoring its Contact Status attribute can be reliable. If the resource model is a Port model, monitoring its Port Status can be a good choice. Perhaps the resources of the service are other services. In that case, you can monitor the Service Health of the resources models.

For the first phase of service creation, stick to the obvious resources and obvious resource faults. You can identify several attributes that can be used to express the status of a resource model. For the first phase, select the attribute that provides the broadest representation. You can refine the service later by adding more attribute monitoring, or narrowing the specific faults that are associated to an attribute.

Most of the models in DX NetOps Spectrum have the Condition attribute. This attribute is commonly associated to alarms on the model. For example, if a model has a major or orange alarm, the value of its Condition attribute is Major. Condition is often the simplest attribute to use for monitoring a resource. If you cannot identify another attribute that expresses a model's status Condition is probably a good starting point.

> **NOTE**
> It is often easiest when creating services to start by monitoring the Condition of the service resource models. Although this can be a good starting point, the value of Condition is often influenced by many different types of outages some of which may not be appropriate for the service.

Consider resource failure affects on service health

The next part of specifying the service policy is to consider how each individual resource impacts the service. More specifically, consider how the failure of each resource affects the health of the service.

The health of a service model can be expressed by four values, such as Up, Down, Degraded, and Slightly Degraded. Consider each resource that has been identified, and the health of the service should indicate, if that resource fails.

A fault matrix table can be a useful tool to document how various resource failures can impact the health of a service. The fault matrix is a table with columns for each resource, and a column for service health. Here is an example of a simple fault matrix for a pair of servers that support a web-based application. The columns Server 1 and Server 2 contain potential status for each server, the Service Health column contains the logical service health given the status of each Server.

Server 1	Server 2	Service Health
Up	Up	Up
Down	Up	Slightly Degraded
Up	Down	Slightly Degraded
Down	Down	Down

After reviewing the table, you can describe the behavior of the service with the following statements:

- If either Server 1 or Server 2 are Down the Service Health is Slightly Degraded
- If both Server 1 and Server 2 are Down the Service Health is Down

Stated in a resource independent manner, the following two expressions can be identified:

- When any 1 resource is Down the service is Slightly Degraded
- When all resources are Down the service is Down

In terms of a DX NetOps Spectrum service policy, these two expressions can be considered rules, and collectively they make up a rule set. The combination of an attribute and a rule set is the service policy.

By first identifying which resource model attribute represents its status, and next identifying a set of rules which describes how each resource impacts service health you have now specified the service policy.

DX NetOps Spectrum service manager provides a number of policies out-of-the-box, but you can create your own policies whenever an out-of-the-box policy does not match the resource behavior that you have identified.

You have determined the basic structure of the service, and can model it in DX NetOps Spectrum. You can consider this model to be a first-phase model. Chances are throughout this process you identified some areas where the monitoring capacity of the service is not adequate. As mentioned before service modeling is an iterative process, each phase expands, or refines the monitoring capacity of the service to make it more accurate.

The next section refining the service definition covers some of the common issues you will find when first defining a service, and provides some techniques for how to improve your service models.

Refining the Service Definition

Regardless of the role that is defined for a service model, your goal can be to create a service such that the health of the service model accurately depicts the logical status of the service. You can eliminate scenarios where the service is logically down, but the service model indicates that it is up. Similarly, a service model cannot indicate that its health is impacted if logically the service is functioning normal. The following section introduces some techniques for refining your service models. These techniques are best applied in phases. Do not make too many enhancements at once. Define a strategy to improve your services, and implement that strategy. Break down the service revision process into multiple phases that can be managed and validated.

You can find various reasons for inaccuracy of service models monitoring capacity. This section covers a few of these reasons and discuss ways to improve your service models.

The following reasons are potential scenarios that can affect the accuracy of a service model:

Missing Resource Models

The most common issue when building service models is failure to identify all of a service resource. The principle reason for this is that most of the time we identify the resources that are providing the service directly to an end user, and not the capabilities that those resources rely on.

Continuing with the example of a web-based application, often when focusing on user faces applications, we start with the servers which host the application. This is a great starting point, but you may find that you have missed servers that support the application. Consider how the web application relies on a database to provide content. The following table displays the changes of our fault matrix:

Database	Server 1	Server 2	Service Health
Up	Up	Up	Up
Down	Up	Up	Down
Down	Down	Up	Down
Down	Up	Down	Down
Down	Down	Down	Down
Up	Down	Up	Slightly Degraded
Up	Up	Down	Slightly Degraded
Up	Down	Down	Down

The table is a bit more complicated, but you can see that with the addition of the Database, the previously identified rule set no longer works. You can see that regardless of the status of Server 1 and Server 2, if the Database is Down, the service is Down.

This type of pattern is common, and you have identified that the service relies on more than the web servers it requires extra resource which behaves differently than the web servers. To support this new set of resources, use a new model type named Resource Monitor.

The job of a resource monitor is to manage diverse sets of resources which do not follow a behavior pattern that can easily be captured in a single policy. A resource monitor is similar to a service in that it applies a policy to a set of resource values to determine its own health. It can be useful to think of resource monitors as a type of sub-service. A resource monitor by itself does not represent a logical service, but rather a critical aspect of that service.

In this example, you can see how resource monitors can be used. The following behavior is captured for the web servers and their relationship to the health of the service.

Server 1	Server 2	Service Health
Up	Up	Up
Down	Up	Slightly Degraded
Up	Down	Slightly Degraded
Down	Down	Down

If we specifically consider how the database impacts the service that a simple matrix can be created.

Database	Service Health
Up	Up
Down	Down

Capture each of these patterns into its own resource monitor.

The Database RM monitors the database resource with the following rule:

- When all resources are down the service is down

The Webserver RM monitors the web servers with this rule set:

- When any 1 resource is Down the service is Slightly Degraded
- When all resources are Down the service is Down

The service monitors the two resource monitors, and the service health reflects the status of the worst resource monitor. The following table show the fault matrix for the service:

Database RM	Webserver RM	Service Health
Up	Up	Up
Down	Up	Down
Down	Slightly Degraded	Down
Down	Down	Down
Up	Down	Down
Up	Slightly Degraded	Slightly Degraded

From this matrix you can see the following rule set for the service:

- When any 1 resource is Down the service is Down
- When any 1 resource is Slightly Degraded the service is Slightly Degraded

By expanding the service from monitoring the condition of two host models to instead monitoring the health of two resource monitors we have improved the accuracy of the service model. You can notice that the database resource is never referred to as a host model or a database server. Because the database is likely to be a service itself.

This scenario of missing resources is common that you can consider it when creating service models. Even if the database server was not identified as a service resource in the initial phase the service could still have been created to monitor a single resource monitor for the web servers. In later phases, as additional resources are identified, it is easy to add new resource monitors to the service itself.

Resource Models Which Are Not Discrete Enough

Missing service resources is certainly one common cause of inaccurate service model. Another cause for inaccuracy in the services model is resource models that are not discrete enough. Usually, DX NetOps Spectrum monitors devices at a fairly high level, and reports their condition that is based on fairly basic criteria. For example, a host model that responds to SNMP traffic and has normal CPU and memory utilization can be considered Up or Normal.

Going back to the web application service example, consider if the simple host monitoring is adequate to determine if the web application is working. Imagine, if on webserver 1, a critical file system has become full, and on webserver to the actual webserver process was not running. In this case, the web application service is not available to users, but the given basic host monitoring the service model indicates a health of Up.

In addition to the host being contactable, various other aspects can be monitored to determine if the web application is running. To determine the application status, you can use the following common components:

- Monitoring critical processes
- Monitoring critical file systems
- monitoring application connectivity and response time

Looking at each web server, you can define a fault matrix using the following example:

Host	Process	File System	App Connection	Service Health
Up	Up	Up	Up	Up
Down	Down	Down	Down	Down
Up	Down	Up	Down	Down
Up	Up	Down	Up	Down
Up	Up	Up	Down	Down

In the preceding abbreviate matrix, you can notice that for the webserver to be considered up, consider more than simply a host model. A webserver process, a critical file system, and the web application can also be responsive to connections and requests.

You can notice that rather than simply a host model each webserver is actually a service in and of itself. From the preceding example, you can envision a service with three resource monitors.

The Host RM, simply monitors the Condition of the host model. The Proc and FS RM monitors the Condition of a process model, and a file system model. The App Conn RM monitors the status of a series of response time tests that send requests to the server.

At first, given that the Host RM and the Proc and FS RM are both monitoring the Condition attribute, you can combine them into a single resource monitor. You can separate the resource models into two resource monitors as they represent different classes of models. In the next section, you can notice, isolating the host model within its own resource monitor that gives you the ability to exclude host-related alarms (that do not impact the service).

The Host RM and Proc and FS RM models have simple fault matrix tables.

For the Host RM model:

Host	Service Health
Up	Up
Down	Down

For the Proc and FS RM Model:

Process	File System	Service Health
Up	Up	Up
Down	Up	Down
Up	Down	Down
Down	Down	Down

The App Conn RM is based on response time monitoring. It is at your discretion to use the multi-level threshold capability of DX NetOps Spectrum service performance manager to create more health values. For simplicity, look at a Timeout, Critical Threshold, and Major Threshold configuration. The following matrix looks like a condensed set of response time test values:

Response Time	Service Health
Normal	Up
Timeout	Down
Critical Threshold	Down
Major Threshold	Degraded

Consider the following fault matrix for a single webserver:

Host RM	Proc and FS RM	App Conn RM	Service Health
Up	Up	Up	Up
Down	Down	Down	Down
Up	Down	Down	Down
Up	Up	Down	Down
Up	Up	Degraded	Degraded

You can determine that all resource monitors must be Up for the webserver to be considered Up. Remember, this is the fault behavior for a single web server.

In the review, we earlier expanded the web application service to include two resource monitors: Database RM, and Webserver RM. At the time, the webserver RM was monitoring the Condition of two host models. From the preceding matrix, you can identify that rather than two host models, the webservers can be represented as two service models each with three resource monitors. The previous configuration still applies, but now the webserver RM monitors the service health attribute of two services models, instead of the Condition of two hosts. You can see that despite vastly improving the accuracy of the web application service the structure of the service remains intact. The fault matrix that is determined early is still accurate:

Database RM	Webserver RM	Service Health
Up	Up	Up
Down	Up	Down

Down	Slightly Degraded	Down
Down	Down	Down
Up	Down	Down
Up	Slightly Degraded	Slightly Degraded

What has been improved is the definition of Up for the webserver RM.

Resource Faults that Should Not Impact the Service

Missing resources and resources that are not discrete enough can often lead to a service indicating a health of Up when it should not. Sometimes a service model may indicate a health of Down or Degraded when logically it is not. This can best be described as a non-service impacting resource fault.

Two common situations are available that produce this type of problem, both of which are different.

The first situation is when a resource model status is influenced by associated child models. This situation can happen, if you select a model that is logically a service resource but has a status that can be affected by other models (that are not service resources). For example, when you specify a network device as a resource for a particular service, rather than the interfaces that actually support the service. Another example, when dealing with host models or servers to specify that the host itself alarms for a failed process or file system instead of the monitored process or file system model. If the process or file system is not logically a part of the particular service, the service may indicate an affected health when it should not. Such situations are easy to resolve by selecting the most appropriate model as the service resource, for example, a specific interface, or monitored process.

The second situation occurs exclusively when a service monitors the Condition of its resource models. The Condition of a resource model holds the value of the most severe alarm on the model. DX NetOps Spectrum generates thousands of different types of alarms. Some of these alarms are indicative of service impacting faults, but many are not. By default, DX NetOps Spectrum monitors devices for various reasons, but principally to ensure that are functioning correctly from a network infrastructure perspective. You can see various alarms which faults have to do with network management, but these alarms may have no logical impact on the capability that is being monitored by the service. Similarly, a given model can be a resource of multiple services, such that a particular alarm can be significant for some services and not others. This occurs frequently when the purpose of the service is to monitor compliance that is based on some specific responsibility.

Service Manager alarm type exemption functionality is designed to support resource alarms which should not impact service health. When you discover that certain resource faults should not impact a service, you can specify an alarm type exemption configuration. This configuration can be specified for an individual service or resource monitor, or if the behavior is common it can be specified in a service policy.

By applying alarm type exemptions, you can again improve the accuracy of your service models by eliminating false service outages.

Service impact of non-alarm producing Events

Sometimes, you can encounter a situation where a resource model experiences an event that does not produce an alarm on the resource. However, it logically affects the capability being monitored by the service. When deployed with a primary focus on network infrastructure management, often many events that are produced by or passed to DX NetOps Spectrum which do not produce alarms. Because the events are too common or insignificant. That said there are circumstances where these events are ignored and can be considered for a specific service. In some situations, it can be appropriate for you to map the event to a new alarm, which affects the Condition of a resource model. As the alarm configuration takes effect for all models of a given type, it may not be a good practice if the event is significant only for a few models.

When the situation arises that an event is service impacting, but does not produce an alarm, you can consider creating a Correlation Domain to represent the service resource. Much like a service determines its own health by monitoring attributes of its resources, a correlation domain can determine its own condition by monitoring events occurring on its resources.

To support non-alarm producing events, create a Correlation Condition for the event. You can perform this task by specifying the event code for the logical set, and then the corresponding event code for the logical clear.

Revisit the Database service that discussed early in this section, for example, the Database service is comprised of two host models. At some time during the day, each database server can initiate an automatic data archival process. Consider that DX NetOps Spectrum was integrated with a database management tool which can produce an event on the DX NetOps Spectrum host model when the archival process begins and another event when the archival process is complete. Since this behavior is normal for the server, there is no alarm. However, during the archival process, the server may not respond as quickly to requests.

In general, the data archival process does not impact the database service as a whole. However, if both database servers were archiving data simultaneously, the service becomes Slightly Degraded. If one of the database servers is down, and the other is archiving data, the service becomes Degraded.

You can create a Correlation Condition and a pair of Correlation Rules to capture this behavior.

The Correlation Condition consists of a set event code and clear event code which correspond to the start and completion of the archival process.

Two new Correlation rules can be created which specify an Implied Cause of either Service Impact Slightly Degraded or Service Impact Degraded. Service Manager adds these out-of-the-box Conditions with a Service Impact Down Condition. Consider the following correlation rules:

- DB Archiving Exists AND Device Contact Lost Exists Implied Cause Service Impact Degraded
- DB Archiving Count = 2 Implied Cause Service Impact Slightly Degraded

Both rules specify the Correlation Domain as the root cause target. The new rules can be combined into a new Correlation Policy that is named Data Archival Policy. You can create a Correlation Domain using the Data Archival Policy, and can specify both database servers as resources.

Finally, to extend the Database service, a new resource monitor, Data Archival RM can be created to monitor the Condition of the correlation domain.

By using this approach, you are able to show the service impact of non-alarm producing events.

Patterns of Resource Faults Which Impact Services

You can encounter resource monitoring scenarios where the relationship of particular service resources is too complex to capture simply by monitoring the Condition attribute of the models. More specifically, the Condition of the model can have a greater or lesser significance depending on some additional resource monitoring criteria.

Consider the following scenario. An account management team working in a remote office uses a local database server to access customer information. If the local system is down, the account management team can access the information that they require by connecting to a server at the company head quarters. The service representing the customer account system behaves in the following manner:

- If the local server is down, and the head quarters server is up, the customer account service can be Slightly Degraded
- If the local server is up and the head quarters server is down, the customer account service can be Up.
- If the local server is down and the head quarters server is down, the customer account service can be Down
- If the local server is down and the head quarters server is in maintenance, the customer account service can be Down
- If the local server is in maintenance and the head quarters server is in maintenance, the customer account service can be Down

As you can see, the service impacting scenarios are fairly complex. Even though there are two servers providing the service, the servers are not treated equally.

For handling a resource monitoring scenario, such as Correlation Domain can be used that manages the complexities of monitoring the individual resources.

A service utilizing two resource monitors can be used to implement this example. First, you can isolate the local server into its own resource monitor, which uses a simple fault matrix. The Local Server RM uses a policy to support this pattern.

Local Server	Service Health
Up	Up
Down	Slightly Degraded

The second resource monitor that is Local and Remote Domain RM also has a simple fault matrix:

Local & Remote Domain	Service Health
Up	Up
Down	Down

The Local and Remote Domain RM, monitors a Correlation Domain which contains the host model for the local server and the host model for the head quarters server.

Verify the following status table of the Condition of the Correlation Domain:

Local Server	Remote Server	Domain Condition
Normal	Normal	Normal
Contact Lost	Normal	Normal
Normal	Contact Lost	Normal
Contact Lost	Contact Lost	Critical
Contact Lost	Maintenance	Critical
Maintenance	Maintenance	Critical
Maintenance	Contact Lost	Critical

The domain uses a policy with the following rules, all utilizing the domain for the root cause target.

- Device Contact Lost Count = 2 Implied Cause Service Impact Down
- DeviceInMaintenance Count = 2 Implied Cause Service Impact Down
- Device Contact Lost Exists AND DeviceInMaintenance Exists AND Device Contact Lost Model Does Not Equal DeviceInMaintenance Model Implied Cause Service Impact Down

The Customer Account Service monitors the service health of the two resource monitors in the following way:

Local Server RM	Local and Remote Domain	Server Health
Up	Up	Up
Slightly Degraded	Up	Slightly Degraded
Slightly Degraded	Down	Down
Up	Down	Down

The Customer Account Service bases its health on the worst status of the two resource monitors which is the common high sensitivity pattern.

Using this approach, you can create accurate resource monitoring capacity for your services even when the scenarios have complex requirements.

Service Attributes and Relationships

When creating a service, specify the number of attributes and associate the service to its resources and other service management models. This section explains the attributes that can be configured and the potential relationships that you can create for each service model.

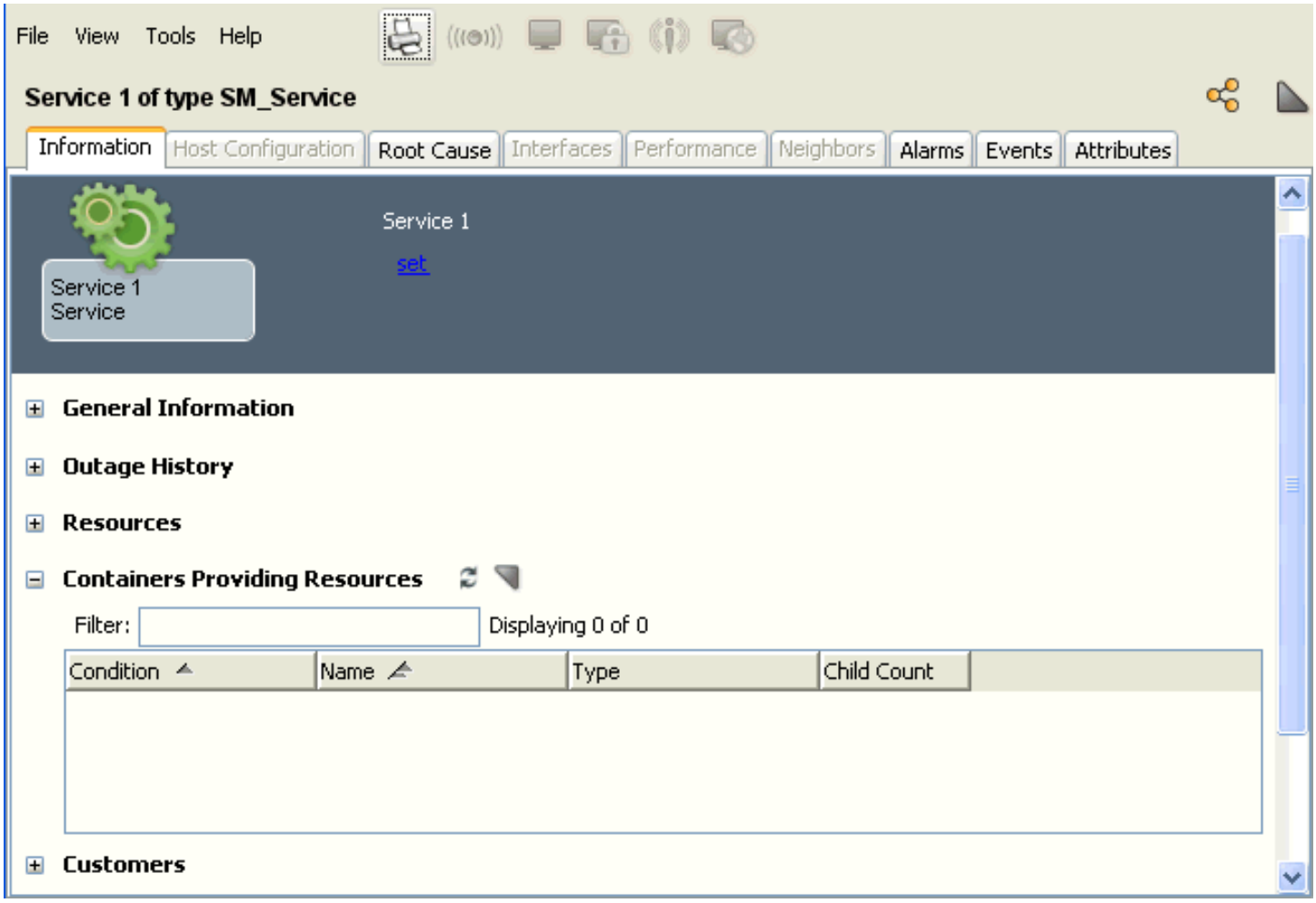
- **Name and Description**
  The service name and description identify the service model. You can define multiple services with the same name, provided they have unique descriptions.
  You do not have rules for defining service names. However, you are recommended to use some naming scheme or convention that lets you identify a service model, if you encounter it outside of the service management hierarchy. Some naming schemes include multiple parts that let you categorize the service geographically or organizationally. Other naming schemes associate the service to a particular customer or function.
  It is a useful practice to state the service role in the service description. The service role is determined in the planning phase, and states the capability and purpose of the service model.
- **Criticality**
  Service criticality is an enumerated value ranging from a Low value of 10 to a High value of 30. All or a portion of a service's criticality can be added to the Impact value of any resource alarms affecting the service.
  If a service is down, the entire value is factored into the calculation. If a service is degraded as a result of the resource alarm, one half of the service criticality value is factored into the impact calculation for the alarm. If a service is slightly degraded, one-fifth of the value is factored into the calculation.
  Verify the following Criticality values:
  - Low (default) = 10
  - Medium Low = 15
  - Medium = 20
  - Medium High = 25
  - High = 30

  For example, a Degraded High criticality service has an impact of (50percent) * 30 = 15, and a Down Low criticality service has an impact of (100percent) * 10 = 10. The alarm that caused the Degraded High criticality service has a comparatively greater impact than the alarm the caused the Down Low criticality service. If alarms are sorted and prioritized by impact value, the alarms impacting the most critical services are given the highest priority.
- **Landscape**
  The landscape field specifies the landscape where the service model resides. The landscape option appears only when multiple SpectroSERVERs are deployed in a distributed environment.
  To optimize performance, the service can be created on the landscape where all or most of its resources reside. If the service has resources spanning multiple landscapes review Service Models in a DSS environment.
- **Security String**
  The security string secures access to the service model in DX NetOps Spectrum. For more information, see the OneClick Administration section.

  NOTE
  Security in the service dashboard differs from the OneClick Console. If a user does not have access to a service model, all icons and list entries for that service are absent from the service dashboard. This dashboard differs from the OneClick console where icons are exposed, but no model data is available.
- **Maintenance Mode**
  Service models support maintenance mode as do many other models in DX NetOps Spectrum. When a service is "in maintenance", it is not actively monitoring any of it resources.

You can create a service in maintenance to avoid the generation of any service outages while you are still building your service hierarchy and identifying resources. In this capacity maintenance, the model is used to show that the service is under construction.
Service models also support scheduled maintenance. Schedule maintenance defines preconfigured periods of time where the service stops actively monitoring its resources. Commonly service level agreements specify periods of time which are reserved for service maintenance.

- **Generate Service Alarms**
  Each service can be configured to generate alarms that correspond to changes in service health.
  Disabling the generation of alarms for the service means that an alarm cannot be generated, the service health is still modified based on policy evaluation. All icons within the OneClick Console and Service Dashboard shows the appropriate color for service health regardless of whether an alarm is generated. Regardless of the generate service alarms setting, all or a portion of the service's criticality are added to any resource alarms impacting the health of the service. The service is shown in the service impact table of such resource alarms, even if no alarm is generated for the service model itself. Any guarantee models associated with the service tracks outage time regardless of whether an alarm is generated for the outage.
  There are several reasons to disable the generation of service alarms. The first is to reduce the number of alarms that are produced in DX NetOps Spectrum. If you are sure that all resource alarms indicate the service impact, then the service alarm can be unnecessary.
  Another reason to disable service alarms is when the alarm is redundant. Often multiple services are created which monitor many of the same resources, but with a different role. For example, you can create a service model that focuses on the real-time status of a service and can have it generate alarms. Other services models monitor specific aspects or resources of the service model for SLA purposes and can be configured such that they do not produce an alarm.
- **Containers**
  The containers setting specifies how a service monitors its resources, if the specified resource is a type of container model. This setting applies only to those resources which are containers and is not used for non-container resources. You can add different types of containers to a service. Depending on the type of container, you can configure the service to monitor the container model itself or the contents of the container. When set to Monitor Contents (default), the service applies the policy to the models within the container. When using Monitor Container, the service applies the policy to the container itself.
  When Monitor Contents is specified the service monitors the containment relations of the container model, and updates its resources as models are added or removed from the container.
  Consider, for example, the effect of physically removing a router and replacing it with a new router during a network upgrade. If the original router model was placed in a container model that is monitored by Service Manager, it automatically removes from the service. If you place the new router in the same container, it automatically monitors as part of the service.
  Many environments tend to be dynamic (the service resources can periodically change). Consider the addition of new infrastructure components that increase capacity and mitigate the risk of failure. As these resources are added services, you can take them into account. Structured containers and services can adapt easily to these types of changes.
  You can view a list of containers in a service under Containers Providing Resources in the OneClick Console or Service Dashboard Information view. Verify the following image:



The Containers Providing Resources subview indicates the container condition, name, type, and child count (the number of resources in the container).

The Containers setting applies to all resources which are derived from the Container model type. If you want different behavior for different containers within the same service, consider the use of multiple resource monitors, which each has its own container monitoring setting.

- **Service Policy**
  During the planning phase, you identified the resource attribute to monitor and the rule set by which to determine service health. You can select an existing policy that matches this behavior or can create a new one.
  The policy should reflect the behavior of the resources that the service monitors. You can select the policy first by which resource attribute can be monitored, and next by which policy best reflects the behavior of the resources collectively. For example, if the service monitors a pair of redundant servers, a redundancy type policy would be appropriate. If you find no single policy accurately reflects the behavior of the resource, you can create multiple resource monitors to organize resources that are based on their specific monitoring requirements.
  Service Manager provides a set of standard policies that represent common service monitoring requirements. It also lets you create custom policies to meet your particular requirements.

If the service uses resource monitors, or have other services as its resources only Service Health based policies can be used.

Create a Service

Have an understanding of service and the resources that provide the service, before creating a service. Consider how the resources can be monitored and the relative impact of resource outages on the availability of the service. Identifying the resources and how they impact the service makes policy and resource selection easy.

Services that you create appear under the Service tab in Service Editor, in Service Dashboard, and in OneClick Console under Service Management in the Navigation panel.

Follow these steps:

1. Open the Service Editor.
2. Click Tools, Utilities, Service Editor from the main menu.
3. Click the Services tab.
   At the bottom of the Services table you can see the following options/tabs:
   - List - Creates a service as a child of the top-level services model
   - Hierarchy - Creates a service as a child of the service which is selected when the Create button is clicked

   If you are going to create a service as a child or resource of another service the parent service must use a Service Health based policy.
4. Click Create.
   The Create Service dialog appears. The Create Service dialog lets you specify service properties, the IT resources that support the service, and the service policy or resource monitors that define which resource attributes are monitored.
5. Specify the following service properties, some are required for service creation other are optional. Most required fields have default values, but always consider if these values are appropriate.
   - **Name (Required)**
     Multiple services can have the same name provided they have unique descriptions. Consider using a naming scheme or convention that allows for quick identification of the service model.
   - **Description (Optional)**
     Describes the service. You can enter unique descriptions for services that have the same name to facilitate finding each service using a list filtering utility. The service description appears within Service Availability and Service Health Reports.
   - **Criticality (Required)**
     Specifies the criticality value that is factored into the impact calculation for a service resource model in an alarm state (the root cause alarm).
   - **Landscape (Required)**
     Specifies the landscape where the service model can be created. The landscape field appears only when you are working in a DSS environment.
   - **Security String (Optional)**Specifies the security string for the service model.
   - **In Maintenance (Optional)**
     Selecting this option puts the service model into maintenance mode.
   - **Generate Service Alarms (Required)**
     Specifies whether DX NetOps Spectrum generates alarms for the service model which correspond to changes in service health.
   - **Containers (Required)**
     Specify how the service should monitor resources which are containers.
6. Specify the service policy and click the Set button to select or create a service policy.
   For services that define resource monitors or monitor of have other services as their resource only Service Health based policies can be used.
   Note: When you specify Service Policy, the Service Editor automatically selects the most appropriate table to display the resources of the service.

Note: The service policy dictates which attribute of the resources can be monitored. Make sure the service resources that you specify support the attribute that you select. If a resource is specified which does not support the attribute, DX NetOps Spectrum generates a yellow (minor) alarm for the Service Management model.

7. If the service uses resource monitors, click the gear icon to create each resource monitor.
   For each resource monitor, specify a name and policy. You can also specify the Container behavior, and an alarm type exemption.
8. Configure alarm type exemptions.
   This feature is only available for services that are using a Condition-based Policy.
9. Select containers and resources for a service by clicking the binoculars icon to launch the resource locater; search for resources and add them to the service.
   If you have created resource monitors first select the resource monitor, and then launch the resource locater.
   When specifying resources which are container mode, the Containers configuration that is specified in step 5 applies.
   For more information about working with searches, see the OneClick Administration section.
   If you are creating the service on the main location server, select resources from any landscape. If the service is created on a landscape which is not the MLS, you are restricted to select resources only from the local landscape, and the MLS. Notify your organization DX NetOps Spectrum administrator if the search does not find the resources that you expect to find.
10. Select the resources (models) you want to include from the returned search list and click Add Selected to Monitored Resources. Close the Locate Resources dialog when you have all the resources.
11. Click the Create button to create a service model.
    The service now appears in the table view of the service editor.

Resource Monitors

Resources in Maintenance Mode

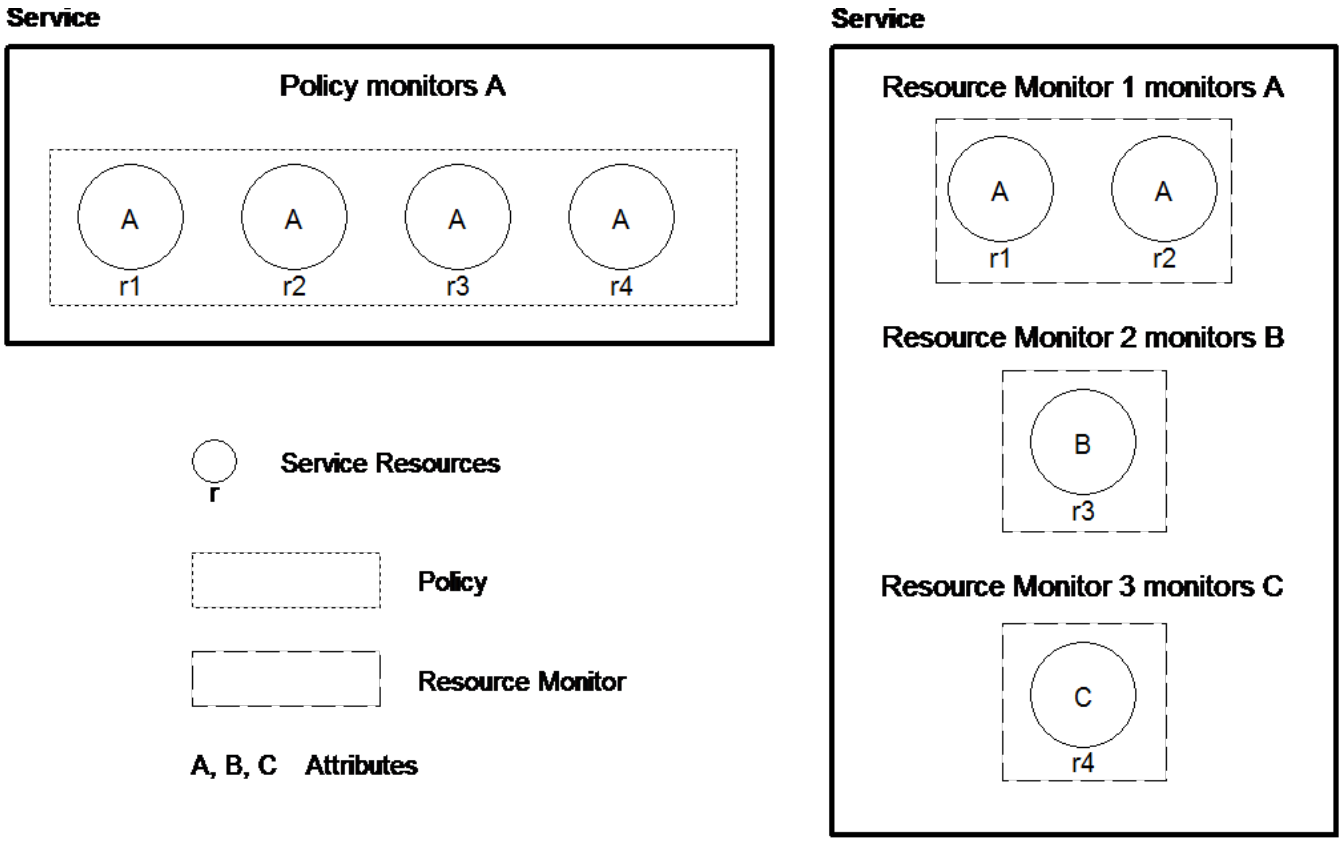
When a service resource is put into maintenance mode, Service Manager stops monitoring the resource during the maintenance mode period. For certain types of resources (port models, monitored process models, and monitored file system models, for example), Service Manager respects the maintenance status of the parent device model.

When the parent model for the following resource types is put into maintenance mode, the service stops monitoring the resource:

- interface models
- monitored process models
- monitored disk (file system) models

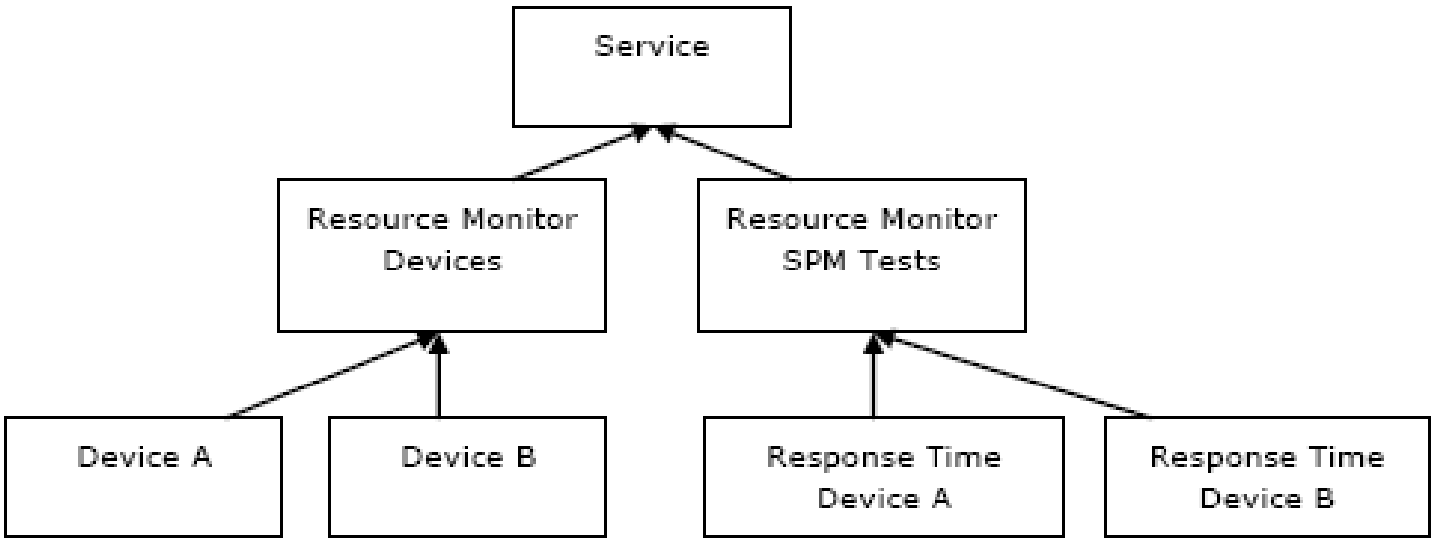
A resource monitor is a DX NetOps Spectrum model. A policy, which is not a model, specifies a *single* watched attribute common to *all* resources monitored by a service, a resource monitor monitors an *attribute* common to *one or more* service resources to determine its own service health status, the same way a service determines its health status. Using resource monitors lets you implement a finer mode of monitoring service health than is provided by a single policy.

The following diagram illustrates the difference between the two by showing how each could be implemented for the same service:



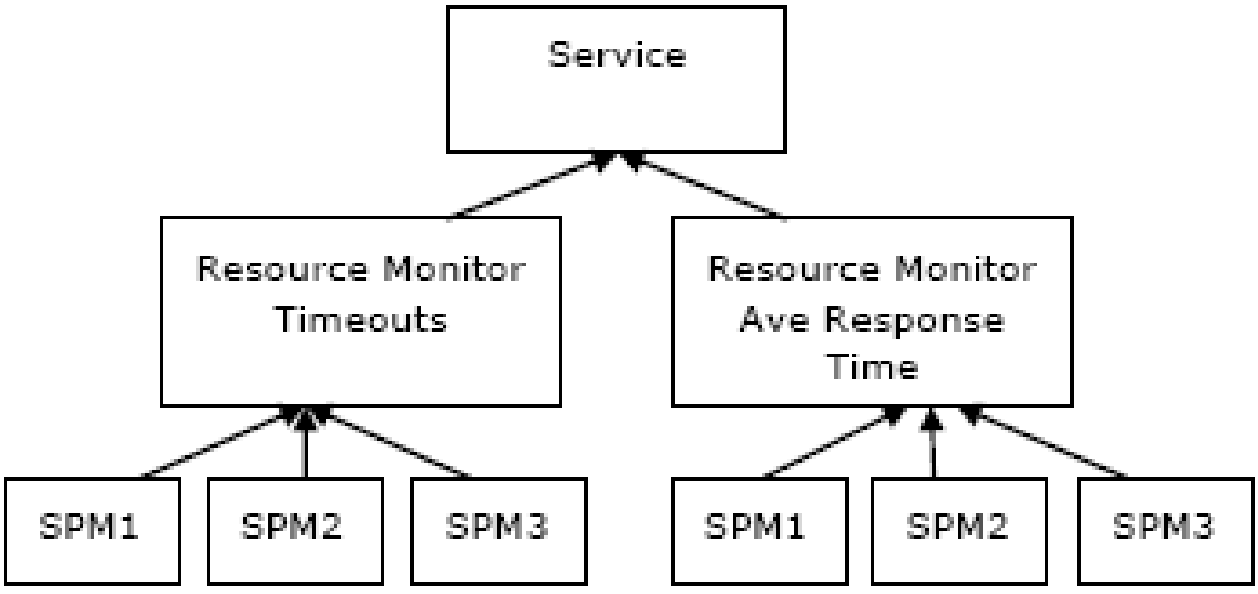
A resource monitor is a resource of a service, and you can apply multiple resource monitors to monitor a service. A service monitors the service health attribute of each resource monitor to determine its own health.

A simple scenario of when resource monitors can be used is when the resources of a service are of mixed types. For example, when monitoring a pair of device models, and a pair of response time test models. Different policies are used to monitor device models, and SPM test models, resource monitors could be created to organize the device models, and the SPM test models in the following way:



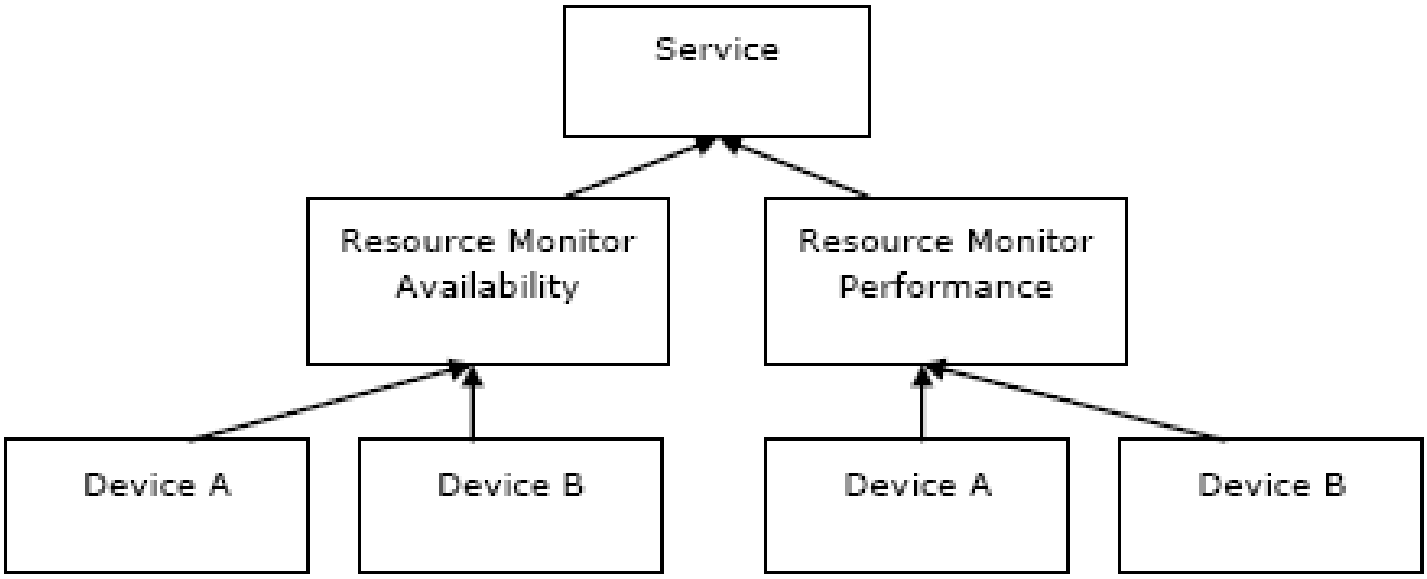
The Service model would use a policy such as Service Health High Sensitivity, the Resource Monitor for Devices uses the policy Condition Redundancy, and the Resource Monitor for Response Time tests uses Response Time Redundancy.

In other cases, you can monitor multiple attributes of the same resources. In other words, apply multiple policies to the same resources. For example, you may be interested in monitoring a set of SPM Tests with a response time policy such that the service is impacted when too many response time tests are timing out. You also want to monitor the average response time of the same response time tests such that the service is impacted when the average response time exceeds some specific threshold. Again resource monitors can be used to create this type of service.



In this scenario, the service uses the Service Health High Sensitivity policy. The Resource Monitor for timeouts could use a custom policy focusing specifically on the Latest Error Status value of Timeout for the SPM Tests. The Resource Monitor for average response time uses another custom policy that compares the average Latest Result for the SPM tests to a particular set of threshold values.

In yet another scenario, multiple resource monitors can monitor the Condition of common resources, but can use different policies to organize resource outages by fault type. For example, resource monitors could be used to categorize availability impacting faults and performance impacting faults:



In this configuration, the Service uses a service health policy, the availability resource monitor uses a customer policy with an Alarm Type exemption configuration designed to isolate resource faults that are designated as availability impacting. The performance resource monitor would in turn use of custom policy which isolates resource faults that are designated as performance impacting.

Create a Resource Monitor

You can monitor multiple attributes on a single resource with different resource monitors.

Follow these steps:

1. Open the Service Editor.
2. Click the Services tab and click Create.
   The Create Service dialog appears.
3. Click 
   The Create Resource Monitor dialog appears.
4. Perform the following steps:
   a. Enter a name for the resource monitor and, optionally, a security string.
   b. Select whether the resource monitor monitors the resources in a container or the container itself.
   c. Select a policy.
   d. (Optional) Specify which alarms you want to affect or not affect the health of the resource monitor.
   e. Click OK.
   The resource monitor appears in the Containers in Resources to Monitor list in the Create Service dialog.
5. Add the resources with the attribute you want to monitor (the attribute that is specified by the resource monitor policy) with the resource monitor:

a. Select the resource monitor.
b. Click the Locate resources and containers icon.
c. Select the containers and resources for the resource monitor.
d. Click Add Selected to the Monitored Resources.

The resources appear under the resource monitor icon in the Containers and Resources to the Monitor panel.

Specify the Alarm Types That Affect or Do Not Affect Service Health

Content

You can specify resource alarm types to affect or not affect service health when you create a service or resource monitor with a policy that specifies the Condition attribute. You can perform this task from the Exemptions Panel by selecting the appropriate alarm impact options.

Note: This functionality replaces the manual setting of the Exempt_Cause_List attribute which was documented for previous releases.

Consider the following information before you specify resource alarm types:

- Individual services or resource monitors can specify a list of alarm types to be excluded from or included in the service health calculation.
- Service alarm type exemptions can be used to establish a specific behavior for individual services. For example, only this service is affected by this alarm type, or this particular service should not be affected by this alarm type.
- Service alarm type exemptions take precedence over any configuration that is defined at the policy.

> **NOTE**
> When you specify alarm type exemptions for a service, Service Manager ignores any exemption specification that is defined for the policy that is used by the service.

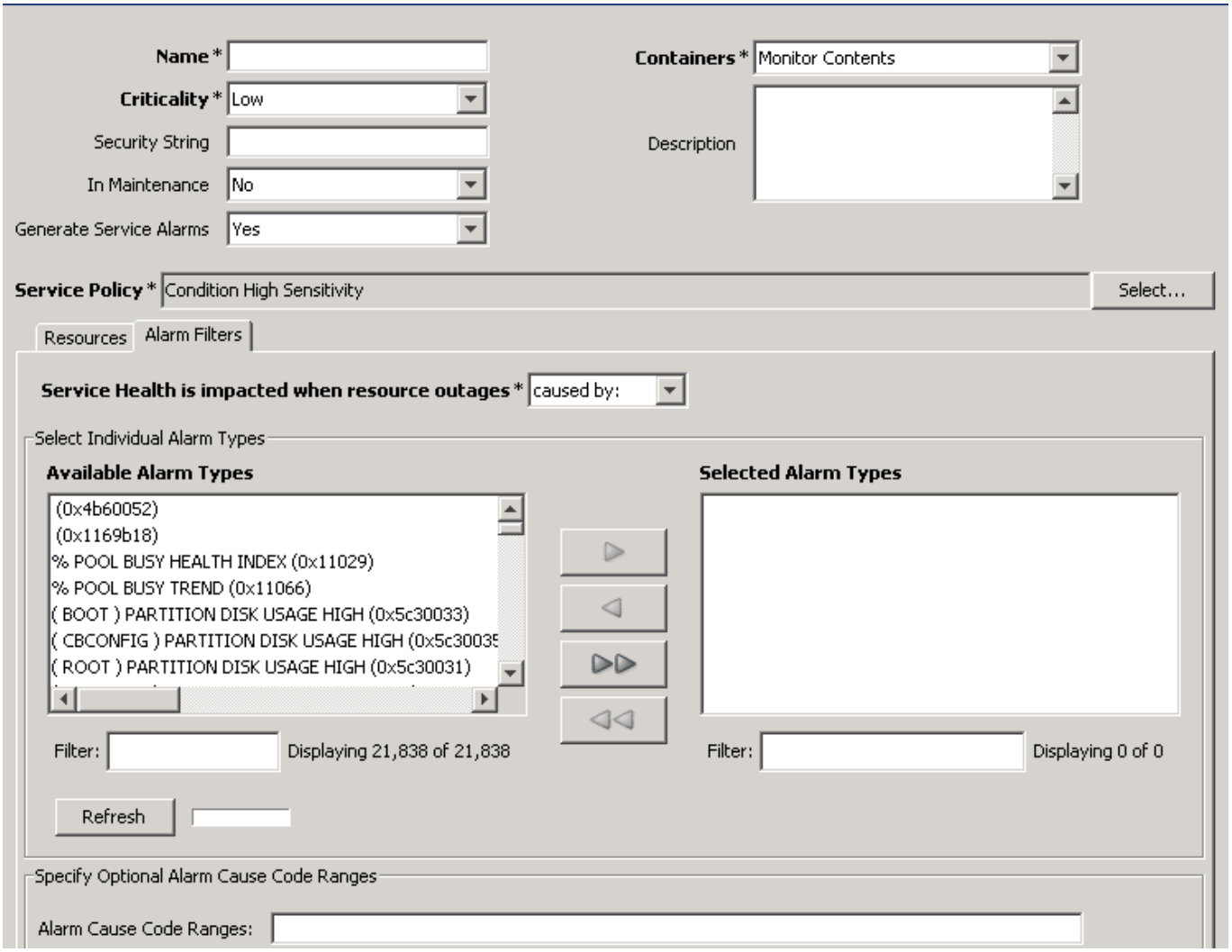
Follow these steps:

1. Perform one of the following tasks:
   – Create a service.
   – Create a resource monitor.
2. Click the Exemptions tab and select one of the following alarm impact options from the Service Health Impacted Only When Resource Outages * drop-down list:
   – **Caused By**
   Only these selected alarm types impact the service health.
   – **Not Caused By**
   Exclude these alarms from impacting the service.
   – **Disabled (default)**
   Do not use service level alarm type exemptions. If the policy defines exemptions, you can use it.
3. Move the available alarm types that you want to affect or not affect a service to the Selected Alarm Types box, or specify a range of alarm cause codes.
   The alarm types are specified.

Alarm Filters

In DX NetOps Spectrum, the health of a service can be impacted based on the alarms that are generated on the resources on which the service is monitored. If a filter is applied to any specific alarm, that particular alarm participates in affecting the health of the service. The alarm is also considered while calculating the health of a service.

The following image shows the Alarms Filter tab and the available options:



If the alarm filters tab is disabled, the alarm functionality does not work, irrespective of alarm types that are selected to impact the health of the service. By default, the alarm filters tab is disabled.

- Service Health is impacted when resource outages*
  - **caused by**
    Impacts the health of the service if the generated alarm on Service Monitored Resources (SMR) is of the selected alarm type.
  - **not caused by**
    Impacts the health of the service if the generated alarm on SMR is not of the selected alarm type.
- **Alarm Cause Code Ranges**
  Impacts the health of the service if the generated alarm on SMR falls in the specified alarm cause code range.
- **Specify Alarm Title Keywords**
  Impacts the health of the service if the generated alarm title on SMR matches with the specified alarm title. For example, CHASIS.

  NOTE
  You can define the same functionality (Alarm Title Keyword, Alarm Types) on Alarm Filter at the service policy level.

Add a Resource to a Service

You can create a service and can add resources to the service, or can add resources to an existing service or resource monitor using the OneClick Console as an alternative to using the Service Editor.

Follow these steps:

1. Select the resource (model) you want to add to an existing service or a service you plan to create.
2. Right-click the model and click Add To, Service.
   The Add to Service/Resource Monitor dialog appears. This dialog lists available services.

   NOTE
   To display resource monitors for a service, select the service.
3. Create a service to which you can add the resource or can add the resource to an existing service or resource monitor:
   – To create a service, click Create.
   The Create Service dialog appears. It lists the selected resource as a resource of the service. Specify service properties as described in Create a Service.
   – To add the selected resource to an existing service or a resource monitor, select the service or resource monitor to which you want to add the resource and click OK.
   The resource is added to the service or resource monitor.

Delete a Resource from a Service

You can remove a resource from a service using the OneClick Console as an alternative to using the Service Editor.

Follow these steps:

1. Select the resource (model) that you want to remove from an existing service.
2. Right-click the model and click Remove From, Service.
3. Click OK.
   The resource is removed from the service.

Edit a Service

You can change all service properties anytime after you create a service.

Consider the following information before you edit a service:

- If you change a service name, all historical data in reports for the service that is generated with DX NetOps Spectrum Report Manager is listed under the new name.
- If you delete a service that is monitored by an SLA, the SLA guarantee goes to the Initial (Blue) state.
- If a service has a resource monitor monitored by an SLA and you delete the resource monitor, the SLA guarantee watching it goes to the initial (Blue) state.

  WARNING
  Editing services can be performed by authorized personnel who are aware of the potential ramifications of these actions, particularly as they relate to services monitored by SLAs.

Follow these steps:

1. Open the Service Editor.
2. Click the Services tab, select the service that you want to edit from the list, and then click Edit.
   The Edit Service dialog appears.
   Edit the settings, as described in Create a Service, and click OK.
   The service is edited.

Delete a Service

Before you delete a service, verify that it is not associated with an SLA you intend to implement that is scheduled to be activated.

WARNING
Deleting services can be performed by authorized personnel who are aware of the potential ramifications of these actions, particularly as they relate to services monitored by SLAs.

Follow these steps:

1. Open the Service Editor dialog.
2. Click the Services tab, select the service that you want to remove from DX NetOps Spectrum, and then click Delete.
3. Respond to the confirmation message that appears to complete the deletion.
   The service is deleted.

Cut a Service

DX NetOps Spectrum Service Manager lets you cut a resource monitor as part of editing a service.

Create a service and add a resource monitor. The policy and resources do not matter. Save and edit the service.

NOTE
With the resource monitor selected, the cut icon is disabled.

Follow these steps:

1. Open the Service Editor dialog.
2. Click the Services tab, select the service that you want to cut from the resource monitor, and then click Cut.
   The service is cut.

Service Maintenance Schedule Management

When you schedule maintenance for a service, DX NetOps Spectrum puts the service in maintenance mode (brown state) for the duration that is specified by the schedule. A service in maintenance mode is a planned outage. If the service is monitored by an SLA, the frequency and duration of the scheduled planned outages are typically defined by the SLA contract between a service provider and a service customer. Planned service outages are not accumulated as down or degraded time by SLA guarantees.

You can manage service maintenance schedules in the following ways:

- Create and save multiple one-time and recurring schedules.
- Add schedules to the list of schedules to be implemented on an as-desired basis.
- Remove schedules from the list of schedules to be implemented.

Create a Maintenance Schedule

When you create a schedule, Service Manager saves it to DX NetOps Spectrum. The schedule is added by default to the list of schedules that can be implemented for the service. If you do not want the schedule to be implemented, you can remove it from the list and can add it to the list at another time. You have two ways to create or specify a maintenance schedule for a service. You can specify scheduled maintenance from the Component Detail view of a service model much like other DX NetOps Spectrum models, or you can configure scheduled maintenance from the service editor.

Follow these steps:

1. Open the Service Editor.

2. Click the Services tab, select the service for which you want to manage maintenance schedules, and then click the Scheduled Maintenance tab.
3. Click Create.
   The Create Schedule dialog appears.
4. Configure the new schedule and click OK.
   Service Manager adds the schedule to the Current Schedules list.

Add a Maintenance Schedule to the Current Schedules List

If you want to implement a maintenance schedule for a service, include it in the Current Schedules list for the service.

Follow these steps:

1. Open the Service Editor.
2. Click the Services tab, select the service for which you want to manage maintenance schedules, and then click the Scheduled Maintenance tab.
3. Select the schedule to add from the Available Schedules list and move it to the Current Schedules list.
   The maintenance schedule is added to the Current Schedules list.

Remove a Maintenance Schedule from the Current Schedules List

If you do not want to implement a maintenance schedule, you can remove it from the Current Schedules list.

Follow these steps:

1. Open the Service Editor.
2. Click the Services tab, select the service for which you want to manage maintenance schedules, and then click the Scheduled Maintenance tab.
3. Select the schedule that you want to remove from the Current Schedules list, and then move it to the Available Schedules list.
   The maintenance schedule is removed from the Current Schedules list.

Associate an Owner with a Service

Service Manager lets you designate one or more users as owners of a service. A service owner serves as the contact person for the service. Service owner information and also be available from Service Availability and Service Health reports.

Follow these steps:

1. Open the Service Editor.
2. Click the Services tab, select the service to associate with an owner, click the Owners tab, and then click Select Service Owners.
   The Select Owners dialog appears.
3. Move the owner to associate with the service from the Available Owners list to the Service Owners list and click OK.
   The owner is associated with a service.

Associate a Customer with a Service

Service Manager lets you associate one or more customer models with a service to help you track and manage services and customers. You can generate service reports with DX NetOps Spectrum Report Manager based on service customer associations. When a service is associated with one or more customer models, the Criticality of each customer can be

factored into the alarm impact for any service impacting outage. This assures that resource outages which impact highly critical customers can have a greater impact value.

Follow these steps:

1. Open the Service Editor.
2. Click the Services tab, select the service that you want to associate with a customer, click the Customers tab, and then click Select Service Customers.
3. The Select Customers dialog appears.
4. Move the customer that you want to associate with the service from the Available Customers list to the Customers that use this Service list and click OK.
   The customer is associated with a service.

Service Models in a DSS Environment

When creating service models in a distributed SpectroSERVER (DSS) environment there are some important factors to consider. Service models can be associated with resource models from other landscapes. The supported behaviors are as follows:

- A service model that is created on the MLS can monitor resources from the MLS or any second tier landscape.
- A service model that is created on a second tier landscape can monitor resources models on its own landscape or the MLS, but not other second tier landscapes.

The following behaviors apply to services which use global collections to define their resources:

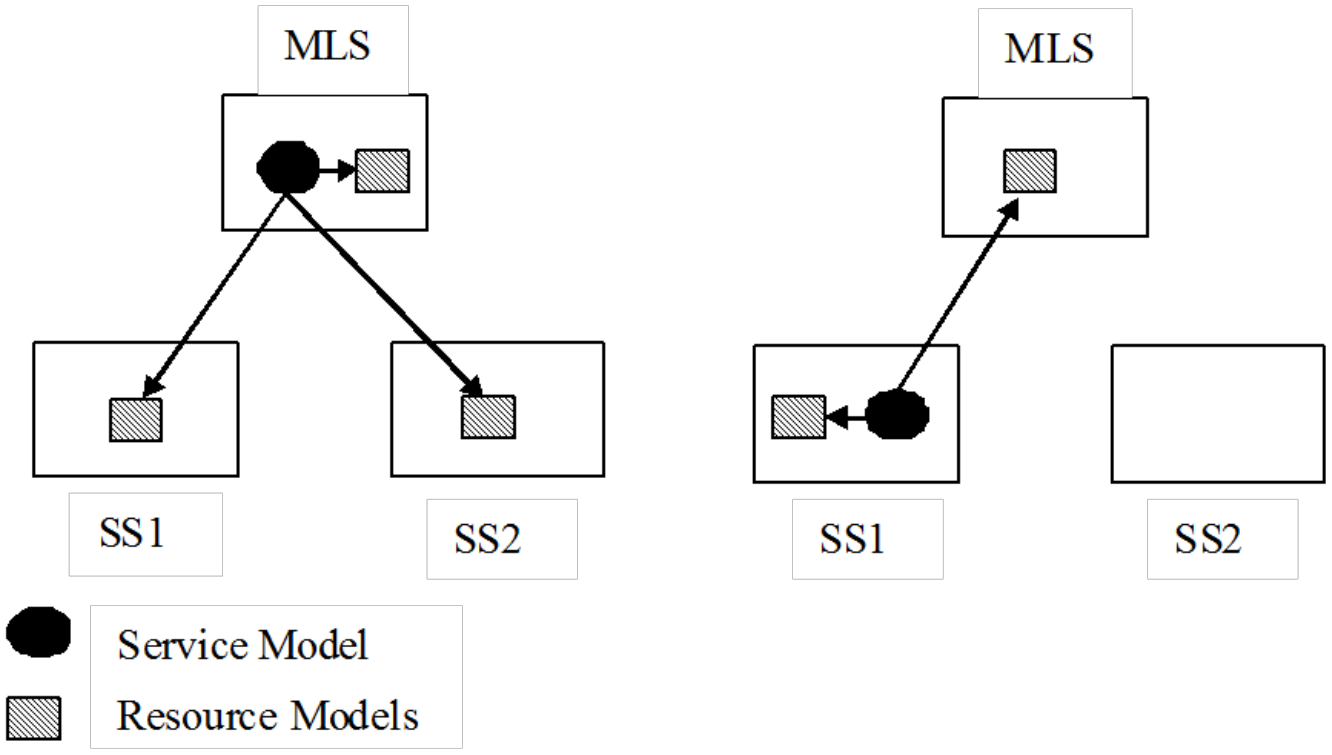
- A service model that is created on the MLS using a global collection can monitor all resources that are contained by the global collection.
- A service model that is created on a second tier landscape using a global collection can monitor only those resources in the collection which reside on its own landscape or the MLS.

In 9.2, the performance-heavy cross landscape watches are replaced with asynchronous action notifications, relieving some of the burdens from the SpectroSERVERs. In addition, a relay mechanism allowing the MLS to forward second tier notifications from one landscape to another allows second tier services to monitor resources from other second tier landscapes.

This lets you construct services with less concern over performance impact. In addition, it lets you create services on landscapes where the service should logically reside without concern over what landscape required remote resources reside upon. This does not eliminate the need for an efficiently designed service hierarchy, but it provides flexibility which makes it easier to design the service hierarchy. All resources from all landscapes are visible, regardless of where the service mode is being created.

Example: Supported Service to Remote Resource Configurations

The following image depicts the supported service to remote resource configurations:



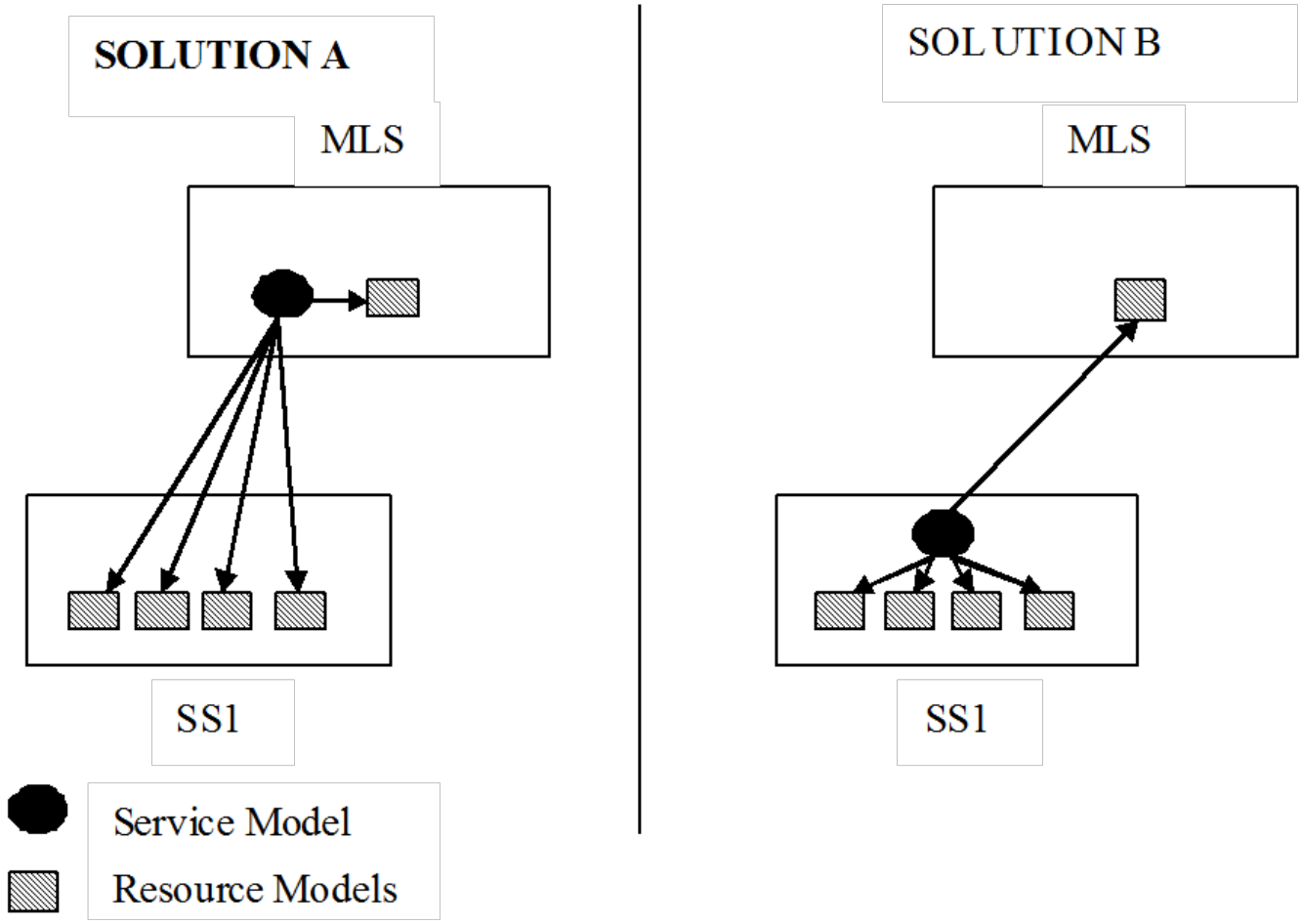
In addition to understanding which configurations are supported it is also important to consider the efficiency of the service model, and resource distribution impact. Monitoring a resource that resides in the same landscape is more efficient than monitoring a resource that resides in another landscape. When creating service models, strive to minimize the number of service resources that reside on remote landscapes. Consider the following example. Service XYZ has five resources, 1 of these resource models resides on the MLS while the other 4 reside on SpectroSERVER 1 (SS1). The following image depicts two potential designs for service XYZ:

Service Model Example: Resource Distribution Impact

Because of the reduced number of resources residing on a remote landscape, Solution B is a more efficient design. If the service can reside on either landscape, select the landscape where most resources exist. You can summarize this statement by saying ‘build the service closest to the bulk of its resources’.

There are some circumstances where the service has resources on multiple second tier landscapes and must reside on the MLS. In these scenarios, more efficient service design can be created by consolidating the remote resources into a sub-service which is monitored by the parent service on the MLS.

The following image depicts two solutions which can be created when the service must reside on the MLS:

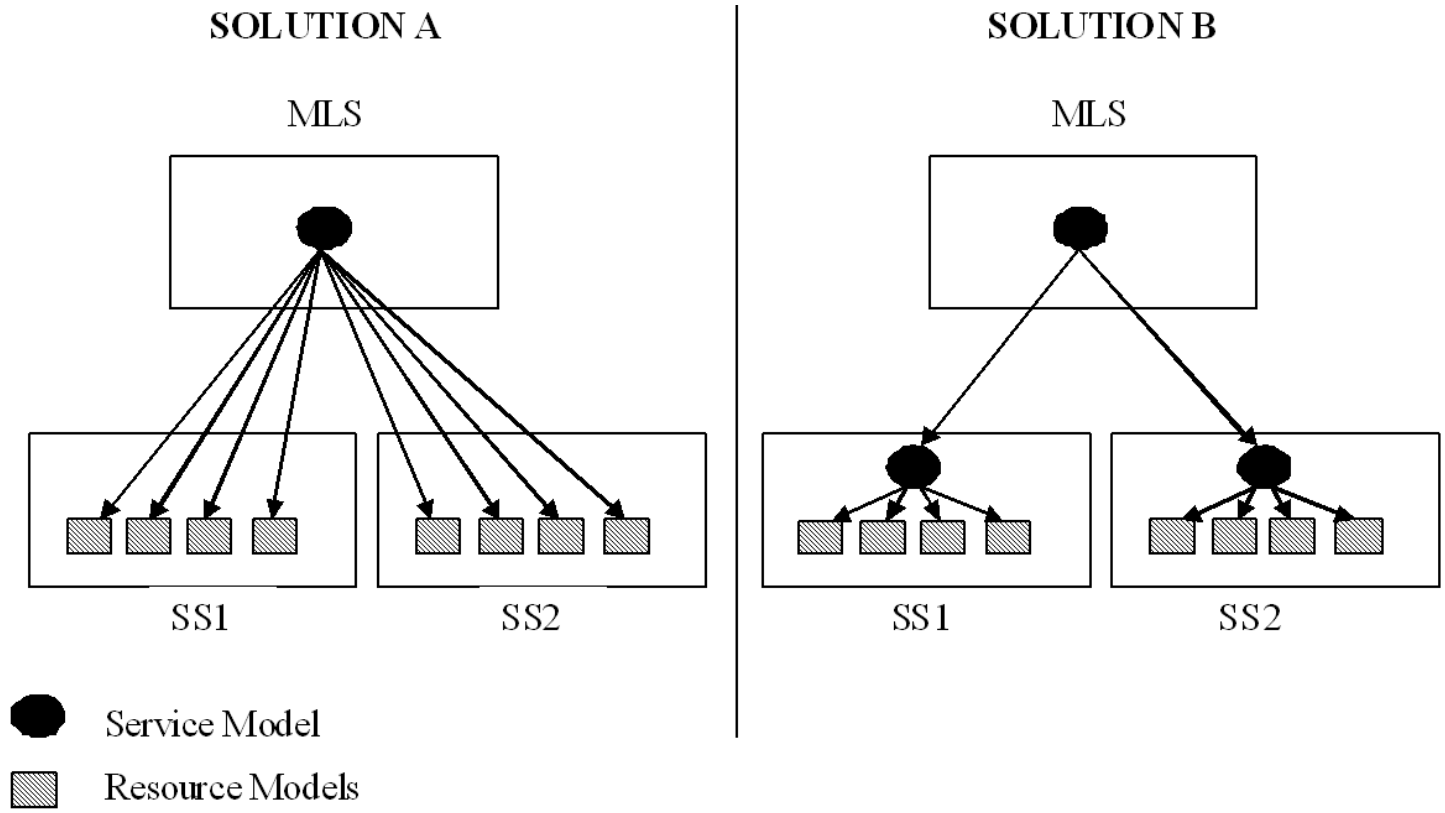


Service Model Example: Service Resides on the MLS

Solution B represents a more efficient design by minimizing the number of resources that are monitored remotely. In Solution B, the service models that are created on SS1 and SS2 become resources of the service that is created on the MLS. When creating service hierarchies, such as the one depicted in Solution B, it is important to verify the policies that are used for each service, reflects the behavior of the resources from each landscape.

When creating services that use global collection consider the number of remote resources that result from the use of the global collection. When any type of container is used to specify service resources by default, the service monitors the contents of the container. A global collection can contain models from multiple landscapes.

The following image illustrates the use of a global collection that can potentially create an inefficient service design.



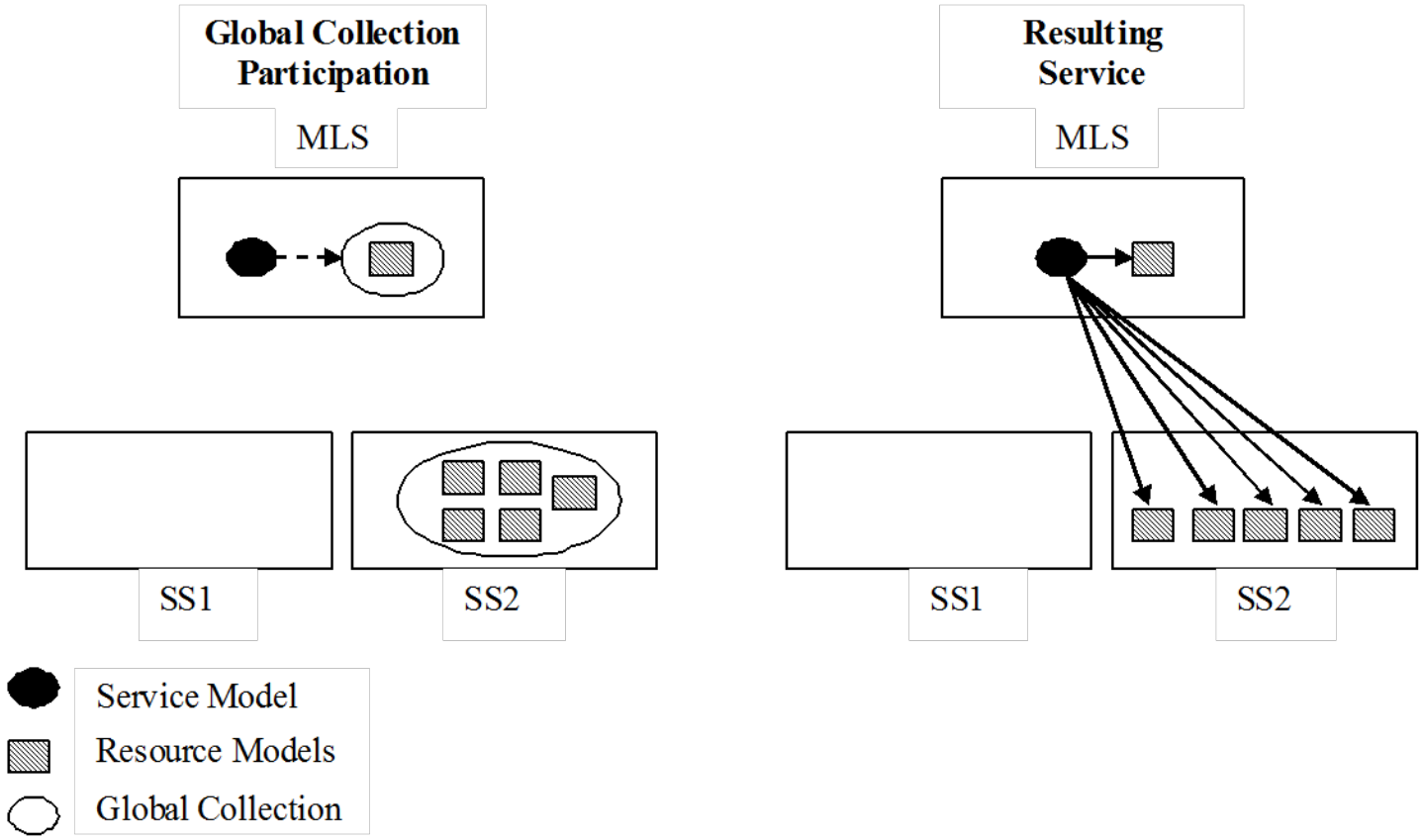
Again Solution B, represents a more efficient design by minimizing the number of resources that are monitored remotely. In Solution B, the service models that are created on SS1 and SS2 become resources of the service that is created on the MLS. When creating service hierarchies, such as the one depicted in Solution B, it is important to verify the policies that are used for each service, correctly reflects the behavior of the resources from each landscape.

When creating services that use Global Collection consider the number of remote resources that result from the use of the Global Collection. When any type of container is used to specify service resources by default, the service monitors the contents of the container. A Global Collection can contain models from multiple landscapes. Consider the following images, and how the use of a Global Collection could potentially create an inefficient service design.

Example: Global Collection

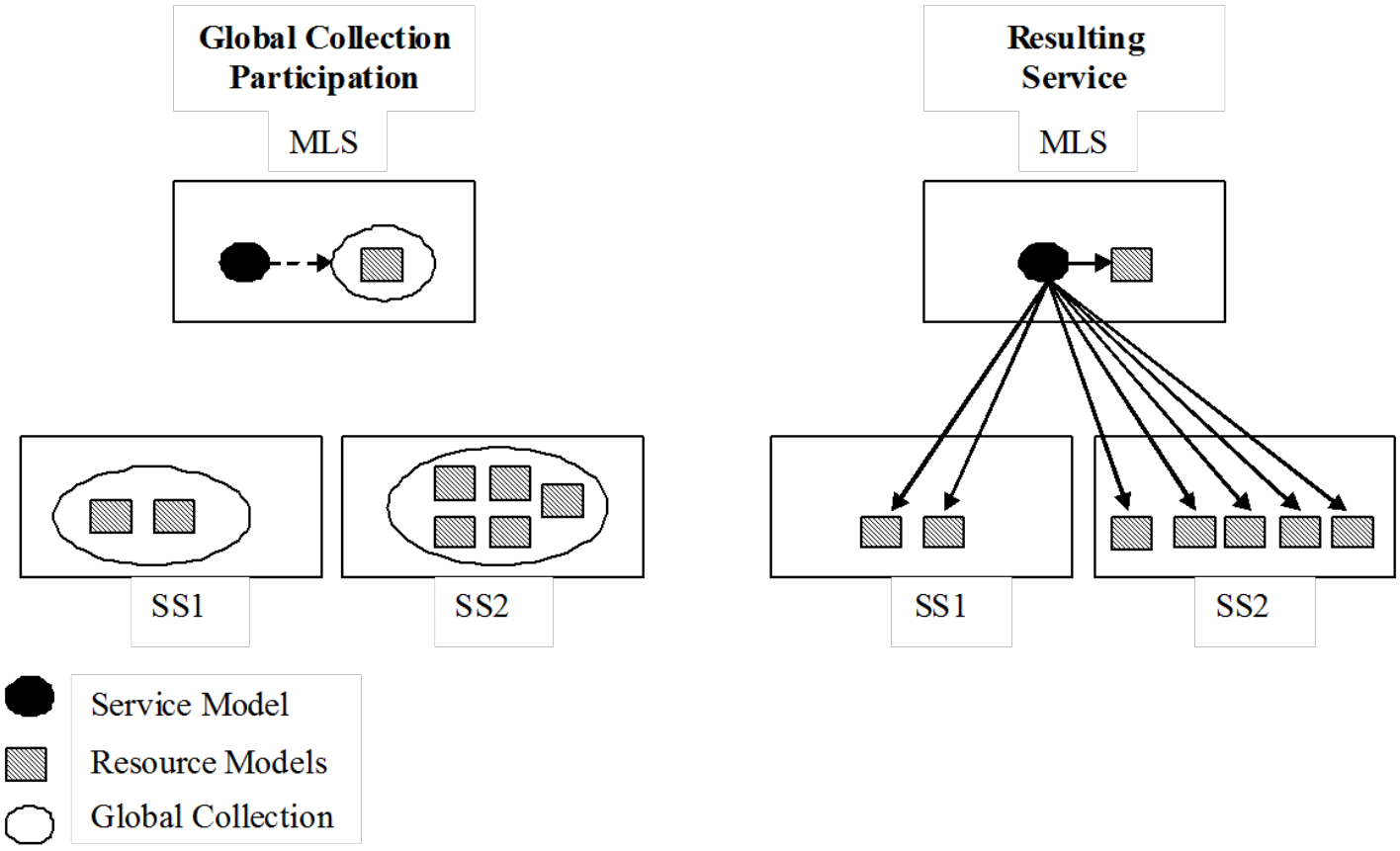
In the following scenario, if the service is created on the MLS it would be monitoring five remote resources. If the service was created on SS2, it only has to maintain one remote resource.

However, improving efficiency of this scenario cannot be as simple as moving the service to SS2. The service is using a global collection to specify its resources to support a set of potentially dynamic resources. Consider if two more models are created on SS1 which participate in the global collection.



Global Collection Example: Add Two Additional Models

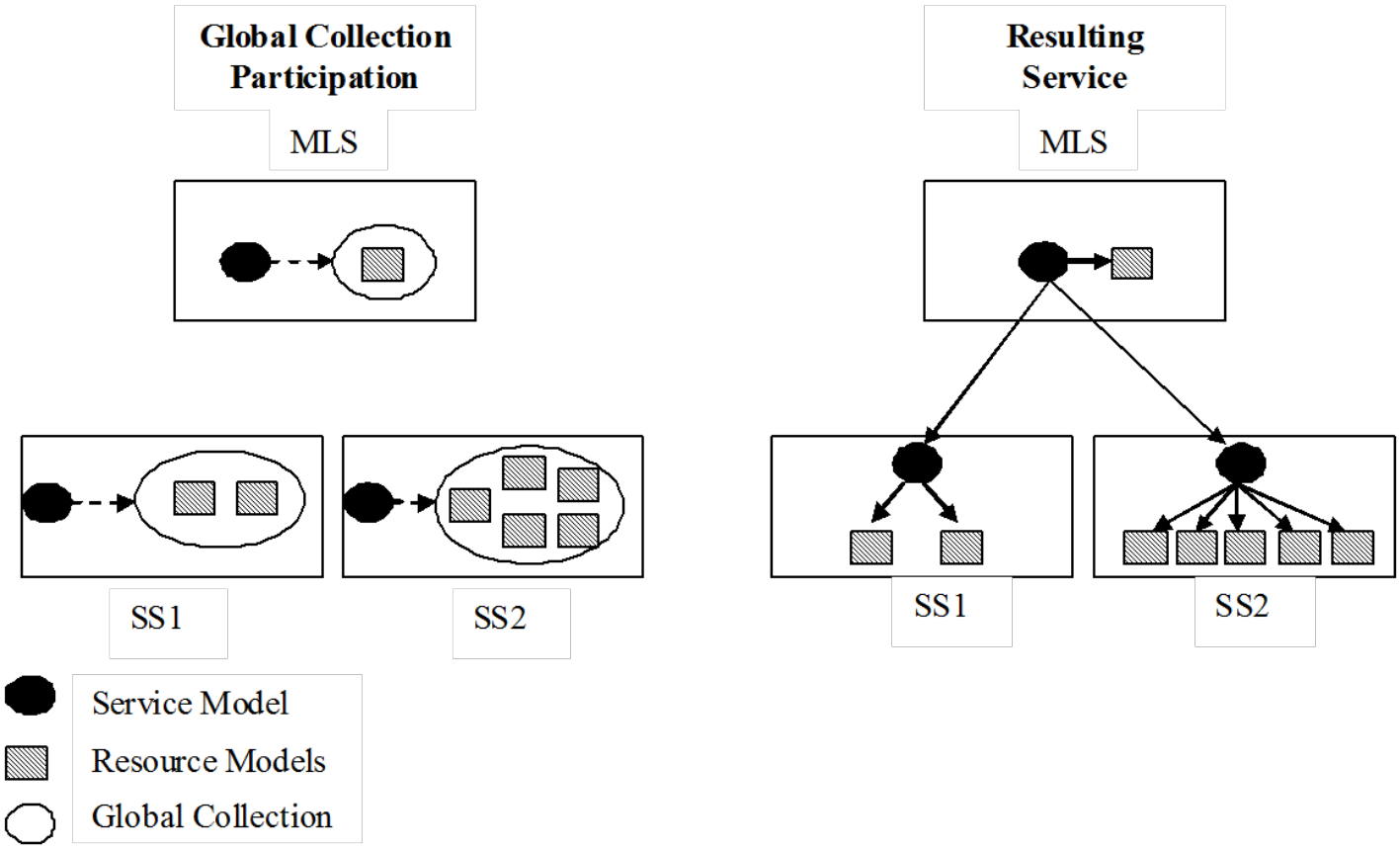
If the global collection contained resources from all landscapes, the service has to reside on the MLS. An alternative to consider is the use of multiple services that monitor the local copy of the global collection, residing on the same landscape. Although logically a global collection is a single model, it is implemented as a set of duplicate models with a model residing on each landscape for which the global collection is specified.



Example: Multiple Services Monitoring the Local Copy of the Global Collection

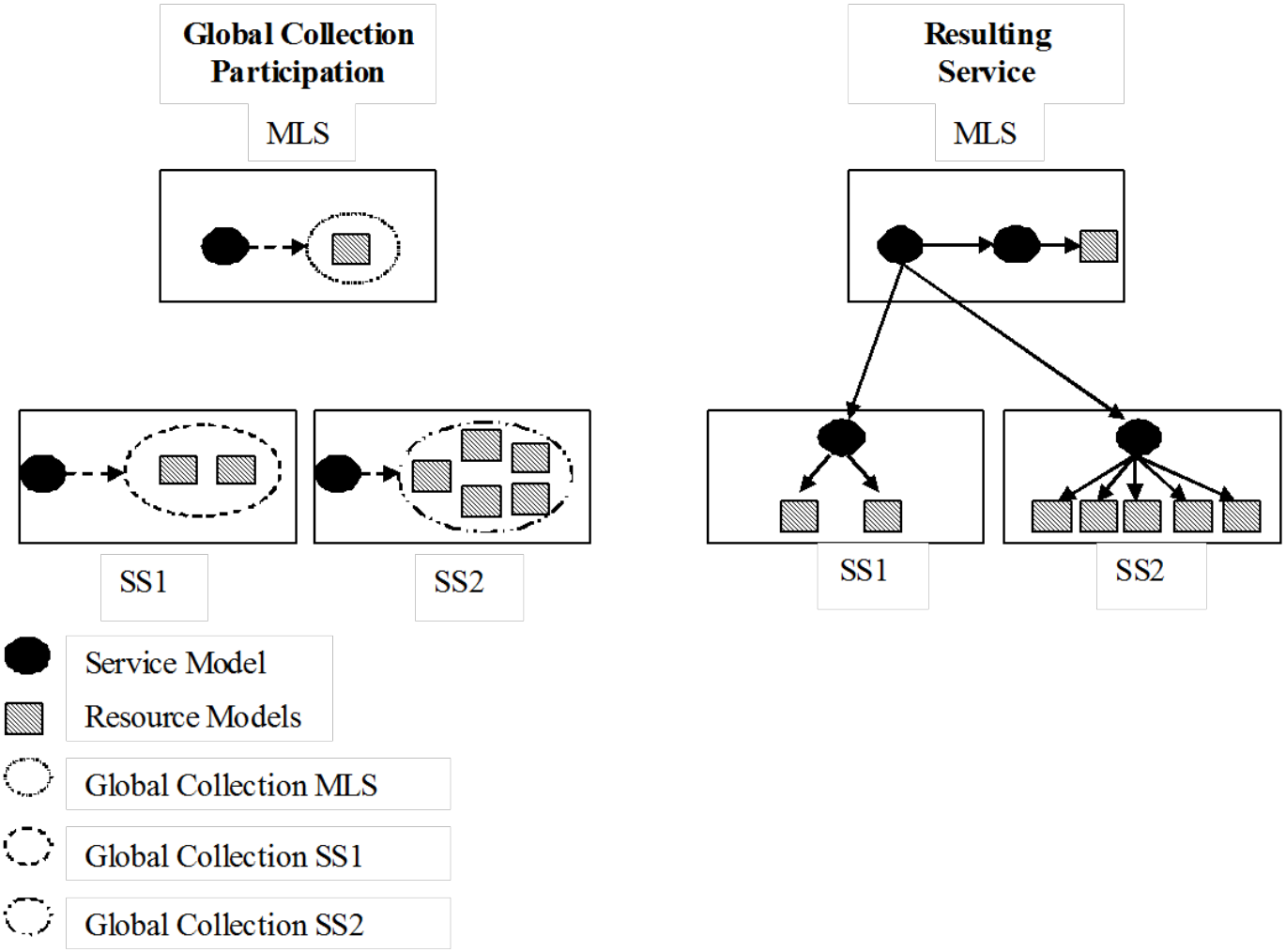
Although this approach does produce a more efficient service, it can be complex to maintain and may require the use of DX NetOps Spectrum Command Line Interface (CLI) to implement.

An alternate technique would be to create multiple global collections which are bound to a single landscape. Services can then be created on each landscape, with a parent service to monitor them.



Example: Multiple Global Collections Bound to a Single Landscape

This design allows for the benefits of a dynamic collection, but also provides an efficient service design. It may not be ideal to maintain multiple Global Collections in this way so you must weigh the costs and benefits of this solution.



Policies

A service policy reflects the behavior of a set of resources that logically impacts a service. The policy specifies which resource attribute is monitored and how those attributes are interpreted to determine the health of a service. A number of common policies are available out-of-the-box, and users can create their own policies to more accurately monitor service resources.

A policy includes the following basic components:

- **Attribute map**
  The attribute map serves two purposes. First, it specifies which resource attribute is monitored. Second, it maps resource attribute values to a set of resource health values. The mapping is done based on the logical severity of the attribute value. For example, if the attribute map represents the Port Status attribute, a disabled port can be logically considered as a down resource. If the attribute map represents the status of a response time test, a minor threshold violation and a slightly degraded resource can be considered.

This mapping allows policy rule sets to handle various resource types in a common way by having only to consider the resource health values of down, degraded, and slightly degraded.

- **Rule set**
  A rule set consists of a number of statements evaluating the cumulative mapped resource health values of a set of service resources against some criteria. Each rule within the set specifies the criteria and the resulting service health value if the criteria are met. For example, a rule can look something like: When all resources are down the service is down. This means that given the mapping of the monitored attribute if all resources have a resource health of down the resulting health of a service using this policy can be down.
  As mentioned, the rule set consists of multiple statements or rules. The rules are evaluated from the top down, and the first rule which is satisfied determines the health of any service using that policy. If none of the rules in the rule set are satisfied, the service health can be up.

  WARNING
  Editing and deleting policies and attribute maps and rule sets can be performed by authorized personnel who are aware of the potential ramifications of these actions, particularly policies for services that are watched by Service Level Agreements (SLAs).

Policy Types

Before you create a custom policy, have a clear understanding of what you want to monitor (the watched attribute and its status values) and the method that is used for the monitoring. Two distinct types of service monitor policies are available, such as:

- Status policies use an attribute map to compare the status of monitored attributes that are indicated by the rule sets. These policies use the *All*, *Any*, or *Percentage* rule sets.
- Statistical policies use Aggregate rules and compare the value of the monitored attribute to the computed value set by each rule. A statistical policy requires an attribute map to specify which attribute can be monitored, but the mapped values are ignored in favor of the attributes pure values. The attributes values can be summarized in several ways, such as average, minimum, maximum. The summarized values are then compared against numeric thresholds that are specified in the rule-set.

For example, a status policy can monitor the operational status of an interface, where a statistical policy can monitor the error rate of an interface.

Create a Policy

You can create policies using any combination of standard and user-defined attribute maps and rule sets. When you create a policy, it becomes available to other Service Manager users. You can also create a policy by saving a uniquely named version of an existing policy.

Follow these steps:

1. Open the Service Policy Editor.
2. Click the Policies tab and click Create.
   The Create Policy dialog appears. Service Manager autofills the Author field with your DX NetOps Spectrum user name.
3. Enter a unique name for the policy in the Policy field.
4. Select an attribute map.
   Properties for the map you select appear. If the attribute maps available do not meet your requirements, you can create an attribute map or can edit an existing user-created attribute map.
5. Select a rule set.

Rules for the rule set you select appear. The rule set specifies the status (or health) of a service is based on the mapped health values of the service resources. A rule set consists of conditional statements that assert what the health of the service is based on the collective set of resource health values.
If the available rule sets do not meet your requirements, you can create a rule set or can edit an existing user.

6. (Optional, and only applicable if you chose a Condition attribute map).
   Specify alarm type exemptions for the policy. This option consists of a set of alarm types and how they are applied either inclusively (caused by) or exclusively (not caused by).
   You can specify alarm type exemptions at the policy level, if the policy is used by multiple services. If the alarm type exemption configuration is only relevant for a single service, it can be better to define if for the service itself rather than at the policy level.
   Consider the following points before you allow or disallow alarm type lists in policies:
   - Alarm type exemptions that are configured at the policy are enforced only for services using the policy which do not define their own alarm type exemptions. If a service using the policy is later edited and has alarm type exemptions that are configured for it, the policies configuration can be ignored from that point forward.
   - Any changes that you make to the alarm type exemption configuration for a policy affect all services that use the policy.

   Follow these steps to specify alarm type exceptions for a policy:
   a. In the policy create/edit dialog Click Set. (If the set button is disabled, it is likely that the specified attribute map is not for the Condition attribute, only Condition-based policies can specify alarm type exemptions).
      The Configure Inclusive or Exclusive Alarm Types dialog appears.
   b. Specify Caused By or Not Caused By.
   c. Select the appropriate alarm types and click OK.
   d. The Create Policy dialog shows the alarm type inclusion or exclusion list that you specified.
7. Click Create.
   The policy appears in the Policy list in the Service Policy Editor dialog.

Add Alarm Types to a Custom Condition Policy

You can add resource impacting alarm to the list of alarm types that are specified in any user created policy which uses a Condition attribute. This option is available when selecting an Alarm in the Alarm tab of the OneClick Contents Panel.

Follow these steps:

1. Select one or more alarm types to include in the policies from any OneClick alarm view.
2. Right-click and click Add Alarm Type to Service Policies.
   The Specify Alarm Types for Service Policies dialog appears. This dialog lists selected alarm types and all custom (user-created) Condition policies.

   NOTE
   You can include the Alarm Behavior column by right-clicking any table column heading. This indicates how the policy interprets the alarm exemption Caused By or Not Caused By.
3. Select the policies to which you want to add the alarm types and click OK.
   The alarm types are included in the alarm type list for each selected policy. If you selected a policy without an alarm type configuration, Service Manager defaults to a setting of Not Caused By for the policy.

Create a Policy from a Copy

You can create a policy by copying an existing policy and saving it with the settings you require under a different name.

Follow these steps:

1. Open the Service Policy Editor.
2. Click the Policies tab, select the policy from which you want to create a policy, and click Copy.

The Create Policy: <*Policy Name*> dialog appears. It includes the settings for the policy you are going to use as the basis for your new policy.

3. Enter a unique name for the new policy in the Policy field.
4. Edit policy properties, as described in Create a Policy, and click Create.
   The new policy appears in the Policy list in the Service Policy Editor dialog.

Edit a Policy

You can modify all user-created policies. Service Manager implements modified policy settings immediately after you save them.

Follow these steps:

1. Open the Service Editor.
2. Click the Policies tab, select the policy that you want to edit, and then click Edit.
   The Edit Policy: <*Policy Name*> dialog appears.

   NOTE
   Policy edits are applied to all services or resource monitors which are currently using the policy. Depending on the nature of the edit, it results in a change to the service health of one or more services.
3. Edit the settings, as described in Create a Policy, and click OK.
   The policy is edited.

The following limitations apply to the service policy editing:

- You can edit or delete user-created policies, attribute maps, or rule sets. However, Service Manager prevents you from deleting any policy (or any attribute map or rule set that are part of it) that is used by a service or resource monitor.
- You cannot edit or delete CA-authored Service Manager policies.
- You can copy any policy and can save it under a unique name and can edit it or can delete.
- You cannot edit or delete CA-authored attribute maps and rule sets.
  You can copy any attribute map or rule set and can save it under a unique name.

Delete a Policy

You can delete any user-created policy that is not currently in use. If the policy used by a service deletion fails, an error dialog indicating the failure is displayed.

NOTE
When you delete a policy**,** do not delete its attribute map and rule set.

Follow these steps:

1. Open the Service Editor dialog.
2. Click the Policies tab and then select the policy that you want to delete.
3. Click Delete.
   The policy is removed from the Policy list.

Attribute Maps

Attribute maps associate natural resource attribute values to equivalent resource health values, and let you specify a root cause reason string for each mapped value. Any attribute which has whole number values can be mapped. Multiple attribute maps can exist for the same attribute, and can have different value mappings or simply different root cause reason strings. Each attribute map has its own name which describes its purpose.

The attribute map serves two purposes when added to a service policy. First it specifies which resource attributes monitored. It provides a mapping of how the resource attribute values indicate the health of the resource. When the specified attribute changes for a resource model, the service evaluates the value to determine its relative resource health and apply that health value with all of the other resources health values to the rule set.

If the out-of-the-box attribute maps are not sufficient, you can create a custom attribute map. All user-created attribute maps can be edited and deleted by any Service Manager user.

Consider the following information about attribute maps:

- An attribute map must have a unique name.
- You cannot edit or delete CA-authored attribute maps.
- Any attribute can have multiple mappings.
- Service resource(s) must support the attribute that is specified in the attribute map. DX NetOps Spectrum generates a minor (yellow) alarm on the Service Management model, if a service uses a policy that specifies a watched attribute that is not supported by the service resources.
- You must know what values an attribute returns and what they indicate about the state of the resource(s) on which they are monitored to map those values to service health values.
- At a minimum, an attribute map must include one mapped value, and must define a default root cause reason.
- You can map multiple attribute values to the same service health value. For example, port statue values of disabled, down, and unreachable are all mapped to a resource health of down.
- If the attribute you select returns enumerated values, each value must be mapped to up, down, degraded, or slightly degraded before you can save the new mapping.
- Service Manager attribute maps now support range values in addition to single value mapping.

For example, an attribute map value can appear in this form single value 100, range value 100-200. Using the following information, you can configure attribute maps:

1-99 = Slightly Degraded

- 100-199 = Degraded
- 200-300 = Down

Also supported is greater than and less than operators, so a set of mapped values may look like the following equations:

- <100 = Down
- 100-200 = Degraded
- 300-400 = Degraded
- >400 = Down

Create an Attribute Map

You can create a custom attribute map using Service Policy Editor.

Follow these steps:

1. Open the Service Policy Editor.
2. Click the Attribute Maps tab and click Create.
   The Create Attribute Map appears.
3. Specify the following properties:
   - **Attribute Map**
     Identifies the attribute map. Provide a unique name.
   - **Default Root Cause Reason**
     Identifies the underlying reason for the service health state. Later, when you map particular attribute values to service health values, you can overwrite the default reason for each mapping.
4. Click Attribute.

The Attribute Selector dialog appears.

5. Select an attribute of the allowed type (counter, integer, date, time ticks, and gauge), and then click OK.
   The Create Attribute Map dialog appears.
6. Click Add.
   The Add Value to Service Health Mapping dialog appears.
7. Enter the following values:
   - **Attribute Value**
     Specifies the attribute value that you want to map to a service health value.
   - **Service Health**
     Specifies the service health value that you want to map to the attribute value.
   - **Root Cause Reason**
     Describes the root cause that you want to provide for this particular mapping. The root cause reason is displayed in the root cause tab of the OneClick Component detail panel, and the Outage Details in a service outage history table.
     You can create new attribute maps specifically to provide better root cause reasons. For example, the standard Condition attribute map defines generic root cause reasons.
     It is useful to create multiple purpose-specific attribute maps for the Condition attribute. For example, you can define a Condition attribute map specific for services monitoring cable modems and can define root cause reasons like: “Modem disconnect, and Data transfer fault.” These attribute maps are more descriptive strings, which are used in place of standard Condition strings of “A critical problem was detected on the resource, and A major problem was detected on the resource.”
8. Click OK.
   The Mapped Values panel lists the new attribute value and service health mapping.
9. Click Create to save the new attribute map.
   The new attribute map appears in all attribute map lists in Service Manager, and it can be used in all user-created policies.

Create an Attribute Map from a Copy

You can create an attribute map by copying an existing map and saving it with the settings you require under a different name.

Follow these steps:

1. Open the Service Policy Editor.
2. Click the Attribute Maps tab, select the map that you want to copy, and click Copy.
   The Create Attribute Map dialog appears. It includes the settings for the map you want to copy.
3. Specify a unique name, modify settings as required (using the set command where applicable), and click Create.
   The new attribute map appears in all attribute map lists in Service Manager, and it can be used in all user-created policies.

Edit an Attribute Map

You can edit any user-defined attribute map regardless of whether it is included in a policy that is in use. If edits to the attribute map are saved, all services or resource monitors using related policies reevaluate their health that is based on the edit and can result in a change in service health for the models.

Consider the following information before editing an attribute map:

- You can change the attribute map name, the service health designation for an attribute value, the default root cause reason, and the root cause reason for each service health mapping.
- You cannot change the attribute that is originally specified.

Follow these steps:

1. Open the Service Policy Editor.
2. Click the Attribute Maps tab, select the attribute map that you want to edit, and then click Edit.
   The Edit Attribute Map: <*Attribute Map Name*> dialog appears.
3. Modify settings as required (using the set command where applicable), and click OK.
   The attribute map is edited.

Delete an Attribute Map

You can delete any user-created attribute map that is not part of a policy currently in use.

Follow these steps:

1. Open the Service Policy Editor.
2. Click the Attribute Maps tab, select the map that you want to delete, and click Delete.
   The attribute map is removed from the attribute maps list.

Rule Sets

A rule set consists of a set of rules. Each rule is a conditional statement that is comprised of a comparator and a resulting health value. A rule is considered satisfied if the cumulative resource health matches its criteria. For example, when all resources are Down the service is Down. If all resources within the service have a resource health of down, the rule can be satisfied, and the policy can be evaluated to a resulting health of down.

Rules are evaluated from top-down, the first rule within the ruleset which is satisfied dictates the health of any service or resource monitor using the policy. It is important to consider rule evaluation when creating a rule set, and verify that rules of lesser significance do not hide rules of greater significance. For example, consider a rule set with the following logic:

Rule 1 = When any 1 resource is Down the service is Degraded.

Rule 2 = When all resources are Down the service is Down.

A rule set like Rule 1 and Rule 2 can never return anything other than Degraded, because even if all resources are down, Rule 1 is still satisfied as one resource is down.

Service Manager supports four different categories of rules which can be used to form a rule set. All, Any, and Percent Rules use the mapped values that are provided by the attribute map. Aggregate rules use the resource attributes natural values.

Verify the following rule types:

- **All**
  When all the monitored service resources or all resources that are watched by a resource monitor have the service health value (down, degraded, or slightly degraded), the service or resource monitor is (down, degraded, or slightly degraded).
- **Any**
  When a particular number of monitored service resources or a particular number of resources that are watched by a resource monitor have the service health value (down, degraded, or slightly degraded), the service or resource monitor is (down, degraded, or slightly degraded).
- **Percent**
  When the percentage of monitored service resources or the percentage of resources that are watched by a resource monitor have the value (down, degraded, or slightly degraded), the service or resource monitor is (down, degraded, or slightly degraded).
- **Aggregate**

When the (Sum, Minimum, Maximum, and Average) for all monitored service resources or the (Sum, Minimum, Maximum, and Average) of resources that are watched by a resource monitor is (less than, greater than, or equal to) the Integer or attribute value you designate, the service or resource monitor is (down, degraded, or slightly degraded).

A status policy rule set can consist of any combination of All, Any, and Percentage rules. Because mapped values are ignored, a statistical rule set can use only Aggregate rules.

Consider the following information about rule sets:

- You can create any number of uniquely named rule sets.
- Uniquely named rule sets can include identical rules.
- You can create new versions of existing rule sets.
- You cannot edit or delete CA-authored rule sets.
- The order in which you list a rule in rules set is important. The first rule that is satisfied dictates the service health value returned.

Create a Rule Set

You can create an original custom rule set or another version of an existing rule set in the Service Policy Editor if the CA-authored rule sets do not meet your requirements for a policy. All user-created rule sets can be modified and deleted by any Service Manager user.

Follow these steps:

1. Open the Service Policy Editor.
2. Click the Rule Sets tab and click Create.
   The Create Rule Set dialog appears.
3. Enter a name for the new rule set in the Rule Set field.
   The Author field auto-fills with the current Service Manager user name.
4. Click Add to create a rule for the ruleset.
   The Create Rule dialog appears.
5. Configure rule parameters, including the type and the conditions for the type, and click OK.
   The rule appears in the Create Rule Set dialog.
6. Rearrange rules as necessary using the Up Arrow and Down Arrow buttons, and click Create.
   The rule set is created.

Create a RuleSet from a Copy

You can create a rule set by copying an existing ruleset and saving it with the settings you require under a different name.

Follow these steps:

1. Open the Service Editor.
2. Click the Rule Sets tab, select the ruleset you want to copy, and click Copy.
   The Create Rule Set dialog appears. This dialog includes the settings for the ruleset you are going to use as the basis for your new ruleset.
3. Specify a unique name and edit rules and modify rules as required, and then click Create.
   The rule set appears in all ruleset lists in Service Manager, and it can be used in all user-created policies.

Edit a Rule Set

You can edit any user-defined ruleset regardless of whether it is included in a policy that is in use. If edits to the ruleset are saved, all services or resource monitors using related policies reevaluate their health that is based on the edit, which results in a change in service health for the models.

Follow these steps:

1. Open the Service Editor.
2. Click the Rule Sets tab, select the ruleset you want to edit, and then click Edit.
   The Edit Rule Set: <*Rule Set Name*> dialog appears.
3. Modify the name of the rule set in the Rule Set field.
4. (Optional) Use the arrow keys to rearrange rules.
5. Edit a rule by selecting the rule and clicking Edit.
   The Edit Rule dialog appears.
6. Change rule settings as needed and click OK.
   The Edit Rule dialog appears.
7. Click OK.
   Your edits are saved.

Delete a Rule Set

You can delete a rule set that is not part of a policy currently in use by a service or resource monitor.

Follow these steps:

1. Open the Service Editor.
2. Click the Rule Sets tab view, select the ruleset you want to delete, and click Delete.
   The rule set is removed from the rule sets list.

Creating and Managing Customers

This section describes the procedure to Create and Manage customers.

Customers and Customer Groups

Customers are DX NetOps Spectrum models that represent a person or organization that is associated with services or SLAs. The use of customer models enables you to track and monitor each customer service and SLAs.

A customer model status attribute reflects the status of the customer's services. The customer status can be equivalent to the worst service health value for all of the customer services. For example, assume that a customer is associated with services A, B, and C. If service A is up, service B is degraded, and service C is down, the customer status attribute has a value of severe to reflect a service down. If service C is restored to up and B remains degraded, the customer status attribute indicates a significant impact to reflect a service degraded. Customer icons within OneClick and the Service Dashboard indicate the value of the customer status attribute. No alarms are associated to changes in customer status. The visual indication is only the icon color.

Each customer model also has a criticality attribute with values ranging from low to high. Similar to the criticality of a service model, all or a portion of the customer model's criticality is added to the impact of any resource alarms that are affecting the customer's services. This confirms that alarms impacting highly critical customers have a high impact value.

Customer models can be added to customer groups. The customer group model provides a way to organize similar customers. The condition of a customer group can be equivalent to the worst status of all customers within the group. No alarms are associated to condition changes for the customer group model, the only visual indication is icon color.

Customers and customer groups you create appear under the Customers tab in the Service Editor, in Service Dashboard, and in OneClick under Service Management in the Navigation panel.

Create a Customer

A customer identifies a person or an organization that is associated with a service, an SLA, or both. In addition to the Criticality and Security String parameters included for all DX NetOps Spectrum models, it includes customer identity and contact information and other fields in which you can enter additional information.

Follow these steps:

1. Open the Service Editor.
2. Click the Customers tab.
   A list of customers that are created in Service Manager appears.
3. Click Create Customer.
   The Create Customer dialog appears.
4. Complete the required fields (denoted by asterisks).
5. (Optional) Specify the customer group (using the Customer Group tab) to which you want to add the customer and click Create.
   The customer is created.

Customer Criticality and a Service's Outage Alarm Priority

A customer criticality is factored into the impact calculation of any alarms that cause a service outage for one of the customer services. For example, Customer A has a Medium criticality value of 15 and Customer B has a Low criticality value of 5 and both Customer A and B are associated with services that have high criticality values and are down.

The root cause alarm for the outage on Customer A service has an increased impact of + 30 for the service and + 15 for the customer. The root cause alarm for Customer B service has an increased impact of +10 for the service and +5 for the customer.

Because Customer A has a higher criticality value, the alarms that affect Customer A service have a higher impact. Although both example alarms affect high criticality services, the root-cause alarm that affects Customer A service has a higher impact value. If the alarm view in OneClick is ordered by alarm impact, the alarm affecting Customer A can appear higher in the alarm table.

Create a Customer Group

Service Manager lets you organize customers in groups in any way that meets your requirements for tracking and managing customers.

Follow these steps:

1. Open the Service Editor.
2. Click the Customers tab and click Create Group.
   The Create Customer Group dialog appears.
3. Enter a Customer Group Name, the landscape where you want to create the group, and, optionally, a security string.
4. Under Group Location, select the group in which you want the new customer group saved.

   NOTE
   By default, Service Manager saves all customers and customer groups under the customer manager model on the landscapes where the customer model or customer group model exists.
5. Click Create.
   The customer group is created.

Edit Customer Settings

You can edit customer settings as required.

Follow these steps:

1. Open the Service Editor.
2. Click the Customers tab, select the customer that you want to edit, and then click Edit.
   The Edit Customer dialog appears.
3. Modify the settings:
   – Under the Contact Information tab, you can edit all contact information except the landscape where the customer was created.
   – Under the Customer Group tab, you can move the customer to a new location.

   Click OK.
   The customer settings are edited.

Edit a Customer Group

You can edit customer group settings as required.

Follow these steps:

1. Open the Service Editor.
2. Click the Customers tab, select the customer group that you want to edit, and click Edit.
   The Edit Customer Group dialog appears.
3. Modify the settings.

   NOTE
   You can modify the group name and security string. You cannot modify the landscape where the group was created.
4. Click OK.
   The customer group is edited.

Move a Customer or a Customer Group

As your customer list grows, you can reorganize your customers and customer groups by moving them from their current locations to new locations. When you move a customer group, you also move its customers with it.

Follow these steps:

1. Open the Service Editor.
2. Click the Customers tab, select the customer or customer group you want to move.
3. Drag and drop the customer or customer group to the new location.
   The customer or customer group is moved.

Delete a Customer or a Customer Group

You can delete customers and customer groups you no longer use. When you delete a customer group, you can select to delete or retain its customers.

Follow these steps:

1. Open the Service Editor.
2. Click the Customers tab, select the customer or customer group you want to delete, and then click Delete.

   NOTE
   If you are deleting a customer group, you are prompted to retain the group customers or delete them with the group.
3. Respond to the confirmation message that appears to complete the deletion.
   The customer or customer group is deleted.

Associate a Service or an SLA with a Customer

Service Manager lets you associate customers with services and SLAs. Verify the following benefits:

- You can track services and SLAs associated with customers in Service Dashboard, Service Editor, and OneClick
- You can generate service and SLA reports about specific customers with DX NetOps Spectrum Report Manager.

Follow these steps:

1. Open the Service Editor.
2. Click the Customers tab, select the customer that you want to associate with a service or SLA.
3. Click the Services or SLAs tab and click Select Customer Services or Select Customer SLAs.
   The Select Services or Select SLAs dialog appears.
4. Move the services or SLAs you want to associate with the customer from the Available Services or Available SLAs list to the Customer Services or Customer SLAs list (Do the opposite to remove services or SLAs).
5. Click Ok,
   The service or SLA is associated with the customer.

   NOTE
   Only the real-time health of services impact the status of associated customers. The customer status attribute indicates the real-time status of a customer in terms of any impacted services. Changes to the SLA status of SLA models associated to a customer does not alter the customer models status as the SLA status does not indicate a real-time value.

Creating and Managing Service Level Agreements

About Service Level Agreements

DX NetOps Spectrum represents a service level agreement or operational level agreement with an SLA model. A Service Manager SLA model or SLA, incorporates measurable provisions which are defined by the service or operational level agreement. These provisions are implemented as service models within DX NetOps Spectrum which are in turn monitored by the SLA model. The SLA monitors the real-time health of each associated service, and records outage time when the service is down. The recorded time is compared against a number of thresholds to determine the status of the SLA for a given period. SLA models can be created as individual SLAs or from preconfigured SLA Templates.

Guarantees

Each SLA specifies one of the more guarantees. A guarantee is a DX NetOps Spectrum model that represents and monitors a provision of the SLA. Each guarantee is associated with a service or resource monitor. The guarantees record service outage time and compares the amount of recorded time to a threshold specified by the user. A guarantee is considered violated if the amount of accumulated service downtime exceeds the threshold that is specified by the users. If the guarantee has not recorded any outage time, or the recorded outage time is less than the threshold, the guarantee is considered the complaint. The status of an SLA equals to the status of its worst guarantee. In other words, if any SLA's guarantee is violated, the SLA is violated. The status of an SLA is always expressed in terms of a period. Once an SLA is violated, it remains violated during the current period, unless outages contributing to the violation are edited in such a way that recorded outage time is removed from the guarantee.

Many SLAs include some form of an availability guarantee. For example, an SLA can stipulate that the service is guaranteed to be available 99.9 percent of each month. These statements define both the availability threshold and the period for which the threshold applies.

SLAs commonly include performance or response time provisions. For example, an SLA can state that service response time is 100 ms or better 99.9 percent during the hours of 8 AM to 5 PM each weekday.

Service Manager provides both Availability and Response Time guarantees. Functionally availability and response time guarantees are similar in the way that outage time is recorded. The Availability guarantees offer three additional thresholds for mean time to repair, such as maximum outage time and mean time between failures. The distinction of availability that is compared to response time is used for organizational convenience rather than functional purposes, as an availability guarantee can be configured to perform identically to a response time guarantee.

An SLA can include as many guarantees (either availability or response time) as required, to monitor all measurable provisions in a service level agreement. The SLA is considered complaint when all of it guarantees are compliant, or violated if any of its guarantees are violated. Guarantees and SLAs can also have a status of warned or at risk if a guarantee has record service outage time such that a violation is likely to occur. When the status of the SLA changes to a warned or violated state, an alarm is generated on the SLA model. This alarm remains on the model until the SLA period ends, or a user-initiated outage edit causes the recorded time to fall below the threshold.

Guarantee thresholds can be configured in one of two ways. The first and most common is by the percentage of availability. When configuring percentage-based thresholds a user specifies is the desired availability for the guaranteed service. Users can also configure guarantee thresholds by the number of seconds of outage time not to be exceeded for the period. Configuring a threshold that is based on a set number of seconds can eliminate some of the variability found with percentage-based thresholds for monthly periods.

Regardless of how the threshold is configured, the guarantee determines its status that is based on the amount of recorded outage time compared to the amount of allowable outage time for the period. Consider an SLA that stipulates that a service must be available 99.5 percent of the time for the SLA period. The SLA must include a guarantee that specifies the availability threshold (99.5 percent). Stated another way, the service cannot be down more than 0.5 percent of the total time for the SLA period, in this case, a calendar month. For a thirty-day month, out of 720 available hours, the service cannot be down more than 3.6 hours.

In addition, the SLA stipulates that no individual service outage can exceed 15 minutes, and the average time to repair an outage cannot exceed 10 minutes. These statements indicate that one addition to the overall threshold of 99.5 percent, the guarantee can also specify the threshold for MOT (maximum outage time), and MTTR (mean time to repair). If any of the guarantees thresholds are violated the guarantee, and likewise the SLA is considered violated for the period.

Period

A *SLA period* is the interval for which the guarantees can compute their status. The most common SLA periods are monthly. The guarantees that are specified for the SLA records time and calculates status on a monthly basis, with a specific start and end time. For example, if the SLA period is monthly, the first period begins at midnight on the first of the month, and end at midnight on the first day of the following month. The next SLA period begins immediately after the current SLA period ends.

In addition to the overall SLA period, some service level agreements can define specific hours that a guarantee is relevant for. For example, the SLA may state availability of 99.9 percent from 8 AM to 5 PM, Monday through Friday. These time frames within the period are named *business hours*.

Business hours can also be described as the times when the guarantee is active. A guarantee does not record outage time for service outages occurring outside of its business hours. Percentage-based guarantee thresholds are also calculated specifically to include only the time for which the guarantee is active. Therefore, if a guarantee defines a percentage-based threshold then the business hours has far less allowable outage time than a guarantee that specifies the same threshold, but no business hours.

SLA Considerations

SLA status must be accurate based on the stipulations of the guarantee in the SLA. In most cases, you can notice that not all service outages can impact SLA. One of the most common problems that users can encounter when using SLA models is guarantees recording outage time for issues that are not guaranteed by the SLA. The reason for this is that services designed for accurate real-time monitoring generally do not lend themselves well to SLAs.

When designing a service to encompass all possible fault scenarios, you can build a vast set of resource monitoring capabilities. SLAs on the other hand tend to be focused on specific types of service outages. For example, consider a critical application service. For real-time monitoring, you may want to understand the availability of the physical servers, the critical processes, system resources, network connections, application, and network response time. For accurate real-time management, the service must consider a wide range of potential resource faults.

In support of the application service, the server team has an SLA that stipulates the physical servers, available 99.9 percent of the time during regular business hours. The server team is responsible for the physical servers, but not for the applications running on them, or the network which provides access to them. So, the availability guarantee for the server teams, SLA should only record time if there is a failure of the server. SLA should not record time if there is a failure of the application or a network failure that prevents access to the server.

Therefore, the design of a service that is designed for real-time monitoring is likely to be different from a service that is designed to isolated specific resource faults for an SLA. Let us look at a couple of aspects of the previous scenario, and consider how service design can be different.

First, consider how the availability of the physical servers is monitored in CA Spectrum. If contact is lost to the server, then the server can be considered not available. In terms of SLA; however, it is not enough to say that contact is lost, the SLA must consider the distinction between a server outage and a network outage preventing access to the server. In CA Spectrum, the availability guarantee must only record outage time if the server is red, and must not record outage time if the server is gray. If the application service is designed for real-time monitoring, it is unlikely that its designers included resource monitoring components to distinguish between these two types of outages.

Next, consider how server faults must be distinguished from application or process failures. The server team is not responsible for the processing running on the server itself. To begin any monitored processes outages probably should be isolated to the process model and not the server. Other system resource-related failures may or may not be the responsibility of the server team, for example, high CPU is likely the result of the applications on the server, failures of local file systems on the other hand could be the responsibility of the server team. Again it is unlikely the original designers of a service for real-time monitoring would have considered making these distinctions in their service design.

You can use the two techniques to verify that SLAs accurately monitor the services they guarantee.

The first option is to simply create a parallel set of services that specifically monitor the components that are guaranteed by the SLA. These services are built in addition to services designed for real-time monitoring. They can monitor a smaller set of resources, possibly use different service policies, and do not generate alarms. SLA-specific services can also define periods of maintenance which makes them inactive outside of the SLA specified business hours. This is probably the easiest technique, but it does not always scale well. Consider the example of the server team SLA, imagine that the application team and the network also have SLAs. For one logical application service, you can easily produce four service hierarchies to support, one for real-time monitoring, one server-specific, one application-specific, and one network specific. The implementation and maintenance of these services can be daunting.

The second option, an alternate technique, is to verify that all services are built with a modular design. It is valuable to remember that the use of resource monitors makes it easier to extend the monitoring capability of service. This is true not only when adding new resources, but also when adding support for SLAs. Resource monitors are used to isolating specific faults such that guarantees can be associated with the specific resource monitor. If well-designed, the same service can be used for multiple SLAs, and still used for real-time monitoring.

Create an SLA

When you create an SLA, name it, specify the period that it is in effect, and the guarantee thresholds for the services or resource monitors that are monitored by the SLA. After you create an SLA, associate it with an SLA customer. You can also modify an SLA as necessary when service delivery requirements change.

Consider the following points before creating an SLA:

- An SLA can have multiple guarantees.
- You can associate a single service to an SLA, but guarantees can be created for that service, or any of it sub services or resource monitors.
- You may need to create new service components to isolate the specific set of faults for which the guarantee is responsible.
- SLA models reside on the same landscape as the service the SLA is associated to.

Follow these steps:

1. Open the Service Editor.
2. Click the SLAs tab and click Create.
   The Create SLA dialog appears.
3. Specify the following SLA properties:
   - **SLA Name**
     Identifies the SLA model. OneClick lists SLA names under the Service Management - SLAs category in the Explorer view for each landscape where you have created SLAs. You can use duplicate names for SLAs. However, to facilitate filtering when searching through a lengthy list of identically named SLAs, provide different descriptions for each SLA.
   - **Control**
     Specifies whether the SLA is activated during the current SLA period (Active), the default setting, or the next period (Inactive Until Next Period).

     NOTE
     If you activate an SLA during an SLA period, Service Manager does not prorate allowable outage time. The service that is associated with the SLA is allowed the amount of down or degraded time that is specified for the entire period. For example, if an SLA allows five hours of outage for a 30-day month and activate an SLA on 15th of the month, the service that is associated with the SLA can be unavailable for up to five hours over the course of remaining 15 days. In terms of an availability obligation, this situation allows for twice as much service outage than a service customer would expect over the remaining 15-day period. In this case, you can modify the availability threshold for the SLA guarantees to an amount of time for the partial period proportional to the entire period. For example, in the previous example, you could change the availability threshold to two and half hours for the partial 15-day period.
   - **Description**
     (Optional) Describes the SLA. You can enter unique descriptions for SLA that have the same name to facilitate finding each SLA using a list filtering capabilities.
   - **Security String (Optional)**
     Identifies the security string for the SLA model. The security string secures access to the SLA model in DX NetOps Spectrum. For more information, see the OneClick Administration section.
   - **Notes**
     (Optional) Includes any information about the SLA you want to enter not covered in the Description field.
   - **Expiration Date**
     Specifies the date a recurring SLA period expires. Check the box, and enter the date in the field that appears. If the date falls within an SLA period, the SLA stays in effect to the end of the period.
   - **Period**
     Specifies the interval during which the SLA is in effect. Select the period from the drop-down list, or click Create to create an SLA period.

4. Configure one or more guarantees for the SLA.
5. Associate a service to the SLA by moving it from the Available Services list to the SLA Services list.
6. Click Create.
   The SLA is created.

Create an SLA from an SLA Template

An SLA template is an SLA configuration that you create and save as a template and from which you can create multiple SLAs. When you create an SLA from an SLA template the SLA inherits the template settings. You can tailor SLA settings that are inherited from a template except for the following items:

- Guarantees -- The guarantees in an SLA template can only be modified in the SLA template workspace, and any changes to the guarantees extend to only those SLAs created from the template that have been kept in sync with the template.
- Period -- The period in an SLA template can only be modified in the SLA template workspace, and any changes to it extend to only to those SLAs created from the template that have been kept in sync with the SLA template.

Consider the following information before you create an SLA from an SLA template:

- When you create an SLA from an SLA template, you can select whether to keep the SLA in sync with (or associated with) the SLA template. If you select to keep the SLA in sync with the template, all changes to the template guarantees and SLA period extend to the SLA. If you select not to keep the SLA in sync with the template, changes to the template guarantees and the SLA period do not extend to the SLA created from the template.
- The settings for any SLA created from an SLA template that has been deleted convert to local settings.

Follow these steps:

1. Open the Service Editor.
2. Click the SLAs tab and then click Create From Template.
   The Select SLA Template dialog appears.
3. Select the SLA template that you want to use to create the SLA and click OK.
   The Create SLA From Template dialog appears.
4. Configure the settings that can be edited as required.
5. Select or clear the option to keep the SLA in sync with the template. You can also clear the option anytime after you have created the SLA from the template.
6. Click Create.
   The SLA is created from an SLA template.

Guarantee Types

You can specify one of the following guarantees for a service or a resource monitor:

- **Availability**
  You can specify an availability threshold that is expressed as a percentage of the period for which the service is available or the amount of time in seconds that the service must not exceed for the period. This guarantee interprets the time that a service or a resource monitor service health value is down as outage time. You can also specify these supplemental thresholds when you specify an availability guarantee:
  - **Mean Time Between Failure (MTBF)**
    (total time between failures) / (total number of failures - 1)
    If the interval between failure falls below this value, the status of the guarantee changes to At Risk and DX NetOps Spectrum generates a Major (Orange) alarm for the SLA. Any time the interval between failures exceeds this value, DX NetOps Spectrum clears the At Risk alarm. If the guarantee is At Risk at the end of the period, DX NetOps Spectrum generates a Critical (Red) alarm for the SLA.
  - **Mean Time to Repair (MTTR)**

(total outage time) / (total number of outages)
If the average outage duration exceeds this value, the status of the guarantee changes to At Risk and DX NetOps Spectrum generates a Major (Orange) alarm for the SLA. Any time the average duration falls below this value, DX NetOps Spectrum clears the “At Risk” alarm. If the guarantee is At Risk at the end of the period, DX NetOps Spectrum generates a Critical (Red) alarm for the SLA.

- **Maximum Outage Time (MOT)**
  If the service or resource monitor has an outage that exceeds this value, The SLA is violated, and DX NetOps Spectrum generates a Critical alarm for the SLA.

- **Response Time**
  Lets you define a threshold for monitoring a service or resource monitor that determines its service health from the results of response time (or performance) tests. This guarantee interprets the time that a service or a resource monitor service health value is degraded as outage.

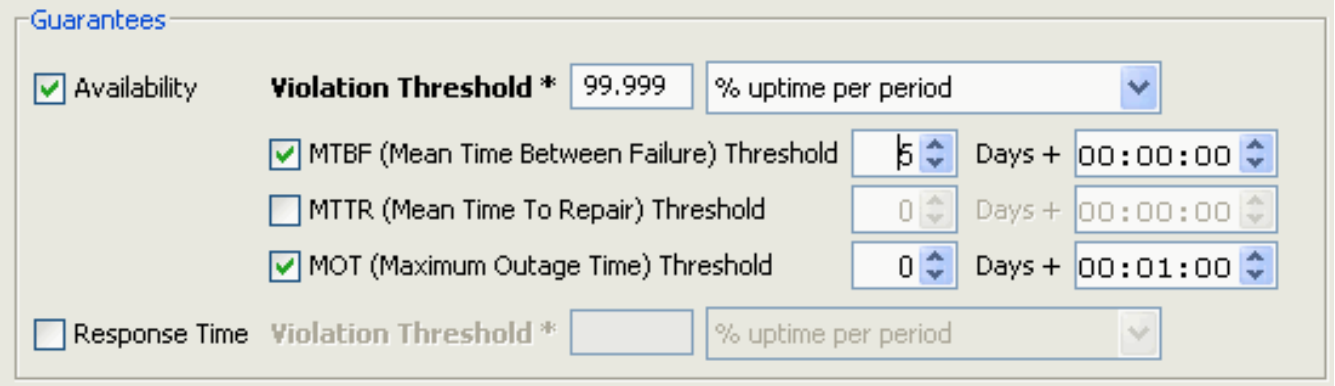
Create a Guarantee for a Top-Level Service

You can create a guarantee for a top-level service, which lets you create a guarantee for an entire service but not to any sub-services and resource monitors.

Follow these steps:

1. Select a threshold type: Availability or Response Time:
   - Specify a Violation Threshold value for % uptime per period or seconds of outage time per period for the threshold(s). The default uptime percentage is 99.9.
   - If you specify an availability threshold, specify values for any or all of the MTBF, MTTR, and MOT thresholds.

   Service Manager provides a name for the guarantee using this format: *Threshold type - SLA name*. For example: Availability - Web Service SLA. The following is an example availability guarantee that includes an uptime threshold and MTBF, MTTR, and MOT supplemental thresholds:



Guarantee is created for a top-level service.

Create a Guarantee for a Service, Sub-Service, or Resource Monitor

You can create a guarantee for a service, a sub-service, or a resource monitor.

Follow these steps:

1. Open the Service Editor.
2. Click the SLAs tab, select the SLA for which you want to create the guarantee, click the Guarantees tab, and then click the Create button.
   The Create Guarantee dialog appears.

3. Specify the following guarantee properties:
   - **Guarantee Name**
     Identifies the guarantee model.
   - **Control**
     Specifies whether the guarantee is Enabled or Disabled during the current SLA period. When you disable a guarantee, it does not accumulate outage time during the SLA period. Generally a guarantee is created in the disabled state if SLA and the services it guarantees are still be defined. However, there may be scenarios where due to alterations of a specific service it makes sense to periodically disable one of more guarantees.
   - **Guarantee Type**
     Specifies the type of guarantee: Availability or Response Time.
   - **Outage Type**
     Specifies whether the guarantee accumulates Down or Degraded time. response time guarantees accumulate only degraded time, while availability guarantees can be configured to accumulate either down time or degraded time.
   - **Accumulation Method**
     Service Manager provides two accumulation methods: Straight time and Per Resource.
     Straight time means that the guarantee records outage time that corresponds exactly to service outage time. One minute of service outage time produces one minute of outage time for the guarantee. Straight time is almost always the appropriate configuration for a guarantee. Under rare circumstances per resource configurations are used to record time that is based on the resources contributing to the service outage.
     Per resource guarantee is only used for a specific type of SLA which actually guarantees specific availability of individual service resources as opposed to the service as a whole. The result is the guarantee can record outage time in excess of the actual service outage time. Once again per resource guarantees are specialized and uncommon.
   - **Violation Threshold *n*% uptime per period or seconds of outage time per period**
     Specifies a service availability threshold that if during the current SLA period results in a critical alarm on the SLA model. The default uptime percentage is 99.9.
   - **(Optional) Generate warning alarm after accumulating *n*% of allowed outage time**
     The warning percentage is a percentage of the violation threshold in terms of allowable outage time. When a guarantee becomes warned a major alarm is generated on the SLA. This option lets you take action before the SLA becomes violated.
4. Select the Service or Resource Monitor tab, and then select a service or resource monitor by moving it from the Available Services and Groups box to the Service or Group being Measured box.
5. (Optional) Specify "business hours" intervals during which you want the guarantee in effect.

   NOTE
   By default a guarantee is always active, that means it records outage time 24x7.
6. Click Create.
   The guarantee is created.

Specify Business Hours for a Guarantee

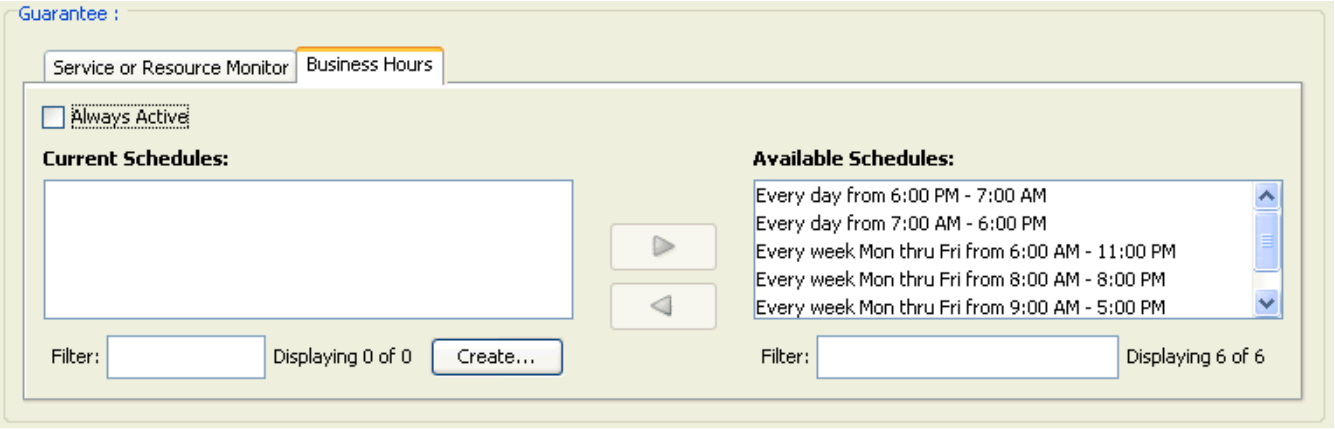
If you are creating a guarantee for an SLA that measures service availability or performance for a particular portion of a day, or “business hours,” you can identify those hours in the guarantee. This means that the guarantee accumulates service outage time only during the business hours, and the SLA availability threshold applies only to the business hours.

You can also specify multiple intervals for a guarantee, as long as they do not overlap. For example, if you want a guarantee to watch a service or resource monitor from 7 AM to 5 PM on Monday, Wednesday, and Friday, 6 AM to 6 PM on Tuesday and Thursday, and continuously throughout the weekend. You can specify these three schedules in the guarantee.

For guarantees which specify a percentage of availability, it is important to understand that the availability calculation is made against the time the guarantee is active. This means that a guarantee defining business hours can have less available outage time than one with the same availability threshold that does not define business hours.

Follow these steps:

1. Open the Service Editor dialog.
2. Click the SLAs tab, select the SLA that contains the guarantee you want to modify, and click the Guarantees tab.
3. Select the guarantee that you want to specify business hours for and click Edit.
4. Click the Business Hours tab, and clear Always Active, as shown in the following figure:



NOTE
If you want the guarantee in effect continuously throughout the SLA period, click the Business Hours tab and select Always Active (default).

5. Select and move one or more intervals from the Available Schedules list to the Current Schedules list.
6. (Optional) If you require a custom interval, take the following steps:
   a. Click Create.
   b. Configure the interval and click OK.
      The custom interval is added to the Available Schedules list.
   c. Select and move the custom interval from the Available Schedules list to the Current Schedules list.
7. Click OK.
   Business hours are specified for the guarantee.

Edit a Guarantee

You can edit a guarantee anytime after you create it; however changes to thresholds or business hours are not recommended.

Consider the following points before editing a guarantee:

- Modified thresholds take effect during the current period, this may not be desirable.
- A guarantee begins a new down or degraded time tally when you change the service or resource monitor with which it is associated during the current period.
- A guarantee modifies its outage time tally during a period if an outage has occurred and ended during the period and the outage has had its status modified (to exempt for example) during the period.

Follow these steps:

1. Open the Service Editor.
2. Click the SLAs tab, select an SLA from the list of SLAs, click the Guarantees tab (in the lower panel) and click Edit.
   The Edit Guarantee dialog appears.
3. Edit the settings, as described in Create a Guarantee for a Service, Sub-Service, or Resource Monitor and click OK.
   The guarantee is edited.

Delete a Guarantee

You can delete a guarantee for an SLA as your requirements for the SLA change.

WARNING
Delete with caution. Deleting the only guarantee for an SLA renders it inoperative.

Follow these steps:

1. Open the Service Editor.
2. Click the SLAs tab, and then select the SLA from which you want to delete a guarantee.
3. Click the Guarantees tab, select the guarantee that you want to delete, and then click Delete (in the lower panel).
4. Respond to the confirmation message that appears to complete the deletion.
   The guarantee is deleted.

Create an SLA Period

Service Manager provides two default SLA periods that commence at the following times for the time zone where the SpectroSERVER is located:

- On the first of every month at 12:00 AM
- On the 15th of every month at 12:00 AM

You can specify one of these periods when you create or edit an SLA, or you can create and specify a custom period that meets your particular service contract requirements. Service Manager saves the periods that you create so you can use them for other SLAs.

Follow these steps:

1. Click the Create button next to the Period field in the Create SLA dialog or the Edit SLA dialog.
   The Create Period dialog appears.
2. Specify a period, and optionally a description, and then click OK. Repeat as necessary to create and save more periods.
   The SLA period is created.

Edit an SLA

Periodically it is necessary to alter the configuration of a SLA model. Service Manager allows certain edits, but you can take care in the types of changes that are made. If the required change involves alteration of guarantee thresholds, business hours or the SLA period it is recommended that you do not make these edits to an active SLA. The period and guarantee thresholds are the essence of the SLA. If the guarantees stipulations are changed, it implies that a new SLA is available.

Therefore it is important that the real-time status of an SLA, and its historical reported status are based on complete periods with consistent thresholds. Consider how confusing data may look if an SLA recorded that same amount of outage time across two periods, but is compliant for one period and violated for the next due to a threshold change.

It is recommended to set an expiration date of the SLA to correspond with the end of the current period, instead editing a standing SLA. Later create a SLA with the new threshold settings, and set its control value to Inactive Until Next Period. This confirms that the current SLA completes the period with its existing configuration and the new SLA takes over seamlessly at the beginning of the next period.

If you change the service that is associated with the SLA, you can see the following output:

- Service Manager resets the start time of the SLA.
- All of the SLA guarantees go to the Initial (Blue) state, which means that you must associate a new service or resource monitor to each guarantee.

Follow these steps:

1. Open the Service Editor.
2. Click the SLA tab, select the SLA you want to edit from the list, and then click Edit.
   The list field, Template Name, indicates whether the SLA was created from an SLA template. If it was, the field lists the source SLA template name.
   The Edit SLA dialog appears. If you select an SLA created from an SLA template, the dialog includes the Keep in sync with the template selection box. You can clear the sync option if you want to disassociate the SLA from its source SLA template and edit all SLA fields. Otherwise, you can edit only those settings that are not managed in the source SLA template.
3. Edit the settings, as described in Create an SLA and click OK.
   The SLA is edited.

Delete an SLA

You can delete an SLA anytime after you create it. However, you would typically not delete an SLA that is actively monitoring a service.

> **WARNING**
> Be cautious when deleting SLAs because you may inadvertently delete an SLA that is actively monitoring a service.

Follow these steps:

1. Open the Service Editor.
2. Click the SLAs tab, select the SLA you want to remove from DX NetOps Spectrum, and then click Delete.
3. Respond to the confirmation message that appears to complete the deletion.
   The SLA is deleted.

Associate a Customer with an SLA

Service Manager lets you associate one or more customers with an SLA to help you track and manage SLAs and customers. The status of the customer model is not impacted by the status of the SLAs associated to it. You can generate SLA reports with DX NetOps Spectrum Report Manager based on SLA-customer associations.

Follow these steps:

1. Open the Service Editor.
2. Click the SLAs tab, select the SLA you want to associate with a customer, click the Customers tab, and then click Select SLA Customers.
   The Select Customers dialog appears.
3. Move the customer that you want to associate with the SLA from the Available Customers list to the Customers that use this SLA list and click OK.
   The customer is associated with an SLA.

SLA Templates

An SLA template includes configuration settings that are inherited by the SLAs that you can create from the template. It lets you create multiple SLAs with similar settings for different customers and services without having to configure each SLA individually. The use of SLA templates is common in traditional service provider environment that offers similar SLAs to multiple customers. The SLA template also allows you to make changes or additions to all associated SLAs by editing the template itself. Although this can be a convenient feature, be cautious when making edits guarantee thresholds or business hours setting.

The following SLA template settings cannot be edited in an SLA that has been created from a template while the SLA in sync with the template:

- Period
- Guarantees

All other inherited settings can be modified in the SLA. All settings can be modified, however, if an SLA is disassociated (in sync option deselected) from its parent template.

The guarantees that you create for SLA templates are referred to as guarantee templates. You create and manage guarantee templates in the SLA Template workspace the same way you do with guarantees in the SLA workspace.

Create an SLA Template

You can create as many SLA templates as you require, but the templates must have unique names.

Follow these steps:

1. Open the Service Editor.
2. Click the SLA Templates tab and click Create.
   The Create SLA Template dialog appears.
3. Specify settings for the SLA template.

   NOTE
   SLA Templates are often created with a control of Inactive. It lets you instantiate the actual SLAs in the inactive state, and activate them when appropriate.
4. (Optional) Configure basic guarantee settings for the template. You can specify more detailed guarantee settings or a new guarantee for the template (such as guarantee template) after you create the template.
5. Click Create.
   The SLA template is created.

Edit an SLA Template

You can edit an SLA template anytime before or after it is in effect.

Consider the following information before editing an SLA template:

- Changes to the period and guarantees in a template extend to all SLAs created from it that are in sync with the template. This may not be desirable.
- When you delete a template, the associations between the SLAs created from it that are in sync with the template are severed. This means you can edit all period and guarantee settings in the SLAs that were once managed exclusively in the former template.

Follow these steps:

1. Open the Service Editor.
2. Click the SLA Templates tab, select the template that you want to edit from the list, and click Edit.
   The Edit SLA Template dialog appears.
3. Edit the settings, as described in Create an SLA, and click OK.
   The SLA template is edited.

Delete an SLA Template

You can delete an SLA template anytime after you create it. When you delete an SLA template, all SLAs created from the template that are kept in sync with it become fully editable.

Follow these steps:

1. Open the Service Editor.

2. Click the SLA Templates tab, select the template from the list you want to remove from DX NetOps Spectrum, and click Delete.
3. Respond to the confirmation message that appears to complete the deletion.
   The SLA template is deleted.

Guarantee Templates

You can create guarantee templates for SLA templates. You can edit guarantee template settings, and those changes extend to all SLAs created from the SLA template that includes the guarantee template.

You can modify guarantee templates only from the SLA Templates workspace. You cannot access guarantee templates from the Guarantees tab in the SLA workspace.

You cannot specify watched services or Resources Monitors for guarantee templates. You can associate the guarantee to services and resource monitors you specify when you create an SLA from an SLA template that includes the guarantee template.

NOTE
Be cautious when editing guarantee templates for the reasons that are outlined in the Edit an SLA section.

Create a Guarantee Template

Guarantee templates can be created in the Create SLA Template dialog or in the Create Guarantee Template dialog. This section describes the latter method.

Follow these steps:

1. Open the Service Editor.
2. Click the SLA Templates tab, select the SLA template for which you want to create the guarantee template, click the Guarantee Template tab, and then click Create.
   The Create Guarantee Template dialog appears.
3. Configure settings, and then click Create.
   The guarantee template appears under the Guarantee Template tab for the SLA template and is inherited by all SLAs created from the template.

Edit a Guarantee Template

You can edit a guarantee template any time after you create it. Because they are changed within the context of an SLA template and all SLAs created from it that are in sync with the template are also changed. Be cautious when editing guarantee templates for the reasons that are outlined in the Edit an SLA section.

Follow these steps:

1. Open the Service Editor.
2. Click the SLA Templates tab and select the SLA Template that includes the guarantee template you want to edit from the list of templates.
3. Click the Guarantee Templates tab to display guarantee templates for the selected SLA template.
4. Select the guarantee template that you want to edit from the list, and then click Edit (in the lower panel).
   The Edit Guarantee Template dialog appears.
5. Edit the settings as necessary as described in Edit a Guarantee, and click OK. For more information, see Create a Guarantee for a Service, Sub-Service, or Resource Monitor.

Delete a Guarantee Template

You can delete a guarantee template as your requirements change.

WARNING
Delete with caution. Deleting the only guarantee template for an SLA template renders the SLAs created from and in sync with the template inoperative.

Follow these steps:

1. Open the Service Editor.
2. Click the SLA Templates tab and select the SLA template from the list of templates from which you want to delete a guarantee template.
3. Click the Guarantee Templates tab, select the guarantee template that you want to delete, and then click Delete (in the lower panel).
4. Respond to the confirmation message that appears to complete the deletion.
   The guarantee template is deleted.

Creating Service Management Components with Modeling Gateway

You can create service management component models using the DX NetOps Spectrum Modeling Gateway Toolkit to define service component model configurations in XML input files and import the files into DX NetOps Spectrum. Defining service management models with the Modeling Gateway Toolkit and importing them into DX NetOps Spectrum instead of creating them with Service Editor is advantageous in the following ways:

- Modeling Gateway lets you define new models in bulk. Once you have created a particular service management model, you can use the XML input file for the model as a template for creating other models of that type.
- You can edit service management models that are imported through Modeling Gateway either by using the Service Editor or by simply editing and re-importing the original XML file.
- There can be an opportunity to automate the creation of service management models by producing an xml file for import that is based on an external data source. You can verify the Modeling Gateway capabilities and prerequisites. For more information, see the Modeling Gateway Toolkit section.

About the XML Framework

The hierarchy models (SM_Service_Mgt, SM_ServiceMgr, SM_SLA_Mgr, CustomerManager) must be arranged and named in the XML input file according to the following example.

The following example illustrates the basic framework of the XML input file. Each service management component is added within the appropriate section of the file. Service models can be created within the SM_ServiceMgr block, or within other services. Customer models and Customer Group models are created within the CustomerManager block. Finally the SLA models are created within the SLA_Mgr block.

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE Import SYSTEM ".import.dtd">

<Import>

<SM_Service_Mgt
      name="Service Management"
      containment_relation="SlmHasServiceComponent">

    <SM_ServiceMgr
        name="Services"
```

```
        containment_relation="SlmContains">

    </SM_ServiceMgr>

    <SM_SLA_Mgr
        name="SLAs"
        containment_relation="SlmContainsSLAs">

    </SM_SLA_Mgr>

    <CustomerManager
        name="Customers"
        containment_relation="Groups_Customers">

    </CustomerManager>

</SM_Service_Mgt>

</Import>
```

The following image displays the hierarchy that is represented in the OneClick Console.

WARNING
Although the name and containment_relation attribute values in the framework tags must be adhered to in the XML file you import, the individual service, resource monitor, customer, SLA, and guarantee models you import must have unique names. This differs from the OneClick client. Because modeling gateway uses name for uniqueness. Services with the same name are interpreted as the same model.

Service Models

Service models must be defined within the SM_ServiceMgr tag in the XML file that you import. For each landscape where Service Manager is installed, there is only one SM_ServiceMgr model with the model name, Services.

Each service model in a given landscape must be either SlmContains by the ServiceManager or SlmMonitors by another service model.

To use Modeling Gateway effectively, understand the mapping between policies and policy IDs. For example, if a service monitors other services or resource monitors, it can be configured with a service health policy (IDs 6-9). Also, you can determine a policy ID for a user-created policy by setting up the Policies table in the Service Policy Editor to display IDs.

Policies and Watched Attributes

When using Modeling Gateway to model services, the value for the MonitorPolicy_ID can be set to the id number associated with the policy the service uses. Service policy ids can be seen in the service policy editor by adding the Service Policy ID column to the service policies table. Out-of-the-box policies start at 1, user created policies start at 1000.

In addition to viewing policy ID in the service policy editor, you can see the policy id that is displayed at the following link.

http://<server>/spectrum/slm/ policyrep.jsp

Remember that when specifying the Monitor Policy_ID, verify that the AttrToWatch matches the attribute for the policy that you have selected.

NOTE
If your XML file creates a mismatch between a monitor policy ID and watched attribute (for example, watching Contact Status using policy ID 2), Service Manager puts the service in a defunct condition, which is reported as an outage and can be viewed in Service Dashboard. When a service becomes default an alarm is generated on the Service Management model. This alarm is major for services which are associated to an SLA, and minor for services which are not.

Example: Services That Monitor Resources Directly

The following XML document configures a service named “Test Service.” It monitors the contact status of two Cisco routers and generates a critical alarm if contact to either router is lost:

```
<!-- Each SpectroSERVER will have only one -->
<!-- SM_ServiceMgr model, named “Services”. -->


<SM_ServiceMgr
    name="Services"
    containment_relation="SlmContains">

    <SM_Service
        containment_relation="SlmMonitors"
        name="Test Service"
        AttrToWatch="Contact_Status"
        MonitorPolicy_ID="11"
        Criticality="10"
        Generate_Service_Alarms="true">
        <Device ip_dnsname="10.253.9.7" />
        <Device ip_dnsname="10.253.9.8" />
    </SM_Service>


</SM_ServiceMgr>
```

This relation associates a service with resources.

Enter either a SPECTRUM-provided Policy ID (1-21) or ID (1000, 1001, ...) for a policy created in Service Editor.

Enter device (resource) IP address or DNS name.

- The Services model contains (has an SlmContains relationship with) Test Service. This can make the Test service a direct child of the service manager. The Test service appears under the Services icon in the OneClick navigation panel.
- Test Service monitors (has an SlmMonitors relationship with) the 10.253.9.7 and 10.253.9.8 devices and watches their Contact_Status attributes using the Contact Status High Sensitivity policy.
- The value of Generate_Service_Alarms is true, indicating that when the service health value is down, degraded, or slightly degraded, DX NetOps Spectrum generates an alarm for the service.

Example: Services That Monitor Resources in Resource Monitors

The following XML document defines a service, XYZ Service, that monitors two other services (Core Routers and DNS) directly. In addition, the XYZ service defines three resource monitors by specifying SM_AttrMonitor elements. The first resource monitor XYZ Condition monitors the contents of the XYZ Network container. The second resource monitor XYZ Response Time monitors an SPM test call XYZ_RTM_1. Finally, the service also defines a resource monitor, XYZ Port Status which monitors an interface model.

```
<SM_Service
        containment_relation="SlmMonitors"
        name="XYZ Service"
        Criticality="25"
        AttrToWatch="Service_Health"
        MonitorPolicy_ID="8"
        Generate_Service_Alarms="true">

        <SM_AttrMonitor
            containment_relation="SlmWatchesContainer"
            name="XYZ Condition"
            AttrToWatch="Condition"
            MonitorPolicy_ID="2">

            <Topology_Container model_type="Network" name="XYZ                    Network Servers" />
        </SM_AttrMonitor>

        <SM_AttrMonitor
            containment_relation="SlmMonitors"
            name="XYZ Response Time"
            AttrToWatch="LatestErrorStatus"
            MonitorPolicy_ID="18">

            <RTM_Test name="XYZ_RTM_1" />
        </SM_AttrMonitor>

        <SM_AttrMonitor
            containment_relation="SlmMonitors"
            name="XYZ Port Status"
            AttrToWatch="Port_Status"
            MonitorPolicy_ID="15">

            <Port ip_dnsname="10.253.50.5"
            identifier_name="ifIndex"
            identifier_value="45" />
        </SM_AttrMonitor>

        <SM_Service name="Core Routers"/>
        <SM_Service name="DNS"/>

    </SM_Service>
```

Example: Using XML to Define a Service Template

If you encounter a scenario where many services share a common pattern or structure. You can define that structure in xml, and can use it as a common template. For example, you can build services to monitor a set of applications which are

all different, but have common service modeling components. You can define the structure and import as many service models as you need from it. Some of the data can be added for import, or you can create empty services and resource monitors. You can add resources to them using the OneClick client.

The following syntax shows an example of the xml for a small service hierarchy that define a set of reusable service and resource definitions. The TMPL text represents wildcard test that can be changed to a more meaningful name for each set of services to be imported.

For this example, you can see a service which includes monitoring capability for some application servers, and associated database servers. As well as some additional some response time and performance monitoring. This example is not intended to match any specific requirements, but serves as an example to create an xml template for a set of services with common requirements.

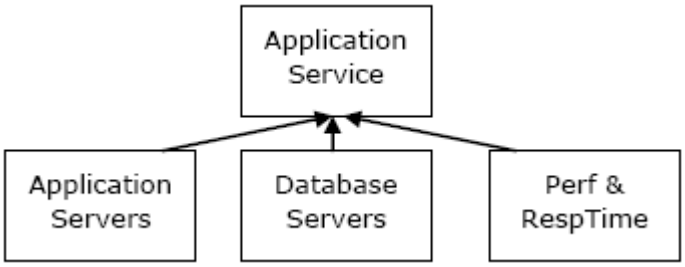
```
<SM_Service
    containment_relation="SlmMonitors"
    name="TMPL Application Service"
    Criticality="10"
    AttrToWatch="Service_Health"
    MonitorPolicy_ID="7"
    Generate_Service_Alarms="true">

    <SM_Service
        containment_relation="SlmMonitors"
        name="TMPL Application Servers"
        Criticality="10"
        AttrToWatch="Service Health"
        MonitorPolicy_ID="9"
        Generate_Service_Alarms="true">

        <SM_Service
            containment_relation="SlmMonitors"
            name="TMPL Application Server 1"
            Criticality="10"
            AttrToWatch="Service Health"
            MonitorPolicy_ID="7"
            Generate_Service_Alarms="true">

            <SM_AttrMonitor
                containment_relation="SlmMonitors"
                name="TMPL App Host 1"
                AttrToWatch="Condition"
                MonitorPolicy_ID="3"
                Cause_List_Control="2"
                      Special_Cause_List="0x1106f-0x11232">

                // Excludes all eHealth alerts

            </SM_AttrMonitor>

            <SM_AttrMonitor
                containment_relation="SlmMonitors"
                name="TMPL App Server 1 Critical Processes"
                AttrToWatch="Condition"
                MonitorPolicy_ID="3">
```

```
        </SM_AttrMonitor>

        <SM_AttrMonitor
            containment_relation="SlmMonitors"
            name="TMPL App Server 1 System Resources"
            AttrToWatch="Condition"
            MonitorPolicy_ID="3">

        </SM_AttrMonitor>

        <SM_AttrMonitor
            containment_relation="SlmMonitors"
            name="TMPL App Server 1 Connection"
            AttrToWatch="Response Time"
            MonitorPolicy_ID="19">

        </SM_AttrMonitor>

        <SM_AttrMonitor
            containment_relation="SlmMonitors"
            name="TMPL App Server 1 Performance"
            AttrToWatch="Condition"
            MonitorPolicy_ID="3"
            Cause_List_Control="1"
                  Special_Cause_List="0x1120a,0x11219">

            // Includes 2 specific eHealth alerts only
        </SM_AttrMonitor>

    </SM_Service>

</SM_Service>

<SM_Service
    containment_relation="SlmMonitors"
    name="TMPL Database Servers"
    Criticality="10"
    AttrToWatch="Service Health"
    MonitorPolicy_ID="6"
    Generate_Service_Alarms="true">

    <SM_Service
        containment_relation="SlmMonitors"
        name="TMPL Database Server 1"
        Criticality="10"
        AttrToWatch="Service Health"
        MonitorPolicy_ID="7"
        Generate_Service_Alarms="true">

        <SM_AttrMonitor
            containment_relation="SlmMonitors"
            name="TMPL DB Host 1"
```

```
            AttrToWatch="Condition"
            MonitorPolicy_ID="3"
            Cause_List_Control="2"
                  Special_Cause_List="0x1106f-0x11232">

            // Excludes all eHealth alerts

        </SM_AttrMonitor>

        <SM_AttrMonitor
            containment_relation="SlmMonitors"
            name="TMPL DB Server 1 Critical Processes"
            AttrToWatch="Condition"
            MonitorPolicy_ID="3">

        </SM_AttrMonitor>

        <SM_AttrMonitor
            containment_relation="SlmMonitors"
            name="TMPL DB Server 1 System Resources"
            AttrToWatch="Condition"
            MonitorPolicy_ID="3">

        </SM_AttrMonitor>

        <SM_AttrMonitor
            containment_relation="SlmMonitors"
            name="TMPL DB Server 1 Connection"
            AttrToWatch="Response Time"
            MonitorPolicy_ID="19">

        </SM_AttrMonitor>

    </SM_Service>

</SM_Service>

</SM_Service
    containment_relation="SlmMonitors"
    name="TMPL Application Performance & Response Time"
    Criticality="10"
    AttrToWatch="Service Health"
    MonitorPolicy_ID="7"
    Generate_Service_Alarms="true">

    <SM_Service
        containment_relation="SlmMonitors"
        name="TMPL Application Response Time"
        Criticality="10"
        AttrToWatch="Response Time"
        MonitorPolicy_ID="20"
        Generate_Service_Alarms="true">
```

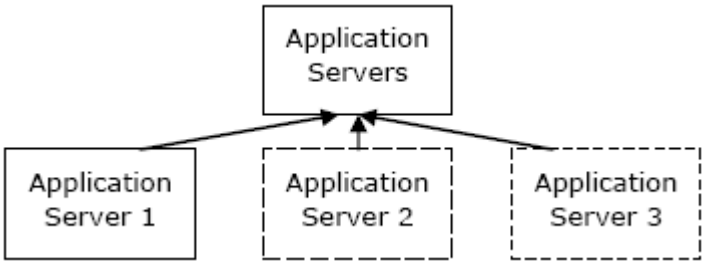
```
        <SM_Service>

        <SM_Service
            containment_relation="SlmMonitors"
            name="TMPL Application Performance
            Criticality="10"
            AttrToWatch="Condition"
            MonitorPolicy_ID="4"
            Cause_List_Control="1"
                  Special_Cause_List="0x1106f-0x11232">

                // Includes eHealth alerts only

            Generate_Service_Alarms="true">

        <SM_Service>

    </SM_Service>

</SM_Service>
```

Now let us review each component in the xml to understand its purpose and how it fits within the service hierarchy that is defined by the xml. At the top of the hierarchy we have the application Service which has three direct child services:



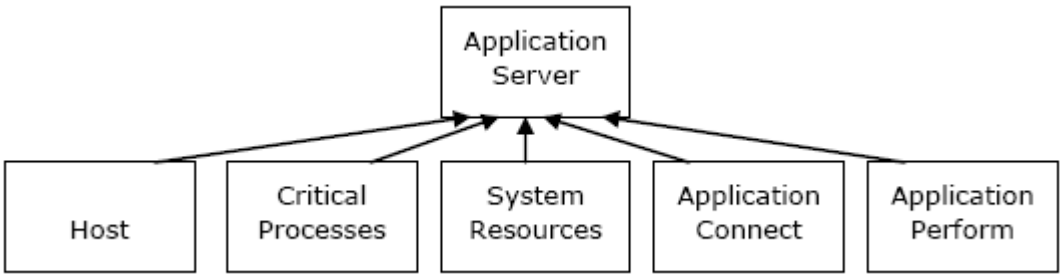
The Application Service monitors each of its child service with a high sensitivity policy. Therefore, the Application Service can have a service health equal to that of its worst direct resource.

The Application Servers service is designed to monitor child services which represent individual servers. Each server is represented by a service model with five resource monitors. The xml example contains only Application Server 1, but you could add this section for as many servers as necessary.

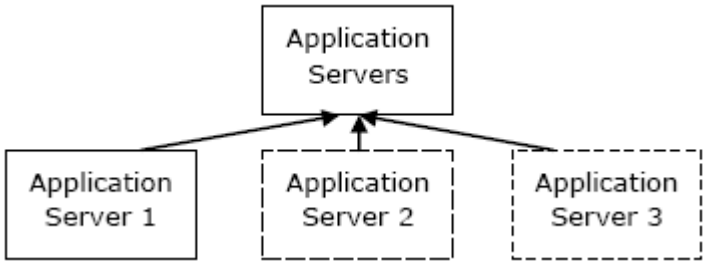


The Application Servers service could monitor the health of individual child services with a redundancy, percentage of low sensitivity policy.

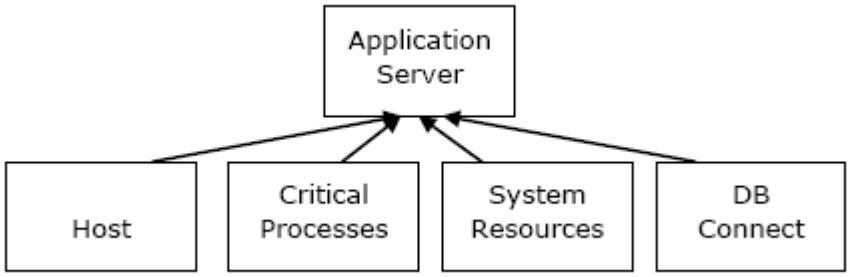
Each Application Server service; however, would monitor its resource monitors with a high sensitivity policy. The resource monitors focus on the host itself, critical processes, system resources, application connection, and application performance. You can notice that the Host resource monitor excludes CA eHealth notifications which are performance-based. The Application Performance resource monitor is only impacted by CA eHealth notifications. Both resource monitors would likely have the same host model as a resource, but are affected by different types of resource outages. This allows you to determine if the service outage is related to availability or performance.



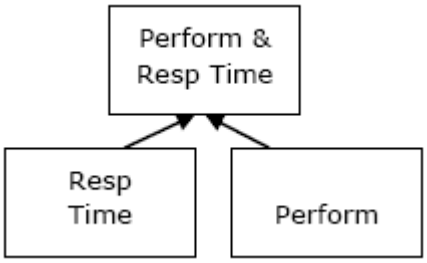
The Database Servers service has a similar structure to Application Servers. Multiple individual servers can be supported.



Each individual server supports four resource monitors. These detect faults on the host model, critical database processes, system resources, and database connections. Once again, you can see that the Host resource monitor excludes CA eHealth notifications.



Finally the service also includes a component for monitoring performance and response time. Application Response Time sub service would focus on monitoring SPM tests in DX NetOps Spectrum. You can use default response time policies, or perhaps develop a custom policy for average response time. The Performance sub service would detect resource faults that are based on CA eHealth performance notifications that are sent to DX NetOps Spectrum. The performance service would likely use a set of host models as its resources where CA eHealth notifications are mapped to resource alarms.



This example is not designed to fit a specific scenario, but provide you with an example of how modeling gateway can be used to model common patterns in your environment.

If you are able to identify all of the resources, they can be added to the service and resource monitor elements in the xml file. Even if you are not sure of all service resources, services and resource monitors can still be imported and empty of resources. The empty services and resource monitors appear with blue icons in DX NetOps Spectrum prompting you to the need to fill in the resources.

Example: Define a Service Maintenance Schedule

In addition to defining the structure of a service and its resources you can also specify a maintenance schedule for service. The following XML segment defines a maintenance schedule for a service named "ABC Service", using an existing schedule model.

```
<SM_Service
     containment_relation="MaintenenceScheduledBy"
     name="ABC Service">

          <Schedule name="Every day from 6 PM - 7 AM"
                    SCHED_Recurrence="2"
                    SCHED_Duration="46800"
                    SCHED_Start_Hour="18"
                    SCHED_Start_DoM="0"
                    SCHED_DayBitMask="0"
                    SCHED_Start_Day="0"
                    SCHED_Description=""
                    SCHED_Start_Year="0"
                    SCHED_Start_DoW="0"
                    SCHED_Start_MoY="0"
                    SCHED_Start_Minute="0"
                    SCHED_Start_Month="0"
                    SCHED_Daily_Repeat_Limit="2"
                    SCHED_Recurrence_Multiplier="1"/>

</SM_Service>
```

Example: Define an Alarm Exemption List for a Service or Resource Monitor

You can specify an alarm type exclusion list for a service using modeling gateway. This setting applies to the service model, and can be used in lieu of any setting that is made at the policy level. This xml configuration is equivalent to specifying the alarm type exemption in the Exemptions tab of the Service Editor. If the configuration you want to specify for this service is defined within a policy, specify the policy ID.

The following XML segment specifies the three alarms (0xabcd0001, 0xabcd0001, 0xabcd0002) that are the only alarm types to affect (Cause_List_Control="1") the service:

```
<SM_Service
          containment_relation="SlmMonitors"
          name="Access Routers"
          Criticality="30"
          AttrToWatch="Condition"
          Cause_List_Control="1"
          Special_Cause_List="0xabcd0001,0xabcd0001,0xabcd0002"
          MonitorPolicy_ID="2"
          Generate_Service_Alarms="true">

          <Device ip_dnsname="10.253.9.16" />
          <Device ip_dnsname="10.253.9.17" />
          <Device ip_dnsname="192.168.152.5" />
          <Device ip_dnsname="172.19.17.174" />
      </SM_Service>
```

The following XML document specifies a range of alarms (0xeeee0000-0xeeee002b) that can be excluded from impacting the health of the service (Cause_List_Control="2") the resource monitor:

```
      <SM_AttrMonitor
          containment_relation="SlmMonitors"
          name="Access Routers"
          Criticality="30"
          AttrToWatch="Condition"
          Cause_List_Control="2"
          Special_Cause_List="0xeeee0000-0xeeee002b"
          MonitorPolicy_ID="2">

          <Device ip_dnsname="10.253.9.16" />
          <Device ip_dnsname="10.253.9.17" />
          <Device ip_dnsname="192.168.152.5" />
          <Device ip_dnsname="172.19.17.174" />
      </SM_AttrMonitor>
```

Example: Associate an SLA to a Service

Service models can be associated to SLA models using modeling gateway. This association does not create any specific guarantee models, or associate the service to any existing guarantee model. Associations for guarantees must be done explicitly and can be covered later in this document.

The following sample XML shows how to associate an SLA to a service:

```
<SM_SLA
        containment_relation="SlmGuarantees"
        name="Acme Service Level Agreement">

        <SM_Service name="Acme"/>
</SM_SLA>
```

Example: Create a Guarantee for an SLA

Guarantee models are created within an SLA element, and can be associated to a service or resource monitor model. The following XML shows how to create a guarantee for an SLA call Acme Service Level Agreement. You can call the guarantee as Engineering Guarantee and record outage time when the Engineering service is Down.

```
<SM_SLA
     containment_relation="SlmHasGuarantee"
     name="Acme Service Level Agreement">

     <SM_Guarantee
        containment_relation="SlmIsMeasuredBy"
        name="Engineering Guarantee"
        GuaranteeControl="1"
        GuaranteeType="0"
        ServiceHealthType="1"
        WarningThresholdPercent="80.5"
        ViolationThresholdPercent="99.5"
        GuaranteeNotes="Tracks Down Time For Engineering Service"
        GuaranteeDescription="Availability Guarantee for Acme Engineering"
        MOT_Threshold="300"
        MTBF_Threshold="300"
        MTTR_Threshold="300">

        <SM_Service name="Engineering"/>

     </SM_Guarantee>
</SM_SLA>
```

Guarantee business hours can be specified using xml, by defining the schedule. The following example shows how a schedule named Business Hours can be associated to the Engineering Guarantee model.

```
        <SM_Guarantee
            containment_relation="SlmSchedulesGuarantee"
            name="Engineering Guarantee"
            GuaranteeType="0">

            <Schedule
                name="Business Hours"
                SCHED_Recurrence="2"
                SCHED_Daily_Repeat_Limit="0"
                SCHED_Duration="25200"
                SCHED_Recurrence_Multiplier="1"
                SCHED_Start_DoM="0"
                SCHED_Start_DoW="0"
                SCHED_Start_Hour="8"
                SCHED_Start_Minute="0"
                SCHED_Start_Month="0"
                SCHED_Start_Day="0"
                SCHED_DayBitMask="0"
                SCHED_Start_Year="0"
                SCHED_Start_MoY="0"
                SCHED_Description="Standard Business Hours 8AM Start"/>
        </SM_Guarantee>
```

Example: Define an SLA

The SLA Manager is the top-level model for all SLAs. Each SpectroSERVER has one SLA Manager model.

In addition to creating the SLA and its guarantees, the SLA period can be defined using modeling gateway. The following XML example shows how to define an SLA period, by specifying the period schedule.

```
<SM_SLA_Mgr
    name="SLAs"
    containment_relation="SlmContainsSLAs"
    >
<SM_SLA
        containment_relation="SlaPeriod"
        name="Acme Service Level Agreement"
        SLA_Control="1"
        SLA_ExpirationDate="1514696400"
        SLA_Notes="Manages SLA for Acme Services"
        SLA_Description="Acme Management Technologies Internal Service Level Agreement">

         <Schedule
            name="Daily SLA Schedule"
            SCHED_Recurrence="2"
            SCHED_Duration="0"
            SCHED_Start_Hour="0"
            SCHED_Start_DoM="0"
            SCHED_DayBitMask="0"
            SCHED_Start_Day="0"
            SCHED_Description=""
            SCHED_Start_Year="0"
            SCHED_Start_DoW="0"
            SCHED_Start_MoY="0"
            SCHED_Start_Minute="0"
            SCHED_Start_Month="0"
            SCHED_Daily_Repeat_Limit="0"
         SCHED_Recurrence_Mutiplier="1"/>

</SM_SLA>

</SM_SLA_Mgr>
```

Example: Define a Customer and a Customer Group

Customer models must be defined within the CustomerManager tag in the XML file. Each SpectroSERVER has one Customer Manager model, and it must be named as Customers.

The following example XML document defines a customer that is named Product Development within a customer group named XYZ Group:

```
<CustomerManager
        name="Customers"
        containment_relation="Groups_Customers">

        <SM_CustomerGroup
                name="XYZ Group"        <-- Specify Customer Group name here.
                containment_relation="Groups_Customers">

                <!-- This code defines a Customer and associates -->
                <!-- it with a Service. A Customer contains -->
                <!-- primary and secondary contact information, -->
                <!-- and a criticality that can effect the -->
                <!-- severity of a Service outage for Service -->
                <!-- models used by the Customer. -->

                <SM_Customer
                        containment_relation="SlmUses"
                        name="Product Development"      <-- Specify Customer name here.
                        CustomerField4="Pease International TradePort"
                        CustomerField5="123 Big Dr."
                        CustomerField6="Portsmouth, NH 03801"
                        CustomerField7="USA"
                        CustomerID="11DU-156"
                        Criticality="10"
                        Contact_Name="Fred Flintstone"
                        Contact_Title="Product Development Manager"
                        Email_Address="fred@proddev.com"
                        Phone_Number="123-456-7890"
                        Mobile_Phone_Number="123-456-1111"
                        Secondary_Contact_Name="Barney Rubble"
                        Secondary_Contact_Title="Product Development Manager"
                        Secondary_Phone_Number="123-456-9999"
                        Secondary_Mobile_Phone_Number="123-456-8888"
                        Secondary_Email_Address="barney@proddev.com">

                        <SM_Service name="Development"/>    <-- Specify service name here.

                </SM_Customer>


                <!-- This code associates a Customer with an SLA. -->
                <SM_Customer
                        containment_relation="SlmAgreesTo"
                        name="Product Development">

                        <SM_SLA name="XYZ Service Level Agreement"/>

                </SM_Customer>
```

Example: Import XML Input Files

You can use Modeling Gateway to import your Service Manager configuration files. Use the Modeling Gateway to test the XML document in Example: Services That Monitor Resources Directly.

Follow these steps:

1. Create a file containing the example XML document and save it to the <*$SPECROOT*>/SS-Tools directory with the file name, slm_test1.xml.
2. To run the modelinggateway tool for importing a single file, use the following syntax.

   Windows

   ```
   modelinggateway.bat -vnm vnm_name -i import_file [-o outputfile] [-debug debugfile]
   ```

   Linux

   ```
   modelinggateway -vnm vnm_name -i import_file [-o outputfile] [-debug debugfile]
   ```

   – **-vnm *vnm_name***
   Specifies the SpectroSERVER host name.

   – **- i *import_file***
   Specifies the XML file name which contains the necessary input information (that is compiled with .modelinggateway.dtd.)

   NOTE

   If you are importing from multiple files, specify the files names that are enclosed in comma-separated {} brackets. For example, refer to the syntax sample for importing multiple files.

   – **-o *outputfile***
   (Optional) Logs the error information to the file named in the *outputfile* parameter.

   NOTE

   If you don't specify the debug/output file names, the error information is logged to a file named *import_file*.log;where, *Import_file* is the name of the XML file.In case of multiple imports, the debug/output files are appended and you can see consolidated logs.

   – **-debug *debugfile***
   (Optional) Indicates that you would like to create a debugging output file during the import process. When using the -debug option, you can provide your own debug file name for output. If you do not supply a value for *debugfile*, the debug file name defaults to the *import_file* name suffixed with ".debug."

   NOTE
   The -debug option requires disk space on the machine where Modeling Gateway is run. For example, a large debugging output file can result when the number of models in the *import_file* is large or when the device models have large interface densities.

   ### Import Multiple Files

   The modelinggateway tool now supports import from multiple XML files. You can now specify any number of import files.

   To run the modelinggateway tool for importing multiple files, use the following syntax:

   ```
   modelinggateway -vnm vnm_name [-user SS user][-i{importfile1,importfile2...}]|[ -cmdb]-e exportfile][-o
    outputfile] [-debug debugfile]
   ```

3. If no errors are reported, device and container models are created. The output can be similar to the following output:

   ```
   Import session started Fri, December 29, 2006, at 02:53:32 EST
   Start parsing file slm_test1.xml
   ```

```
Start importing file slm_test1.xml
Container models created: 1
Identifying ports...
Import session finished Fri, December 29, 2006, at 02:53:38 EST
```

Service Attributes (SM_Service)

When creating service management models with Modeling Gateway, the XML code must provide values for the service attributes (sm_service) listed in the following table.

Attribute	Description	Possible Values
containment_relation	The set of supported relations from which associations can be created.	SlmMonitors SlmWatchesContainer MaintenenceScheduledBy
name	The model name of the service.	Text string, up to 256 characters
Criticality	The impact severity of the alarm relative to other current alarms. A higher Criticality value contributes to a higher impact severity value for alarms in OneClick.	10 = Low 15 = Medium Low 20 = Medium 25 = Medium High 30 = High
AttrToWatch	The attribute monitored on the service resource models. The AttrToWatch value should be consistent with the MonitorPolicy_ID. For example, if you specify AttrToWatch="Condition", you should specify a MonitorPolicy_ID for a Condition Policy (1-5).	Condition RM_Condition(service health) Contact_Status Port_Status LatestErrorStatus (Response Time)
MonitorPolicy_ID	The ID of a specific monitor policy as defined on the GlobalConfig model type. The ID should be consistent with the AttrToWatch value.	1 - 21 (DX NetOps Spectrum default) 1000 - *n* (User-defined)
Generate_Service_Alarms	Determines whether the SM_Service model generates alarms upon a change in service health.	True or False
Special_Cause_List	The alarm cause codes to include in an alarm type exemption list for a service.	Text string in the form of comma separated alarm causes or ranges separated with a hyphen (-)
Cause_List_Control	The integer that defines which alarm types affect or do not affect a service.	0=Unused (Ignore Cause) 1=Inclusive (Caused By) 2=Exclusive (Not Caused By)

Monitor Resource Monitor Attributes (SM_AttrMonitor)

When creating service management models with Modeling Gateway, the XML code must provide values for the monitor resource monitor attributes (SM_AttrMonitor) listed in the following table.

Attribute	Description	Possible Values
containment_relation	The set of supported relations from which associations can be created.	SlmMonitors SlmWatchesContainer
name	The model name of the resource monitor.	Text string, up to 256 characters
AttrToWatch	The attribute monitored by the resource monitor. The AttrToWatch value should be consistent with the MonitorPolicy_ID. For example, if you specify AttrToWatch="Condition", you should specify a MonitorPolicy_ID for a Condition Policy (1-5). **Note:** See Policy ID Mappings for more information.	Condition RM_Condition (service health) Contact_Status Port_Status LatestErrorStatus (Response Time)
MonitorPolicy_ID	The ID of a specific monitor policy as defined on the GlobalConfig model type. It should be consistent with AttrToWatch value. **Note:** See Policy ID Mappings for more information.	1 - 21 (DX NetOps Spectrum default) 1000 - *n* (User-defined)
Special_Cause_List	The alarm cause codes to include in an alarm type exemption list for a resource monitor.	Text string in the form of comma separated alarm causes or ranges separated with a hyphen (-)
Cause_List_Control	The integer that defines which alarm types affect or do not affect a resource monitor.	0=Unused (Ignore Cause) 1=Inclusive (Caused By) 2=Exclusive (Not Caused By)

Customer Group Attributes (SM_CustomerGroup)

When creating service management models with Modeling Gateway, the XML code must provide values for the customer group attributes (SM_CustomerGroup) listed in the following table.

Attribute	Description	Possible Values
containment_relation	The set of supported relations from which associations can be created.	Groups_Customers
name	The model name of the customer group.	Text string, up to 256 characters

Customer Attributes (SM_Customer)

When creating service management models with Modeling Gateway, the XML code must provide values for the customer attributes (SM_Customer) listed in the following table.

Attribute	Description	Possible Values
containment_relation	The set of supported relations from which associations can be created.	SlmUses
name	The model name of the customer.	Text string, up to 256 characters
CustomerField4	Address Line 1	Text string, up to 256 characters
CustomerField5	Address Line 2	Text string, up to 256 characters
CustomerField6	City, State, Postal Code	Text string, up to 256 characters
CustomerField7	Country	Text string, up to 256 characters
CustomerID	Any identification number.	Alpha-numeric string
Criticality	The impact severity of the alarm relative to other current alarms. A higher Criticality value contributes to a higher impact severity value for alarms in OneClick.	10 = Low 15 = Medium Low 20 = Medium 25 = Medium High 30 = High
Contact_Name	The person associated with the customer model.	Text string, up to 256 characters
Contact_Title	The contact person title.	Text string, up to 256 characters
Email_Address	The contact person email address.	Text string, up to 256 characters
Phone_Number	The contact person phone number.	Text string, up to 256 characters
Mobile_Phone_Number	The contact person mobile phone number.	Text string, up to 256 characters
Secondary_Contact_Name	The alternate contact person name.	Text string, up to 256 characters
Secondary_Contact_Title	The alternate contact person title.	Text string, up to 256 characters
Secondary_Phone_Number	The alternate contact person phone number.	Text string, up to 256 characters
Secondary_Mobile_Phone_Number	The alternate contact person mobile phone number.	Text string, up to 256 characters
Secondary_Email_Address	The alternate contact person email address.	Text string, up to 256 characters

SLA Attributes (SM_SLA)

When creating service management models with Modeling Gateway, the XML code must provide values for the SLA attributes (SM_SLA) listed in the following table.

Attribute	Description	Possible Values
containment_relation	The set of supported relations from which associations can be created.	SlaPeriod SlmHasGuarantee SlmGuarantees
name	The model name of the SLA.	Text string, up to 256 characters

SLA_Control	Specifies whether the SLA is active during the current SLA period or becomes active at the onset of the next period.	0 (inactive until next period) or 1 (active)
SLA_ExpirationDate	The UNIX timestamp, measured as the number of seconds from January 1, 1970.	For example, the value 1514696400 - Dec 31, 2017
SLA_Notes	Any text notes about the SLA.	Text string, up to 256 characters
SLA_Description	Any text description of the SLA.	Text string, up to 256 characters

Guarantee Attributes (SM_Guarantee)

When creating service management models with Modeling Gateway, the XML code must provide values for the guarantee attributes (SM_Guarantee) listed in the following table.

Attribute	Description	Possible Values
containment_relation	The set of supported relations from which associations can be created.	SlmIsMeasuredBy
name	The model name of the guarantee.	Text string, up to 256 characters
GuaranteeControl	Specifies whether the guarantee is active or inactive during the current period.	0 (inactive) or 1 (active)
GuaranteeType	Specifies whether the guarantee monitors service availability or performance (response time).	0 (availability) or 1 (performance)
ServiceHealthType	The type of service health time that is accumulated by the guarantee. An availability guarantee can accumulate both down and degraded time. A performance guarantee accumulates only degraded time.	1 (Down) or 2 (Degraded)
WarningThreshold	The number of seconds of outage time allowed per period before a warning alarm is issued.	0 - *n*
WarningThresholdPercent	The percentage of outage time allowed before a warning alarm is issued.	0 - 100%
ViolationThreshold	The number of seconds of outage time allowed per period before a violation occurs.	0 - *n*
ViolationThresholdPercent	The percentage of uptime per period below which a violation occurs.	0 - 100%
GuaranteeNotes	Any text notes about the guarantee.	Text string, up to 256 characters
GuaranteeDescription	A text description of the guarantee.	Text string, up to 256 characters
MOT_Threshold	Maximum outage time in seconds	0 - *n*
MTBF_Threshold	Mean time between faults in seconds	0 - *n*
MTTR_Threshold	Mean time to repair in seconds	0 - *n*

Schedule Attributes (Schedule)

When creating service management models with Modeling Gateway, the XML code must provide values for the schedule attributes (Schedule) listed in the following table.

Attribute	Description	Possible Values
name	The model name of the schedule. **Note:** DX NetOps Spectrum renames the schedule name that you provide.	Text string, up to 256 characters
SCHED_Recurrence	Specifies when the schedule is implemented.	1 = Always (24 x 7) 2 = Daily 3 = Weekly 4 = Monthly 5 = Yearly
SCHED_Start_Hour	The hour the schedule starts.	0 - 23
SCHED_Start_Minute	The minute the schedule starts.	0 - 59
SCHED_Start_DoW	The day of the week the schedule starts	0 - 6
SCHED_Start_DoM	The day of the month the schedule starts.	1 - 31
SCHED_Start_Month	The month of the year the schedule starts.	0 (Jan.) - 11 (Dec.)
SCHED_Start_Year	The year the schedule starts. Entering 0 starts the schedule in the current year.	0
SCHED_Start_MoY	The month of the year the schedule starts. Entering 0 starts the schedule in the current month.	0
SCHED_Description	The description of the schedule.	Text string, up to 256 characters
SCHED_Duration	The duration the schedule is in effect in seconds.	0 - *n*
SCHED_Recurrence_Multiplier	The number of times the schedule is implemented.	1 - *n*
SCHED_Daily_Repeat_Limit	The number of consecutive days to repeat a daily schedule at the start of each recurrence period. Applicable to Weekly, Monthly, or Yearly recurrence only.	0 - *n*

Monitoring Service Management Components with the Service Dashboard

The Service Dashboard is Service Manager dedicated operational and administrative console. The Service Dashboard includes many of the same operational features as the OneClick Console, but focuses purely on service management components. The service dashboard offers an at-a-glance view to the real-time status of services, SLAs, and customers. Using Service Dashboard, you can also view history outage information, outage trends and summarized service availability.

The Service Dashboard

The Service Dashboard provides a service-centric view of your service management environment. Unlike the OneClick Console, which enforces user security at the data level, the Service Dashboard shows only the service, SLA, and customer models to which you have security access. If you do not have access to a specific model, you can still see explorer and topology icons in the OneClick Console. Within the service dashboard, if you do not have access to a model, that model is absent from all dashboard views. In some circumstances, you can allow the DX NetOps Spectrum Administrator to provide the service dashboard to specific DX NetOps Spectrum users to view only the appropriate service and SLA components.

Using the Service Dashboard, you can navigate to your service management environment through the dashboard explorer and topology panels. Component detail information for services, SLA, and customer models is equivalent to that available in OneClick with the addition of some new panels which display outage history and SLA/Guarantee trend information.

Unique to the Service Dashboard is the service topology view. The semi-customizable topology icons indicate service health. The topology icons are expandable and provide navigation to lower layer services. You can edit and annotate the topology view for those service hierarchies you create.

Open the Service Dashboard

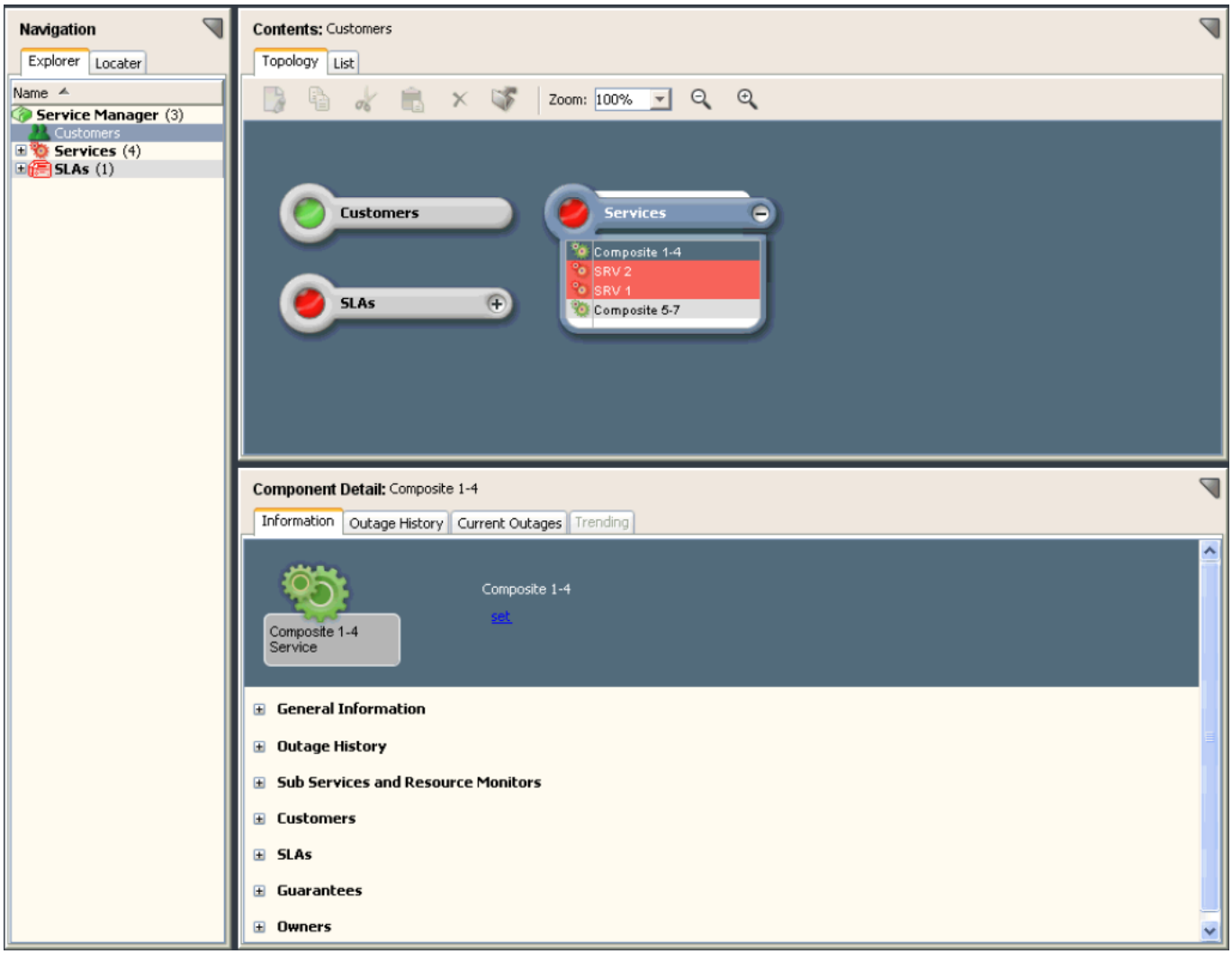
You can use various methods to open the Service Dashboard.

NOTE
You must have Service Manager license privileges to access the Service Dashboard.

Follow these steps:

1. Take *one* of the following steps:
   - Click the Service Dashboard link on the OneClick home page.
   - Click Tools, Utilities, Service Dashboard, from the main menu.
   - Right-click Service Manager in the Navigation panel and select Utilities, Service Dashboard in the OneClick Console.

   The following image shows the areas of the Service Dashboard you work with to monitor service components.



The Service Dashboard interface includes three main information panels that you work with to monitor your organizations services management components:

- **Navigation Panel**
  Displays components in a hierarchical folder structure. It provides the following options:
  - **Explorer tab**
    Lets you select the component that you want to view in the Contents panel and the Component Details panel. It groups components by services, SLAs, and customers. The explorer tab in the service dashboard condenses all landscapes into a single tree. You can notice that services which reside on different landscapes are all organized into the same services folder. This tab allows you to view your service management implementation without consideration of how many SpectroSERVER are deployed.
  - **Locater tab**
    Lets you search for the services, SLAs, or customers you want to view in the Contents panel. For example, can specify a particular component name or you can specify all components from a service management component category.
- **Contents Panel or Topology Panel**
  Displays summary information about the status of components you specify in the Navigation panel. You can select Topology and List views of service management components that are selected in the Navigation panel. The panel

topology view provides basic editing tools that you can use to arrange component icons, create basic shapes in the view, and annotate the view.
Note: Only the topology of user created hierarchies can be edited. Specifically the topology of the top tier folder that is named Services cannot be edited. This folder is comprised of multiple landscapes, and is shared among users. Given the variability in user access it would not be practical to allow topology editing to the top tier services topology.

- **Component Detail Panel**
  Displays detailed information about components that are selected in the Navigation or Contents panels. Tables and sub-view available in the Component Detail panel contains service configuration information and real-time and historical outage information.

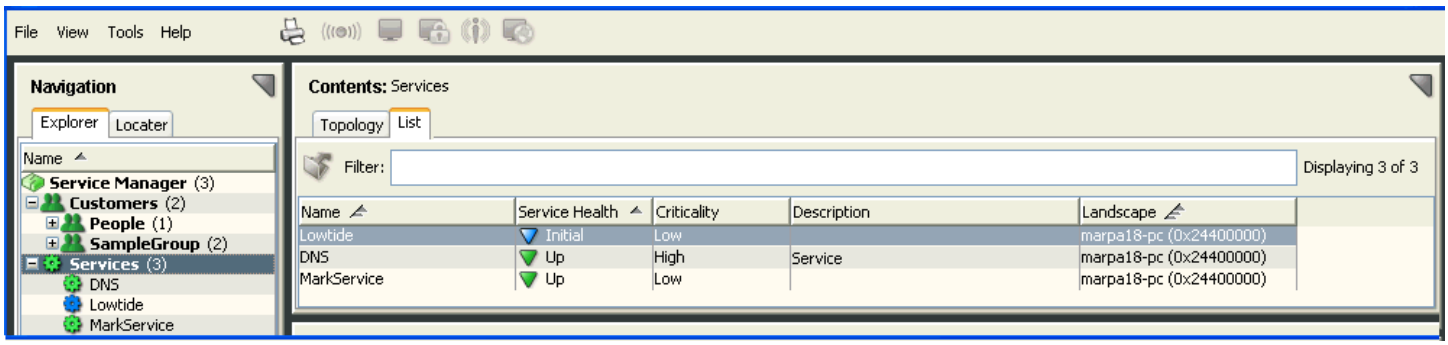
Topology and List Views in the Contents Panel

The Contents Panel contains a Topology and List view. The topology view provides expandable icons which indicate the real-time status of each service management component. The List view provides a table of service component models and specific attribute data. When a model is selected in the navigation panel, both the List and Topology views show the selected models children. If the selected model has no logical children, the List and Topology views show the selected model as contained by its parent.

NOTE
The List view is the default view for component results that are found through the Locater tab. You can specify a List view for components that are selected from the Explorer tab.

The Component Details panel displays comprehensive information about the component that is selected in the Contents panel.

To display a List view of a service management component group that is selected in the Explorer tab, click the List tab in the Contents panel, as shown in the following image:



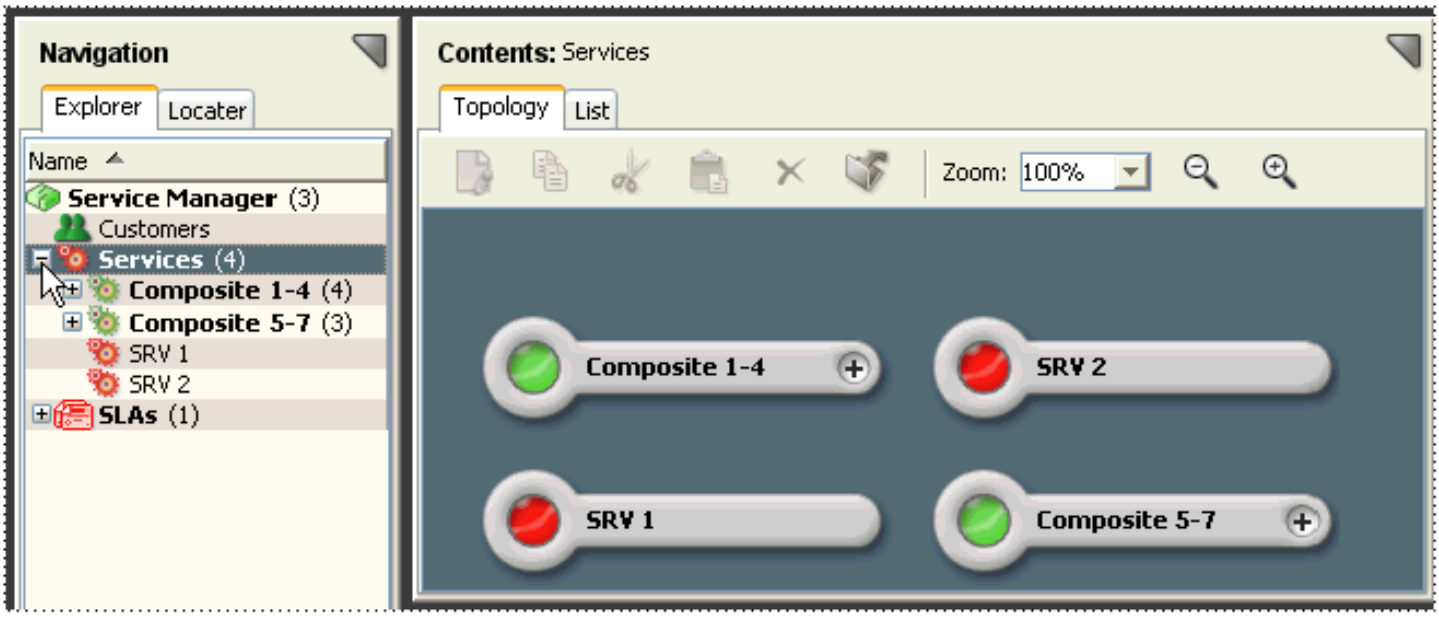
Explorer Folders and Topology Icons

The service dashboard does not display the service hierarchy about the landscapes where the models reside. Within the service hierarchy, you can see only the parent/child relationships of their service models. Within the OneClick Console, a service can appear as a top tier service directly under the service manager model on its landscape. If the service is a child of a service on another landscape, it can be seen in the dashboard only as a child, and not as both a child and top tier service. This behavior has changed from earlier releases of Service Manager in which the service would have appeared both directly under the Services folder and a child of the other service.

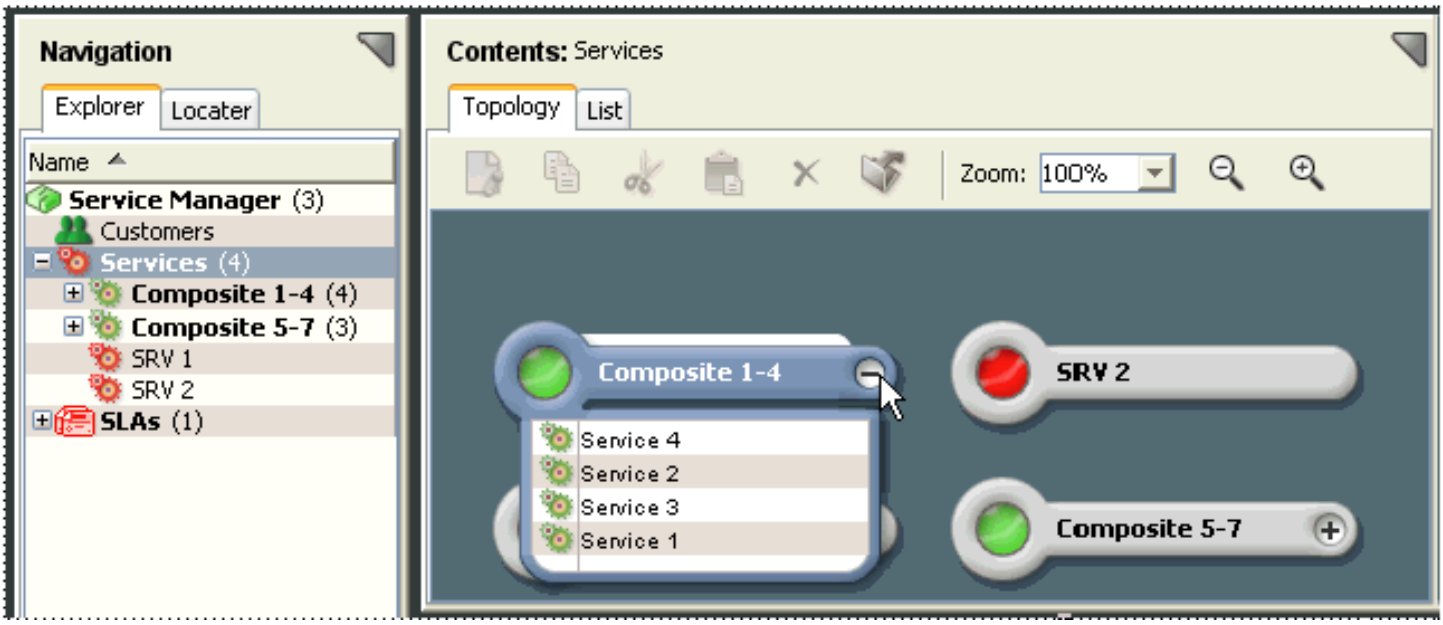
If a service management component displayed in the Explorer tab of the Navigation panel has a (+) next to it, this behavior indicates that the model or folder has one of more children. If you select a model, the List or Topology view displays the child models.

Icons within the topology view can also have (+) displayed in the icon itself. This behavior indicates that the model has children, click the plus sign (+) to expand and show a table of the associated child models.

For example, click the plus sign (+) next to a folder in the Navigation panel or on an icon in the Contents panel, or double-click a folder or icon:



For example, click the plus sign (+) on an icon in the Contents panel and then click a component from the drop-down list:



Status Indicators

The top-level service management component icons -- Services, SLAs, and Customers -- indicate the most severely affected corresponding top-level category for each landscape. If the Services icon is red, it indicates that the service manager model (Services) in at least one landscape has a Condition of critical.

The following table describes service component icon colors, and the associated state attribute values:

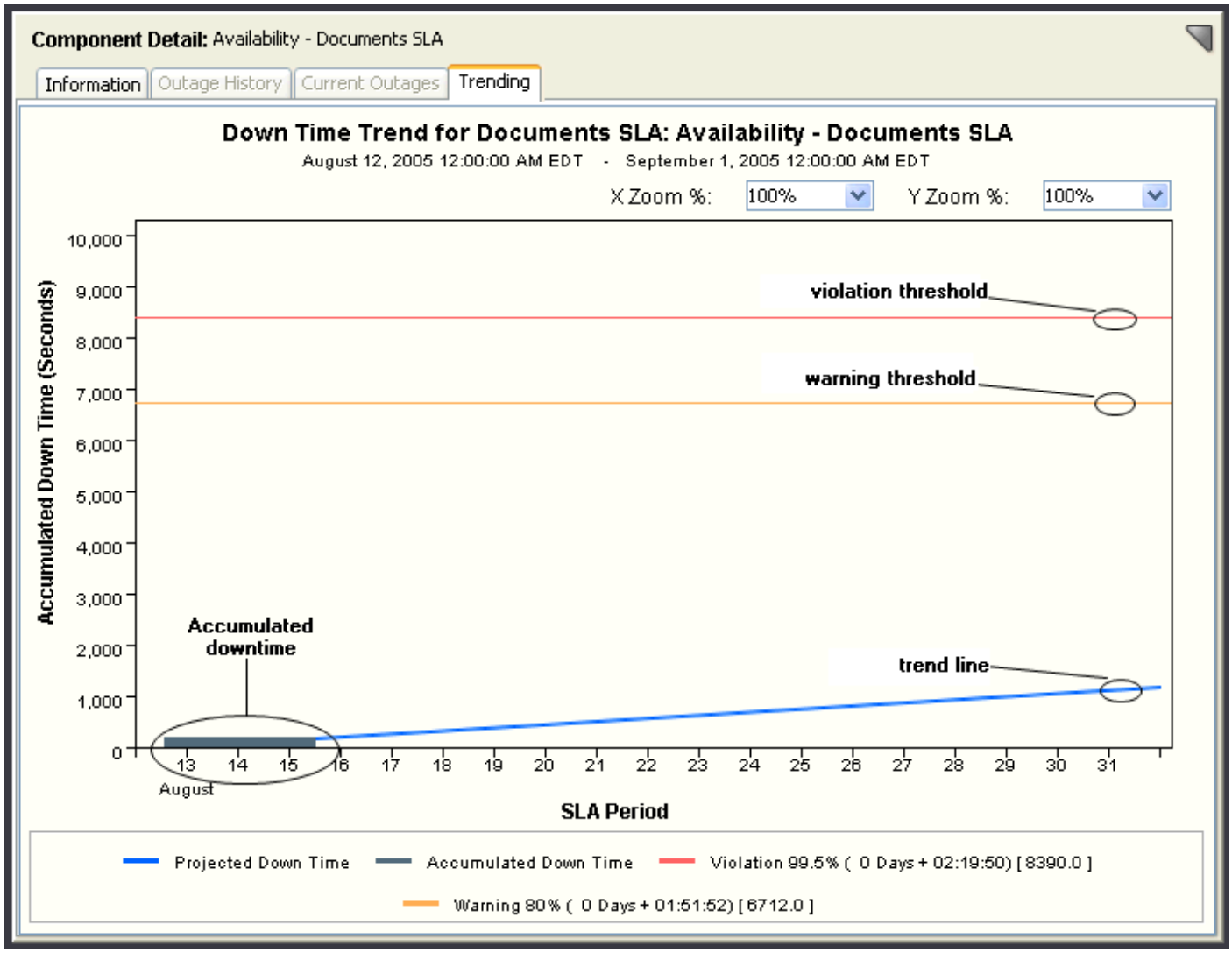
Color	Service	SLA	Customer
Green	Normal	Unaffected	Not Impacted
Yellow	Slightly Degraded	Compliant	Slightly Impacted
Orange	Degraded	Warned (at risk)	Significantly Impacted
Red	Down	Violated	Severely Impacted

Access Information about a Service Management Component

When you select a service management component from the Navigation panel or the Contents panel, the Service Dashboard Component Detail panel displays detailed information about it.

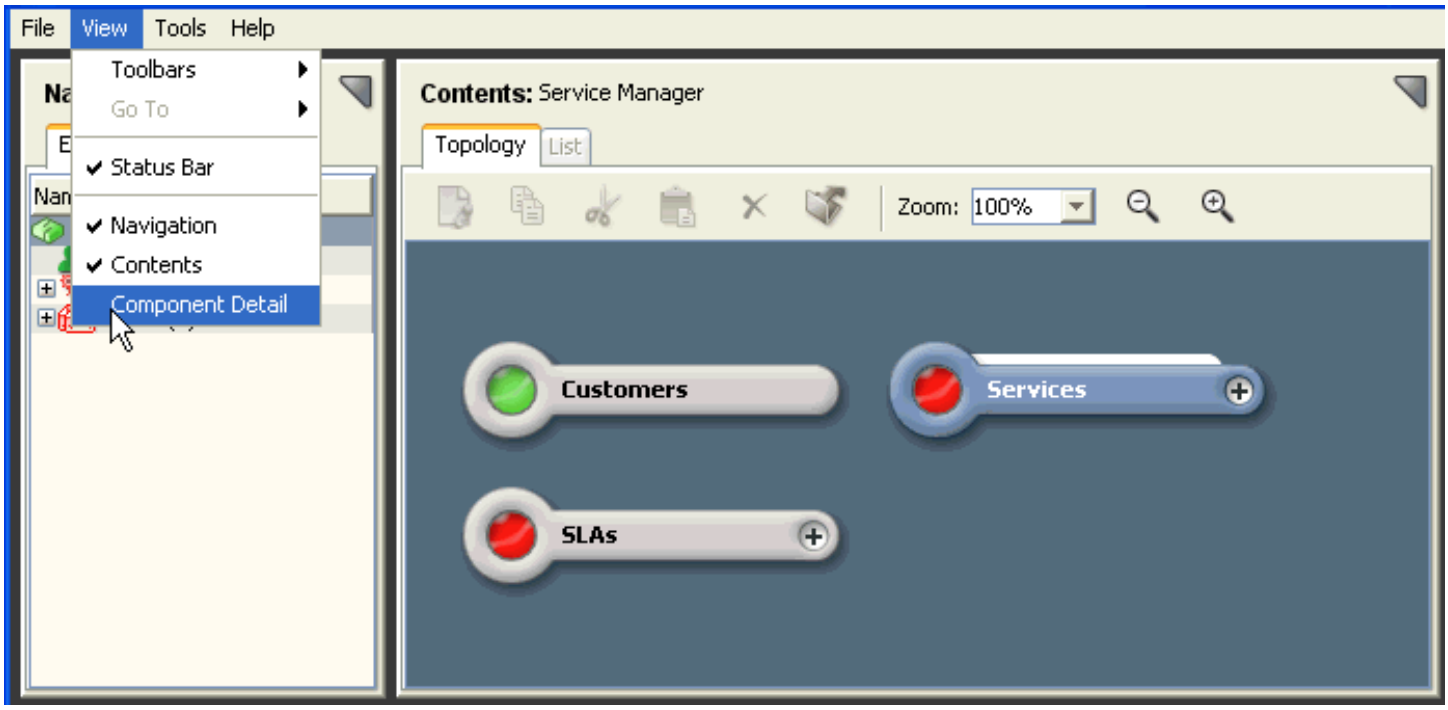
Follow these steps:

1. Open the Service Dashboard.
2. To access information in the Component Detail Panel, navigate to a model from either the Navigation or Contents views.
3. When the model is selected the Component Detail panel contains the following tabs:
   - **Information tab**
     Displays detailed information about any service management component (customers, services, SLAs, guarantees). Contains a number of sub-view each with detailed configuration information or historical data.
   - **Outage History tab**
     Displays the outage history for the last 31 days for the selected service.
   - **Current Outages tab (service and customer models only)**
     Displays information for any current service outage, including the cause of the current outage, assignment, trouble ticket ID, and status note of root cause alarm.
   - **Trending tab (SLAs and guarantee models only)**
     Displays accumulated outage time for a guarantee and a trend line to indicate the probability for the guarantee to become warned or violated within the current SLA period.
     The following shows an example of a trend chart:



Service Dashboard Interface Management

To show and hide Service Dashboard panels and the Status bar, select the interface component to display from the View menu, or clear the interface component to hide. The following image shows that the Component Detail panel has been deselected and is not included in the Dashboard interface:



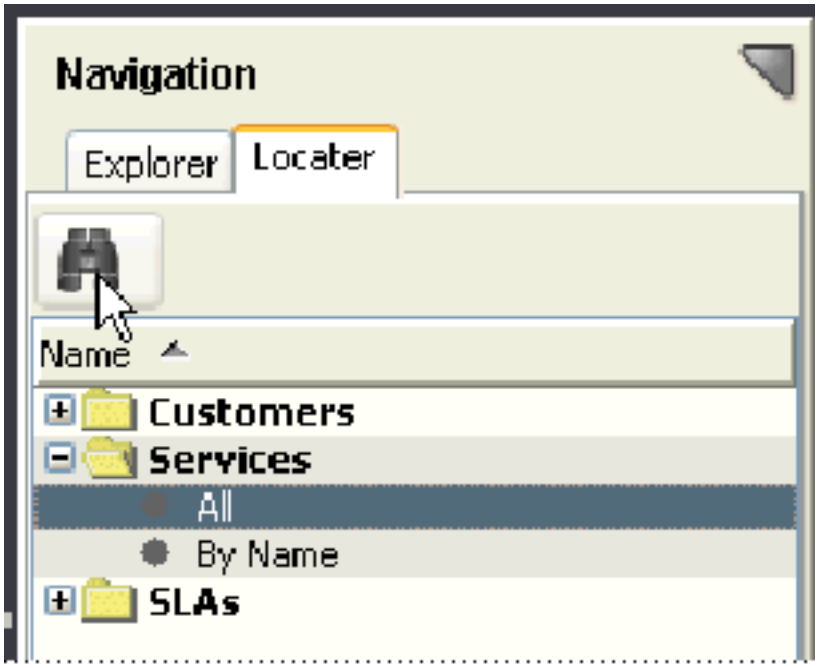
To dock and undock Dashboard panels, which include the Navigation, Content, and Component Detail panels and Component Detail information panels, click the Dock/Undock icon. The following image shows and example of Docking/ Undocking icon locations (circled). An undocked panel includes many of the same interface controls and options available from the main Service Dashboard.

Locate Service Management Components

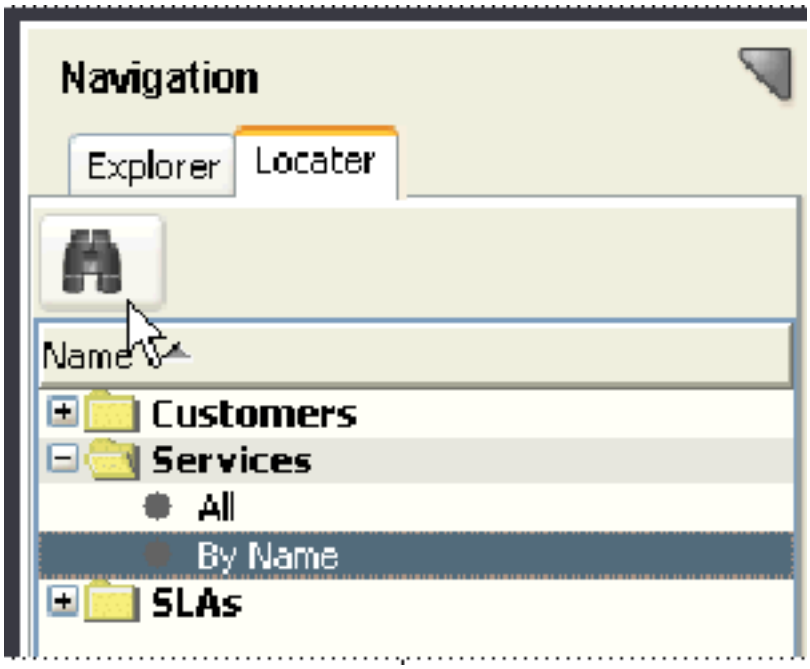
You can use the Dashboard Locater tab to display those service management components you specify, in the Contents panel and Component Detail panel.

Follow these steps:

1. Open the Service Dashboard.
2. Click the Locater tab in the Navigation panel.
3. Expand the customers, services, or SLAs folder from which you want to locate a component.
4. Specify whether to locate all components or a particular component from the selected component category:
   - Select All and to locate all components, click 

     The Select landscape to search dialog appears.
     Specify the landscape that you want to search and click OK.
   - Select By Name and to locate a specific component, click  .



The Search dialog appears.
Enter the component name in the Model Name field and click OK.

NOTE
To specify which landscapes to search, click the Landscapes button to open the Select Landscapes to Search dialog.

Results are displayed in the Contents panel.

Print Dashboard Views

Service Dashboard lets you print content from the Explorer and Locater tabs. You can also print the content from the Topology and List tabs. The Results view in the Contents panel and the Information view of the Component Details panel can be printed.

Follow these steps:

1. Click File, Print.
   The Print dialog appears.
2. Select the content that you want to print, and click OK.

Export Dashboard Views

Service Dashboard lets you save content from the Contents panel in multiple formats, using the export function.

You can export to a PNG (portable network graphics) format from the Explorer tab Topology view.

You can export to CSV (comma-separated values - spreadsheet compatible), text, and HTML formats from the Explorer tab List view and the Locater tab Results view.

Follow these steps:

1. Click 
   The Save As dialog appears.
2. Select an available format and click Save.

The dashboard view is exported.

Use the Service Dashboard Editing Tools

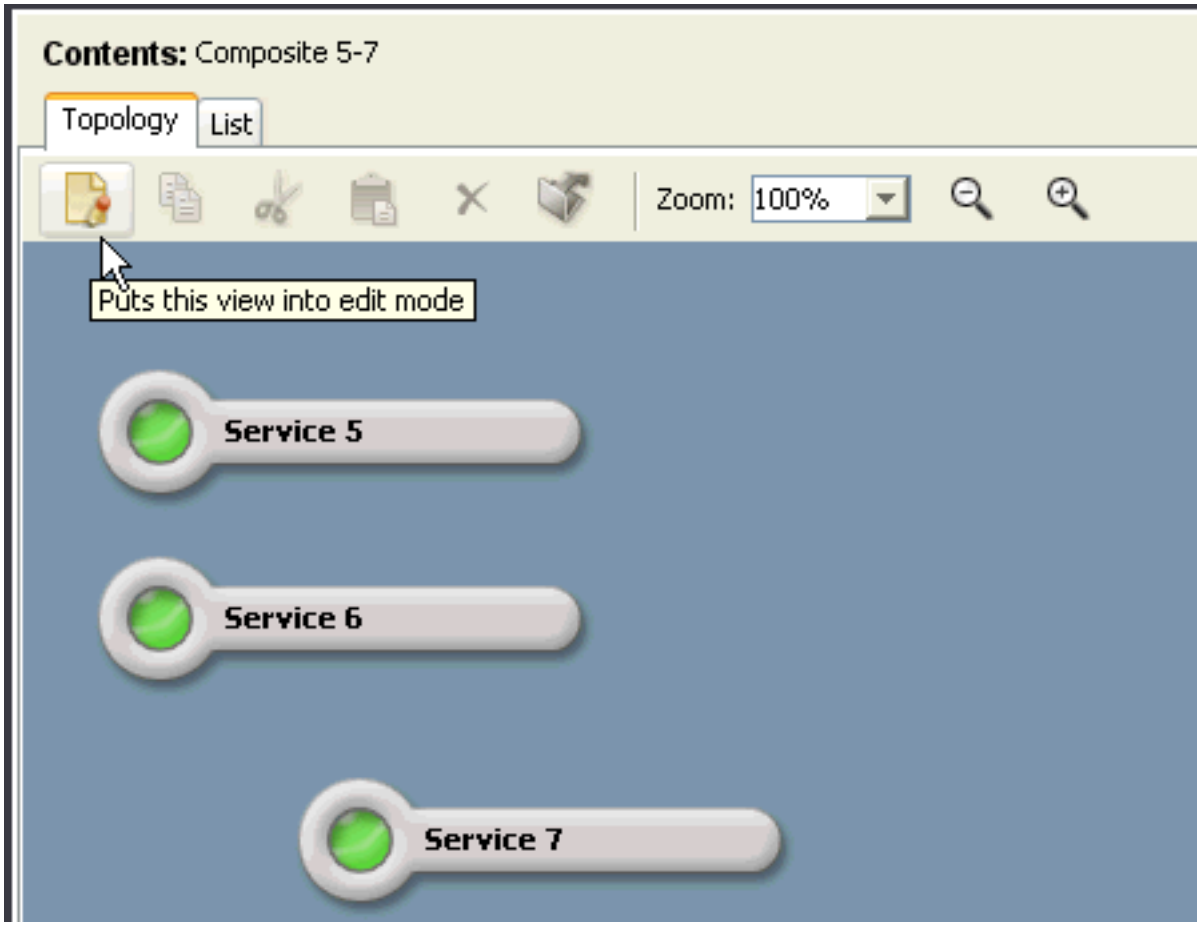
The Service Dashboard editing tools let you customize the topology views for Service, SLAs, and Customer Groups. Only the topology views of user created models can be edited. You cannot edit the topology views of the top tier service management folders.

You can perform the following tasks:

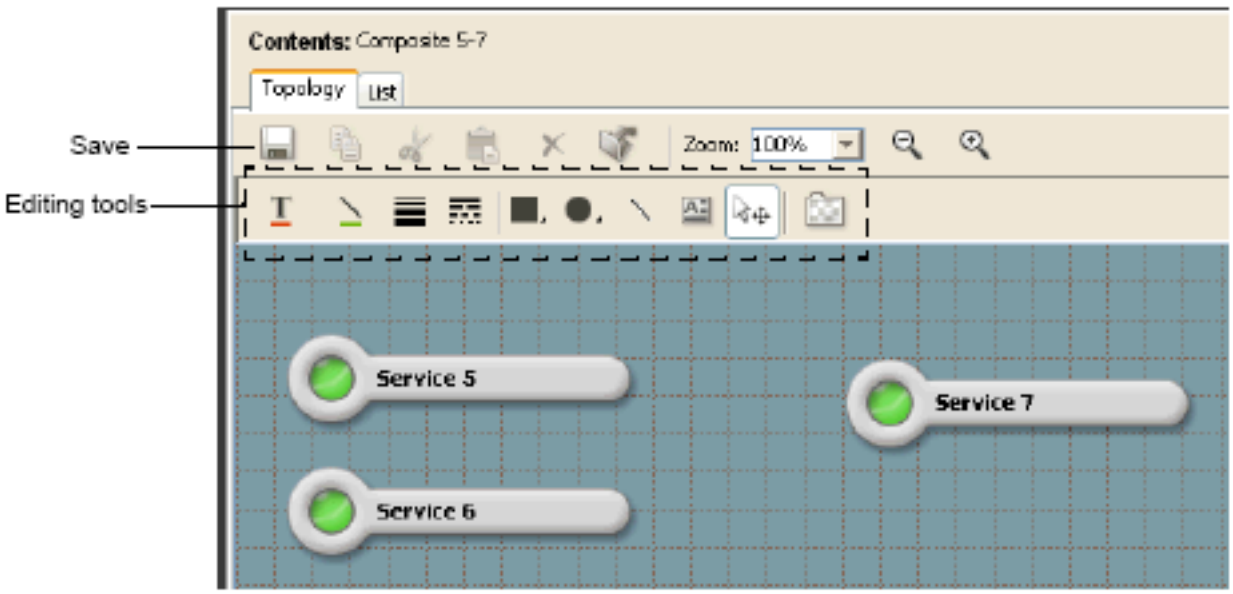
- Arrange component icons.
- Create basic shapes (rectangles, ovals) and lines.
- Enter annotations.
- Modify annotation text color, font, style, and size.
- Modify the topology view background color, grid dimensions (which you can use to align and resize shapes and lines), and size.

Follow these steps:

1. Click the Edit Mode icon in the Content panel Topology view, as shown in the following image:



The Topology view changes to editing mode and the editing toolbar appears. You can invoke a description of an editing tool by holding the cursor over the icon for the tool. The editing tools are a subset of tools available for editing the OneClick topology, and function in the same mapped.
Icon placement can be done either in tiled format or in custom format by selecting and dragging icons.



2. Edit the view as required and click Save.

Service Outage Management

Service dashboard provides various options for viewing current or historical service outages. From the current outage tab you can the view the following outage details:

- The health or severity of the outage
- The start time of the outage
- The resource or resources that are contributing the outage
- Any trouble shooter assignment or notation added to the outage

You can view and edit service outages from the Outage History tab panel or the Outage History sub-view of the Information tab panel.

The following information is available from the outage history table:

- Start time, end time, and duration of all outages.

  NOTE
  With Service Manager r9.2, the outage limitation of 31 days has been removed. You can now specify the date range of all outages.
- The resource or resources that caused each outage
- The name of the person that is assigned to troubleshoot the current service outage
- The status of an outage (unplanned, planned, or exempt)

The Outage History tab panel also offers a pie chart depicting service health or historical availability for the past 31 days.

The service outage editor is available from either the Outage History tab panel of the outage history sub-view. The outage editor allows user to add outage note or change the status of an outage by marking it as exempt or planned.

View Current Outages

Service Dashboard lets you view information about a service that is in an outage state.

Follow these steps:

1. Open the Service Dashboard dialog.
2. Select the service with the outage that you want to review.
3. Click the Current Outage tab in the Component Detail panel.
   The Component Detail panel displays information about the current outage, including the resource or resources that are contributing to the outage.

View Outage History

Service Dashboard lets you view information about all past and current outages for a service over the last 31 days.

Follow these steps:

1. Open the Service Dashboard.
2. Select a service that you want to review.
3. Click the Outage History tab in the Component Detail panel.
   The Component Detail panel displays information about the service outage history, including a chart with a summary of outage information and an itemized list of recent outages.
   The panel also displays Outage Details pertaining to the resources which contributed to the outage.

Service Outages

A service outage can have one of three potential status values: Unplanned, Planned, or Exempt. Usually the status of a service outage influences only how the historical record for that outage can be interpreted.

Most service outages have the status of Unplanned unless they are edited by the user. Unplanned outage time counts against the historical availability of a service model. It can be recorded if an SLA has associated one of more guarantees to the service.

You can use the service outage editor to change the status of an Unplanned outage to either Planned or Exempt. A status of planned indicates that the outage was expected to happen, and can be considered a maintenance outage. A status of Exempt indicates that the outage was not expected, but the cause or nature of the outage is such that the outage time should not be counted against the historical availability of the service. Outages which have a status of planned or exempt does not contribute time to SLAs.

Editing an outage does not change the real-time status of a service. If a service is Down and the user marks the outage as exempt the service remains down, as that is the true health of the service. Even though the real-time health of the service is unchanged and if a guarantee records time for the outage, you can remove the time from the guarantee that can change the status of an SLA.

Edit a Past or Ongoing Outage Status

You can edit the status of an ongoing or historical outage.

Consider the following information before editing a past or ongoing outage status:

- If an ongoing outage is edited, the outage status persists during the outage regardless of changes to the resources that contribute to the outage. Therefore, if the outage is marked as exempt, it remains exempt even if the resources contributing to the outage change. The current outage status can be seen in the Information tab for any service model.
- Outage time that is exempted or marked as planned is subtracted from the outage time tallied by a guarantee. This means that the status of an SLA that includes the guarantee changes accordingly. For example, an SLA with a status

of violated resulting from a service outage that exceeded an availability guarantee threshold is restored to compliant when the outage is exempted.

Follow these steps:

1. View the Outage History tab or open the Outage History sub-view of the Information tab.
2. Expand the Recent Outages subview in the Component Detail panel and select the outage that you want to edit from the Recent Outages list.

   NOTE
   Outages with health values of Down, Degraded or Slightly Degraded can be edited. You can also notice periods of Maintenance time, Initial time, and Loss of Management time in the outage table.
3. Click Outage Editor.
   The Edit Service Outage dialog appears.
4. Select an outage type for the service outage from the Set Outage Type of the drop-down list. The following options are available:
   - **Unplanned**
     Outages count against service availability.
   - **Planned**
     Outages do not count against service availability.
   - **Exempt**
     Outages are the unplanned outages that you have determined and must not count against service availability.
5. (Optional) Enter comments in the Notes/Reason field for the outage you changed. You can also enter comments for any outage regardless of whether you changed its status.
6. Click Save.
   The login name of the user who edited the outage is recorded and is displayed in the outage history table and in reports.
   The outage status is edited.

   NOTE
   You can also edit outages with the Affected Services Editor in Report Manager. For more information, see the Install Report Manager section.

Monitoring Service Management Components with Unicenter Management Portal

This section describes the procedure to Monitor Service Management Components with Unicenter Management Portal.

About the Service Level Manager Portlet

The Service Level Manager portlet provides summary status information about services, SLAs, and customers to Unicenter users with security access to Service Manager models. The portlet can be incorporated into the Unicenter Management Portal (UMP).

The portlet displays information about services, SLAs, and customers in a customizable tabular format. The portlet also provides context-sensitive links to the OneClick Console and Service Dashboard where you can view more detailed information about the services, SLAs, and customers.

Publish the Service Level Manager Portlet in UMP

Before viewing and accessing resources that are provided by the Service Level Manager portlet, first publish the portlet in UMP. For more information, see *UMP Documentation*.

Follow these steps:

1. Click the Knowledge tab and select Publish File.
2. Enter the URL for the Service Level Manager portlet in the Content box.

For example: http://abcde-sun/spectrum/slm

3. Enter a title for the Service Level Manager portlet and click OK.
   The Library tab displays the portlet title.
4. Click the portlet title to open the portlet.
   You are prompted to log in with a DX NetOps Spectrum user name and password. After you log in, the portlet appears.
5. Click the Work Place tab and select the workplace where you want to add the portlet.
6. Select Add Content, select the Service Level Manager portlet name, add it to the desired column, and click OK.
   The portlet is published in UMP.

View Service Information

You can use the Service Level Manager portlet to view service information.

Follow these steps:

1. Click the Services tab.
   The table displays status information about service models, as shown in the following image:

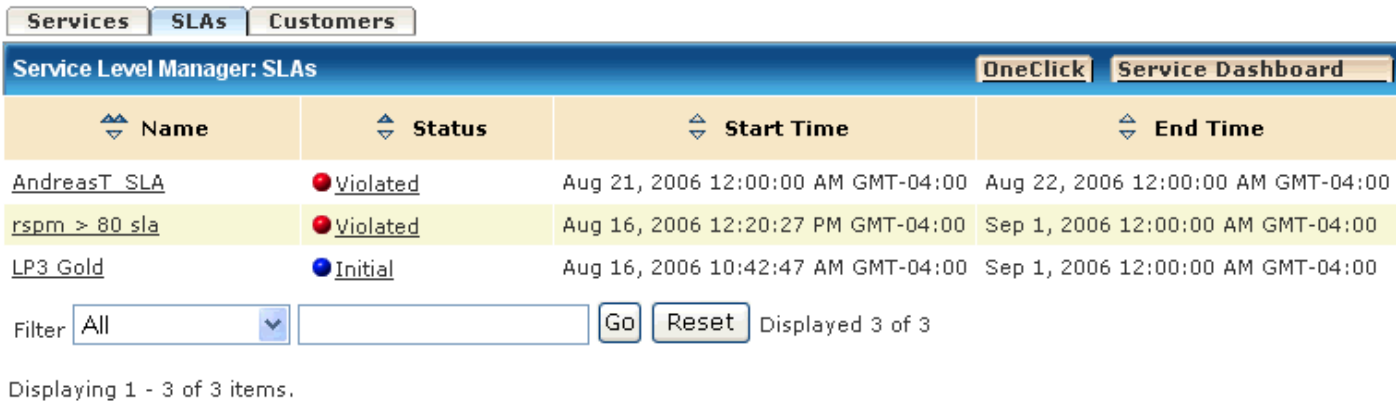
2. To view context-sensitive views of a service in OneClick and Service Dashboard, perform one of the following actions:
   - Click a service Name to view a detailed information view of the service in OneClick Console.
   - Click the Service Health indicator for a service to view information about the service in Service Dashboard.

View SLA Information

You can use the Service Level Manager portlet to view SLA information.

Follow these steps:

1. Click the SLAs tab.
   The table displays status information about SLA models, as shown in the following image:



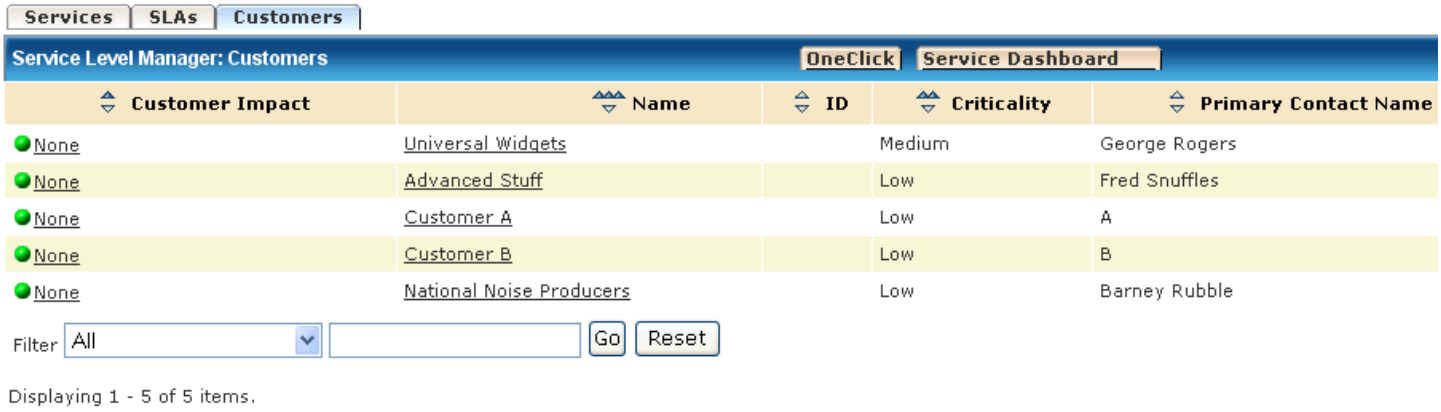
2. To view context-sensitive views of an SLA in OneClick Console and Service Dashboard, perform one of the following actions:
   – Click an SLA Name to view a detailed information view of the SLA in OneClick Console.
   – Click the Status indicator for an SLA to view information about the SLA in Service Dashboard.

View Customer Information

You can use the Service Level Manager portlet to view customer information.

Follow these steps:

1. Click the Customers tab.
   The table displays status information about customer models, as shown in the following image:



1. To view context-sensitive views of a customer in OneClick and Service Dashboard, perform one of the following actions:
   – Click a customer Name to view a detailed information view of the customer in OneClick Console.
   – Click the Customer Impact indicator for a customer SLA to view information about the customer in Service Dashboard.

Open the OneClick Console and the Service Dashboard

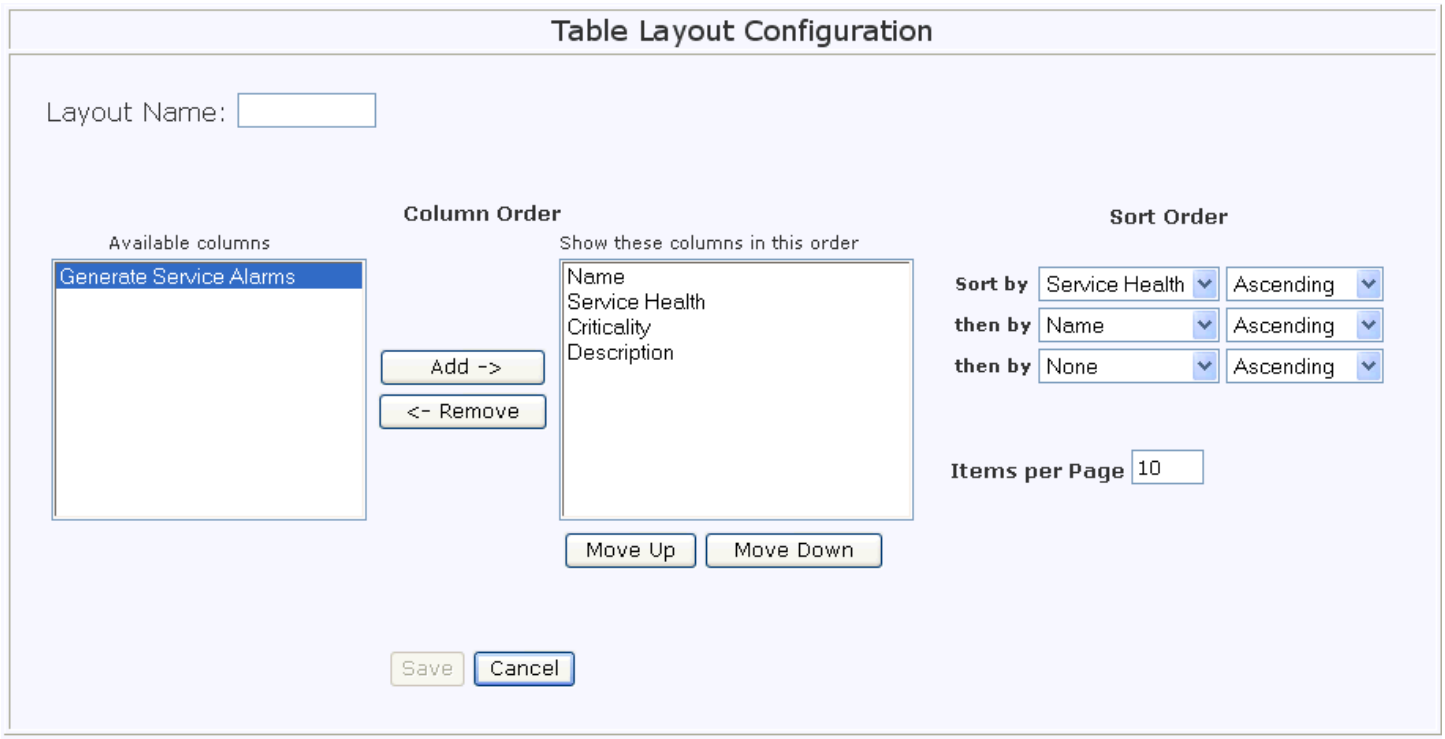
You can open the OneClick Console and the Service Dashboard from any component view in the Service Level Manager portlet. Perform one of the following actions:

- Click the OneClick button.
- Click the Service Dashboard button.

Apply and Manage Layouts

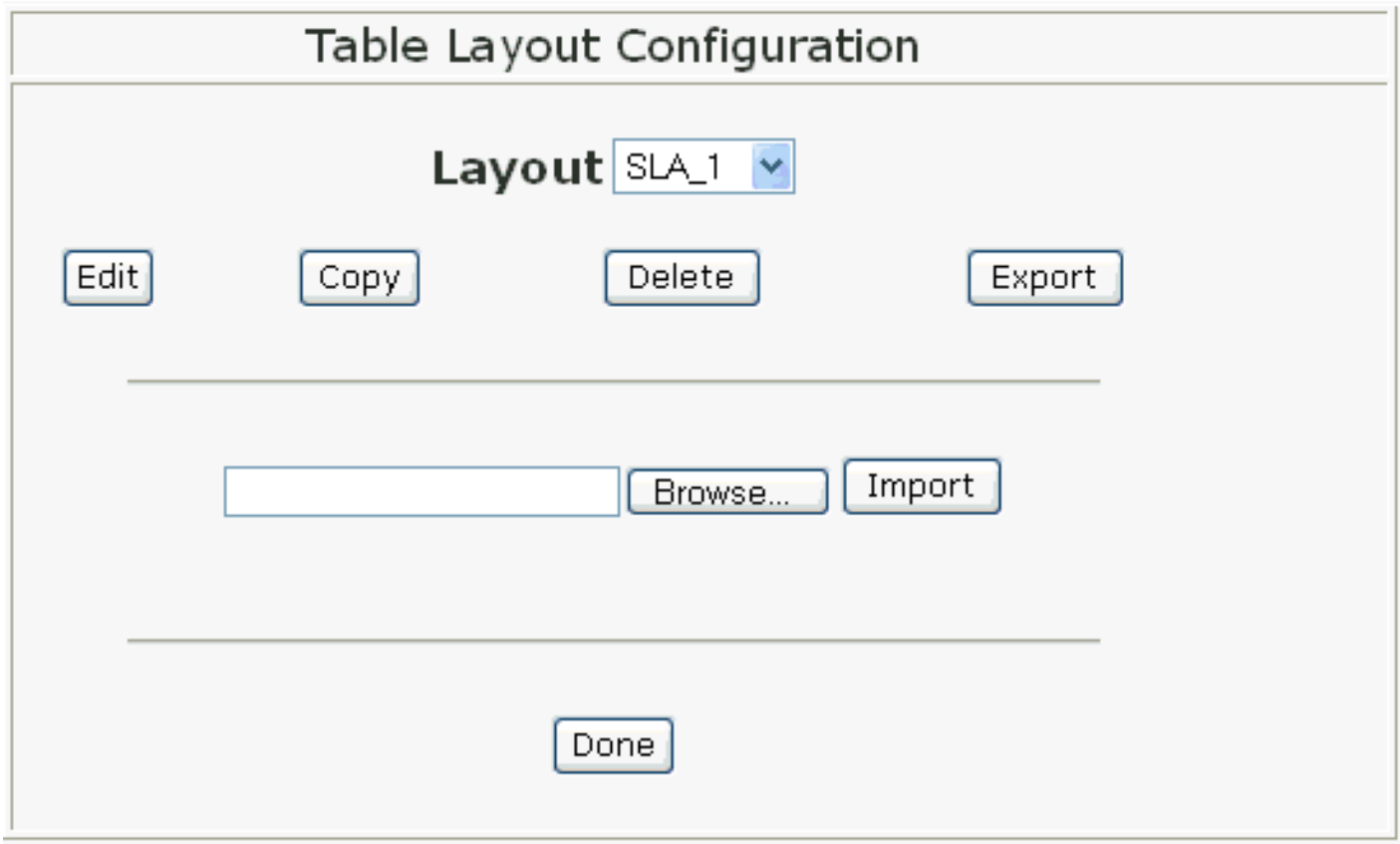
The Service Level Manager portlet lets you customize the type of information to include in the services, SLAs, and customers views. It lets you customize how you want to organize the information. You can also save multiple customized layouts for each view, edit all user-created layouts, export and import layout files, and remove layouts from the Service Level Management portlet.

The following image shows an example default layout for a service:



Perform the following actions to apply and manage layouts:

- To apply a layout to a component view, select it from the Configuration drop-down list. The table displays the columns and sort order that is specified for the layout.
- To create a layout for a component view, apply the default layout to the view, click Configure to open the Table Layout Configuration window, specify settings, and click Save.
- To manage a layout, apply the layout that you want to edit, copy, or delete to the current component view and click Configure.
  The Table Layout Configuration windows appears with available layout management options, as shown in the following image:



Select the following actions that you want to perform:

- Click Edit if you want to modify the layout.
- Click Copy if you want to create another version of the layout.
- Click Delete if you want to remove the layout from Service Level Management. You can export a layout to your hard drive before you delete it in case you want to re-use the layout in the future by importing it back into Service Level Management.
- Click Export if you want to save a copy of the layout to your hard drive (as a .prx file).
- Click Browse to locate a layout file on your hard drive and then click Import to add a layout to the list of available layouts.

Service Manager Policy Descriptions

Policy ID Mappings

The following table lists associations between watched attributes and monitor Policy IDs (1-21) for standard monitoring policies that are shipped with Service Manager.

NOTE
User-defined policies begin with ID value 1000 (and are incremented by 1, 1001, 1002, and so on).

To view user-defined Policy IDs, access the following link: http://<server>:CA_Portal/spectrum/slm/policyrep.jsp.

Watched Attribute (AttrToWatch)	Monitor Policy ID (MonitorPolicy_ID)	Policy
Condition_Value	1	Condition Value Sum Greater Than Or Equal

Condition	2	Condition Redundancy
Condition	3	Condition High Sensitivity (Default Policy)
Condition	4	Condition Low Sensitivity
Condition	5	Condition Percentage
RM_Condition	6	Service Health Redundancy
RM_Condition	7	Service Health High Sensitivity
RM_Condition	8	Service Health Low Sensitivity
RM_Condition	9	Service Health Percentage
Contact_Status	10	Contact Status Redundancy
Contact_Status	11	Contact Status High Sensitivity
Contact_Status	12	Contact Status Low Sensitivity
Contact_Status	13	Contact Status Percentage
Port_Status	14	Port Status Redundancy
Port_Status	15	Port Status High Sensitivity
Port_Status	16	Port Status Low Sensitivity
Port_Status	17	Port Status Percentage
LatestErrorStatus	18	Response Time Redundancy
LatestErrorStatus	19	Response Time High Sensitivity
LatestErrorStatus	20	Response Time Low Sensitivity
LatestErrorStatus	21	Response Time Percentage

Condition Value Sum Greater Than Or Equal

Watched attribute: Condition_Value

Default reason: The condition value has violated the allowable threshold.

This policy is different from the other standard Service Manager policies. Other policies include an attribute map, which defines the attributes that are monitored. This policy monitors the aggregate value of the Condition_Value (0x1000b)attribute for all resources that are associated with a service.

Rule Set

- When the sum of the Condition_Value attribute for all monitored resources is equal to or greater than the value of Red_Threshold (0x10012), the service is down.
- When the sum of the Condition_Value attribute for all monitored resources is equal to or greater than the value of Orange_Threshold (0x10011), the service is degraded.
- When the sum of the Condition_Value attribute for all monitored resources is equal to or greater than the value of Yellow_Threshold (0x10010), the service is slightly degraded.

You can adjust the values for Red_Threshold, Orange_Threshold, and Yellow_Threshold on the service model, or can adjust Value_When_Red (0x1000e), Value_When_Orange (0x1000d), or Value_When_Yellow (0x1000c) on the service component resource models to obtain the desired behavior.

Port Status Policies

Watched attribute: Port_Status

Default Reason: Bad Port Status

Value Mapping

- Service Health = Down if the watched attribute value is down, disabled, or unreachable
- Service Health = Up if the watched attribute value is up

Port Status High Sensitivity

- When any 1 resource is Down then the service is Down.

Port Status Low Sensitivity

- When all resources are Down then the service is Down.
- When all resource(s) are Down then the service is Degraded.

Port Status Redundancy

- When all resources are Down then the service is Down.
- When any 1 resource(s) is Down then the service is Slightly Degraded.

Port Status Percentage

- When 75 percent of the resources are Down then the service is Down.
- When 50 percent of the resources are Down then the service is Degraded.
- When 25 percent of the resources are Down then the service is Slightly Degraded.

Port Status - Automatically Configured

PollPortStatus has been automatically configured for interface models which are added or removed from services or resource monitors. Verify the following scenarios for Port Status:

- If an interface model has a PollPortStatus of False (No), when the model is added to a service or resource monitor PollPortStatus can be set to True (Yes). (OneClick View Attributes which imply false are displayed as ‘No’ and true are displayed as ‘Yes’.)
- If an interface model has a PollPortStatus of True, when the model is added to a service or resource monitor, no changes are made. When the model is removed from the service or resource monitor the PollPortSTatus is left as true.
- There is a caveat to the functionality described earlier. In scenarios where the PollPortStatus is automatically updated from false to true, that information is only preserved for the lifecycle of the running SpectroSERVER. The PollPortSTatus would not be restored to False when the interface is removed from the service, if the SpectroSERVER shutsdown between the time the interface was added and removed.

Condition Policies

Watched attribute: Condition_Value

Default reason: Bad Condition

Value Mapping

- Service Health = Down if the watched attribute value is Critical or Suppressed
- Service Health = Degraded if the watched attribute value is Major
- Service Health = Slightly Degraded if the watched attribute value is Minor
- Service Health = Up if the watched attribute value is Normal

Condition High Sensitivity (Default Policy)

- When any 1 resource(s) are Down then the service is Down.
- When any 1 resource(s) are Degraded then the service is Degraded.
- When any 1 resource(s) are Slightly Degraded then the service is Slightly Degraded.

Condition Low Sensitivity

- When all resources are Down then the service is Down.
- When all resources are Degraded then the service is Degraded.
- When all resources are Slightly Degraded then the service is Slightly Degraded.
- When any 1 resource(s) are Down then the service is Degraded.
- When any 1 resource(s) are Degraded then the service is Slightly Degraded.

Condition Redundancy

- When all resources are Down then the service is Down.
- When all resources are Degraded then the service is Degraded.
- When all resources are Slightly Degraded then the service is Slightly Degraded.
- When any 1 resource(s) are Down then the service is Slightly Degraded.

Condition Percentage

- When 75 percent of the resources are Down then the service is Down.
- When 50 percent of the resources are Down then the service is Degraded.
- When 25 percent of the resources are Down then the service is Slightly Degraded.

Response Time Policies

Watched attribute: LatestErrorStatus

Default reason: Bad Response Time

Value Mapping

- Service Health = Down if the watched attribute value is Timeout or Threshold_Critical
- Service Health = Degraded if the watched attribute value is Threshold_Major
- Service Health = Slightly Degraded if the watched attribute value is Threshold_Minor
- Service Health = Up if the watched attribute value is OK

Response Time High Sensitivity

- When any 1 resource(s) are Down then the service is Down.
- When any 1 resource(s) are Degraded then the service is Degraded.

Response Time Low Sensitivity

- When all resources are Down then the service is Down.
- When all resources are Degraded then the service is Degraded.
- When any 1 resource(s) are Down then the service is Degraded.
- When any 1 resource(s) are Degraded then the service is Slightly Degraded.

Response Time Redundancy

- When all resources are Down then the service is Down.
- When all resources are Degraded then the service is Degraded.
- When any 1 resource(s) are Down then the service is Slightly Degraded.

Response Time Percentage

- When 75 percent of the resources are Down then the service is Down.
- When 50 percent of the resources are Down then the service is Degraded.
- When 25 percent of the resources are Down then the service is Slightly Degraded.

Service Health Policies

Watched attribute: RM_Condition

Default reason: Bad Service Health

Value Mapping

- Service Health = Down if the watched attribute value is Down
- Service Health = Degraded if the watched attribute value is Degraded
- Service Health = Slightly Degraded if the watched attribute value is Slightly Degraded
- Service Health = Up if the watched attribute value is Up

Service Health High Sensitivity

- When any 1 resource(s) are Down then the service is Down.
- When any 1 resource(s) are Degraded then the service is Degraded.
- When any 1 resource(s) are Slightly Degraded then the service is Slightly Degraded.

> **NOTE**
> This is the default Policy for a SM_Service model that monitors resource monitor models/monitor groups/ SM_AttrMonitor models.

Service Health Low Sensitivity

- When all resources are Down then the service is Down.
- When all resources are Degraded then the service is Degraded.
- When all resources are Slightly Degraded then the service is Slightly Degraded.
- When any 1 resource(s) are Down then the service is Degraded.
- When any 1 resource(s) are Degraded then the service is Slightly Degraded.

Service Health Redundancy

- When all resources are Down then the service is Down.
- When all resources are Degraded then the service is Degraded.
- When all resources are Slightly Degraded then the service is Slightly Degraded.
- When any 1 resource(s) are Down then the service is Slightly Degraded.

Service Health Percentage

- When 75 percent of the resources are Down then the service is Down.
- When 50 percent of the resources are Down then the service is Degraded.
- When 25 percent of the resources are Down then the service is Slightly Degraded.

Contact Status Policies

Watched attribute: Contact_Status

Default reason: Bad Contact Status

Value Mapping

- Service Health = Down if the watched attribute value is Lost
- Service Health = Up if the watched attribute value is Established

Contact Status High Sensitivity

- When any 1 resource(s) are Down then the service is Down.

Contact Status Low Sensitivity

- When all resources are Down then the service is Down.
- When any 1 resource(s) are Down then the service is Degraded.

Contact Status Redundancy

- When all resources are Down then the service is Down.
- When any 1 resource(s) are Down then the service is Slightly Degraded.

Contact Status Percentage

- When 75 percent of the resources are Down then the service is Down.
- When 50 percent of the resources are Down then the service is Degraded.
- When 25 percent of the resources are Down then the service is Slightly Degraded.

Resource Monitor Implementation

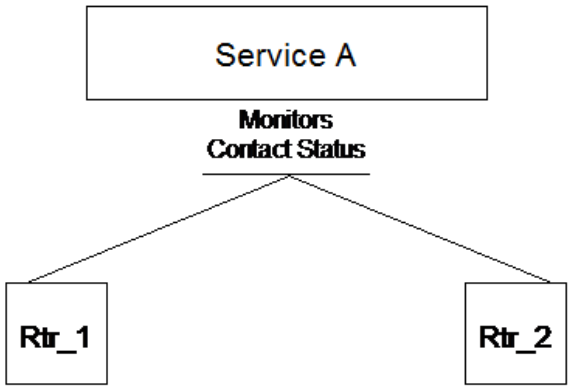
Policy Implementation Monitor Routers

Service A monitors two routers: Rtr_1 and Rtr_2. Service A functions if Rtr_1 or Rtr_2 is up, but it cannot function if both routers are down. Service A uses the Contact Status Low Sensitivity policy to monitor the service.

This policy rule set stipulates the following information:

- If all resources are down, service is down.
- If any 1 resource(s) are down, service is degraded.

Service with Policy



Service A seems to adequately monitor Rtr_1 and Rtr_2, but, on second glance, not enough to determine Service A actual viability. What about the case where Service A is not only dependent on the routers but also on particular router ports? The Contact Status Policy only monitors whether the routers are up or down.

The contact status of the routers could be established, simultaneously the ports on which Service A depends could be unavailable. Because of the limited scope of monitoring provided by its policy, Service A appears viable when actually it is not. Service A requires the more precise method of monitoring its resources that resource monitors provide.

Resource Monitor Implementation Monitor Routers and Their Ports

Scrutiny of the policy implementation reveals that it does not monitor the status of router ports that support Service A. The ports must be monitored with contact status of the routers. In addition, ports 4 and 5 on each router must be available for Service A to function optimally and that at least two of the four ports must be available for Service A to function adequately.

The following two resource monitors can be created as resources of Service A:

- Router Contact Resource Monitor -- Monitors the contact status of Rtr_1 and Rtr_2 (same as the policy implementation).
- Port Status Resource Monitor -- Monitors the port status of ports 4 and 5 on each router. Because at least two of four ports must be available. This resource monitor uses a Port Status Percentage Policy that reports down when 75 percent of the ports are unavailable and degraded when 50 percent of the ports are unavailable.

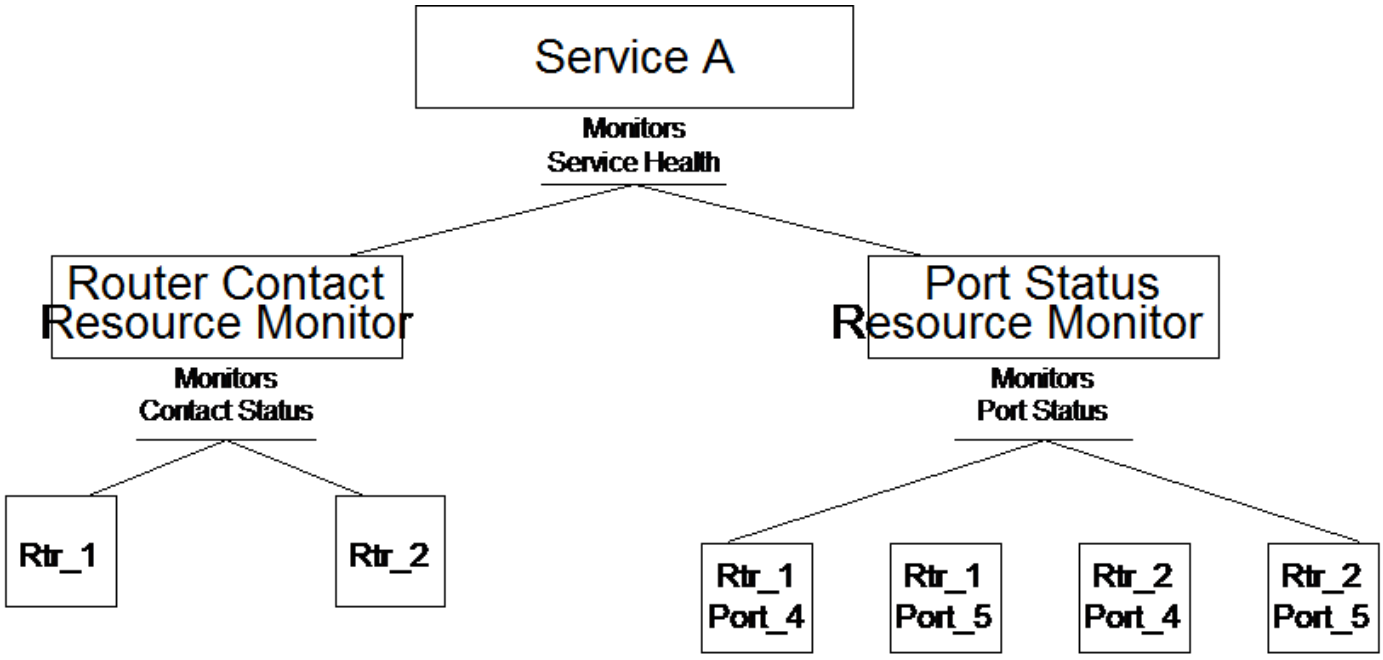
The result is that Router Contact Resource Monitor and Port Status Resource Monitor become resources of Service A. Rtr_1 and Rtr_2 become resources of the Router Contact Resource Monitor. Ports 4 and 5 on each router become resources of the Port Status Resource Monitor.

Service A monitors the service health attribute of Router Contact Resource Monitor and Port Status Resource Monitor with the Service Health High Sensitivity Policy. This policy rule set stipulates the following information:

- When any 1 resource(s) are Down then the service is Down.
- When any 1 resource(s) are Degraded then the service is Degraded.
- When any 1 resource(s) are Slightly Degraded then the service is Slightly Degraded.

So, if either one of the resource monitors is Down, Service A is down.

Resource Monitors Monitor Routers and Ports



Refined Resource Monitor Implementation Monitor Routers, Ports, and Response Time Tests

Further scrutiny of service A reveals that database server responsiveness and FTP transfer time are critical to its functionality. Assume that two RTM_Test models in DX NetOps Spectrum define critical, major, and minor thresholds for

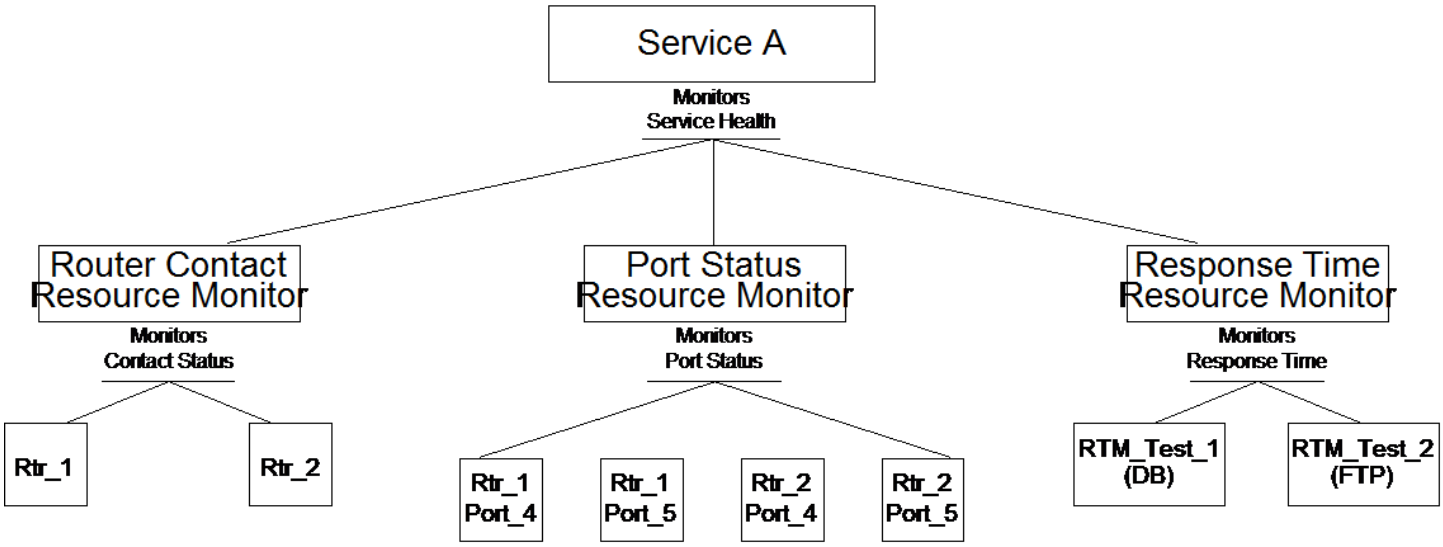
response time to the database server and the FTP server. These tests can be resources of another resource monitor (the Response Time resource monitor), which becomes another resource of Service A.

Response Time Resource Monitor monitors the response time tests with the Response Time High Sensitivity Policy. This Policy rule set stipulates the following information:

- When any 1 resource(s) are down then the service is down.
- When any 1 resource(s) are degraded then the service is degraded.

Therefore, if this resource monitor reports down (either test indicates unacceptable response time latency) then Service A is down as stipulated by the Service Health High Sensitivity Policy which monitors all Service A Resource Monitors.

Resource Monitors Monitor Routers, Ports, and Response Time



RollMaintenanceToResourceMonitor Attribute

When you use a Resource Monitor for multiple Services that are in active and Maintenance Mode states, the following attribute allows you manage the resource monitor behavior for services in maintenance mode.

Attribute	Description	Possible Values
RollMaintenanceToResourceMonitor	Allows you to manage the Resource Monitor status for Services in Maintenance Mode.	YES (default) NO

Example:

When the RollMaintenanceToResourceMonitor attribute value is set to ‘YES’ (default value)

If you add a resource monitor from an active service to a service that is in maintenance mode, the resource monitor turns into maintenance mode in all the services where it is being used.

When the RollMaintenanceToResourceMonitor attribute value is set to ‘NO’

If you add a resource monitor from an active service to a service that is in maintenance mode, the resource monitor does not turn into maintenance mode in all the services. The resource monitor continue to be in its state in other active services.

Administration and Maintenance

Customize a Service Editor Information Table

You can specify the information tables for services, customers, SLAs, and SLA templates in the Service Editor dialog. Specify the information types (columns) to include, the sort order (by status, by name, or by date for example), and the font and text size. You can revert to default settings. For more information about customizing interface settings, see the Using OneClick section.

Follow these steps:

1. Open the Service Editor.
2. Select the tab for which you want to customize an information table and right-click any column heading.
   The Table Preferences dialog appears.

   NOTE

   In the Table Preferences dialog, the column name 'Current MOT' is now renamed as 'Peak Outage Time'. The 'Peak Outage Time' indicates the maximum outage time a service has undergone.
3. Configure the table properties and click OK.
   The information table is customized.

Customize a Service Policy Editor Information Table

You can specify the information tables for policies, attribute maps, and rule sets in the Service Policy Editor dialog. Specify the information types (columns) to include, the sort order (by status, by name, or by date, for example), and the font and text size. You can also revert to default settings. For more information about customizing interface settings, see the Using OneClick section. .

Follow these steps:

1. Open the Service Editor.
2. Select the tab for which you want to customize an information table and right-click any column heading.
   The Table Preferences window appears.

   NOTE

   In the Table Preferences dialog, the column name 'Current MOT' is now renamed as 'Peak Outage Time'. The 'Peak Outage Time' indicates the maximum outage time a service has undergone.
3. Configure the table properties and click OK.
   The information table is customized.

Remove Service Manager Historical Data from All Landscapes

You can remove Service Manager historical data from all landscapes.

NOTE
Once you remove historical data from the database, you can no longer generate reports about them.

To remove Service Manager historical data from all landscapes, run the following script:

- Windows:
  ```
  <$SPECROOT>\bin\SMIntializeDB.bat
  ```
- UNIX/Linux:
  ```
  <$SPECROOT>/bin/SMIntializeDB
  ```

Remove Service Manager Historical Data from a Single Landscape

You can remove Service Manager historical data from a single landscape.

> **NOTE**
> Once you remove historical data from the database, you can no longer generate reports about them.

To remove Service Manager historical data from a single landscape, run the following script:

- Windows:
  ```
  <$SPECROOT>\bin\SMInitializeLandscape.bat <lh>
  ```
  - **lh**
    Indicates landscape handle.
- UNIX/Linux:
  ```
  <$SPECROOT>/bin/SMInitializeLandscape <lh>
  ```
  - **lh**
    Indicates landscape handle.

Remove Destroyed Service Manager Models from All Landscapes

You can remove destroyed Service Manager models from all landscapes.

Note: Once you remove destroyed Service Manager models from the database, you can no longer generate reports about them.

To remove destroyed Service Manager models (services, SLAs, and so on) and historical data for those models from all landscapes, run the following script:

- Windows:
  ```
  <$SPECROOT>\bin\SMRemoveDestroyedModels.bat
  ```
- UNIX/Linux:
  ```
  <$SPECROOT>/bin/SMRemoveDestroyedModels
  ```

Custom Resources Table

If a service using a policy with a custom attribute map is created, customize the Resources table to display the data you want to view about the monitored attribute that is specified in the custom attribute map. The Resources table appears in the following locations:

- Resources link under the OneClick Information tab
- List tab in OneClick Contents panel
- Service Dashboard List tab
- Resources tab in Service Editor

For more information about customizing OneClick interface elements, see the OneClick Customization section.

The following table lists Resources table configuration files for attributes that are monitored by standard Service Manager attribute maps:

Attribute	Attribute ID	File
Contact Status	0x10004	table-resources-0x10044-config.xml
Condition	0x1000a	table-resources-0x1000a-config.xml
ConditionValue	0x1000b	table-resources-0x1000b-config.xml
Port Status	0x10f1b	table-resources-0x10f1b-config.xml
Service Health	0x12a40	table-resources-0x12a40-config.xml

Response Time	0x456008c	table-resources-0x456008c-config.xml

These default files are located in the following directory:

```
<$SPECROOT>/tomcat/webapps/<sp>/WEB-INF/slm/config
```

Suppose, for example, you have a service that monitors “load in” data on a set of port resources. In this case, you can create a custom Resources table to display NRM_PortLoadIn (0x12aad) attribute data for a set of ports that are monitored by the service instead of the default Condition attribute data. The NRM_PortLoadIn (0x12aad) attribute example is used in the sections that describe how to configure the custom file.

Create the Custom Table File

You can create the custom file from scratch and can save it to the custom file directory, or you can save a modified version of the default table-resources-config.xml file from the following directory to the custom file directory:

```
<$SPECROOT>/tomcat/webapps/spectrum/WEB-INF/slm/config
```

Follow these steps:

1. Create the following custom file directory:
   ```
   <$SPECROOT>/custom/slm/config
   ```
2. Save all custom Resources table configuration files for Service Manager to this directory.
3. Create the custom file using the following naming convention:
   ```
   table-resources-<attribute ID>-config.xml
   ```
   For the load in attribute example:
   ```
   table-resources-0x12aad-config.xml
   ```

Example: Resources Table Configuration File

The following script shows an abbreviated example of the table-resources-0x12aad-config.xml configuration file. OneClick loads the table that is specified by this file to display the “load in” data column with other types of data for the port resources that are monitored by the service.

Elements pertaining to the “load in” example are highlighted in bold.

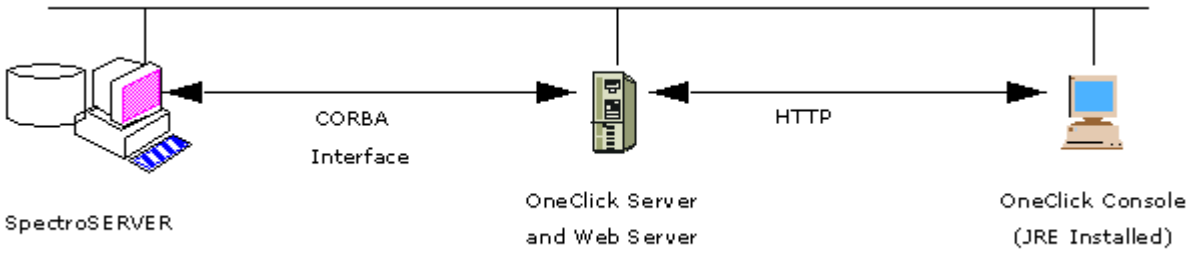
```
<?xml version="1.0" encoding="utf-8"?>

<table id="table-resources-0x12aad-config"
       xmlns="http://www.aprisma.com"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://www.aprisma.com
                           ../../common/schema/table-config.xsd">

  <orientation>horizontal</orientation>

  <swing-table-template>
    <show-vertical-lines>true</show-vertical-lines>
    <show-horizontal-lines>false</show-horizontal-lines>
  </swing-table-template>

  <swing-header-row-template>
    <static-color idref="row-header-color-config"/>
  </swing-header-row-template>
```

```
  <swing-row-template>
    <enumerated-color idref="alternatingrow-color-config"/>
  </swing-row-template>

  <column-list>
    <column>
      <name>Load In</name>
      <content>
        <attribute>0x12aad</attribute>
       </content>
      <default-width>125</default-width>
    </column>

    <column idref="column-normalizedstatus-config">
      <default-width>125</default-width>
    </column>

    <column idref="column-modelname-config">
      <default-width>125</default-width>
    </column>
</column-list>

  <default-sort>
    <sort-column-list>
      <sort-column>
        <name>
          com.aprisma.spectrum.app.topo.client.interfaces.render.NormalizedStatusColumn
        </name>
        <direction>ascending</direction>
      </sort-column>
      <sort-column>
        <name>
          com.aprisma.spectrum.app.util.render.ModelNameColumn
        </name>
        <direction>ascending</direction>
      </sort-column>
    </sort-column-list>
  </default-sort>
</table>
```

Using OneClick

OneClick delivers DX NetOps Spectrum information to network operators and troubleshooters using an intuitive graphical user interface. OneClick provides customized access to information and tools for users who monitor or troubleshoot specific portions of a network that DX NetOps Spectrum manages.

The OneClick architecture uses the Java Network Launch Protocol (JNLP) and the Java Web Start application to let remote systems and users access the OneClick server. JNLP is a standard for application delivery that does not require traditional installers or the launching of executable code. After installation, the OneClick Console communicates with the web server on the OneClick server using port 80 by default for Windows, or port 8080 for UNIX. The web server communicates as a single client of the SpectroSERVER using the CORBA interface, as illustrated by the following diagram:



The following DX NetOps Spectrum and OneClick content provides information that is related to installing, customizing, and maintaining OneClick:

- Fresh Install provides detailed instructions for installing the OneClick server and client.
- OneClick Administration provides information about how to configure and administer the OneClick environment, applications, and users.
- OneClick Customization provides information about customizing the OneClick interface by modifying XML files and other techniques.
- Modeling and Managing Your IT Infrastructure describes how to configure OneClick to discover and model the elements on the network.

How to Set Up the OneClick Client

The following process describes how to set up and start using the OneClick client.

> **NOTE**
> For more information about each of these steps, see Fresh Install .

Follow these steps:

1. Verify that your workstation meets the minimum OneClick client requirements before installing or running the OneClick client.
2. Install JRE. The OneClick Console and OneClick add-on applications require Java Runtime Environment (JRE). The JRE includes the Java Web Start client, which is required to run Java Network Launching Protocol (JNLP) applications like OneClick.
   You can install JRE 1.8.0_112 for 10.2/JRE 1.8.0_121 for 10.2.1 from OneClick home web page. To install JRE, click the "**Install JRE**" option on OneClick home page and follow the steps thereafter.
   After you install the JRE, you can start OneClick.
3. Associate .jnlp files with Java Web Start.
4. Launch the OneClick Console.

View the Client Details Web Page

The OneClick Client Details page lets you view the clients that you have opened. You can also log out of clients from this page. From 10.4.1, the Client Details table shows the client type in the WebApp column. In case the client is a WebApp, the Host Name and the Host Address columns show the **respective client** details. Previously, these columns displayed the **localhost** details, when the client was a WebApp.

NOTE
In case WebApp is launched on the same machine where OneClick is installed, and you launch the WebApp using **localhost** as the hostname, the client details page shows the hostname and host address as "0:0:0:0:0:0:0:1".

NOTE
This web page is not automatically updated with the latest client information. To verify that you have the latest information, reload the page in your browser.

How to View the Client Details Page

Follow these steps:

1. Navigate to http://<*webserver*>/spectrum/index.jsp in a web browser.
   The OneClick home page opens.
2. Click the Client Details link.
   The Client Details web page opens, displaying a Client(s) Logged On table.

How to Log off Clients Using the Client Details page

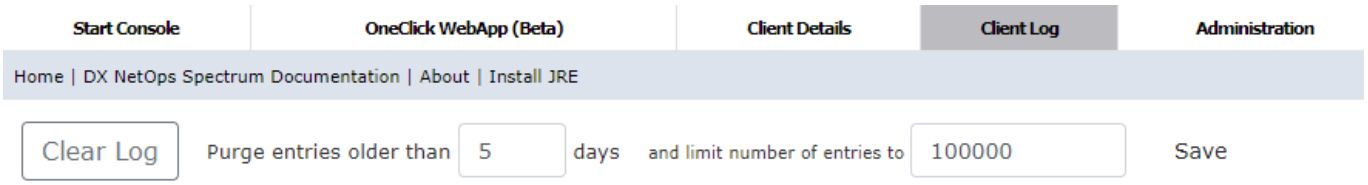
Follow these steps:

1. In the Client(s) Logged On table, select the checkboxes next to your user name for the clients that you want to log out.
2. Click Log off Clients.
   A confirmation dialog opens.
3. Click OK.
   The clients are logged out.

NOTE
Administrators accessing the Client Details page can view all currently logged in users. For more information, see OneClick Administration.

View Client Log

The OneClick Client Log page lets you view log history of the logged in and logged out users of the OneClick client. The details include login date, time, and host details. Using the Client Log page you can view logs, clear logs, purge entries, and limit the number of entries to display.



Follow these steps:

1. Navigate to http://<*webserver*>/spectrum/index.jsp in a Web browser.
   The OneClick home page opens.
2. Click the Client Log link.
   The Client Log web page opens, displaying log of the logged in users.
3. Click the 'Clear Log' button to clear the entire log details.
4. Enter the number of days in the 'Purge entries older than days' field to delete the entries, which are older that the given days.

NOTE

Starting from 10.2.2 release, the Purge functionality of the log entries is enhanced to purge by size as well as by time. The purge timer gets triggered with Tomcat startup and runs every hour and purges the client log data based on the configuration provided (i.e, number of days and number of entries).

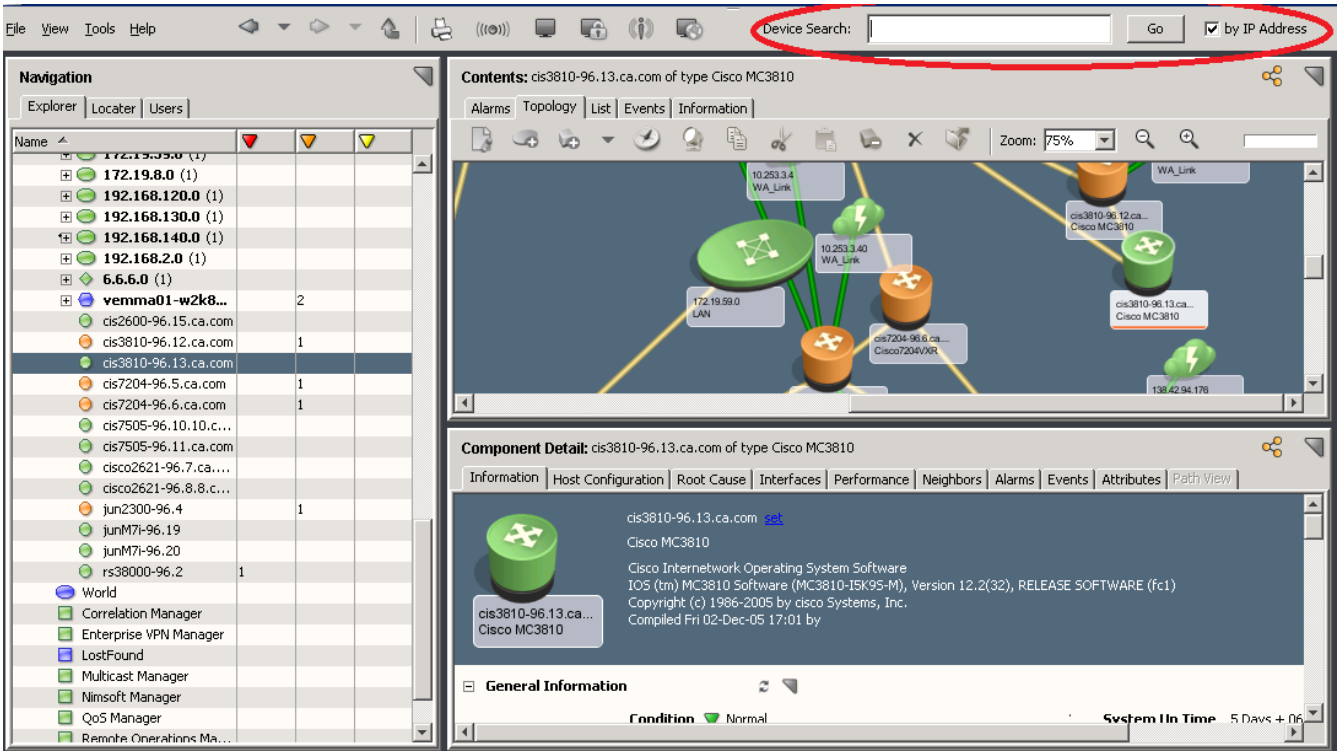
5. Enter the number of entries in the 'Limit number of entries to' field to display only the given entries. (applies from 10.2.2)
6. Click the Save button to save the configuration.

NOTE

Information! When multiple users access the Client Log file, If the client log is huge then you may encounter Out of Memory issue for OneClick Web server. To resolve this issue, you need to change the purge interval accordingly so that the entries older than the given number are purged automatically and the log will have less entries. You can also limit the number of entries to display so that only those many entries are fetched in the log.

OneClick Console User Interface

The OneClick Console user interface comprises three panels that display information about your network assets: the Navigation panel, the Contents panel, and the Component Detail panel. The following image shows an example of the OneClick Console user interface:



The information that is displayed in each panel depends on the item that is selected in the Navigation panel. Each panel displays a different context. The titles of the Content and Component Detail panels describe the context. Tabs in the Contents and Component Detail panels provide detailed lists of devices, alarms, events, and other information about specific items.

You can customize the display by docking, cloning, or removing panels.

NOTE
The OneClick Console supports multiple add-on applications, such as VPN Manager, Service Performance Manager, Multicast Manager, and Service Manager. For more information, see the user documentation that is provided with those applications.

Navigation Panel

You can use the Navigation panel to access information about your network assets. The Navigation panel includes the following features:

- Alarm views
- Topology views
- Device lists
- Event views
- Detailed device information
- Containers
- Landscapes
- OneClick applications
- Searches

The Navigation panel is on the left side of the default OneClick user interface. Two tabs are available to OneClick operators: the Explorer tab and the Locater tab.

Explorer Tab

The Explorer tab in the Navigation panel displays a hierarchical view of landscapes, containers, OneClick applications, and device models. In this view, container and device icons indicate the model class and status of each container and device model. The OneClick administrator at your organization created the views in the Explorer tab by modeling devices to represent your network infrastructure.

NOTE
OneClick filters containers and devices from the Explorer view if they are child objects of a container to which you lack view permissions.

The Explorer tab shows a high-level view of alarms that are active for devices in each container and application. You can modify the default alarm view in the Explorer tab. For more information, see Customize Columns.

Use the Explorer tab to select a container. You can then view information, alarms, events, lists, and topologies for that selection in the Contents panel. You can also select the OneClick application in the Name column to view information, alarms, events, lists, and topologies for that selection. You can also expand and collapse containers and applications in the Explorer tab as necessary.

NOTE
Items in the Explorer tab are sorted numerically.

Locater Tab

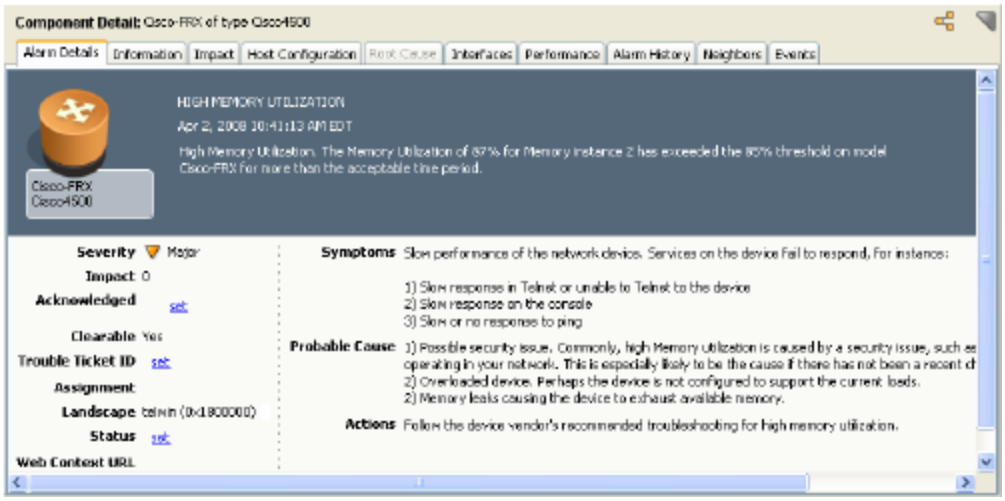
The Locater tab provides search functionality for locating network assets (device and application models) and viewing details about them. Search results appear in the Results tab of the Contents panel. Detailed information about network assets selected in the results list appears in the Component Detail panel.

Contents Panel

The Contents panel is located in the upper right of the OneClick interface. The information in the Contents panel depends on the context that is set from the Navigation panel. If the Locater tab in the Navigation panel is the active tab, the Results tab appears in the Contents panel. The Results tab shows the results of the last search performed in the current user session. If the Explorer tab is the active tab in the Navigation panel, the Contents panel displays the Alarms, Topology, List, Events, and Information tabs for the selected device, container, or application. By default, the Alarms tab is the active tab in the Contents panel.

Component Detail Panel

The Component Detail panel displays more detailed information for the item that is selected in the Contents panel. The following image shows an example of the Component Detail panel.



OneClick Tabs

OneClick categorizes information by tabs that appear in both the Contents and Component Detail panels. The tabs that you see depend on the context, which depends on the current selection in the Navigation panel. The following list describes these OneClick tabs in detail.

- **Alarms tab**
  Appears in either the Contents or Component Detail panel depending on context.
- **Alarm Details tab**
  Appears in the Component Detail panel and shows detailed information about the alarm that is selected in the Alarms tab.
- **Topology tab**
  Displays network topology models that are created manually or by Discovery.
- **List tab**
  Displays all models of the container that is selected in the Explorer tab.
- **Events tab**
  Displays events for the container, model, or application that is selected in either the Explorer tab or the Contents panel. The display includes all alarms and events for the selected model.
- **Information tab**
  Displays details about the container, device, or application that is selected.
- **Host Configuration**

Capture, view, upload, and export device configuration files.

NOTE
For more information, see the Network Configuration Manager section.

- **Impact tab**
  Displays the impact and symptoms for a selected alarm.

  NOTE
  The state of devices in the Impact tab does not always reflect the current device state.
- **Root Cause tab**
  Displays the root cause for a device that is down. The root cause can be helpful when a device has multiple alarms. You can view the date and time of occurrence of each alarm in the "Date/Time" column. You can also view the Condition, Name of the model, Alarm Title, and more information in other columns.
- **Interfaces tab**
  Displays interface information for the selected alarm or device.
- **Performance tab**
  Displays performance information for the selected device, including CPU & Memory Utilization.
- **Alarm History tab**
  Displays the historical information for the selected alarm including associated events, status, when created, and cleared. For general device alarm history, use the Events tab.
- **Neighbors tab**
  Displays the model that is selected from either the Explorer tab or the Contents panel and any models that are directly connected to it.

Alarms Tab

The Alarms tab displays any alarms that exist for the device, container, or application that is selected in the Navigation panel Explorer tab. Your OneClick administrator would have preconfigured the view of the alarms to show only a subset of alarms available.

Select an alarm in the Alarms tab to display detailed information for that alarm in the Component Detail panel. The toolbar that is displayed at the top of the Alarms tab lets you quickly process alarms in OneClick.

Alarm Severity Colors

Color	Severity	Description
Blue	Initial	Contact with device is not established.
Gray	Suppressed	Device cannot be reached due to a known error condition that exists on another device.
Brown	Maintenance	Device is offline for maintenance purposes.
Red	Critical	A loss of service has occurred; immediate action is required.
Orange	Major	A loss of service has occurred or is impending; action is required within a short period of time.
Yellow	Minor	A situation has occurred that does not require immediate action. This severity is also used for alarms created to convey information only, such as "Duplicate IP."

Green	Normal	Contact has been made with the device. Device is operating normally. No alarms are associated with this device.

Alarms List Columns

The columns in the table categorize the information for each alarm that is displayed in the Alarms tab. The default alarm information categories include Severity, Date/Time, Name, Network Address, Type, Acknowledged, Alarm Type, and Landscape. Other categories are present if OneClick add-on applications are installed. And the OneClick administrator can create custom alarm categories.

You can select the columns to display in the Alarms list. As with all table columns in OneClick, you can sort on the content for each column by clicking the column heading. Click a column header to toggle the sort order.

Filter the Alarms List

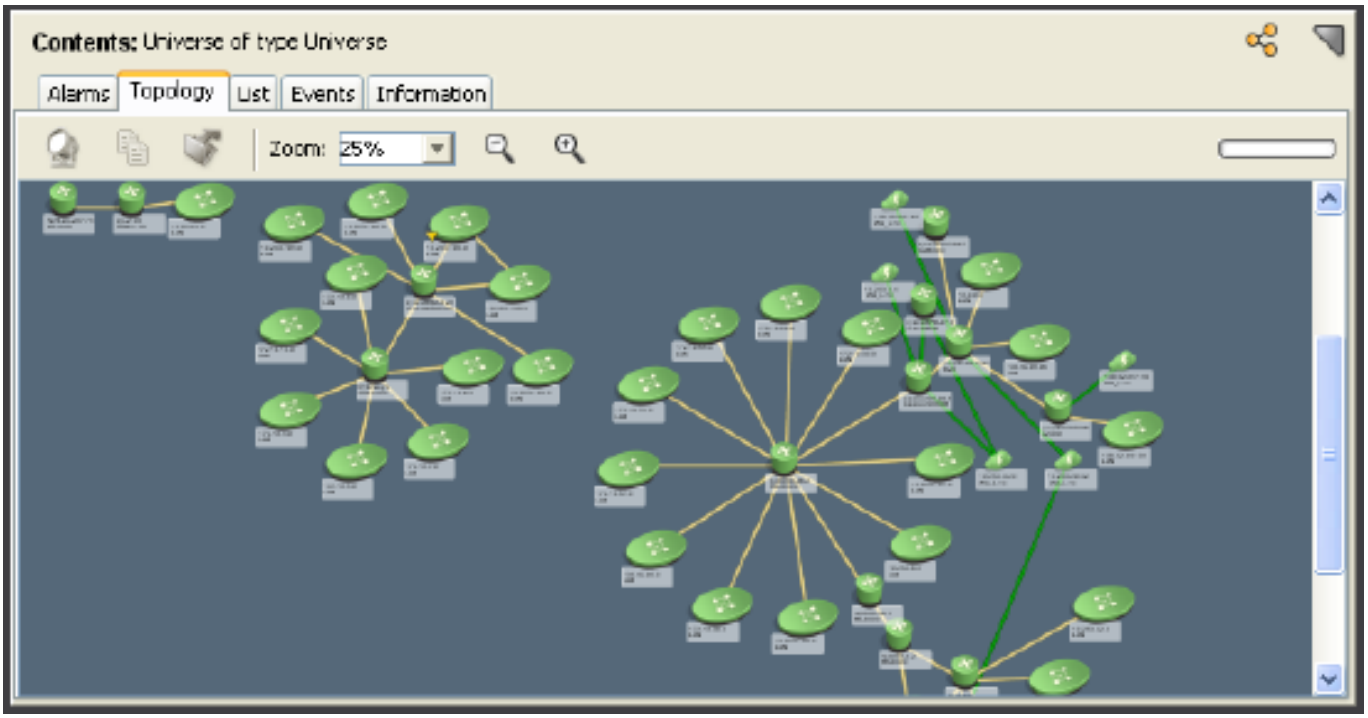
You can filter the alarms list as follows:

- Select Show or Hide from the Filter drop-down list and enter text in the Filter text box. As you type, the Alarms list displays or hides only those alarms with attributes that match the current text in the Filter field.
  This behavior depends on your Filter drop-down list selection. For example, view only critical alarms by selecting Show from the Filter drop-down list and typing **crit** in the text box.
- You can also create alarm filters to save and reuse.

Topology Tab

The Topology tab appears in the Contents panel and shows network topology diagrams for selected containers and collections. Expand the view of a container or collection in the Explorer tab, and then select a specific device in that container. The Topology view shows that device in the Contents panel. Double-click an icon in a topology view to change the context of the Navigation panel to display that device, container, or application. The Topology tab is not available if either My Spectrum or a landscape is selected in the Explorer tab.

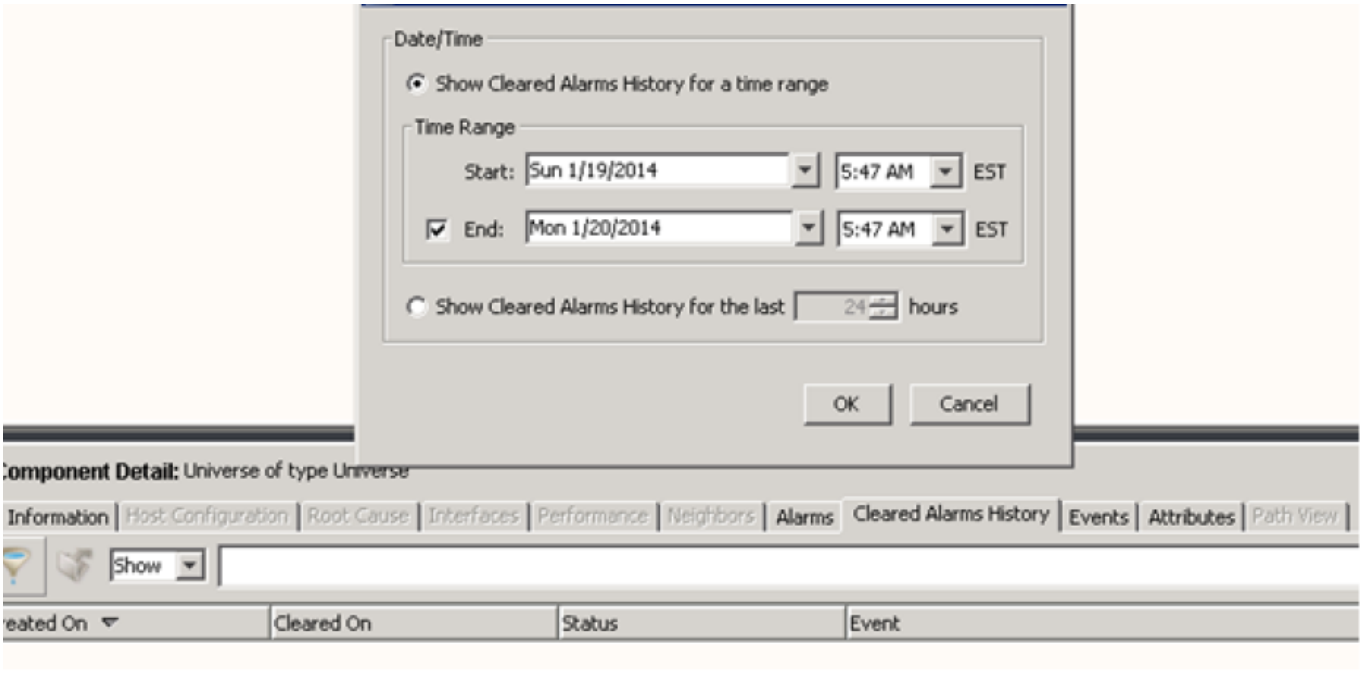
The following image shows an example of the Topology tab view:



Cleared Alarms History Tab

You can view the Cleared Alarms History tab from the Component Detail panel. The Cleared Alarms History tab displays historical information about the cleared alarms for the selected model. Historical information includes associated events and the status of created and cleared alarms. By default, cleared alarm history is displayed for the last 24 hours. You can use the Time Range option to view the cleared alarms history of a specific period.

The following image shows the Cleared Alarms History tab view:



List Tab

The List tab displays all models of the container that is selected in the Explorer tab.

If a device is selected in the Explorer tab, all models are displayed for the container of that device. This view is updated when models are added or removed or when attributes are updated. Double-click an entry in the List tab to select that model in the Explorer tab. This tab is not available when My Spectrum or any landscapes are selected in the Explorer tab.

Results Tab

The Results tab displays the results of searches performed in the Locater tab in the Navigation panel.

Events Tab

The Events tab appears in either the Contents or Component Detail panel. The Events tab displays all events for the item that is selected in either the Explorer tab or the Contents panel. If you select My Spectrum, the Events tab appears only in the Component Detail panel.

Information Tab

The Information tab appears in either the Contents or Component Detail panel, depending on the context set. The Information tab displays detailed device configuration information, VLAN and VPN configuration settings, and more. The Information tab is not available when My Spectrum or any landscapes are selected in the Explorer tab.

OneClick Toolbars

Toolbars are available in several OneClick panels and tabs. OneClick toolbars use graphical buttons and icons to provide quick access to features and functionality.

Hide Toolbars

By default, all available toolbars are shown in the OneClick Console. You can hide the toolbars that you do not use.

Follow these steps:

1. Select View, Toolbars.
   A submenu lists the available toolbars. Toolbars that are visible are checked.
2. Click a checked toolbar to hide it.
   The menu closes and the toolbar is removed from the applicable view.

Main Toolbar

The Main toolbar, which appears at the top of the OneClick Console, contains buttons for completing tasks common to many OneClick applications. The following table describes the buttons of the Main toolbar:

Button	Description
	**Navigation:** Moves you among views that you have recently accessed. Arrows select views from a list.
	**Go Up:** Moves you up to the next level in the hierarchy. A tooltip indicates the next level.
	**Ping:** Sends an ICMP Ping to the selected devices from the SpectroSERVER modeling the device.
	**TraceRoute:** Performs the traceroute operation on SpectroSERVER- and SDC-moldeled devices. Traceroute is a tool that helps you diagnose the network. It displays the route. It also helps in measuring the transit delays of packets that take place across the Internet Protocol network. For directly connected devices, this option performs the traceroute operation through SpectroSERVER. For SpectroSERVER-modeled devices, the traceroute prints the TTL for each hop with respect to SpectroSERVER. For devices modeled through SDC, check the connection status and perform the traceroute operation. In this case, the traceroute operation is performed through SDC. Additionally, the traceroute prints the TTL for each hop with respect to SDC.
	**Telnet:** Establishes a communication session with the selected device using Telnet from the SpectroSERVER modeling the device.
	**Secure Shell:** Establishes an encrypted communication session with the selected device using Secure Shell (SSH), from the SpectroSERVER modeling the device.
	**Poll:** Initiates contact with the selected devices from the SpectroSERVER modeling the device.
	**Web Administration:** Opens a browser using the IP address of the selected device. Available only for models that have the WebAdminURL attribute.

NOTE
You can also access the functions of the Main toolbar from the File, View, and Tools menus or, from the right-click menu. This selection depends on the current view.

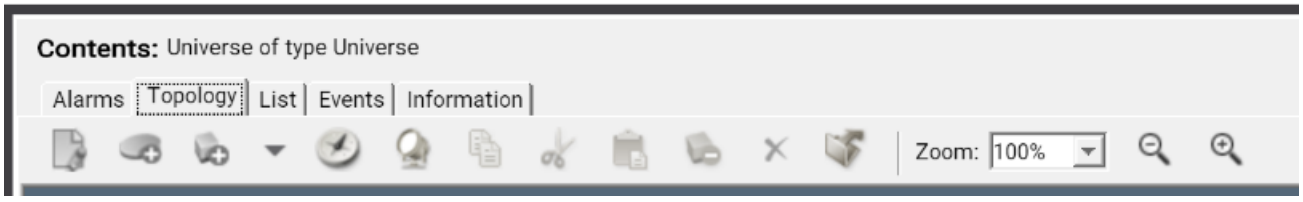
Alarms Toolbar

The Alarms toolbar contains buttons for quickly process alarms in OneClick. The following table describes the buttons and functionality that are available in the Alarms toolbar.

Button	Description
	**Information:** Opens a dialog containing details about the alarm currently selected in the table.
	**Clear selected alarms:** Clears the selected alarms. Cleared alarms are removed from the **Alarms** table.
	**Acknowledge selected alarms:** Acknowledges the selected alarms.
	**Unacknowledge selected alarms:** Unacknowledges the previously acknowledged, selected alarms.
	**Assign a troubleshooter:** Opens the Select Troubleshooter dialog from which you can assign a troubleshooter to the selected alarms.
	**Unassign the troubleshooter:** Removes a troubleshooter from the selected alarms.
	**Update alarm attributes on the selected alarm(s):** Opens the Update Alarm Attributes dialog from which you can update values for certain alarm attributes, such as a trouble ticket number or an acknowledgment.
	**Mail:** Opens the Mail Selected Alarms dialog, from which you can email alarms to a recipient. The mail is populated with alarm information. The mail content has URLs of OneClick console and WebApp (added in 10.4.2.1).
	**Unsnooze:** Unsnoozes all previously snoozed alarms.
	**Alarm Filter:** Create an alarm filter.
	**Export:** Specify a file format and location to export the Alarms list.
Filter	**Filter:** Supply text to filter the current **Alarms** list view. Select Show or Hide from the Filter drop-down list to specify whether to show or hide filter matches.

Topology Toolbar

The Topology toolbar appears in the Topology tab in the Contents panel. The following image shows an example of the Topology toolbar.



You can use the Topology toolbar to perform the following tasks:

- Spotlight router redundancy, configured VLANs, VPNs, and LSP Paths.

  NOTE
  LSP Path spotlighting is not available if you do not have MPLS Manager installed. VPN spotlighting is not available if you do not have VPN Manager installed.

- Copy selected items to the paste buffer so that you can paste them to another OneClick field or another application.
- Export the contents of the selected Topology view to a file.
- Adjust the Topology tab view for the current session only by zooming in or out.

  NOTE
  Users with administrative privileges also have access to editing tools in the Topology toolbar. For more information, see the Modeling and Managing Your IT Infrastructure .

Neighbors Toolbar

The Neighbors toolbar appears in the Neighbors tab in the Component Detail panel. The Neighbors toolbar functions like the Topology tab toolbar. However, it lacks a Spotlight button.

OneClick Status Bar

The Status bar is at the bottom of the OneClick Console and provides the following functionality:

- Displays information about the OneClick infrastructure. For example, you can see the connection status of the servers and services that let OneClick provide accurate, real-time network information.
- Lets you view messages from OneClick administrators.
- Identifies the username that is used to log in to the current OneClick session, and the name of the OneClick server to which the current client is connected.
- Lets you change your password using a "Change Password" link.

Using and Customizing OneClick

Change Your OneClick Password

You can change your OneClick password from the OneClick home page or the status bar in the OneClick Console.

Follow these steps:

1. Navigate to *one* of the following locations:
   – The bottom of the OneClick home page.
   – The status bar at the bottom of the OneClick Console.
2. Click the 'Change Password' link.
3. Enter your current password, your new password, and reenter your new password.
   Your password is changed.

Favorites Folder

The Favorites folder is something all OneClick users can populate and maintain for their own benefit, without administrator assistance.

In the Explorer tab of the Navigation panel, you can add any OneClick element below the landscape level to your Favorites folder by right-clicking the element and choosing Add To, Favorites. You can also add Global Collections to your favorites by right-clicking your Favorites folder and choosing Add Collections.

To remove an element from the Favorites folder, right-click the element within the Favorites folder and choose Remove.

> **WARNING**
> If you right-click an element and select Delete, the element is not only removed from the Favorites folder, but some models can also be deleted permanently from the system. For more information, see the Modeling and Managing Your IT Infrastructure .

You can create subfolders by right-clicking Favorites (or a subfolder within Favorites) and selecting New Folder. Use the right-click menu to cut, copy, paste, rename, and delete subfolders.

Set SPECTRUM_BROWSER Variable

By default, OneClick uses the Mozilla Firefox browser on Linux systems. For Windows systems, OneClick uses the default browser as defined on the system. You can override the default OneClick settings with the SPECTRUM_BROWSER environment variable.

Define SPECTRUM_BROWSER as part of your environment. Include the full path for the command that opens the browser of your choice. Use a placeholder to specify the URL in the SPECTRUM_BROWSER variable using {0} (<*full_path_browser*>/firefox.exe {0}).

When you click a link in OneClick, The URL link replaces the {0}.

The SPECTRUM_BROWSER variable overrides other platform or system browser selection variables.

Set OneClick Preferences

The Set Preferences dialog lets you customize your view of OneClick. You can set preferences in OneClick for a number of categories and add-on applications such as General settings, the Alarms tab, and the Explorer tab. The Set Preferences dialog provides access to these settings. When you select the top-level preferences group in the Set Preferences dialog, all available preferences and the tools to edit them appear.

Other preferences that are only available to administrators can be set for all users or for categories of users.

Follow these steps:

1. Click View, Preferences.
   The Set Preferences dialog opens.
2. Expand the folder for the individual preference or preference group that you want to change in the Name column.
3. Set new values for the selected preference in the right panel as desired.

   > **NOTE**
   > If Make Changes Permanent is selected at the bottom of the dialog, any preferences that you set become your default settings. If you clear this option, your changes only apply to the current session.
4. Click Apply.
5. Click OK.
   The preferences are set, and the Set Preferences dialog closes.

Alarms Tab Preferences

Use the Alarms Tab in the Set Preferences dialog to specify settings for OneClick alarms. Alarm preferences and categories available for modification include the following attributes:

- **Acknowledge When Assign**
  Specifies whether to auto-acknowledge alarms when assignments are made.
- **Alarm Filter**
  Sets the filter that is used for all displayed alarms tables. Click Set Alarm Filter to access the Alarm Filter dialog.
- **Alarm Notification**
  Specifies settings for alarm popup alerts and sounds:
  - **New Alarm Popup Alert**
    Specifies whether you see a popup alert for new alarms. You can also specify the duration and transparency of the popup.
  - **New Alarm Sound**
    Enables or disables sound notifications for new alarms. Sound notification is a male, English-speaking voice announcing the number and severity of alarms when they are generated.
- **Alarms Table**
  Specifies settings for how the alarms table displays including column order, available columns, sorting, and font.
- **Default Alarm Snooze**
  Specifies the default alarm snooze time. The value has to be greater than zero and less than 24 hours.
- **Email Subject Templates**
  Specifies the available templates that can be included in the Subject heading of an email message.
- **Email Templates**
  Specifies the available templates that can be included in an email message.
- **Expert Clear**
  Suppresses confirmations for clearing selected alarms.
- **Show secondary when in maintenance**
  Specifies whether to show secondary alarms for a device that is in maintenance mode.
- **Trouble Ticket URL**
  Specifies a URL in which to enclose the trouble ticket ID. Set the URL for a trouble ticket management system at your organization. The ticket number for an alarm appears as a hyperlink that opens a Web browser to the trouble ticket system URL. The trouble ticket ID can be substituted for the URL at run time by specifying “{0}” in the URL string. The following example shows this substitution:
  ```
  http://acme/ticket?id={0}
  ```

Display Initial and Suppressed Alarms

> **WARNING**
> Displaying Initial and Suppressed alarms is not recommended in OneClick. These alarms can create a significant amount of network traffic.

If the Disable Initial Alarms and Disable Suppressed Alarms settings for the Virtual Network Machine (VNM) managing your network are disabled, you can view initial and suppressed alarm conditions. Only users with OneClick administrator privileges can change these settings. Go to the Disable Initial and Suppressed Alarms attributes in the Alarm Management submenu for the VNM.

Follow these steps:

1. Click View, Preferences.
   The Set Preferences dialog opens.
2. Expand the Alarms Tab folder, select Alarm Filter, and click Set Alarm Filter.
   The Alarm Filter dialog opens.
3. Click Add.
   The Enter Filter Name dialog opens.
4. Enter a name for the new alarm filter and click OK.
5. Click the Severity tab in the Alarm Filter dialog.

6. Select the Initial and Suppressed alarm categories and click to move them to the Show list.
   Click OK.
7. The Alarm Filter dialog closes.
8. Click OK.
   The Set Preferences dialog closes.
9. Right-click the Name column header in the Explorer tab.
   The Table Preferences dialog opens.
10. Select the Initial Alarm Count and Suppressed Alarm Count check boxes.
11. Click OK.
    Initial and Suppressed alarms are displayed in OneClick.

Events Tab Preferences

Select Events in the Set Preferences dialog to set preferences for the Events tab. The following preferences are available for customization:

- **Default Time Interval**
  Specifies the default time interval that is used to retrieve events for display in the Events tab. OneClick uses this value initially to display events for a model. You can change this value using the Event Filter dialog.
- **Email Subject Templates**
  Specifies the available templates that can be included in the Subject heading of an email message.
- **Email Templates**
  Specifies the available templates that can be included in an email message.
- Events Table
  - **Columns**
    Specifies the columns of information to display in the Events table.
  - **Font**
    Specifies the font family and type size used in the Events table.
  - **Sort**
    Specifies the default sorting methodology for the Events table.
- **Filtered Event Types**
  Specifies the types of events to exclude from the Events table.
- **Show events for subcomponents**
  Specifies whether to show events for the subcomponents of the selected model in the Events tab. For example, ports are subcomponents. OneClick uses this value to display events for a model. You can change this value in the Event Filter dialog.

Exclude Event Types

You can exclude event types from displaying in the Events table.

How to Add Filtered Event Types to the Excluded Event Types List

Follow these steps:

1. Click View, Preferences.
   The Set Preferences dialog opens.
2. Expand the Events Tab folder in the Name column and click Filtered Event Types.
3. Click Browse under the Excluded event types list.
   The Select Event Type dialog opens displaying all the available event types.

   NOTE
   If you know the event code for the event type to exclude, enter it in the Filter box beneath the list of excluded event types and click Add.

4. Select the desired event from the Select Event Type dialog and click OK.
   The Select Event Type dialog closes. The event code appears in the text box beneath the list of excluded event types.
5. Click Add.
   The event type is added to the Excluded event types list.
6. Click Apply in the Set Preferences dialog.
   The event types you selected are now excluded and are not displayed in the Events table.

How to Remove Filtered Event Types from the Excluded Event Types List

Follow these steps:

1. Click View, Preferences.
   The Set Preferences dialog opens.
2. Expand the Events Tab folder in the Name column and click Filtered Event Types.
3. Select the excluded event type that you want to remove from this list.
4. Click Remove.
5. Repeat for other event types that you want to include in the Events table.
   The event types that you selected are again displayed in the Events table.

Email Templates

OneClick contains email templates that you can use to email alarms or events. These email templates let you automatically include specific values from the related alarm or event in your email messages. You can modify existing email templates from either the Preferences dialog or the email message dialog itself.

The following types of email templates are available in OneClick:

- **Subject Templates**
  Specifies the fields to include in the subject line of the email messages that use this template.
- **Message Templates**
  Specifies the fields to include in the body of the email messages that use this template.

Create Email Templates

You can create new email templates by editing existing email templates.

> **NOTE**
> The following procedure describes how to create email templates for alarms. The same steps apply to setting up email templates for events. For more information, see Events Tab Preferences.

Follow these steps:

1. Click View, Preferences.
   The Set Preferences dialog opens.
2. Expand the Alarms Tab folder in the Name column. Take one of the following steps:
   – Click Email Templates
   – Click Email Subject Templates
3. Select a template from the Templates drop-down list.
4. Click Edit to modify the selected template.
   The Edit Template dialog opens.
5. Enter a name for this new template in the Save As field.
6. Select the information that you want to display in the new message template.
   Each option corresponds to a column in the alarm.
7. (Optional) Click Move Up or Move Down to change the order in which the information appears in the message.
8. Click OK when you have finished creating the template.

The new template appears in the Templates drop-down list.

Modify Email Templates

You can modify existing email templates.

> **NOTE**
> The following procedure describes how to modify email templates for alarms. However, the same steps apply to setting up email templates for events.

Follow these steps:

1. Click View, Preferences.
   The Set Preferences dialog opens.
2. Expand the Alarms Tab folder in the Name column, and take one of the following steps:
   – Click Email Templates.
   – Click Email Subject Templates
3. Select a template from the Templates drop-down list.
4. Click Edit to modify the selected template.
   The Edit Template dialog opens.
5. Select the information to display in the message template.
6. (Optional) Click Move Up or Move Down to change the order in which the information appears in the message.
7. Click OK when you have finished modifying the template.
   The modified template is now available in the Templates drop-down list.

General Preferences

This section describes the general preferences available in the Set Preferences dialog. Depending on your access rights, you may not have access to all of the following settings.

- **SNMP Community Strings List**
  Lets you edit the stored SNMP community strings list. Use this setting to add SNMP community strings to the list, remove stored SNMP community strings that were typed incorrectly, or clear the entire list.

  > **NOTE**
  > The OneClick administrator can lock this preference.
- **Default Field Font**
  Specifies the default font that is used for all field views in the Information panels.
- **Default Table Font**
  Specifies the default font used in all OneClick tables. This setting can be overridden for a specific table by using the Table Preferences dialog.
- **Email Address List**
  Specifies the email addresses available from the To/Cc fields in Mail dialogs. You can add and remove addresses as needed. Also, any email address that you type manually into a Mail dialog in OneClick is automatically saved to this list. You can also add names along with the email address using the following format: *<RecipientA Name>*:*<RecipientA Email Address>.*
- **Locale**
  Specifies the regional locale that is used to format dates, time, and numbers. This setting overrides the operating system setting. You can maintain the same locale setting independent of the system where you are logged on.

  > **NOTE**
  > Restart the OneClick client to apply this setting.
- **Look and Feel**
  Specifies the look and feel for the OneClick client. The default setting is the native look and feel for the system running the client, such as Windows. If you choose to override the system default, OneClick attempts to use the same look and

feel setting independent of the system you are logged on to. If the system does not support the specified look and feel, OneClick uses the system default.

NOTE
Restart the OneClick client to apply this setting.

- **New Message Sound**
  Specifies whether there is a sound indicator when you receive a new message from OneClick administrators.
- **Scrollbar Increment**
  Specifies the amount that each scrollbar adjusts to when you click the scrollbar arrow.
- **Time Format**
  Specifies the time format in OneClick as either 12-hour or 24-hour.
- **Time Zone**
  Specifies whether to use Coordinated Universal Time (UTC) to display dates and time in OneClick. By default, OneClick uses the local system time zone.
- **Tool Tip Delay**
  Specifies the amount of time, in seconds, that your cursor remains over a button, field, or other component in the OneClick interface before a tooltip displays.

Explorer Tab Preferences

The following Explorer preference options are available in the Set Preferences dialog in the Explorer Tab section.

- **Expansion Limit**
  Displays a warning when you expand an Explorer node whose number of elements exceeds the limit specified.
- **Explorer Table**
  Specifies the following preferences for the Explorer tab:
  - **Columns**
    Specifies the Alarm category columns that appear in the Explorer.

    NOTE
    For more information, see Display Initial and Suppressed Alarms.
  - **Fonts**
    Specifies the font and type size that are used to display text in the Explorer.
  - **Sort**
    Specifies the default sorting method for the Explorer.
- **Initial View**
  Specifies how the Explorer tab appears in the OneClick Console each time you start the application.

Expand and Collapse the Explorer View

You can collapse the hierarchy in the Explorer tab with one click.

Follow these steps:

1. Click the Explorer tab.
2. Select the node that you want to collapse, right-click, and select Collapse All.
   Everything beneath the selected node is collapsed to the level of the selected node.
3. (Optional) Select a node that you want to expand, right-click, and select Expand All.
   Everything beneath the selected node is fully expanded.

Locater Tab Preferences

This section describes the Locater tab preferences available in the Set Preferences dialog. Depending on your access rights, you may not have access to all the following settings.

- **Prompt for Landscapes**
  Specifies whether to prompt you for the landscape you want to search when executing a search from the Locater tab.
  Default: Yes
  – **Results Table**
  – **Columns**
    a. Specifies the columns of information to display in the table.
  – **Font**
    a. Specifies the font family and type size used in the table.
  – **Sort**
    a. Specifies the default sorting methodology for the table.

Topology Tab Preferences

This section describes the Topology tab preferences that are available in the Set Preferences dialog. Depending on your access rights, you may not have access to all the following settings.

- **Annotation Font**Specifies the default font settings for topology annotation text. You can modify font, style, size, and background and foreground colors.
- **Grid Properties**
  Affect the topology view in Edit mode. Only administrators can place OneClick topologies into Edit mode.
- **Initial Zoom**
  Affects the default view of the Topology tab. Select from the following options:
  – The system default zoom percentage. The OneClick administrator sets the value.
  – A custom zoom percentage.
  – Fitting the topology into a visible window with the zoom level at or above a minimum setting.
- **Model By Type**
  Specifies the model types that are available from the Model by Type dialog. Model types are available when you manually create models by model type in the Topology tab.

  NOTE
  For more information, see Modeling and Managing Your IT Infrastructure .
- **Performance**Specifies whether to enable anti-aliasing when rendering pipes.
  Default: Yes
- **Show Off Page Reference Models**
  Specifies whether off-page reference models are displayed in the Topology tab.
- **Show Pipe Label**
  Specifies whether to show pipe (connection) labels in the Topology tab.
  Default: No
- **Telnet/SSH Connection Type**
  Specifies the connection type for telnet and ssh.
  Default: Connect to the device through OneClick web server and SpectroServer
- **Telnet/SSH Terminal Type**
  Sets the terminal type for telnet and ssh.
  Default: vt100
- **VLAN Duplicates**
  Specifies whether to show VLAN duplicate IDs even if the same device appears more than once in a Global Collection.

  NOTE
  If you change any of the Topology preferences, and ‘Make Changes Permanent’ is selected, the changes are in effect each time you use OneClick.

Table Preferences

You can change the way columns appear in OneClick by right-clicking a column heading. In the Table Preferences dialog, you can select the columns that you want to display. The available choices vary depending on the OneClick application you are using.

You can resize a column by clicking and dragging a column header to the left or right. You can also resize a column to fit the longest text string. Double-click the column header boundary to the right of the column. Click a column heading to sort a list by the attribute values in a particular column.

You can also set table preferences in each of the Set Preferences dialog categories that support tables.

View Row Details

You can view more information about the rows in OneClick Console lists from the Row Details dialog. The Row Details dialog displays the value for each available field. Values for fields that are not currently displayed in the list are also included.

Follow these steps:

1. Right-click the desired row item and select Row Details.
   The Row Details dialog opens.
2. (Optional) Click another row in the same list.
   The Row Details dialog displays values for the newly selected row.
3. Click Close when you have finished reviewing row details for items in this list.

Filter OneClick Lists

You can filter the items that are displayed in OneClick lists using the Filter text box. A Filter text box appears in many OneClick panels and tabs. The filtering feature lets you enter text to filter the data that appears in columns and lists. As you type in the Filter text box, the list displays or hides only the items that contain the text you entered. You can include or exclude items by selecting Show or Hide from the Filter drop-down list.

Select Landscapes

A DX NetOps Spectrum landscape is the network view of a specific DX NetOps Spectrum server. In a distributed DX NetOps Spectrum environment, multiple DX NetOps Spectrum servers are used to manage the network. Each server has its own view of the network. Depending on how your DX NetOps Spectrum environment is configured, you may have access to more than one DX NetOps Spectrum server.

As such, the ‘Select Landscapes to Search’ dialog can appear when you perform certain actions. This dialog asks you to select those landscapes on which you want to perform the actions. This dialog lists the included landscapes on the left, and the excluded landscapes on the right. If you have only a single landscape, that single landscape appears in the list of included landscapes.

OneClick Panels

By default all three panels appear in the OneClick interface, however, you can modify your view of the panels as needed.

- **View menu:** Each panel is listed in the View menu. If the panel has a checkmark next to it, it is viewable. If the panel does not have a checkmark next to it, it is not currently viewable in the OneClick Console interface.
- **Docking and Cloning:** Each OneClick panel can be docked, undocked, or cloned using the following buttons:

- 
  Undock: Click to undock a OneClick Console panel.
- 
  Dock: Click to dock a previously undocked OneClick Console panel.
- 
  Clone: Click to clone a OneClick Console panel.

Dock and Undock Panels

Undocking a panel opens it in its own window, at the same time removing it from the main OneClick Console view. Undocking panels can help you to make better use of your screen space.

You can dock an undocked panel by clicking Dock or by using the View menu. To display panels that you have closed, click the View menu and select the panel to display.

Clone Panels

Click Clone in either the Contents panel or the Component Detail panel to open a separate window containing another instance of the panel. Clicking Clone in the Contents panel while the Component Detail panel is visible opens a new window containing instances of both panels.

Use cloning to view more than one area of your network. The display of information of a cloned window is not affected when you navigate away from the original source to view other network assets.

Copy Text from the Component Detail Panel

You can copy text from the Component Detail panel.

Follow these steps:

1. Select the text in the Component Detail panel you want to copy by taking *one* of the following steps:
   - Place the cursor over the beginning of the range of text you want to copy, press and hold the left mouse and drag the cursor across the range of text.
   - Double-click a word that you want to select, or triple-click to select an entire line of text.

   The text range is highlighted.
2. Take *one* of the following steps with the text range highlighted:
   - Right-click and select Copy.
   - Press Ctrl+C.
3. Paste the text into the writable field in OneClick or in another document or email program.

Insert URLs in OneClick

You can insert URLs into writable fields in the Component Detail panel.

Follow these steps:

1. Find the field in the Component Detail panel where you want to add a URL.
   For example, select the Notes field from the General Information submenu and click set.
2. Enter the text and the URL to include in the note. For example:
   Issue is described at http://internal.info.xyz.com
3. Click Save.

NOTE
Including spaces or commas in a URL can cause some browsers to have problems locating the web resource. To include spaces or commas in a URL that you are including in the Component Detail panel, use the hexadecimal equivalent and proper encoding:

- For a comma, use **%2C**
- For a space, use **%20**

NOTE
The OneClick administrator can provide more information about URL formatting.

Monitoring Your Network with OneClick

This section contains the following topics:

Global Collections

Global Collections help organize operator views of network devices that span containers or landscapes. OneClick administrators create Global Collections, and operators monitor Global Collections by selecting them in the Explorer tab, and then viewing the Alarms, Events, and List tabs in the Contents panel.

A Global Collection can be empty for the following reasons:

- Collections are not configured.
- Your user account does not have access to existing collections.
- Dynamic collections do not currently contain any models.

OneClick administrators maintain Global Collections and grant or restrict access to them.

NOTE

When the **Global Collections, List tab** contents search exceeds a pre-defined number of models, a relevant dialog box stating: **"There are <number> models to be displayed. This may take time. Would you like to proceed?"** appears. If you select **No**, the **List tab** and other tabs for that **Global Collection** will remain empty. To initiate the display of its contents, switch back and forth to any other container item (with **List Tab** enabled) in **Explorer Tab** like another **Global Collection**, **Cluster Manager**, **Policy Manager** etc. The pre-defined threshold is configurable by modifying **ModelIdCountThreshold** parameter value in **<SPECROOT>\tomcat \webapps\spectrum\WEB-INF\web.xml** on **OneClick Server** and restarting **OneClick Server**.

NOTE
For more information, see the Modeling and Managing Your IT Infrastructure .

Network Searches

Searching your network with DX NetOps Spectrum is a fundamental network management task. As an operator, you can run predefined searches using the Locater tab to identify specific models or groups of models on your network. You can choose from several categories of search criteria when performing a search. For example, not only can you search for network assets, but you can also search for configuration items, such as schedules. You can determine which schedules are in effect and can determine the models to which a schedule is applied.

Some predefined searches that are available from the Locater tab include the following objects:

- All Devices
- Devices, By IP Address
- All Application Models
- All SNMPv3 Devices
- All Schedules

If you are operating in a Distributed SpectroSERVER (DSS) environment, some searches require you to select the landscapes to include in your search using the landscape selection dialog.

NOTE
For more information, see the OneClick Administration.

Search Your Network

You can run predefined searches of your network from the Locater tab.

Follow these steps:

1. Click the Locater tab in the Navigation panel.
2. Find the search that you want to run in the Locater tab and do *one* of the following actions:
   - Double-click the search.
   - Select the search and click the search icon to launch the selected search 

   If no further information is required, the search runs and the results are displayed in the Results tab. If more information is required, the Search dialog opens.
3. Take *one* of the following steps:
   - Type the value you want to search for.
   - Select the value that you want to search for from the drop-down list:

   The value that you want to search for is now listed in the Search dialog.
4. (Optional) Select the Ignore Case checkbox for a search that is not case-sensitive.
5. (Optional) Click the List button to search for multiple values for a single attribute.
6. Click OK.
   The search runs. The results are displayed in the Results tab of the Contents panel.

NOTE

When the search results exceed a pre-defined number of models, a relevant dialog box stating: **"There are <number> models to be displayed. This may take time. Would you like to proceed?"** appears. This pre-defined threshold is configurable by modifying **ModelIdCountThreshold** parameter value in **<SPECROOT>\tomcat\webapps\spectrum\WEB-INF\web.xml** on **OneClick Server** and restart OneClick Server.

Search for Multiple Values for a Single Attribute

You can specify a list of values to search on using the List button in the Search dialog after executing a search.

> **NOTE**
> The List button is not available for regular expression searches or for 'negative' searches, such as Does Not Contain, Does Not Start With, Does Not End With, Not Equal To.

Follow these steps:

1. Run the desired search for which you want to enter a list of values.
   The Search dialog opens.
2. Click List.
   The List of Values dialog opens.
3. Do *one* of the following actions:
   – Type the values that you want to search for in the List of Values dialog.
   – Click Import, select the file that contains the values you want to search for, and click Open.

   > **NOTE**
   > The values that you enter are logically OR'ed together. For example, if you enter "router 1, router 2, router 3", the search returns "router 1 OR router 2 OR router 3."

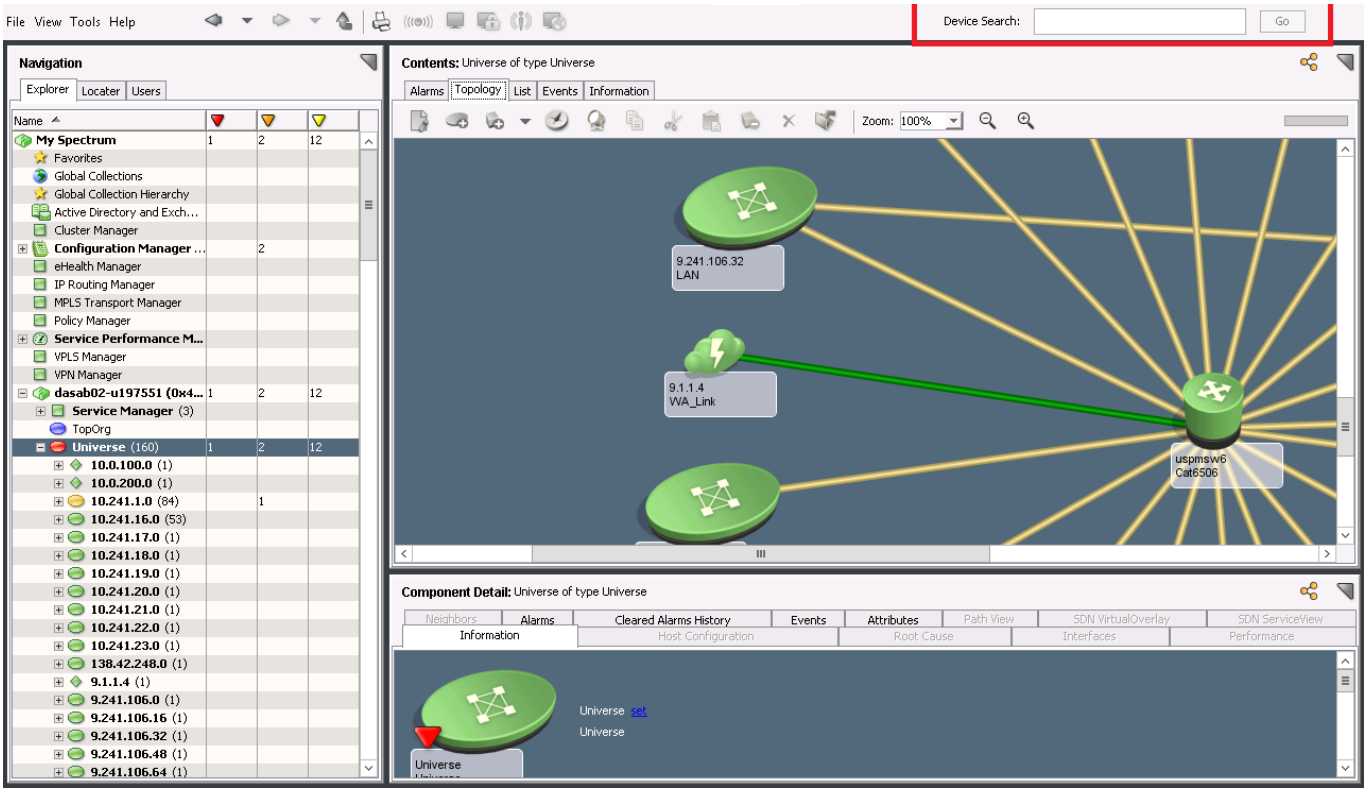
   The values that you are searching for are displayed in the List of Values dialog.
4. (Optional) Select the Ignore Case checkbox.
   The search is now case-insensitive.
5. Select one of the following characters from the Delimiter drop-down list, depending on which delimiter you are using to separate each value in the list:
   – <new line>
   – <space>
   – ,
   – ;
6. Click OK.
   The List of Values dialog closes.
7. Click OK
   The search runs and results are displayed in the Results tab in the Contents panel.

About OneClick Quick Device Search

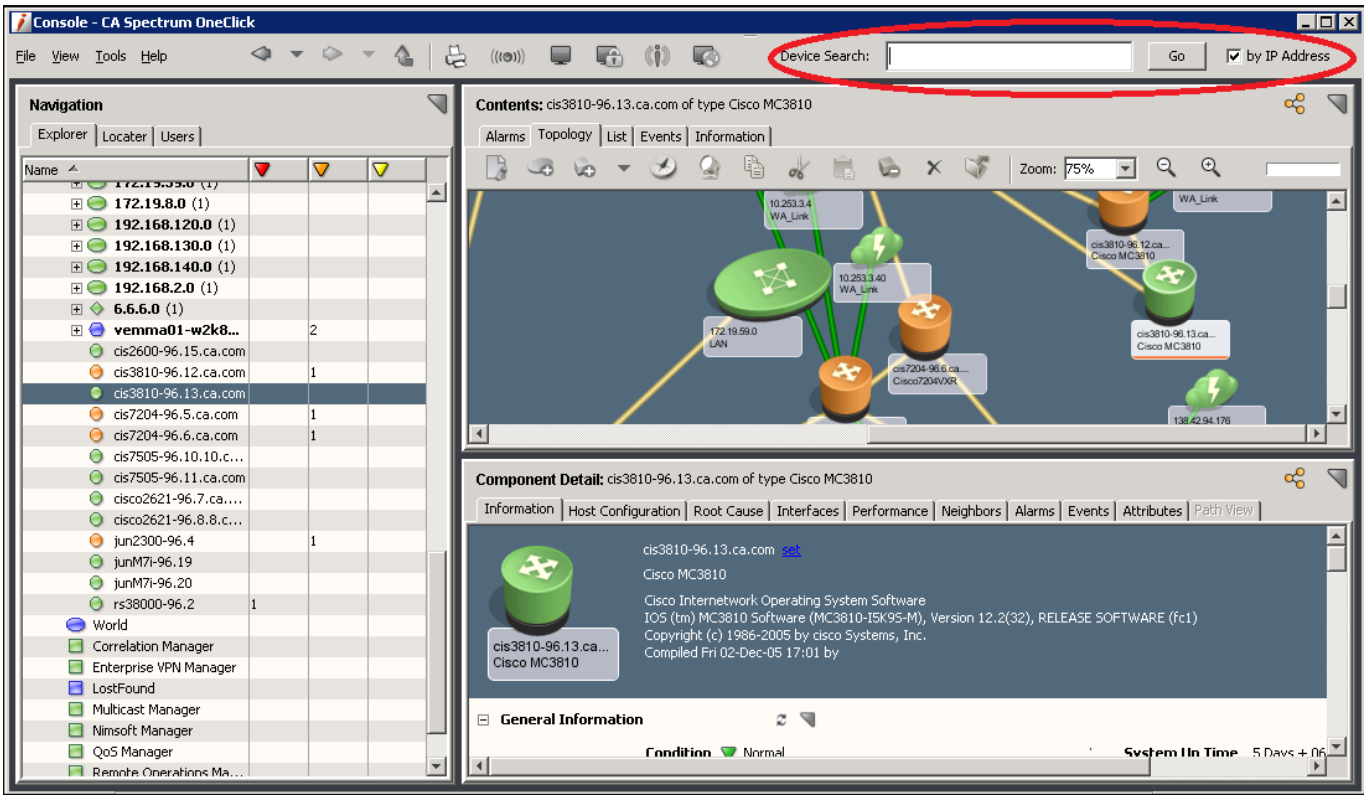
OneClick allows you to directly find a device model without going to the Locater Search. Enter the full IP address of the device in the device search bar of the OneClick console. You can also find a device by its redundant IP addresses, if that device has **Redundancy Preferred Addresses**. To find a device by its full name or a string, uncheck the "by IP Address" check box and enter the full name or a string.

From the 10.2.2 release, the Device Search functionality is enhanced to find a device by its full name, IP Address or a string.

The following image shows how the device search bar looks like (from 10.2.2 release):



The following image shows how the device search bar looks like (till 10.2.1 release):



Searching by a String

A search result that includes a list of all devices containing the matching string in their names is displayed. The search result includes device name, IP address, and the device landscape.
When you double-click the required device from the search result, the device is highlighted in the navigation panel.

Searching by Full Device Name or IP Address

The exact device is highlighted in the navigation panel. In a DSS environment when the device exists in multiple landscapes, you get the search result with the same information.

Select the device belonging to the required landscape.

In both cases, alarms and other information of the device are displayed in the Contents and Component Detail pane.

NOTE

Ensure to enter a valid IP address when you find by IP address. Only a valid IPv4 address enables the search.

Searching by a String, Full Device Name or IP Address (from 10.2.2 release)

From the 10.2.2 release, the Device Search functionality is enhanced to find a device by its full name, IP Address or a string. Enter the full name, IP Address or a string and click Go.

A search result that includes a list of all devices containing the matching string in their names is displayed. The search result includes device name, IP address, and the device landscape.
When you double-click the required device from the search result, the device is highlighted in the navigation panel.

In a DSS environment when the device exists in multiple landscapes, you get the search result with the same information. Select the device belonging to the required landscape.

In both cases, alarms and other information of the device are displayed in the Contents and Component Detail pane.

Manage Alarms

OneClick provides tools to identify and manage the alarms that are displayed in the Alarms tab. Some settings let administrators customize the alarms that are generated in OneClick. Other settings are available to let operators manage the alarms that they see in OneClick.

Alarm Filter Dialog

The Alarm Filter dialog lets you create alarm filters to determine how alarms appear in your OneClick Alarms tab. The dialog contains the following tabs:

- **Landscape**
  Defines the landscapes for which to display alarms.
- **Severity**
  Defines the alarm severities that are applied to this filter.
- **State**
  Specifies the states that you want to show. Options are as follows:
  - **Acknowledged State**
    Acknowledged, Not Acknowledged, Both.
  - **Clearable State**
    Clearable, Not Clearable, Both.
  - **Primary/Secondary State**
    Show Primary For Containers/All For Devices, Show Only Primary, Show All Alarms.

    NOTE
    Select Both under Acknowledged State or Clearable State to see both options for that state.
  - **Symptoms**
    Specifies whether to show alarms that are determined to be the cause of symptoms.
- **Network Address**
  Specifies a range of network addresses for which to show or hide alarms.
- **Assignment**
  Specifies which assigned troubleshooters can view alarms. Please note that all troubleshooters will show if you have access to the Alarm Filter.
- **Model Class**
  Specifies the model classes for which you do not want to display alarms.
- **Model Type**
  Specifies the model types for which you do not want to display alarms.
- **Alarm Type**
  Specifies the alarm types for which you do not want to display alarms.
- **Attribute**
  Attribute: Select an attribute of a device to filter.
  Comparison Type: Specifies how to compare the value of the attribute ID with the value in the Attribute Value field. Only the comparison types appropriate to the data type of the attribute are displayed.
  Ignore Case: Select the Ignore Case check box if you do not want the comparison to be case-sensitive. This selection is only enabled when it is appropriate for the data type of the attribute that you selected.
  Attribute Value: Enter or select the desired attribute value that you want to use in the comparison.
  The Show Advanced button in this tab lets you use complex attribute filtering.

The following buttons are available in the Alarm Filter dialog from every tab:

- **Clear Tabs**
  Clears all fields in all tabs, and clears any filters that are set in the tabs.
- **Clear All**
  Clears all fields in all tabs and in the Advanced filter section. If you click Clear All and click OK, all alarms appear because no filters are set.
- **Show Advanced**
  Opens the Advanced Filter panel.
- **Available Filters**
  Contains saved filters so that you can apply, edit, or delete them.
- **Add**
  Creates an alarm filter using the Enter Filter Name dialog. The new alarm filter displays in the Available Filters drop-down list.
- **Delete**
  Removes the selected filter from the Available Filters list.

Filter Alarms

You can determine how alarms appear in your OneClick view using alarm filters created in the Alarm Filters dialog.

To open the Alarm Filter dialog, do *one* of the following actions:

- In the Set Preferences dialog, select Alarm Filter from the Alarms Tab folder and click Set Alarm Filter.
- Click the Filter icon (in the Alarms toolbar) 

You can create alarm filters to customize which alarms OneClick displays in the Alarms tab. You can also create advanced alarm filters, as described in Advanced Alarm Filter.

Multiple Alarm Filters

You can create multiple alarm filters to screen for specific alarm conditions on specific devices, containers, or other models. You can use these filters to view different alarm conditions simultaneously in multiple Alarm views. Create multiple Alarm views by cloning the Component Detail or Contents panel, and displaying the Alarm tab in each panel. Select a different alarm filter that you have created to troubleshoot or watch for specific conditions in each Alarm view.

Create and Save Alarm Filters

You can create and save alarm filters so that you can retrieve and use them later.

Follow these steps:

1. Click the Alarms tab in the Contents panel.
2. Click the Filter icon (in the Alarms toolbar) 
   The Alarm Filter dialog opens.
3. Click Add.
   The Enter Filter Name dialog opens.
4. Enter the name of the filter you want to create and click OK.
5. Click the tab that you want to use to configure the filtering criteria. These tabs are described in Advanced Alarm Filter.
6. Click Apply.
   The filter settings are saved and the filter is applied to the Alarms tab view.
7. Click OK.

The alarm filter is now created and saved and the Alarm Filter dialog closes.

Use Advanced Alarm Filtering

The Advanced Filter provides more flexibility when compared to simple filtering because it lets you make multiple selections of the types of filters you want to apply. Simple filtering, on the other hand, simply groups all filter selections and applies them in a linear fashion, (for example Filter by Landscape *and* Secondary Alarms *and* Model Type). In simple filtering, all criteria must be met; in advanced filtering, any of the criteria you define can be met.

Advanced alarm filtering has two requirements:

- At least two sets of filter criteria.
- Alarms are filtered in an “either/or” fashion.

For example, with advanced filtering, you can display red (Critical) HubCat5000 model types or yellow (Minor) Pingable model types. In this case, neither red (Critical) Pingable model types nor yellow (Minor) HubCat5000 model types show up in the Alarms list. Simple alarm filtering does not make such a fine distinction. Instead, all Minor and Critical Pingables and all Minor and Critical HubCat5000 display with simple filtering.

This procedure continues the example and describes advanced filtering for critical HubCat5000 model types or minor pingable model types.

Follow these steps:

1. Click View, Preferences.
   The Set Preferences dialog opens.
2. Expand the Alarms Tab folder in the Name column and click Alarm Filter, Set Alarm Filter.
   The Alarm Filter dialog opens.
3. Click the Model Type tab and hide all model types except HubCat5000 models by doing the following actions:
   a. Click the double-right arrow button to move all model types from the Show list to the Hide list.
   b. Type **HubCat5000** in the Filter field on the right side of the dialog.

      NOTE
      Scroll down to see the Filter field.

   c. With the HubCat5000 model type selected, click the single-left arrow button.

   HubCat5000 is moved to the Show list.
4. Click the Severity tab and move Major and Minor alarms to the Hide list.
5. Click Show Advanced.
   The Alarm Filter dialog expands to display the Advanced Filter section.
6. Click Add in the Advanced Filter section.
   Your selections are placed into the Advanced Filter panel; the panel now displays the following filter:
   ```
   Severity (Hide Suppressed, Major, Minor, Initial, Maintenance) AND Model Type (Show HubCat5000)
   ```
7. Click Clear Tabs.
   The filters that you just set are cleared.

   NOTE
   The filter that you created still appears in the Advanced Filter panel, but now you are going to add to it.
8. Click the Model Type tab and move all model types to the Hide list except Pingable.
9. Click the Severity tab and move Major, Critical, Initial, and Suppressed alarms to the Hide list.
10. Click Add in the Advanced Filter panel to move your selections to the Advanced Filter panel.
    The panel now displays the following filter:
    ```
    Severity (Hide Suppressed, Major, Minor, Initial, Maintenance) AND Model Type (Show HubCat5000)
    OR Severity (Hide Critical, Suppressed, Major, and Initial) AND Model Type (Show Pingable)."
    ```
11. Click Add next to the Available Filters list.
    Your settings are saved and the Enter Filter Name dialog opens.

12. Enter a name for the filter you want to save.
13. Click OK.
    The Enter Filter Name dialog closes and the filter you saved appears in the Available Filters drop-down list.
14. Click OK.
    The Alarm Filter dialog closes.
15. Click OK.
    The Set Preferences dialog closes.

Refresh

Starting from the 10.2.3 release, a new option (Refresh) is available in the Alarms tab to refresh the alarm filters without logging out of the OneClick. When new alarm filters are created and applied to a user or group of users, the 'Refresh' button becomes active. You can click this button to update the Available Filters drop-down list with newly created and applied advanced filters.

System Cleared Alarms

System cleared alarms are alarms that are cleared automatically by the system without user acknowledgment. The result is a device that returns to a normal (green) condition. You can track these alarms as part of network monitoring.

Enable system cleared alarm tracking on a per-model basis. With tracking enabled, you can locate these system cleared alarms from the Locater tab by running the Devices > All Devices with System Cleared Alarms search. You can then acknowledge them as needed.

> **NOTE**
> This search only finds devices that had an alarm that was cleared by the system. An aged out alarm which is cleared displays the "System.Alarm_AgeOut" value in its corresponding "Cleared By" column under "Cleared Alarms History" tab. The corresponding cleared event also displays this value in its "Cleared By" column under Events tab.

Example: System Cleared Alarms

The following example describes two devices, each with tracking enabled.

Device A has one critical alarm. The system clears the critical alarm; the device returns to normal condition and is found by the Devices, All Devices with System Cleared Alarms search.

Device B has one critical alarm and one major alarm. The system clears the critical alarm, but not the major alarm. Because the condition is *not* normal, the search does not find the device.

Track System Cleared Alarms

You can track system cleared alarms as part of monitoring your network.

How to Enable System Cleared Alarm Tracking

Follow these steps:

1. Select the model for which you want to track system cleared alarms.
2. Right-click and select Track System Cleared Alarms.
   Any system cleared alarms that occur on this model are now tracked.

How to Locate System Cleared Alarms

Follow these steps:

1. Click the Locater tab in the Navigation panel.
2. Double-click the Devices, All Devices with System Cleared Alarms search.
   If additional input is not required, the search runs immediately; the search results appear in the Contents panel.

All system cleared alarms are displayed.

How to Acknowledge a System Cleared Alarm

Follow these steps:

1. Select the model with the system cleared alarm on it.
2. Right-click and select Acknowledge System Cleared Alarms.
   The alarm is acknowledged.
   The model no longer appears in the Devices, All Devices with System Cleared Alarms search.

How to Disable System Cleared Alarm Tracking

Follow these steps:

1. Select the model for which you want to disable system cleared alarm tracking.
2. Right-click and select Ignore System Cleared Alarms.
   Any system cleared alarms that occur on this model are no longer tracked.

Update Alarm Attributes

By default, DX NetOps Spectrum lets you update the following two alarm attributes from the Alarms tab:

- Alarm Status
- Trouble Ticket ID

Updating an attribute of an alarm lets you provide more information about the alarm to other users. For example, update the status of an alarm to let other operators know how the situation related to an alarm is being handled.

> **NOTE**
> You can also define your own custom alarm attributes as needed. For more information, see OneClick Customization .

Follow these steps:

1. Click the Alarms tab in the Contents panel.
2. Select the alarm that you want to update.
3. Click the update alarm attributes
   icon 
   The Update Alarm Attributes dialog opens.
4. Select the attribute that you want to update from the Attribute drop-down list.
5. Type the new value for the attribute in the Attribute Value field.
6. Click OK.
   The Update Alarm Attributes dialog closes. The alarm attribute is updated.

Snooze Alarms

You can snooze alarms for any period shorter than 24 hours. The Snooze feature is helpful, for example, if some alarms are not as critical as others. Snooze an alarm to postpone action on a less critical issue so that you can focus on the more serious alarms.

Follow these steps:

1. Click the Alarms tab in the Contents panel.
2. Select the alarms that you want to snooze.
3. Right-click the selected alarms, select Snooze, and select *one* of the following options:

- Snooze Selected Alarms.
- Snooze Selected Alarms for All (You must have the Administrator privileges to use this option)
- Snooze Alarms From Corresponding Sources.

The Snooze Alarms dialog opens.

4. Take the following steps:
   a. Complete the fields to indicate how long the alarms remain snoozed.
   b. (Optional) Select the Save Current Time as Default check box.
   c. Click OK.

   The alarms that you selected are no longer visible in the Alarms tab and reappear when the snooze time has expired.

Snooze Alarms for All Users

If you have the Administrator privileges, you can snooze alarms for all users.

Follow these steps:

1. Click the Alarms tab in the Contents panel.
2. Select the alarms that you want to snooze for all users.
3. Right-click the selected alarms, select **Snooze**, then select **Snooze Selected Alarms for All Users**.
4. Take the following steps:
   a. Complete the fields to indicate how long the alarms remain snoozed.
   b. (Optional) Select the Save Current Time as Default check box.
   c. Click OK.

   The alarms that you selected are no longer visible in the Alarms tab for all users and reappear when the snooze time has expired.

Unsnooze Alarms

You do not have to wait for the default snooze time to expire before you can view snoozed alarms again. You can unsnooze these alarms whenever you are ready to focus your attention on them.

Follow these steps:

1. Click the Alarms tab in the Contents panel.
2. Click the Unsnooze icon (in the Alarms toolbar) 

   All previously snoozed alarms are visible again in the Alarms tab.

Unsnooze Alarms for All Users

If you have the Administrator privileges, you can unsnooze alarms for all users.

Follow these steps:

1. Click the Alarm tab in the Contents panel.
2. Select the alarms that you want to snooze for all users.
3. Right-click in the Alarms view, select **Snooze**, then select **Unsnooze Alarms for All Users**.

NOTE

The option 'Unsnooze Alarms for All Users' is enabled only when the option 'Snooze Selected Alarms for All Users' is applied to any alarm.

Alarm Troubleshooters

You can assign individuals, named troubleshooters, the responsibility of investigating alarms and solving problems. Troubleshooters are assigned to alarms using the Alarms toolbar. When you assign a troubleshooter to an alarm, they automatically receive an email about the alarm. You can edit the email before it is sent.

Create Troubleshooters

First create troubleshooters before you can assign an alarm to them.

Follow these steps:

1. Click Tools, Utilities, Troubleshooters.
   The Troubleshooters dialog opens.
2. Click Create.
   The Create Troubleshooter dialog opens.
3. In the Create Troubleshooter dialog, do the following:
   a. Enter the name of the troubleshooter and email address.
   b. Select the landscapes to which you are assigning the troubleshooter.
   c. Click OK.
4. To add the troubleshooter to a new landscape, do the following:
5. 1. Select the troubleshooter from the Troubleshooters list.
   2. Click Landscapes.
      Landscapes for Troubleshooter dialog opens.
   3. Select the new landscape from ‘Does not exist in’ list and move it to the ‘Exist in’ list.
   4. Click OK.
      The troubleshooter is added to the selected landscape.
6. Click Close.
   The troubleshooter is now created and you can now assign this troubleshooter to an alarm.

Assign and Unassign Troubleshooters

You can use the Alarms toolbar to assign and unassign troubleshooters to the alarms that are displayed in the Alarms list. Troubleshooters must already exist in OneClick to assign them.

> **NOTE**
> The administrator must configure email services on the OneClick server to enable the sending of notification email messages to the assigned troubleshooter. For more information, see the OneClick Administration .

How to Assign a Troubleshooter and Send a Notification Email

Follow these steps:

1. Click the Alarms tab in the Contents panel.
2. Select the alarms to which to assign a troubleshooter.
3. Click the Assign Troubleshooter
   icon 
   The Select Troubleshooter dialog opens.
4. Select a troubleshooter from the list.
5. (Optional) Click Edit Mail to edit the message before sending it to the troubleshooter.
6. Click OK.
   The Select Troubleshooter dialog closes. An alarm notification email message is sent to the troubleshooter you selected.

How to Unassign a Troubleshooter

Follow these steps:

1. Click the Alarms tab in the Contents panel.
2. Select the alarms from which to unassign a troubleshooter.
3. Click the Unassign Troubleshooter icon  .
4. Confirm that you want to unassign the troubleshooter from the alarm.
   OneClick sends an email message to the troubleshooter with information about the change.

View Troubleshooter Assignments

You can display the Assignment column in the Alarms list to see the names of troubleshooters that are assigned to each alarm.

Follow these steps:

1. Click the Alarms tab in the Contents panel.
2. Right-click the column header in the Alarms list.
   The Table Preferences dialog opens.
3. Select Assignment in the Columns tab and click OK.
   The Assignment column now appears in the Alarms list, displaying the names of troubleshooters that are assigned to alarms.

Email Alarms

You can send email messages which contain alarm details to troubleshooters and nontroubleshooters as needed.

Follow these steps:

1. Click the Alarms tab in the Contents panel.
   The Alarms list opens.
2. Select the alarms that you want to send in an email.
3. Click the Email icon  .
   The Mail Selected Alarms dialog opens.
4. Do any of the following actions:
   - Type one or more new recipient email addresses in either the To field or the Cc field, separated by semi-colons. These new addresses are saved to the Email Address List in your General preferences.
   - Use an existing email address by doing the following action:
     a. Click To or Cc.
        The Select Email Address dialog opens.
     b. Select one or more email addresses from the list and click To or Cc.
        The email addresses are added to the To or Cc field in the Select Email Address dialog.
     c. Click OK.
        The Select Email Address dialog closes and the email addresses you selected appear in either the To field or the Cc field in the Mail Selected Alarms dialog.
5. Type a subject in the Subject field, or select a subject template from the drop-down list.
6. Select the message template to use from the Templates drop-down list, or edit the template.
7. Click Send.
   The message is sent.

Manage Events

View events for containers and modeled devices by selecting the container or device in the Explorer tab, and then clicking the Events tab in the Contents panel or the Component Detail panel. You can filter the Events tab using the Event Filter dialog.

Email Events

You can email events using the same steps that you use to email alarms. For details, see Email Alarms.

Event Filtering

You can filter the events displayed in the Events tab by entering text to filter on in the Filter text box. You can also create event filters to save and reuse.

Event Filter Dialog

The Event Filter dialog lets you set more conditions for filtering events. You can access this dialog from the Events tab by clicking

(Filter) in the toolbar, or by right-clicking an event in the Events list and selecting Filter from the menu.

You can filter your current view of the Events list as follows:

- Display Events for a Date and Time Range
- Display Events for a Range of Hours
- Display Events for Ports and Applications
- Exclude and Include Events in the Event Table by Type
- Advanced Event Filter

> **NOTE**
> When you change many of these settings in the Event Filter dialog, the change only applies to the current instance of the Events tab. Default values for the Date/Time and Show events for subcomponents options are applied each time that you select the Events tab for a model. These default values can be set using the Set Preferences dialog (View, Preferences). Default preference values for these and other options can be set globally for all users by the OneClick administrator using the Set Preferences dialog.

Display Events for a Date and Time Range

In the Date/Time section of the General tab in the Event Filter dialog, you can limit the events shown in the Events tab to a particular range by selecting the ‘Show events for a time range’ option. Enter a start date and time for the range. If you do not select an end date and time for the range, OneClick displays all the events starting for that date and time onward. Create an end date and time for the range by selecting the End check box, and entering a date and time.

When you select the ‘Show events for a time range’ option, the Events tab shows the time range for which events are currently being displayed just above the Events list.

Display Events for a Range of Hours

You can show only those events that fall within the range of specific hours.

Follow these steps:

1. Click the Events tab in the Contents panel.

2. Click

(Filter) in the Events tab toolbar.
The Event Filter dialog opens.
3. Select ‘Show events for the last <*X*> hours.’
4. Enter a number in the selection box to indicate how many hours of recent events you want to see in the Events list.
5. Click OK.
The Events list now displays only those events from the past number of hours you specified.

NOTE
When you select ‘Show events for the last <*X*> hours,’ the Events tab shows the time range for which events are currently being displayed just above the Events table.

Display Events for Ports and Applications

You can show events for device model subcomponents including port and application models by selecting the ‘Show events for subcomponents’ option. By default, this option is not selected.

Exclude and Include Events in the Event Table by Type

You can add and remove events that are in the Exclude Event Types List. Events listed in the Exclude Event Types list are not displayed in the Events table.

NOTE
The following procedures only impact the Events tab view in the context in which you perform the task. You cannot save event view settings to use later.

You can exclude event types directly from the Events table.

To exclude event types from being displayed in OneClick, right-click the event that you want to exclude in the Events table and select Exclude from the menu.

The event is removed from the Events table.

How to Exclude Event Types from Being Displayed in OneClick using the Event Filter Dialog

Follow these steps:

1. Click the Events tab in the Contents panel.
2. Click

(Filter) in the Events tab toolbar.
The Event Filter dialog opens.
3. Click the Event Type tab.
4. Do *one* of the following actions:
   – Enter the value for the event type directly in the text field.
   – Click Browse, select the desired event value from the list that appears, and click OK.

   The selected event value is entered in the text field.
5. Click Add.
The event is added to the ‘Excluded event types’ list.

How to Include Event Types in the Events Table

Follow these steps:

1. Click the Events tab in the Contents panel.
2. Click
   
   (Filter) in the Events tab toolbar.
   The Event Filter dialog opens.
3. Select the Event Type tab.
4. Select the event type that you want to display in the Events table from the ‘Excluded event types’ list.
5. Click Remove.
   The event type is removed from the ‘Excluded event types’ list and appear in the Events table when it is generated.

   NOTE
   OneClick administrators can also use the Set Preferences dialog to specify whether any event types are excluded from the Events table.

Create Advanced Event Filters

You can select or create an advanced event filter using the Advanced tab in the Event Filter dialog. The advanced filters that you create are stored and can be reused. To use an existing advanced filter, select one from the Available Filters drop-down list.

How to Create an Advanced Event Filter

Follow these steps:

1. Click the Events tab in the Contents panel.
2. Click
   
   (Filter) in the Events tab toolbar.
   The Event Filter dialog opens.
3. Click the Advanced tab.
4. Select the Attribute from the drop-down list.
5. Select the Comparison Type from the drop-down list.
6. Enter the Attribute Value in the text box.
7. If you selected either Event Type or Model Type Name as the Attribute and you do not know the attribute value, click Browse.
   Either the Select Event Type or Select Model Type dialog opens.
8. Select the desired Event Type or Model Type from the dialog, and click OK.
   The Attribute value appears in Attribute Value field.
9. Click Add.
   The Enter Filter Name dialog opens.
10. Enter a name for the filter, and click OK.
11. Take *one* of the following steps:
    – Click OK to apply the filter and close the Event Filter dialog.
    – Click Show Advanced to continue and create a complex filter by using And/Or relationships between multiple advanced filters.

How to Clear Advanced Event Filters

Follow these steps:

1. Click the Events tab in the Contents panel.

2. Click

(Filter) in the Events tab toolbar.
The Event Filter dialog opens.
3. Click the Advanced tab.
4. Click Clear.
5. Click OK.

How to Delete an Advanced Event Filter

Follow these steps:

1. Click the Events tab in the Contents panel.
2. Click

(Filter) in the Events tab toolbar.
The Event Filter dialog opens.
3. Click the Advanced tab.
4. Select the filter you want to delete from the Available Filters drop-down list.
5. Click

(Delete).
6. Click OK.

Interface Information

You can view information about the interfaces of a device model by selecting the model and selecting the Interfaces tab in the Component Detail panel. The Interfaces tab displays a list of the configured interfaces and subinterfaces for the selected device, along with the parameters defined in this section.

> **NOTE**
> The colors of icons in the Name, Condition, and Status fields have the same meaning as the colors that indicate device model status throughout OneClick.

Not all of the parameters that are listed here appear in the default Interfaces tab view. See Customize Columns for information about displaying hidden columns in tables.

- **Name**
  Specifies the name of the interface.
- **Condition**
  Specifies the contact status for the device, in addition to any alarm conditions in effect for the device model.
- **Status**
  Indicates whether the interface is operational or nonoperational. An interface may be nonoperational for various reasons including being administratively disabled. Some of the possible values include up, down, off, and dormant.
- **Type**
  Identifies the physical layer interface standard that the interface uses, such as Ethernet, SONET, and V.35.
- **Description**
  Describes whether the interface is physical or logical, and the interface ID, such as et.2.1
- **Device Connected**

Specifies the name and status (green for up or red for down) of the device that the current interface is connected to. The device name is a hyperlink that displays the Information tab for the connected device.

- **Port Connected**
  The name of the port on the device that the current port is connected to. The port name is a hyperlink that displays the Interfaces tab for the device that the current port is connected to.
- **QoS Policy**
  Specifies the QoS policy name that applies to this interface.
- **Index**
  Specifies the value of the index object in the standard RFC or proprietary MIB that uniquely identifies this interface within the device.
- **Board.Port**
  Identifies the board and port number on the device for the corresponding port. For example, if the port is port 4 on a module in the third slot device, the Board.Port value is 3.4.
- **MAC Address**
  Specifies the MAC address of the corresponding interface.
- **IP Address**
  Specifies the IP address of the corresponding interface.
- **Port Speed**
  Specifies the connection speed of the corresponding interface.
- **Duplex Status**
  Specifies the duplex state of the corresponding interface, either full, half, unknown, or N/A.
- **Trunk Membership**
  Identifies if an interface is a member of an 802.3ad trunk. Trunk Membership displays either the trunk ID that the interface is a member of, or a zero for no membership.
- **Network Link Type**
  Describes the type of network device the interface is connected to. Possible values are:
  - End Station Link
  - Internal Link
  - No Link
  - Router Link
  - Shared Access Link
  - Switch Link
  - Unknown Link
- **% Total Utilization**
  Utilization rate of the corresponding interface that is expressed as percentage of the total capacity of the interface. For interfaces that share bandwidth between inbound and outbound traffic (such as Ethernet interfaces in half duplex-mode) thresholds against % Total Utilization are helpful in monitoring the “load” on the interface.
- **% In Utilization**
  Utilization rate of the corresponding interface that is expressed as percentage of the total inbound capacity of the interface. For interfaces that provide dedicated bandwidth to inbound and outbound traffic separately (such as Ethernet interfaces in full-duplex mode) individual thresholds against % Inbound Utilization and % Outbound Utilization can be more helpful in monitoring the “load” on the interface. This type of threshold configuration proves useful when an interface is expected to experience predominantly outbound or inbound traffic (such as one on a web server or a load balancer).
- **% Out Utilization**
  Utilization rate of the corresponding interface that is expressed as percentage of the total outbound capacity of the interface.
- **IF Alias**
  Specifies the value of the MIB II object ifAlias for the corresponding interface.
- **IF Name**

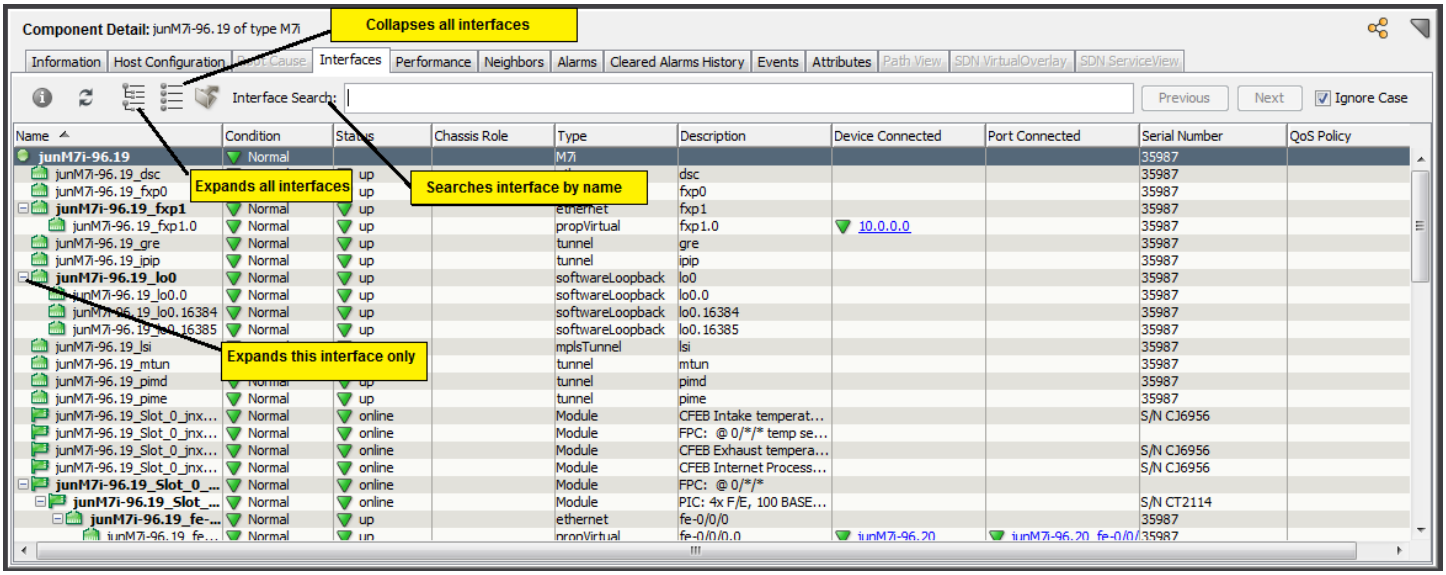
Specifies the value of the MIB II object ifName for the corresponding interface.

NOTE
For performance reasons, clicking (Refresh) in the Interfaces tab does not update external attributes (like ifAlias, for example). To update all values, instead select the specific rows in the list that you would like to update and click Refresh.

Subinterfaces

When a device supports virtual or subinterfaces, and subinterface modeling is enabled for the device model, DX NetOps Spectrum models the endpoints associated with multiplexed physical connections as subinterfaces. For example, Cisco IPSEC tunneling on a physical Ethernet interface, Permanent Virtual Circuits (PVCs) on a physical ATM interface, and Data Link Connection (DLC) circuits on a physical Frame Relay interface.

Some modeled interfaces also have subinterfaces available for viewing. The plus sign (+) next to modeled interfaces indicates that subinterfaces are available.



You can expand subinterfaces individually by clicking the plus sign (+) to expand the view of the interface. You can also click the Expands all interfaces icon

.

This icon expands the view of every subinterface belonging to this modeled interface.

Modeling of subinterfaces must be enabled on a model or device by a network administrator. For more information, see Modeling and Managing Your IT Infrastructure .

Interface Search

The Interface Search option allows you to search for an interface in the interface list by interface name.

The search can be a case-sensitive or case-insensitive search. The interface tree is automatically expanded when you click in the search box. You can use Previous or Next buttons to navigate the matching entries.

NOTE

The search time depends on the number of interfaces or subinterfaces. If the interfaces are more, then it may slow down the search.

Interface Component Detail Window

The interface Component Detail window provides access to tabs and subviews displaying information about the selected interface and its parent device. To display the interface Component Detail window, select the interface in the Interfaces tab and click the information icon  .

Interface Thresholds Subview

The Thresholds subview displays the current settings of pairs of parameters that are used to define interface alarm trigger and reset levels. Each parameter has the following threshold settings:

- A threshold level above which an alarm can be generated.
- A reset level that defines the value below which an existing threshold alarm condition is cleared.
- An allowed threshold violation duration which defines the duration, in seconds, that a threshold level can be violated before generating an alarm.

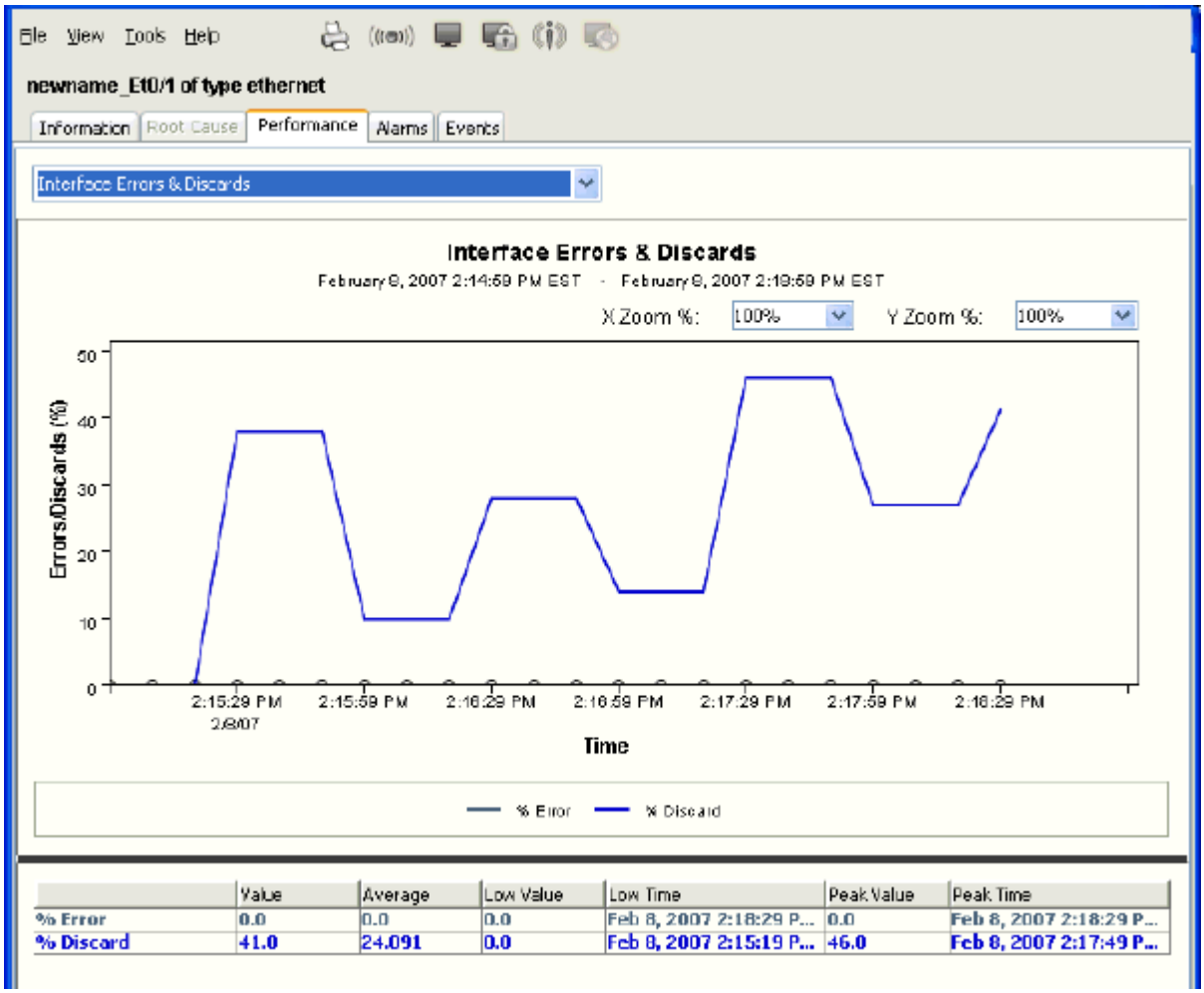
The following interface thresholds parameters appear in the Thresholds subview:

- % Total Utilization: Defines the level of port capacity used that triggers an alarm condition for a port.
- % Inbound Utilization: Defines the level of inbound port capacity used that triggers an alarm condition for a port.
- % Outbound Utilization: Defines the level of outbound port capacity used that triggers an alarm condition for a port.
- Total Packet Rate (packets/sec): Defines the number of packets per second that triggers an alarm condition for a port.
- % Errors: Defines the error rate on a port that triggers an alarm condition.
- % Discarded: Defines the percentage of discarded packets on a port that triggers an alarm condition.

> **NOTE**
> Network administrators can set the values for these parameters. For more information, see the Modeling and Managing Your IT Infrastructure .

Interface Performance View

The Performance view for the selected interface displays real-time graphs of interface utilization, throughput, and errors and discards. You can select from the performance views available using the drop-down list. You can set the zoom level for the X and Y scales for each graph, and each graph includes a legend explaining the data that appears in the graph. The following image shows an example of the interface Performance view:



Spotlighting Model Relationships

The spotlighting feature in OneClick lets you isolate and visualize the following model relationships within your network that are not readily visible from the Topology view:

- Router redundancy
- VPNs
- VLANs
- LSP Paths

The Topology view does not visually distinguish these model relationships, making it more difficult to picture them within the context of your network. With spotlighting, these model relationships are accentuated, showing you where they appear in the network topology.

For example, you can use the spotlighting feature to select an LSP Path to view in the Topology view. Viewing LSP Path information from this view can help you more easily understand which devices make up a Path in an MPLS environment. From this view, you can also see if any alarming devices are impacting a Path's performance.

> **NOTE**
> LSP Path spotlighting is not available if you do not have MPLS Manager installed. VPN spotlighting is not available if you do not have VPN Manager installed.

Spotlight Model Relationships in the Topology Tab

You can use the spotlighting feature in OneClick to see the VLANs, VPNs, LSP Paths, and router redundancy groups that are configured on your network. You can only spotlight these items if they have been enabled and configured on your network. For example, network administrators configure VLANs.

The following example describes how to spotlight VLANs, however, this procedure also applies to VPNs, LSP Paths, and router redundancy groups.

Follow these steps:

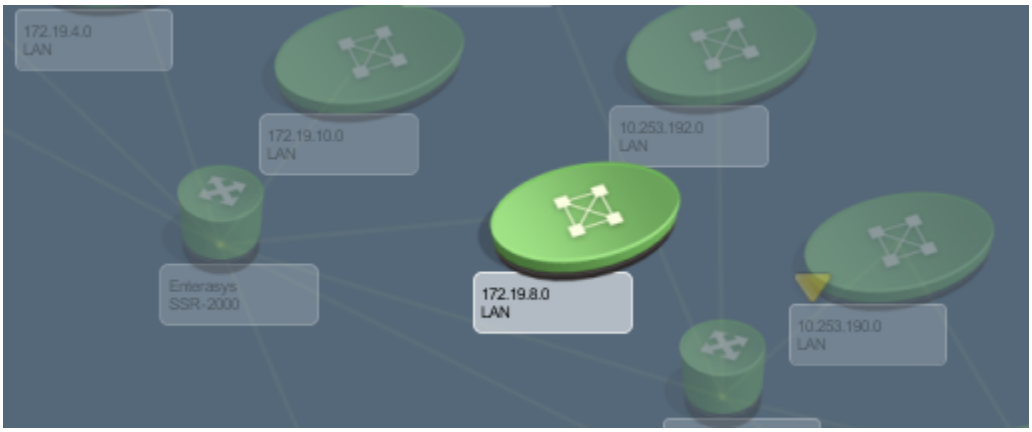
1. Select the desired Topology or container in the Navigation panel.
2. Click the Topology tab.
3. Click
   
   (Spotlight View), VLAN List in the Topology tab toolbar.
   The Topology tab view highlights the VLANs in the network and the VLAN List dialog opens, listing all the VLANs in the selected topology.
4. Select the VLAN that you want spotlight from the list and click the Information button.
   The VLAN List dialog expands to display OneClick tabs, which provide information about the selected VLAN.

> **NOTE**
>
> Starting from the 10.2.3 release, a new OneClick view is added in the Device Information tab to display (Port) interface name from JUNIPER_VLAN MIB. For Juniper EX Series Switch, VLAN information reads the data from JUNIPER-VLAN-MIB instead of dot1q bridge. The corresponding VLAN ID and VLAN Name mapping is displayed in the VLAN Spotlight View.

Highlight Modeled Devices

The Topology view for a Universe or other container can include many models, making it difficult to find a specific device or model. The OneClick highlighting feature can enable you to locate a model in the Topology view.

You can use the OneClick highlight mode to highlight modeled devices and containers in a topology view. In highlight mode, all devices except the highlighted device become translucent. The highlighted device stands out, as shown in the following image:



How to Find a Single Model using Highlight Mode

Follow these steps:

1. Select the container in the Explorer tab that you want to display in the Topology view.
2. Click the Topology tab in the Contents panel to view the topology.
3. Locate and select the model in the Explorer tab that you want to highlight.
4. Press the Shift key.
   The topology view and the device you selected appear.
5. (Optional) Navigate around the topology view using the horizontal and vertical scrollbars to locate the highlighted model if it does not appear in the viewable area of the topology.
6. Press the Shift key again to exit the highlight mode.
   You are returned to your original view of the Topology tab.

How to Find Multiple Models using Highlight Mode

Follow these steps:

1. Select the container in the Explorer tab that you want to display in the Topology view.
2. Click the Topology tab in the Contents panel to view the topology.
3. Press and hold the Shift key. The icons in the topology view become translucent.
4. Place the cursor over the model you want to highlight in the Explorer tab, without selecting it.
   The model is highlighted in the topology view.

   NOTE
   Use the scroll bars in the Topology tab to locate the highlighted device. The topology view does not adjust to show the highlighted device because it is not selected.
5. Move the cursor to the next model you want to highlight. As you move the cursor over any model in the Explorer tab, it becomes highlighted in the topology view.
   – If you place the cursor over a model that contains other models, such as a global collection, the devices in the global collection that are visible in the topology are highlighted.
   – If you place the cursor over a device that is part of a multicast group configured in OneClick, all of the devices in that multicast group that are visible in the topology are highlighted.
6. Release the Shift key to exit the highlight mode.
   You are returned to your original view of the Topology tab.

Connection Status Indicator

The OneClick Console provides visual indicators when the connection status to the SpectroSERVER changes. If the connection is lost, the borders around the OneClick Console turn red. If the connection has switched to a secondary SpectroSERVER, the borders turn yellow. The Information, Interfaces, and Performance tabs also display an orange border if the connection is lost to the selected device. A brown border indicates that the selected device model is in maintenance mode.

Check Connection Status

OneClick provides the status of connections to servers and services. The Connection Status dialog provides connection status and shows status logs for the servers and services used by the OneClick Console and OneClick add-on applications. The dialog provides the following information:

- Web services provided by the OneClick server
- Landscape service provided by the SpectroSERVER
- Events services provided by the SpectroSERVER

NOTE
The status of other services and server connections is available when you view the Connection Status dialog from other OneClick applications.

Follow this step:

- In the OneClick status bar, click the Connection <*status*> icon 
  The Connection Status dialog shows the status of web, landscape, and event services.

OneClick Messages

You can receive messages from OneClick administrators if your DX NetOps Spectrum environment is configured to support this option.

In the status bar, the Messages icon appears 

The Messages icon displays a + sign when you have unread messages. Retrieve your messages by clicking the Messages icon, which opens the Messages dialog. The Messages dialog lets you access messages that are sent to you by a OneClick administrator.

OneClick Schedules

You can schedule OneClick actions to occur at a given time with a recurrence if desired. Schedules include the following information:

- Start date
- Start and end times
- Total duration in hours
- Recurrence
- Description

When you apply a schedule to a modeled device, the event starts and ends at the specified start and end times in the time zone of the SpectroSERVER managing the device.

Access Schedules

You can locate existing schedules in OneClick by using the Schedules search function in the Locater tab. Once you have performed a search and one or more schedules appear in the Results list, you can access information about a schedule.

Schedule Information View

The schedule Information tab contains subviews which display schedule parameters. The schedule information that appears depends on the OneClick add-on applications that are installed as part of your DX NetOps Spectrum environment.

You can access the schedule Information view after running a search for schedules. Select a schedule from the Results list and select the Information tab in the Component Detail panel.

Schedule General Information Subview

The following parameters appear in the General Information subview in the Information tab for the selected schedule.

- **Creation Author**

Identifies the user who created the schedule. Schedules that ship with DX NetOps Spectrum show CA as the author.

- **Creation Time**
  Identifies when the schedule was created.
- **State**
  Identifies whether a schedule is active.

  NOTE
  A schedule with no duration, such as a schedule associated with Discovery Configurations, always appears as Inactive.
- **In Use**
  Identifies whether or not the schedule is applied to any devices, services, or other models.
- **Description**
  Optional text describing the schedule.

Items Scheduled for Maintenance Subview

The Items Scheduled for Maintenance subview displays all the devices that the schedule is applied to as a maintenance schedule. You can right-click any of the table headings to display a list of other columns available for viewing in this table.

Discoveries Planned with this Schedule Subview

The Discoveries Planned with this Schedule subview displays all the Discovery configurations to which this schedule is applied. See Modeling and Managing Your IT Infrastructure for more information about OneClick discovery and modeling configurations.

Create Schedules

You can create your own schedule by clicking **Schedule**, for example, in the **In Maintenance** field in the General Information subview.

Follow these steps:

1. Click **Schedule**.
   The Add/Remove Schedules dialog opens.
2. Click **Create**.
   The Create Schedule dialog opens.
3. Complete the fields as desired.
4. Click **OK**.
   The Create Schedule dialog closes and the schedule you created appears in the **Current Schedules** list.
5. Click **OK**.
   The Modifying Schedules dialog opens, indicating that the changes you made are being applied.

OneClick Schedules in a DSS Environment

In a Distributed SpectroSERVER (DSS) environment, there are likely to be SpectroSERVERs located in different time zones. Each SpectroSERVER interprets all schedules as local time. When you create schedules and apply them to devices that are managed in different landscapes, the scheduled item begins and end at the specified times local to each time zone. OneClick and DX NetOps Spectrum do not correlate schedules so that they start and end simultaneously across time zones. The following example illustrates how time zones and schedules work in a DSS environment.

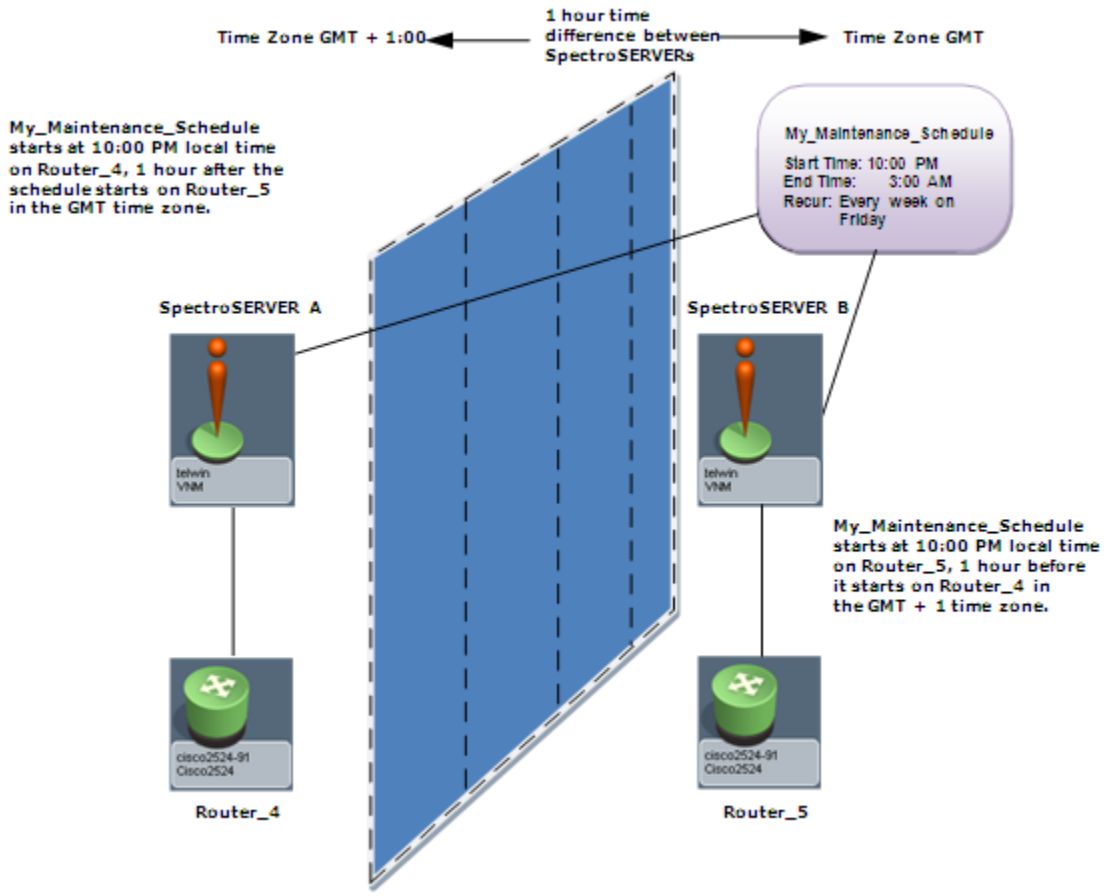
Apply a schedule to devices in different time zones

A schedule named My_Maintenance_Schedule specifies putting a device into maintenance mode starting at 10:00 PM and ending at 3:00 AM. My_Maintenance_Schedule is applied to Router_5 in the GMT time zone, and to Router_4 in the GMT+1 hour time zone.

Applying My_Maintenance_Schedule to these two devices results in the following situations:

- Router_5 enters maintenance mode at 10:00 PM GMT and exits maintenance mode at 3:00 AM GMT.
- Router_4 enters maintenance mode at 10:00 PM GMT+1 (11:00 PM GMT) and exits maintenance mode at 3:00 AM GMT+1 (4:00 AM GMT).

The following diagram illustrates this example:



Recurring Schedules

When creating a schedule, you can specify a recurrence. Consider the following things when you use recurring schedules:

- The Start Date controls when the schedule goes into effect. Consider the following points when specifying the Start Date:

  - The Start Date defaults to today's date. You can also specify today's date by using the Today button from the drop-down calendar. For recurring definitions, the default value of today's date is ignored unless you explicitly specify today's date by using the Today button.
  - If the Start Date field remains unchanged, the schedule goes into effect immediately; any scheduled action occurs on the next instance of a date and time that complies with the schedule.
  - If the Start Date field is modified in any way, regardless of if it is today's date or a future date, it must fall on a day that complies with the recurrence definition. For example, if the recurrence definition specifies that an action is performed every Saturday, the Start Date must be a Saturday.
- The recurrence definition for Daily, Weekly, and Monthly supports a year's time frame. For example, when specifying an action to occur every *x* weeks, you cannot exceed 51, which is one unit less than a year. To use anything greater, use Yearly.

  NOTE
  Although you can manually enter a value beyond a year's time frame, the recurrence definition defaults to the last valid value entered.
- If a schedule with no duration is set, as is the case for a schedule associated with Discovery Configurations, viewing it in a Locater search result list always shows its State as Inactive. Check the In Use column to determine whether it is associated with any tasks. To see its associated tasks, select the schedule and view its Information tab in the Component Detail panel.
- After a schedule is created, you cannot modify its definition. To change the time or duration for which a task is scheduled, a new schedule must be created and the associated task must be altered to use the new schedule.

Maintenance and Hibernation Mode for Devices

Maintenance and hibernation modes in OneClick let you suspend management traffic to a modeled device and its components. When a modeled device is in maintenance or hibernation mode, the SpectroSERVER continues to receive and process all SNMP traps for that device. However, it does not generate events or alarms for the device or its components.

Maintenance mode differs from hibernation mode by requiring you to disable the maintenance mode option before the device can resume normal management traffic. By contrast, hibernation mode automatically restarts normal management traffic as soon as the SpectroSERVER detects successful communication with the device after a set of successful polls.

Hibernation mode takes precedence over maintenance mode on a device model. However, if the device model has interface models in maintenance mode, those models remain in maintenance mode after the hibernation device model resumes normal management communication.

By default, placing a device model into maintenance or hibernation mode also places its interface models and application models into maintenance or hibernation mode. When a modeled device is in maintenance or hibernation mode, its topology icon displays a brown condition color. Brown alarms are shown for all device models in maintenance or hibernation mode, but they are not shown on the application and interface models that have inherited the mode from the device model.

Place Devices in Maintenance Mode

The device maintenance mode setting is in the General Information subview of the Information view. Find the Information view in the Contents panel or in the Component Detail panel.

Follow these steps:

1. Select the device in the Navigation panel, in a Topology view, or in a List view.
2. Click the Information tab.

3. Expand the General Information subview.
4. Click Set next to the In Maintenance setting and select Yes from the drop-down list.
   The device is now in maintenance mode, and its icon changes to brown.

Schedule Maintenance Mode

Schedule Maintenance Mode

You can schedule when a device enters maintenance mode by applying a maintenance schedule. You can apply an existing schedule or can create a maintenance mode schedule.

> **NOTE**
> See OneClick Schedules for information about how OneClick and DX NetOps Spectrum apply schedules across time zones in DSS environments.

Follow these steps:

1. Select the device for which you want to set up a maintenance mode schedule.
2. Click the Information tab.
3. Expand the General Information subview if necessary, locate 'In Maintenance', and click Schedule.
   The Add/Remove Schedules dialog opens. Any maintenance schedules applied to the device appear in the Current Schedules column.

   > **NOTE**
   > You can also open the Add/Remove Schedules dialog by clicking Tools, Utilities, Schedule Maintenance.
4. Do *one* of the following:
   - **To apply an existing schedule to the device**, select the schedule from the Available Schedules column, and click the left arrow button to move it to the Current Schedules column.

     > **NOTE**
     > A device can have more than one schedule that is applied to it.
   - **To remove an existing schedule from the device**, select the schedule from the Current Schedules column and click the right arrow button to move it to the Available Schedules column.
   - **To create a new schedule**, click Create; the Create Schedule dialog opens. Configure a schedule by selecting a Start Date, a Start Time, and either an End Time or Duration. Select the Recurrence factor.

     > **NOTE**
     > You can create a one-time maintenance mode window by leaving the Recurrence set to None. Enter a Description that adequately identifies the schedule you are creating.
5. Click OK.
   The new schedule appears in the Available Schedules column in the Add/Remove Schedules dialog.
6. Click OK.
   The maintenance mode scheduling changes are applied to the device.

Maintenance Mode Schedule Synchronization

From 10.2.3, the Maintenance Mode schedules on devices from DX NetOps Spectrum will be synchronized to CA UIM and from CA UIM to DX NetOps Spectrum, using the integration via spectrumgtw probe v8.65.

> **WARNING**
>
> Deploying and configuring the maintenance mode probe v8.53 is a prerequisite before configuring the spectrumgtw probe for Maintenance Mode Schedule synchronization. For more information refer Maintenance Mode probe Release Notes.

After you create a schedule on a device on either DX NetOps Spectrum and CA UIM), the schedules will get synchronized (bidirectional) after the next inventory sync that spectrumgtw performs.

The Maintenance Mode schedules will get synchronized from DX NetOps Spectrum to CA UIM only for the devices that are part of the Global Collection in DX NetOps Spectrum. However, for devices synced from CA UIM to DX NetOps Spectrum, maintenance schedules will be synchronized irrespective of whether the devices are placed in the Global Collection or not. The sync interval is 30 minutes. Any updates, creation or deletion of schedules are synchronized either if the spectrumgtw probe is restarted or after the sync interval.

NOTE

- DX NetOps Spectrum will synchronize or create CA UIM schedules, only if a minimum of one associated model exists in the UIM schedule.
- Maintenance schedules without any devices will not be synchronized.
- CA UIM does not support Yearly schedules, consequently yearly maintenance schedules existing in DX NetOps Spectrum will not be synchronized.
- If you manually place a device in maintenance in DX NetOps Spectrum, the device state is not synchronized to CA UIM. The synchronizations only applies to a device which enters maintenance mode by applying a schedule.

Synchronization for different time zones

If DX NetOps Spectrum is in Daylight Saving time zone and CA UIM is in non-Daylight Saving time zone or vice versa then the Maintenance Mode Schedule Synchronization is not supported.

The following table explains the supported/not-supported scenarios:

DX NetOps Spectrum (SpectroSERVER)	CA UIM	Support
Daylight Saving	Daylight Saving	Supported
Non-Daylight Saving	Non-Daylight Saving	Supported
Daylight Saving	Non-Daylight Saving	Not Supported
Non-Daylight Saving	Daylight Saving	Not Supported

Determine Whether Devices are Scheduled for Maintenance

You can determine if a device is scheduled for maintenance from the List tab or from the Information tab of a device.

How to Determine If a Specific Device is Scheduled for Maintenance

Follow these steps:

1. Select the device from either the List tab or the Topology tab.
2. Click the Information tab in the Component Detail panel.
3. View the In Maintenance section in the General Information subview.
   The Assigned Maintenance Schedules list displays the maintenance schedules that are assigned to this device.

How to Determine If Any Devices are Scheduled for Maintenance

Follow these steps:

1. Click the List tab.
2. Review the information in the Assigned Maintenance Schedules column.

NOTE
If you do not see the Assigned Maintenance Schedules column**,** add it to complete this procedure.The Assigned Maintenance Schedules column displays the maintenance schedule that is assigned to each device. If a device has more than one schedule that is assigned to it, a 'view' link is displayed.

3. (Optional) If a device has multiple schedules that are assigned to it, do the following actions:
   a. Click the 'view' link.
      The Assigned Maintenance Schedules dialog opens.
   b. Review the schedules that are assigned to this device.
   c. Click Close to close the Assigned Maintenance Schedules dialog.

Suppress Events and Alarms for Devices in Maintenance or Hibernation

When a model is in maintenance or hibernation mode, no events are processed for that model. That includes events that would typically clear an alarm on the model, and events that would create an alarm. For example: If a link_down event generated an alarm on a device model before the model being placed in maintenance mode and a link_up event occurred while the device model is in maintenance mode, the SpectroSERVER does not clear the alarm because the link_up event is not processed.

In this example, the SpectroSERVER would not resume normal management traffic to the maintenance modeled device until you manually disable the maintenance mode option for this device in the Component Detail panel.

If, in this example, the modeled device was placed in hibernation mode instead of maintenance mode, the SpectroSERVER would have to make a set of successful communication attempts to the device before it could resume normal management traffic with the device.

NOTE

The poll result of a device in maintenance, if pinged manually only has a 'successful' or a 'unsuccessful' message with no description. Generally users execute a poll before resetting the device to 'normal'. With the latest release of 10.3.1, this scenario is addressed with an enhancement that displays the actual status of a device in poll results to help users with descriptive result status (example that is shown in **Fig.1 and Fig.1.1**).

Fig.1 Example of a device which is in maintenance and is up with the descriptive polling result.

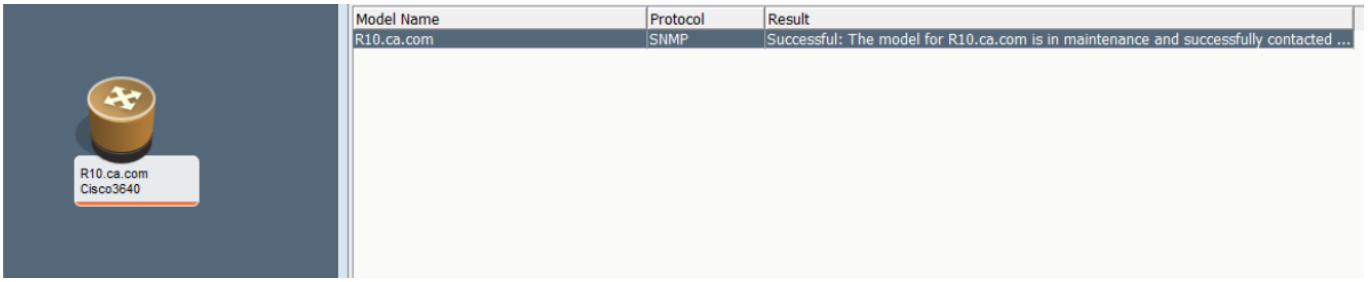
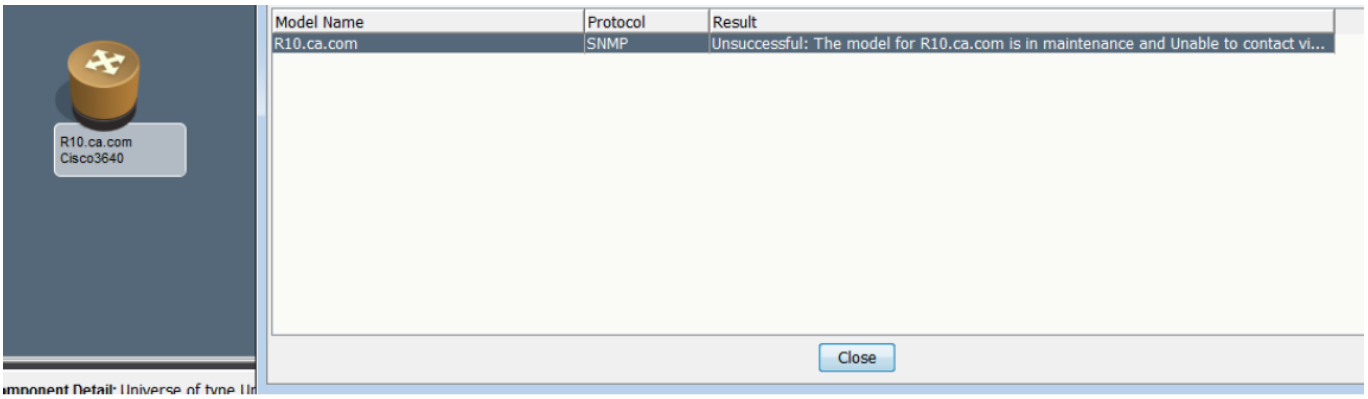


Fig 1.1 Example of a device which is in maintenance and is down with the descriptive polling result.



Secondary Alarms and Devices in Maintenance Mode

You can configure DX NetOps Spectrum to show or hide secondary alarms when a device is in maintenance mode. The ‘Show Secondary Alarms When Device is in Maintenance’ parameter in the Set Preferences dialog controls this behavior. If this parameter is enabled, secondary alarms are shown when a device is in maintenance mode.

NOTE
You cannot show or hide secondary alarms for devices in hibernation mode.

The 'Show Secondary Alarms' option is disabled by default. Secondary alarms are hidden when a device is in maintenance mode and are shown later when the device is taken out of maintenance mode.

NOTE

The ‘Show Secondary Alarms' setting only applies when the primary and secondary alarms are enabled in the Alarm Filter settings in the Set Preferences dialog.

Show Brown Alarms for Interfaces and Applications

You can display brown alarms for interfaces and applications that have inherited maintenance or hibernation mode from the device model by using the DX NetOps Spectrum Command Line Interface (CLI).

- To generate brown alarms on interface models that have inherited maintenance mode, set the device model attribute 0x00012a7a (rollMMAlarmToIF) to TRUE.
- To generate brown alarms on application models that have inherited maintenance mode, set the device model attribute 0x00012a7b (rollMMAlarmToApp) to TRUE.

NOTE
For more information, see Command Line Interface .

Place Devices in Hibernation Mode

In the Component Detail panel for a modeled device, the In Hibernation attribute indicates whether a device is in hibernation mode. When a device model is in hibernation mode, management traffic to the device and its components is suspended until a predefined number of communication attempts have succeeded. When the device can be contacted, the device model automatically resumes normal management communication.

Change the Number of Communication Attempts

The default number of successful communication attempts is 3 with the polling interval time (default 60 seconds) between each attempt. You can change the default number of successful communication attempts using the DX NetOps Spectrum

Command Line Interface (CLI) or the OneClick Attribute Editor. For more information about the DX NetOps Spectrum CLI, see Command Line Interface .

Either method requires changing the GlobalConfig mtype 0x00010470; the attribute HibernationCommSuccessTries 0x12acb is initially set to the default value of 3.

> **NOTE**
> You can also change this value using the Attribute Editor. For more information, see Modeling and Managing Your IT Infrastructure .

Hibernate After Maintenance

You can specify whether a device goes into hibernation mode when it comes out of scheduled maintenance. Maintenance Schedules include an option to automatically hibernate.

Follow these steps:

1. Select the device that you want to put into hibernation after maintenance.
2. Set up a maintenance mode schedule for the device as described in Schedule Maintenance Mode.
3. Click the Information tab in either the Contents or Component Detail panel and expand the General Information subview if necessary.
4. Locate the Hibernate After Maintenance setting, click set, and select Yes from the drop-down list.
   The device now automatically hibernates after a scheduled maintenance window closes. In hibernation, the device is polled 3 times and, if successful, the device comes out of maintenance.

Place Interface Models in Maintenance or Hibernation Mode

In the Component Detail panel of a device interface, enable the In Maintenance option to place the interface model into maintenance mode. This action suspends the management of the interface. However, DX NetOps Spectrum still performs regular management on the device and on its other interface.

> **NOTE**
> Unlike the maintenance mode, hibernation only applies to devices; you cannot place the interfaces alone into hibernation mode.

In the maintenance or hibernation mode, the following conditions apply to the interface model:

- Brown alarms are shown for interfaces in hibernation mode.
- Alarms are not created for the port.
- Events are logged for the port.
- No polling, logging, or other device communication is performed for the port model until the interface resumes normal management.
- Link Down traps that are sent are ignored, and no alarms are generated.
- If the Live Pipes option is enabled for a connection and one of its endpoints is in hibernation mode, the color of the pipe in the topology view turns brown. Status polling for that port is discontinued.

If a connection is modeled with a WA_Link model connection to two ports, and one of those ports is in hibernation mode (or maintenance mode), an alarm is created on the WA_Link and WA_Segment models. The WA_Link icon in the OneClick topology views turns brown. If Live Pipes are enabled on this link, the pipe remains green as long one port is up. If the second port is down or unreachable, the pipe condition color turns gray.

If DX NetOps Spectrum loses contact with a device model that is connected to a port in hibernation or maintenance mode, the ‘Device Has Stopped Responding to Polls’ alarm is suppressed for that device and for all adjacent devices. If device_contact_lost alarms are suppressed because of their position relative to a port in hibernation (or maintenance)

mode, the hibernation or maintenance mode alarm reflects these lost devices in its Impact and Severity attributes. View these lost devices in the **Impact** tab of the **Alarm Details** panel for that maintenance alarm.

Maintenance and Hibernation for WLC and AP interface models

In the maintenance or hibernation mode, the following conditions apply to the WLCs and AP interface models:

1. If WLC is in Maintenance/Hibernation mode, maintenance alarm is generated on WLC and all other alarms on WLC device are suppressed. The Access Points connected to the WLC are put into maintenance mode.
2. If AP and its Parent WLC are kept in Maintenance/ Hibernation then both will be in maintenance and all other alarms (except maintenance alarm) on both devices will be suppressed.

NOTE

- If AP goes into Maintenance/ Hibernation, only maintenance alarms are generated on the AP and all other alarms are suppressed.
- If AP is in maintenance and its parent WLC goes into maintenance at a later point, even then, all connected APs will go in maintenance. Once WLC is up, connected APs will be reverted to the previously managed state.

Place Wide Area Link Models in Maintenance or Hibernation Mode

A wide area link model represents a wide area connection between two router interfaces and includes:

- A WA_Link model that appears in the topology view.
- A WA_Segment model that exists within the WA_Link model and connects the two router interfaces together.

To place a wide area link into maintenance or hibernation mode, you have to modify settings for both the WA_Link and WA_Segment models.

In the Component Detail panels of both the WA_Link and the WA_Segment models, set the In Maintenance or In Hibernation setting to Yes to place the wide area link model into maintenance mode or hibernation mode. Management of the wide area link is suspended while regular management of the connected router interfaces continues.

When in maintenance or hibernation mode, both the WA_Link and WA_Segment models have a brown condition. The two connected router interfaces remain managed, and events and alarms are still generated on them.

NOTE
You can also put the router interfaces into maintenance mode, which allows full customizable control over how events and alarms are generated for wide area links. See Place Interface Models in Maintenance or Hibernation Mode.

To take a wide area link model out of maintenance or hibernation mode, modify settings for both the WA_Link and WA_Segment models accordingly.

Place ESX Host in Maintenance Mode or Hibernation

An *ESX host* is a physical computer that uses ESX Server virtualization software to run virtual machines. Hosts provide the CPU, memory, storage resources, and network connectivity for the virtual machines.This article describes how you can place an EXS host in maintenance and hibernation modes. Hibernation mode automatically restarts normal management traffic as soon as the SpectroSERVER detects successful communication with the device after a set of successful polls. Whereas, the Maintenance Mode requires you to disable the maintenance mode option before the device can resume normal management traffic.

To place an ESX Host in maintenance mode, enable the In Maintenance option in the Information tab of the Component Detail panel. This action suspends management of the ESX Host. However, DX NetOps Spectrum still performs regular management on the ESX Host and on its virtual machines.

In the maintenance mode, following conditions apply to the ESX Host:

WARNING

From 10.2.1, you can choose to place only the ESX Host server in maintenance mode, and not the VMs under the ESX Host. See Maintenance Mode Correlation for more information.

- By default, all the VMs under the ESX Host are automatically placed in maintenance mode. If the maintenance mode is turned off on the ESX Host, all the VMs are removed from maintenance mode.

  NOTE

  If a VM is already in maintenance mode before the ESX Host placed in maintenance mode; even after the ESX Host is out of maintenance mode, the state of VM remains maintenance mode only.
- Traps are not processed.
- Events are generated.
- Maintenance Alarms of VMs are correlated to ESX Host Maintenance Alarm.

Schedule Maintenance Mode for ESX Host

Schedule the maintenance mode of an ESX Host by applying a maintenance schedule or by creating schedule. You can apply multiple schedules to an ESX Host. For more information on applying schedule and creating a schedule see the Schedule Maintenance Mode section.

NOTE

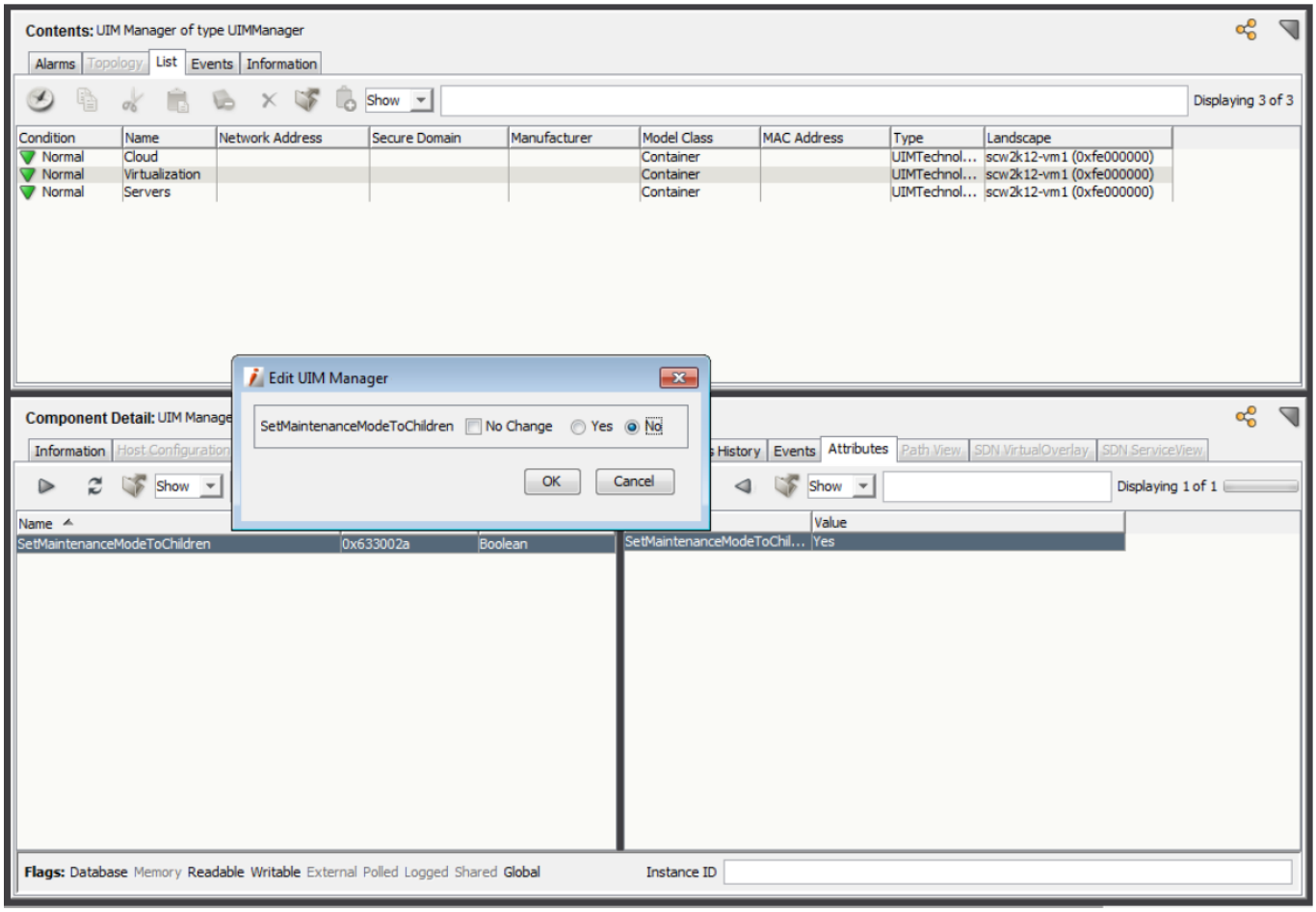
See OneClick Schedules for information about how OneClick and DX NetOps Spectrum apply schedules across time zones in Distributed SpectroSERVER environments.

Maintenance Mode Correlation

The default, Maintenance Mode Correlation behavior is such that, when an ESX Host is placed in maintenance mode, all the VMs under the ESX Host are automatically placed in maintenance mode. The maintenance alarms on VMs are correlated to the ESX Host maintenance alarm. From 10.2.1, you can modify the default behavior, (where you can edit an attribute "SetMaintenanceModeToChildren /0x633002a") enabling you to choose only the ESX Host Server to be placed in maintenance, while ensuring that the the maintenance mode correlation does not apply to the VMs under the ESX Host.

To enable maintenance mode only for the ESX Host and not for the VMs, follow these steps:

1. In the DX NetOps Spectrum OneClick console, Explorer Tab hierarchy, navigate to UIM Manager, select the List tab and then the Attributes tab in the Components Detail panel.
2. In the Component Details panel, Search bar, enter one of the following:
   - Attribute Name: **SetMaintenanceModeToChildren**
   - Attribute ID: **0x633002a**
     By default, the value of this attribute is set to Yes.



3. Double-Click the Attribute result on the right pane.
   The Edit UIM Manager dialog appears.
4. Clear the No Change check-box and select No in the dialog.
5. Click OK.
   The Attribute Edit Results dialog appears with a confirmation of the change in attribute value, along with landscape details.

   NOTE

   Click Undo on the Attribute Edits Results dialog to revert the Maintenance Mode correlation back to default.

Once you have successfully edited the attribute; the maintenance mode correlation no longer applies to the VMs under the ESX Host. Now, only the ESX Host server will be placed in maintenance while the VMs under the same, will no longer be placed under maintenance. The VMs will remain in active state and will be monitored as usual for alarms and events.

Place ESX Host in Hibernation Mode

To place an ESX Host in hibernation mode, enable the In Hibernation option in the Information tab of the Component Detail panel. This action suspends management of the ESX Host.

When the ESX Host is placed in Hibernation mode, all the VMs under the ESX Host are automatically placed in maintenance mode. The maintenance alarms on VMs are correlated to ESX Host hibernation alarm.

In the Hibernation mode, following conditions apply to the ESX Host:

- Brown alarms are shown for interfaces in hibernation mode.
- No polling, logging, or other device communication is performed until normal management is resumed.
- Hibernation Alarm is shown on ESX Host and Maintenance Alarm is shown on VM.
- Maintenance Alarms on all the VMs are correlated to single ESX Host Hibernation Alarm.

If an ESX Host is placed in Hibernation, which is already in maintenance, then Spectrum shows both Maintenance and Hibernation alarms on the ESX Host.

NOTE

The Hibernation Alarms of VMs under an ESX are not correlated to ESX Host Maintenance Mode or Hibernation Alarm.

Hibernate After Maintenance

To place an ESX Host in hibernation mode after scheduled maintenance, refer to the steps explained in the Hibernate After Maintenance section.

Migrating a VM from an ESX Host to Another ESX Host

The following section describes scenarios of VM migration from a source ESX to a destination ESX.

Scenario 1	Result
(A) • ESX1 (source) is in Maintenance Mode. • VM is in inherited Maintenance. (the 'inheritedMaintenance' attribute is 'YES') • ESX2 (destination) is in Maintenance Mode.	The migrated VM under ESX2 is in Inherited Maintenance Mode.
(B) • ESX1 (source) is in Maintenance Mode. • VM is in inherited Maintenance. (the 'inheritedMaintenance' attribute is 'YES') • ESX2 (destination) is NOT in Maintenance Mode.	The migrated VM under ESX2 (destination) is NOT in Inherited Maintenance Mode.

Scenario 2	Result
(A) • ESX1 (source) is NOT in Maintenance Mode. • VM is in Maintenance Mode. • ESX2 (destination) is in Maintenance Mode.	The migrated VM under ESX2 (destination) is in Maintenance Mode.
(B) • ESX1 (source) is in Maintenance Mode. • VM is in Maintenance Mode. • ESX2 (destination) is NOT in Maintenance Mode.	The migrated VM under ESX2 (destination) is in Maintenance Mode.

Scenario 3	Result
(A) • ESX1 (source) is in Maintenance Mode. • VM is in Maintenance Mode. (the 'inheritedMaintenance' attribute is 'NO') • ESX2 is in Maintenance Mode.	The migrated VM under ESX2 (destination) is in Maintenance Mode.

(B) • ESX1 (source) is in Maintenance Mode • VM is in Maintenance Mode (the 'inheritedMaintenance' attribute is 'NO') • ESX2 (destination) is NOT in Maintenance Mode	The migrated VM under ESX2 (destination) is in Maintenance Mode.

Exporting Data and Images from OneClick

You can export table data from OneClick to a file. Some OneClick views, such as the Topology view, the Neighbors Topology view, and the Link Information view, can be exported as images.

Export Table Data

You can export table data from OneClick to a file. Table data can be exported from the Alarms tab, the List tab, and other tabs.

Follow these steps:

1. Navigate to a table that contains the Export button
   
   (Export) in the toolbar.
2. Click
   
   (Export).
   The 'Export table data to file' dialog opens.
3. Complete the following information:
   – **Save in**
   Specifies the location to save the exported data file.
   – **Save as type**
   Specifies the file type that you want to use when saving the exported data.
   – **File name**
   Defines the name for the exported data file.
   – **Files of type**
   Specifies the type of file format to use. The following file formats are supported for export:
   • Comma separated values (CSV)
   • Tab-delimited text
   • HTML.
4. Select a location to save the file and click Save.
   The file is saved in the directory that you selected.

Copy and Paste Table Data

You can copy and paste OneClick table data to an external application. In the following procedure, the Alarms table is used as an example, but you can use this procedure in other OneClick tables too.

Follow these steps:

1. Select the alarms to export in the Alarms tab of the Contents panel.

   NOTE
   To select all alarms in the Alarms list, click any alarm and press Ctrl+A.

2. Copy the selected alarms as tab-delimited text (Ctrl+C).
3. Open a spreadsheet application or text editor.
4. Paste Ctrl+V to paste the tab-delimited text into a document.
   The data from the Alarms table appears in the spreadsheet application or text editor that you selected.

Fix Exported CSV Files Containing Board.Port Data

When you export a table that includes Board.Port data to CSV and open it in Microsoft® Excel, trailing zeros are truncated. For example, if the Board.Port value is 2.10, it appears as 2.1 in the spreadsheet. These trailing zeros are not truncated if you export to TXT or HTML format. However, you can take a few steps to fix an exported CSV file in which trailing zeros have been truncated.

Follow these steps:

1. Rename your CSV file from <*filename*>.csv to <*filename*>.txt.
2. Select File, Open in Microsoft Excel.
3. Select 'Text Files' from the Files of Type drop-down list.
4. Select your file and click Open.
   The Text Import Wizard dialog opens.
5. Select Delimited, and click Next.
6. Select Comma, and click Next.
7. Select the column that contains data with trailing zeros.
8. Select Text from the Column Data Format section.
9. Click Finish.
   Microsoft Excel opens the file. The trailing zeros are preserved.

Export Topology Views

Some OneClick views, such as the Topology view, the Neighbors Topology view, and the Link Information view, can be exported outside DX NetOps Spectrum.

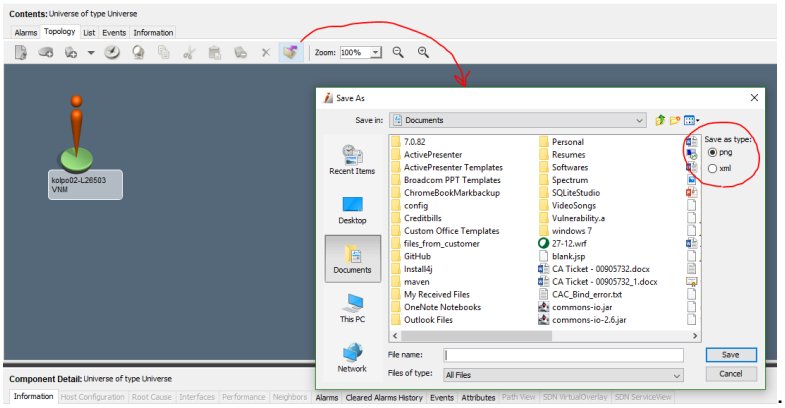
You can now export the topology in XML and PNG formats. You can import the topology to third-party imaging applications supported by mxGraph (for example, draw.io) and change the topology based on your requirements. You can then export the updated topology to any supported output formats of that third-party imaging application.

NOTE

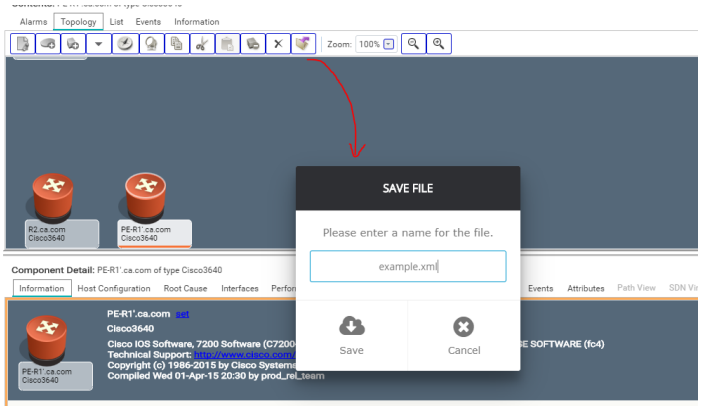
- The image is saved according to the current zoom level in the view.
- To get clearer and bigger images, zoom the topology before exporting it.
- Only the model name and device type are displayed along with the image.
- Overlapping images are not shown properly.

Follow these steps:

1. Navigate to the entity that has a topology image.
2. Select the desired topology or container in the Navigation panel.
3. Click the Topology tab.
4. Click (Export View to PNG or XML)  .
   The Save As dialog opens. By default, the topology is saved in PNG format.
5. Perform one of the following tasks:
   a. In Console: enter the desired file name and select **XML** as the file type from the **Save as type**.



b. In WebApp: enter the desired file name and the file extension as **.XML**.



6. Select a location to save the file and click **Save**.
   The topology is exported to the specified location.

You can now import the saved XML file to a third-party imaging application, re-arrange the topology, and export it to any supported file format.

> **WARNING**
> The default setting for image size to export is 640x480 pixels. You can create a large image when exporting the Topology view (4000x4000 pixels or larger). Excessive size can cause an out-of-memory error in OneClick. Reduce the size of the image that you are exporting by zooming out in the Topology view. Consult your OneClick administrator to increase your client memory settings.

Keyboard Shortcuts

The following keyboard shortcuts are available in the OneClick Console.

- **CTRL + P**
  Opens the Print dialog from which you can specify what you want to print and which printer you want to use.
- **CTRL + G**
  Sends an ICMP Ping to the selected devices, from the SpectroSERVER modeling the device.
- **CTRL + R**
  Opens the TraceRoute dialog that lets you perform the traceroute operation on SpectroSERVER- and SDC-moldeled devices. For directly connected devices, it performs the traceroute operation through SpectroSERVER. For devices modeled through SDC, check the connection status and perform the traceroute operation. In this case, the traceroute operation is performed through SDC.
- **CTRL+T**

Establishes a communication session with the selected device using Telnet, from the SpectroSERVER modeling the device.

- **CTRL+H**
  Establishes an encrypted communication session with the selected device using Secure Shell (SSH), from the SpectroSERVER modeling the device.
- **CTRL+L**
  Polls the selected devices from the SpectroSERVER modeling the device.
- **CTRL+W**
  Web administration. Launches a browser using the IP address of the selected device. Available only for models that have the WebAdminURL attribute.
- **ALT+LEFT ARROW**
  Goes back to a previous container or device.
- **ALT+RIGHT ARROW**
  Goes forward to a container or device after navigating back.
- **ALT+V, S**
  Shows or hides the Status bar.
- **ALT+V, N**
  Shows or hides the Navigation panel.
- **ALT+V, C**
  Shows or hides the Contents panel.
- **ALT+V, D**
  Shows or hides the Component Detail panel.
- **ALT+H**
  Opens the Help menu from which you can access DX NetOps Spectrum support, DX NetOps Spectrum training information, and DX NetOps Spectrum documentation.

Mobile Application

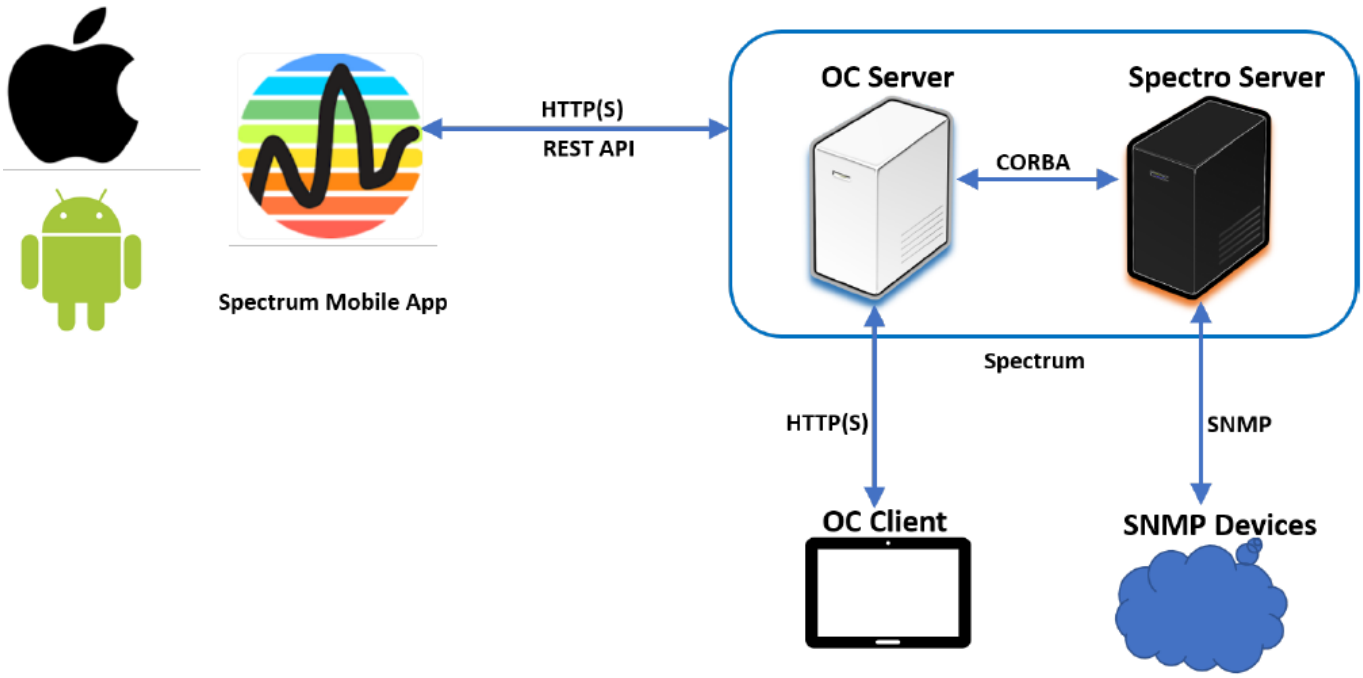
DX NetOps Spectrum Mobile lets you view the alarms through your mobile devices. This application is built on the Sencha framework and it supports both the iOS and Android platforms. The application displays detailed information of alarms such as landscape, description, date, and time. You can configure the application to poll the DX NetOps Spectrum webserver periodically and to obtain the recent alarms. The application lets you perform alarm actions such as view, search, filter, acknowledge, and assign troubleshooter. To connect the application to the OneClick server, you must specify the host name or IP address of the OneClick server host and the server port number.

The application lets you view the top alarms through your mobile devices. You can download the mobile application from App Store or Play Store.

- iOS version 8 or higher
- Android version 4.1 or higher

Architecture

This application runs on the Flutter framework. The application contacts the OneClick server using REST API to obtain the alarm-related information. The OneClick server collects the alarm information from the SpectroSERVER through CORBA.



The latest version of the application includes the following enhancements:

- Support for the alarms filtering capability.
- CPU/memory performance graph support for the SNMP devices.
- Ability to view events for the devices generated in the last four hours.
- Ability to change the device state to the maintenance mode.
- Ability to search the devices on the server.
- Support for the global collection alarm and device views.
- Support for the dark mode theme.
- Various bug fixes and performance improvements.

Using the Mobile Application

Using the mobile application, you can view an overview and details of the alarms and the associated devices and perform various actions, such as the following:

- View alarms by severity on the dashboards and also access the devices that are in a critical condition
- Search for alarms based on the severity, alarm title, device name, device IP address
- View alarms details for approximate 20000 alarms. The details include information, such as description, possible causes, symptoms, recommendations, and the devices that are impacted by the alarms.
- Acknowledge alarms
- Clear alarms and refresh the alarms list
- View device details of 10000 devices and the associated alarms
- Report the alarms or provide feedback
- Share the details on various social media platforms

This article contains the following topics:

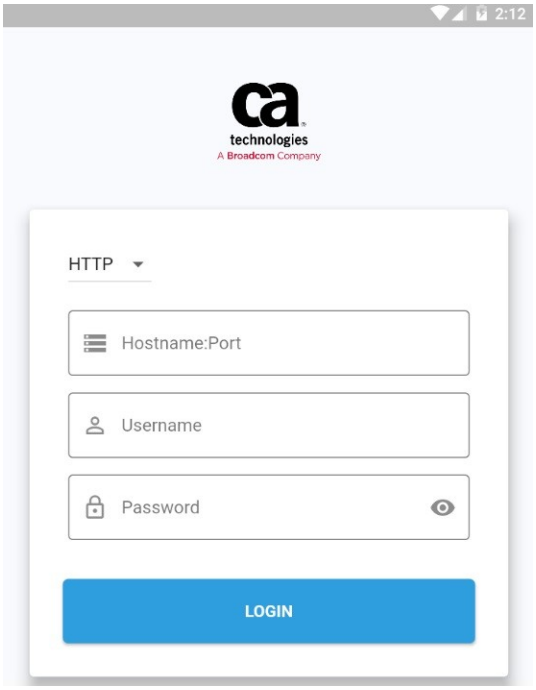
Prerequisite

Ensure that the following actions are completed before you access the mobile application.

- You have installed the mobile application
- You are connected to the enterprise network through the VPN or Wi-fi.
- The OneClick server runs on the enterprise network that you are connecting to from your mobile.

Accessing the Mobile Application

Open the mobile application from your mobile device, and connect it to a OneClick server host.



- Select the HTTPS option from the drop-down to connect to the OneClick server. By default, you connect using HTTP.
- **Hostname:Port**: specify the hostname or IP address and the port number to connect to the OneClick Server. Define the value in this format: *<hostname or IP address>:<port number>*
- The default port number for HTTP is 80 and for HTTPS is 443. The mobile application automatically adds these port numbers. However, if the OneClick server is configured with a different port number, you must specify the port number in the format provided.
- **Username**: define the username to connect to the OneClick server.
- **Password**: define the password to connect to the OneClick server.
- **Login**: click the button to log in to the mobile application.
- The mobile application saves the credentials, by default.

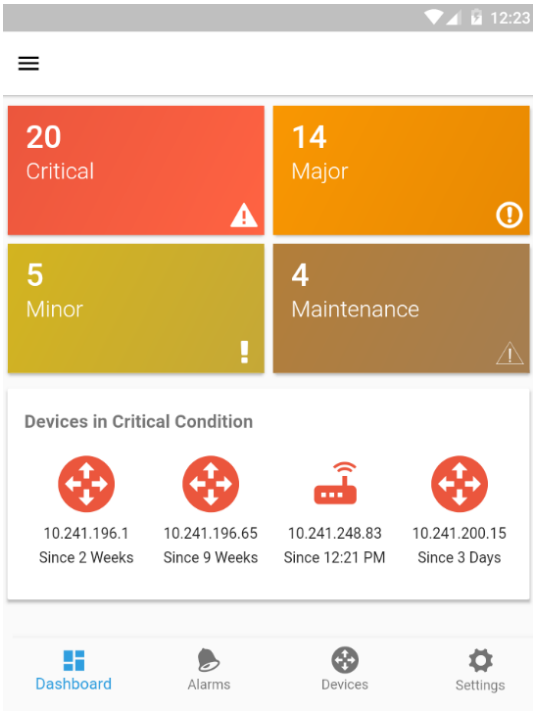
Options Available with the Application

After you log in to the mobile application, you can access the following screens and perform actions to manage the alarms from the DX NetOps Spectrum Database:

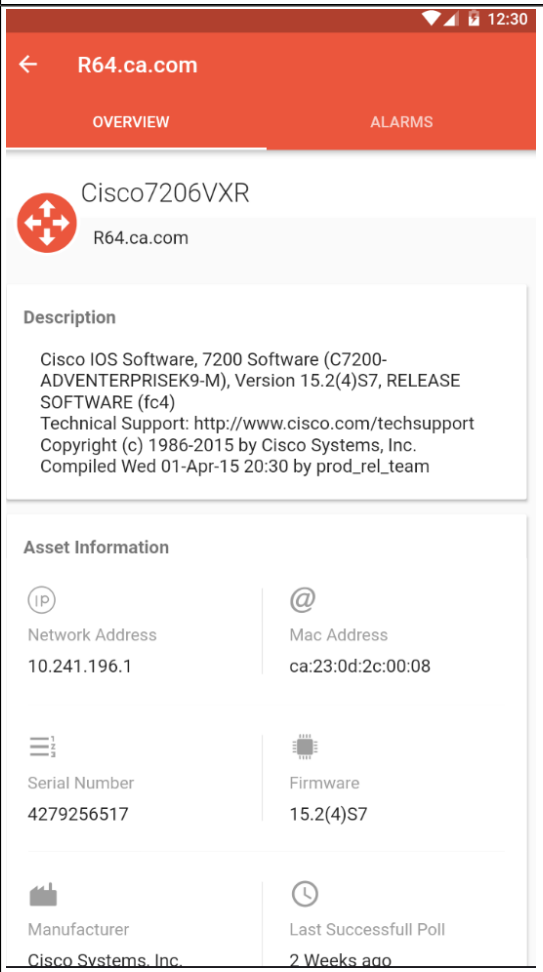
- Dashboard
- Alarms
- Alarm Details
- Device Details
- Settings

Dashboard Screen

The Dashboard is the default view when you log in to the mobile application. On this screen, you can view a summary of alarms based on the severity and any device and the type of device that is in a critical condition.

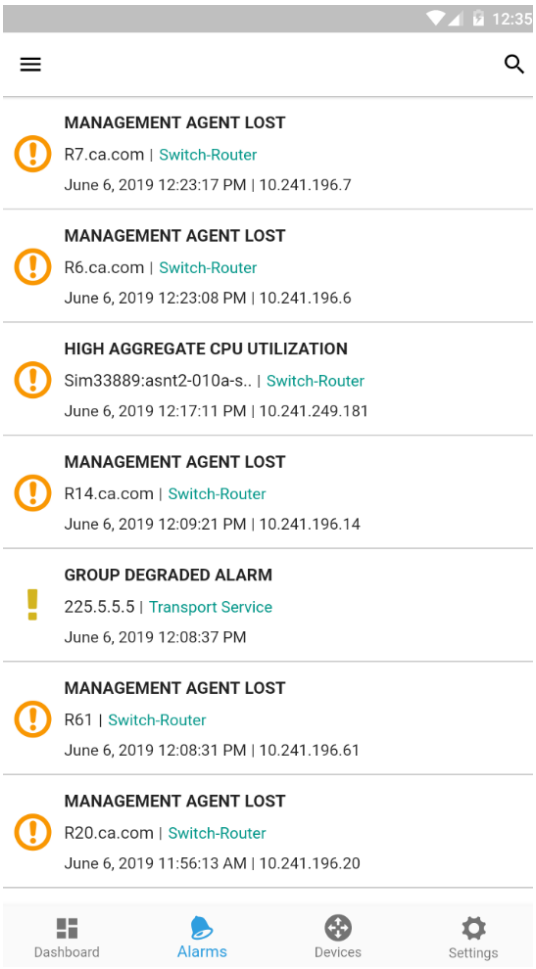


Actions Available on Dashboard Screen	Description
View Device Details	Click on any device on the Dashboard to view the overview of the device and any associated alarms.

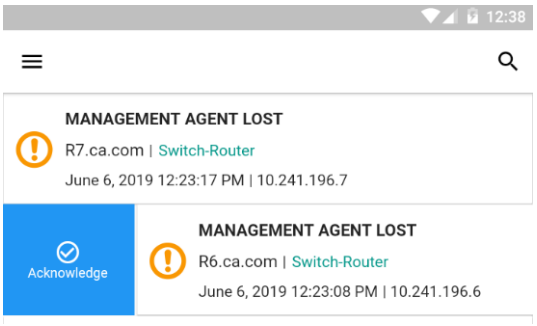
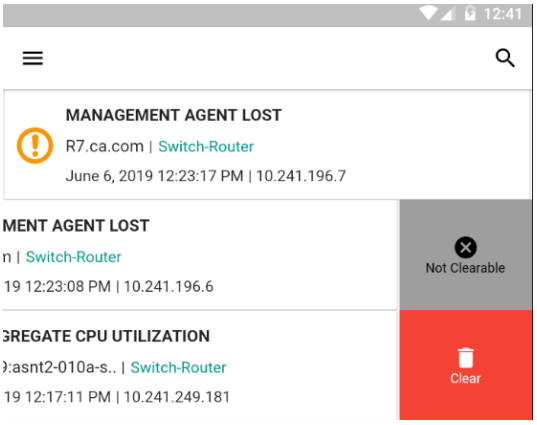
Actions Available on Dashboard Screen	Description
	

Alarms Screen

Once connected to the specified OneClick server, the mobile application fetches the latest 20000 alarms (Critical, Major, and Minor) and displays on the Alarms screen. This list of 20000 alarms is grouped in the descending order of the date of occurrence of alarms. The alarms are polled automatically and refreshed.



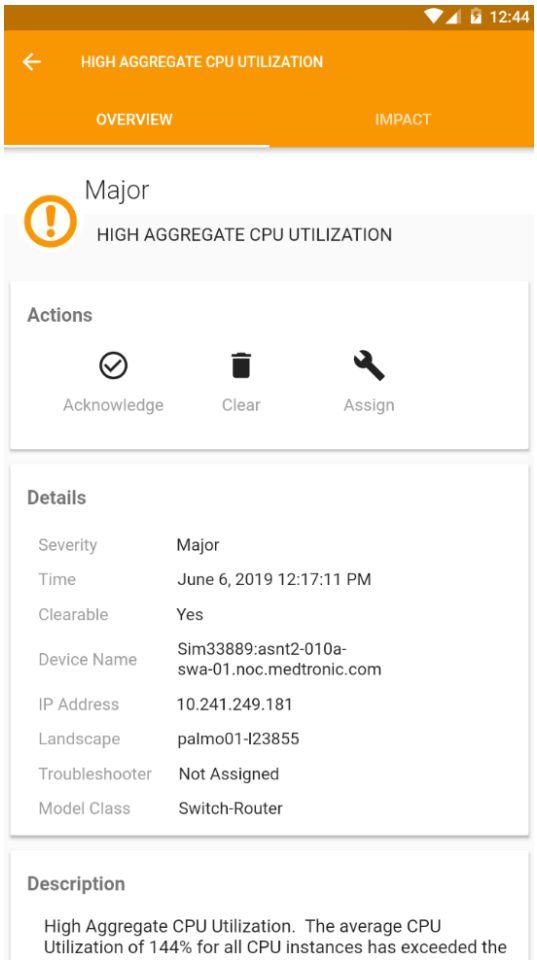
Actions Available on the Alarms Screen	Description
Search Alarms	The Alarms screen lets you search alarms by alarm title, device name, device IP address.
Filter Alarms	The Alarms screen lets you filter the latest alarms by severity. You can sort Critical, Major, and Minor alarms from the list of alarms. By default, all alarms are displayed irrespective of the severity. To view alarms of a severity type, click on the Critical, Major, or Minor option at the bottom of the screen.

Actions Available on the Alarms Screen	Description
Acknowledge Alarms	To acknowledge an alarm, select an alarm and slide to the right to view the acknowledge option.  When you click on the Acknowledge option, a message Alarm updated appears indicating that you have acknowledged the alarm.
Clear Alarms	To clear an alarm, select an alarm and slide left. You can clear alarms only in instances where the **Clear** option is available.  After you click on the Clear option, the alarm is cleared from the OneClick server and the DX NetOps Spectrum database. After you clear the alarm, the list of alarms is refreshed.

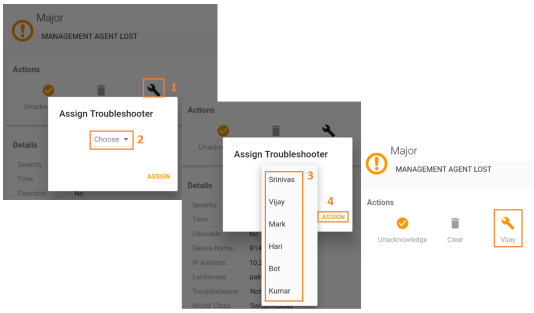
Alarm Details Screen

When you select an alarm from the list, the alarm detail screen displays the following details:

- **Overview** tab:
  - Severity: Critical, Major, Minor
  - Alarm details
  - Description of the alarm
  - Alarm symptoms
  - Possible causes of the alarm
  - Any recommended action
- **Impact** tab: In this tab, you can view the list of impacted alarms. The alarm details screen is as follows:

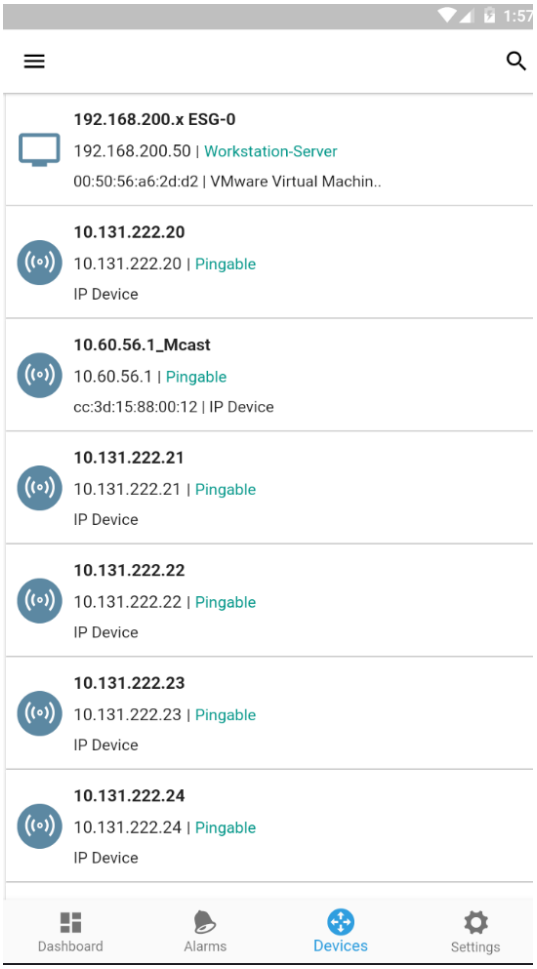


Actions Available on the Alarms Details Screen	Description
Acknowledge Alarms	When you select an alarm, you view the details of the alarms, which lets you acknowledge that alarm. Click the **Acknowledge** icon in the **Actions** section to acknowledge the alarm. To unacknowledge, double-click **Unacknowledge** in the **Actions** section
Clear Alarms	Click the Clear icon in the Actions section, to remove the alarm from the OneClick server and the DX NetOps Spectrum database. After you delete the alarm, the list of alarms is refreshed. This option is enabled only for alarms that can be cleared.

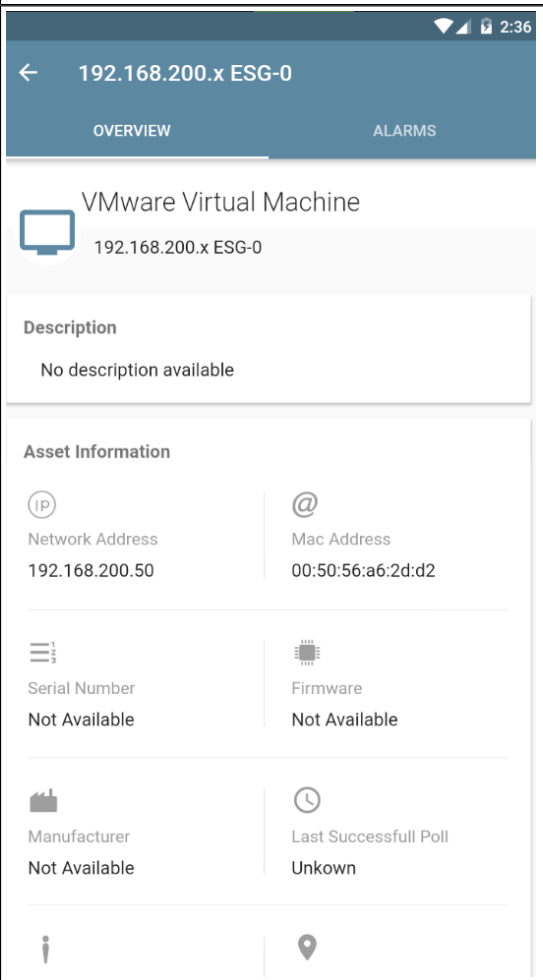
Actions Available on the Alarms Details Screen	Description
Assign Troubleshooter	When you select an alarm, you can view the alarm details, which lets you assign a troubleshooter for that alarm. Before you assign a troubleshooter, ensure that you have configured a list of troubleshooters in the OneClick server. When the assignment is complete, an email notification is sent to the troubleshooter about the assignment. Click the Choose drop-down, to view the list of names that you can assign the alarm to for troubleshooting.  After you select a name, click the Assign button. The alarm appears assigned to the name that you selected. You can unassign only the assigned alarms to a troubleshooter. To unassign, click the Assign icon and the name gets removed.

Device Details Screen

Click on the **Devices** option at the bottom of the screen to view the list of devices from the DX NetOps Spectrum database. The device details screen is as follows:

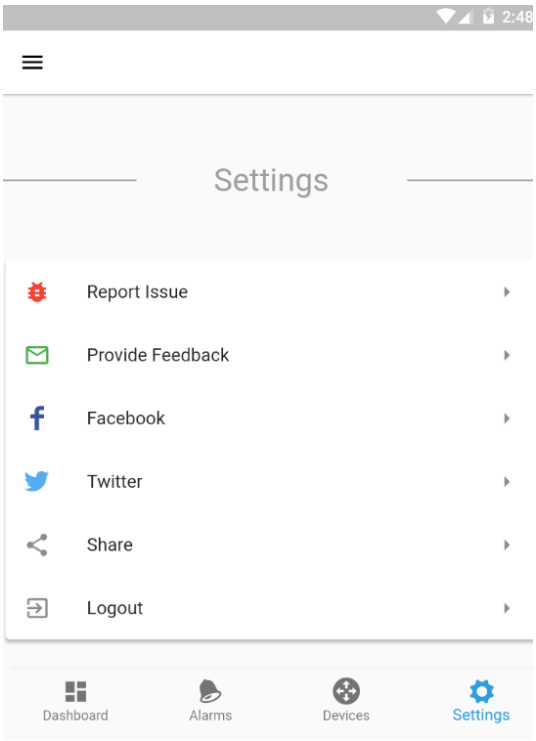


Actions Available on Device Details Screen	Description
View Device Details	To view the device details, click on any device from the list. The device details page displays an overview of the asset in the Overview tab, such as network address, Mac address, Serial Number, Firmware, Manufacturer, polling details, landscape, location. In the Alarms tab, you can view the list of alarms associated with the device.

Actions Available on Device Details Screen	Description
	

Settings Screen

Click the **Settings** option at the bottom of the screen to report an issue, provide feedback, share the details, or to logout from the mobile application. Logging out also logs you out from the OneClick server.



OneClick WebApp

About OneClick WebApp

From 10.4, instead of launching the WebApp separately, the WebApp is embedded in DX NetOps Spectrum. The OneClickWebApp launches in a separate tab with the same port details as DX NetOps Spectrum. The OneClickWebApp address is http://<OC-HostName>:<OC-Port>/spectrum/webapp. Manual configuration is not required in the console-menubar.jsp page. Logging out of the WebApp redirects you to the DX NetOps Spectrum administration page.

> **NOTE**
> If javaSript is not enabled in your web browser, WebApp does not work. To enable javaSript, see How to enable JavaScript in Windows.

OneClick WebApp Improvements in 10.4.2

The following improvements are done in the current release:

- Window Undocking is supported. Click the **UP arrow** (in the browser's title bar) as shown in the following image to toggle window docking:

- DX NetOps Spectrum OneClick WebApp is 508 compliant.
- The memory footprint of webtomcat is reduced by around 60%, hence the OneClick WebApp is that much lighter.
- Launch the **OneClick WebApp in-context** to open the alarm, explorer, and topology. For more information, see Launch OneClick Clients with Context.

OneClick WebApp Improvements in 10.4.2.2

This release includes the following OneClick WebApp-related enhancements:

OneClick WebApp Audio Alarm

An audio message announces the new alarm in WebApp. For more information, see Alarms Tab Preferences.

OneClick WebApp Security

The OneClick WebApp administrator can use the **Admin** console to view all the open WebApp sessions from multiple machines. When you try to launch an open session, you can see a warning message `Session is mirrored by OneClick WebApp Administrator.`

Memory Requirements for OneClick WebApp Web Server:

For every new session that opens in the client, OneClick WebApp takes around 300 MB to 500 MB of server memory, as it creates a Java process for each client.

NOTE

DX NetOps Spectrum now ensures that the OC WebApp port is updated to 9443 if it is available, or to any other available port.

NOTE

10.3.2 introduces the download and upload option as an improvement to the OneClick WebApp. These upload and download improvements are made where the client file system is made available.

The Service Desk ticket launch now opens in a separate browser.

Configure OneClick WebApp in Docker

This section describes the procedure to configure the OneClick WebApp in Docker. You need the X Windows System virtual framebuffer X server package (`xorg-x11-server-Xvfb)` to start the webswing.

Follow these steps:

1. While creating the container, create a port mapping like it is done for the OneClick port.
   For OneClick WebApp:

   ```
   docker run -e ROOT_PASSWORD=???.qaperf184 -e MAIN_LOCATION_SERVER=719de9a39c46 -e
    MAIN_LOCATION_SERVER_IP=172.17.0.2 -e TOMCAT_PORT=8080 -p 9090:8080 -p 9443:9443 -e MASTER_NODE=docker-
   rh74vm2 -it 1032ocimage
   ```

   Here 9443 is the port number that WebApp uses, once the OC container is created.
2. Perform one of the following tasks to install the package.
   a. If the package is **available**, follow these steps:
      a. Copy the `xorg-x11-server-Xvfb` file to the `<docker_host>/container` directory using the ‘docker copy <filename> <containerName:/path>’ command.
      b. Install the package.
   b. If the package is **not available**, follow these steps:
      a. In the docker container, go to the `/etc/yum.repo.d` directory.
      b. Move all the repo file in this directory except the `Vi test.repo` file to a temporary location.
      c. Open the `Vi test.repo` file.
      d. Insert the following code chunk:

      ```
      [test]
      name=test
      ```

```
baseurl=http://vault.centos.org/<version>/os/x86_64
gpgcheck=0
enabled=1
```

e. Save and close the file.

f. Run the following commands:

```
yum clean all
yum repolist
```

g. Install the package using the yum install `xorg-x11-server-Xvfb` command.

3. When package installation is completed, move the repo files that you put in a temporary location to the current directory.

Steps to Run OneClick WebApp on Windows

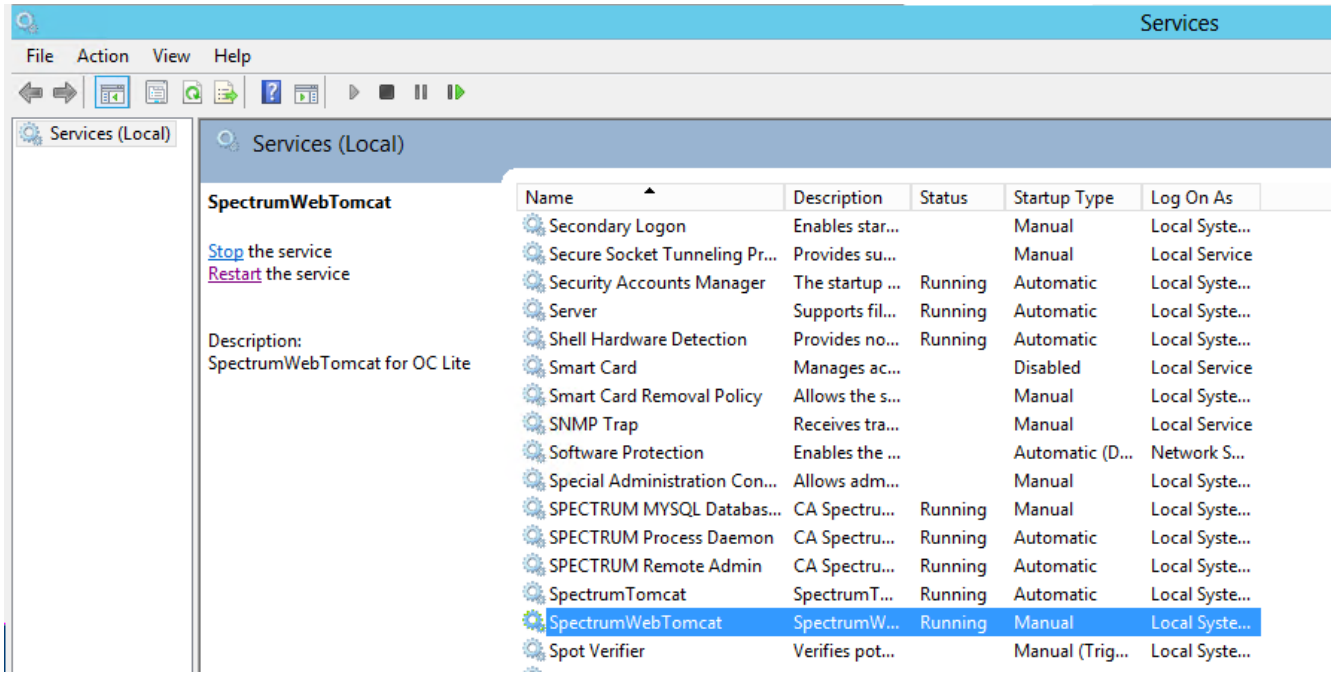
This section describes the steps to run OneClick WebApp on Windows.

NOTE
The OneClick WebApp application runs in a separate tomcat web server, also called as the OneClick WebApp web server. The OneClick WebApp web server runs on a service that is referred to as the SpectrumWebTomcat. By default, the SpectrumWebTomcat service is in a running state.

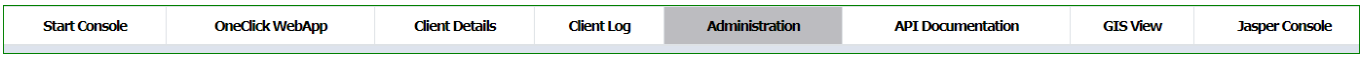
Following these steps:

1. To start the SpectrumWebTomcat service, launch **services.msc** from your windows machine. Find and select the SpectrumWebTomcat service option (which as mentioned, is in the stopped state) and start it. The OneClick WebApp web server takes a couple of minutes to start.

   NOTE
   The Webtomcat web server and the OneClick web server, run on separate ports. DX NetOps Spectrum ensures the assigning of the next available port number (by reading the OneClick webserver port number) to the Webtomcat webserver.



a. Launch the OneClick Admin page and select the OneClick WebApp option (as shown here) to launch the OneClick WebApp application. Directly launching the WebApp URL is not supported. You must launch it from the OneClick Admin page.



NOTE

To change the port number that has been assigned to the OneClick WebApp web server, navigate to the following location: **%SPECROOT%Webtomcat>conf>server.xml** file and look in the connector folder for the HTTP protocol. For example:

```
<Connector port="9443" protocol="HTTP/1.1" connectionTimeout="20000"
 redirectPort="8443" />
```

In the example that is given above, the port number that is assigned to the OneClick WebApp web server is 9443.

2. If the OneClick WebApp web server does not start, analyze, and fetch the webtomcat logs for troubleshooting, from the following log locations:

```
%SPECROOT%webtomcat/bin/webswing.log
%SPECROOT%webtomcat/logs/stdout.><date-stamp>.log
```

Steps to Run OneClick WebApp on Linux

This section describes the steps to run OneClick WebApp on Linux. Install the 'X virtual frame buffer' (Xvfb) for Linux machines.

Prerequisites

1. Download and Extract the installation package (xorg-x11-server-Xvfb-1.10.4-6.el6.x86_64.rpm) to a temporary directory.

```
wget http://vault.centos.org/<version>/os/x86_64/Packages/<package_name> tar <package_name>
```

2. Install the package using one of the following methods.

```
yum install <package_name>
    OR
yum localinstall <package_name>
```

3. Verify that the following Linux packages are available, else install them:
   These packages enable the graphical environment that is required to use the Swing framework.
   DX NetOps Spectrum Dependencies:
   - xorg-x11-server-Xvfb
   - libXdmcp
   - xorg-x11-server-common
   - libXfont2
   - libxkbfile
   - xorg-x11-xkb-utils

   OneClick WebApp Dependencies
   - libXext
   - libXi
   - libXtst
   - libXrender

   NOTE
   For more information see the Webswing documentation.

Following are the steps to run OneClick WebApp on Linux:

NOTE

The OneClick WebApp application runs in a separate tomcat web server, also called as the OneClick WebApp web server. The OneClick WebApp web server runs on a service referred to as the SpectrumWebTomcat.

1. (Optional) To start the OneClick WebApp web server, navigate to the **$SPECROOT>webtomcat>bin** directory and execute the following command:
   ```
   ./startWebTomcat.sh
   ```

   NOTE
   The OneClick WebApp web server and the OneClick web server, run on separate ports. 10.3.1 ensures the assigning of the next available port number (by reading the OneClick web server port number) to the OneClick WebApp webserver.
2. Launch the OneClick Admin page and select the OneClick WebApp option (as shown here) to launch the OneClick WebApp application. Directly launching the WebApp URL is not supported. You must launch it from the OneClick Admin page.

   NOTE
   To change the port number that has been assigned to the OneClick WebApp web server, navigate to the following location: **$SPECROOT>webtomcat>conf>server.xml** file and look in the connector folder for the HTTP protocol.
   For example:
   ```
   <Connector port="8081" protocol="HTTP/1.1" connectionTimeout="20000"
    redirectPort="8443" />
   ```
   In the above example, the port number that is assigned to the OneClick WebApp web server is 8081.
3. Troubleshoot any issue in starting the OneClick WebApp web server, analyze, and fetch the web tomcat logs from the following log locations:
   ```
   %SPECROOT%webtomcat>logs>Catalina.out
   %SPECROOT%webtomcat/bin/webswing.log
   ```
4. To stop the OneClick WebApp application, run the following command:
   ```
   ./stopWebTomcat.sh
   ```

SSL Support for OneClick WebApp

The SSL feature is enabled with OneClick and available with previous versions of DX NetOps Spectrum. Installation takes care of the client certificates and setting the SSL to true in the configuration. Selecting the link the OneClick administration page loads the DX NetOps Spectrum console without errors.

NOTE
If OneClick is SSL enabled before upgrading, the OneClick WebApp URL appears in **https**.

Enable SSL support

This section discusses the steps to enable SSL support or change the OneClick WebApp URL.

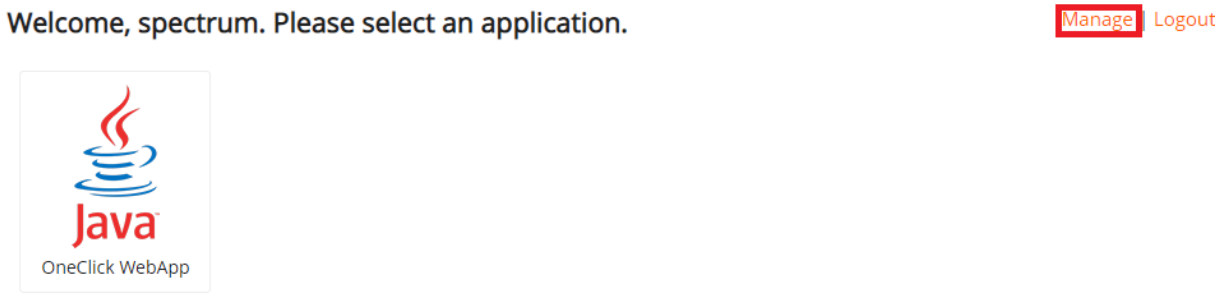
Follow these steps:

1. Navigate to the **tomcat >conf>server.xml** file and copy the connector https ports-related information and paste it in the **webtomcat>conf folder>server.xml file** under the **https connector** section.
2. Change the SSL connector **port entry** that you pasted in the **webtomcat/conf/server.xml** file, so that it does not conflict with the standard OneClick Tomcat SSL **port**.
3. Restart the web tomcat service.

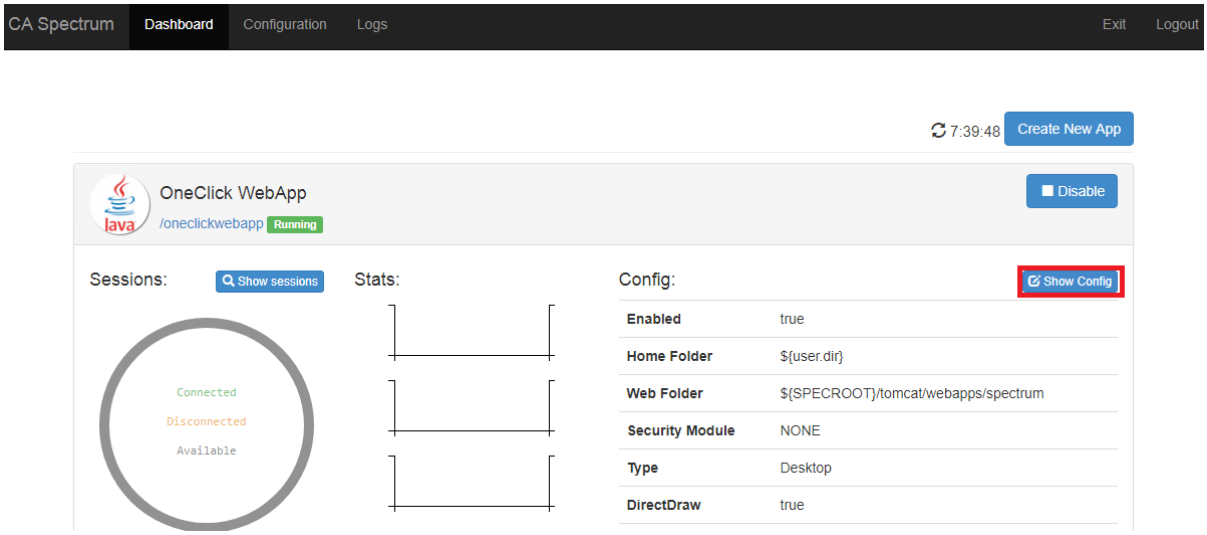
Post Upgrade Tasks

Follow these steps if you have enabled SSL in OneClick after upgrading to 10.3.x or if you have changed the OneClick ports after upgrading to 10.3.x:

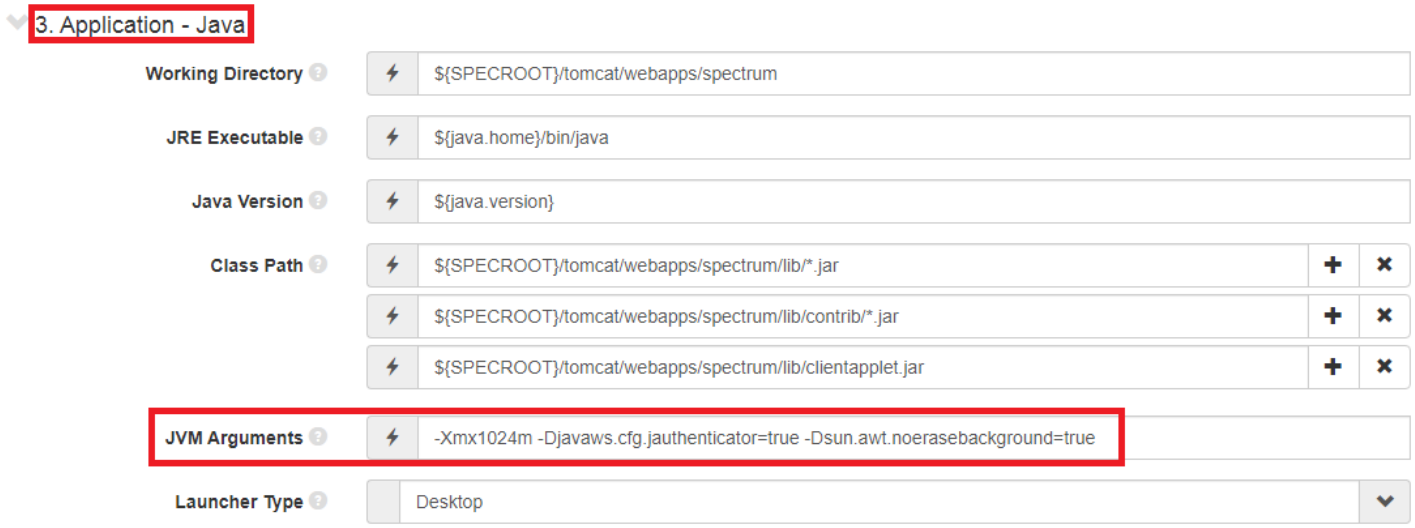
1. Log in to the OneClick WebApp administration page. The default URL is:
   `https://<webapp-hostname>:<webapp-port>/spectrum`
2. Log in with the **spectrum/spectrum** username/password credentials and select **Manage**.



3. Select **Show Config**, to view the configuration.



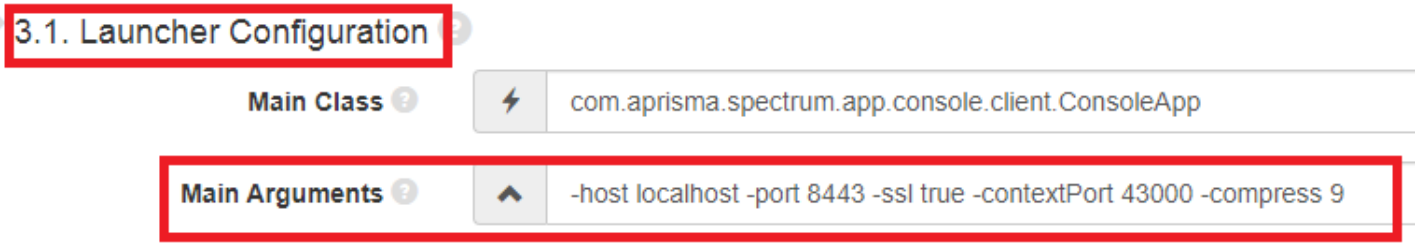
4. Add the following for the JVM Arguments under the **Application – Java** section after the `-Dsun.awt.noerasebackground=true` entry:
   `-Djavax.net.ssl.trustStore=${SPECROOT}/custom/keystore/cacerts-Djavax.net.ssl.trustStorePassword=changeit`



Cross verify these values with the OneClick tomcat server.xml configuration (only if SSL is enabled).

NOTE
Do not set the ${SPECROOT} to an absolute path. However, change the path/filename of the keystore (the default is cacerts) and password (the default is changeit) if they are not the default entries.

5. Navigate to the **Launcher configuration>Main Arguments** section and select the OneClick Hostname, OneClick port, and the SSL values, if SSL is enabled, change the –ssl to true.



NOTE
These values should be in sync with the OneClick details. The SSL port should be the same as your original Tomcat SSL port, that is 8443 (if SSL is enabled). If you changed the OneClick port, change it.

6. After you have made the changes, select **Apply** on the top left corner and re-launch the OneClick WebApp from the administration page.

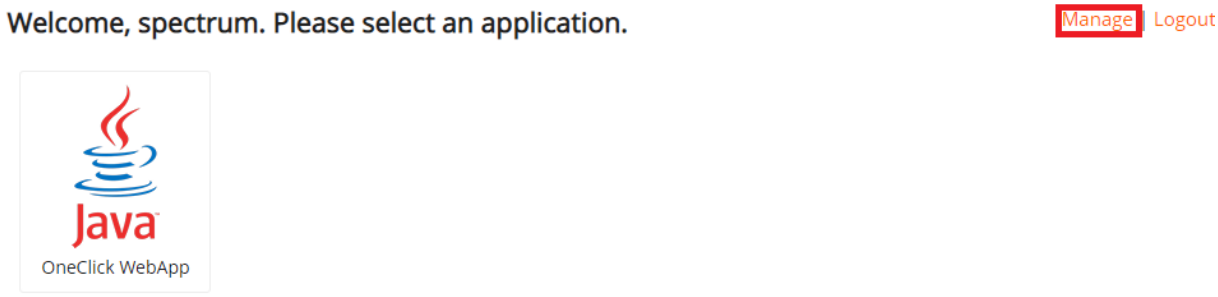
7. Launch the OneClick Admin page and select the OneClick WebApp option, to launch the OneClick WebApp application.

CAC Support for the OneClick WebApp

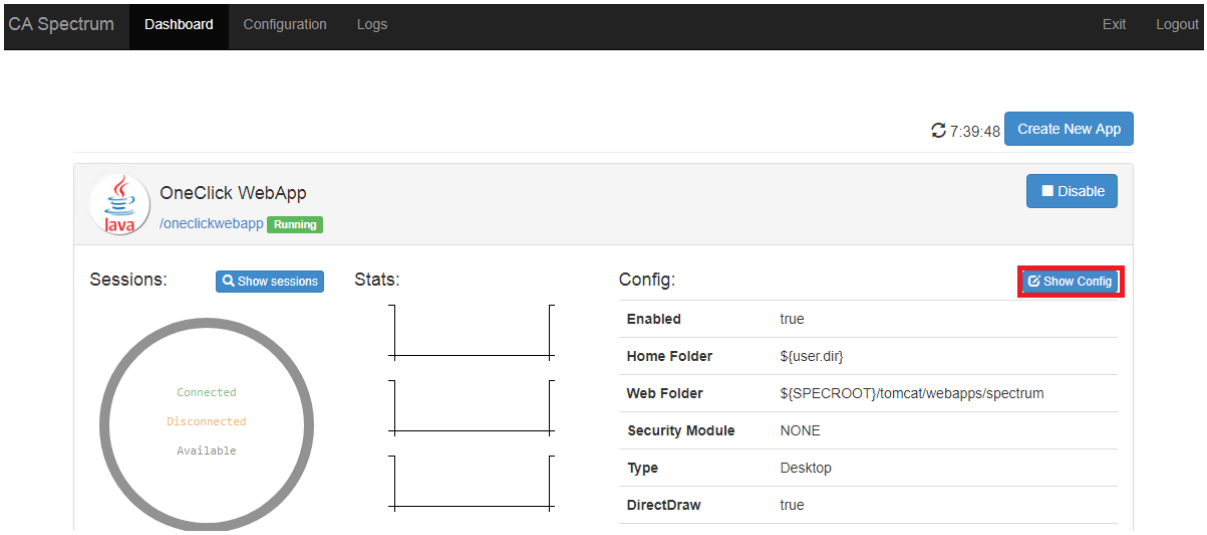
After configuring SSL, configure Common Access Cards (CAC) authentication on the OneClick WebApp.

Follow these steps

1. Configure SSL for the OneClick WebApp.
2. Configure CAC for DX NetOps Spectrum OneClick, see How to Configure DX NetOps Spectrum for SSL and CAC Authentication.
3. Log in to the OneClick WebApp administration page. The default URL is:
   `https://<webapp-hostname>:<webapp-port>/spectrum`
4. Log in with the **spectrum/spectrum** username/password credentials and select **Manage**.

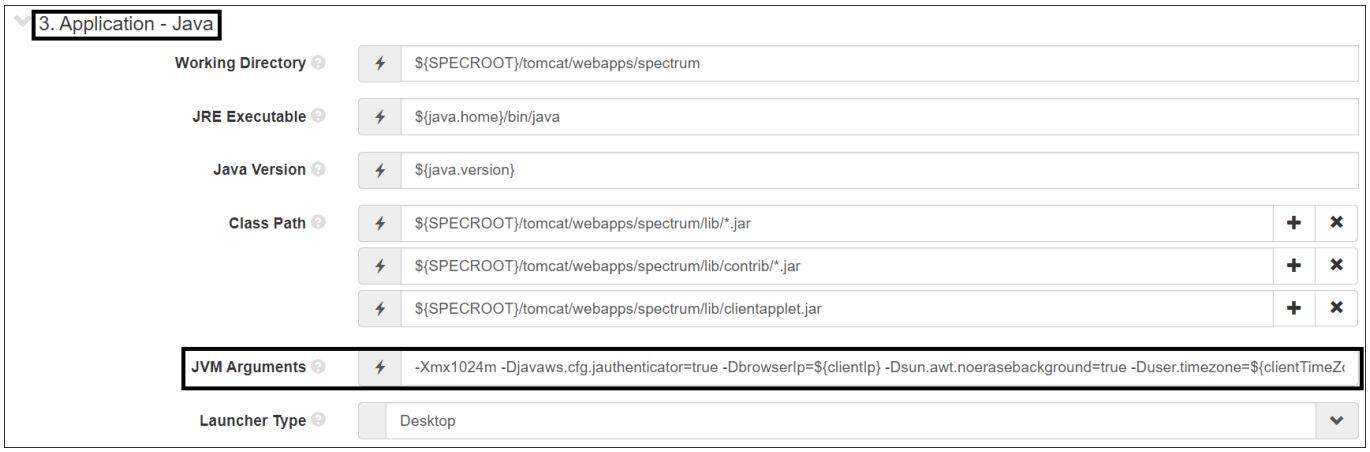


5. Select **Show Config**, to view the configuration.



6. Add the following for the JVM Arguments under the **Application – Java** section after the `-Djavax.net.ssl.trustStorePassword=changeit` entry:

```
-Djavax.net.ssl.keyStore=${SPECROOT}/custom/keystore/cacerts -Djavax.net.ssl.keyStorePassword=changeit
```



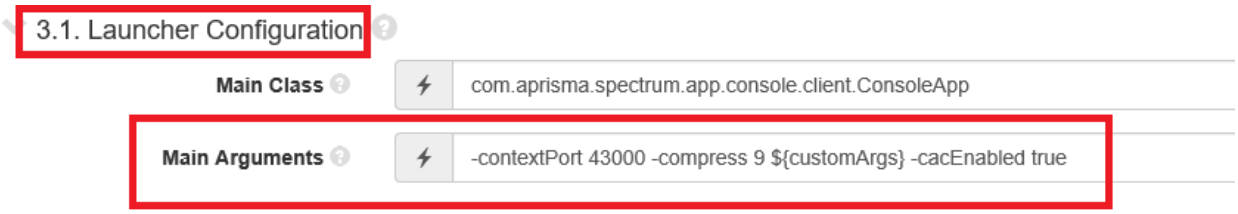
Cross verify these values with the OneClick tomcat server.xml configuration (only if SSL is enabled).

7. Verify that your JVM arguments are as follows:

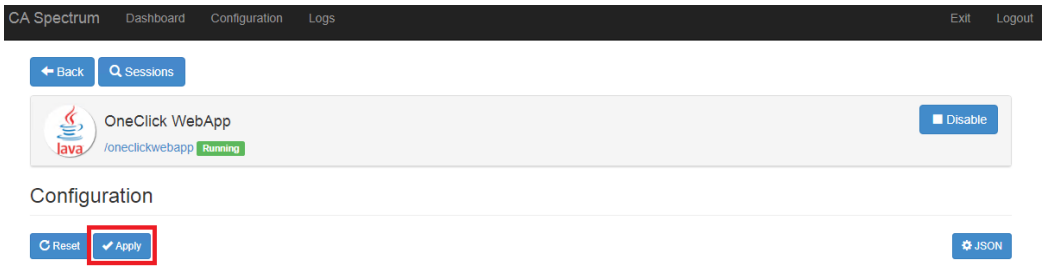
```
--Xmx1024m -Djavaws.cfg.jauthenticator=true
-DbrowserIp=${clientIp}
-Dsun.awt.noerasebackground=true
-Duser.timezone=${clientTimeZone}
-Djavax.net.ssl.trustStore=${SPECROOT}/custom/keystore/cacerts -Djavax.net.ssl.trustStorePassword=changeit
-Djavax.net.ssl.keyStore=${SPECROOT}/custom/keystore/cacerts
-Djavax.net.ssl.keyStorePassword=changeit
```

NOTE
Do not set the ${SPECROOT} to an absolute path. However, change the path/filename of the keystore (the default is cacerts) and password (the default is changeit) if they are not the default entries.

8. Navigate to the **Launcher configuration>Main Arguments** section and add `-cacEnabled true` to the end of the entry.



9. After you have made the changes, select **Apply** on the top left corner and re-launch the OneClick WebApp from the administration page.



10. Launch the OneClick Admin page and select the OneClick WebApp option, to launch the OneClick WebApp application.

(Optional) Configure WebApp User Session Timeout

When a session is inactive for a specific duration, the session automatically logs out after the defined time. You can configure the user session timeout for WebApp users. By default, the session timeout option is disabled. You can enable the session timeout and set the inactive time after which the session must be timed out.

Follow these steps:

1. Log in to the OneClick WebApp administration page.
   Default URL: https://<webapp-hostname>:<webapp-port>/spectrum
2. Select Manage.
3. Select Application, Config.
4. Navigate to the **Application - Session** section.
5. Enter a duration for which session must be active, in the **Session Timeout** field.
   Default: 300 (seconds)
6. Toggle the **Timeout if Inactive** field to ON or OFF.
7. Click **Apply**, to save your changes.
8. Relaunch the OneClick WebApp.

Troubleshooting OneClick WebApp

This section describes the steps you can take to troubleshoot some common issues that you may face during WebApp configuration.

Troubleshooting Enabling the SSL support

This section discusses the procedure to troubleshoot any error while enabling the SSL Support.

Follow these steps:

1. Log in to the OneClick WebApp administration page. Refer to the $SPECROOT/webtomcat/conf/server.xml file for webapp port details.
   Default URL: https://<webapp-hostname>:<webapp-port>/spectrum
2. Log in with the spectrum/spectrum as a username/password.
3. Select Manage.
4. Check whether the OneClick WebApp is enabled or not. If it is not enabled you see the Enable button on the top right corner.
5. Select **Enable**, if the option is in the disable state.
6. Select Show Config, If it is already enabled. The JVM arguments contain the trustStore path and password under the application.
7. Verify the data with OneClick Tomcat server.xml configuration.
8. Navigate to the Launcher configuration, Main arguments.
9. Verify that arguments point to the localhost, where the SSL port and the -ssl is be set to true.

Troubleshooting Launching/Starting Webtomcat

Follow these steps:

1. Check the **tomcat/conf/server.xml** and the **webtomcat/conf/server.xml** files for any port conflicts (often port conflicts will shut down the port).
2. For Linux, the webtomcat does not start if the XVFB is not installed. Refer to the section on Steps to run OneClick WebApp on Linux on this page.

Common Errors with OneClick and Action Required

Error: Spectrum Tomcat is running in SSL mode, Please configure Spectrum WebTomcat also in SSL mode.

Solution: When OneClick is in HTTPS, WebApp also must be in HTTPS.

Error: Not able to reach OneClick Webapp. The process might be stopped. Please start the Spectrum WebTomcat Process and re-launch.

Solution: In case the WebApp process is down, start the WebApp process.

Error: Request Processing failed. Please refer tomcat log for more details.

Solution: In case of any other issues, WebApp shows an appropriate error. Refer to the tomcat log for more details and take appropriate action.

Error: When I launch the WebApp from OneClick administration page, application gets redirected to Administration page. I see the following errror in the tomcat log.

```
java.lang.reflect.InvocationTargetException
Caused by: java.lang.NoClassDefFoundError: org/webswing/ext/services/ImageService
```

Solution: The error occurred because the webtomcat jars are not properly loaded. Restart the webtomcat (stop and start) to solve the issue.

Managing Network

This section describes how to use this product to manage your network.

ATM Circuit Manager

This section discusses how you can discover, manage, and monitor your ATM network.

ATM and Modeling Concepts

This section explains the ATM network technology, and the ATM network model types available in DX NetOps Spectrum. This section also describes the typical ATM infrastructure scenarios.

ATM Technology

Asynchronous Transfer Mode (ATM) is a connection-oriented, network communication architecture that transmits data through pre-established virtual channels called circuits that are similar to telephone calls. Circuits can be established automatically by switched virtual circuit (SVC) signaling, or they can be set up manually to form permanent virtual circuits (PVCs).

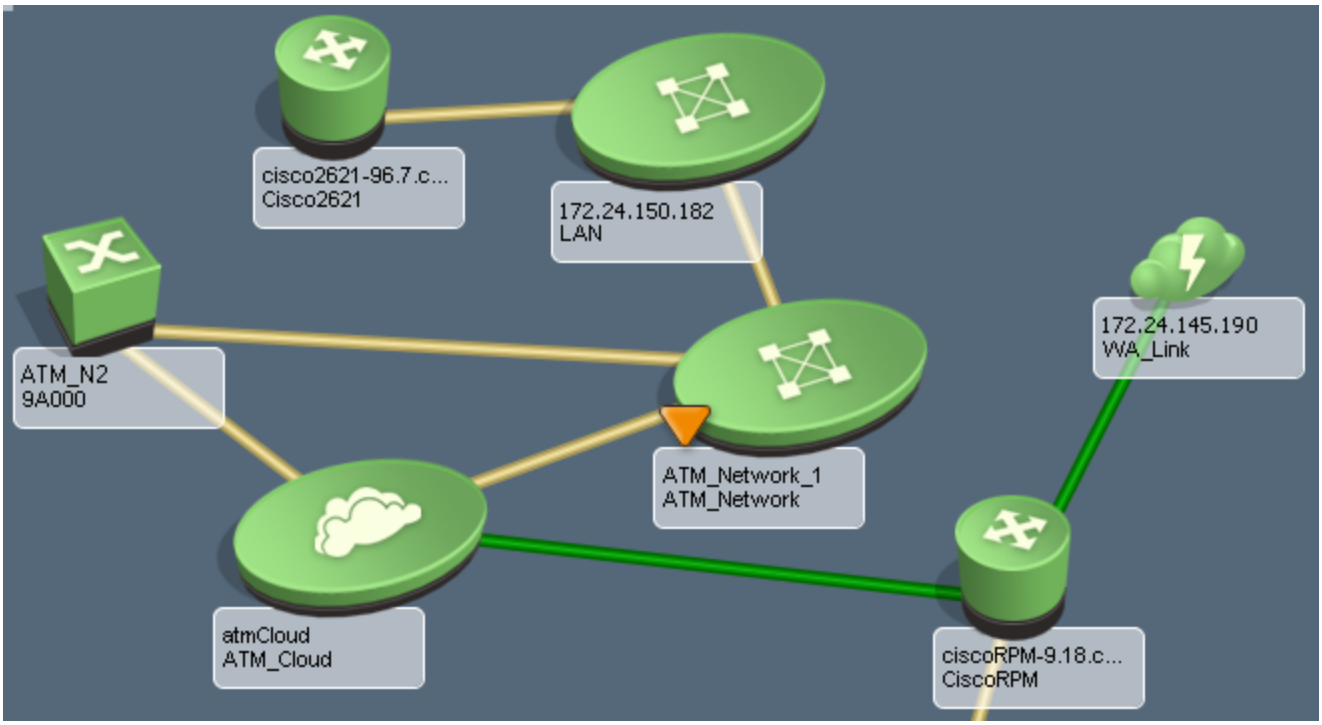
ATM is based on the transmission of fixed-length (53-byte) cells of data. ATM's use of small, fixed-length data cells allows for improved traffic management and traffic shaping. Each ATM cell contains a 5-byte header and 48 bytes of payload. The header includes a virtual path identifier (VPI) and a virtual channel identifier (VCI). These identifiers are used by ATM switches to determine the correct channels on which to transmit particular cells. Transmission is controlled by statistical multiplexing, which awards bandwidth (channels) to devices on a first come, first serve basis.

The combination of small, fixed-length data cells and the efficient use of bandwidth enable ATM switches to communicate time-critical video and audio data, as well as other computer data, across the ATM network. In an end-to-end transmission across a mixed LAN/ATM/LAN network, packets transmitted by a LAN workstation to an ATM switch are segmented into

cells for high speed transmission through ATM channels. At the destination, the cells are then reassembled into packets for use by another LAN workstation.

Overview on ATM Circuit Manager

DX NetOps Spectrum's ATM Circuit Manager lets you model both the physical and logical connectivity of your ATM infrastructure. You can represent an ATM infrastructure that you own and manage, or one that is partially or fully owned and managed by a service provider. The following illustration shows a sample modeled ATM network that includes partial management by a service provider.

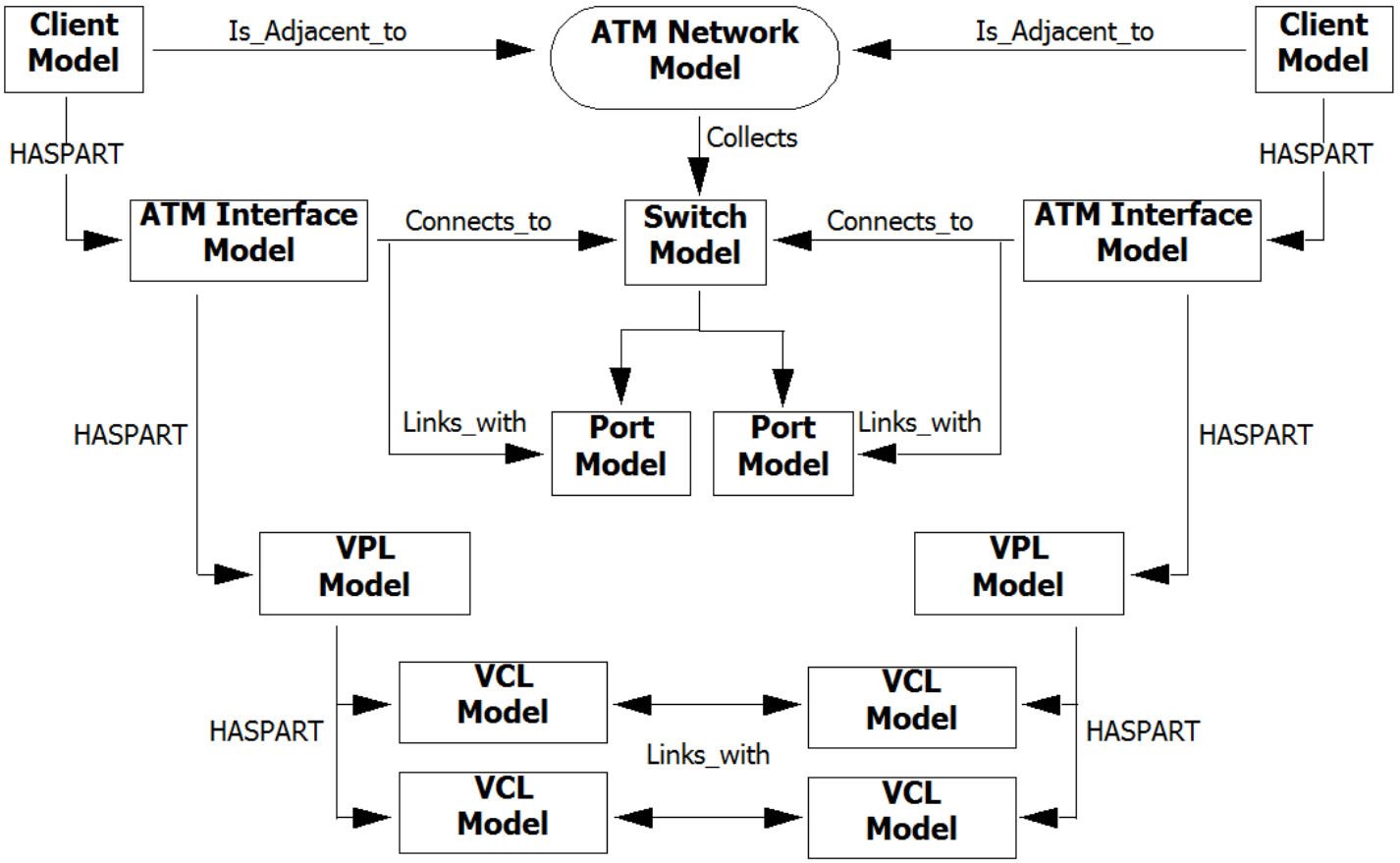


In a fully or partially meshed ATM network, each physical ATM interface may have logical connections with many other ATM devices. However, without ATM Circuit Manager, DX NetOps Spectrum's modeling functionality will only model physical ATM interfaces, leaving logical ATM interfaces unmanaged and limiting connectivity to physical links. Therefore, without ATM Circuit Manager, it is impossible to accurately represent the logical connectivity of the switches and routers in the ATM network.

The physical connectivity of an ATM network is represented by *Connects_to* associations between the ATM physical interface models and the connected device models. This lets you represent true data relay paths between the clients and helps ensure proper fault isolation. You can represent the physical connectivity of the ATM network by connecting a physical ATM interface model to another physical interface model, to a device model, or to an ATM_Cloud model.

The logical connectivity of an ATM network is represented by logical connections between virtual link models. These virtual link models represent the endpoints of a virtual connection. Virtual link models include VPLs (virtual path links, also referred to as virtual path trunks or VPTs) and VCLs (virtual channel links). Virtual link models are associated with a lower-layer interface model using the HASPART relation. The lower-layer interface may be a physical ATM interface model, or, in the case of a VCL, it may be a VPL or VPT model.

The following illustration shows the relationships between the modeled physical and virtual components in an ATM network.



NOTE
An actual ATM network would contain (collect) multiple switches. For simplicity, the illustration shows just one switch.

NOTE
Once you have modeled your ATM network, you can use ATM Circuit Manager to isolate faults within the network, evaluate performance, and monitor quality of service.

ATM Model Types

ATM Circuit Manager uses several model types to enable you to accurately represent your ATM infrastructure. These include ATM_Cloud, ATM_Network, ATMVplLink, ATMVclLink, and UnmanAtmLink.

ATM_Cloud

The ATM_Cloud model type is used in modeling situations where a service provider supplies wide area connectivity to remote sites using ATM links that are leased channels or paths.There is no management access to the service provider's network. In this scenario, management of the ATM links must be provided by the ATM clients also referred to as ATM Edge devices and/or ATM switches connected to the service provider's equipment.

If the devices connected to a cloud have their virtual interfaces modeled (as VPLs/VPTs, VCLs, or both), you can use the interfaces to create logical connections between endpoints.

ATM_Network

The ATM_Network model type is used in modeling situations where the ATM switched fabric is completely managed.

If the devices within the network and the devices connected to it have their virtual interfaces modeled (as VPLs/VPTs, VCLs, or both), you can use the interfaces to create logical connections between endpoints.

ATMVplLink and ATMVclLink

The ATMVplLink model type represents a VPL/VPT, and the ATMVclLink model type represents a VCL. These virtual link models represent the endpoints of a virtual connection, and connectivity between them represents the logical connectivity of the ATM network.

Management of the PVPs and PVCs in the ATM network is achieved by polling the attributes of VPL/VPT and VCL models. DX NetOps Spectrum uses this data to do the following:

- Monitor circuit status and other statistics
- Generate alarms based on the status of a model, for example, when the load on the model exceeds a predefined threshold

In OneClick, you can view a device's physical interface models and virtual link models on the Interfaces tab. The virtual link models in the table are identified by interface type (atmLink and, more specifically, VPL, VPT, or VCL). You can identify the underlying model type of a specific virtual link model by displaying the value for Modeltype_Name using the Attributes tab.

UnmanAtmLink

Some ATM paths or circuits may be manageable from only one endpoint. The device on one side may not have an SNMP agent, may be inaccessible for some reason, or you simply may not have management of the device. If this is the case, you can create a model of type UnmanAtmLink to represent the unmanaged endpoint, and then include the unmanaged model in a logical connection with a managed endpoint.

ATM Network Topologies

There are four typical ATM infrastructure scenarios:

1. You own and administer the switches and clients that comprise the ATM infrastructure.
2. You use a completely leased network through an ATM service provider to provide wide-area connectivity, and you have no management access to the ATM switched fabric.

3. You use a completely leased network through a service provider, and you have a mixture of ATM and Frame Relay interfaces on either side of the leased network.
4. You own and manage your own local area ATM network and lease additional wide-area channels through a service provider's network.

The following subtopics explain how the ATM-related model types are used to represent these different types of ATM infrastructures. For an illustration of the associations that are described, see Overview on ATM Circuit Manager.

A Completely Owned ATM Network

If you own your ATM switches, you will use an ATM_Network model to represent the switched fabric. Switch models will appear within the ATM_Network model and have a *Collects* association with that model. Client models will have an *Is_Adjacent_to* association with the ATM_Network model because the connection between the clients and the switches within the network have been fully modeled at the physical interface level.

A Completely Leased ATM Network

If an ATM service provider supplies wide-area connectivity to remote sites, there is no management access to the service provider's ATM switches. The ATM clients must provide all of the data to monitor the ATM circuits. To support the modeling of this type of network, an ATM_Cloud model is used. All ATM interfaces that connect to the service provider's network will have a *Connects_to* association with the ATM_Cloud model. This modeling association can be established manually as described in Connecting the Physical Interfaces to the Service Provider's Network.

An ATM to Frame Relay Network

It is possible to have a hybrid ATM network that includes multiple Frame Relay to ATM links over a completely leased network. In this scenario, signals transmitted from a local ATM interface through the service provider's network are converted to Frame Relay by a translational bridge before being received by the remote Frame Relay interface, and vice versa. The modeling procedure for this scenario is identical to that for modeling an ATM to ATM logical connection over a completely leased network.

A Completely Owned ATM Switched Fabric with Leased ATM Wide-Area Channels

If you own your local ATM switches, but you connect to a service provider's network for wide-area access, the physical interface(s) of one or more ATM switches may be connected to the ATM_Cloud model. In this case, DX NetOps Spectrum intelligence will automatically create VPL and VCL models to represent the VPLs and VCLs of a physical interface connected to the ATM_Cloud model. If the ATM switches are modeled within an ATM_Network, the ATM_Network model will be adjacent to the ATM_Cloud model.

Modeling the ATM Network

This section discusses how you can accurately model your ATM network.

How to Create an Accurate Model of Your ATM Network

To create an accurate model of your ATM network, you must perform the following high-level tasks:

1. Model the physical elements and physical connectivity in the network using Discovery or manually.
2. Connect the physical ATM interfaces to the service provider's network.
3. Model the logical connectivity between virtual path links (VPLs, also referred to as virtual path trunks or VPTs) and virtual channel links (VCLs).

   NOTE
   This section refers to VPLs and VCLs, collectively, as virtual link models.
4. Create logical connections between the virtual link models. You can create the connections using Discovery (for VCLs on Cisco devices only), create the connections manually, or import them using a file that defines them.

Physical Components Modeling in the ATM Network

To model the ATM circuits correctly, the physical connectivity must be modeled first. You can do this using Discovery or manually. If you use Discovery, you may need to manually customize the results to verify that the model accurately represents the network infrastructure.

Physical Components Modeling Using Discovery

When you use Discovery to model the physical components in an ATM network, and you select in the Protocol Options dialog that ATM protocols be used to map the connectivity between models, Discovery maps the physical connectivity between the ATM devices and places ATM switch models inside ATM_Network container models.

If Discovery does not fully map the physical ATM connectivity of your network, including switch-to-switch and router-to-switch connections, you must rerun Discovery or complete the mapping manually.

> **NOTE**
> Discovery does not create ATM_Cloud models to represent the ATM networks of service providers. After you model your network using Discovery, you must create these models manually and then name them appropriately (for example, Sprint's Network).

If you are using Cisco ATM devices and the VCL interfaces have unique IP addresses, Discovery also resolves the connections between VCL interfaces or between VCL interfaces and Frame Relay DLCI interfaces.

> **NOTE**
> For complete information on modeling using Discovery, see the Modeling and Managing Your IT Infrastructure section.

Once you have modeled the physical components and connections in the ATM network, you can then model the logical connectivity between virtual path links (VPLs, also referred to as virtual path trunks or VPTs) and virtual channel links (VCLs).

Manual Physical Components Modeling

You may need to manually model the ATM infrastructure for various reasons:

- You prefer to manually create all ATM models and connections.
- You have used Discovery to model the devices and map their physical connectivity, but you now need to make some manual modifications to make the network representation fully accurate.
- You have used Discovery to model the ATM infrastructure, but you now need to create ATM_Cloud models to represent the ATM networks of service providers. Discovery does *not* create these models.

After you manually model the devices, you can create connections between container models, device models, and physical interface models. You can connect the physical ATM interface models directly to the switch or client models, but they must be collected by an ATM_Network model to have access to the Logical Connection Table.

> **NOTE**
> For complete information on manually modeling devices and connections, see the Modeling and Managing Your IT Infrastructure section.

After you model the physical components and connections in the ATM network, you can then model the logical connectivity between virtual path links (VPLs, also referred to as virtual path trunks or VPTs) and virtual channel links (VCLs).

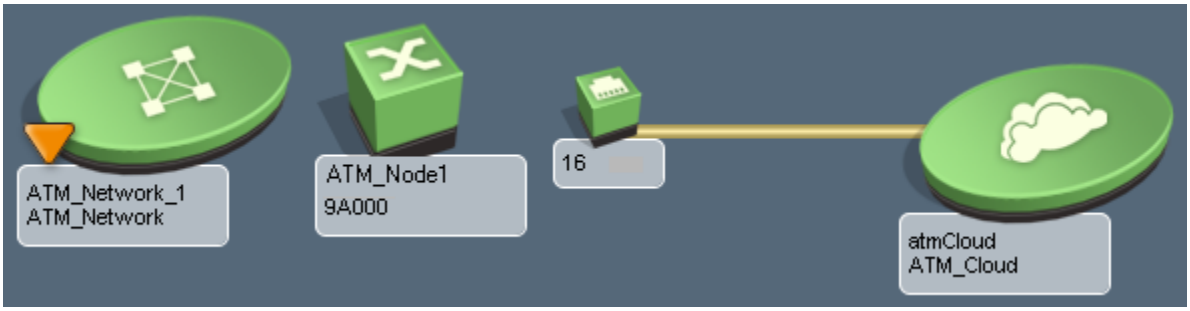
Connect Physical Interfaces to Service Provider's Network

In order for DX NetOps Spectrum to accurately determine the point of an ATM network failure during fault isolation, every physical ATM interface that connects to the service provider's network should be connected to the ATM_Cloud model.

Connecting the physical ATM interfaces to the service provider's network is a manual step regardless of whether you have modeled the ATM network manually or used Discovery.

To connect an ATM interface to the service provider's network

1. On the Interfaces tab of the associated device, select the physical ATM interface to connect to the service provider's network, right-click, and click Start Connection.
2. On the Topology tab, select the ATM_Cloud model that represents the service provider's network, right-click, and click Connect With *<ATM Interface Name>*, where *<ATM Interface Name>* is the name of the interface you selected in the first step.
   A pipe between the ATM_Cloud and the physical ATM interface (and the appropriate network container model) is created to represent the connection. You can double-click the pipe to view the connection between the ATM_Cloud and the physical ATM interface itself, as shown in the following example.



Modeling the Virtual Interfaces in the ATM Network

In DX NetOps Spectrum, physical connections between devices are managed by polling the status and performance data of the endpoints, such as FDDI or HSSI interfaces.

There is no difference between managing logical connections and managing physical connections except that the logical connection endpoints are Virtual Path Links (VPLs) or Virtual Channel Links (VCLs). This section refers to VPLs and VCLs collectively as *virtual links*.

Like physical interfaces, virtual links have objects in a MIB that contain the status, bandwidth, and-depending on the MIB-performance statistics. Physical interfaces, as represented in the MIB-II *ifTable*, have a single-term index, but VPLs have a two-term index, and VCLs have a three-term index.

- A VPL index is in the form *ifIndex.VPI*, where *ifIndex* is the index of the physical interface that the VPL runs on, and *VPI* is the Virtual Path Identifier (the identifier given to the path when it was created).
- A VCL index is in the form *ifIndex.VPI.VCI*, where *VCI* is the Virtual Channel Identifier (the identifier given to the channel when it was created).

Virtual links can be modeled for all ATM clients and switches that have supported ATM MIBs.

Required MIBs

DX NetOps Spectrum can use different sets of SNMP MIBs to manage ATM links and logical connections. When DX NetOps Spectrum creates a device model to represent an ATM device, it also creates an application model based on the ATM MIBs that the device supports.

The following table lists the DX NetOps Spectrum management modules that support ATM devices, the MIBs required to support those devices, and the type of application model that is created to support the required MIBs.

Management Module	MIBs	Application Model Type
CA Spectrum Core Product	ATM-MIB	ATMClientApp
CA Spectrum Core Product	ATM-MIB ATM2-MIB ATM-FORUM-MIB	ATMSwitchApp
Bay Networks Centillion 100 (SM-BAY1001)	CENTILLION-ATMCFG-MIB	CentATMApp
Cisco Catalyst 85xx (SM-CAT1008)	ACCOUNTING-CONTROL-MIB ATM-MIB ATM-FORUM-MIB ATM-RMON-MIB CISCO-ATM-ACCESS-LIST-MIB CISCO-ATM-CONN-MIB CISCO-ATM-IF-MIB CISCO-ATM-IF-PHYS-MIB CISCO-ATM-RM-MIB CISCO-ATM-SERVICE-REGISTRY-MIB CISCO-ATM-SWITCH-ADDR-MIB CISCO-ATM-TRAFFIC-MIB CISCO-PNNI-MIB	CiscoSwitchApp
Cisco LightStream 1010 (SM-CIS1002)	ACCOUNTING-CONTROL-MIB ATM-MIB ATM-FORUM-MIB ATM-RMON-MIB CISCO-ATM-ACCESS-LIST-MIB CISCO-ATM-CONN-MIB CISCO-ATM-IF-MIB CISCO-ATM-IF-PHYS-MIB CISCO-ATM-RM-MIB CISCO-ATM-SERVICE-REGISTRY-MIB CISCO-ATM-SWITCH-ADDR-MIB CISCO-ATM-TRAFFIC-MIB CISCO-PNNI-MIB	LS_Switch_App
Cisco Router (CIS-1000)	CISCO-AAL5-MIB	CiscoAAL5App
Cisco Stratacom BPX8600 and IGX8400 Series (SM-CIS1003)	STRATACOM-MIB	StComATMSwApp
ForeRunner ATM Switch Modules (SM-FOR1000)	Fore-Switch-MIB	ForeSwitchApp
Lucent Ascend CBX (SM-LUC1002)	CASCADE-MIB	CascadeATMApp
Nortel Passport Multiservice Carrier Switch Series (SM-NTL1005)	Nortel-MsCarrier-MscPassport-AtmCoreMIB	PpAtmClientApp
SmartSwitch 9000/9500 Series (SM-CSI1073)	ATM-MIB ATM2-MIB ATM-FORUM-MIB	PredSwitchApp

Specify Virtual Links to Model for a Device

DX NetOps Spectrum models the virtual links (VPLs and VCLs) on ATM devices by periodically reading the MIBs on the devices to determine the virtual links that currently exist. By default, ATM links (VPLs and VCLs) are modeled for ATM clients, and, for performance reasons, they are not modeled for ATM switches.

An exception to this default behavior is when a physical interface of an ATM switch is connected to an ATM_Cloud model. In this situation, the virtual links associated with that interface are modeled. These virtual links are necessary to resolve logical connectivity across the ATM_Cloud.

You can change the configuration of an ATM device so that only VPLs are modeled, both VPLs and VCLs are modeled, or so that no virtual links are modeled at all. The last option is especially useful if the virtual links for a device are dynamic (such as the case when SVCs are in use) and modeling them holds no inherent value.

NOTE
If a particular ATM device has a very large number of VCLs (in the order of thousands), modeling and reconfiguration of the virtual link models can impact SpectroSERVER performance.

To specify the virtual interfaces to model on an ATM device

1. Use the Navigation panel or the Topology tab to locate and select the device.
   Information about the device is displayed in the Component Detail panel.
2. Click the Information tab in the Component Detail panel, expand Reconfiguration, and expand ATM Link Modeling Options and Reconfiguration.
3. Specify whether to model ATM links (virtual interfaces) for the device and, if so, whether to model only VPLs or both VPLs and VCLs:
   – To enable the modeling of ATM links, click set in the 'Create models to represent ATM links' field and select Enabled. Alternatively, if you do not want to model any ATM links, select Disabled instead. (By default, this option is enabled for ATM Client models and disabled for ATM Switch models.)

     NOTE
     If the ATM links for a device are currently modeled, and you disable this option, all existing ATM link models for the device will be deleted after you click 'Reconfigure ATM Link and Virtual Interface Models' in the next step.
   – To enable the modeling of virtual channel links (VCLs), click set in the 'Create models to represent VCLs' field and select Enabled. Alternatively, if you do not want to model VCLs, select Disabled instead.

     NOTE
     This option only controls the modeling of VCLs. If 'Create models to represent ATM links' is set to Disabled, this setting has no effect.
4. Click Reconfigure ATM Link and Virtual Interface Models.
   DX NetOps Spectrum reads the MIBs on the devices to determine the virtual links that currently exist and then creates and destroys the virtual link models according to the settings you specified.

Specify Frequency for Updating Virtual Link Models

DX NetOps Spectrum models the virtual links (VPLs and VCLs) on ATM devices by periodically reading the MIBs on the devices to determine the virtual links that currently exist and then updates the virtual link models accordingly. If desired, you can change the interval at which this read operation is performed.

To specify the frequency for updating virtual link models

1. Use the Navigation panel or the Topology tab to locate and select the appropriate device.
   Information about the device is displayed in the Component Detail panel.
2. Click the Information tab in the Component Detail panel, expand Reconfiguration, and expand ATM Link Modeling Options and Reconfiguration.
3. Click set in the Configuration Discovery Interval (sec) field, type a new time interval in seconds, and press Enter.

Logical Connections Links between Virtual Link Models

Virtual link models can have Links_with associations with other virtual link models to indicate logical connections. A *managed virtual circuit* is a circuit whose virtual link models are associated by a logical connection. ATM circuits run from one client, through the ATM_Network, to another client, or directly between ATM switches within the switched fabric. By default, none of the circuits through the switched fabric are managed.

You can create logical connections (*Links_with* associations) between virtual link models in the following ways:

- Use Discovery (Cisco devices only)
- Manually create the connections
- Import a file that defines the connections to create

DX NetOps Spectrum uses the logical connection information during the fault isolation process.

Logical Connections Links for Cisco Devices Using Discovery

When you run Discovery to model your ATM network, if the VCL interfaces on the Cisco ATM devices have unique IP addresses, DX NetOps Spectrum resolves the connections between these interfaces. If there are connections between Cisco VCL interfaces and Frame Relay DLCI interfaces, DX NetOps Spectrum also resolves these connections.

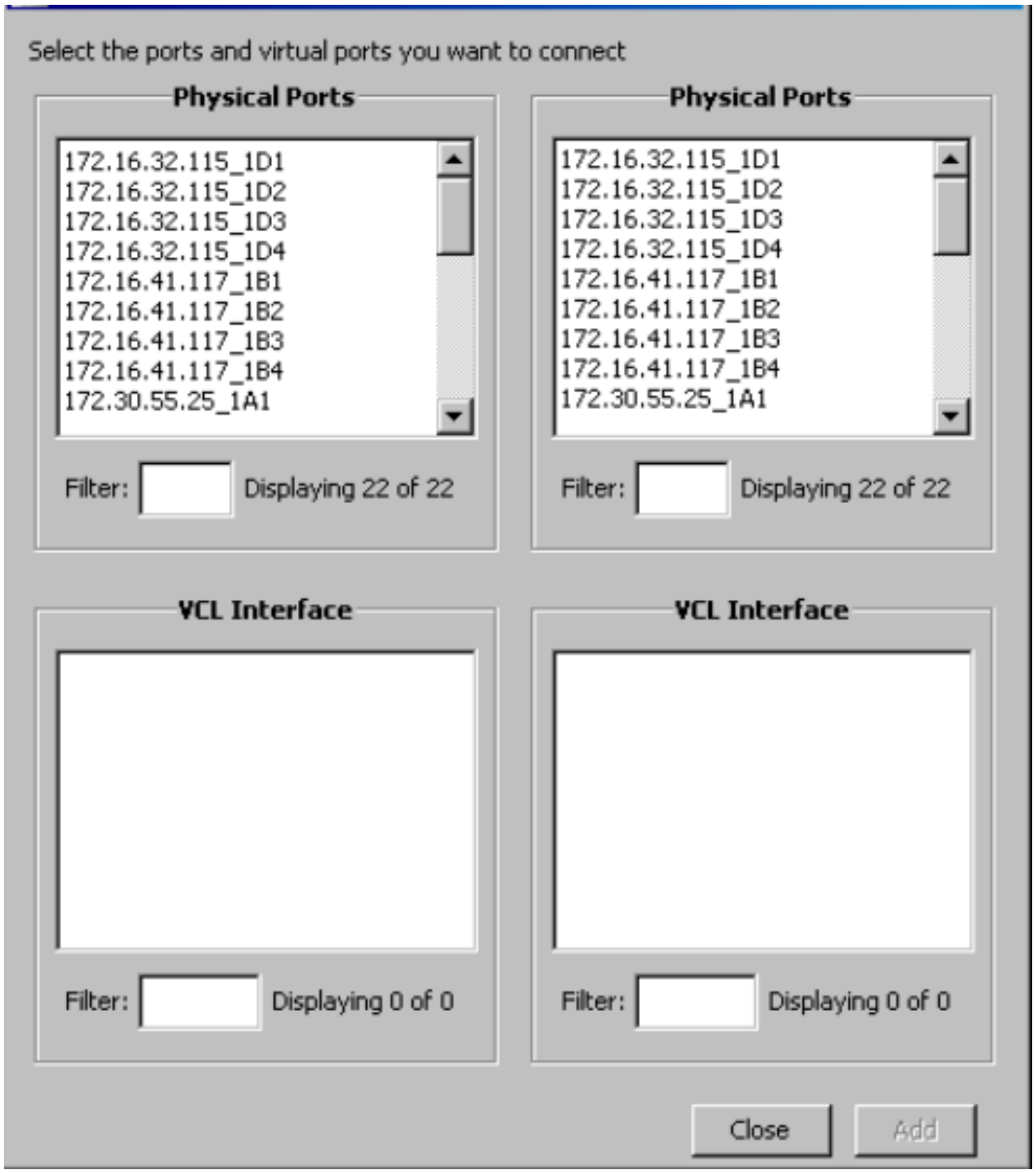
NOTE
For information on using Discovery, see the Modeling and Managing Your IT Infrastructure section.

Create Logical Connections Manually

NOTE
You can include a virtual link in only one logical connection.

To manually create logical connections between virtual link endpoints

1. Use the Navigation panel or the Topology tab to locate and select the ATM_Cloud or ATM_Network model that is connected to or contains the relevant device and interface models.
2. Click the Information tab in the Component Detail panel, expand Logical Connection Table, and click Add. The Logical Connection Manager dialog opens.



The dialog lists the physical ports on the ATM devices connected to the selected cloud, or connected to or contained within the selected network. Only the ports that currently have virtual link models are included in the lists.

3. Select one of the physical ports that contains the virtual link to connect in the left Physical Ports box.
   The virtual interfaces for the selected port are displayed in the left VCL Interface box.
4. Select the virtual link that represents the first endpoint of the logical connection in the left VCL Interface box.
5. Repeat the Step 3 and Step 4 using the right-side boxes to locate and select the virtual interface that represents the second endpoint in the logical connection, and click Add.
   A *Links_with* association is added between the virtual link models, and the logical connection is displayed in the Logical Connection Table.
6. Repeat the preceding steps to create additional logical connections as needed, and click Close.

Create Logical Connections to Unmanaged Models

It is possible for a DX NetOps Spectrum user to be responsible for a circuit but not have SNMP contact with the device on one side. If this is the case, you can still create a logical connection that includes the unmanaged link model and manage the circuit from the other endpoint.

When you create a connection to an unmanaged link model, ATM Circuit Manager creates a model of type UnmanAtmLink and associates it with the ATM_Network or ATM_Cloud model using a *Contains* relation. As with all logical connections, a *Links_with* association between the managed and unmanaged virtual link models is created.

To create a logical connection to an unmanaged link model

1. Use the Navigation panel or the Topology tab to locate and select the ATM_Cloud or ATM_Network model that contains or is connected to the relevant device and interface models.
2. Click the Information tab, expand the Logical Connection Table, and click Add.
   The Logical Connection Manager dialog opens.
3. Under Physical Ports on the left side of the dialog, select the physical port that contains the managed virtual link.
4. Under VCL Interface on the left side of the dialog, select the managed virtual link.
   This virtual link will be the single point of management for the connection.
5. Under Physical Ports on the right side of the dialog, select a physical port on the unmanaged device.
6. Under VCL Interface on the right side of the dialog, select Unmanaged Link, and click Add.
7. Enter the following information about the unmanaged endpoint:
   - **Name**
     The name of the virtual link model. For convenience, the name of the associated ATM_Network container model is used as the default name, so you can easily use it as a prefix for the name you specify.
   - **Component OID**
     The component OID of the unmanaged endpoint (Port_ID.VPL.VCL).
   - **Network Address**
     The network address of the unmanaged interface (if there is one assigned).
   - **Circuit Name**
     The name of the circuit. This can be any name that has meaning, such as “London to Paris link.”
   - **Circuit ID**
     The ID of the circuit. This can be any ID that has meaning, such as “Leased Circuit 119.”
8. Click OK.
   The unmanaged link model is created, and the logical connection between it and the managed link model is also created.

Logical Connections Link in ATM and Frame Relay Models

If you have a mixture of ATM and Frame Relay interfaces on either side of a leased service provider's network, you can create logical connections between their virtual interfaces (VPLs, VPTs, VCLs, and DLCIs) using *Links_with* associations. The procedure for modeling these types of connections is the same as that for ATM-to-ATM logical connections with two exceptions:

- When following the procedure outlined in Connect Physical Interfaces to Service Provider's Network , the physical interface selected in Step 1 will be a Frame Relay interface, such as a serial port.
- When following the procedure outlined in Creating Logical Connections Manually, the virtual interface selected as an endpoint in Step 3 or Step 4 will be a DLCI virtual interface.

It is also possible to use a file to import a list of ATM-to-Frame Relay connections. To do this, you can use the import process described in Importing Logical Connections, but you must access the Logical Connection Import subview that is available on the Information tab of the *VNM model*, not the Logical Connection Table subview that is available on the ATM_Network or ATM_Cloud model.

NOTE
For more information about Frame Relay, see the Standards-Based Protocol Reference section.

Import Logical Connections

You can create logical connections between virtual link models by importing a comma-delimited, ASCII file (text file or XML file) that defines the connections. You can define connections that include two ATM models or an ATM model and a Frame Relay model.

The file in which you define the logical connections must be a comma-delimited, ASCII file (text file or XML file). Use the following syntax to define a connection:

```
<device_IP_1>,<subinterface_OID_1>,<device_IP_2>,<subinterface_OID_2>,
<circuit_name>,<circuit_ID>,<Pipe>
```

The parameters are as follows:

- **device_IP_1**
  Specifies the IP address of the first device in the connection.
- **subinterface_OID_1**
  Specifies the component OID of the first virtual interface in the connection.
- **device_IP_2**
  Specifies the IP address of the second device in the connection.
- **subinterface_OID_2**
  Specifies the component OID of the second virtual interface in the connection.
- **circuit_name**
  (Optional) Specifies the name of the circuit. This can be any name that has meaning.
- **circuit_ID**
  (Optional) Specifies the ID of the circuit. This can be any ID that has meaning.
- **pipe**
  (Optional) Specifies CREATE_PIPE to create a live pipe to graphically represent the connection. Otherwise, specify NO_CREATE_PIPE.

To import the logical connections defined in a file

1. In OneClick, use the Navigation window or the Topology view to select one of the following:
   – The ATM_Network that contains or has connections to the physical interfaces of the virtual link models for which you want to define connections
   – The ATM_Cloud that has connections to the physical interfaces of the virtual link models for you which you want to define connections
2. In the Component Detail window, click the Information tab and expand Logical Connection Table.
3. Click the
   icon 
   This icon helps import a file describing links between virtual interfaces.
4. In the Open dialog, navigate to the file that contains the connection definitions, select the file, and click Open.
   The connection definitions are imported, and the logical connections are created. If the import process fails, you are notified in the Import Results dialog that describes the results of the operation.
5. In the Import Results dialog, click OK.

Note: You can find additional information about the import operation (and previous ones) on the Information tab under the Modeling Gateway subview for the VNM model. This view displays the name of the file that was imported, the location of the log file that was created for the operation, and other information.

Create Pipes Between Virtual Link Models

In some cases, you might want to create pipes (resolved connections) to represent important logical connections between virtual link models. This allows you to monitor the status of a circuit based on the pipe's color in the Topology view.

NOTE
For an introduction to pipes, see the Modeling and Managing Your IT Infrastructure section.

To create a pipe between virtual link models

1. Use the Navigation window or the Topology view to locate the device that contains one of the virtual link models, and select the device.
   Information about the device is displayed in the Component Detail window.
2. In the Component Detail window, click the Interfaces tab.
   The physical and virtual interfaces for the device are displayed on the tab.
3. Expand the physical interfaces to view the associated link models. You can right-click the interface to get the Expand All or Collapse All option for your IP address.
4. Select one of the virtual link models you want to connect with a pipe, right-click, and click Start Connection.
5. Select the other virtual link model to connect with the pipe, right-click, and click Connect With *<name of virtual link model>*, where *<name of virtual link model>* is the name of the interface model you selected in the previous step.

NOTE
You can also create pipes when you import a set of logical connections.

Monitoring and Managing the ATM Network

DX NetOps Spectrum lets you set up thresholds, monitor network performance, and diagnose several common network problems.

The management of PVPs and PVCs on ATM switches and ATM clients consists of:

NOTE

If you are monitoring Cisco devices with ATM Circuit Manager, and you are making use of DX NetOps Spectrum's Live Pipes feature or the PortPollStatus attribute to monitor connectivity, it is recommended that you use Cisco's Operation, Administration, and Maintenance (OAM) feature. Turning on this feature helps ensure that you can detect communication problems on a permanent virtual connection (PVC) when network connectivity is lost but the PVC remains up on the end devices. In this situation, if OAM is not configured, and a CiscoATMVclLnk goes down, an alarm is not generated. For more information, see the ATM technical tips on the Cisco web site.

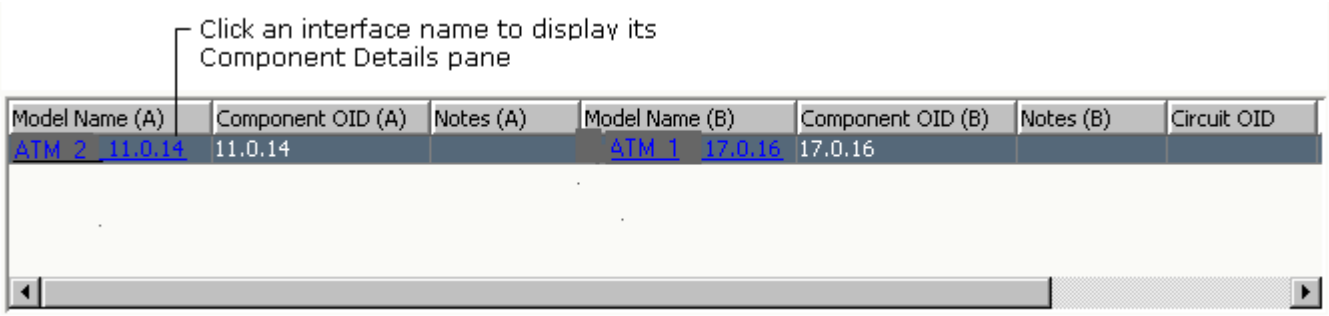
Monitor the Logical Connections Between Virtual Link Models

As you monitor your ATM network, you need to view the logical connections between virtual link models, for example, so you can determine the status of a particular circuit represented by a connection.

To view the logical connections between virtual link models

1. Use the Navigation panel or the Topology tab to locate and select the ATM_Cloud or ATM_Network model that the logical connection you want to view traverses or is a part of.
2. Click the Information tab and expand Logical Connection Table in the Component Detail panel.

The Logical Connection Table opens.



The connection table provides information about each connection; that is, the endpoint models and the logical connections they collectively represent. You can view a circuit's status by moving the scroll bar to the right.

3. To export the list of connections to a CSV, TXT, or HTML file, click

(Export).

4. To delete one or more connections, select them in the table and click

(Delete).

5. To display additional information in the Logical Connection Table, right-click the table heading, select the desired columns in the Table Preferences dialog, and click OK.

Monitor Virtual Link Models

As you monitor your ATM network, you need to view a device's virtual link models, for example, to determine their condition or status.

To view the virtual link models on an ATM device

1. Use the Navigation panel or the Topology tab to locate and select the device.
   Information about the device is displayed in the Component Detail panel.
2. Click the Interfaces tab in the Component Detail panel.
   The physical and virtual interfaces for the device are displayed on the tab.
3. Expand the physical interfaces to view its associated virtual link models. You can right-click the interface to get the Expand All or Collapse All option for your IP address.
   The interface table identifies the condition, status, and general type of the interface (for example, VPL or VCL), among other information. If the interface is used in a logical connection, the device and port at the other end of the connection are displayed, respectively, in the Device Connected and Port Connected fields.

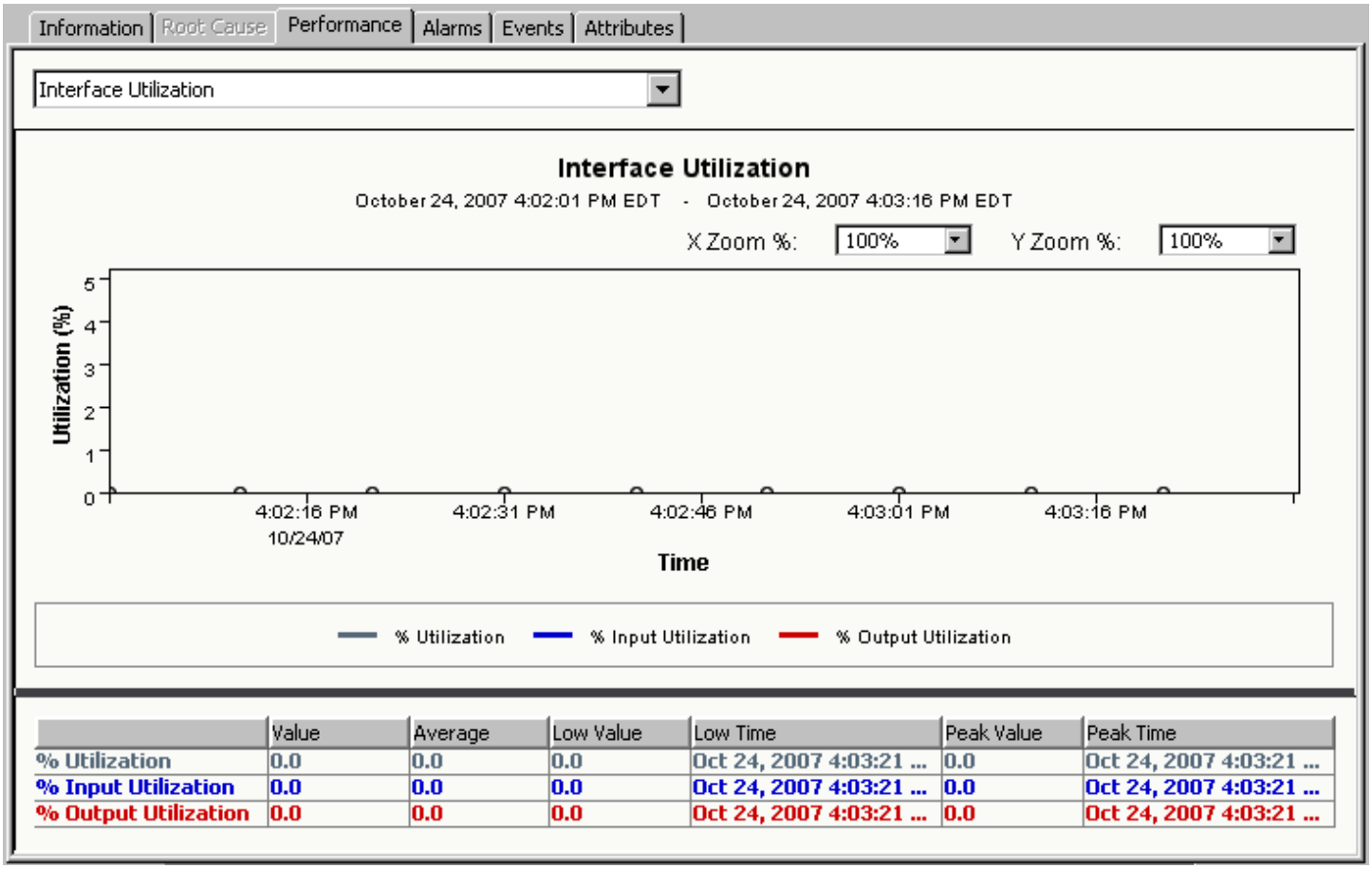
   NOTE
   You can set the interface IP address as the primary IP address for DX NetOps Spectrum management. To do this, right-click the interface, select Configure Primary Address for Device, and select Use Interface IP as Primary Address.

NOTE
To display additional information in the interface table, right-click the table heading, select the desired columns in the Table Preferences dialog, and click OK.

View Virtual Link Models Performance

You can use the Performance view of a virtual link model to access the following:

- The percentage of the virtual link's utilization for input and output.
- The throughput for input and output using the virtual link.



	Value	Average	Low Value	Low Time	Peak Value	Peak Time
% Utilization	0.0	0.0	0.0	Oct 24, 2007 4:03:21 ...	0.0	Oct 24, 2007 4:03:21 ...
% Input Utilization	0.0	0.0	0.0	Oct 24, 2007 4:03:21 ...	0.0	Oct 24, 2007 4:03:21 ...
% Output Utilization	0.0	0.0	0.0	Oct 24, 2007 4:03:21 ...	0.0	Oct 24, 2007 4:03:21 ...

The current values are the last known values when the associated ATM device was polled. The other values are the average, low, and peak values since the Performance tab was opened.

The interface utilization calculations use the following statistical fields that indicate load:

- For input utilization, the value of the *rcvLoad* attribute
- For output utilization, the value of the *xmtLoad* attribute

The formulae for these attributes are as follows:

```
rcvLoad = rcvCellsPerSecond * 100 / rcvBandwidth
xmtLoad = xmtCellsPerSecond * 100 / xmtBandwidth
```

The *rcvBandwidth* and *xmtBandwidth* attributes are defined by either the Peak Cell Rate (PCR) or Sustainable Cell Rate (SCR) depending on the Quality of Service (QoS) type. For Variable Bit Rate (VBR) circuits, the bandwidth is defined as the SCR. For all other types of service, the bandwidth is defined as the PCR. This means that the load can exceed 100% for VBR circuits.

To always use the Peak Cell Rate (PCR) in bandwidth calculations-even for Variable Bit Rate (VBR) circuits-enable the corresponding setting in the VCL QoS Information view.

The *rcvCellsPerSecond* attribute is calculated by reading the attribute pointed to by *rcvCells_Attr* and the *upTime* attribute over a particular interval, and then subtracting the first values from the second values. This yields a delta of received cells and a delta of elapsed microseconds. By dividing the delta of cells by the delta of microseconds, and then multiplying by 100, the *rcvCellsPerSecond* value is determined.

You can log the values of the *rcvLoad* and *xmtLoad* attributes for historical reports.

NOTE
To use this Performance view, the ATM device must support the ATM2 MIB or one of the supported proprietary MIB extensions. Performance information is not available without the cell counters inherent in this MIB.

To view the virtual link model performance

1. Use the Navigation panel or the Topology tab to locate and select the device that has the virtual link model. Information about the device is displayed in the Component Detail panel.
2. Click the Interfaces tab in the Component Detail panel.
   The physical and virtual interfaces for the device are displayed on the tab.
3. Expand the physical interfaces to view the associated link models.

   NOTE
   You can right-click to get the expand all and collapse all option for your IP address.
4. Select the link model for which you want to view performance information, and click

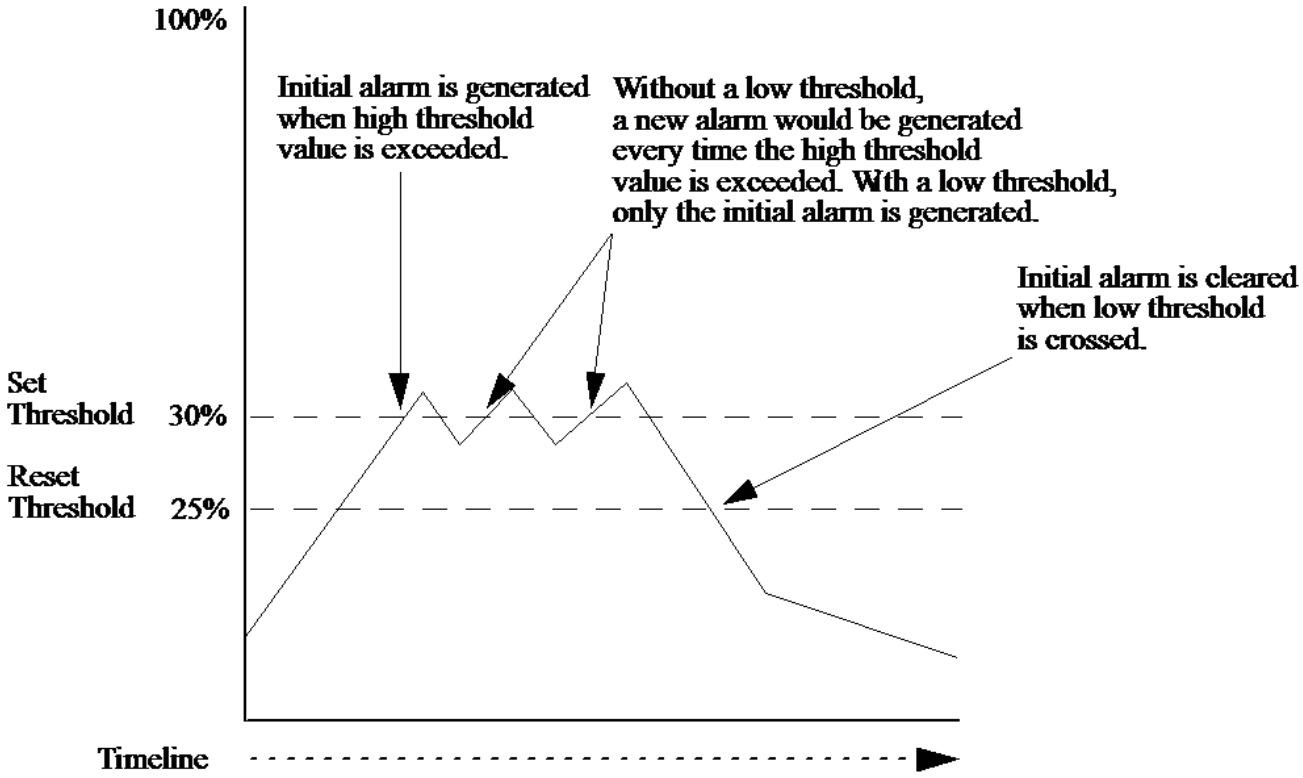
   (View the Component Detail for the selected model).
5. Click the Performance tab in the Component Detail panel.
   The performance information for the selected virtual link model is displayed.

Set Traffic Thresholds for Virtual Link Models

You can establish the levels of activity for virtual link models that will generate alarms. You do this by setting high and low thresholds for the following attributes:

- **Receive Load**
  The average number of bits received by the virtual link model since the last poll.
- **Transmit Load**
  The average number of bits transmitted by the virtual link model since the last poll.

Each high and low threshold has a “set” value and a “reset” value. For the high thresholds, the “set” value is the value that, if exceeded, generates an alarm for the attribute. The “reset” value is the value that, if gone below, automatically clears the alarm for the attribute. The “set” and “reset” values for the low thresholds work in the converse manner. The following illustration further describes the behavior.



By default, the high thresholds are set to 90% and reset at 80%, and the low thresholds are set to 0 and reset at 0. In other words, the low thresholds are disabled. The threshold values are recalculated at every poll cycle and represent the average number per poll.

WARNING
Do not set the “reset” field to “0.” If the “reset” field is set to “0” and the “set” field is crossed, the subsequent alarm is never cleared automatically. If the alarm is cleared manually by the user, the DX NetOps Spectrum threshold intelligence is not reset, and the alarm for that model is not generated again. To reset the threshold intelligence, the SpectroSERVER must be restarted.

NOTE
To use this Threshold view, the ATM device must support the ATM2 MIB or one of the supported proprietary MIB extensions. Thresholds cannot be set for a circuit without the cell counters inherent in this MIB.

To set traffic thresholds for a virtual link model

1. Select the device that has the virtual link model you want to configure from the Navigation panel or the Topology tab. Information about the device is displayed in the Component Detail panel.
2. Click the Interfaces tab in the Component Detail panel.
   The physical and virtual interfaces for the device are displayed on the tab.
3. Expand the physical interfaces to view the associated link models.

   NOTE
   You can right-click the interface to get the Expand All or Collapse All option for your IP address.
4. Select the VPL or VCL model that you want to configure, and click

   (View the Component Detail for the selected model).

5. Click the Information tab and expand VCL Threshold Information in the Component Detail panel.
6. Specify a value for any of the thresholds by clicking set, entering the value, and pressing Enter.

Specify Service Information for Virtual Link Models

For reference purposes, you can add information about the service provider for individual virtual link models.

To specify service information for a virtual link model

1. Use the Navigation panel or the Topology tab to locate and select the device that has the virtual link model you want to configure.
   Information about the device is displayed in the Component Detail panel.
2. Click the Interfaces tab.
   The physical and virtual interfaces for the device are displayed on the tab.
3. Expand the physical interfaces to view the associated link models.

   NOTE
   You can right-click the interface to get the Expand All or Collapse All option for your IP address.
4. Select the link model that you want to configure, and click

   (View the Component Detail for the selected model).
5. Click the Information tab and expand VPL/VCL Service Information in the Component Detail panel.
6. Enter the service information by clicking 'set,' entering the text, and pressing Enter.
   You can enter the following information:
   - **Provider**
     Specifies the name of the service provider associated with the ATM network.

     NOTE
     ATM customers who use multiple carriers can use this field to indicate the carriers that provide service for specific circuits.
   - **Customer**
     Users who are service providers and manage other companies' ATM networks can use this field to indicate the customer.
   - **Primary Contact**
     Specifies the name, phone number, and email address of the person to contact if there is a problem with the circuit.
   - **Secondary Contact**
     Specifies the name, phone number, and email address of a secondary person to contact if there is a problem with the circuit.
   - **Service Notes**
     Miscellaneous information about the circuit, such as the circuit ID or the monthly cost.

Specify Service Information for unmanATMLink model

NOTE
For reference purposes, you can add information about the service provider for unmanaged virtual link models.

To specify service information for an unmanATMLink model

1. Use the Navigation panel or the Topology tab to locate and select the ATM_Cloud or ATM_Network model where the unmanATMLink in question is defined as an endpoint of a logical connection.
2. Click the Information tab and expand Logical Connection Table in the Component Detail panel.
   The Logical Connection Table opens.

3. Locate the relevant row in the table and click the name of the unmanATMLink to launch the Component Detail for the model.
4. Click the Information tab and expand VPL/VCL Service Information in the Component Detail panel.
5. Enter the service information by clicking 'set,' entering the text, and pressing Enter.
   You can enter the following information:
   - **Provider**
     Specifies the name of the service provider associated with the ATM network.

     NOTE
     ATM customers who use multiple carriers can use this field to indicate the carriers that provide service for specific circuits.
   - **Customer**
     Users who are service providers and manage other companies' ATM networks can use this field to indicate the customer.
   - **Primary Contact**
     Specifies the name, phone number, and email address of the person to contact if there is a problem with the circuit.
   - **Secondary Contact**
     Specifies the name, phone number, and email address of a secondary person to contact if there is a problem with the circuit.
   - **Service Notes**
     Miscellaneous information about the circuit, such as the circuit ID or the monthly cost.

Monitor Quality of Service (QoS) Information for VCLs

If you have configured your network for Quality of Service (QoS), you can use ATM Circuit Manager to view the following QoS information for VCL link models:

- The QoS class of the connection.
- Performance information for the connection, for example, the peak cell rate when receiving and transmitting data.
- The bandwidth used to receive and transmit data on the connection, as well as the parameters used to calculate the bandwidth.

NOTE
For information on how to configure, discover, and manage the QoS elements of your network using DX NetOps Spectrum QoS Manager, see the QoS Manager section.

To monitor QoS information for a VCL

1. Use the Navigation panel or the Topology tab to locate and select the device that has the VCL model that you want to examine.
   Information about the device is displayed in the Component Detail panel.
2. Click the Interfaces tab.

   NOTE
   You can right-click the interface to get the Expand All or Collapse All option for your IP address.

   The physical and virtual interfaces for the device are displayed on the tab.
3. Expand the physical interfaces to view the associated link models.
4. Select the VCL model that you want to examine, and click

   (View the Component Detail for the selected model).
5. Click the Information tab and expand VCL QoS Information.
   The QoS information for the selected VCL model is displayed.

6. (Optional) Click set in the Use Peak Cell Rate (PCR) for bandwidth calculation field and select Enabled to specify that the peak cell rate (PCR) of the connection is always used in bandwidth calculations.

   NOTE
   Alternatively, you can disable this setting if you want the parameters used in bandwidth calculations to be determined based on the QoS class.

Receive and Transmit QoS Parameters

The VCL QoS Information view displays information on the following receive and transmit parameters:

- **QoS Class**
  The QoS class used for the connection.
- **QoS Peak Cell Rate**
  The maximum number of cells that the connection can receive or transmit per second on the network.
- **QoS Sustained Cell Rate**
  The average number of cells that the connection can receive or transmit per second on the network.
- **QoS Max Burst Size**
  The maximum allowable burst size (in cells) that can be transmitted contiguously at the peak cell rate over this link.
- **QoS Tagging**
  If On (enabled), the Cell Loss Priority (CLP) bit of cells is marked (tagged) because the cells do not confirm to the subscribed QoS contract. Tagged cells have a lower priority than other cells, and they are the first cells to be dropped by the network when traffic is congested.
- **QoS CLPO Peak Cell Rate**
  The maximum number of cells with the CLP bit set that the connection can receive or transmit per second on the network.
- **QoS CLPO Sustained Cell Rate**
  The average number of cells with the CLP bit set that the connection can receive or transmit per second on the network.
- **QoS CLPO Max Burst Size:**
  The maximum allowable burst size (in cells) that can be transmitted contiguously with the CLP bit set at the CLP peak cell rate.

Bandwidth Parameters

The VCL QoS Information view displays the maximum bandwidth that can be used by the connection to receive and transmit data.

Receive Bandwidth Calculations

The *Receive Bandwidth value* is the maximum number of bits per second that can be received by the connection.

By default, if the connection's QoS class is Variable Bit Rate, the Receive Bandwidth value is calculated using the Receive - QoS Sustained Cell Rate as follows:

```
Receive Bandwidth (bits per second) = (Sustained Cell Rate (cells per second) * 53 (bytes per cell)) * 8 (bits per byte)
```

If the connection's QoS class is any other class, the Receive Bandwidth value is calculated using the Receive - QoS Peak Cell Rate as follows:

```
Receive Bandwidth (bits per second) = (Peak Cell Rate (cells per second) * 53 (bytes per cell)) * 8 (bits per byte)
Transmit Bandwidth Calculations
```

Transmit Bandwidth Calculations

The *Transmit Bandwidth value* is the maximum number of bits per second that can be transmitted by the connection.

By default, if the connection's QoS class is Variable Bit Rate, the Transmit Bandwidth value is calculated using the Transmit - QoS Sustained Cell Rate as follows:

```
Transmit Bandwidth (bits per second) = (Sustained Cell Rate (cells per second) * 53 (bytes per cell)) * 8
 (bits per byte)
```

If the connection's QoS class is any other class, the Transmit Bandwidth value is calculated using the Transmit - QoS Peak Cell Rate as follows:

```
Transmit Bandwidth (bits per second) = (Peak Cell Rate (cells per second) * 53 (bytes per cell)) * 8 (bits per
 byte)
```

Using the Peak Cell Rate (PCR) in Bandwidth Calculation

The QoS Information view includes a Use Peak Cell Rate (PCR) for bandwidth calculations setting that you can use to specify how bandwidth calculations are performed.

By default, the setting is disabled, which means that the values for Receive Bandwidth and Transmit Bandwidth are calculated as described previously in this topic. However, if you enable this setting, the bandwidth values are calculated as follows:

- If the receive and transmit values for QoS Peak Cell Rate are specified, the Peak Cell Rate is always used to calculate the bandwidth.
- If the receive and transmit values for QoS Peak Cell Rate and QoS Sustained Cell Rate are not specified, the bandwidth is set to the value of If Speed.

Graphical Representations of ATM Interface Connections

In OneClick, in the Topology tab of your ATM network, you can click any pipe (connection) to view a graphical, port-to-port representation of the connection in the Component Detail panel.

This is helpful when you need to quickly identify the physical interfaces involved in the connection represented by a pipe.

If you have created logical connections between virtual link models and created pipes during the import process, or if you have manually created resolved connections between virtual link models (which automatically creates pipes), this view can also show the logical connections represented by a pipe.

NOTE

In this view, you can click any interface model to display the Interfaces tab for the associated device in the Component Detail panel.

Managing Faults

Once you have created an accurate model of your ATM network, ATM Circuit Manager can detect faults on the ATM circuits, isolate the root cause of an outage or service degradation, and alert the user to a problem using an alarm.

DX NetOps Spectrum uses three different methods to monitor virtual interface models and manage faults across the switched fabric in an ATM environment:

- For devices that support the standard ATM MIB, the status of each virtual link model is monitored using the following MIB objects:
  - atmVclAdminStatus and atmVclOperStatus (for VCL models)
  - atmVplAdminStatus and atmVplOperStatus (for VPL models)

  If a model is determined to be inactive, an alarm is generated.
- You can set high and low traffic thresholds for VPLs and VCLs so that an alarm is generated if the load on a particular circuit falls above or below what what you expect.
- If one endpoint of an ATM circuit is a Cisco router, by default DX NetOps Spectrum periodically initiates a “remote ping” from the router to the IP address on the other side of the circuit. If the ping fails, an alarm is generated.

Fault Management Using Link Status

For devices that support the standard ATM MIB, DX NetOps Spectrum reads the Internal_Link_Status attribute of each virtual link model every polling interval. This attribute is calculated based on the values of the following administrative and operational status objects:

- atmVclAdminStatus and atmVclOperStatus (for VCL models)
- atmVplAdminStatus and atmVplOperStatus (for VPL models)

If the value of the Internal_Link_Status attribute is not active, it is assumed that an error condition has occurred, and an attempt is made to isolate the problem to the model itself or to its parent model. This is done by reading the Internal_Link_Status attribute of the parent model, which could be a trunk or a physical interface.

If the parent model is active, a red alarm is asserted on the inactive model to indicate that the problem has been isolated. If the parent model is *not* active, a gray alarm is asserted on the inactive model to indicate a a suppressed condition, and the fault isolation process continues until the source of the problem is identified.

Manage Faults Using Threshold Alarms

ATM virtual link models have an ATM Threshold view that allows you to specify the levels of activity that generate alarms. This can be useful if a specific amount of traffic on an ATM model is the norm, and you want DX NetOps Spectrum to generate an alarm if less or more traffic occurs (because this indicates a problem exists).

For this functionality to be active, you need to set the virtual link model's PollingStatus attribute to Yes (for TRUE). This attribute is set to No by default to limit management traffic over ATM links.

To set the PollingStatus attribute for a virtual link model

1. Use the Navigation panel or the Topology tab to locate and select the device that has the virtual link model you want to configure.
   Information about the device is displayed in the Component Detail panel.

2. Click the Interfaces tab in the Component Detail panel.
   The physical and virtual interfaces for the device are displayed on the tab.
3. Expand the physical interfaces to view the associated link models.
4. Select the virtual link model that you want to configure, and click
   
   (View the Component Detail for the selected model).
5. Click the Attributes tab in the Component Detail window.
6. Set the PollingStatus attribute to Yes.

> **NOTE**
> For information on using the Attributes tab to change attribute values, see the Modeling and Managing Your IT Infrastructure section.

Fault Management on Cisco Routers

If you have Cisco routers in your ATM network, ATM Circuit Manager uses the CiscoPingApp application model to initiate remote pings to determine the status of ATM PVCs. An inference handler examines the ATM connections for a particular router, and then it instructs the router to ping the IP addresses of the routers on the other side of the ATM PVC. If a ping fails, an event with an event code of 0x02dc0001 is sent to the ATM virtual link model that represents this router's side of the ATM PVC. This generates a red alarm on the model; the alarm includes the probable cause information shown below.

```
REMOTE PING FAILURE MAY INDICATE A PVC FAILURE

SYMPTOMS:

The <ss> initiated a remote ping from one router to another over an ATM PVC. Not all ICMP echos were received
 back by this router.

PROBABLE CAUSES:

The PVC connecting this router to the IP address that was pinged may be down.

RECOMMENDED ACTIONS:

1) Check the Event tab to see what IP address was pinged.
2) Verify that all PVCs on this device are operating normally.
```

If desired, you can change the severity of this alarm or prevent it from being generated using the Event Configuration application.

> **NOTE**
> For information on using the Event Configuration application, see the Event Configuration section.

> **NOTE**
> If the ATM network has redundant paths set up or the OSPF (Open Shortest Path First) routing protocol is being used, remote pinging may return information that is not completely useful in determining the health of the network and, therefore, will use bandwidth unnecessarily.

Also note that the ATM Circuit Manager only initiates 5 remote pings per router at a time until all of the remote pings have been tried. This prevents the possibility of the router becoming overloaded.

Configure CiscoPingApp Application Model

You can configure the following attributes of a CiscoPingApp application model:

- **CommunityNameForSNMPSets**
  The community name that is used when performing SNMP sets. By default, this value is inherited from the CommunityNameForSNMPSets attribute on the device model.
  In order for DX NetOps Spectrum to initiate remote pings from a Cisco router, DX NetOps Spectrum must perform SNMP sets. To allow sets to be performed, the CiscoPingApp application model must have a community name that allows DX NetOps Spectrum to write to the device's MIB (in other words, a community name with read and write capabilities).
  If the value of this attribute for the application model differs from the value for the device model, the value for the application model takes precedence.
  If this attribute is unset on both the device model and the application model, DX NetOps Spectrum uses the value of the Community_Name attribute. Thus, if the value of the Community_Name attribute is a name with read and write capabilities, and the CommunityNameForSnmpSets attribute is unset, remote ping works properly.

  NOTE
  For more information on these attributes, see the Getting Started section.
- **EnableRemotePings**
  Enables and disables remote pinging. If set to No (for FALSE), remote pinging from the router is disabled. The default value is Yes (for TRUE).
- **NumberOfPingPackets**
  The number of ping packets the router will send to the remote IP address.
  Default: 3
- **PingFailuresAllowed**
  The number of ping request failures allowed before an alarm is generated.
  Default: 2
- **PingInterval**
  The interval in seconds between remote ping requests.
  Default: 300
- **PingPacketSize**
  The size in bytes of the ping packet that the router will send to the remote IP address.
  Default: 128

  NOTE
  NumberOfPingPackets, PingFailuresAllowed, and PingInterval, collectively, are used to increase the frequency of remote pinging and the speed of any network fault detection. For example, if PingInterval were lowered to 60, NumberOfPingPackets remained set at 3, and PingFailuresAllowed were decreased to 0, the ping requests would be initiated more frequently, and no failure of any of these requests would be allowed. This would result in the ATM network being more closely monitored for remote link problems, and, if problems were discovered, alarms would be generated more quickly.

To configure one or more CiscoPingApp application models

1. Use the Locater tab to find all models of model type CiscoPingApp.
   The CiscoPingApp models are listed on the Results tab in the Contents panel.
2. Do one of the following:
   – To configure a single model, select it, and click the Attributes tab in the Component Detail panel. Then use the Attributes tab to configure the CiscoPingApp model attributes.
   – To configure multiple models, right-click them all and click Tools, Utilities, Attribute Editor. Add the CiscoPingApp model attributes that you want to configure to the User Defined folder in the Attributes tree, and then use the Attribute Editor to modify the attributes.

NOTE
For more information about using the Attributes tab and the Attribute Editor to change attribute values, see the Modeling and Managing Your IT Infrastructure section.

Fault Isolation Across Switched Fabric

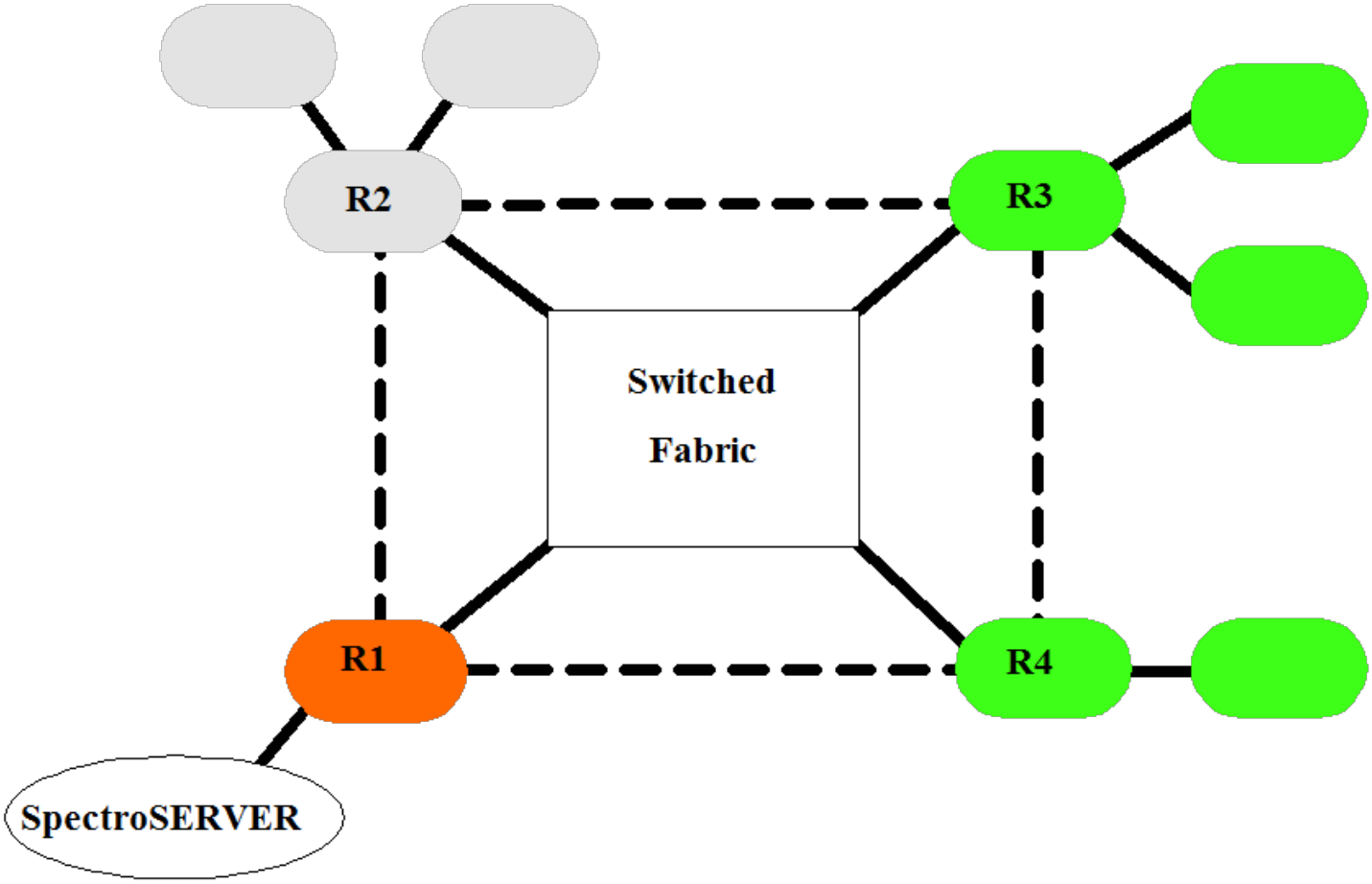
ATM Circuit Manager allows DX NetOps Spectrum to extend its fault isolation capability through the switched fabric of an ATM network. The switched fabric can be part of your own network infrastructure, or it can be part of a leased infrastructure on a service provider's network.

If you are managing your own switched fabric, there will be one or more ATM_Network models in your network topology. The ATM switches that make up the switched fabric are collected inside the ATM_Network model, and the ATM clients (routers, bridges, and so on) will be adjacent to the ATM_Network model. In this scenario, the managed circuits go from one client-through the switched fabric in the ATM_Network model-to another client. The channels and trunks within the switched fabric are not managed unless this functionality has been specified.

If you are leasing channels or trunks through a service provider's network, you will use an ATM_Cloud model in the DX NetOps Spectrum topology to represent the service provider's switched fabric. In this scenario, the managed circuits go from one client-through the switched fabric in the ATM_Cloud model-to another client. The channels and trunks within the switched fabric are not managed.

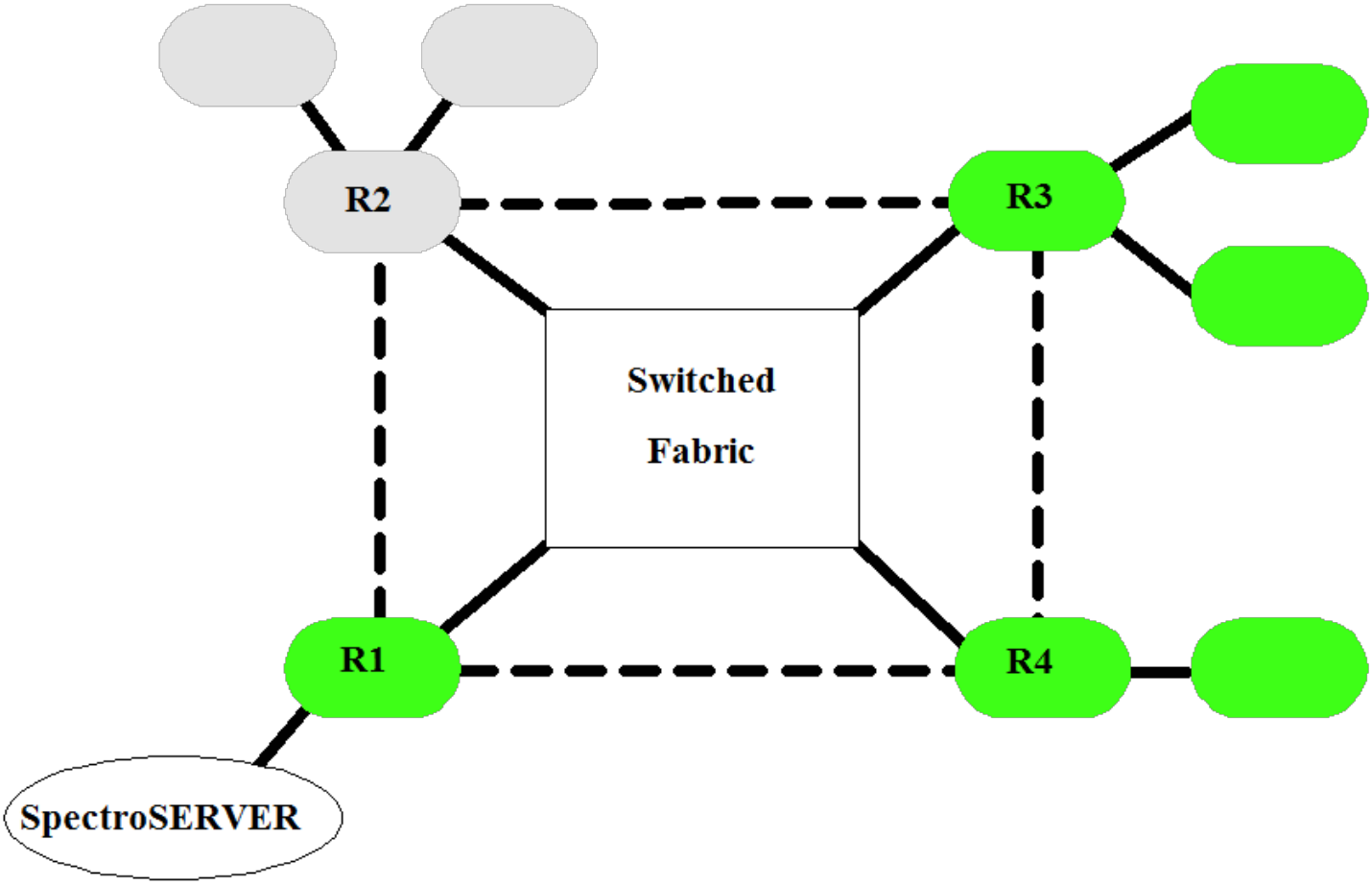
Connections between virtual link models are used to determine an ATM device's neighbors. The neighbors of a device are queried during fault isolation to determine where the source of the problem is located.

As an example, the following illustration shows four routers (R1, R2, R3, and R4) connected across a switched fabric. The logical connections between the routers are indicated by dotted lines, and the physical connections are indicated by solid lines. If the SpectroSERVER has a problem reaching R2, it will check the status of R2's neighbors to isolate the fault. Since the ATM Circuit Manager uses logical connectivity to determine device neighbors, both R1 and R3 will be considered neighbors of R2.



If, for example, R3 responds that it is available, but R1 does not respond, DX NetOps Spectrum creates a red alarm on R1 and changes R2 and the models that depend on its connectivity to gray to indicate a suppressed state.

If ATM Circuit Manager did not have the capability to use logical connectivity to determine device neighbors, it would not be able to isolate the fault to R1. Instead, R2 and its connected devices would change to gray to indicate a suppressed state, and an unresolved red alarm would be created on the generic Fault Management model to indicate the communication problem. The red alarm would be viewable in the Alarms tab in OneClick, but it would not appear linked to any device model in the Topology tab.



Cable Broadband Infrastructure

This section discusses how you can manage and monitor your Cable Broadband network using DX NetOps Spectrum.

Getting Started with Cable Broadband Solution

This section discusses the basic elements of a cable broadband network, cable broadband models available in DX NetOps Spectrum, device-specific MIB support available.

All this information helps you getting started with understanding your cable broadband infrastructure and modeling it in DX NetOps Spectrum.

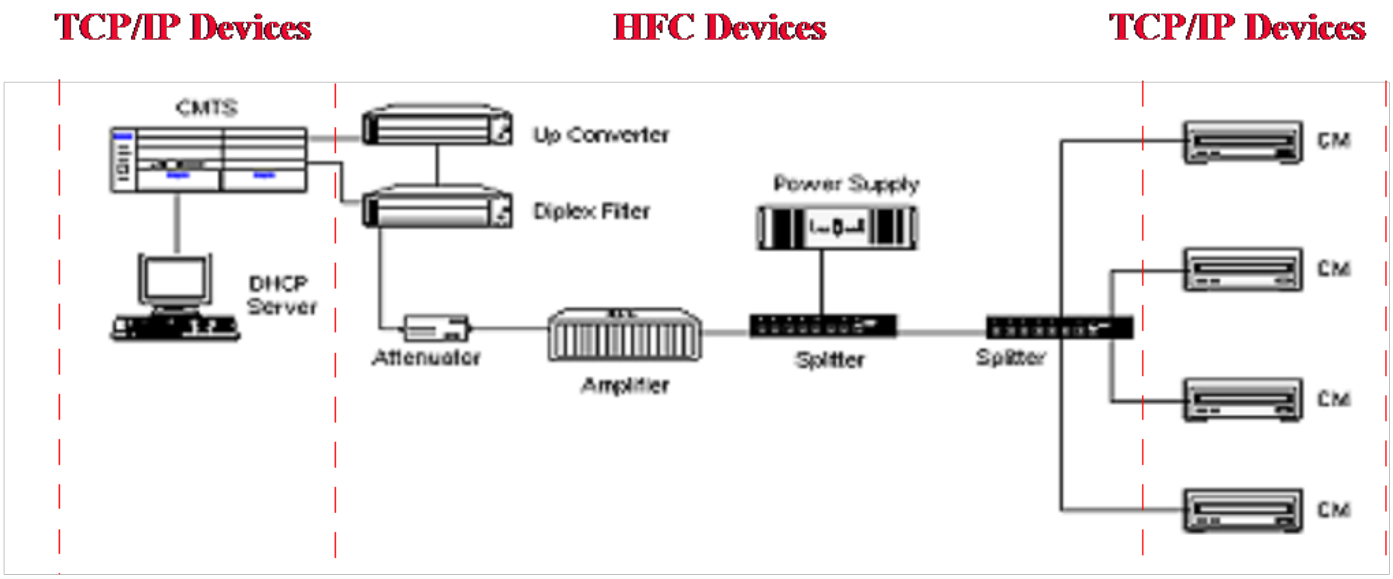
Network Management Factors

This section describes the elements of a broadband network and the Cable Broadband models in the DX NetOps Spectrum Cable Broadband Solution.

For a cable broadband management system to be effective, it must meet the challenge of acquiring an array of information available from a variety of devices diversely deployed to meet the demands of many different business requirements. To be useful to managers, the acquired information must be displayed in a meaningful manner.

The following image provides an example of a simple cable broadband network deployment.

Basic Components of a Cable Broadband Network



The cable broadband network includes Hybrid Fiber Coax (HFC) devices and TCP/IP devices. The HFC devices consist of Up Converters, Diplex Filters, Attenuators, Amplifiers, Splitters, and Power Supplies. The TCP/IP devices consist of DHCP servers, Cable Modem Termination Systems (CMTS), and Cable Modems (CM). The HFC components gather HFC information through proprietary communication methods. The TCP/IP devices gather network information through the Simple Network Management Protocol (SNMP).

In addition to the challenges imposed by the diversity of devices and deployments, an added complexity for network managers is the need to control SNMP traffic. Using SNMP, CMs communicate information to CMTSs over the HFC network using shared channel frequencies. The downstream frequency channel aggregates information from the CMTSs to the CMs and the upstream frequency channels aggregate information from the CMs to the CMTSs. Managers must have visibility into the amount of SNMP traffic that is generated on this shared media in order to maximize network efficiencies, plan for network growth, and have visibility into the operational status of devices for purposes of fault isolation and root cause analysis.

DX NetOps Spectrum's cable broadband network management solution meets the challenges summarized previously by:

- Supporting devices and MIBs produced by a variety of vendors.
- Limiting ICMP and SNMP traffic by introducing modeling techniques designed for cable broadband networks.
- Providing a mechanism for logically grouping device models.
- Providing a means for aggregating alarms from selected devices and setting aggregate threshold values based on mission criticality.
- Providing fault isolation and root cause analysis down to the port level.

Vendor MIB Support

Depending on the management module, vendor-specific information can be found either off the device model or in the Component Detail panel view in the form of an application. The device-specific DX NetOps Spectrum management modules related to cable broadband devices are mentioned in the following list. For example, the AM Communications device type supports the SM-AMC1000 management module.

- **AM Communications**
  SM-AMC1000
- **Arris Cadant C4 CMTS**
  SM-ARS1000
- **Broadband Service Containers**
  SM-BSC1000
- **Cheetah Gateway Integration**
  SM-SFA1000
- **Cisco uBR72xxCMTS**
  SM-CIS1008
- **DOCSIS Applications**
  SM-DCSCMN
- **DOCSIS Devices**
  SM-DCS1000
- **LANCity Cable TV Modem**
  SM-LCH1000
- **Motorola CDLP Cable Router**
  SM-MOT1001
- **RiverDelta BSR 1000/64000**
  SM-RVD1000
- **Riverstone SmartSwitch Router**
  SM-RST1000
- **Scientific Atlanta Explorer HCT**
  SM-SFA1000
- **Telecom CUDA 12000**
  SM-ADC1000
- **Terayon BroadbandEdge2000/TeraLink 1000**
  SM-TRN1000
- **Terayon BW3500 CMTS**
  SM-TRN1001

Device Support

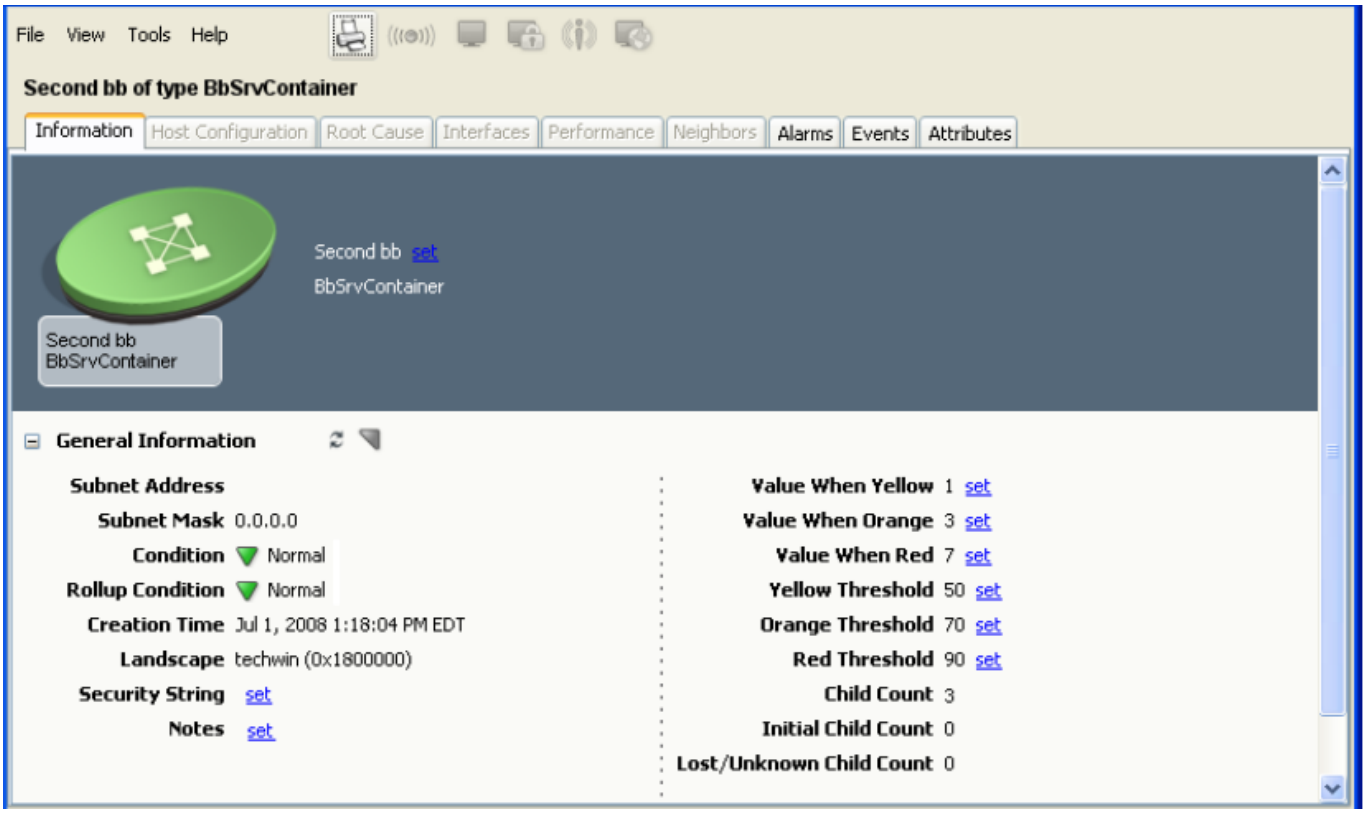
This section describes the cable broadband models and their functionality in DX NetOps Spectrum.

DX NetOps Spectrum provides efficient information gathering by limiting ICMP and SNMP traffic across the cable broadband network. This is accomplished by providing five model paradigms as follows:

- CMTS models provide the full information gathering aspects of a GnSNMPDev model type.
- Lightweight models provide focused information gathering for CMs and set-top boxes.
- Broadband Service Container models provide a method of logically grouping Lightweight models.
- HFC Device Event Modeling
- DOCSIS MIB Support

Each of these models is designed to focus information gathering to meet management's need for useful information, to eliminate distracting information, and to reduce the impact of polling on an efficiently operating cable broadband network.

The following image provides an example of how DX NetOps Spectrum displays cable broadband component details:



CMTS Models

The following section describes DX NetOps Spectrum's support of CMTS devices and DOCSIS-compliant devices.

CMTS Model Device Support

DX NetOps Spectrum supports a variety of CMTS devices from vendors including Cisco, Terayon, Motorola, Nortel LANCity, River Delta (acquired by Motorola), ADC, and Riverstone. The CMTS device models have standard device support including model creation for applications and interfaces, Topology and Device Interface views, interface connectivity/port resolution, Discovery, and fault isolation capabilities. These device models are based on the standard support provided by the GnSNMPDev model type.

CMTS Model DOCSIS Support

DX NetOps Spectrum also supports CMTS devices that comply with DOCSIS.

CMTS Fault Isolation, Polling, and Logging

For information about fault isolation, polling, and logging, see the Modeling and Managing Your IT Infrastructure section.

Lightweight Models

Lightweight models include CMs and set-top boxes. The lightweight model paradigm supports devices that do not require the full functionality that GnSNMPDev models provide.

Lightweight Model Device Support

DX NetOps Spectrum supports the Motorola cable modem, a DOCSIS-compliant cable modem, and the Scientific Atlanta set-top box. A cable broadband network can include thousands of these devices; therefore, the models of these devices are designed as lightweight models. The lightweight series of models provide a way to represent and collect meaningful SNMP data from all cable modems and set-top boxes while limiting the effects of polling on the network.

The implementation of the new lightweight model paradigm has many significant differences over conventional GnSNMPDev models. The lightweight models have increased model capacity on a SpectroSERVER, reduced SNMP traffic, and reduced memory and CPU usage. However, the lightweight models do not create interface models for port resolution, and do not participate in fault isolation. They only communicate SNMP to their real world counterpart, whereas a GnSNMPDev model will also try to use ICMP ping to contact the device.

Lightweight Model DOCSIS Support

Support also includes a generic DOCSIS-compliant CM device model for those vendor devices that DX NetOps Spectrum does not support but which are DOCSIS-compliant.

Lightweight Model Fault Isolation

Because the Lightweight Model Architecture does not participate in fault isolation, there is no value from a fault isolation standpoint of connecting these models by pipes. Also, by default lightweight models do not alarm. If contact with the device via SNMP has been lost, the model will turn gray and go into a suppressed state. This is done to keep the alarm manager from being flooded with red alarms from cable or set-top models losing contact. This functionality is configurable and can be set to alarm or not to alarm.

Lightweight Model Polling and Logging

Lightweight models do not log or poll any attributes. For this reason, they keep in contact with the device at three times the polling interval. This is why lightweight models do not turn active over a polling interval. Lightweight models can only be modeled by model type; you cannot model them using Discovery or by using Model by IP.

> **NOTE**
> Modeling cable modems over the HFC network is not advised. Because most cable modems change IP addresses on a regular basis, there would be too much SNMP traffic generated to update cable modem models.

In the future, DX NetOps Spectrum will contain auto-population features that will create and update cable modem models from the CMTS MIB tables.

Broadband Service Container Models

The broadband service container model (BbSrvContainer) provides a mechanism for the logical grouping of lightweight models. To see how the broadband service container differs from other standard containers, first consider the process used by a standard container to monitor the condition of models it collects.

Standard Container Characteristics

With standard containers, such as a Network or LAN, the container is responsible for summing the condition value of every device (or container) it collects. This sum is written to an attribute called composite condition. The composite condition is then compared to the rollup threshold values of yellow, orange, and red. The rollup thresholds are defined for minor, major, and critical severities. For every severity state, there is a significance level that can be defined. If the

composite condition exceeds a rollup threshold then the rollup condition assumes that threshold condition and color. This process for standard containers is described in detail in the Modeling and Managing Your IT Infrastructure section.

Broadband Service Container Characteristics

In contrast to standard containers, the broadband service container monitors the percentage of models it has collected that have lost contact compared with the total sum of models, excluding any model still in the initial state. The percentage thresholds are defined for minor, major, and critical severities. For every severity state, there is a significance level that can be defined. However, when a threshold has been violated, the broadband service container assumes the condition associated with the threshold. The broadband service container assumes a condition to reflect the condition of the network represented by the devices grouped in the container. This is done because the alarms on the individual cable or set-top models are suppressed.

The container model's General Information subview contains the following two attributes which determine the condition shown on the container in the Topology view: Condition Value and Lost/Unknown Child Count.

The Lost/Unknown Child Count displays the percentage of the devices collected by this container that have lost contact. The value of Lost/Unknown Child Count is compared with the rollup thresholds. If the value of Lost/Unknown Child Count exceeds a threshold, the container will set the Condition Value attribute to that criticality. The table below shows possible condition values.

Once the Condition Value of the Broadband Service Container assumes a criticality of yellow, orange, or red, the significance levels are taken into consideration by DX NetOps Spectrum. That is, the current Condition Value of the BbSrvContainer is compared with the significance level values and the significance value of the BbSrvContainer is then set, based on this comparison.

Models of cable modems and set-top boxes should never be directly connected in DX NetOps Spectrum to models of CMTS devices. To help ensure proper fault isolation for CMs and set-top boxes, the broadband service container model must be used to group models of these lightweight devices.

HFC Device Event Modeling

DX NetOps Spectrum does not model HFC devices. Most HFC devices communicate with a Headend Communications Controller (HEC). The communication protocol between the HEC and HFC devices is usually proprietary. For example, Acterna (Cheetah) and AM Communications developed a software application to communicate with their respective HECs. These software applications are capable of sending SNMP traps. DX NetOps Spectrum collects and processes these traps (sent from either software application) using the Southbound Gateway.

When an SNMP trap is sent from the software application, the Southbound Gateway analyzes the data and creates a new event model, if one has not already been created. From that point forward, traps sent from the application are mapped to that event model. The traps will be further processed and events and alarms created on the event model.

DOCSIS MIB Support

Support also includes a generic DOCSIS-compliant CMTS device for those vendor devices that DX NetOps Spectrum does not support but are DOCSIS-compliant. In the case of these devices, the CMTS models will have the DOCSIS information available in the application view as applications. There is an application for each of the DOCSIS MIBs shown in the following table that will discover automatically if the device supports the MIB.

DOCSIS MIB Listing	1.0 Standard	1.1 Standard	2.0 Standard	Supported by DX NetOps Spectrum
RFC 2669: Cable Device MIB	Yes	Yes	Yes	Yes
RFC 2670: Radio Frequency Interface	Yes	Yes	Yes	Yes

RFC 3083: Baseline Privacy Interface	Yes	Yes	Yes	Yes
Quality of Service	No	Yes	Yes	Yes
Baseline Privacy Interface Plus	No	Yes	Yes	Yes

Broadband Service Container Model

This section describes how to model broadband network devices and access MIB attribute information from the BbSrvContainer model's General Information subview.

Create a Broadband Service Container Model

The following procedure describes how to create a Broadband Service Container model.

> **NOTE**
> For more information about creating models and container models, see Modeling and Managing Your IT Infrastructure section.

To create a Broadband Service Container model

1. In the Explorer tab of the OneClick Navigation panel, select the Universe topology view where you want to add the new container.
   The selected topology view appears in the Topology tab of the Contents panel.
2. In the Topology tab of the Contents panel, click
   
   (Creates a new model by type) in the Topology tab toolbar.
   The Select Model Type dialog opens.
3. Click 'BbSrvContainer' in the Containers tab and click OK to add the container to the Topology tab.
   The 'Create Model of Type BbSrvContainer' dialog appears.
4. Type a Name and Security String for the container and click OK.
   The container is added to the Topology view and you can view details in the Component Detail panel.

> **NOTE**
> You can create more than one container model to separately monitor different parts of a network. A container model can be created inside other container models, and it can be copied and pasted into the topology on an appropriate interface to provide port resolution for the broadband devices. Where and how you model containers and devices depends on your network configuration and how you want to view it in DX NetOps Spectrum.

Model Devices

Once the Broadband container model has been created, you can model devices in its Topology view as needed. The following procedure describes how to do so manually. For more detailed information about creating models and container models, see the Modeling and Managing Your IT Infrastructure section.

To manually add device models to the Broadband Service Container

1. In the Explorer tab of the OneClick Navigation panel, select the broadband service container to which you want to add new device models.
   The selected topology view appears in the Topology tab of the Contents panel.

2. Click
   
   (Creates a new model by type) in the Topology tab toolbar.
   The Select Model Type dialog opens.
3. Select the desired model type and click OK.
   The 'Create Model of Type' dialog appears.
4. Complete the fields as needed and click OK.
   The device model is added to the Topology view and you can view details in the Component Detail panel.

General Information Subview

From the Broadband container model's General Information subview you can access information about the status of the model and its children, its rollup thresholds, and its significance levels. It contains the settings listed here and in the following sections.

- **Condition**
  Reflects the current contact or alarm status of the model itself.
- **Rollup Condition**
  Applies to container models; reflects the composite status of all the other models in the container, which are sometimes referred to as its children. The percentage of devices in the container that are down, excluding those devices whose models were never active.
  The following table shows possible Rollup Condition values:

Condition Value	Alarm Status	Label Color
0	Normal	Green
1	Minor	Yellow
2	Major	Orange
3	Critical	Red

- **Child Count**
  Specifies the total number of devices in the container, which includes active, Initial Child, and Lost Child models.
- **Initial Child Count**
  Specifies the number of devices in the container whose models are in the Initial state.
- **Lost/Unknown Child Count**
  Specifies the number of devices in the container that were active but have lost contact with the network.

Significance Levels

Significance levels for the BbSrvContainer model weigh the importance of the model for each possible alarm severity the model may reach.

In the case of the BbSrvContainer model, the significance levels represent the importance of the cable modems and set-top models. When a BbSrvContainer model is collected by a parent container and the BbSrvContainer model reaches a particular alarm severity, the significance value will be used to calculate the parent container's alarm severity.

- **Value When Yellow**
  Specifies the point value of a Yellow alarm condition existing in a child towards the rollup alarm threshold value for the parent container.

Default: 1

- **Value When Orange**
  Specifies the point weight of an Orange alarm condition existing in a child towards the rollup alarm threshold value for the parent container.
  Default: 3
- **Value When Red**
  The point weight of a Red alarm condition existing in a child towards the rollup alarm threshold value for the parent container.
  Default: 7

Rollup Thresholds

The rollup thresholds are three read-write values that control when the BbSrvContainer model's rollup condition icon changes color and also controls when alarms are triggered. Each of the threshold values represents the percentage of active devices in the BbSrvContainer that have gone down. When the actual percentage of devices that are down equals or exceeds a threshold value, the BbSrvContainer model's rollup condition icon changes to the color associated with that threshold and a commensurate alarm is triggered.

Expressed mathematically, the rollup threshold value is:

```
Lost Child Count divided by (Child Count - Initial Child Count) x 100
```

In other words, models that have never been active are excluded from the percentage value.

You can set the thresholds to suit your requirements or use the default values. Recommendations for each threshold value are as follows:

- **Yellow Threshold**
  Minor alarm threshold. Specifies the minimum points needed to trigger a Yellow rollup alarm for a container. You might use this threshold to indicate a network condition that is less than optimum but does not threaten service.
  Default: 50
- **Orange Threshold**
  Major alarm threshold. Specifies the minimum points needed to trigger an Orange rollup alarm for a container. You might use this threshold to indicate a network condition that should be examined before it threatens service.
  Default: 70
- **Red Threshold**
  Critical alarm threshold. Specifies the minimum points needed to trigger a Red rollup alarm for a container. You might use this threshold to indicate a network condition that has a serious impact on service.
  Default: 90

  NOTE
  Change threshold levels carefully; you may see an increase in generated alarms if threshold levels are set lower, or a decrease in generated alarms if levels are set higher.

Certifications

This section explains the concepts of DX NetOps Spectrum Device Certification:.

Out-of-the-Box Certification Support

This section discusses how DX NetOps Spectrum provides out-of-the-box support for devices, interfaces, and applications. This section also discusses how DX NetOps Spectrum models a device using the device certification, how you can reconfigure existing models with new certification support.

Overview on Certification

Support for monitoring many devices is provided out-of-the-box in DX NetOps Spectrum. Basic monitoring support is supplied through simple or enhanced certifications:

- **Simple Support -** The device is modeled using the DX NetOps Spectrum generic certification. This level of certification provides core DX NetOps Spectrum capabilities, including discovery, identification, standard MIB and trap support, and standard views. Simple support also includes interface modeling and participation in fault isolation and root cause analysis.
- **Enhanced Support** - The device is modeled using one of the DX NetOps Spectrum enhanced certifications, which extend simple certification support. At a minimum, enhanced certification support indicates that support for this device has been extended with proprietary MIB and trap support. Typical extensions include proprietary OneClick views, CPU and memory device thresholding, and serial number support.

About Generic Certification

DX NetOps Spectrum provides a generic certification to represent an SNMP-compliant network device that lacks a corresponding DX NetOps Spectrum enhanced certification. Management Information Bases (MIBs) support SNMP-compliant devices. MIBs are SNMP structures that describe particular devices. MIBs are imported into the DX NetOps Spectrum database and made available through device, application, and interface model types.

Note: For more information about simple and enhanced certification support, see the Standards-Based Protocol Reference section. For more information about enhanced certification support, see the Device Management Reference section, Cisco Device Management section, and Host System Resources Management section.

The generic model type, GnSNMPDev, can represent a broad range of devices by creating the following models:

- A model to represent the device.
- Application models to represent each of the standard (IETF) MIBs that the device supports.
- Interface models to represent device ports.

GnSNMPDev lets DX NetOps Spectrum dynamically create models to manage devices when a specific management module is unavailable.

GnSNMPDev rapidly queries the device to determine its characteristics and capabilities and then creates a model to represent the device. GnSNMPDev also creates the following models:

- Submodels, referred to as application models, to represent each of the standard MIBs that the device supports.
- Interface models to represent each device port that is defined in the standard MIB-II interface table.

The application and interface models are associated with the GnSNMPDev device model. Together, they provide management capabilities for the device.

Devices that are modeled with the GnSNMPDev model type can be used with all DX NetOps Spectrum management tools. GnSNMPDev models participate fully in DX NetOps Spectrum root cause analysis, fault isolation, and downstream alarm suppression algorithms. As a result, they can alert users to network and device problems.

Device Modeling

When modeling a device using Discovery or the Model by IP Address icon, DX NetOps Spectrum automatically chooses the GnSNMPDev model type when an enhanced certification for the device is not available. You can also model a device using the GnSNMPDev model type when you use the Model by Type feature.

You can map the connectivity of interface models automatically using Discovery, or you can map connectivity manually.

The GnSNMPDev model type supports the Cisco Proprietary Discovery Protocol (CDP). A CiscoCDPApp application model is created for Cisco devices that are modeled with GnSNMPDev and that support CDP. This application model lets DX NetOps Spectrum use the Proprietary Discovery tables for the device when discovering device connectivity information.

NOTE
For more information, see the Modeling and Managing Your IT Infrastructure section.

How DX NetOps Spectrum Identifies the Device Type

When modeling a device, DX NetOps Spectrum assigns a descriptive identifier or device type. The device icon shape and label reflect device functionality in OneClick.

The DeviceType attribute (0x23000e) is a text string that identifies the type of device being modeled. In OneClick, this string is displayed below the device icon. DX NetOps Spectrum lets you search, filter, and report on device models using the DeviceType attribute.

The following process describes how DX NetOps Spectrum determines the device type to assign to a device model:

1. If the device type setting is locked, DX NetOps Spectrum does not reevaluate the device model. The device type that is set for the device model remains.
2. If the device type setting is not locked, DX NetOps Spectrum runs custom intelligence for some models to set the device type name.
3. DX NetOps Spectrum checks the System Object Identifier-to-Device Type mapping list. If a device type name (for example, “Cisco 2621”) is found for the device System Object Identifier, it becomes the model device type. If no match is found, DX NetOps Spectrum extracts the device enterprise ID from the System Object Identifier. DX NetOps Spectrum uses the enterprise ID to identify the manufacturer.
4. DX NetOps Spectrum then checks device capabilities and appends an abbreviation (for example, Rtr or Bdg) to the manufacturer name. This entire string becomes the device type name in OneClick (for example, “Cisco Rtr”).
5. If DX NetOps Spectrum cannot determine an appropriate device type, the default value “SNMP DV” is assigned.

How DX NetOps Spectrum Identifies the Model Class

DX NetOps Spectrum evaluates the model class when a device is modeled for the first time and when you reconfigure a device model.

The following process describes how DX NetOps Spectrum determines the model class to assign to a device model:

1. If the model class setting is locked, DX NetOps Spectrum does not reevaluate the device model. The model class that is set for the device model remains as is.
2. If the model class setting is not locked, DX NetOps Spectrum checks device model support for a specific MIB object. If DX NetOps Spectrum detects that a device model supports a certain MIB object, the model class for that device model is set to a specified value.
3. If the search for a supported MIB object fails, DX NetOps Spectrum attempts to determine the model class for the device model. DX NetOps Spectrum uses the mappings in the Device Certification utility in this search. This utility provides a mapping from System Object ID (which is more general than a MIB object) to Model class.
4. If Device Certification does not contain any model class mappings for a device, DX NetOps Spectrum defaults to setting the model class based on whether the device appears to be routing (“Router”), switching (“Switch”), both switching and routing ("Switch-Router”) or simply repeating (“Repeater”).

When assigning the icon and label for a device model, DX NetOps Spectrum uses the icon for the model class that is assigned as described here. This icon appears throughout OneClick.

Support for Chassis Devices

When a device model uses a DX NetOps Spectrum certified proprietary chassis MIB or the Entity MIB, DX NetOps Spectrum identifies that device model as a chassis device. DX NetOps Spectrum models and arranges all the identified chassis devices with their components or modules under the **Chassis Manager** node in the navigation pane of OneClick. This arrangement is based on the vendor names of your chassis devices.

From 10.4.2, you can discover the Cisco FEX module. The FEX module is displayed under the Chasis Manager view.

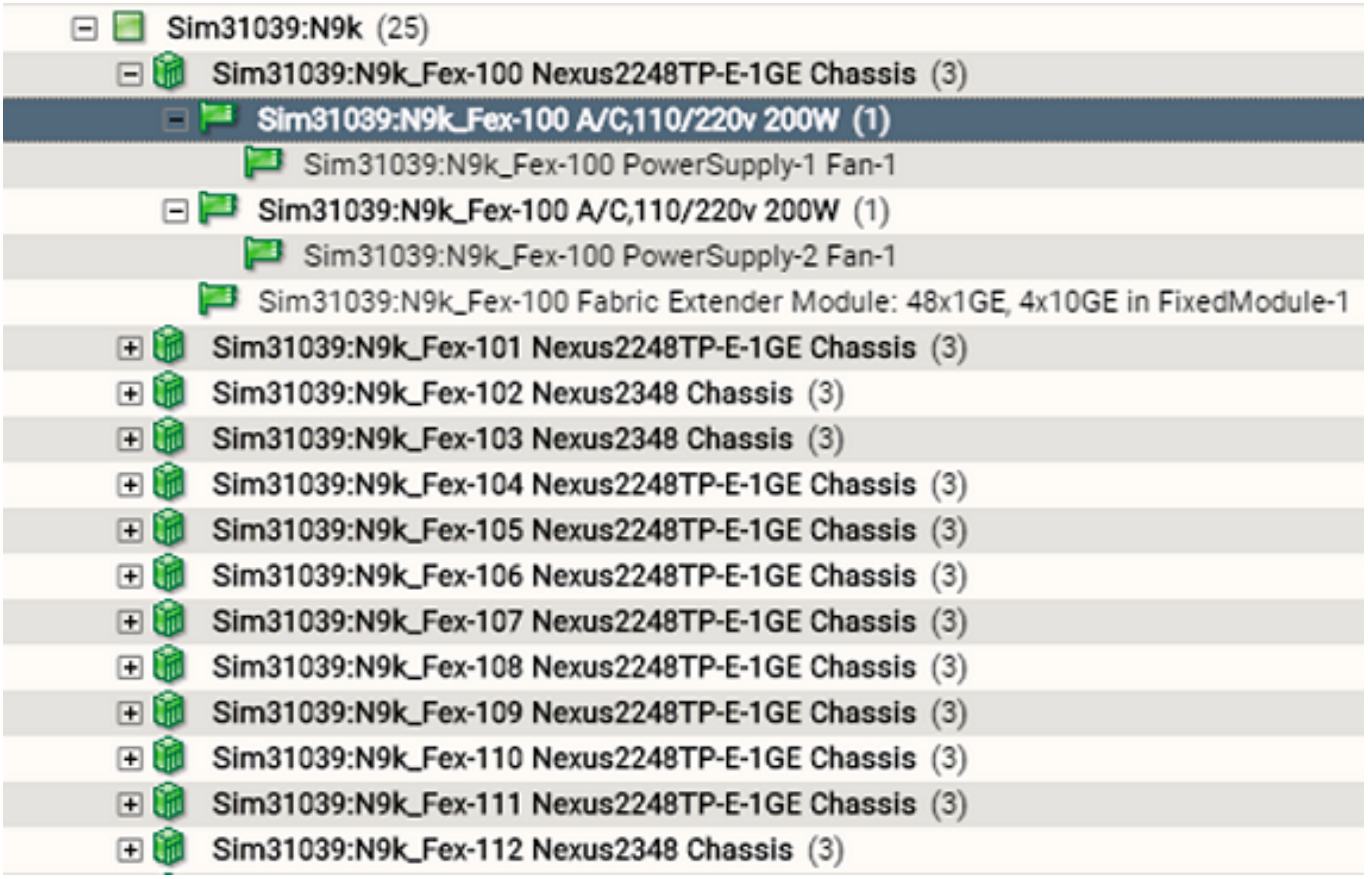
The following CISCO FEX devices are supported:

- Nexus 5596
- Nexus 7000
- Nexus 5000
- Nexus 6001
- Nexus 5672

> **NOTE**
> For active-active FEX configuration, both monitoring devices display all the FEX modules because both entity MIBs contain all the FEX modules.

For example, devices of "Cisco" that are identified as chassis devices are arranged in a folder with the name "Cisco". Similarly, devices of "Juniper" identified as chassis devices are arranged in the "Juniper" folder.

The following image shows how chassis devices are modeled and arranged under the **Chassis Manager** node. Each vendor folder contains its chassis devices:



As a result, you can monitor the health of your chassis devices and their modules at one place in OneClick, the **Chassis Manager** node. This node provides a consolidated location to view and manage chassis devices that are modeled in the Universe topology. After selecting a chassis device from this node and then accessing chassis views from the Component Detail pane, you can view the status of all interfaces, and can assess the health of each module. For more information about the chassis views that give detailed information about each module of your chassis device, see Chassis Views.

Identification of Chassis Devices

DX NetOps Spectrum identifies your device as a chassis device based on the following two types of MIBs:

- **Proprietary MIB**
  When a device supports a DX NetOps Spectrum certified proprietary chassis MIB, it is identified as a chassis device. For example, when a "Juniper" device supports the "jnxBoxAnatomy", it is identified as a chassis device based on that MIB.

  NOTE
  DX NetOps Spectrum always prefers the DX NetOps Spectrum certified proprietary chassis MIB to the Entity MIB of a device model for its identification as a chassis device. Only when the proprietary chassis MIB is absent, the device model is identified as a chassis device using the Entity MIB.

  NOTE

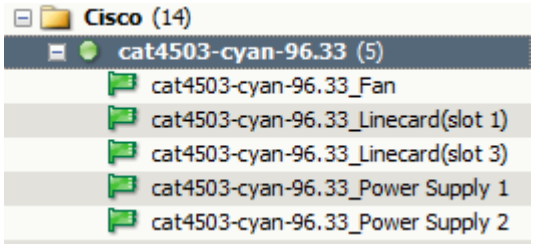
  An attribute(createChassisModules) is added (available on device) to enable or disable modeling of Slots/ Chassis Modules for Juniper devices. By default, this attribute is set to True and all the chassis modules are discovered and modeled for Juniper devices. To disable modeling of chassis modules, set this attribute value to False and perform reconfiguration on the device.
- **Entity MIB**
  When a device model supports the Entity MIB and the value of the "**EnableEntityModuleModeling**" attribute is "Yes" for that device model, it is identified as a chassis device. By default, the value of this attribute is "Yes" on a case-by-case basis for the following reasons:
  – Some vendors do not implement this MIB indexing scheme correctly.
  – Some vendors support the Entity MIB even for non-chassis devices.

  NOTE
  If you do not want DX NetOps Spectrum to identify a device model as a chassis device, set the value of this attribute to "No" and reconfigure the model.

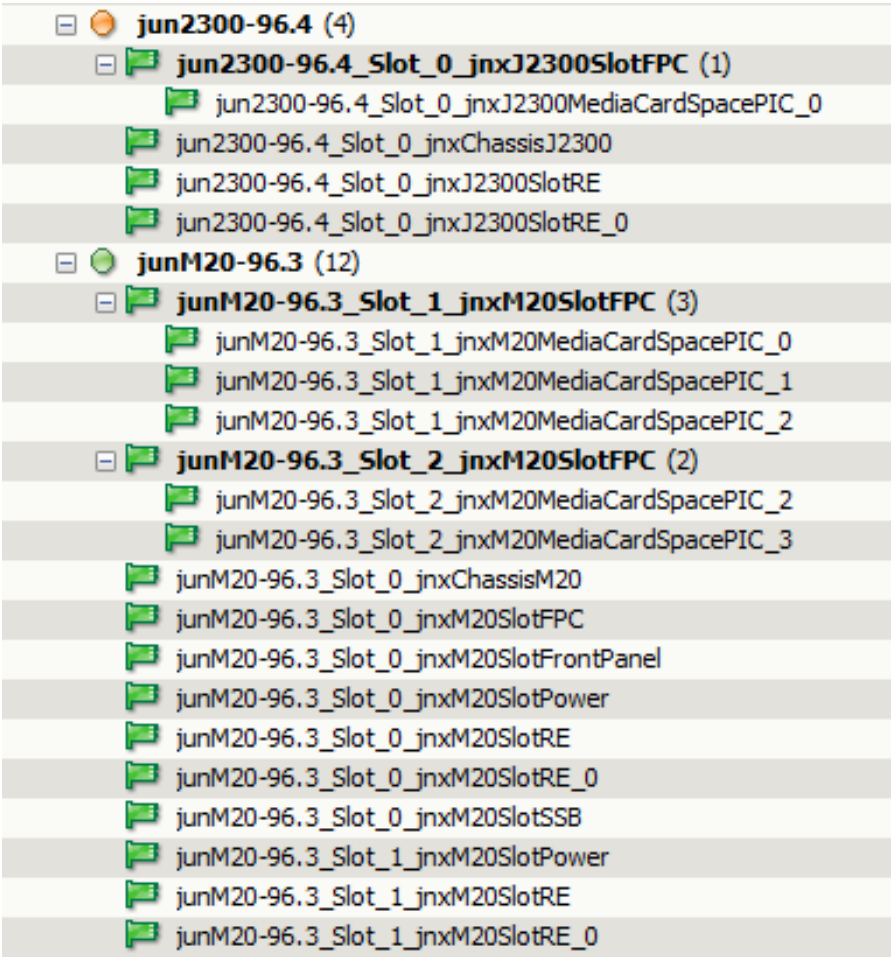
Chassis Views

To view the details of your chassis devices, DX NetOps Spectrum displays the following three types of chassis views in OneClick:

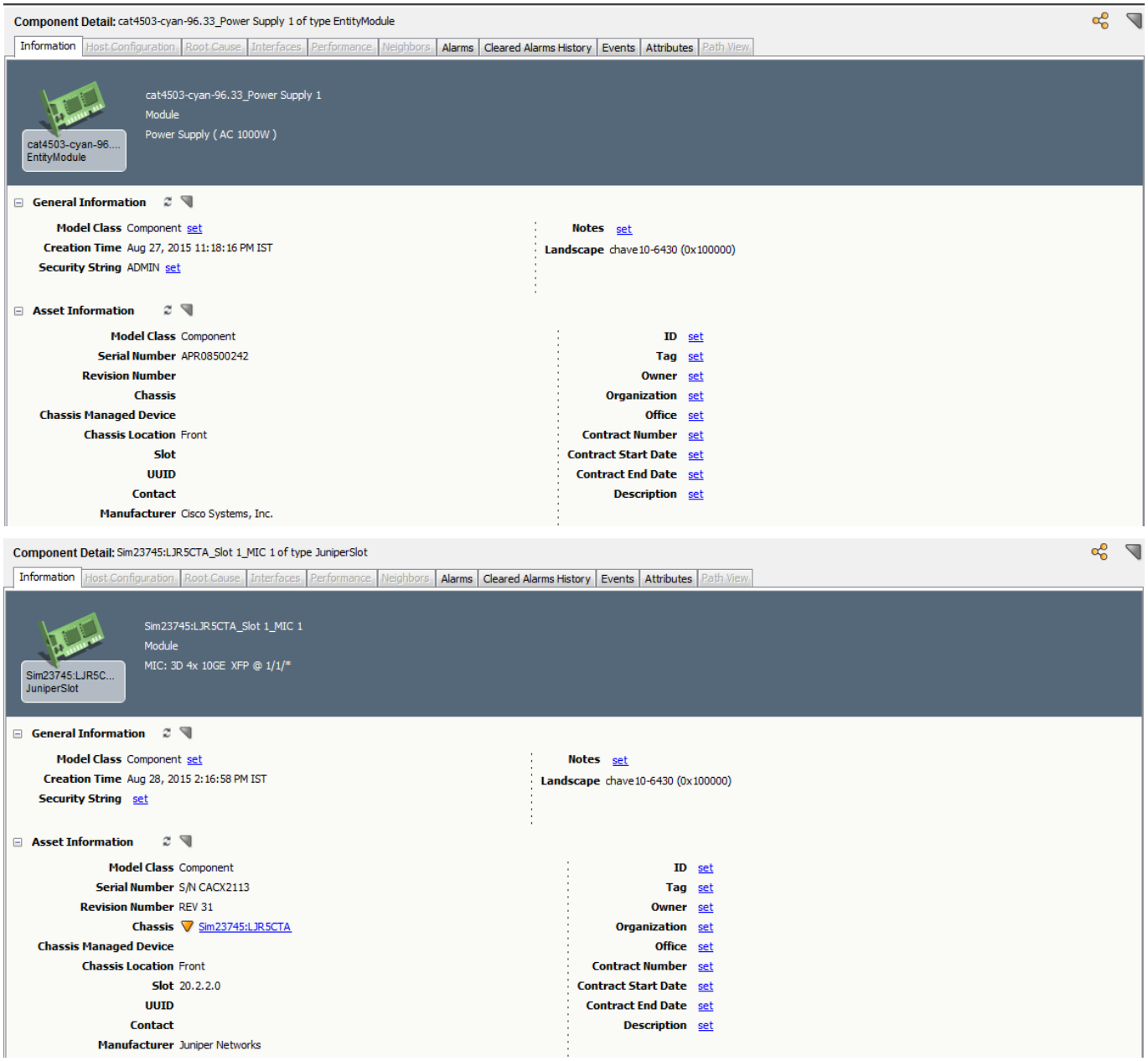
- **The Basic Module-Level View**
  This view of your chassis device can be viewed at the Chassis Manager node. This view gives you visibility into your chassis device by displaying all the modules of it. The following image shows the various modules displayed under a Cisco chassis device:



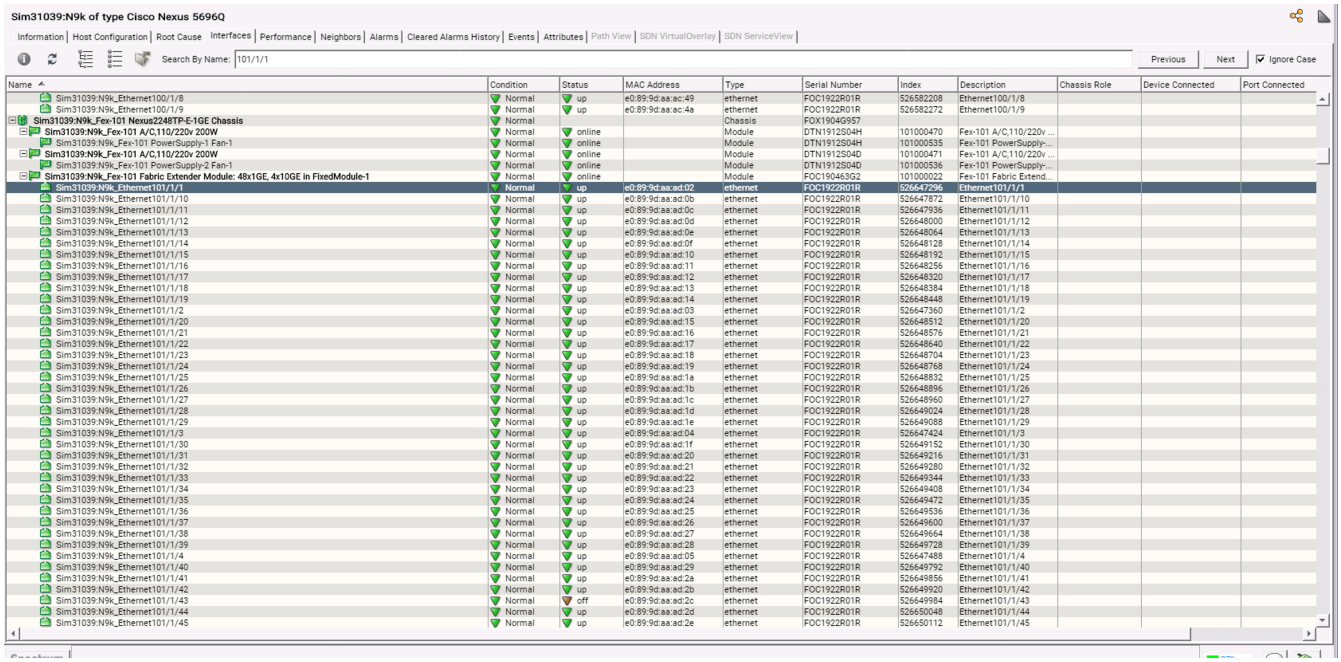
  The following image shows the various modules displayed under a Juniper chassis device:



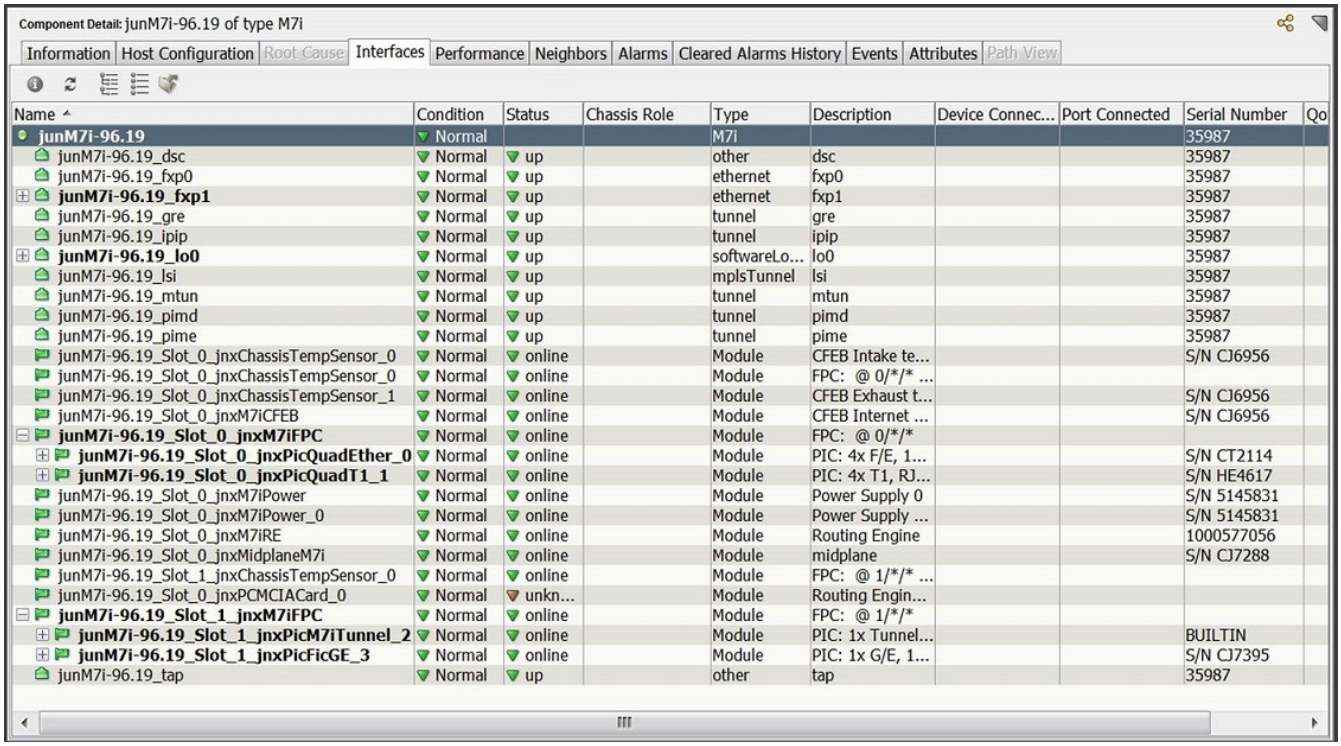
If you select a particular module under any chassis device, you can see the hardware description of that module. The hardware description lets you know whether the module is Fan, Linecard, Power Supply, Routing Engine, PIC, MIC, and so on. The following images show how the hardware description of a Cisco and Juniper module is displayed in the **Component Detail> Information:**



- **The Interfaces View**
  This view is an elaborated view of all interfaces present in each module of your chassis device. For a selected chassis device, this view shows all of its interface modules, interfaces within each module, the status of modules and its interfaces, and other information under the Interfaces tab of the Component Detail pane. From 10.4.2, all the FEX down ports are listed under the FEX modules. The following image shows how a "Cisco" chassis device is populated under the Interfaces tab:

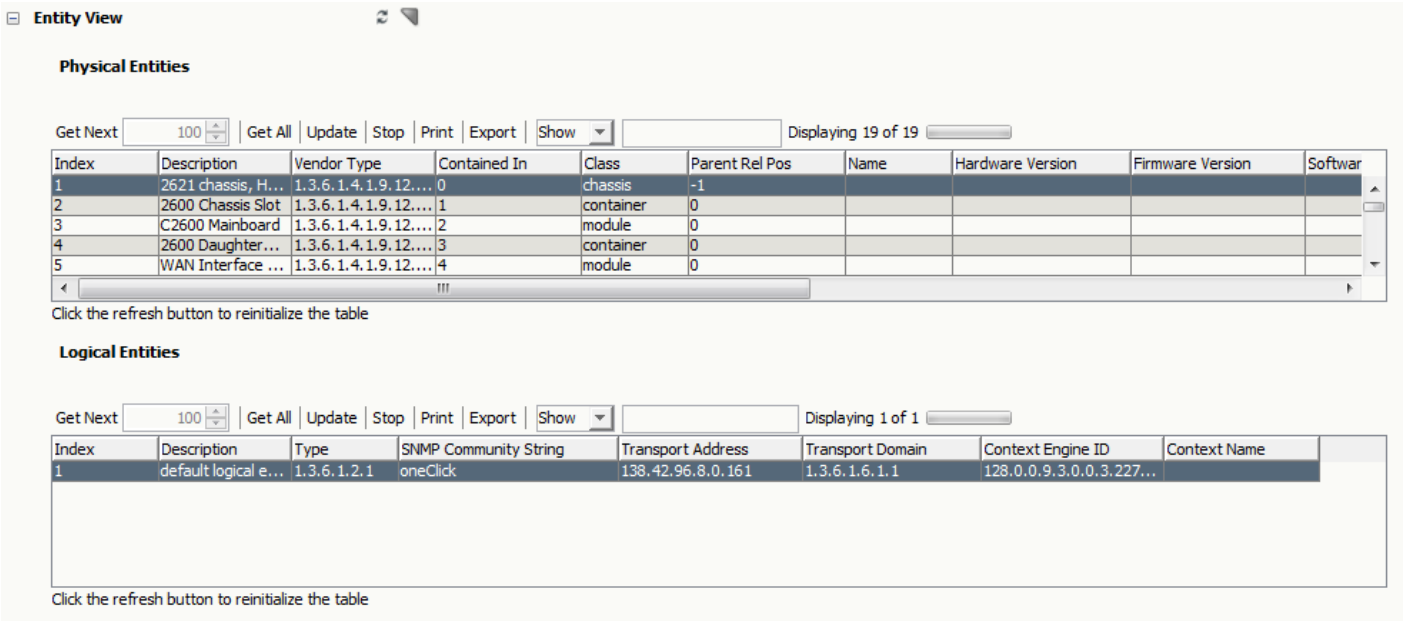


The following image shows how a "Juniper" chassis device is populated under the interfaces tab:



- **The Entity View**
  DX NetOps Spectrum populates this view only for those chassis devices which support the Entity MIB. The **Entity View** is populated when you expand the "**Entity View**" under the **Information** tab of the **Component Detail** pane. This view has the following two sections:
  - **Physical Entities**
    This section populates the information about each module that exists in your chassis device.
  - **Logical Entities**

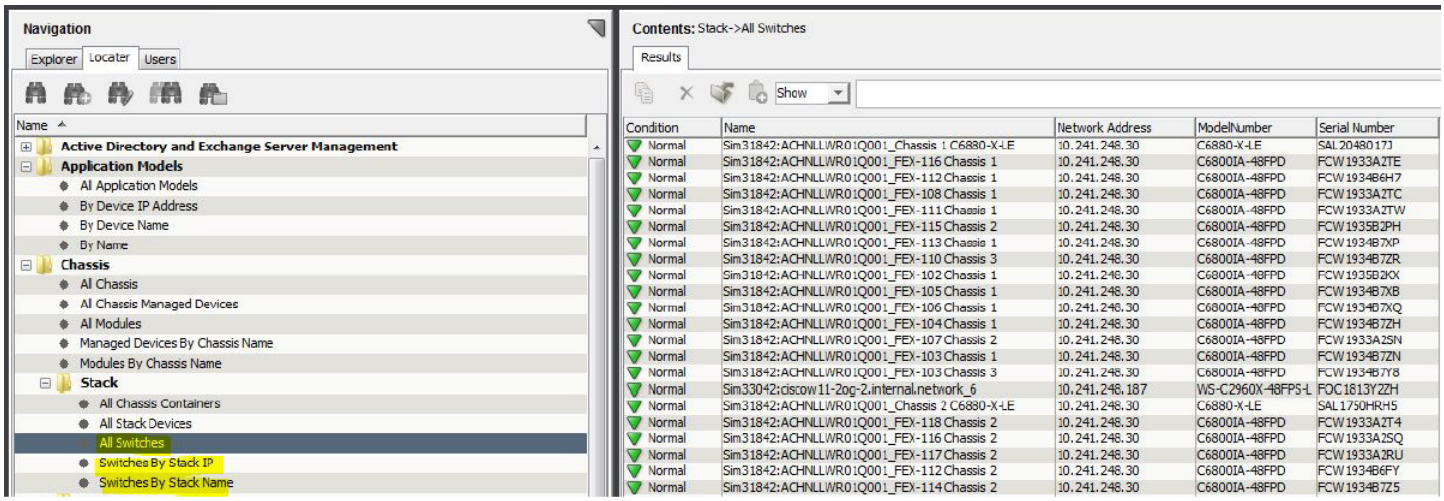
This section populates the information about the logical entities that exist in each module of your chassis device. The following image shows how the **Entity View** is populated for a selected "Cisco" chassis device view under the Chassis Manager node:



The Locater Search

DX NetOps Spectrum allows you to find all your modeled chassis devices and their modules in the **Chassis** node of the Locater tab in the Navigation pane. To find all your modeled chassis devices and their modules, use the following five search criteria available under the **Chassis** node:

- **All Chassis**
  This search criteria finds all your chassis devices that are modeled and arranged under the **Chassis Manager** node of your landscape. The result of this search lists all of your modeled chassis devices.
- **All Chassis Managed Devices**
  This search criteria finds each SNMP capable device model existing on all your modeled chassis devices. The result of this search lists each SNMP capable device model with name of its chassis device.
- **All Modules**
  This search criteria finds all your existing modules that are modeled and arranged under each modeled chassis device. The result of this search lists all modeled modules existing in each modeled chassis device.
- **All FEX Modules**
  This search criteria finds all your existing modules that are modeled and arranged under each modeled chassis device. The result of this search lists all modeled modules existing in each modeled chassis device.
- **Managed Devices By Chassis Name**
  This search criteria finds each SNMP capable device model by the name of its chassis device. The result of this search lists all SNMP capable devices that are mounted on the chassis device you specify.
- **Modules by Chassis Name**
  This search criteria finds each modeled module by the name of its chassis device. The result of this search lists all the modeled modules of the chassis device you specify.
- **Stack**
  From Spectrum 10.3.2 onwards, the 'All switches' option (as shown in the screenshot) fetches details such as the ModelName, Serial Number, ModelNumber, IPAddress of all the switches in a selected landscapes, including the details of the members of the stack devices.



Chassis Alarms

When DX NetOps Spectrum identifies a device model as a chassis device, the **Chassis Fault Domain** is associated with that chassis device. This condition correlation domain correlates various alarms on a chassis device and its modules to raise different root cause alarms. For more information about condition correlation domains, see Condition Correlation section. The existing alarm modules take care of the FEX down scenarios.

The following alarms are the Chassis Fault Domain alarms:

- **Chassis Down (0x00010f69)**
  This alarm is raised when the contact with a chassis device is lost. This alarm is the root cause alarm that suppresses the following alarms that are raised on a chassis device:
  – ContactLost_Red (0x00010d35)
  – Blade Status Unkown (0x00010f71)
  – InferConnectorContactLost_red (0x00010d90)
  – Linkdown (0x00010d11)
- **Blade Status Unkown (0x00010f71)**
  This alarm is raised when DX NetOps Spectrum is not able to contact the chassis on-board agent. This alarm is the root cause alarm that suppresses the following alarms that are raised on a module of a chassis device:
  – Catalyst Dev Module Failed (0x011c0488)
  – Dev Module Failed (0x00010f70)
  – Dev Module Offline (0x00010f86)
  – Dev Module Pulled (0x00010f6b)
  – Module Offline (0x00010f87)
  – Module Pulled (0x00010f6d)
- **Module Offline (0x00010f87)**
  This alarm is raised when the state of a Module is reported as "Offline". This alarm is the root cause alarm that suppresses the following alarms that are raised on a module of a chassis device:
  – ContactLost_Grey (0x00010d36)
  – ContactLost_red (0x000103d5)
  – Physical Host Down (0x056e000c)
- **Module Pulled (0x00010f6d)**
  This alarm is raised when a Module is pulled out from the chassis. This alarm is the root cause alarm that suppresses the following alarms that are raised on a module of a chassis device:

- ContactLost_Grey (0x00010d36)
- ContactLost_red (0x000103d5)
- Physical Host Down (0x056e000c)

Reconfigure Existing Models with New Certification Support

Existing models do not reevaluate their model class and device type at server startup. Therefore, new mappings that are available in a patch or with an upgrade are not applied to existing models. To pick up the new mappings, reconfigure your existing models.

Follow these steps:

1. Select the device models to update in any OneClick view.
2. Right-click the selected models and select Reconfiguration, Reconfigure Model.
   The selected models are reevaluated. If more current certifications for the models exist, the models are reconfigured.

Interface Modeling

GnSNMPDev creates an interface model for every instance in the MIB-II Interface table. Interface models are instantiated and associated with the device during DX NetOps Spectrum modeling. They represent the physical or logical connections on a device.

The device model Interfaces tab in the Component Details panel shows all of the interfaces that DX NetOps Spectrum has discovered on a device. The view shows interface status (UP or DOWN) and other information.

Connections between devices can be mapped to the port level, which lets DX NetOps Spectrum isolate faults with more granularity. For example, if a port on a device goes down, an alarm is generated on the individual interface model rather than at the device level. Interface model statistics can be polled and logged, letting you monitor and manage device performance with detailed data.

Potential interface model types include the following types:

- Gen_If_Port
- Serial_If_Port
- VLAN_IF
- FrameRelayPort

If Frame Relay Manager is installed and the device supports either of the Frame Relay standard MIBs (RFC1315 or RFC2115), the DLCI circuits are modeled using the DLCI_port model type.

NOTE
For more information, see the Standards-Based Protocol Reference section.

If ATM Circuit Manager is installed and the device supports the ATM MIB RFC1695, the ATM logical connections are modeled using the ATMVclLink or ATMVplLink model types.

Note: For more information, see the ATM Circuit Manager section.

Application Modeling

When a device is modeled with GnSNMPDev, DX NetOps Spectrum creates application models to represent each of the standard (IETF) MIBs that the device supports. Application models are instantiated and are associated with the device during DX NetOps Spectrum modeling.

For example, GnSNMPDev intelligence detects that a modeled device performs routing functions (a routing MIB is present). A Routing Application model is created and associated with the device model. Non-routing devices are not

burdened with the functionality and attributes that are required to manage routers; each device model carries only the required functionality.

Additional support for standard or proprietary MIBs can be added to the GnSNMPDev model type by customizing the GnSNMPDev management module.

The Locater tab in OneClick lets you search for and access the application models that are associated with a given device model. Several predefined searches are available for application models, but you can also perform a search using custom criteria.

NOTE
For information about standard MIB applications and accessing their views in OneClick, see the Standards-Based Protocol Reference section and the Host System Resources Management section. For information about creating a search, see the OneClick Administration section.

Traps, Events, and Alarms

The following table summarizes the trap support that is available with the GnSNMPDev management module for the six generic traps:

Trap Name	OID	Variable Binding	Event Generated	Alarm Generated	Alarm Severity
coldStart	0.0	N/A	0x10306	N/A	N/A
warmStart	1.0	N/A	0x10307	N/A	N/A
linkDown	3.0	1.3.6.1.2.1.2.2.1.1 1.3.6.1.2.1.2.2.1.2 1.3.6.1.2.1.2.2.1.3 1.3.6.1.2.1.2.2.1.7 1.3.6.1.2.1.2.2.1.8	0x220002	0x220001	Yellow alarm on the device (can be configured per port); red alarm on the port
linkUp	2.0	1.3.6.1.2.1.2.2.1.1 1.3.6.1.2.1.2.2.1.2 1.3.6.1.2.1.2.2.1.3 1.3.6.1.2.1.2.2.1.7 1.3.6.1.2.1.2.2.1.8	0x220001	N/A	N/A
authenticationFailure	4.0	N/A	0x1030a	0x1030a	Yellow
egpNeighborLoss	5.0	1.3.6.1.2.1.8.5.1.2	0x1030b	0x1030b	Yellow

In addition, the GnSNMPDev model type supports various RFC and IEEE standard applications traps. This model type also supports any traps that are defined at the global level. You can enhance this support to include other traps and event processing.

NOTE
For more information about global traps, see the Event Configuration section.

Access the Device Certification Database Online

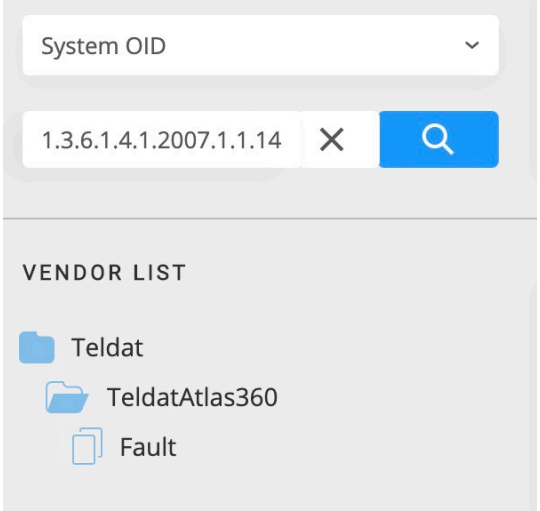
An application on the CA Technical Support website lets you search on all DX NetOps Spectrum certified devices. You can determine whether DX NetOps Spectrum supports a specific device model and filter by firmware version and release. You can also determine whether a device is supported with a Simple certification or an Enhanced certification.

Follow these steps:

1. Access the Device & Technology Certification page.
2. Click the 'Search All Certified Devices' link.

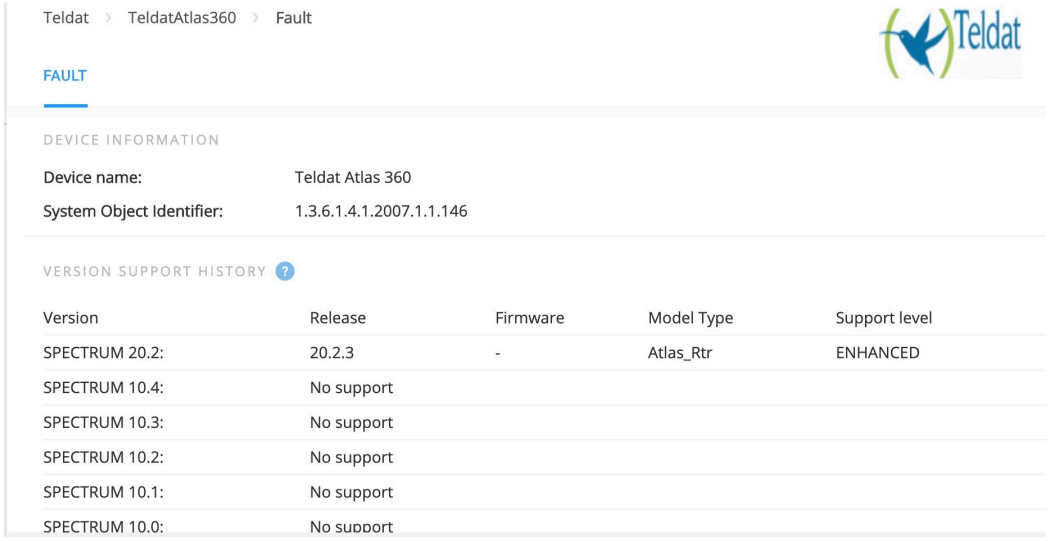
The Certification Web Database Search application appears.

3. Select Spectrum from the Product Line dropdown.



4. Complete the following search criteria fields as needed to locate your device:
   - **Certified Vendors**
     Corporations or organizations that manufacture one or more devices that DX NetOps Spectrum has certified. A vendor filter limits your search to all devices owned or acquired by the selected vendor.
   - **Keyword Search**
     Searches in the Device Type Name field of each device. A keyword search limits your search to all devices that contain the specific keyword in the Device Type Name field.
   - **System Object Identifier**
     Searches for a System Object Identifier, or a portion of the System Object Identifier. All devices containing the sequence you enter are returned.
     For example, 1.3.6.1.4.1.9.1.685 identifies the Cisco 1240AP device.

     NOTE
     Not all devices have a unique System Object Identifier. In addition, some devices lack a System Object Identifier.
   - **Support Level**
     Indicates the current level of DX NetOps Spectrum certification support. Two levels of certification support are available. For more information, see the Overview on Certification section.
5. Click the Search Database button to initiate a search based on your search criteria.
   Results are displayed, one line per device. Details at this level include the device name and model, System Object Identifier and Support Level.
6. Click a specific entry in the results table.
   Detailed information about the selected device appears, as shown:



Self-Certification

You can extend and customize the DX NetOps Spectrum simple or enhanced certification support. The following options are available:

- modifying attribute settings
- customizing identification with Device Certification
- managing MIBs and traps with MIB Tools
- developing new certifications

The subsequent sections discuss customizing identification with Device Certification, managing MIBs and traps with MIB Tools, and developing new certifications.

> **NOTE**
> For more information about customizing certification support, see the Event Configuration section, the Watches section, and the OneClick Customization section.

Adding Trap Support

DX NetOps Spectrum uses traps, events, and alarms to notify you about significant occurrences in your infrastructure. These terms apply to specific DX NetOps Spectrum entities, as described in the following list:

- Traps are alerts that are sent from SNMP-compliant devices. DX NetOps Spectrum receives traps and converts them into events for further processing.
- An alert is an unsolicited message that a managed node on a network sends. The management protocol affects the specific implementation of alerts. In general, DX NetOps Spectrum uses SNMP as the management protocol to communicate with devices on a network.
- An event indicates that something significant has occurred. Events are generated for observed behavior within DX NetOps Spectrum itself or in the managed environment. DX NetOps Spectrum events always occur in relation to a

model. When a managed element on the network generates an alert, it is mapped to a DX NetOps Spectrum event in the appropriate AlertMap file. The event is then generated with the event code specified in the AlertMap.

- An alarm indicates that a user-actionable, abnormal condition exists on a model. A model usually detects an abnormal condition when an event occurs and the EventDisp file states that an alarm is generated.

By default, the GnSNMPDev model type supports various traps, events, and alarms.

You can also add support for additional traps using the MIB Tools application and the Event Configuration application in OneClick. The high-level process is as follows:

1. Identify the MIB that contains the desired trap definitions.
2. In MIB Tools, import the MIB into the MIB Tools database.
3. Map the traps to events using MIB Tools. Specify the events that generate alarms and alarm severity.
   MIB Tools automatically creates and installs the appropriate event and alarm support files.
4. Launch Event Configuration directly from MIB Tools:
   a. In the Trap Support table, select the mapped traps whose events and alarms you want to configure.
   b. Edit traps for selected items in the trap support table.
5. Complete the configuration of the events and alarms in Event Configuration.
   For example, specify the symptoms, probable causes, and recommended actions for each alarm. The corresponding messages are displayed in OneClick when the alarms are generated.
   You can also add optional event processing for one or more events. For example, set up logging and create event rules that determine whether the event clears an alarm or generates another event.
   In addition, you can customize the default event message that is displayed in OneClick when the events are generated.

   NOTE
   For more information, see the Event Configuration section.

Watches to Monitor and Manage Model Conditions

You can create one or more watches for a particular model. A *watch* is a mechanism for adding thresholds for model attributes. Watches let you monitor network elements, such as routers, with a high level of detail. They also provide current data that can be used with other DX NetOps Spectrum tools in network analysis.

Set up a watch to monitor and analyze the changing internal and external attribute values of a model. Watches can include expressions that incorporate one or more attribute values. These attribute values, or an expression that is derived from these values, can then be measured against a defined threshold value. DX NetOps Spectrum evaluates the attribute values defined in a watch by polling the attributes when they are updated or when the watch value is read. Results can be used to generate events and alarms. Results can be logged for historical tracking and report information or sent to script files.

Keep in mind that watches can have an impact on network traffic and system resources. Delete watches that are no longer useful.

NOTE
For more information, see the Watches section.

Lock the Device Type Setting for a Device Model

You can lock the device type setting for a model so that the type is not reevaluated when you reconfigure a device. By default, the device type setting is not locked.

Follow these steps:

1. Locate the device model in the Topology tab of the Contents panel.

2. Select the device model whose device type setting you want to lock.
3. Click the Information tab in the Component Detail panel.
4. Expand the DX NetOps Spectrum Modeling Information subview.
5. Click 'set' in the Lock Device Type field, and select Yes.
   The device type setting is locked for the selected device model.

Lock Device Model Settings

You can lock settings for a device model. Locked settings are not reevaluated when you reconfigure a device. Both the device type setting and the model class setting for a device can be locked. By default, neither setting is locked.

Follow these steps:

1. Locate device models in the Topology tab of the Contents panel.
2. Select the device model whose device type or model class setting you want to lock.
3. Click the Information tab in the Component Detail panel.
4. Expand the DX NetOps Spectrum Modeling Information subview.
5. Take one or both of the following steps:
   – Click 'set' in the Lock Device Type field, and select Yes.
   – Click 'set' in the Lock Model Class field, and select Yes.
   The setting is locked for the selected device model.

Customizing Identification with Device Certification

This section describes the Device Certification component of OneClick. This section also discusses concepts and operations that are performed to customize the identity of a device using the Device Certification component of OneClick.

Device Certification in OneClick

The Device Certification component of OneClick lets you view, create, and edit Device Certification entries. DX NetOps Spectrum maps the System Object ID to the device type, model class, and model type. Device certification mappings appear in the Device Certification dialog.

Device certification entries let DX NetOps Spectrum initialize the device type, model type, and model class attributes on models during Discovery, Modeling, and device creation. You can create Device Certification entries for devices not directly supported in OneClick.

You can search, filter, and report on device models using the following attributes from the Device Certification list:

- Device Type attribute (0x23000e)
- Model Class attribute (0x11ee8)
- Model Type attributes (0x10000 for Modeltype_Name and 0x10001 for Modeltype_Handle)

These attributes provide a fine level of granularity when managing your network infrastructure.

Device Certification supports a distributed SpectroSERVER environment. Consistent device model identification occurs across a distributed deployment.

> **NOTE**
> To access and edit the device certification entries, log in as an administrator with read and write permissions.

Open the Device Certification Dialog

You can open the Device Certification dialog by taking one of the following steps:

- Open the Device Certification dialog from OneClick by selecting Tools, Utilities, Device Certification.
  The Device Certification dialog opens, displaying device type mappings for all modeled devices.
- Open the Device Certification dialog within the context of a device model. Select the model in the Explorer tab or the List tab of the Navigation panel, or in the Topology tab of the Contents panel. Right-click the selected device and select Utilities, Device Certification.
  The Device Certification dialog opens. The entry for the selected device type is highlighted.

About the Device Certification Dialog

The Device Certification dialog lets you maintain a custom list of system object identifiers and their corresponding device type name, model type, and model class. When you create or modify one of these entries, the corresponding attribute for all device models with the given system object ID is set to your customized value. This setting is applied to both existing and future device models. Used with the GnSNMPDev model type, this feature lets you model and monitor any SNMP-compliant device in the network. Devices that lack a specific DX NetOps Spectrum management module can also be modeled.

The Identification list in the Device Certification dialog also contains unregistered devices. An "unregistered" device has been modeled using Discovery or Model by IP. Such a device has system object IDs but lacks a device type name, model type, or model class.

You can use the unregistered devices in the Identification list to set up entries for all devices that are modeled with GnSNMPDev. Instead of trying to determine which devices use the GnSNMPDev model type, first model the devices. Once the devices are modeled, their system object IDs are added to the Identification list. You can then filter and sort the list and specify device type names for the unregistered devices that are modeled with GnSNMPDev.

> **NOTE**
> The mappings in the Device Certification dialog are preserved during upgrades and database migrations.

You can open the Device Certification dialog from the OneClick Tools menu or from a device model context:

- Select Tools, Utilities, Device Certification.
  The Device Certification dialog displays device type mappings for all modeled devices.
- Right-click a device model in OneClick. Find the model in the Explorer tab or the List tab of the Navigation panel, or in the Topology tab of the Contents panel. Select Utilities, Device Certification.
  The Device Certification dialog highlights the entry for the selected device type.

Device Certification Table

The Device Certification dialog lists the mappings of the device type name to System Object Identifier (sysObjectID or system OID), model classes, and model types. This list includes all standard DX NetOps Spectrum predefined mappings, all user-defined mappings, and any unregistered mappings. The list *does not* include mappings that have specialized device type name handling.

The Device Certification table displays the following information about each mapping:

- **Vendor Name**
  Displays the name of the company that manufactures the device, such as Cisco Systems.
- **System Object ID**
  Displays the MIB II sysObjectID entry that was retrieved from the device.
- **Device Type Name**

Displays the Device Type value that is mapped to the associated system OID.

- **Model Type**
  Identifies the name of the specific model type. If the system OID is supported with a simple certification (the system OID is *not* associated with a specific model type), displays “GnSNMPDev”. By default, this column is not visible.
- **Model Class**
  Identifies the model class (such as Router, Switch-Router, or Port) for the device model. If no model class is mapped to the system OID, "Auto" is displayed.
- **Modification**
  Identifies the mappings that have been modified in the current DC session.
- **Support Level**
  Identifies whether the device for the system OID has a simple or enhanced certification (MM). If the system OID is modeled with the GnSNMPDev model type, the support level is "Simple." If the system OID has a specific certification, the support level is "Enhanced."
- **Author**
  Identifies the user who created the mapping. This column is hidden by default.

You can modify the table display using the standard OneClick table preferences and column sorting methods. You can export the data in the Device Certification table to a file in either a comma-separated (.CSV), tab-delimited (.txt), or web page (.HTML) format.

NOTE
For more information, see the Using OneClick section.

Search Device Certification Mappings Using Filters

You can search for specific text or numeric strings in the Device Certification table by typing them in the Filter field. As you type in the Filter field, only the mappings that contain matching character strings appear in the table.

NOTE
The Filter field searches only the *visible* columns.

Use this feature to search for the following attributes:

- **Vendor Name:** Type the name of a specific vendor to display all supported system OIDs for that vendor. This information can help you determine whether DX NetOps Spectrum supports a specific model.
- **Custom mappings:** Enter "custom" in the Filter field to view only the mappings that you or someone else has modified using Device Certification.
- **Unregistered devices:** Enter "unregistered" in the Filter field to view only mappings of unregistered devices in the table.

Device Certification Changes

Once you have created or modified a mapping and have applied the changes, all device models in your distributed SpectroSERVER (DSS) environment is updated if they have Device Certification mapping changes. The updated device type names, model classes, and model types appear in the Topology views, the Navigation panel, and the List views for all landscapes.

The DX NetOps Spectrum modeling catalog is also updated so that all future device models with this system OID are assigned the corresponding device type, model class, and model type values.

Device Mappings

Device Certification categorizes device mappings into the following types:

- **CA mappings:** The predefined mappings that are included with DX NetOps Spectrum.
- **Custom mappings:** The mappings that are created or customized using the Device Certification utility.
- **Unregistered mappings:** Entries that do not have a mapping.

  NOTE
  The mapping type appears in the Author column in the Device Certification dialog.

Custom Device Type Mappings

You can customize any mapping, or you can create new mappings to improve any of the following administrative settings:

- Modify device type names to be more descriptive.
- Modify model classes to more accurately reflect the type of a device.
- Modify the model type to acquire functionality that is associated with a more appropriate model type.
- Add new mappings to accommodate devices.
- Assign device names to unregistered devices that DX NetOps Spectrum has identified on your network.

Modify Device Certification Entries

You can modify the device type name of an existing Device Certification mapping. You can also map the model type name and model class of a device.

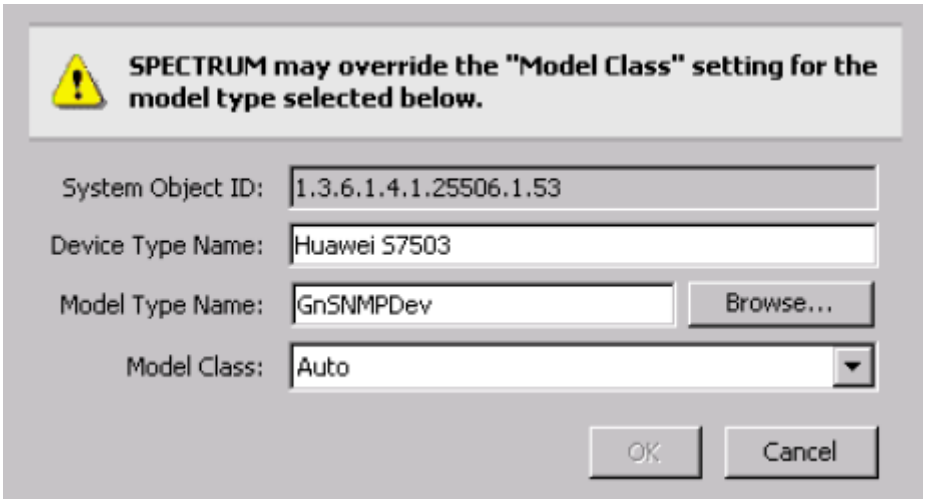
If you modify the device type name of an existing Device Certification mapping, or if you map the device OID to the appropriate model type name or model class, you create a custom mapping. Custom mappings override original mappings.

However, the original mapping remains intact on the system. If you delete a custom mapping with a modified device type name, model type name, or model class value, the original mapping reappears in the table.

Follow these steps:

1. Open the Device Certification dialog.
   The Device Certification dialog displays the device certification mappings on the Identification tab.
2. Select an entry to modify, and click the edit
   icon 
   The DC: Edit Mapping dialog opens.

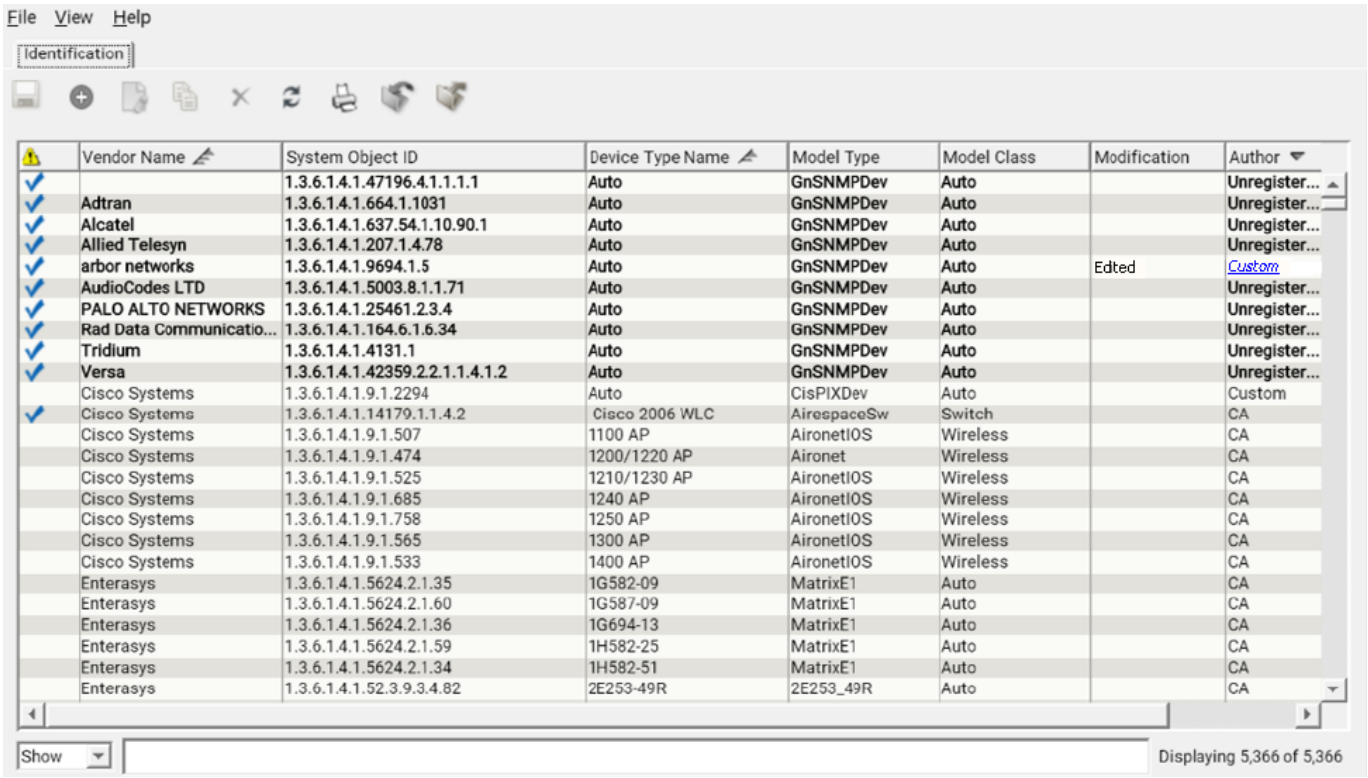
   NOTE
   For some models, DX NetOps Spectrum overrides your custom settings. In these cases, a warning message appears.



– Edit the following fields, and click OK.

  - Device Type Name
  - Model Type Name
  - Model Class

  Your changes to the selected model appear in the table on the Identification tab.



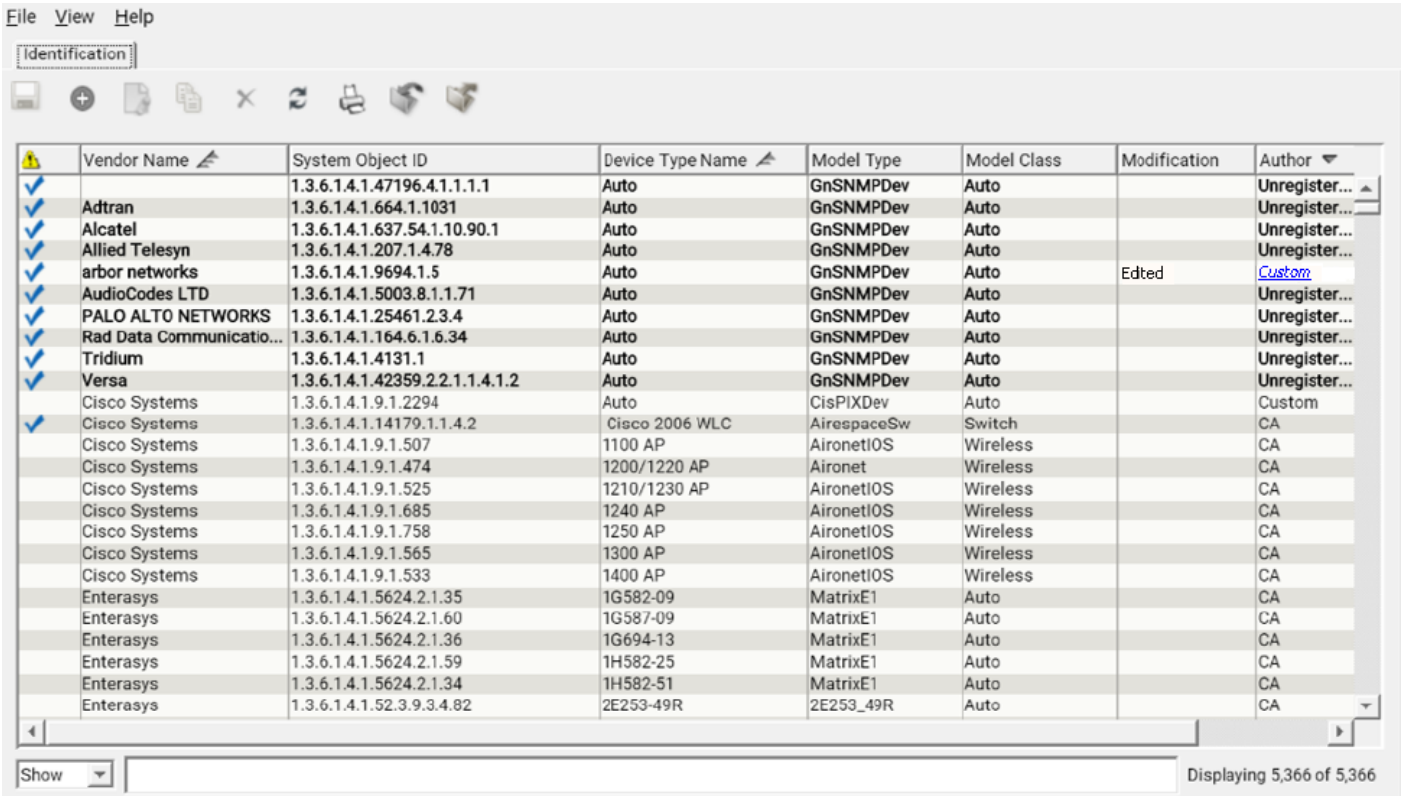
3. Click Save.
   Your changes are saved and are immediately applied to all device models. When complete, the DC: Operation Results dialog displays the results, either successful or unsuccessful, for each landscape in a DSS environment.
4. Click Close.
   Your selected Device Certification mapping is updated.

View Default Values Masked by Custom Device Certifications

Default device certifications are set up during the initial DX NetOps Spectrum modeling. When you customize a device certification mapping, your values mask the default values. After customizing your mapping values, you can view the default values that your custom mapping is obscuring. Viewing this information can be useful when determining whether your custom values are still required.

Follow these steps:

1. Open the Device Certification dialog.
   The Device Certification dialog displays the device certification mappings on the Identification tab.
2. Locate the custom device certification, and click the 'Custom' link.



   The DC: Custom Mapping Details dialog opens. This dialog displays the custom values and the default values for each modified device certification mapping.

Copy Device Certification Mappings to Create New Mappings

You can create a new device certification mapping by copying an existing device certification mapping. Using the existing mapping can be more efficient when your new mapping is very similar, because the System Object ID, device type, model class, and model type values are prefilled.

Follow these steps:

1. Open the Device Certification dialog.
   The Device Certification dialog displays the device certification mappings on the Identification tab.
2. Select the entry to copy and click the copy
   icon 
   The DC: Copy Mapping dialog opens.

NOTE
For some models, DX NetOps Spectrum overrides your custom settings. In these cases, a warning message appears.

3. Edit the following fields, as needed, and click OK.
   - System Object ID
   - Device Type Name
   - Model Type Name
   - Model Class

   Your new device certification mapping appears in the table on the Identification tab. The Modification column specifies "New," and the Author column specifies "Custom."
4. Click Save.
5. Click Close.
   Your new device certification mapping is created and is added to the Device Certification table.

Your changes are saved and are immediately applied to all device models. When complete, the DC: Operation Results dialog displays the results, either successful or unsuccessful, for each landscape in a DSS environment.

Map Unregistered Devices

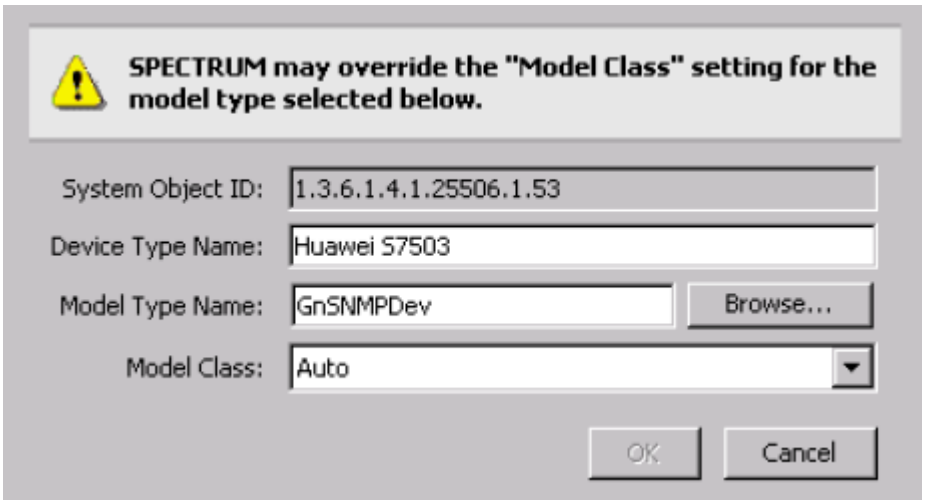
Unregistered devices lack a matching device type, model class, or model type entry. Any unregistered devices appear in bold in the Device Certification dialog, and "Unregistered" appears in the Author column.

To map unregistered devices, know the system OID of the unregistered devices on your network.

Follow these steps:

1. Open the Device Certification dialog.
   The Device Certification dialog displays the device certification mappings on the Identification tab.
2. Select the unregistered device entry to be mapped, and click the edit icon 
   The DC: Edit Mapping dialog opens.

   NOTE
   For some models, DX NetOps Spectrum overrides your custom settings. In these cases, a warning message appears.



- Edit the following fields, as needed, and click OK.
  - Device Type Name
  - Model Type Name
  - Model Class

  Your changes to the selected model appear in the table on the Identification tab. "Custom" appears in the Author column, and "P" appears in the Modification column.

OneClick Views

A device that is modeled with the GnSNMPDev model type offers access to all OneClick views, such as Information, Performance, and Alarms.

NOTE
For more information about OneClick views, see the Using OneClick section.

Delete Custom Device Certification Mappings

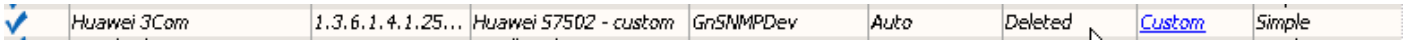
You can delete custom mappings from the Device Certification table. However, default CA Technologies mappings cannot be deleted. When you delete a custom mapping that overrides a default mapping, the default mapping displays after the delete operation completes.

Follow these steps:

1. Open the Device Certification dialog.
   The Device Certification dialog displays the device certification mappings on the Identification tab.
2. Select the entry to delete and click the delete icon 

   NOTE
   The Delete button is not available if the selected entry cannot be removed. For example, a default CA Technologies Device Certification cannot be deleted.

   The entry is flagged for removal from the list.



3. Click Save.
   Your changes are saved and are immediately applied to all device models. When complete, the DC: Operation Results dialog displays the results, either successful or unsuccessful, for each landscape in a DSS environment.
4. Click Close.
   Your selected Device Certification mapping is deleted.

Import the Uncertified Devices

From 10.4.1, you can import the uncertified devices and certify them yourself instead of waiting for the next version of the DX NetOps Spectrum to release a new list of certified devices.

Follow these steps:

1. Open the Device Certification dialog.
   The Device Certification dialog displays the device certification mappings on the Identification tab.
2. Click the **Import the file** icon 
   The Open File dialog is displayed.

3. Browse and select the required CSV (comma-separated values) file.
   Importing the file, update all the unregistered sysoids and update model type of existing sysoids. The updated list of devices is shown in the Device Certification dialog.

   NOTE
   For more information, see **Self Certify the Unregistered Devices** section in Uncertified Devices.

   You have successfully imported the unregistered devices.

Distributed SpectroSERVER Support

The Device Certification utility supports a distributed SpectroSERVER (DSS) environment. Device Certification detects multiple primary SpectroSERVERs in a DX NetOps Spectrum environment and alerts you when it cannot communicate with a SpectroSERVER. Device Certification queries all SpectroSERVERs for Device Certification table entries. Conflicts among SpectroSERVERs for user-specified mappings are detected.

NOTE
When you open the Device Certification utility in a DSS environment, a warning message appears if any of the primary SpectroSERVERs are down. Any unavailable SpectroSERVERs do not receive the Device Certification mapping changes that you make. Custom Device Certification mapping conflicts occur after all SpectroSERVERs are back online.

Resolve Device Certification Mapping Conflicts

Conflicts in Device Certification mappings can occur in an environment when a sysObjectID has more than one device type, model class, or model name defined. Conflicts usually happen when:

- **User customizations are mismatched across SpectroSERVERs** -- This situation can occur when one or more SpectroSERVERs are down prior to starting Device Certification. This situation can also occur when one or more SpectroSERVERs go down after starting Device Certification but before applying changes to custom mappings.
- **Predefined mappings are mismatched across SpectroSERVERs** -- This situation occurs when the device certification mappings provided by DX NetOps Spectrum are updated on some SpectroSERVERs but not others. For example, bringing a new SpectroSERVER online that has a more recent release of DX NetOps Spectrum can lead to differences in the predefined mappings between SpectroSERVERs.

For mismatched predefined mappings, verify the software installation on all SpectroSERVERs so that they use the same device certification table. You can resolve conflicts with customized mappings using the Device Certification dialog.

Follow these steps:

1. Open the Device Certification dialog.
   If conflicting Device Certification mappings are detected when you start the Device Certification utility, a Conflicts Encountered dialog opens.

   WARNING
   Resolve the conflicts so that Device Certification can open.
2. Click the Resolve Conflicts button.
   The Resolve Conflicts dialog opens.
3. Select the appropriate name from the Device Type Name drop-down list for each System Object ID, then click OK.
   The Device Certification mapping conflicts are resolved. The Device Certification dialog displays the device certification mappings on the Identification tab.

   NOTE
   If the Device Certification utility identifies a Device Certification entry that is present on some, but not all, of the servers in the DSS environment, the entry is automatically applied to all servers that lack it. This condition is different from a conflicting entry condition.

Device Certification Changes are Not Saved

Symptom:

I updated my Device Certification mappings, but some of my device models were not updated when I clicked Save. Now the Save button is disabled. Why were some devices not updated, and how can I apply my changes?

Solution:

DX NetOps Spectrum is unable to successfully save the updated Device Certification mappings on the first save attempt in the following cases:

- **A server in a distributed SpectroSERVER (DSS) environment cannot be contacted and does not receive the mapping update.**
  In this case, DX NetOps Spectrum warns you that one or more servers were down before the mappings were applied, or that some of the mappings were not applied on one or more servers.
  To resolve this problem, take the following steps:
  a. Close the Device Certification dialog.
  b. Resolve the communication issue with the server.
  c. Open the Device Certification dialog.
  The Device Certification dialog notifies you that conflicts exist and you must resolve them.
  d. Resolve all conflicts, then click Apply.
  The server synchronizes with your Device Certification mappings and reapplies them to all affected models.
- **Updated mappings are applied to all SpectroSERVERs, but some device models did not reevaluate their Device Certifications.**
  This situation can occur when the model classes or device types are locked. Network failure, stopping a server, or losing contact with the Device Certification client during an update can also cause this scenario. In this case, the SpectroSERVERs are properly updated -- the *device models* did not update -- so the Device Certification dialog does not notify you about conflicts.
  To force individual device models to reevaluate their Device Certification mappings, take the following steps:
  a. Use a Locater search to find all device models that did not update.

  NOTE
  You can check the Device Type Name or Model Class columns for device models that did not update.

  b. Verify that the device types and model classes are not locked.
  c. Select all affected device models and reconfigure the models.

Change the Model Type for a Single Device Type Using a Script

You can use the NewMM.pl post-installation script to automatically change the model type for a single device type. Many key attributes, relationships, and other elements are preserved.

This procedure changes the model type for all models with the specified system Object ID *and* the specified starting model type.

WARNING
Do not perform this procedure until you modify the model type mapping for the device type in the Device Certification utility. Otherwise, your changes are not communicated to the SpectroSERVER database and you see unexpected alarms.

Follow these steps:

1. Verify that the SpectroSERVER is running.
2. Run the following command from the <*$SPECROOT*>/Install-Tools/PostInstall/ directory:

```
NewMM.pl -m
```

NOTE
On Windows, run all necessary scripts from a bash shell. They do not run as expected from a DOS command prompt.

3. Enter the host name or IP address of the VNM and press Enter.
4. Enter the SpectroSERVER landscape handle when prompted.
5. Enter the system Object ID for the model when prompted.
6. Enter the current model type of the model when prompted.
7. Enter the new model type when prompted.
   The model type is changed.
   The log file, NewMM_Log_<*DATE*>, is created in the <*$SPECROOT*>/Install-Tools/PostInstall/ directory.

   NOTE
   Verify that the model type converted successfully by checking the log file, NewMM_Log_<*DATE*>.

Device Certification and Fault-Tolerant Environments

If you are working in a fault-tolerant environment, Device Certification differentiates between a primary and a backup server. For Device Certification to operate, it must be able to connect to the primary SpectroSERVER. The application cannot connect to the backup server.

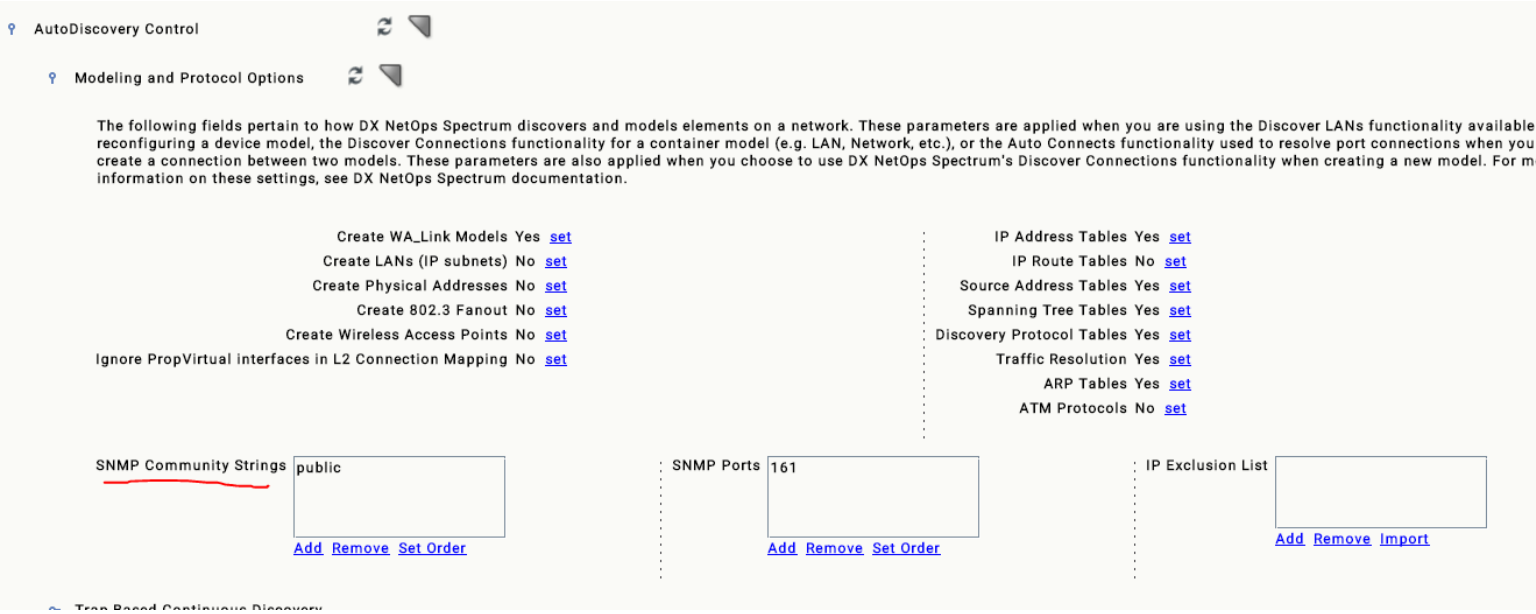
The backup SpectroSERVER obtains the Device Identifier List update and device model updates from the primary SpectroSERVER during the Online Backup procedure.

NOTE
For more information, see the Distributed SpectroSERVER Administration section.

Uncertified Devices

The current release simplifies the certification of uncertified OIDs to an existing model type. The device certification tool has a new option to import CSV files. The CSV file must have the mapping of the OIDs to be certified and model type. Importing the CSV updates the device certification tool with the new OID and corresponding model type. Previously, the device certification tool provided a consolidated list of all certified and uncertified SyOIDs. With this improved search option in this release, you can now fetch only the uncertified devices and self-certify those devices.

Ensure that you add the **SNMP Community String** options of the GnSNMP devices under the VNM Model, AutoDiscovery Control, Modeling and Protocol Options.

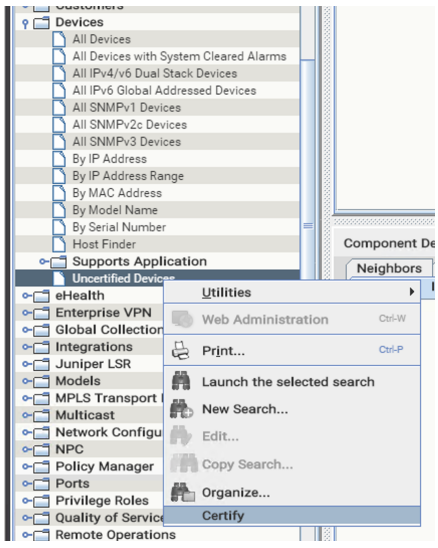


Self-Certify the Uncertified Devices

From 10.4.1, you can import the uncertified devices and can certify them yourself instead of waiting for the next version of DX NetOps Spectrum to release a new list of certified devices.

Follow these steps:

1. Ensure that you import the uncertified devices from the CSV file provided by Product Management.
   For more information, see the **Import the Uncertified Devices** section in Device Mapping.
2. Right-click the **Uncertified Devices** option in Locater and select **Certify**.



   DX NetOps Spectrum lists the uncertified devices in the dialog and prompts you to confirm the certification.
3. Click **Yes** to certify.
   DX NetOps Spectrum identifies all the GnSNMP devices for which the model type has been updated with the import, deletes them, and recreates.

   NOTE
   Manually created connections are lost and must be recreated.

Managing MIBs and Traps With MIB Tools

This section explains the MIB Tools utility of OneClick. This section also describes various concepts of a MIB and operations that are performed to manage MIBs and traps using the MIB Tools utility.

The MIB Tools Utility

The MIB Tools utility lets you compile, import, and browse Management Information Bases (MIBs). In addition, this utility can issue SNMP requests to network elements and can customize the mapping of MIB objects and traps in DX NetOps Spectrum. Use the MIB Tools to create, customize, and troubleshoot network element management in DX NetOps Spectrum.

SNMP and MIBs form the structure of network element management in DX NetOps Spectrum. DX NetOps Spectrum communicates with modeled network entities using SNMP. A MIB is a type of network device database that represents a device as a hierarchical collection of objects. A MIB object represents an individual unit of information in a MIB, such as device uptime. MIBs themselves are text files with special syntax.

MIB Tools include two self-certification tools. First, MIB Tools lets you map MIB objects to attributes in the DX NetOps Spectrum database. You can create OneClick views to display the values of those MIB objects, create Watches on the attributes, and set thresholds to send alarms. Second, the MIB Tools utility lets you add support for traps that your devices send.

DX NetOps Spectrum complies with the following RFCs regarding MIBs and SNMP:

- RFC 1155: Structure and Identification of Management Information for TCP/IP-based Internets
- RFC 1157: A Simple Network Management Protocol
- RFC 1213: Management Information Base for Network Management of TCP/IP-based Internets

DX NetOps Spectrum uses the following standard MIBs for link discovery to establish connectivity with neighboring devices:

- LLDP MIB
- Bridge MIB
- Q-Bridge MIB (if VLAN is configured)
- IP MIB (This MIB takes care of IP-Forwarding tables also)

NOTE

Information!

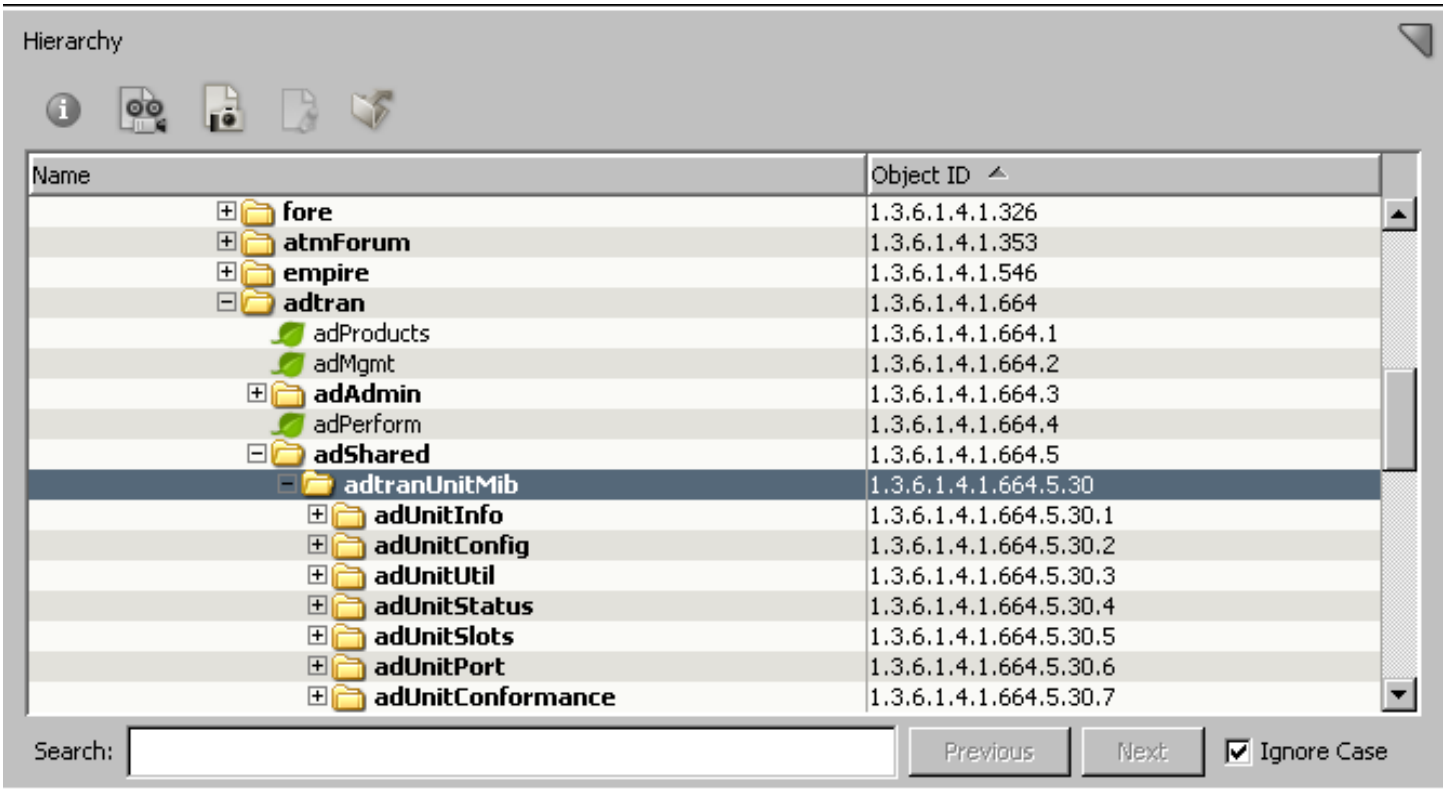
- When a device is certified in Spectrum, it is verified that these standard MIBs are supported by the respective device. If yes, then the connections with neighboring devices get established automatically.
- If these MIBs are supported by the device and still relevant information is not available in those MIBs, then it might be a problem with device configuration or vendor issue.
- If device populates these MIBs information, and yet the connections are not being established then further troubleshooting is required from Spectrum end.
- If these MIBs are not supported by the device and relevant information is available in proprietary MIB (like CDP MIB), then you need to add specific intelligence for these types of devices as part of the enhanced certification.

How a MIB Is Organized

The International Standards Organization (ISO) supplies a standard tree format for the organization of network device management information. This tree structure branches out into subtrees that are organized into branches (groups of related information) and leaves (the individual pieces of information, or objects).

Each layer of this tree is numerically encoded. A unique number, an Object Identifier (OID), identifies each group and object. This identifier lets an SNMP agent locate the object in a device MIB. An ASCII name is also assigned to each branch or OID to identify management objects.

The following image illustrates MIB objects in the MIB Tools Hierarchy with the name and OID. MIB Tools uses folders, acorns, and leaf icons to show branches, traps, and objects within a MIB. Each folder in the display indicates that more objects are contained in that level of the tree structure.



MIB Tools Database

MIB Tools maintains a MIB database on the OneClick web server. The default database consists of standard and proprietary MIBs. You can add MIBs to this database by importing them.

OneClick MIB Tools Overview

MIB Tools is a multifunctional MIB utility. Use it to browse MIBs, issue SNMP requests to network elements, import MIBs, and add MIB object and trap mapping support to DX NetOps Spectrum. You can use MIB Tools to retrieve supported information directly from a given device to aid in troubleshooting and managing that device. You can customize your DX NetOps Spectrum network management environment by using MIB Tools to import the MIBs of network elements that are not yet supported in DX NetOps Spectrum.

The MIB Tools utility lets you complete the following tasks:

- Compile and import MIBs into the MIB Tools database.
- Browse MIBs for details of MIB objects and traps.
- Directly query and set values for MIB objects of network elements.
- Export MIB query result values for use in troubleshooting and creating simulations.
  You can export the data that is displayed in MIB Tools into several different file formats. You can export data from the Results, Attribute Support, and Trap Support tables.
- Create custom network element support in DX NetOps Spectrum by mapping new traps and MIB objects.
- Delete custom MIBs from MIB Tools database. Standard or proprietary MIBs cannot be deleted.

Start MIB Tools

You can start the MIB Tools utility from the Tools menu. Or you can start it within the context of a selected device model.

To open the MIB Tools utility without the context of a specific device model, click Tools, Utilities, MIB Tools without selecting a device model. The MIB Tools dialog shows the progress of retrieving and loading the MIB Tools database.

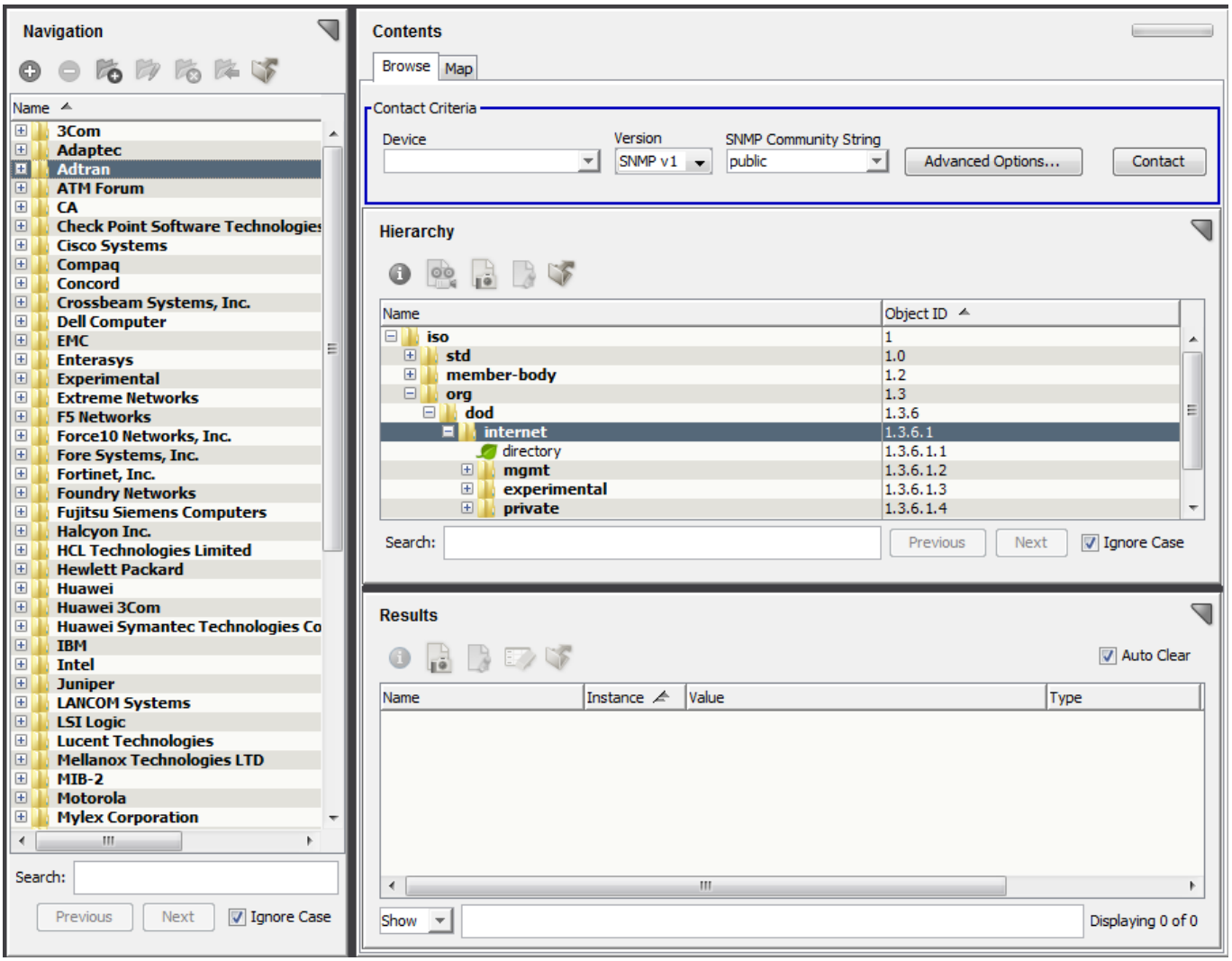
You can also start MIB Tools in the context of a specific device model. This method lets you communicate with the device and perform SNMP queries on the model, whose contact criteria are automatically displayed.

Follow these steps:

1. In the OneClick Console, locate the device model you want to investigate with MIB Tools.
   Locate a model in either the Explorer tab of the Navigation Panel or the Topology tab of the Contents panel.
2. Right-click the device model and click Utilities, MIB Tools, or click Tools, Utilities, MIB Tools from the main menu.
   The MIB Tools utility opens.
   The Contact Criteria display the SNMP contact information for the selected device.
   MIB Tools attempts to contact the device.
   If the attempt fails, an error message appears.
   If the attempt succeeds, the Contact Status indicator turns green.
   A status dialog shows the progress of retrieving and loading the MIB Tools database.

MIB Tools User Interface

The MIB Tools user interface has two panes that let you find MIBs and view MIB information: the Navigation pane and the Contents pane.



Locate and select MIBs in the Navigation pane. You can view a list of the compiled MIBs in the MIB Tools database. By default, MIBs are organized by vendor and are displayed alphabetically. Sorting is supported on table columns. You can also delete custom MIBs that you select in the Navigation pane.

The Contents panel has two tabs:

- **Browse tab**
  Lets you browse MIBs in the MIB Tools database and query devices on your network to obtain or set MIB object values. The Browse tab has three main sections: Contact Criteria, Hierarchy, and Results.
  Perform the following tasks from the Browse tab:
  - Contact a device.
  - View the name, object ID, or access type of an MIB object in the Hierarchy section.
  - View the results of a query in the Results section.
- **Map tab**
  Identifies the objects and traps for a selected MIB that DX NetOps Spectrum supports. You can see the DX NetOps Spectrum Attribute ID for an object and the Event code for a trap. Unsupported objects lack an Attribute ID or Event code.
  Perform the following tasks from the Map tab:

- Create attributes in the DX NetOps Spectrum database from MIB objects.
- Map traps in DX NetOps Spectrum.
- Identify the MIB objects that have been previously mapped to attributes in the DX NetOps Spectrum database using MIB Tools.
- Identify all trap support in DX NetOps Spectrum.

MIB Tree Hierarchy Table

When you select a MIB in the Navigation panel, its details appear in the Hierarchy tree table in the Browse tab of the Contents panel.

The default view of the Hierarchy tree table displays the entire MIB Tools database starting at the iso branch.

Browse MIBs

You can browse the MIB Tools database and view detailed information for each group, object, and trap. For trap objects, you can view variable binding details. To browse an individual MIB, select the MIB in the Navigation panel and use the Hierarchy tree table to navigate the MIB folders, groups, objects, and traps.

Search MIBs

You can search the Hierarchy tree table for a text string by entering it into the Search field. The Next and Previous buttons let you scroll through each successive instance of the search string.

The Hierarchy tree table displays the following information:

- **Name**
  Displays the name of the MIB object.
- **Object ID**
  Displays the Object ID of the MIB object.
- **Access**
  (Hidden by default) Displays the access type for the object. The type is read-only, read/write, read-create, or not-accessible.

NOTE
For more information, see the Using OneClick section.

The Hierarchy toolbar provides the following functionality:

Button	Description
	**Information/ Component Detail button**: Opens a dialog with details about a selected item. This information is taken directly from the MIB.
	**Query button:** Retrieves a subtree of management values using SNMP GET_NEXT requests. All of the objects returned have an OID that is prefixed by the OID of the branch that you select for the query.
	**GET button:** Performs an SNMP GET of a selected scalar object. If a scalar object is not selected, lets you build an SNMP GET or SNMP GET_NEXT request.
	**SET button:** Sets the value of the selected MIB object on a specific device.
	**Export button**: Exports the table contents to an external file

Attribute Support Table

When you select a MIB in the Navigation panel, its attribute support details appear in the Attribute Support table. Find this table in the Map tab of the Contents panel. The Attribute Support table displays information for MIB objects and DX NetOps Spectrum attribute support.

> **NOTE**
> The table does not reflect the DX NetOps Spectrum attribute support from the DX NetOps Spectrum model type catalog. Only the DX NetOps Spectrum attribute support that is created using MIB Tools is included.

The Attribute Support table displays the following information:

- **Name**
  Is the name of the object in the MIB.
- **Object ID**
  Is the object ID in the MIB.
- **Attribute ID**
  Specifies the attribute ID value. If the object is not supported in DX NetOps Spectrum, this field is blank. The word 'Conflict' can appear to indicate a conflict with the assigned attribute ID for the object in a DSS environment.
- **Landscapes**
  Indicates whether the attribute is supported on some, all, or none of your landscapes.
- **Needs Update**
  Indicates whether the attribute requires an update. For example, if the enumerations in the MIB do not match entries in the DX NetOps Spectrum database, the entry needs an update. A checkmark indicates that an update is needed.

The toolbar in the Attribute Support table includes the following functionality:

Button	Description
	**Information/ Component Detail button**: Opens a dialog with details about a selected item. This information is taken directly from the MIB.
	**Create Attributes button:** Creates DX NetOps Spectrum Attribute IDs for the objects that are selected and that lack support in DX NetOps Spectrum. If you do not select any objects in the Attribute Support table, and no attributes are currently mapped, creates attributes for *all* objects in the table. This button is disabled if the selected entry is already mapped, or if no items are selected and any of the entries are already mapped.
	**Export button**: Exports the table contents to an external file

Trap Support Table

When you select a MIB in the Navigation panel, its trap support details appear in the Trap Support table in the Map tab of the Contents panel.

The Trap Support table lets you view the traps defined in the MIB selected in the Navigation panel and all default and custom DX NetOps Spectrum event codes mapped to them.

The Trap Support table displays the following trap and DX NetOps Spectrum event information:

- **Name**
  Specifies the name of the trap in the MIB.
- **Object ID**
  Specifies the trap object ID in the MIB.
- **Event Code**
  Specifies the event code. If the trap is available for only select DX NetOps Spectrum models, the event code appears as ‘Partial.’ If the trap has different event codes mapped on different SpectroSERVERs in a DSS environment, the event code appears as ‘Conflict.’
- **Landscapes**
  Indicates if the attribute is supported on some, all, or none of your landscapes.
- **Event Type**
  Indicates whether the trap has a default mapping, a custom mapping, or both. Default mappings show ‘CA’ in this column; custom mappings show ‘Custom.’ However, if a custom trap mapping is obscuring, or shadowing, a default trap mapping, this column shows a ‘Custom’ link. Click 'Custom' to see the disposition details of the custom mappings.

The toolbar in the Trap Support table includes the following functionality:

Button	Description
	**Information/ Component Detail button**: Opens a dialog with details about a selected item. This information is taken directly from the MIB.
	**Map Traps button:** Maps traps for the selected item or items. If no objects in the Trap Support table are selected and no traps are mapped, creates traps for *all* objects in the table. Does not apply if the selected entry is already mapped, or if no items are selected that lack mappings.
	**Remove Traps button:** Removes the custom trap mappings from the selected item or items in the Trap Support table. When you unmap a trap from an event code, the corresponding "EvFormat" and "Pcause" files are deleted, and that event code is deleted from the database.
	**Edit Traps button:** Opens the Event Configuration application, which lets you edit the mapped trap selected in the Trap Support table. Does not apply to unmapped traps, whose entries lack event codes. For more information, see the Event Configuration section.
	**Export button**: Exports the table contents to an external file
Remap All Conflicts	**Remap All Conflicts check box:** Remaps traps that are partially supported in DX NetOps Spectrum, or that have inconsistent, conflicting support across SpectroSERVERs in a DSS environment. Remapping traps makes the traps available to all model types on all SpectroSERVERs.

Import and Export MIBs

You can add MIBs to the MIB Tools database individually using the MIB Import feature. You can also import multiple MIB files using a script at the command line. Import MIBs into the MIB Tools database for the following reasons:

- To view new MIB objects in a MIB that is not imported.
- To retrieve MIB objects from a device whose MIB is not imported.
- To build OneClick views or create Watches based on MIB objects that are not already supported in DX NetOps Spectrum.

> **NOTE**
> Importing MIBs is only the first step to using them in OneClick. The second step involves mapping the MIB objects.

You can also export data that is displayed in MIB Tools for use outside of MIB Tools and OneClick. You can export data that is displayed in the Results, Attribute Support, and Trap Support tables into several different file formats.

Import Individual MIBs

You can import individual MIBs into the MIB Tools database. The MIBs that you import are stored in the following directory on the OneClick web server:

```
<$SPECROOT>/MibDatabase/userContrib
```

When you import a new MIB using the MIB Tools, DX NetOps Spectrum stores it as a custom MIB.

To import a MIB into MIB Tools, the MIB file must be on a file system that is accessible from the OneClick Console. You can only compile text-formatted MIB files. Files in Microsoft Word or rich text formats (containing control characters) cannot be compiled and are ignored.

> **NOTE**
> If you have many MIBs to import, you can import multiple files in bulk using the BulkMibImport command. The MIB files that are referenced by MIB Tools can only contain a single MIB. This restriction includes the MIB that is being compiled and any dependent or imported MIBs that are in the same directory. If a referenced file contains multiple MIBs, break each MIB into a separate file.

When you import a MIB file, MIB Tools recursively checks the MIB file for any IMPORTS statements that reference other MIBs. MIB Tools identifies any dependent or imported MIBs in files that are in the same directory as the MIB that is being compiled. As long as they are in the same directory, the MIB files that resolve IMPORTS statements are also compiled and are placed in the following directory on the OneClick web server:

```
<$SPECROOT>/MibDatabase/Dependent
```

These files are therefore available for subsequent import requests.

After you locate and compile a referenced MIB, the MIB Tools utility no longer has to locate and compile it each time another MIB file references it.

Follow these steps:

1. Click Utilities, MIB Tools from the main menu.
   MIB Tools opens.
2. On the Navigation panel, click the Add MIB
   icon 
   The ‘MIB Tools: Add MIB’ dialog opens.
3. Click Browse to locate and select the file that contains the MIB that you want to import. Or manually enter the complete path and filename in the MIB File Name field
4. Click Open.
5. Click Compile.

A message appears in the Compiler section of the dialog, relating to the status of the compile request.
On success, a message states that the file was successfully compiled.
Otherwise, errors or warnings that were generated during the compilation appear. You cannot import a MIB that contains errors. Click Show Editor to edit the MIB file and correct any errors.

6. Once the MIB compiles successfully, click Add to add the MIB and keep the 'MIB Tools: Add MIB' dialog open so that you can add additional MIBs, or click Add & Close to add the MIB and close the dialog.
   The MIB is added to the MIB Tools database, the 'MIB Tools: Add MIB' dialog closes, and the MIB is added to the list in the Navigation panel. If the MIB references a new vendor, a new folder appears for that vendor.

   NOTE
   If you import an updated copy of an existing standard MIB, the MIB Tools adds that MIB as a unique custom MIB under the same vendor folder by appending MIB-MODULE-NAME<n>. For example, when you update the ADTRAN-AOS MIB and import it, MIB Tools adds it as ADTRAN-AOS1 after the existing ADTRAN-AOS MIB. You can delete the ADTRAN-AOS1 MIB as it is a custom MIB.

Delete Individual MIBs

You can delete previously imported custom MIBs from the MIB Tools database. When you delete an MIB all the corresponding mapped attributes and events that are associated with traps remain mapped.

Follow these steps:

1. Click Utilities, MIB Tools from the main menu.
   The MIB Tools opens.
2. Expand the required vendor folder, and select the required custom MIB.
3. On the Navigation toolbar,
   click 
   The " Delete Custom MIBs" dialog opens.
4. Click OK.
   The MIB Tools deletes the MIB from its database.

Editing MIBs

MIB Tools includes an editor that lets you locate and correct errors that are identified during the compilation of a MIB file. To troubleshoot compiler errors, click Show Editor in the 'MIB Tools: Add MIB' dialog. You can then view the MIB file in the editor and make changes to remove errors. You can search the file for alphanumeric strings, locate specific lines in the file, and save the changes that you make.

Import Multiple MIBs

Use the BulkMibCompile command to import large numbers of MIB files into the MIB Tools database. This command lets you migrate existing MIBs without using the MIB Tools interface. BulkMibCompile is located in the default DX NetOps Spectrum installation directory on the OneClick server:

```
<$SPECROOT>/MibDatabase/scripts/BulkMibCompile.sh
```

This command uses the following format:

```
BulkMibCompile [-u <MYSQL USER>] [-p <MYSQL PASS>] -d <MIB DIRECTORY> [-f <FILE MASK>] [-skip_search] [-standard_mibs]
```

- **-u**
  Specifies the MYSQL username. If you do not specify a username, BulkMibCompile uses the default MYSQL username 'root'. This parameter is not required if the default username is correct.
- **-p**

Specifies the MYSQL password. If you do not specify a password, BulkMibCompile uses the default MYSQL password, 'root'. This parameter is not required if the default password is correct.

- **-d**
  Specifies the directory containing the MIBs to import.
  – If you are running a Windows Cygwin bash shell, use the following format for specifying a directory:
  ```
  <$SPECROOT>/MibDatabase/scripts/BulkMibCompile.sh -d c:\\MibDirectory
  ```
  NOTE
  The double backslash (\\) is required. A single backslash (\) is an escape character in this environment.
  – Otherwise, use the following format:
  ```
  <$SPECROOT>/MibDatabase/scripts/BulkMibCompile.sh -d /usr/MibDirectory
  ```
- **-f**
  Specifies the file mask. Use a file mask that includes all of the MIB files that you want to import in a directory. Examples of file masks include:
  ```
  RFC*
  *RFC*
  *.mib
  ```
- **-skip_search**
  Speeds up the import process by instructing the compiler to resolve IMPORTS statements. Use this option if the MIBs that are referenced in IMPORTS statements in the MIB files that you are importing are in the MIB directory that you specified and are named using their MODULE-NAME.
  If you do not use -skip_search, the BulkMibCompile code searches each MIB for IMPORTS statements and attempts to resolve them. This process is repeated during the compilation of each MIB.
- **-init**
  Reinitializes the MIB Tools database, clearing the database of all MIBs.
- **-standard_mibs**
  Specifies that all the MIBs being imported into the <MIB DIRECTORY> will be added as standard MIBs.

  NOTE
  If you run the "BulkMibCompile" command without this option at the CLI, all the imported MIBs will be added as custom MIBs. Custom MIBs can be deleted.

Example: Import All MIBs in a Directory

To import all MIBs in a directory, use the following syntax:

```
<$SPECROOT>/MibDatabase/scripts/BulkMibCompile.sh -d /usr/MibsToCompile
```

A MIB that has been successfully compiled is added to the database. The imported MIB overwrites any existing MIBs with the same MIB MODULE-NAME. Any compilation errors are displayed, and compilation continues with the next MIB that is imported.

Example: Reinitialize a MIB Tools Database

To reinitialize a MIB Tools database, use the following command:

```
<$SPECROOT>/MibDatabase/scripts/BulkMibCompile.sh [-u <MYSQL USER> ] [-p <MYSQL PASS> ] -init
```

Example: Populate the MIB Tools Database of Another OneClick Server from the Primary OneClick Server

To populate the MIB Tools database of another OneClick server from the primary OneClick server, take the following steps:

1. Copy the contents of *<$SPECROOT>*/MibDatabase and *<$SPECROOT>*/MibDatabase/Dependent on the primary OneClick server to the same directories on the secondary OneClick server.
2. On the secondary OneClick server, run BulkMibCompile to import the MIBs that you copied into the *<$SPECROOT>*/MibDatabase:

```
<$SPECROOT>/MibDatabase/scripts/BulkMibCompile.sh -d <$SPECROOT>/MibDatabase
-skip_search
```

After the script has completed, the databases on the original and destination OneClick servers are identical.

Create Attribute Support

You can add support for MIB objects that DX NetOps Spectrum does not currently support. Create attribute support for a MIB that you have imported into the MIB Tools database.

To create support for MIB objects in the DX NetOps Spectrum database, create an attribute identifier (ID) for the MIB objects. Attribute IDs are used to create mappings between the DX NetOps Spectrum database and MIB objects. You can use the attribute information to develop custom views, Watches, or leverage other features to manage network devices.

Follow these steps:

1. In the Navigation panel of MIB Tools, select the MIB for which you want to create attribute support.
   The list of MIB objects appears in the Attribute Support table.
2. Check the columns for MIB objects that lack corresponding Attribute IDs.
3. Select a MIB object, and click the Create Attributes icon 
4. Click OK to continue creating the attributes on all SpectroSERVERs.
   The 'MIB Tools: Attribute Creation Results' dialog shows the status, including the number of attributes created for each landscape.
5. Click Close.
   The Attribute Support table now shows the Attribute ID that was created for each object. The new attributes are available for use by all SNMP-capable models in DX NetOps Spectrum.

Modifying MIB Objects in the MIB Tools Database

After you have imported a MIB object into the MIB Tools database and have mapped the object by configuring attribute support, the object definition is locked. The only part of the object definition that you can change is the enumerations.

WARNING
Do not edit a MIB object that has been mapped. Modifications to any parameters other than enumerations (such as name or data type) are not supported. In addition, you cannot use MIB Tools to delete attributes that have already been mapped.

WARNING
The only way to modify a MIB object that has been mapped is to re-import the MIB into the MIB Tools database. For more information, see Import Individual MIBs.

Query (GET_NEXT), GET, and SET Requests

You can run queries (GET_NEXT) and perform GET and SET operations for MIB objects to retrieve and set information about network devices.

You must know the Object IDs of the objects to query. You can query a device that is not modeled in DX NetOps Spectrum.

Query a Subtree of Objects

You can query a MIB object in the MIB database on a network device. You can perform a query to repeatedly perform a request to find multiple instances of a MIB object as long as the objects returned have an OID that is prefixed by the OID of the branch you selected to do the query on.

Follow these steps:

1. Establish contact with the device to query.
2. Navigate to the MIB object you want to query.

   NOTE
   Select an inaccessible object (folder) or a table leaf object to perform the SNMP GET_NEXT query. Performing an SNMP GET_NEXT query on a scalar leaf object yields an empty result set.
3. In the Hierarchy tree table toolbar, click the Query/GET_NEXT icon  .

   NOTE
   The Query/GET_NEXT button is available when a readable object is selected in the Hierarchy tree table.

   The query results appear in the Results table.

   NOTE
   If you select a group object (contains other groups and individual objects) or a table object (contains multiple instances of the same object) in the Hierarchy tree table and click the GET icon, the SNMP GET dialog opens. Change the request type to a GET_NEXT in the Request Type drop-down menu before proceeding. Performing a GET request on a group object fails.

Query an Object

You can query a MIB object in the MIB database. You can perform a query to search for a single instance of an accessible MIB object. You can also perform a query to repeatedly request multiple instances of a MIB object. But the objects that are returned must have an OID that is prefixed by the OID of the branch where you run the query.

Follow these steps:

1. Establish contact with the device to query.
2. Navigate to the group object or leaf that you want to query.
3. In the Hierarchy tree table toolbar, click the GET icon  .

   NOTE
   The GET button is available after you select a readable object in the Hierarchy tree table.

   One of the following things occurs:
   - If a scalar leaf object is selected, the GET request is performed.
   - If a table leaf object is selected, the SNMP GET dialog opens.
4. To retrieve a particular instance of the table object, specify an instance identifier and click OK.

   NOTE
   If the device contains multiple instances of the MIB object, enter the value of the instance to query in the Instance field.

   If the query is successful, the result appears in the Results table.
   If the query fails, an error message appears, indicating the reason for the failure.

If no results appear in the Results table, the object is not supported by the device, the object is not accessible, or the device is unreachable.

Set an Object

You can set the value of a MIB object on a network device. The MIB objects that you modify must have write access, such as read/write. View the access value for MIB objects in the Hierarchy tree table Access column.

Follow these steps:

1. Establish contact with the device on which you want to set the value of the MIB object.
2. Navigate to the MIB object that you want to modify.
3. In the Hierarchy tree table toolbar, click the SNMP SET icon 
   The 'MIB Tools: SNMP Set' dialog opens.
4. Take the following steps:
   - Verify that the correct MIB object is selected in the dialog. Or for a MIB object that is not in the MIB Tools database, enter the OID for the object that you want to modify.
   - Enter the Instance of the object that you want to modify.
   - Enter the new Value for the object instance.

   NOTE
   Depending on the object type, you can often select a value from a list in the 'MIB Tools: SNMP Set' dialog.
5. Click OK.
   A dialog opens, indicating whether the SET action was successful. If the action was unsuccessful, a reason is provided.

If the SET action fails, use the following checklist for troubleshooting:

1. Verify that the device is reachable. Try to contact the device from the Contact Criteria section of the Browse tab.
2. If you can contact the device, verify that you can perform a GET action on the device for the MIB object.
3. Verify that the MIB object has read/write access by checking the value for the Access parameter in the Hierarchy tree table.

   NOTE
   If the MIB is not part of the MIB Tools database, use the MIB itself to verify that the MIB object has read/write access.
4. Finally, verify that the SNMP community string in the Contact Criteria is the correct one for writing to the device. You can verify the community string value in the DX NetOps Spectrum Modeling Information subview in the OneClick Component Detail panel.

Device Query and SET Results

The Results section of the Browse tab displays the results of SNMP GET_NEXT, GET, and SET requests in a table.

The available columns include the following:

- **Name**
  Displays the name of the MIB object queried.
- **Instance**
  Displays the instance of the object queried.
- **Type**
  Displays the object type, such as Integer, Counter, IP Address, Octet String, Gauge, Time Ticks, and so on.
- **Value**

Displays the value of the MIB object read from the device.

- **Object ID**
  Displays the Object ID of the object.
- **Access**
  (Hidden by default) Displays the access type for the object. The access type can be read-only, read/write, read-create, or not-accessible.

NOTE
For more information about setting table preferences, see the Using OneClick section.

The Results toolbar provides the following functionality:

Button	Description
	
	**SNMP GET button:** Retrieves the value of the selected MIB object on a specific device. Take this step after a SET action because the values in the Results table are not automatically updated.
	**SNMP SET button:** Sets the value of the selected MIB object on a specific device. Available only for objects that have read/write access. Take this step after a SET action because the values in the Results table are not automatically updated.
	**Clear button:** Clears the contents of the Results table.
	
Auto Clear	**Auto Clear check box:** Clears the contents of the Results table each time you query a device. To view sequential SNMP queries, clear this check box.
Filter	**Filter text box:** Filters the Results table. Only results that contain the text string that you supplied as a filter appear in the table.

Export Query Results To Support Troubleshooting

To aid in troubleshooting a problem, CA Technical Support sometimes asks you to provide a .sim file. Exported from MIB Tools, this file contains information to build a simulation of your device. You can use a query to obtain a detailed SNMP snapshot of your device to provide to CA Technical Support.

Follow these steps:

1. Perform a query on the device for which you require support (typically from the ‘internet’ branch).
2. In the Results table toolbar, click the Export
   icon 
   The ‘Export table data to file’ dialog opens.
3. Select Simulation (*.sim) from the 'Save as type' list.

   NOTE
   We recommend using the name of the device in the filename for easier recognition.
4. Save the file to your local file system.

The query results are exported.

Custom Vendor Folders

Create Custom Vendor Folders

You can create custom vendor folders to organize your compiled MIBs. You can edit these folders to change their names and can also delete these folders.

Note: A star icon indicates custom vendor folders. You can only modify or delete custom vendor folders, not the predefined vendor folders.

Follow these steps:

1. In MIB Tools, click

   (Creates a new folder) in the Navigation panel.
   The 'MIB Tools: Create Vendor' dialog opens.
2. Enter a name for the vendor folder you want to create and click OK.
   The new folder appears in alphabetical order in the Navigation panel.

Edit Custom Vendor Folders

You can change the name of custom vendor folders.

Note: A star icon indicates custom vendor folders. You can only modify or delete custom vendor folders, not the predefined vendor folders.

To edit a custom vendor folder

1. In MIB Tools, select a custom vendor folder in the Navigation panel and click

   (Edit Folder).

   NOTE
   The 'MIB Tools: Edit Vendor' dialog opens.
2. Edit the name of the folder and click OK.

   NOTE
   The name of the folder is changed.

Delete Custom Vendor Folders

You can delete the custom vendor folders that you created.

Note: A star icon indicates custom vendor folders. You can only modify or delete custom vendor folders, not the predefined vendor folders.

Follow these steps:

1. In the MIB Tools utility, select a custom vendor folder in the Navigation panel and click
   
   (Delete Folder).
   A confirmation dialog opens.
2. Click Yes to confirm the deletion.
   The custom vendor folder is deleted.

   NOTE
   Any MIBs that were in the deleted folder are automatically moved back to their original predefined vendor folders.

Move MIBs to Custom Vendor Folders

You can move a MIB from a predefined vendor folder to a custom vendor folder. You can also move a MIB from a custom vendor folder back to its default folder.

Follow these steps:

1. In MIB Tools, expand the folder where the MIB you want to move exists, select the MIB, and then click
   
   (Move MIB).
   The 'MIB Tools: Move Selected MIB' dialog opens.
2. Select the custom vendor folder where you want to move the MIB, and click OK.
   The MIB appears in the custom vendor folder in the Navigation panel.
3. (Optional) Move the MIB back to its predefined folder.
4. Expand the custom folder, select the MIB, and click Move MIB.
   The 'MIB Tools: Move Selected MIB' dialog opens.
5. Select the predefined vendor folder where the MIB was originally stored and click OK.
   If you select multiple MIBs to move, an [ORIGINAL VENDOR FOLDER] option appears. Click this option to move multiple MIBs to their respective default vendor folders.
   The MIB reappears in the predefined vendor folder in the Navigation panel.

Contact a Device Using MIB Tools

You can contact a device using MIB Tools. The utility lets you specify parameters for the SNMP query that is sent. SNMP security information information is required.

Follow these steps:

1. Open MIB Tools and click the Browse tab.
2. Complete the following fields in the Contact Criteria section:
   - **Device**
     Specifies the IPv4 or IPv6 address or the hostname for the device.

     NOTE
     Check the Device list to see the last ten devices that were successfully contacted.
   - **Version**
     Specifies the version of SNMP you want to use.
   - **SNMP v3 Profile**

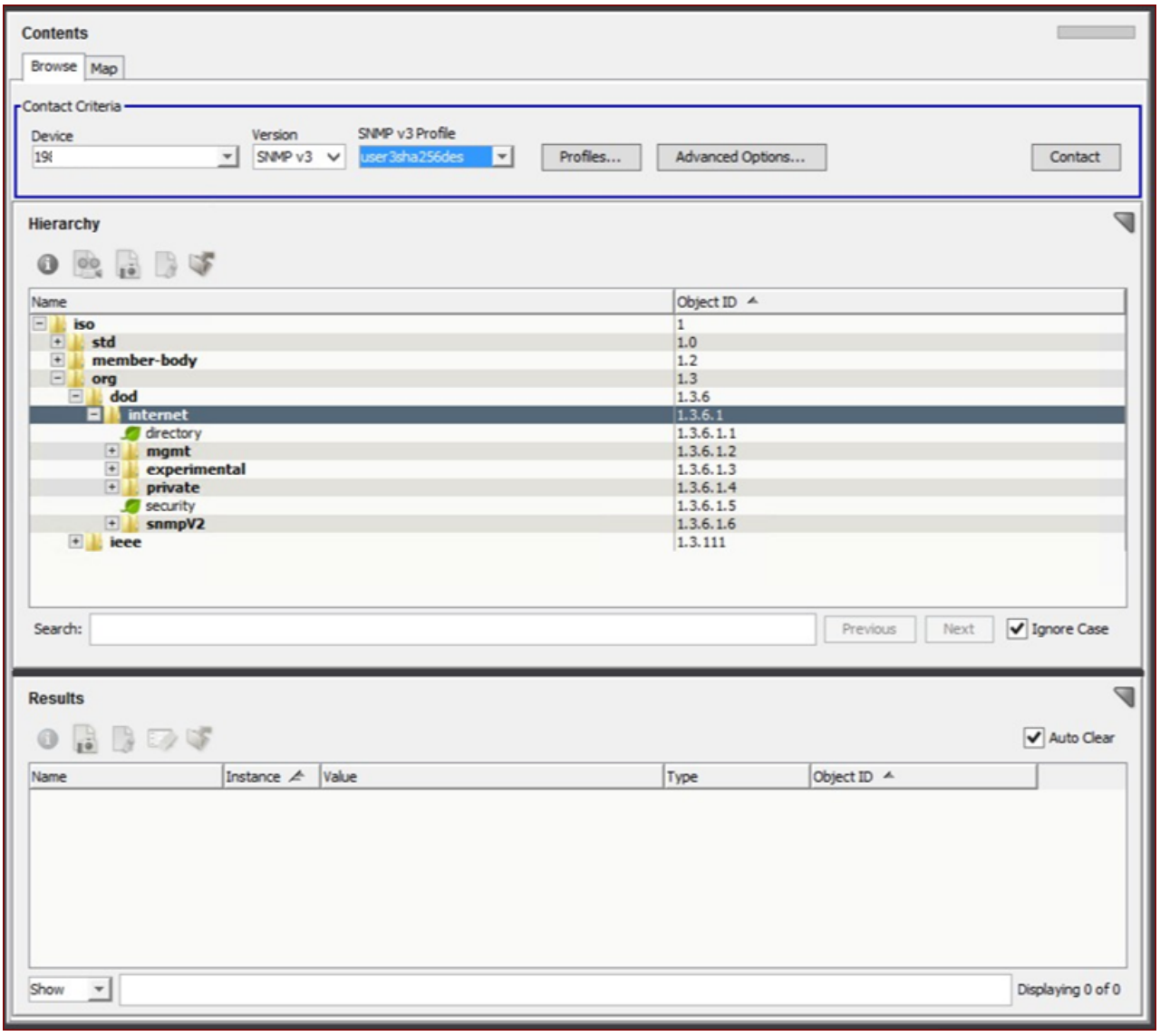
If you are using SNMPv3, select the profile required to contact the device from the SNMP v3 Profile drop-down. If you want to create a profile, click the Profiles button.

NOTE
In 10.4.2, the SNMPv3 profile creation supports SHA-256 and SHA-512 as authentication protocols. Therefore, you can now select those profiles, too.

- **SNMP Community String**
  Specifies the SNMP community string used to access the device.

3. (Optional) If you are using Secure Domain Connector, or if you want to change any of the default values for SNMP to use in contacting the device, click Advanced Options and complete the following fields:
   - **Landscape**
     Specifies the landscape from which the SNMP request should be initiated, if you have a Distributed SpectroSERVER (DSS) environment.
   - **Secure Domain**
     Specifies the secure domain to which the SNMP request should be forwarded, if you have the Secure Domain Manager add-on application installed with secure domains configured.
   - **Port**
     Specifies the value of the UDP port to contact the device on.
     Default: 161
   - **Retry Count**
     Specifies the number of times to retry contacting the device before MIB Tools stops attempting to contact the device.
     Default: 3
   - **Timeout (ms)**
     Specifies the amount of time, in milliseconds, to wait before trying to contact the device again.
     Default: 3000
4. Click Contact to initiate contact with the device.
   The color of the line surrounding the Contact Criteria section indicates the status of contact with the device. The following defines the contact status indicator colors.
   - **Blue:** Contact with the device has not been initiated.
   - **Yellow:** Contact with the device is in progress.
   - **Green:** Contact with the device was successful.
   - **Red:** Contact with the device was unsuccessful.

The following screenshot shows the required information:



Search for a MIB

You can search for MIBs in the Navigation panel of the MIB Tools utility.

Follow these steps:

1. Enter a text string in the Search field in the Navigation panel.

   NOTE
   Wildcards are not supported.
2. (Optional) Select the Ignore Case check box to make the search case-insensitive.
3. Press Enter to search.
   The Next and Previous buttons let you scroll through each successive instance of the search string.
   When a match is found, it is highlighted in the Navigation panel. The Contents panel is populated with the information about the selected MIB.

Trap Support for MIB Objects

The MIB Tools utility lets you create custom traps and modify traps. You can create trap support for MIB objects that are not currently supported by DX NetOps Spectrum. The first step is importing the MIBs.

You can also create support for traps that are defined in the MIBs that you have imported. And you can customize traps for the MIBs that you have imported into the MIB Tools database. Finally, you can add custom trap support for new devices that are not yet supported in DX NetOps Spectrum.

Custom Trap Support File Details

When you map traps using MIB Tools, entries are generated in the following files on all SpectroSERVERs in your DSS environment:

```
<$SPECROOT>/custom/Events/EventDisp
<$SPECROOT>/custom/Events/AlertMap
```

The mapping information for a trap in these files overrides any previous mapping information for that trap in other SpectroSERVER files or directories. And when you map traps, files are generated in the following directories on the OneClick server to which you are connected:

- *<$SPECROOT>*/custom/Events/CsEvFormat -- An Event Format file applies to each DX NetOps Spectrum event. These files define the event text that appears in the OneClick Alarm and Event lists.
- *<$SPECROOT>*/custom/Events/CsPCause -- A Probable Cause file defines the text for each DX NetOps Spectrum alarm that appears in the OneClick Alarm Details view. Select an Alarm Severity level when you map a trap to enable Probable Cause.
- *<$SPECROOT>*/custom/Events/CsEvFormat/EventTable -- An Event Table file determines each varbind sent with a trap that includes enumerated definitions in the MIB.

Manually copy these directories from your primary OneClick server to other OneClick servers in a multiple OneClick server environment.

If you use the CLI commands 'show alarms' or 'show events', or if you deploy DX NetOps Spectrum Alarm Notification Manager (SANM), copy the contents of these directories to all SpectroSERVERs in *<$SPECROOT>*/SG-Support.

> **NOTE**
> The Getting Started section and the Event Configuration section provide information about DX NetOps Spectrum alarm and event support files.

Create Trap Support

You can add support for MIB objects that DX NetOps Spectrum does not currently support by creating trap support. The MIBs must have already been imported into the MIB Tools database.

You can create support in DX NetOps Spectrum for traps that are defined in MIBs and are imported into the MIB Tools database. You can add custom trap support for new devices that are not supported in DX NetOps Spectrum. Or you can change the way that a trap is supported.

In a fault-tolerant SpectroSERVER environment, MIB Tools does not map traps on a secondary SpectroSERVER when the primary SpectroSERVER is down. If you have multiple primary SpectroSERVERs, then at least one of your primary SpectroSERVERs can be available to create trap support.

Follow these steps:

1. Select Tools, Utilities, and MIB Tools from the OneClick main menu.
2. Select MIB in the Navigation panel.

A list of traps for each MIB appears in the Trap Support table. TheMap Traps icon is only enabled if some traps lack DX NetOps Spectrum support.

NOTE
Use multiselect to select specific traps. Support is created only for the traps you select.

WARNING
Any traps with partial support appear in the Trap Support table with Partial in the Event Code column. To create global trap support for these traps, remap these traps before continuing. For more information, see the Modeling and Managing Your IT Infrastructure section.

3. Click the Map Traps icon  .
   The MIB Tools: Assign Trap Alarms dialog opens.
4. Click set in the Alarm Severity column to select the severity for the alarm that DX NetOps Spectrum generates when it receives the trap.
5. Click OK to map the traps on all SpectroSERVERs.
   The MIB Tools: Trap Support Results dialog displays the status of the Map Traps action. When the action completes, the results list the number of traps that were created for each landscape.
   You can use Show Advanced Options when mapping traps to DX NetOps Spectrum events. For more information, see Show Advanced Options for Mapping Traps.
6. Click Close.
   The Trap Support table shows the event code for each trap. The new events are now available for use by all model types in the DX NetOps Spectrum modeling catalog.
   DX NetOps Spectrum processes the trap accordingly.

Review Custom Trap Mapping Information

You can use MIB Tools to determine whether a trap has a default mapping, a custom mapping, or both.

Follow these steps:

1. In MIB Tools, select the MIB in the Navigation panel whose trap support you want to verify. Click the Map tab.
   The list of traps for the MIB appears in the Trap Support table.
2. Right-click the Trap Support table column header.
   The Table Preferences dialog opens.
3. In the Columns tab, select the Event Type check box, and click OK.
   The Event Type column is added to the Trap Support table.
   Default mappings have 'CA' in the Event Type column.
   Custom mappings have 'Custom' in the Event Type column. If a custom trap mapping is obscuring, or shadowing, a default trap mapping, the Event Type column displays a ‘Custom’ hyperlink.
4. Click the ‘Custom’ link.
   The MIB Tools: Custom Trap Mapping Details dialog opens.
5. Review the Custom Disposition Details and the CA Disposition Details for the trap to determine whether to Remove Custom Trap Mappings.
6. Click Close.
   The dialog closes.

Remove Custom Trap Mappings

You can remove custom mappings from traps in MIB Tools.

Follow these steps:

1. In MIB Tools, select the MIB in the Navigation panel whose custom trap support you want to remove. Click the Map tab.
   The list of traps for the MIB appears in the Trap Support table.
   The Remove Traps button is available if at least one item with a custom mapping is selected.
2. Click the Remove Traps
   icon 
   The MIB Tools: Delete Trap Mappings dialog opens.
3. Click OK to confirm the removal of this custom mapping.
   The MIB Tools: Trap Support Results dialog shows the status of the Remove Traps action.
   The results show the number of traps that were removed for each landscape.
4. Click Close.
   If no default mapping exists, the Trap Support table shows a blank entry in the Event Code column. However, if the custom mapping was shadowing a default mapping, the event code of the default mapping is displayed.

Remove Partial Mappings From Traps

You can identify when a trap is only supported on select model types in the DX NetOps Spectrum modeling catalog. You can also change trap support to include all model types.

When a trap is globally supported in DX NetOps Spectrum, it is available for use by all model types. The event code value for a globally supported trap appears in the Event Code column in the Trap Support table.

When a trap is partially supported in DX NetOps Spectrum, it is available for use by certain model types only. The event code value for a partially supported trap appears in the Event Code column in the Trap Support table as ‘Partial.’

We recommend not modifying partially supported traps. However, you can remap a trap to make it globally supported in DX NetOps Spectrum.

> **WARNING**
> If you remap a trap, you can disable useful DX NetOps Spectrum functionality. Proceed with caution.

To change partial trap support to global trap support, remove the partial mapping and then remap the trap.

Follow these steps:

1. In MIB Tools, click the Map tab.
2. Click ‘Partial’ in the Event Code column of the Trap Support table for the trap you want to remap.
   The ‘MIB Tools: Partial Trap Support’ dialog opens, displaying the trap event code, the landscapes it is available on, the MIB object name, and the object ID.
3. Click Remap.
   The ‘MIB Tools: Confirm Remap’ dialog opens.
4. Click Yes to continue remapping the trap.
   The remapped trap appears in the Trap Support table with the previous partial mapping removed; there is no event code value and ‘Partial’ no longer appears in the Event Code column.
5. (Optional)Remap these traps globally.
   The partial mapping is removed.

To remove a partial mapping from all traps in a MIB, select the Remap All Conflicts check box.

Follow these steps:

1. In MIB Tools, select the MIB that contains the traps whose partial support you want to remove in the Navigation panel. Click the Map tab.
   The traps with partial support appear in the Trap Support table.

2. Select the Remap All Conflicts check box.
   All partial trap support is removed for the traps.
3. Verify that the traps lack an event code and that 'Partial' no longer appears in the Event Code column.
   The previous partial mapping is now removed.
4. (Optional) Remap these traps globally.
   The partial mapping is removed.

Show Advanced Options for Mapping Traps

You can use Show Advanced Options when mapping traps to DX NetOps Spectrum events. This option lets you select the event codes to assign to new trap mappings. You can also export the events and alarms to a specific directory.

Follow these steps:

1. Select Tools, Utilities, and MIB Tools from the OneClick main menu.
2. Select MIB in the Navigation panel.
   A list of traps for each MIB appears in the Trap Support table. TheMap Traps icon is only enabled if some traps lack DX NetOps Spectrum support.

   NOTE
   Use multiselect to select specific traps. Support is created only for the traps you select.

   WARNING
   Any traps with partial support appear in the Trap Support table with Partial in the Event Code column. To create global trap support for these traps, remap these traps before continuing. For more information, see the Modeling Your IT Infrastructure section.
3. Click the Map Traps icon 
   The MIB Tools: Assign Trap Alarms dialog opens.
4. Click Show Advanced Options.
   The following advanced options are available in MIB Tools:
   - **Starting Event Code**
     Lets you specify the first code to use for events.
     By default, DX NetOps Spectrum automatically calculates and assigns event codes for new trap mappings. A read-only event code appears in the Starting Event Code field and in the selected Use Default check box.
     However, you can specify a starting event code to assign to new trap mappings. Clear the Use Default check box to enable the Starting Event Code text box. DX NetOps Spectrum then automatically calculates and assigns the event codes that are based on the new starting event code.

     NOTE
     Select a unique starting event code.
   - **Install Trap Support**
     Installs the event and alarm support files in the DX NetOps Spectrum installation area, providing immediate support for the new trap mappings.
   - **Export Trap Support**
     Lets you specify the directory where the event and alarm support files are exported. Selecting this option does *not* provide support for the new trap mappings in DX NetOps Spectrum.

MIB Tools Support for Multiple SpectroSERVERs

The MIB Tools utility offers the following features to supports a Distributed SpectroSERVER (DSS) environment:

Offline SpectroSERVERs: MIB Tools can recognize that a SpectroSERVER is offline while attempting to create Event Codes and Attribute IDs for MIB objects. You receive a notification.

- **Incomplete Trap and Attribute Support:** MIB Tools can identify situations where MIB objects are supported on some, but not all, SpectroSERVERs.
- **Resolve Incomplete Support:** MIB Tools can resolve incomplete trap support.
- **Conflicting Support:** MIB Tools can identify MIB objects that have conflicting Event Code or Attribute ID mappings on multiple SpectroSERVERs.
- **Resolve Trap Conflicts:** MIB Tools can resolve trap disposition conflicts.

MIB Tools Synchronization in a DSS Environment

Use the following guidelines to maintain synchronization among SpectroSERVERs in a DSS setup:

- Do not create attribute or trap support if any of the SpectroSERVERs are down. A warning dialog appears in this situation.
- Always create consistent support for attributes and traps on all SpectroSERVERs.
- Always resolve trap disposition conflicts when they appear.

Attribute Conflicts

When inconsistent support for an attribute occurs in a DSS environment, an attribute conflict condition exists. Usually, this results when an attribute is mapped to different attribute IDs on one or more SpectroSERVERs.

The Attribute ID value for the attribute in the Attribute Support table indicates that a conflict has been detected. To see the landscape and the attribute ID for the attribute, click the 'Conflict' link in the table. At least one SpectroSERVER has a different attribute ID, or no attribute ID, for the attribute in conflict situations.

To resolve attribute conflicts, synchronize the modeling catalogs on your SpectroSERVERs.

NOTE
For more information, see the Distributed SpectroSERVER Administration section.

Create Consistent Support Across a DSS Environment

When a trap or attribute is supported on some, but not all, SpectroSERVERs in a DSS environment, the value 'Some' in the Landscapes column. Click the 'Some' link to see the mapping of the attribute ID or event code to a landscape for a MIB object. Landscapes where the selected MIB object is not mapped lack a value.

Create consistent support across your DSS environment for traps and attributes that are supported on some of your SpectroSERVERs. Remap the traps and recreate the attributes once all SpectroSERVERs are running.

You can create consistent attribute support and consistent trap support across a DSS environment.

Follow these steps:

1. Select the MIB containing the attributes for which you want to create consistent support in the Navigation panel of MIB Tools.
2. Click the Map tab, and verify that attributes appear in the Attribute Support table with the value 'Some' in the Landscapes column.
3. Click Create Attributes 
4. Click OK to confirm.

   The MIB Tools: Attribute Creation Results dialog displays the status and results of the Create Attributes action.
5. Click OK.

The value ‘All’ appears in the Landscapes column for all the attributes.

Follow these steps:

1. In MIB Tools, select the MIB containing the traps for which you want to create consistent support in the Navigation panel.
2. Click the Map tab and verify that traps appear in the Attribute Support table with the value 'Some' in the Landscapes column.
3. Click Map Traps 
   The MIB Tools: Assign Trap Alarms dialog opens.
4. (Optional) Click set in the Alarm Severity column to select the alarm severity for the alarm that DX NetOps Spectrum generates when it receives the trap. Or select None if the trap does not generate an alarm.
5. Click OK to map the traps on all SpectroSERVERs.
   The MIB Tools: Trap Support Results dialog shows the status of the Map Traps action. The results list the number of traps that were created for each landscape.
6. Click Close.
   The Trap Support table shows the event code for each trap.

Synchronize and Update MIB Databases and Support Files on Multiple OneClick Servers

When you import MIBs into the MIB Tools database and create trap and attribute support, information is written to each SpectroSERVER in your DSS environment. However, information required by OneClick to support the traps and attributes is written only on the OneClick server to which you are connected.

If you have multiple OneClick servers in your environment, maintain synchronicity among the MIB Tools databases and support files for new attribute and trap support.

You can synchronize and update MIB databases and support files on multiple OneClick servers.

Follow these steps:

1. Designate one of your OneClick servers as the primary server that contains the primary MIB Tools database.
2. Import all MIBs and create attribute and trap support on this primary OneClick server.
3. Distribute the MIB Tools database to other OneClick servers from the primary server.
4. Distribute OneClick support files created for events and alarms from the primary OneClick server to the other OneClick servers.

Trap Disposition Conflicts

When inconsistent support for a trap occurs in a DSS environment, a trap disposition conflict condition exists. Inconsistent trap support includes the following situations:

- A trap is mapped to different event codes on one or more SpectroSERVERs.
- A trap is disposed differently on one or more SpectroSERVERs.
  - For example, a trap is disposed to a minor alarm on one SpectroSERVER, and a major alarm on another.
  - A trap is using an event rule for complex processing, but another instance of the trap is not using that rule.

When a trap disposition conflict exists, the Event Code value for that trap in the Trap Support table reads ‘Conflict.

Resolve Trap Disposition Conflicts Remap the Trap

You can resolve a trap disposition conflict by remapping the trap on all SpectroSERVERs in your DSS environment. Resolving trap disposition conflicts creates consistent support for the trap on all SpectroSERVERs. Each SpectroSERVER in your DSS environment must be running to resolve a trap disposition conflict.

Follow these steps:

1. In MIB Tools, click the Map tab and locate the trap in conflict in the Trap Support table.
2. Take one of the following steps:
   - For multiple trap disposition conflicts, select Remap All Conflicts.
   - For a single trap, click Conflict in the Event Code column.
     The Trap Disposition Conflict dialog lists each landscape and the event code for the trap.
     Click Remap, and then click Yes to confirm.
     The trap now has a value of 'Some' or 'None' in the Landscape column of the Trap Support table.
3. Click
   
   (Map Traps) to create an event code for the trap on all SpectroSERVERs in the DSS environment.
   The traps are remapped.

Resolve Trap Disposition Conflicts Edit the AlertMap and EventDisp Files

You can resolve a trap disposition conflict by editing the AlertMap and EventDisp files for the trap. Such conflicts occur in a DSS environment when, for example, a trap is mapped to different event codes on one or more SpectroSERVERs.

Follow these steps:

1. In MIB Tools, click the Map tab and locate the trap in conflict in the Trap Support table.
2. Click Conflict in the Event Code column.
   The Trap Disposition Conflict dialog lists each landscape and the event code for the trap.
3. Locate the desired mapping by reviewing the Event details.
4. Synchronize the conflicting events on the other SpectroSERVERs in your DSS environment by editing the appropriate EventDisp file and AlertMap file with a text editor.

   NOTE
   The AlertMap files and EventDisp files are located in the <*$SPECROOT*>/SS/CsVendor directory.
5. Issue a command on each SpectroSERVER to reload its events and alerts.
   For more information, see the Event Configuration section.

Troubleshooting for Trap Mappings missing in MIB Tools

Symptom:

Trap mappings missing after compiling NGD_G42.MIB file in Spectrum MIB Tools.Trap mappings missing after compiling NGD_G42.MIB file in Spectrum MIB Tools. Specifically the trap mappings for the dpsRTUp8177nSet, dpsRTUp8180nSet and dpsRTUp8183nSet traps. The NGD_G42.MIB file has the following NOTIFICATION-TYPE configuration for these traps:

```
dpsRTUp8177nSet NOTIFICATION-TYPE

                OBJECTS { sysDescr, sysLocation, dpsRTUDateTime,
```

```
                                                                dpsRTUAPort,
 dpsRTUCAddress, dpsRTUADisplay, dpsRTUAPoint,
                                                                dpsRTUAPntDesc,
 dpsRTUAState }
                STATUS current
                DESCRIPTION "Granular point 8177 is set."
                ::= { dpsRTU 8177 }


dpsRTUp8180nSet NOTIFICATION-TYPE
                OBJECTS { sysDescr, sysLocation, dpsRTUDateTime,
                                                                dpsRTUAPort,
 dpsRTUCAddress, dpsRTUADisplay, dpsRTUAPoint,
                                                                dpsRTUAPntDesc,
 dpsRTUAState }
                STATUS current
                DESCRIPTION "Granular point 8180 is set."
                ::= { dpsRTU 8180 }


dpsRTUp8183nSet NOTIFICATION-TYPE
                OBJECTS { sysDescr, sysLocation, dpsRTUDateTime,
                                                                dpsRTUAPort,
 dpsRTUCAddress, dpsRTUADisplay, dpsRTUAPoint,
                                                                dpsRTUAPntDesc,
 dpsRTUAState }
                STATUS current
                DESCRIPTION "Granular point 8183 is set."
```

```
                ::= { dpsRTU 8183 }
```

Cause:

The NGD_G42.MIB file contains a NOTIFICATION-TYPE configuration for these traps but does not contain a TRAP-TYPE configuration for these traps.

Resolution:

Follow the steps:

1. Edit the NGD_G42.MIB file
2. Add the TRAP-TYPE configurations for the dpsRTUp8177nSet, dpsRTUp8180nSet and dpsRTUp8183nSet traps. The following is an example for the dpsRTUp8177nSet trap:

```
dpsRTUp8177Set TRAP-TYPE

        ENTERPRISE  dpsRTU

        VARIABLES {     sysDescr, sysLocation, dpsRTUDateTime,

                                    dpsRTUAPort, dpsRTUCAddress, dpsRTUADisplay,
 dpsRTUAPoint,

                                    dpsRTUAPntDesc, dpsRTUAState }

        DESCRIPTION "Generated when discrete point 177 is set."

        ::=  8177
```

Fetch MIBs Information of a Device Using a Script

10.3 introduces a new script (Monitoring_Profile.pl) to fetch MIBs information of a selected device. A user can use this script to fetch the following information of a device:

- List of MIBs
- Trap details
- Pollable Attributes
- Events and Alarms information of a device

Running the Monitoring Script

Following is the procedure to run the Monitoring Profile Script:

Prerequisites:

Following are the prerequisites to run the Monitoring_Profile.pl script:

1. Ensure you log in the SpectroSERVER as one of the DX NetOps Spectrum users. A user should be present in the DX NetOps Spectrum users list to run LogPollAnalyzer.exe , which fetches the attributes list. If the user should be present in the DX NetOps Spectrum users list for the LogPollAnalyzer.exe to work.
2. Ensure that the Tomcat Server and the SpectroSERVER are up and running.

To run the script, follow these steps:

1. Navigate to $SPECROOT/Install-Tools/PostInstall/ directory
2. Open a bash shell and run the following script for a model handle of a device.

   ```
   ./Monitoring_Profile.pl <device model handle> <OneClick host name> <port>
   ```

   Script prompts for the user name and password of OneClick.
   The script contacts vnmsh directory to fetch the MIBs information.
3. If you have the SSL option, select 'Yes' to enable the https support.

   NOTE

   Now fetch MIB information using scripts with HTTPs support with the new Spectrum 10.3.1 release.
4. The script generates following files in the $SPECROOT\SS-Tools folder.
   a. trap_details_<model handle>.csv - lists MIBs and corresponding Traps, Trap OID, Event Code, Alarm Severity, Alarm Name, and Alarm PCause Code details.
   b. pollable_attributes_<model handle>.csv - lists the attributes which are being polled the selected device.

      NOTE

      The attributes to be polled are fetched from the existing feature LogPollAnalyzer.exe.

Enhancement Support for Monitoring Profile in Spectrum 10.3.1

10.3.1 introduces enhancements to support the Monitoring Profile features present in the previous release. These enhancements include:

1. MIB Description, Dynamic Alarm Titles, Event Text, and Event Procedures: A REST API fetches the trap information of a MIB such as the MIB name and the landscape handle is passed as parameters such as the Alarm Titles, Event Text and Event Procedures.

2. Custom AlertMaps: An action code supports the custom AlertMaps that dumps all the custom OIDs, event code and the pcause code.
3. Support for all MIBs

Known Issues

- Conditional alarms are not listed in the output files.
  For example: The script supports only the EventDisp strings in the format like 0x01165306 E 50 A 2,0x01165305
  The script does not support EventDisp strings like 0x0116912a E 20 A { v 4 Ctron_Gen_HOST.alertCurrentStatus },0x0116912a

Developing a New Certification

This section describes various concepts and operations to develop a new or customized certification for your device for managing it properly.

New Certification Management

To model a device, DX NetOps Spectrum uses a device model type and its associated interface and application model types. You can add device model types or you can enhance the functionality of the GnSNMPDev device model type. To enhance device management with DX NetOps Spectrum, a solid understanding of the functional components of a device is required.

The GnSNMPDev device model type and the interface and application models that are known to the GnSNMPDev model type support many device functions. Supported functions comprise both proprietary and standard MIBs. Identify the functionality of the device that GnSNMPDev already supports. For example, if device interfaces map one-to-one with physical ports on a single board, GnSNMPDev supports this device without enhancement. GnSNMPDev includes native support for MIB-II interfaces in the Snmp2_Agent application model.

To test GnSNMPDev device support, use an IP address to model the device in OneClick. DX NetOps Spectrum automatically finds the model type most appropriate for the device. If a specific model type is lacking, DX NetOps Spectrum selects the GnSNMPDev model type and instantiates a GnSNMPDev model to represent the device. You can then evaluate the type of support that DX NetOps Spectrum can provide for your device by default.

Once you have established the default support, consider the required customizations. You can then more easily decide if further customization is necessary to manage your device properly. The following sections outline some scenarios in which expanded support is required.

Additional MIB Support

If device management in your environment requires access to additional MIBs, MIB objects can be made available to a device model. The following methods let you increase access to MIBs:

- (Recommended) Import the new MIB directly into the SpectroSERVER using the import mechanism provided in MIB Tools.
- Create a device model type to represent your device and include the necessary MIBs in the device model type.
- Create an application model type that provides access to the new MIB.

Unique Trap Mapping

Create a device model type if the device that you are modeling requires unique trap processing in response to a common trap. For example, assume that you want core routers to generate a major alarm in response to an authentication failure. However, all other devices generate a minor alarm in response to the same failure.

By default, DX NetOps Spectrum generates a minor alarm in response to an authenticationFailure trap. You can create a device model type, and you can configure support for the trap in the event and alert configuration files for this device model type. This support overrides DX NetOps Spectrum default processing for the authenticationFailure trap for this model type only.

Unique Watches

You can generate events and alarms that are based on the results of a watch. The GnSNMPDev model type provides a number of predefined watches that can be enabled for individual models.

You can customize the watch implementation on the models that represent your device for each applicable GnSNMPDev model. However, you can avoid repeating this customization on each model by creating your own device model type to implement the customized watch.

All of the information that makes up a watch is stored as attributes in the model type specified in the watch. The only exception is the probable cause information that is created for an alarm that results from the watch. This information is stored in the ProbCause model type .

Because you have not created the ProbCause model type with your Developer ID, you lack permission to export and distribute it with your management module. As a result, the probable cause information for the watches that you have created is not distributed. To solve this problem, derive a new model type from the ProbCause model type. The probable cause information for any watches for any of your management modules is automatically stored in this derived model type. Because you have created this derived model type, you can distribute it with your management module.

NOTE
For more information, see the Watches section.

Interface Model Creation

If your device does not advertise interface (port) information in the MIB-II standard interface table but instead uses information from a proprietary MIB, DX NetOps Spectrum cannot model the associated interfaces.

Without interface models, you cannot resolve connections to the interface level, nor can you monitor the status of each interface. To work around this problem, create a new application model type that includes support for the proprietary MIB with the interface information.

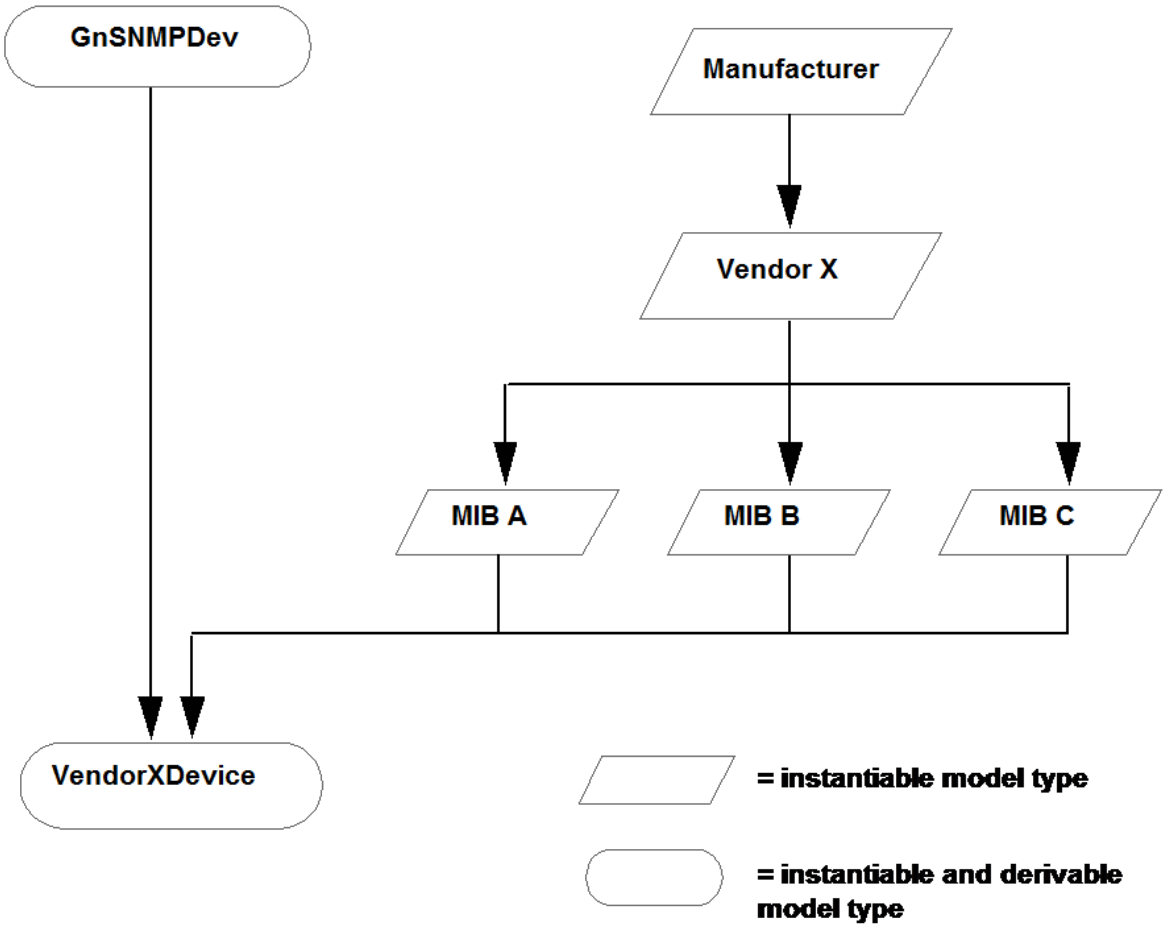
New Device Model Type

DX NetOps Spectrum offers multiple options for creating device model types. The topics in this section describe some of the factors to consider. A new device model type requires some or all of the following tasks:

- Understanding the database derivation and MIB requirements
- Creating the model type and setting required attributes
- Selecting discovery and identification mechanisms
- Making desired customizations
- Making the new model type distributable to other DX NetOps Spectrum hosts

New Device Model Type Design

The device model type database architecture for developing new device model types organizes all of the proprietary MIBs that you import into separate MIB model types. These MIB model types can then be derived directly into the device model type, as shown in the following diagram:



This scheme has the following advantages:

- A single MIB can be derived into multiple model types. For example, you can use the same MIB to create multiple device model types or a device and an application model type while maintaining the attribute IDs.
- Vendor MIBs can be organized in a single collection.
- Convenient access to MIB information is available directly from the device model. For example, you can access OneClick views, watches, and logging and polling information about proprietary vendor attributes from the device model.

If the new device model requires access to additional MIBs from DX NetOps Spectrum, the simplest method is the MIB import mechanism that is available in MIB Tools. This mechanism creates attributes from these objects and makes them available to all models that represent a device in the SpectroSERVER database.

NOTE
For more information, see the Device Management Reference section. The MIB import mechanism distributes the new MIB across all SpectroSERVERs in a distributed environment.

New Device Model Type Creation

After selecting a database scheme, use the Model Type Editor to create your model types. Derive all device model types from the GnSNMPDev model type.

Note: For more information, see the Model Type Editor section.

NOTE
When you are using the Model Type Editor to create a device model type, remember to set the model type flags correctly for the new model type.

New Device Model Type Configuration

A few steps are involved in configuring a new device model type. Complete the following tasks:

- Set model type flags
- Set attribute values
- Map a device or a device family to the new device model type
- Configure serial number handling

The following sections provide high-level information about these configuration steps.

For more information, see the Model Type Editor section.

Model Type Flags Setting

Set the values of model type flags so that models of the new model type behave as expected. Each flag represents a Boolean value and can either be selected (set to TRUE) or not (set to FALSE).

In most cases, you set the Visible, Instantiable, and Derivable flags to TRUE.

- **Visible flag**
  Makes the model type visible to all Model Type Editor users. If set to FALSE, the model type is only visible to a user with the same Developer ID as the one used to create the model type.
- **Instantiable flag**
  Lets you instantiate a model of this model type in OneClick.
- **Derivable flag**
  Lets this model type be used as a base for other model types.

In most cases you should set the No Destroy, Unique, and Required flags to FALSE.

- **No Destroy flag**
  If set to TRUE, prevents users from destroying a model of this type in OneClick.
- **Unique flag**
  If set to TRUE, only lets one model of this model type be instantiated in OneClick.
- **Required flag**
  If set to TRUE, a model of this model type must always exist in the SpectroSERVER database.

Attribute Values Setting

Once you have created your device model type, use the Model Type Editor to set the default value of several attributes. Some of these settings are used to configure built-in capabilities that are inherited by deriving from the GnSNMPDev model type. The following section describes the attributes and settings.

- **CompanyName**
  Is the name of the company that developed the management module.
- **Description**
  Is an attribute that exists in the MMDeveloper group and in the CommonInfo group. The Description attribute in the MMDeveloper group has a default value of Generic SNMP Device Management Module. We recommend resetting this default value to a description of your management module. The Description attribute in the CommonInfo group can be similarly reset or left empty.
- **Desc_Key_Word**

Enables resolution of multiple device model types. If the System_Desc_Verify or Vendor_Object_ID discovery mechanisms identify multiple device model types, a search of sysDescr looks for a substring match for the value of this attribute.

- **DeviceSerialAttr**
  Is the device serial number. Set the value to the attribute ID of the external attribute that contains the serial number. When the model is created, it reads this external attribute and writes it into Serial_Number.
- **DeviceType**
  Identifies the device. A default value is required for this description attribute when the DeviceNameList mechanism is not used for identification. Setting the default value guarantees that a value is present for displaying, sorting, and filtering.
- **DeviceTypeDiscEnable**
  Enables or disables DeviceType naming intelligence. The default value (true) is appropriate for most device model types. However, set this value to false under either of the following conditions:
  – A new device model type has been derived from a base model type other than GnSNMPDev. The new type has specialized DeviceType naming intelligence that is inappropriate for the derived device model type.
  – A more appropriate DeviceType name can be set in the catalog for the derived model type.
- **Disposable_Precedence**
  Is evaluated during device model type discovery when multiple model types are identified as candidates. The higher value is the chosen model type.
  This value is also evaluated when a model is created with the same MAC address as a previously existing model. In this case, the disposable_precedence attributes of both models are evaluated. The model with the higher value replaces the existing model by appropriating its CONNECTs associations.
- **Enable_IH_Enterprise_Disc**
  Enables or disables the automated setting of the Manufacturer and App_Manufacturer attributes based on the enterprise ID term of the device sysObjectID.
  The default value, true, is appropriate for GnSNMPDev because it is used to model devices from various manufacturers. However, for a new device model type that is derived from GnSNMPDev with a known device manufacturer, we recommend setting the value of Enable_IH_Enterprise_Disc to false. With that attribute set to false, set the default values of the Manufacturer and App_Manufacturer attributes to the appropriate names.
- **Manufacturer**
  Is the name of the vendor that manufactures the device.
- **MMName**
  Is the name of the management module.
- **MMPartNumber**
  Is the part number that you plan to assign to the management module.
- **System_Desc_Verify**
  Provides a device model type discovery mechanism that parses the sysDescr for firmware version information. Clear this default value if you are not using this discovery mechanism. It interferes with the other discovery methods if enabled.
- **System_Oid_Verify**
  Is a legacy attribute. Refer to SysOIDVerifyList.
- **SysOIDVerifyList**
  Used in conjunction with DeviceNameList. Populating this list attribute with sysObjectID values enables device model type discovery intelligence to match the list against the device sysObjectID value. If a match is found, this model type is selected as a possible candidate for modeling.
- **Vendor_Name**
  Is the name of the company developing the management module.
- **Vendor_Object_ID**
  Provides a device model type discovery mechanism by which a partial sysObjectID match identifies a device model type.
- **VendorIDVerifyList**

Maps devices to device model types based on whether the device supports specific MIB objects. Used during Discovery with VendorOIDVerifyList.
Specify the list of enterprise IDs to compare against the device to model. If a match is found, the corresponding MIB object specified in VendorOIDVerifyList is read from the device.

- **VendorOIDVerifyList**
  Maps devices to device model types based on whether the device supports specific MIB objects. Used during Discovery with VendorIDVerifyList.
  Specify the list of attribute IDs for the MIB objects to read from the device.
- **Verify_Mismatch_Model**
  Causes DX NetOps Spectrum to perform checks for a device model type match with the device that is modeled. Set this attribute to true.

Device Mapping

Each device on the network requires a unique identifier. Most commonly the MIB-II object sysObjectID provides this unique identifier. Vendors typically assign a unique sysObjectID value for a device, creating a one-to-one mapping. Vendors often advertise this information in a product MIB, where you can find the mapping of sysObjectID to device model type.

You can use the Model Type Editor to identify a device in several ways:

- If your device provides a unique sysObjectID value, use the process described in Map Using Unique sysObjectID Values.
- If your device does not provide a unique sysObjectID but does provide a unique substring within sysDescr, use the process described in Map Using sysObjectID and Strings in sysDesc.
- If your device does not provide a unique sysObjectID but does provide a firmware version text string in sysDescr, use the process described in Map Using Firmware Version Strings in sysDesc.
- If your device does not provide any of the previously described information, you can map the device to a device model type based on whether the device supports specific MIB objects in a proprietary MIB. Refer to Map Using a MIB Object.

Map Using Unique sysObjectID Value

If your device has a unique sysObjectID value, you must relate your new device model type to the sysObjectID to help ensure that DX NetOps Spectrum selects the new device model type to represent the device. To do this, add the sysObjectID value to the SysOIDVerifyList model type attribute. If the new device model type represents a family of devices, then add each sysObjectID value.

> **NOTE**
> If another model type contains the same sysObjectID value in its SysOIDVerifyList attribute, it is possible that DX NetOps Spectrum will choose the other model type to represent a device with this sysObjectID. If this occurs, you should change the disposable_precedence attribute value on your device model type to a higher value than that of the other model type. For example, if the other model type has a disposable_precedence value of 10, change the disposable_precedence value on your model type to 11.

To provide identification to your model, configure the model type to display a different device name for each of the devices that the model type is designed to support. For example, assume your device model type represents the 8480 series of switches made by MySwitch, Inc. Instead of seeing the device name MySwitch_8480XX for all of the switches in the 8480 family, you want to display the model number of the switch, as appropriate. If DX NetOps Spectrum is modeling an 8480-09 switch, the model should display the device name MySwitch_8480-09. If DX NetOps Spectrum is modeling an 06 switch, the model should display the device name MySwitch_8480-06.

Follow these steps:

1. Set the SysOIDVerifyList attribute equal to the sysObjectID(s) of the devices that the model type represents.

2. Set the DeviceNameList attribute equal to the device names that apply to each sysObjectID listed in the SysOIDVerifyList attribute.
3. Specify the same number of names in the DeviceNameList attribute as there are sysObjectIDs listed in the SysOIDVerifyList attribute. List the names in the same order as their corresponding sysObjectIDs.
4. Clear the System_Desc_Verify default value.
   The DeviceNameList attribute only works for device model types that use the SysOIDVerifyList attribute model type discovery mechanism. Verify that both lists have the same number of entries. Otherwise, the DeviceType attribute is not set correctly.

Map Using sysObjectID and Strings in sysDesc

If your device does not provide a unique sysObjectID, a partial or complete match of the sysObjectID in combination with a sysDescr substring can provide unique identification.

You can set up the relevant attributes to enable this functionality on the model type.

Follow these steps:

1. Set the Vendor_Object_ID attribute equal to the partial or complete sysObjectID value returned by your device. Only the first seven terms up to the enterprise ID are used for comparison.
2. Set the Desc_Key_Word attribute equal to the unique, partial sysDescr value returned by your device.
3. Set the DeviceType attribute to be equal to the desired identification string.
4. Clear the System_Desc_Verify default value.

Map Using Firmware Version Strings in sysDesc

If your device does not provide a unique sysObjectID or a unique sub-string within sysDescr, check whether sysDescr provides a unique firmware version. This discovery mechanism searches the sysDescr value for either “Version” or “Revi.” If one of these strings is found, the value of System_Desc_Verify is compared against the text that follows these key words. If a match is found, the device model type is selected. In the case where multiple model types have the same System_Desc_Verify value, a substring in sysDescr can be compared by setting the Desc_Key_Word.

You can set up the relevant attributes to enable this functionality on the model type.

Follow these steps:

1. Set the System_Desc_Verify attribute to be equal to the contents of sysDescr that follow the key text noted above.
2. Set the Desc_Key_Word attribute to be equal to the unique, partial sysDescr value returned by your device.
3. Set the DeviceType attribute to be equal to the desired identification string.

Map Using a MIB Object

If your device does not provide a unique sysObjectID or a unique sub-string or firmware version within sysDescr, check if it supports a proprietary MIB. You can map the device to a device model type based on whether the device supports specific MIB objects.

This discovery mechanism compares the enterprise ID of the device against each enterprise ID specified in the VendorIDVerifyList attribute. If a match is found, the MIB object specified in the same instance of the VendorOIDVerifyList attribute is read from the device. If the SNMP read succeeds, this model type is added to the list of model type candidates (from which the model type with the highest disposable_precedence attribute value is ultimately selected). The enterprise ID match mechanism is implemented for performance reasons: the SNMP read is only initiated for targeted devices.

You can set up the relevant attributes to enable this functionality on the model type.

Follow these steps:

1. Add the enterprise ID of the device to model with this model type to the VendorIDVerifyList attribute.

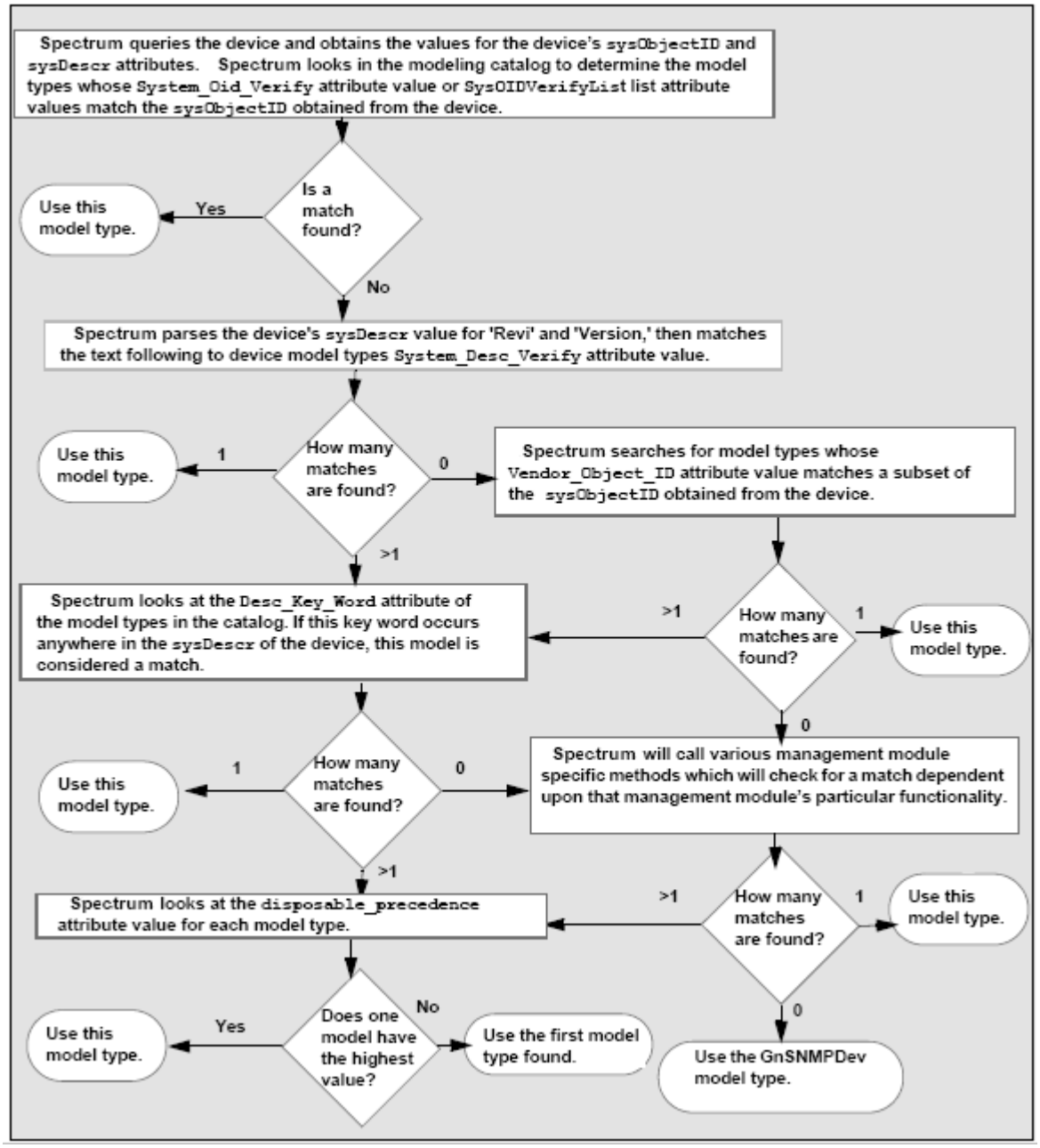
2. Add the attribute ID of the MIB object to read from the device as the corresponding instance in the VendorOIDVerifyList attribute.
3. Repeat the preceding steps for each enterprise ID/attribute ID pair to evaluate in conjunction with one another.
4. Set the DeviceType attribute equal to the desired identification string.

Discovery and Identification Flowchart

This flowchart identifies the steps that DX NetOps Spectrum takes to determine the device model type to represent a device. The flowchart includes discovery and identification mechanisms:

- Using unique values in sysObjectID
- Using strings in sysObjectID and sysDesc
- Using firmware version strings in sysDesc

You can also map a device (or a device family) to a new device model type based on whether the device supports specific MIB objects in a proprietary MIB.



Configure Serial Number Handling

Device model types contain an attribute for setting and displaying the serial number of the modeled device: Serial_Number (0x10030). Enter the appropriate serial number value in any of the views where the attribute is displayed. Or, if the serial number is available as an external device attribute, you can configure the model type to retrieve this value and set the Serial_Number attribute.

Follow these steps:

1. Verify that the external attribute that contains the serial number is not a list attribute and is of type TEXT_STRING or OCTET_STRING.

2. Set the value of the DeviceSerialAttr (0x3d0063) attribute for this device model type in the Model Type Editor. Make this value equal to the ID of the external attribute that contains the serial number.
   When a model of this model type is instantiated, DX NetOps Spectrum sets the Serial_Number attribute so that the value is equal to the value of the external attribute.

Creating a New Application Model Type

This section describes how to expand support for a device using application model types. All application model types are derived from a series of standard model types, called *derivation points*. An Application often corresponds to the functionality of a MIB.

Derivation Points and Model Fragments

Select derivation points and use them as base model types for new application model types. Derivation points have the functionality to support different types of applications. When you derive a new model type from one or more of these derivation points, the model type inherits the derivation point functionality.

Some derivation points require the use of model fragments. The available model fragments are model types with attached inference handlers. These inference handlers provide the model fragments with certain behaviors and intelligence, such as the ability to create port or board models. To use the functionality from these inference handlers, map attribute IDs from the model type that represents the MIB to specific model fragment attribute values.

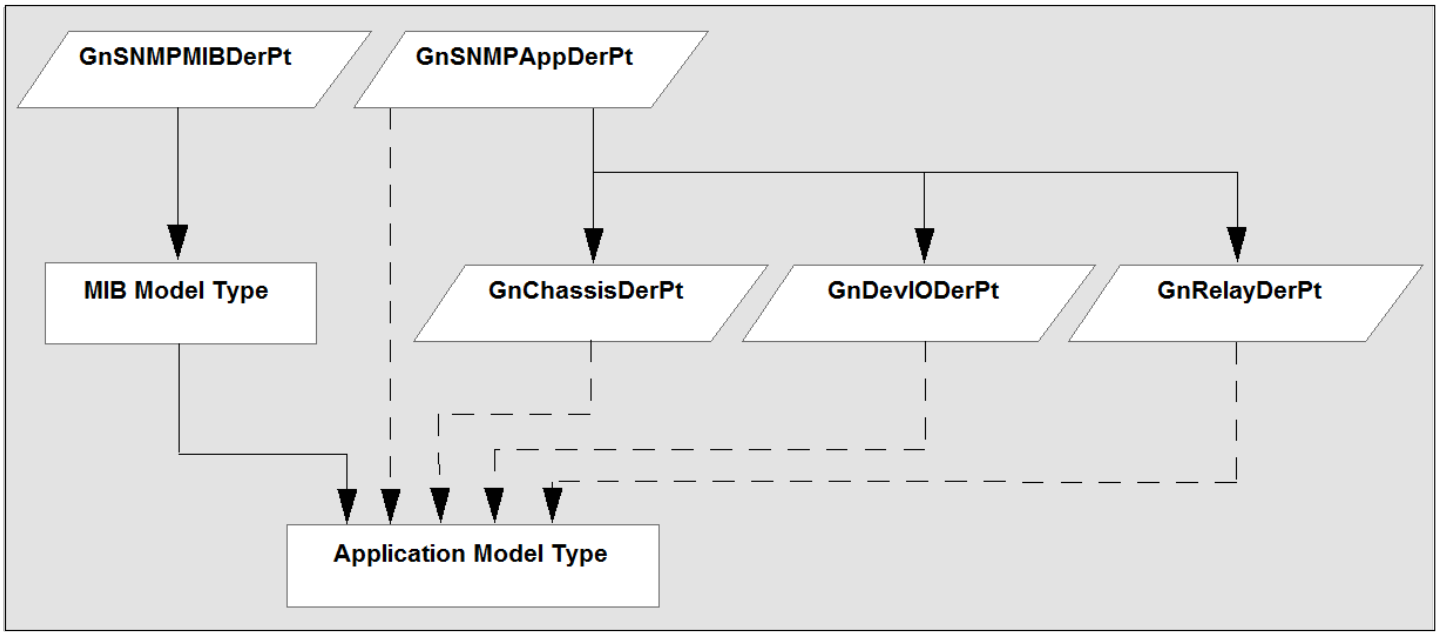
Model fragments are typically included as base model types for the GnSNMPDev derivation points that require them. However, it can be necessary to add a model fragment as a base model type to a new model type to gain the capabilities of the inference handler that is attached to the model fragment.

The following model types can be used as application derivation points:

- GnSNMPMibDerPt
- GnSNMPAppDerPt
- GnChassisDerPt
- GnDevIODerPt
- GnRelayDerPt

The following figure shows the application derivation point hierarchy and sample derived model types. The lines connecting the model types denote the inheritance structure.

NOTE
Select only one of the dotted line paths for the derivation hierarchy of your new application model type.



The derivation points for application model types are all designed to provide specific functionality. The GnChassisDerPt, GnDevIODerPt, and GnRelayDerPt derivation points have model fragments that enhance this functionality. The following table shows each derivation point, the application model type that it creates, and its associated model fragments.

Derivation Point	Model Type	Associated Model Fragment
GnSNMPMibDerPt	MIB Model Type	N/A
GnSNMPAppDerPt	Application model type with no requirement to manage ports or boards.	N/A
GnChassisDerPt	Application model type to model chassis functionality. Provides management for ports and boards.	GnChassis_MF
GnDevIODerPt	Application model type for devices that require port management but not board management (such as switches or terminal servers).	GnDeviceIO_MF
GnRelayDerPt	Application model type for repeater functionality	GnDataRelay_MF

Derivation Point

GnSNMPAppDerPt includes the functionality that is required for an application model type. GnChassisDerPt, GnDevIODerPT, and GnRelayDerPt are derived from GnSNMPAppDerPt and therefore inherit this functionality. Each also includes some specialized functionality for managing ports and boards.

If your device does not manage ports and boards and you are only interested in expanding support, use GnSNMPAppDerPt to derive your application model type.

If your device uses other MIBs to extend the functionality of MIB-II to manage ports and boards, use GnChassisDerPt, GnDevIODerPt, or GnRelayDerPt. Each of these derivation points uses model fragments that contain the attributes and

intelligence to create port models. The topic titled Board and Port Considerations explains how to select an appropriate derivation point for your port or board model type.

Board and Port Considerations

If you are modeling a chassis (a device with multiple modules or boards that can be inserted and removed), create the application model type from the GnChassisDerPt and GnRelayDerPt derivation points. These derivation points are used to model both the boards and ports in the device. The intelligence of these derivation points creates both board and port models.

If the device you are modeling is not a chassis, build your application model type from the GnDevIODerPt derivation point. The intelligence of this derivation point creates only port models (no boards) that are associated with the device model.

The structure and content of the relevant MIBs is important to consider. Chassis and data relay MIBs generally have a standard structure. A chassis MIB usually has a slot and board table. The index of the table represents the slot in the chassis where the board is plugged in.

A data relay MIB usually has two tables: a board table and a port table. The board table is indexed by the slot where the board is plugged in. The port table typically has two indexes: a board index and a port number. And vendors have devised several variations to the standard structures.

Port-Oriented Devices

Use the GnDevIODerPt to model port-oriented, non-chassis devices. Most MIBs for these port-oriented devices conform to the structural requirements to use GnDevIODerPt. The MIB must contain a port table, with at least one index, the port number. The derivation point executes a read_next (which is analogous to the get_next SNMP call) on this attribute. For each successful read of the index attribute, a port model with the appropriate instance ID is instantiated.

Chassis Devices

The structure of the MIBs that are associated with chassis devices is highly varied. We recommend reviewing the requirements of the GnChassisDerPt and the GnRelayDerPt derivation points. You can thus examine the variations and how they affect the modeling of the device.

GnChassisDerPt

The GnChassisDerPt is used to create an application model type that becomes the chassis manager application. This application is responsible for the creation and management of board models in the SpectroSERVER database. This chassis manager relies on three attributes (usually list attributes) for the information it needs:

- slot index
- board type
- board label

A single chassis manager application can be instantiated or managed by the main device model. The chassis manager intelligence expects the MIB to have a slot or board table that is indexed by an integer value. This value represents the slot into which a particular board is plugged. The intelligence performs a read_next on this slot index attribute. For each successful read, the intelligence creates a model in the database to represent that board. Because the intelligence can only reference one index value, all boards in the chassis require an entry in this single table of the chassis MIB.

In addition to finding the slot where a board is plugged in, the manager intelligence must determine the board type and label the board correctly. The board type and board label attributes determine this information. These attributes do not have to exist in the same table as the slot index attribute. The attributes must only exist in a table that uses the same indexing scheme as the table used to discover the boards.

The MIB can have all the board information in non-list attributes rather than in a table. In this case, the information that is supplied within the MIB applies to a single board. The slot index value is not an index into a table, but simply an integer

attribute that returns the slot where the board is located. The chassis manager intelligence tests the slot index attribute. For a non-list attribute, a read is used instead of a read_next to get the board number. If the slot index attribute is not a list attribute, the board type and board label attributes are not list attributes.

GnRelayDerPt

The GnRelayDerPt derivation point is used to model the ports on a chassis. You can use this derivation point with GnChassisDerPt to create one application model. Or you can use it on its own to create an application model that is separate from the chassis manager.

The term *chassis support application* describes an application that was built using GnRelayDerPt. This derivation point provides support to the chassis manager application (such as modeling the ports for each board). Unlike the chassis manager application, multiple chassis support applications can be instantiated under the main device model. This ability lets you model a chassis whose boards support different protocols.

Although all the boards can show up in the slot table of the chassis, a MIB that corresponds to the appropriate protocol can manage the data relay component of each board. Each of these protocol-dependent MIBs must be modeled as separate application models (built from the GnRelayDerPt derivation point). The ports on each board can then be discovered and modeled.

The typical structure of a data relay MIB has two tables: a board table and a port table. Do not confuse the board table with the slot table that is used with the chassis manager. In some cases, they can be the same table. However, the board table in the data relay MIB has an entry for each board that the MIB supports. Typically the board table is indexed by the position of the board in the chassis. For example, if the data relay MIB is an Ethernet MIB, any board that supports the Ethernet protocol (typically a repeater board) has an entry in the board table of this MIB. If a FDDI board is plugged into the chassis, the board creates an entry in the common slot table. However, this new board does not appear in the board table of the Ethernet MIB. Instead, it appears in the board table of the FDDI MIB.

In addition to the board table, the data relay MIB has a port table. For each port that the MIB supports, this table contains a corresponding entry. The tables often contain the status and statistical information for each port. The port table contains two indices: a board index and a port index. Because the port table contains a board index, the chassis support intelligence can associate the port models with the appropriate board models; the board index supplies the mapping of a port to a board.

GnDataRelay_MF is the model fragment within the GnRelayDerPt derivation point, which contains the attributes and intelligence to model the ports of each board and associate those port models with the appropriate board model. The GnDataRelay_MF model fragment intelligence works with only one board table and one port table. This requirement matches the typical structure of a data relay MIB. If your data relay MIB contains sets of tables - for example, a set of board and port tables for each of the major protocols - you must separate these MIB tables or groups into separate model types. Use each model type as a base for the appropriate application that is built with the GnRelayDerPt.

In some cases, the data relay MIB lacks the typical structure: both a board table and a port table, with the port table indexed to provide the physical mapping of ports to boards. For example, the chassis device uses a MIB with a different indexing scheme for accessing the port information. The FDDI MIB indexes the port table by the SMTIndex and the PortIndex. The SMTIndex is not used to identify the board where the FDDI port is physically located.

This situation can also be created if a vendor reuses a MIB from another device. The original device that the MIB was designed to manage was a port-oriented device (no boards, only ports). The vendor supplies the same functionality in a board that can be plugged into its chassis, and has used the original MIB to manage the ports on that board. The port table does not contain a board index; the device does not identify which board has a given port.

In such a case, implement the DataRelay_MF model fragment functionality as you would with a port-oriented device.

Application Model Types

Complete the following tasks when you create an application model type:

- Import the required MIBs.
- Derive the application model type.
- Set up application model discovery.
- Set the model name.
- Map the model fragments.
- Set the model type flags.

Use the Model Type Editor to accomplish each of these tasks. The following sections explain why these tasks are required.

NOTE
For more information, see the Model Type Editor section.

Required MIBs Import

When you create an application model type, in some cases the MIB model type already exists in DX NetOps Spectrum. Or you might need to provide access to the new MIB. To provide access to the new MIB, you have two options:

- Use the Model Type Editor to import the MIB directly into the new application model type.
- Create a MIB model type.

If the MIB will be derived into multiple model types, consider deriving the MIB into a separate model type that can be used as a derivation point. The attribute IDs can then be maintained across the model types. Organizing this MIB model type under a new or existing vendor model type maintains database organization.

To create a MIB model type, derive a new model type from GnSNMPMibDerPt. Import the compiled MIB, and supply the SMI (Structure of Management Information) path.

NOTE
If the wrong SMI Path is used, the Model Type Editor does not produce an error. However, when you view imported attributes, the OID Prefix value is incorrect.

Application Model Type Derivation

To derive an application model type, use the Model Type Editor to set the GnSNMPAppDerPt model type as the current model type. Then create a new derived model type.

After you have created the new application model type, add any MIB model type that you created as a base model type to the new application model type.

The new application model type now contains two base model types:

- GnSNMPAppDerPt model type
- custom MIB model type

Application Model Discovery

When a device model for a specific device is instantiated, DX NetOps Spectrum queries the Model Type catalog. Most application model types that are derived from GnSNMPAppDerPt are queried. The query retrieves the value of the default_attr or default_attr_list attribute of each of these model types. DX NetOps Spectrum then queries those attributes on the device MIB. When a match is found between an attribute value retrieved from the application model type and the corresponding attribute value retrieved from the MIB, DX NetOps Spectrum instantiates a model of this model type.

You can use either the default_attr_list or default_attr to specify attribute IDs from attributes of a MIB model type. DX NetOps Spectrum queries the attributes whose attribute ID is contained in the default_attr or default_attr_list. If default_attr_list is used, DX NetOps Spectrum goes through the list of attribute IDs. The first supported attribute ID that is found is used to instantiate that application model to represent the MIB functionality.

Set the Default Attribute Values

The default_attr_list attribute lets you specify multiple attribute IDs, and the default_attr attribute lets you specify one attribute ID. Each attribute lets DX NetOps Spectrum identify the application model type that represents the MIB functionality.

The default_attr_list attribute is helpful when you have one device that supports a single table in a MIB rather than the entire MIB, and another device that supports other objects in the same MIB, but not in the particular table that the other device supports. In this scenario, use the default_attr_list attribute to specify multiple attribute IDs. This step ensures that the application model type that represents the MIB is instantiated for both devices even though they do not support the same MIB objects.

NOTE
Set the default_attr or default_attr_list in all application model types. When choosing a value, we recommend using an attribute from the MIB model type that represents a mandatory, non-list, external MIB variable. Using such an attribute is especially important when you create a chassis application.

Specify a value for default_attr.

Follow these steps:

1. Find the MIB attribute for the application model type with which you are working.
2. Use the attribute ID of this attribute to set the value of the default_attr attribute in the application model type.
   Look specifically at the attributes of the model type that represents the MIB. You can find the attribute IDs of the attributes of a model type on the Attributes tab in the Model Type Editor.

Specify values for default_attr_list.

Follow these steps:

1. Find the MIB attributes for the application model type with which you are working.
2. Use the attribute IDs of these attributes to specify values in the default_attr_list attribute in the application model type.

   NOTE
   Find the attribute IDs of the attributes of a model type on the Attributes tab in the Model Type Editor.
3. Set the Model_Group attribute to the decimal value of the model type handle of the application model.
4. Verify that the value of Model_Group is set appropriately.
   If Model_Group is set to 0, DX NetOps Spectrum only uses the default_attr attribute to identify the application model type that represents the MIB functionality.

Model Name Setting

Set the Model_Name attribute of the application model type to the appropriate value. By default, this value is used as the model name for any application model of this type.

Model Fragment Mapping

If your new application model type is derived from GnChassisDerPt, GnDevIODerPt, or GnRelayDerPt, use the model fragments that correspond to these model types. This practice ensures correct operation of port and board management. For a model fragment to function properly, use the Model Type Editor to map MIB attribute values from the application model type to model fragment attribute values. The model fragment gains access to information from the MIB that it uses to create and manage ports, boards, and interfaces.

For example, the boardIndex_Attr is one of the required attributes for the GnChassis_MF model fragment, which is used with the GnChassisDerPt derivation point. This attribute lets the model fragment discover the boards that are present in a chassis. The boardIndex_Attr must be set to the index attribute value in the board (group) table of the chassis or repeater MIB. The index attribute usually returns an integer value or a series of values that represents a board number.

Certain derivation points have associated model fragments. The attributes that are associated with that model fragment are available to any model type that was based on these derivation points. To gain the functionality of a model fragment that is not included with one of your base model types, include that model fragment as a base model type.

Model Type Flags Setting

When creating an application model type, set the value of a few different flags to ensure that models of this model type work correctly. These flags are available on the Flags tab of the current model type in the Model Type Editor. Each flag represents a Boolean value and can either be selected (set to TRUE) or deselected (set to FALSE).

In most cases, set the Visible, Instantiable, and Derivable flags to TRUE.

- If the Visible flag is set to TRUE, the model type is visible to all Model Type Editor users. Otherwise, the model type is only visible to a user with the developer ID that was used to create the model type.
- If the Instantiable flag is set to TRUE, you can instantiate a model of this model type in OneClick.
- If the Derivable flag is set to TRUE, this model type can be used as a base for other model types.

The No Destroy, Unique, and Required flags are typically set to FALSE.

- If the No Destroy flag is set to TRUE, users cannot destroy a model of this type in OneClick.
- If the Unique flag is set to TRUE, only one model of this model type can be instantiated in OneClick.
- If the Required flag is set to TRUE, a model of this model type must always exist in the SpectroSERVER database.

Modeling Ports and Boards

When you create application model types from GnChassisDerPt, GnDevIODerPt, and GnRelayDerPt, these applications create the port and board models that are required to represent your device. DX NetOps Spectrum generally uses two model types to model these boards and ports: GnModule and GnPort. You can derive new model types from these model types for customization purposes.

In OneClick, you can view the ports for a device on the Interfaces tab in the Component Details panel. To view the boards for a device, use the Locater tab to search for the boards by model type name.

Modeling Boards with GnModule

Typically a board is modeled for one reason: to be a container for the port models that are physically located on it. In the chassis support of GnSNMPDev, the GnModule model type models many different types of boards.

Derive all new board model types from the GnModule model type. Two GnModule attributes help to define the type of board that a particular model represents:

- **gnType**
  This attribute provides the board type as read from the chassis slot table. When each GnModule model type is instantiated, the chassis manager intelligence supplies the gnType attribute.
- **gnName**
  This attribute is supplied by the chassis manager, which uses information in the chassis slot table when the board is first created.

Modeling Ports with GnPort

Port models are very similar to board models. GnSNMPDev provides one port model type that is sufficient for most modeling needs. The GnPort model type is the default model that is used to model ports using the GnSNMPDev chassis support.

Port and Board Model Information

This following information is not necessary for modeling your ports and boards. We provide this information to help you understand how the information for each board and port is read and displayed in OneClick.

All external attributes that are associated with the boards and ports are read through the application models that support the board and port models. The application models are used because they contain the MIB model types and thus the external attributes that are associated with the boards and ports.

In OneClick, you can view the ports for a device on the Interfaces tab in the Component Details panel. To view the boards for a device, use the Locater tab to search for the boards by model type name.

How to Add Support for Additional Traps

DX NetOps Spectrum notifies you about significant occurrences on your network using traps (alerts from SNMP-compliant devices), events, and alarms.

- An *alert* is an unsolicited message sent out by a managed node on a network. A more specific definition of an alert depends on the management protocol that is used to report the alert. In general, DX NetOps Spectrum uses SNMP as the management protocol to communicate with network devices. Alerts that an SNMP-compliant device generates are called *traps*. DX NetOps Spectrum receives traps and converts them to events for further processing.
- An *event* is an object in DX NetOps Spectrum that indicates that something significant has occurred within DX NetOps Spectrum itself or within the managed environment. Events always occur in relation to a model. When a managed element on the network generates an alert, this alert is mapped to a DX NetOps Spectrum event in the appropriate AlertMap file. The event is then generated and takes on the event code that is specified in the AlertMap file.
- An *alarm* is an indication that a user-actionable abnormal condition exists on a model. A model usually detects an abnormal condition when an event occurs, and the EventDisp file indicates that an alarm is generated.

When you create a model type, typically you add support for additional traps, events, and alarms. You can do this using the MIB Tools application and the Event Configuration application in OneClick. The high-level process is as follows:

1. Enable the OneClick preference that lets you select whether to install or to export the event and alarm support files from MIB Tools:
   a. Click View, Preferences in the OneClick Console.
      The Set Preferences dialog opens.
   b. Expand the MIB Tools folder in the left panel and select Show Advanced Map Options.
   c. Select Yes from the drop-down list.

   Enabling this option lets you use MIB Tools to export the files that support trap, event, and alarm processing to a user-defined directory. You can then package the files in your new management module.
2. Identify the MIB that contains the desired trap definitions.
3. In MIB Tools, import the MIB into the MIB Tools database.
4. Also in MIB Tools, map the traps to events, and specify the events that generate alarms (and the severity of the alarms).
5. While you are still in the Assign Trap Alarms dialog in MIB Tools, take the following steps:
   a. Under Advanced Options, select Export Trap Support.
   b. For Starting Event Code, enter the event code for the first trap that you have mapped.
      The event code is a 4-byte integer that is expressed in hexadecimal format. The first 2 bytes contain the developer ID, and the last 2 bytes identify the event with a unique number. You specify the event code for the first trap. The codes for the remaining traps are assigned sequentially based on the first.

      NOTE
      To identify your custom event codes in OneClick, and to prevent potential conflicts with other DX NetOps Spectrum event codes, we recommend using a starting event code that begins with your CA-assigned developer ID.

c. For Directory, click Browse, navigate to the directory where the event and alarm support files are exported , select the directory, and click Open. For example, browse to C:\win32app\<*vendor_name*>.

6. Click OK in the Assign Trap Alarms dialog.
   MIB Tools creates the appropriate event and alarm support files and exports them to the directory you specified.
7. In Event Configuration, complete the configuration of the events and alarms.
   For example, specify the symptoms, probable causes, and recommended actions for the alarms. These messages are displayed in OneClick when the alarms are generated.
   You can also specify event processing for one or more events, such as logging the event, using the event to clear an alarm, or generating another event using event rules.
   In addition, you can customize the default event message that is displayed in OneClick when the events are generated.

   NOTE
   For more information, see the Event Configuration section.

Distributing a New Certification

After you have created and customized model types, use the DX NetOps Spectrum Extension Integration (SEI) Toolkit to create a virtual CD (VCD) for distributing the new model types to other DX NetOps Spectrum hosts.

The SEI toolkit includes command-line tools for creating required files and for assembling and packaging extensions into a management module that you can distribute. The toolkit helps you create a management module that is compatible with software from CA and other third-party developers. It lets you install a module in your existing DX NetOps Spectrum environment with minimal installation or integration issues.

NOTE
For more information about the DX NetOps Spectrum Extension Integration toolkit, see the Extension Integration (SEI) Developer Reference section.

Cisco Device Management

This section explains how DX NetOps Spectrum supports Cisco device management.

Cisco Device Support Overview

The DX NetOps Spectrum Cisco device certifications provide Cisco MIB and trap support, descriptive device identification, OneClick views, Cisco technology support. The DX NetOps Spectrum Cisco device certifications also provide DX NetOps Spectrum standard capabilities for specific devices and firmware.

Examples of device-family certifications include Catalyst, PIX Firewall, Wireless LAN Controller, and Aironet.

Examples of firmware-based certifications include Cisco IOS, CatOS, and Unified Computing System (UCS).

If no specific device-family certification is available for your Cisco device, then one of the following firmware-based model types is used:

- Rtr_Cisco -- Models Cisco routers that are running IOS firmware.
- SwCiscoIOS -- Models Cisco switches that are running IOS firmware.
- RtrCatOS -- Models Cisco routers that are running CatOS firmware.
- SwCatOS -- Models Cisco switches that are running CatOS firmware.
- CiscoNXOS -- Models Cisco Nexus devices that are running NX-OS firmware.
- GnCiscoDev -- Models Cisco devices that are not running IOS or CatOS firmware.

MIB Sources

Depending on the Cisco device firmware, chassis and board or module information is found in the following MIB sources:

- **OLD-CISCO-CHASSIS-MIB**
  Cisco has deprecated this MIB. Therefore, the information can be incomplete. To view the contents of this MIB, see the Cisco Chassis View subview in OneClick.
- **CISCO-STACK-MIB**
  The CatOS devices support this MIB. This MIB is deprecated in favor of the ENTITY-MIB. To view the contents of this MIB, see the MIB Tools utility.

  NOTE
  For more information about MIB Tools, see the Certifications section.
- **ENTITY-MIB**
  This MIB contains the latest board or module information for new devices. Older devices, however, do not populate this MIB correctly. To view the contents of this MIB, see the Entity View subview in OneClick.

Cisco Unified Computing System

Cisco Unified Computing System (UCS) comprises a set of specialized devices working together, including the blades of the chassis and server. UCS supports the data center by delivering a dynamic IT infrastructure and by unifying network, computing, and virtualization resources.

DX NetOps Spectrum provides visibility into the following key components of Cisco UCS:

- **UCS Manager**
  A web services agent running on a Fabric Interconnect switch. The Cisco UCS manager supports an XML-based API for client interaction.
- Fabric Interconnect (FI) Switch
  - Typically two per UCS system; Cisco recommends a redundant configuration
  - Runs NX-OS
  - Hosts the UCS manager
- **Chassis**
  An agentless, switchless, blade server enclosure, with a maximum configuration of 40 chassis per UCS manager. Each chassis supports 8 half-width or 4 full-width server blades.
- **Blades**
  A server platform that serves as a virtualization host.
- **Service Profiles**
  Logical views of blade servers. Stored in the FI switch, they contain the blade server personality (identity and network information).

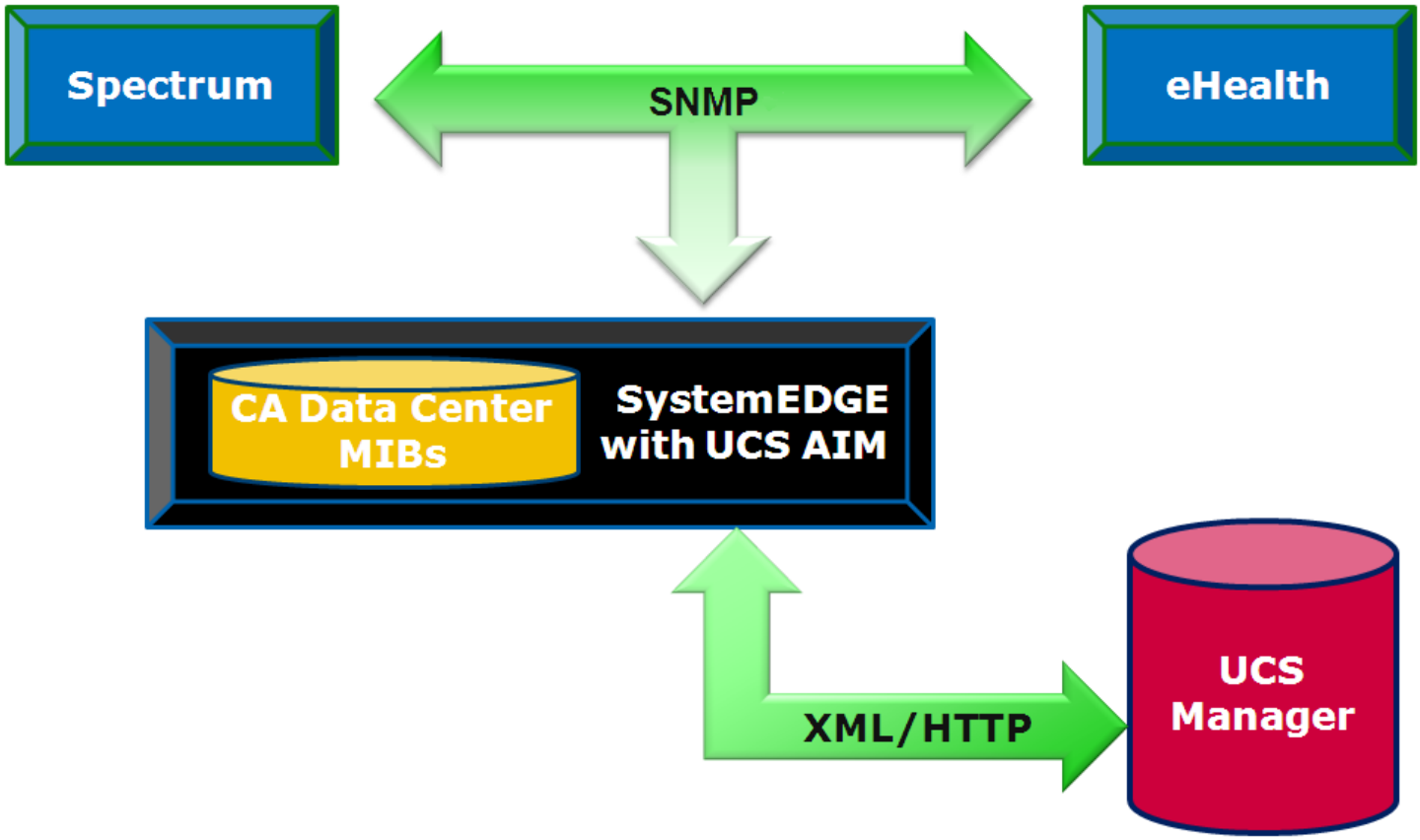
Solution Architecture for Managing Cisco UCS

You can enable DX NetOps Spectrum support for Cisco UCS by employing a specialized CA eHealth SystemEDGE Application Insight Module (AIM). This AIM communicates with the UCS XML-based API to obtain information about the UCS managed environment. The data is then written to a pair of CA-developed MIBs. This solution lets other SNMP clients, such as CA eHealth, leverage the AIM.

The UCS AIM is an extension of the CA eHealth SystemEDGE SNMP agent and can support multiple UCS systems. The CA-developed MIBS are:

- Generic data center MIB (CADATACENTERA)
- UCS-specific data center extension MIB (CACUCSEXTENSIONA)

As the following diagram shows, CA products such as DX NetOps Spectrum and CA eHealth use SNMP to connect to the CA eHealth SystemEDGE that hosts the UCS AIM to obtain Cisco UCS details. The UCS AIM leverages XML/HTTP to connect to the UCS Manager.



UCS Certification Features

DX NetOps Spectrum UCS certification features include:

- Automated device discovery and modeling -- Creates models for UCS components and maintains associations between blade models and any resident ESX hosts
- Connectivity -- Generates accurate physical (Layer 2) topology map of UCS system components
- Enhanced fault management -- Recognizes and suppresses symptomatic alarms, and aids fault isolation with proxy management
- Dedicated UCS views -- Provides visibility into UCS-specific data
- Intelligent trap forwarding -- Enables alarm generation on individual UCS components
- Chassis management (non-UCS-specific) -- Leverages the rich chassis management feature set of DX NetOps Spectrum

Automated Device Discovery and Modeling

The certification automatically models UCS system components on creation of the Host_SystemEDGE model that hosts the UCS AIM. This model can be identified as the Cisco UCS Manager. It creates an application model type of

cacucsaimApp when a UCS AIM MIB is detected. In turn, this application model creates UCS system components such as Fabric Interconnects, chassis, blades, service profiles, and more.

NOTE
Not all UCS components are modeled, such as fabric extenders, power supplies, media adapters, or interfaces.

Next, a search is performed for previously modeled ESX hosts on any of the blades. An association is created between the corresponding blade and ESX models to provide visibility into this hardware-to-software relationship.

Last, container models to represent each UCS system are created. These models reside in the same container (for example, Universe) as the CA eHealth SystemEDGE host. Each container provides a logical topological grouping of the UCS system components.

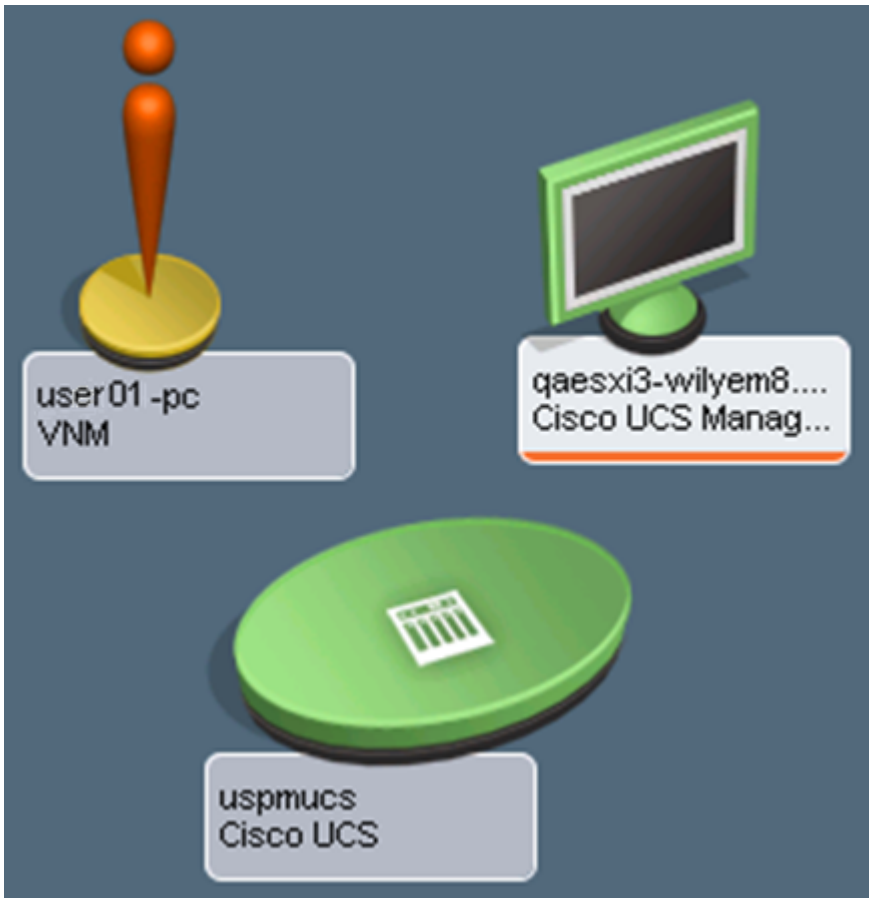
The UCS AIM MIB is polled regularly to collect status and modeling information from the UCS environments. For more information about configuring the polling interval, see Control Cisco UCS AIM Polling.

WARNING
In a given landscape, you cannot model multiple Host_SystemEDGE models whose UCS AIMs monitor the same UCS system. This configuration is not supported. If the Host_SystemEDGE that hosts a UCS AIM is a virtual device, model it before modeling the Host_SystemEDGE that hosts the corresponding virtual technology AIM. Otherwise, UCS containers can be incorrectly created inside a physical host container of the virtual technology. This situation disrupts the DX NetOps Spectrum fault-isolation algorithms.

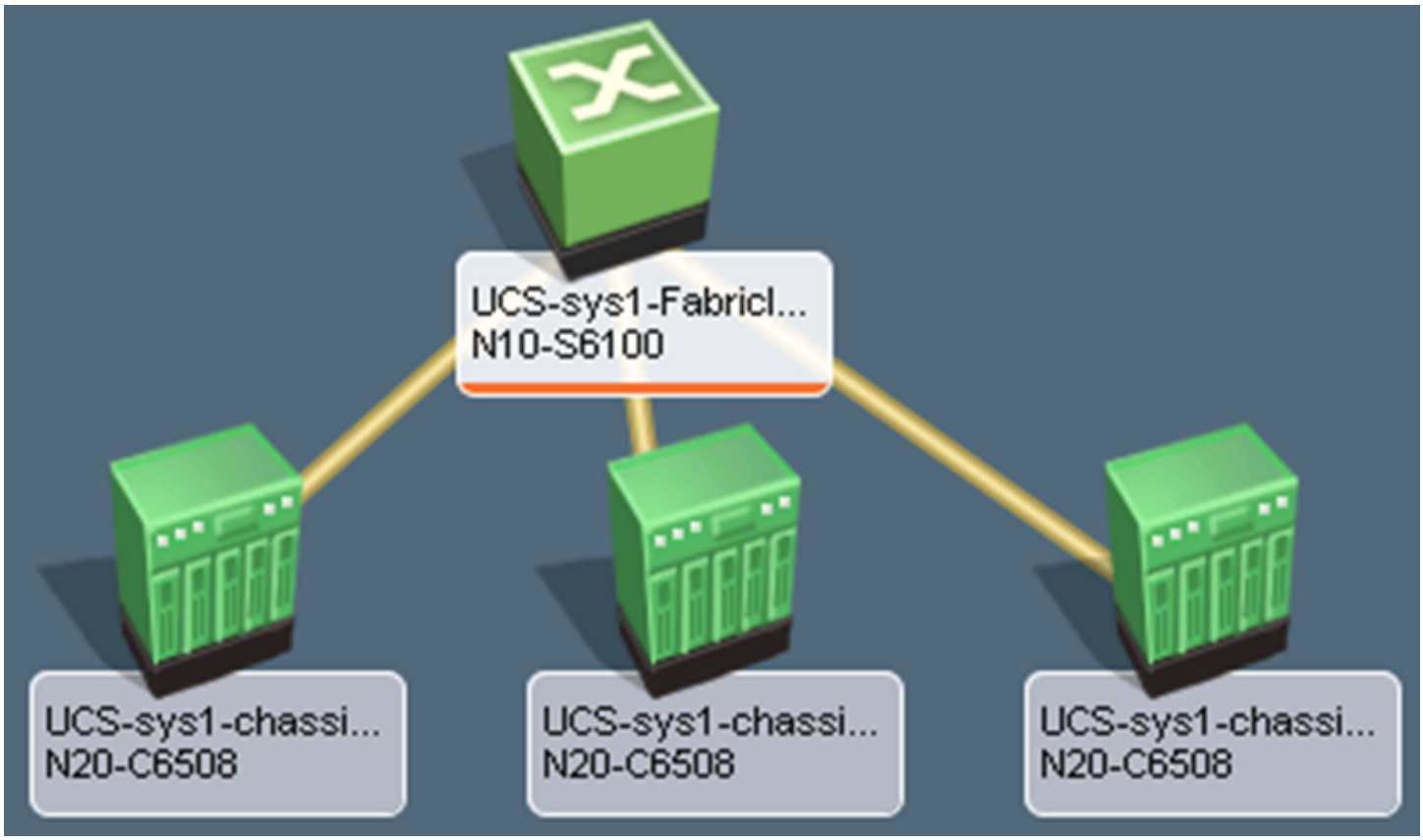
UCS Container Models

To represent a UCS container, DX NetOps Spectrum uses the standard container icon with a chassis logo. UCS containers have a model type of CiscoUCSContainer. Each container gathers together all the topologically significant models of a single UCS system (up to 2 FIs and 40 chassis). The contents of a UCS container cannot be modified.

UCS containers are displayed as in the following image:



The following image shows an example of the contents of a UCS Container:



UCS Fabric Interconnect Models

UCS Fabric Interconnects use the standard switch icon of DX NetOps Spectrum. If this device is modeled through IP address or through discovery, and if its NX-OS SNMP agent is enabled (it is disabled by default), a model of type CiscoUCSFabricInterconnect is created. Otherwise, automated UCS discovery creates a Pingable model. This model includes no device interfaces nor connectivity to upstream devices. Note that the IP address modeling can occur after UCS discovery, in which case the CiscoUCSFabricInterconnect model replaces the Pingable model.

UCS Fabric Interconnect models support dedicated UCS OneClick views and are the target models for UCS Fabric Interconnect traps and alarms.

UCS Chassis Models

UCS chassis use the standard chassis icon of DX NetOps Spectrum and have a model type of CiscoUCSChassis. The Interfaces tab in the Component Details panel of OneClick is enhanced for UCS chassis: the blades within the chassis are displayed to help with blade management.

UCS chassis also extend the fault isolation functionality of DX NetOps Spectrum to provide alarm correlation of collocated hardware resources.

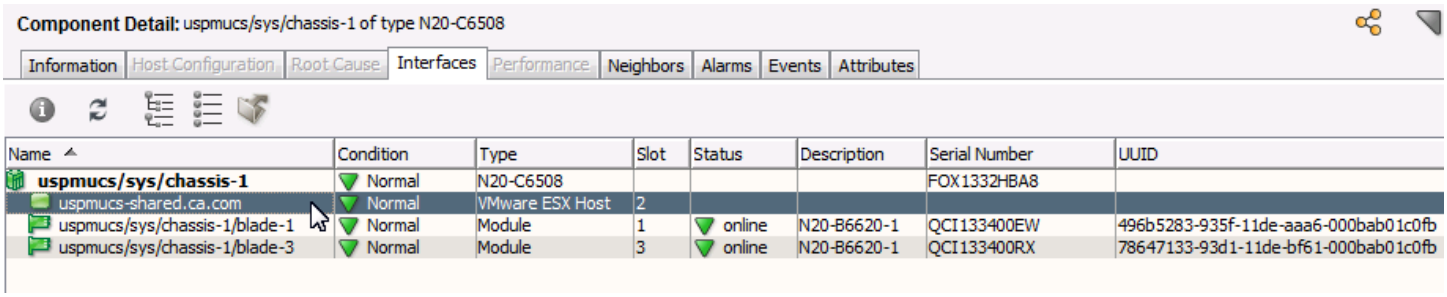
UCS chassis models support dedicated UCS OneClick views and are the target models for UCS chassis traps and alarms.

UCS Blade Models

UCS blades are modeled in DX NetOps Spectrum but, unlike Fabric Interconnect switches and chassis, are not visible inside their parent UCS container nor in any other location of the DX NetOps Spectrum topology. However, the UCS blades for each chassis are listed in the Interfaces tab of the chassis. UCS blades have a model type of CiscoUCSBlade.

DX NetOps Spectrum automatically makes associations between a blade and an ESX host resident on that blade. This association is done by performing a search for previously modeled ESX hosts and obtaining the UUID (Universally Unique Identifier) value. The blade UUIDs are then examined. When a match is found, the ESX host model is associated with the blade model. Only ESX hosts are supported for automatic association. DX NetOps Spectrum understands this blade (hardware)-to-ESX host (software) relationship and leverages it through enhanced fault isolation. You see meaningful alarm details, such as when the ESX host is out of contact because its blade failed.

Once the association is made, the ESX host model takes the place of the blade model in the Interfaces tab of the chassis model.



You can manually associate UCS blades with corresponding SNMP agent models to gain visibility into the blade (hardware)-to-agent (software) relationship.

UCS blade models support dedicated UCS hardware-based OneClick views, which include the following types of information:

- Statistical information such as CPU load, memory, and storage utilization
- Image inventory (BIOS & Firmware) and BIOS H/W configuration
- Physical interfaces of the blade server
- Service profile details

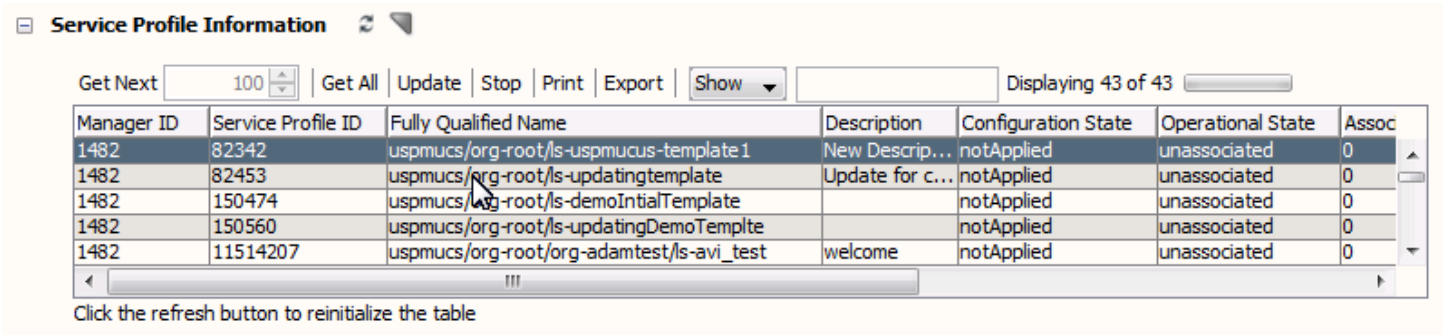
The UCS blade models are the target models for UCS blade traps and alarms.

UCS Service Profile Models

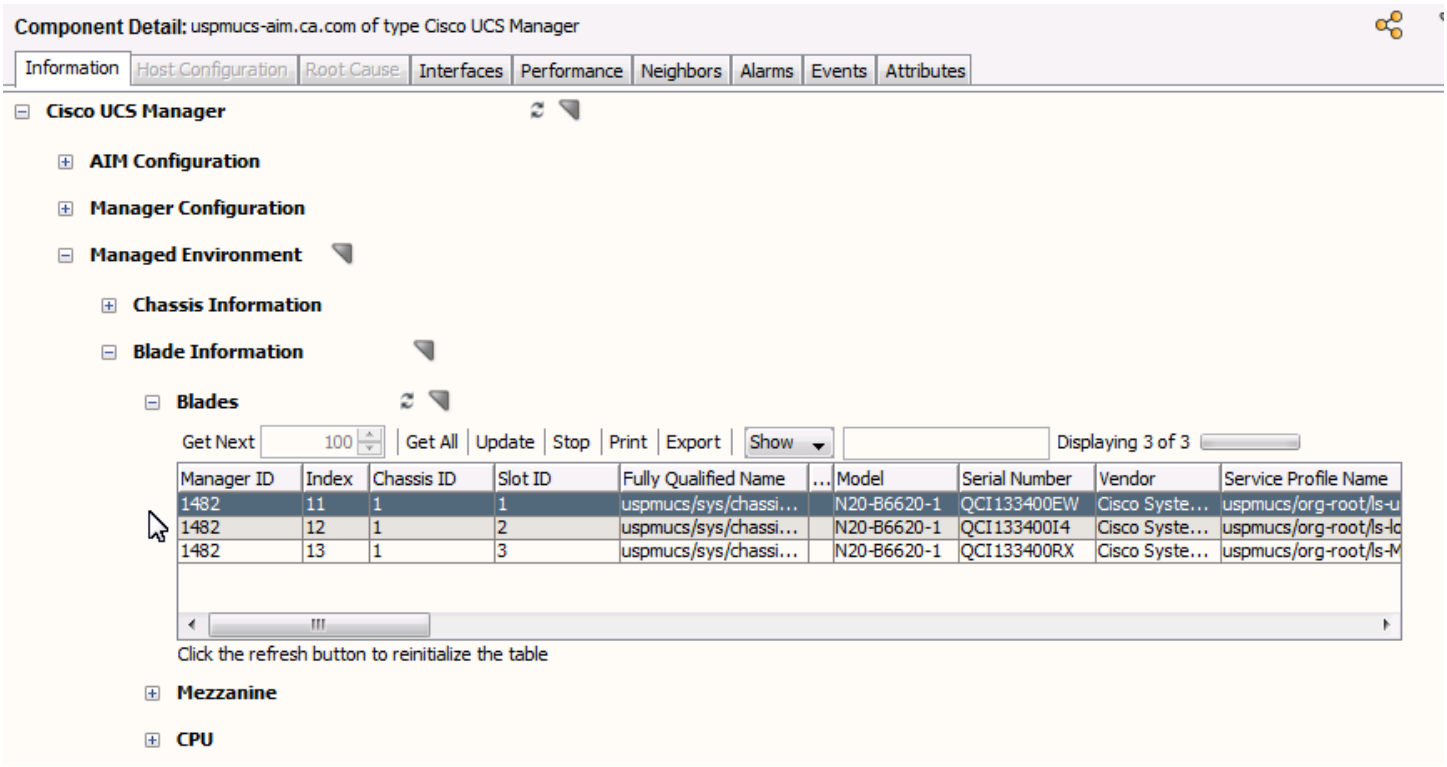
Blade servers that are provisioned in the Cisco Unified Computing System are specified by a service profile. A service profile is a software definition of a server and its LAN and SAN network connectivity. UCS service profiles have a model type of CiscoUCSServiceProfile.

Server, network, and storage administrators create service profiles. Service profiles are stored in the Cisco UCS 6100 Series Fabric Interconnects. When a service profile is deployed on a blade server, the UCS manager automatically configures the blade server, its network adapters, fabric extenders, and fabric interconnects to support the configuration specified in the service profile.

DX NetOps Spectrum creates models for each service profile that is defined by a UCS Manager. They can be viewed from the OneClick Locater tab which now includes a Service Profile Model Class search option. In addition, service profile details are displayed in various OneClick views. Select the Cisco UCS Manager, Managed Environment and the Service Profile Information option on the Host_SystemEDGE model that is hosting the UCS AIM. You can then see the name, ID, description, associated blade, and various states of all the service profiles in the UCS systems that the Host_SystemEDGE manages.



DX NetOps Spectrum also displays the service profile that is associated with each blade that is installed in each UCS Manager Chassis.



The UCS service profile models are the target models for UCS service profile alarms.

Connectivity

UCS Fabric Interconnect models participates in connectivity by providing the boundary switching node between the upstream devices and the blade servers in the chassis.

Upstream from the UCS Fabric Interconnect

The upstream connections from the Fabric Interconnect interface are done through standard bridging tables. These connections require the following steps:

- Enabling the native NX-OS SNMP agent in the Fabric Interconnect .
- Modeling the device ,either by IP address or through discovery.

Downstream from the UCS Fabric Interconnect

The downstream FCoE connections from a UCS Fabric Interconnect to its constituent chassis are shown as standard DX NetOps Spectrum L2 connections. These connections are created programmatically and not through standard bridging tables nor the UCS MIBs.

Enhanced Fault Management

Enhanced Fault Management for UCS involves two types of alarms:

- Fault Alarms
  - Indicate a problem with L2 availability
  - Are enhanced with special correlation logic
- Proxy Lost Alarms
  - Indicate that updated UCS information cannot be obtained from the CA eHealth SystemEDGE UCS AIM host
  - Include a Proxy Unavailable alarm for the host itself

UCS Fault Management enhancements also include chassis- and blade-level availability alarms and trap-generated alarms that indicate infrastructure and environment issues.

UCS leverages the benefits of alarm correlation:

- Pinpoint root cause
- Suppress extraneous alarms
- Correlate symptoms to root cause
- Show impact

UCS alarm correlation occurs at both the chassis level and the UCS System level.

Chassis-Level alarm correlation uses a domain that includes a chassis, its blades, and all SNMP blade agent models in the case of fault (loss of contact) alarms. If contact is lost for each of these domain entities (in other words, the chassis, the blades and the SNMP blade agents) a single Chassis Down alarm is generated on the chassis. The entire set of Contact Lost alarms is correlated with it.

In the case of proxy lost alarms, chassis-level alarm correlation uses a smaller domain that includes a chassis and its blades. Here, Proxy Lost alarms for all blades are correlated with the chassis Proxy Lost alarm.

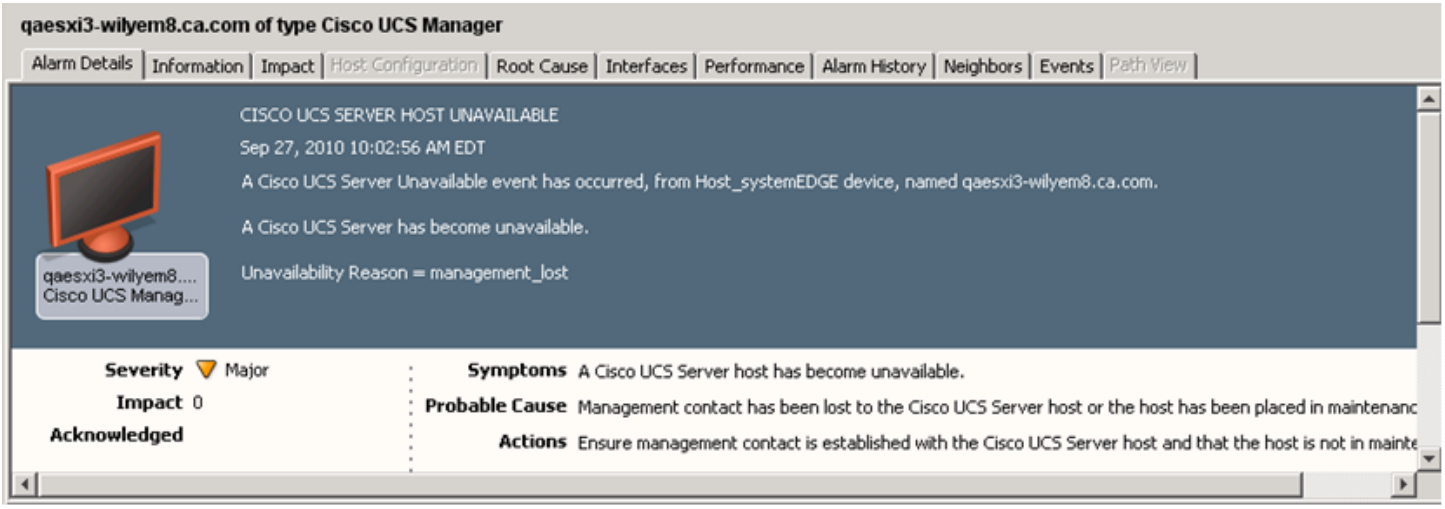
UCS System-Level alarm correlation uses a domain that includes the CA eHealth SystemEDGE host, the FIs, and the child chassis and blades. If communication is lost between DX NetOps Spectrum and the CA eHealth SystemEDGE host, a Proxy Lost alarm is present on all the FIs, chassis, and blades. A Proxy Unavailable alarm is present on the host.

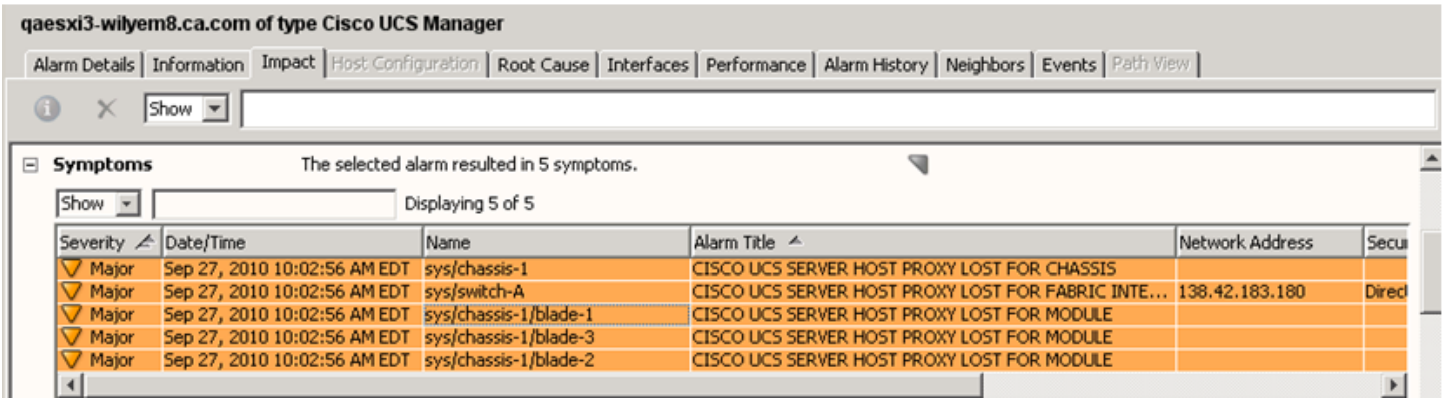
The Proxy Lost alarms for all the components are correlated with the host Proxy Unavailable alarm. These correlations are performed hierarchically. The Proxy Unavailable alarm is itself correlated with the alarm that indicates the reason for the communication failure. For example, it indicates contact lost, management lost, or maintenance mode. This top-level, overarching alarm is then visible to you in the alarm window.

Root Cause Isolation Examples

Root cause isolation resembles the following examples:

- A UCS chassis is inadvertently powered off, affecting the blades (and services running on them). Therefore, the individual Contact Lost alarms on all blades are correlated with the Chassis Down alarm to point the fault to the chassis.
- DX NetOps Spectrum loses contact with the CA eHealth SystemEDGE host. Therefore, the Proxy Lost alarms on all FIs, chassis, and blades are hierarchically correlated with the host’s Proxy Unavailable alarm.





Chassis and Blade Availability Alarms

Chassis availability alarms include Chassis Down and Blade Status Unknown (which is correlated with Chassis Down).

Blade availability alarms include Blade Removed and Blade Failed (the blade is present but has a failed status). Both of these alarms are correlated with an existing Blade Status Unknown alarm on the parent chassis. Note that blade models are subject by default to a two-hour age-out to allow for blade replacement.

Service Profile Alarms

Not only do we display all the service profile details, we also create DX NetOps Spectrum models for each service profile. DX NetOps Spectrum actively monitors the state of the service profile and generates events and alarms based on the operational state of each service profile.

Trap-Generated Alarms

UCS supports trap-generated alarms that indicate infrastructure and environment issues. Discrimination is used where appropriate. Examples include Blade Added, Blade Removed, Power Supply Inoperable, and Temperature Warning.

Dedicated UCS Views

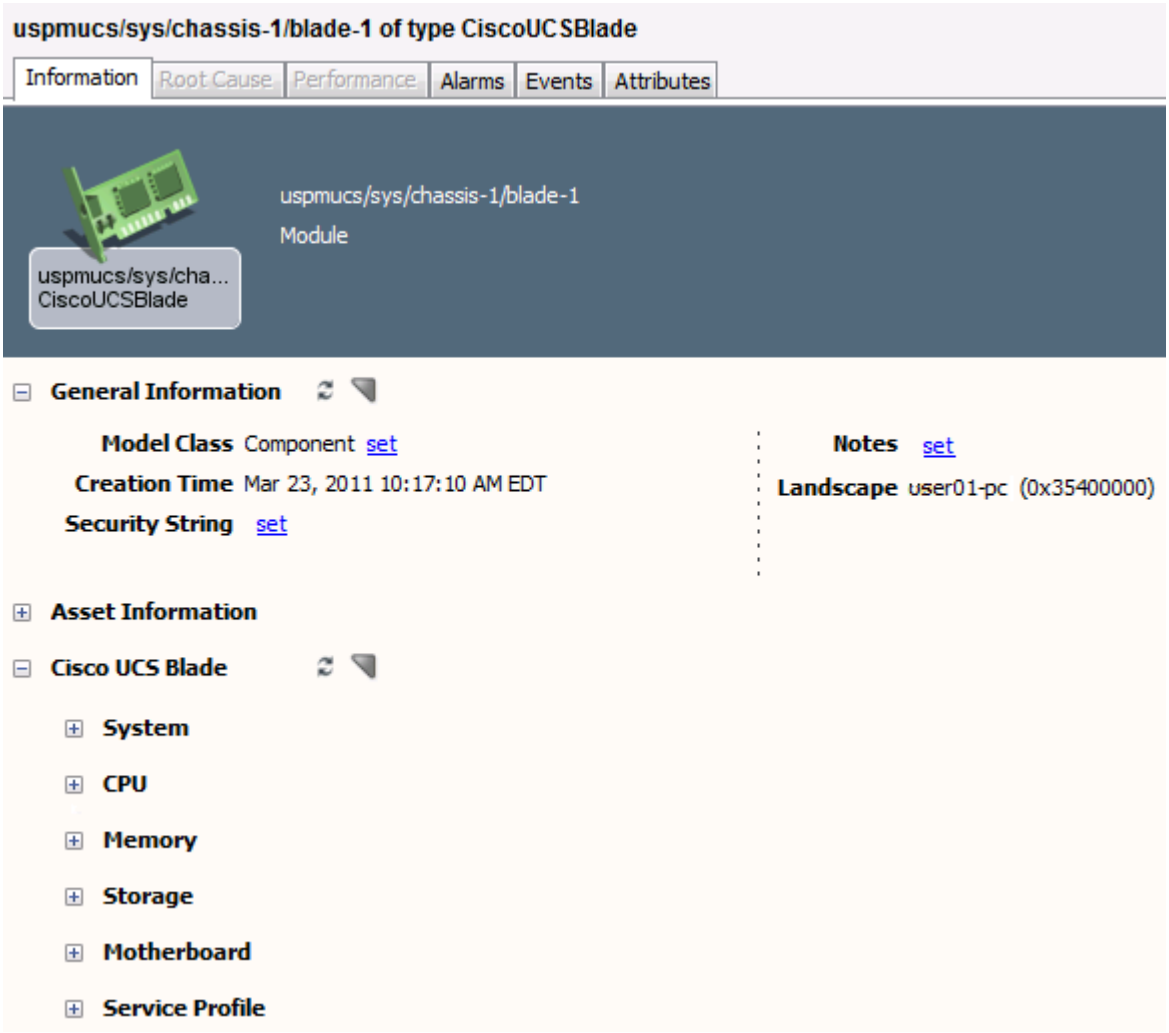
Dedicated UCS views are available for the following device types (indicated in parentheses):

- CA eHealth SystemEDGE host (Cisco UCS Manager)

This view includes table views of the managed environment.

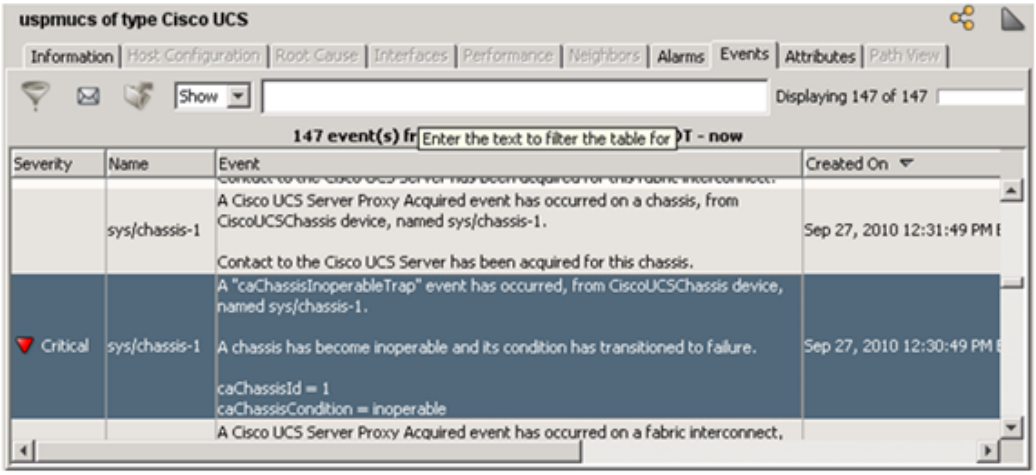
- Fabric Interconnect (Cisco UCS Switch)
- Chassis (Cisco UCS Chassis)
- Blade (Cisco UCS Blade)
- Service Profile (N/A)

OneClick views show hardware details, such as memory DIMMs, mezzanine cards, fabric interconnect extenders, and interfaces.



Intelligent Trap Forwarding

All UCS traps are generated from the UCS AIM and thus arrive into DX NetOps Spectrum from the CA eHealth SystemEDGE host. Therefore, DX NetOps Spectrum employs a forwarding mechanism to generate the event or trap on the correct UCS component. Determination of the correct component is achieved through examination of trap variable values. If the applicable component cannot be found, the trap event is asserted on the CA eHealth SystemEDGE host.



Chassis Management

UCS leverages the rich set of chassis management features that are available in DX NetOps Spectrum:

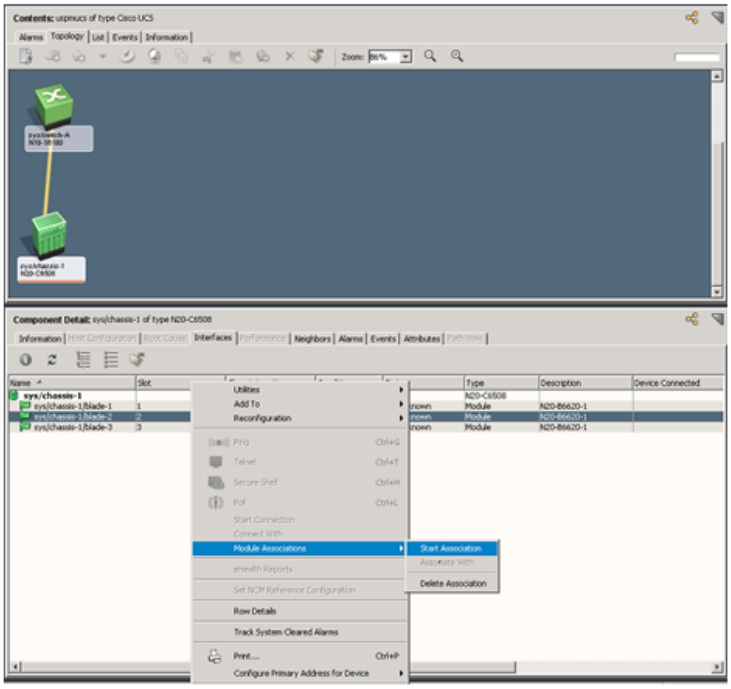
- Manual blade/SNMP device association
- Blade and managed device visibility
- Locater searches

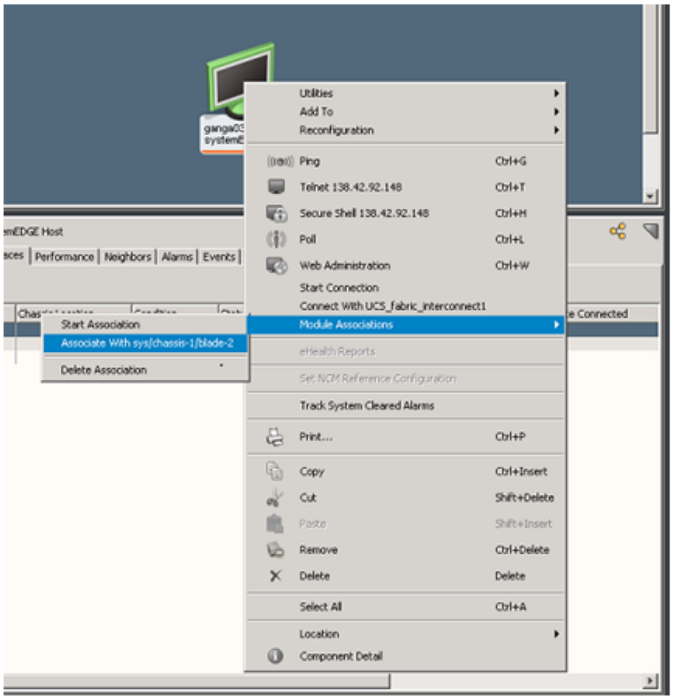
For more information, see the Support for Chassis Devices section of Certifications.

Manual Blade/SNMP Device Association

Manual blade/SNMP device association binds one blade of a chassis to an SNMP-capable blade agent model. This association enables quick determination of the system/chassis location from the agent model. The SNMP model is not moved into the UCS container, but it takes the place of the blade in the Interfaces tab of the chassis.

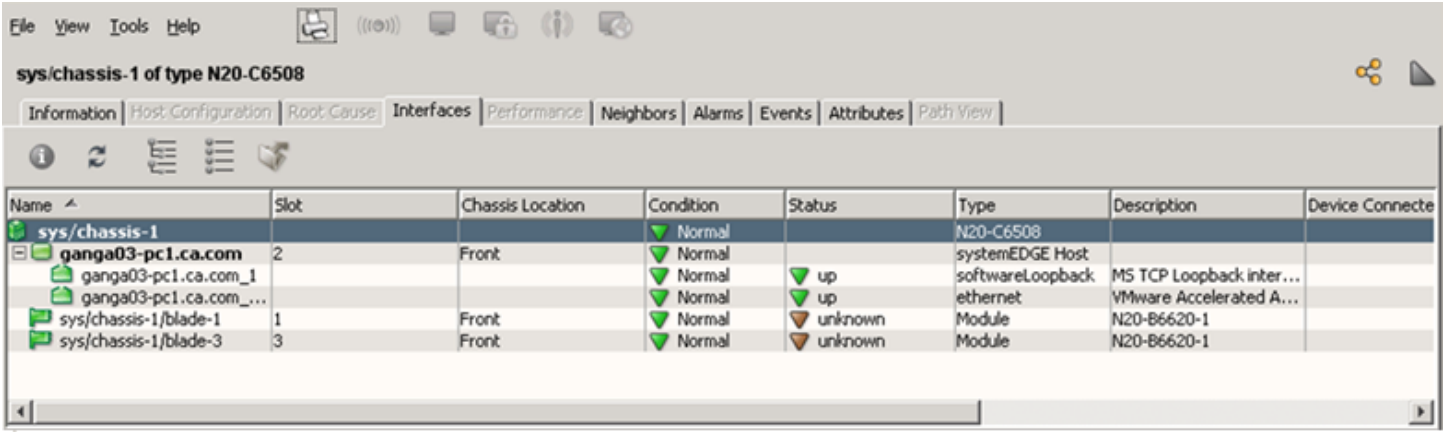
To bolster this integration, the SNMP agent model is incorporated into chassis fault correlation. Blade/agent association also enables SNMP model identification through chassis-based Locater searches.





Blade and Managed Device Visibility

Through the inclusion of associated SNMP devices, the extended Interfaces tab offers visibility into the blades of the chassis and managed devices.



Locater Searches

Chassis-based searches are listed under the Chassis node on the Locater tab. These searches facilitate the quick location of chassis and their components.

Searches include:

- All Chassis
- All Chassis Managed Devices
- All Modules
- Managed Devices By Chassis Name
- Modules By Chassis Name

Control Cisco UCS AIM Polling

When troubleshooting networking issues or tuning the Cisco UCS Manager performance, change the Cisco UCS AIM (cacucsaimApp) polling rate to increase or decrease the frequency. You can configure the polling rate by setting the Poll_Interval attribute on the Cisco UCS AIM application model.

Follow these steps:

1. Open OneClick and select Locater in the Navigation pane.
2. Expand Application Models, and double-click 'By Device IP Address'.
   The Search dialog opens.
3. Enter the IP address of your Cisco UCS Manager in the Device IP Address field, and click OK.
   A list of application models for the Cisco UCS Manager appears in the Contents panel.
4. Select the cacucsaimApp application model.
   The application model details appear in the Component Detail panel.
5. Select Information in the Component Details pane.
6. Expand DX NetOps Spectrum Modeling Information.
7. Click 'set' in the Poll Interval (sec) field, enter a new value, and press Enter.
   Note: Changing the Polling Interval value from any number to 0 also sets the Polling field to Off, disabling UCS AIM polling. If you set the Polling Interval to 0 and the Polling field to On, UCS AIM polling continues, using the polling interval that is set for the Cisco UCS Manager device.
   The Cisco UCS AIM polling rate is configured.

Cisco Catalyst

DX NetOps Spectrum supports Catalyst device families 1200, 1400, 1900, 2820, 3000, 3200, 4000, 4500, 5000, and 6500 with multiple enhanced certifications.

For the Catalyst 2900 and Catalyst 3500 device families, the specific enhanced certification is dependent on the supported MIB set.

DX NetOps Spectrum models Catalyst 2900 series devices as follows:

- The HubCat29xx model type models Catalyst 2900 series switches that run the IOS firmware and support the CISCO-C2900-MIB.
- The SwCiscoIOS model type models Catalyst 2900 series switches that run the IOS firmware, but does not support the CISCO-C2900-MIB. Catalyst 2970 and Catalyst 2948g devices fall into this category.
- The SwCat4xxx model type models Catalyst 2900 series switches that run the CatOS firmware.

DX NetOps Spectrum models Catalyst 3500 series devices as follows:

- The HubCat29xx model type models Catalyst 3500 series switches that run the IOS firmware and support the CISCO-C2900-MIB.
- The SwCiscoIOS model type models Catalyst 3500 series switches that run the IOS firmware, but does not support the CISCO-C2900-MIB. Catalyst 3550 series falls into this category.

Cisco Catalyst Board Fault Isolation Overview

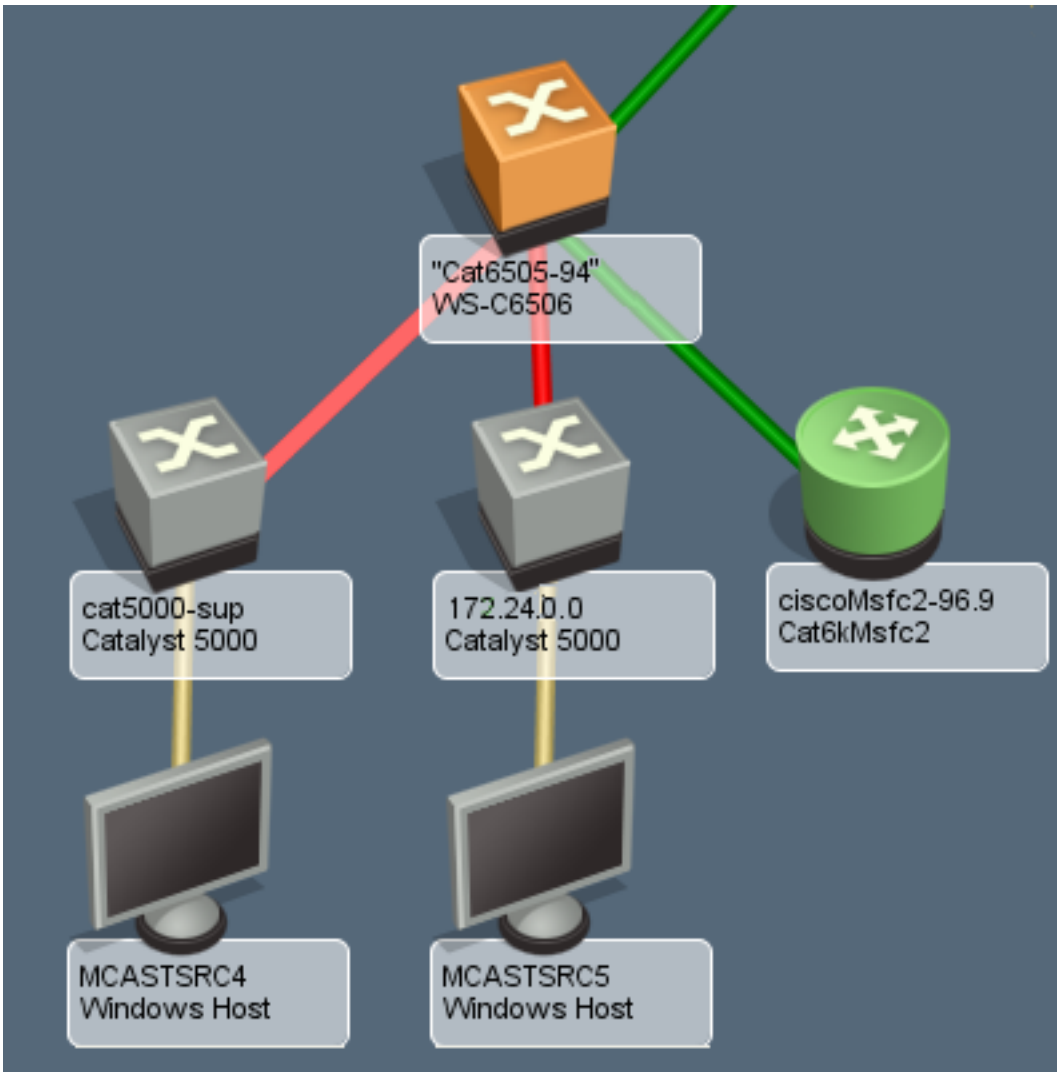
DX NetOps Spectrum supports boards being pulled or failing.

In a traditional fault isolation scenario, when a board in a chassis-based device fails, DX NetOps Spectrum generates critical alarms on all downstream device models. The device model that has the failed board retains its normal condition. However, this behavior does not give an indication as to which device is actually the fault.

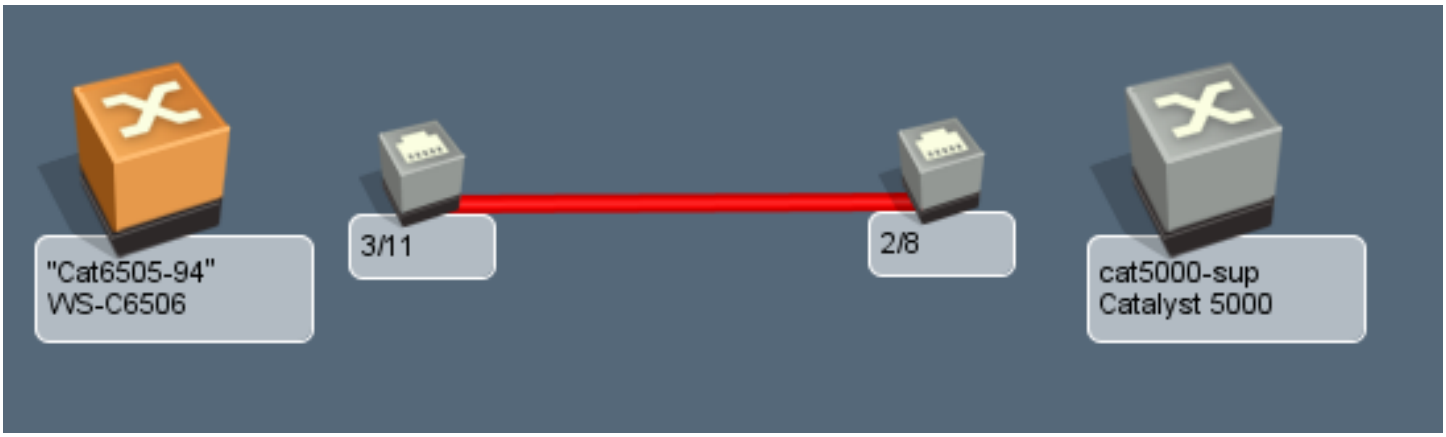
In the same scenario, Catalyst chassis-based devices that support the CISCO-STACK-MIB have enhanced fault isolation functionality to suppress the downstream device models, generate a major alarm on the device model which has the failed board, and generates a critical alarm on the board model. The ports that are associated with the board model are also suppressed giving a clear indication to not only which device is at fault, but also the board that is at fault.

The Catalyst Device with Downstream Devices Example

In the following example, the connected devices have Enable Live Links set to TRUE. When the Catalyst board is pulled, the devices that are connected to ports through that board go down. This event triggers DX NetOps Spectrum to determine the cause of the fault. In this example, two downstream switches and hosts are affected.

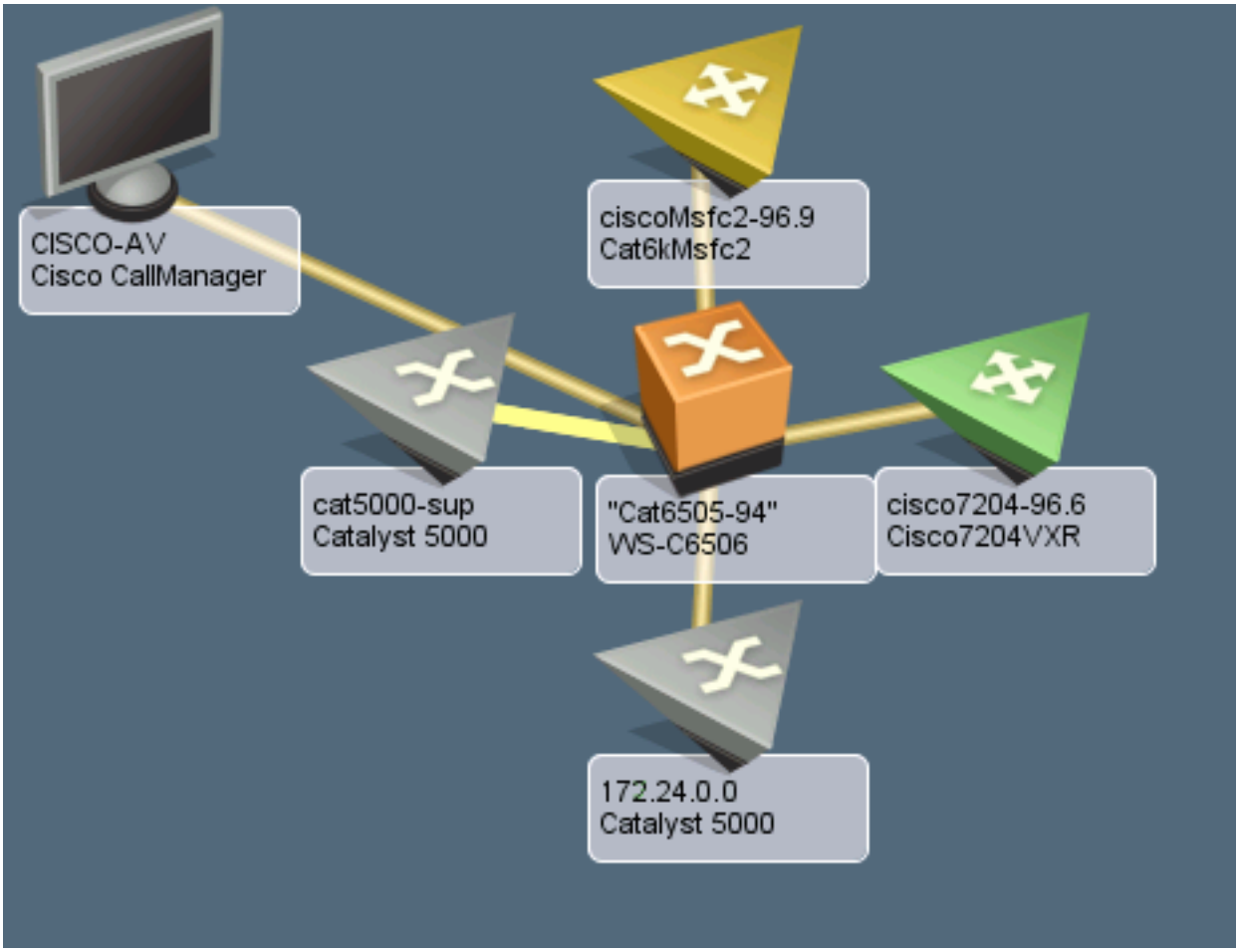


The following illustration shows the Link Information view. The Link Information view shows the root cause of the alarm.

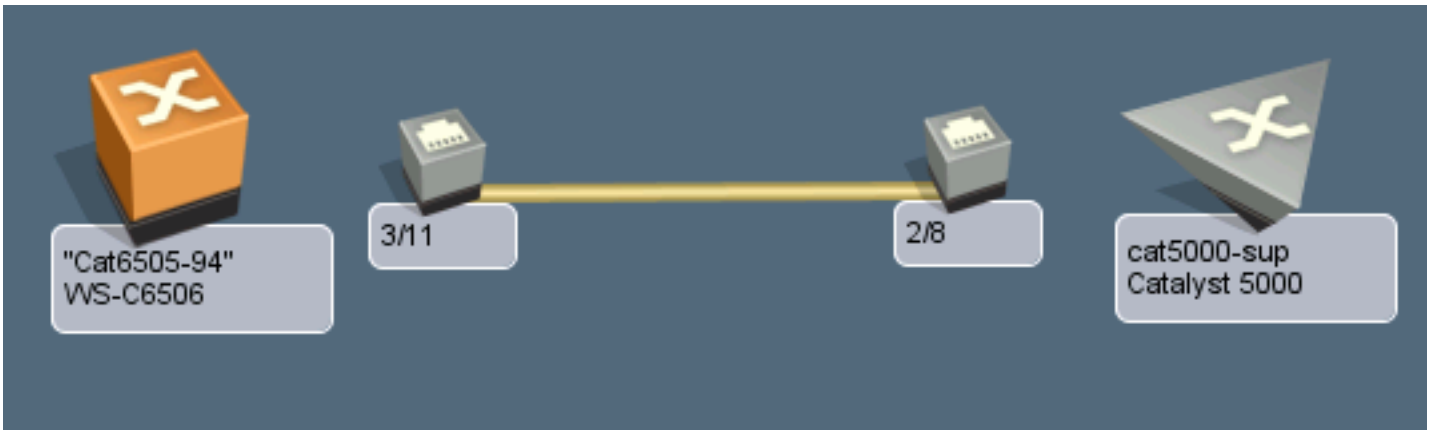


Example of a Catalyst Device with Downstream Devices

In the following example, the connected devices have Enable Live Links set to FALSE. Traps are received when the Catalyst board is pulled, and the devices that are connected to ports through that board go down. This event triggers DX NetOps Spectrum to determine the cause of the fault. In this example, two downstream switches (off-page references) and hosts (not in view) are affected.

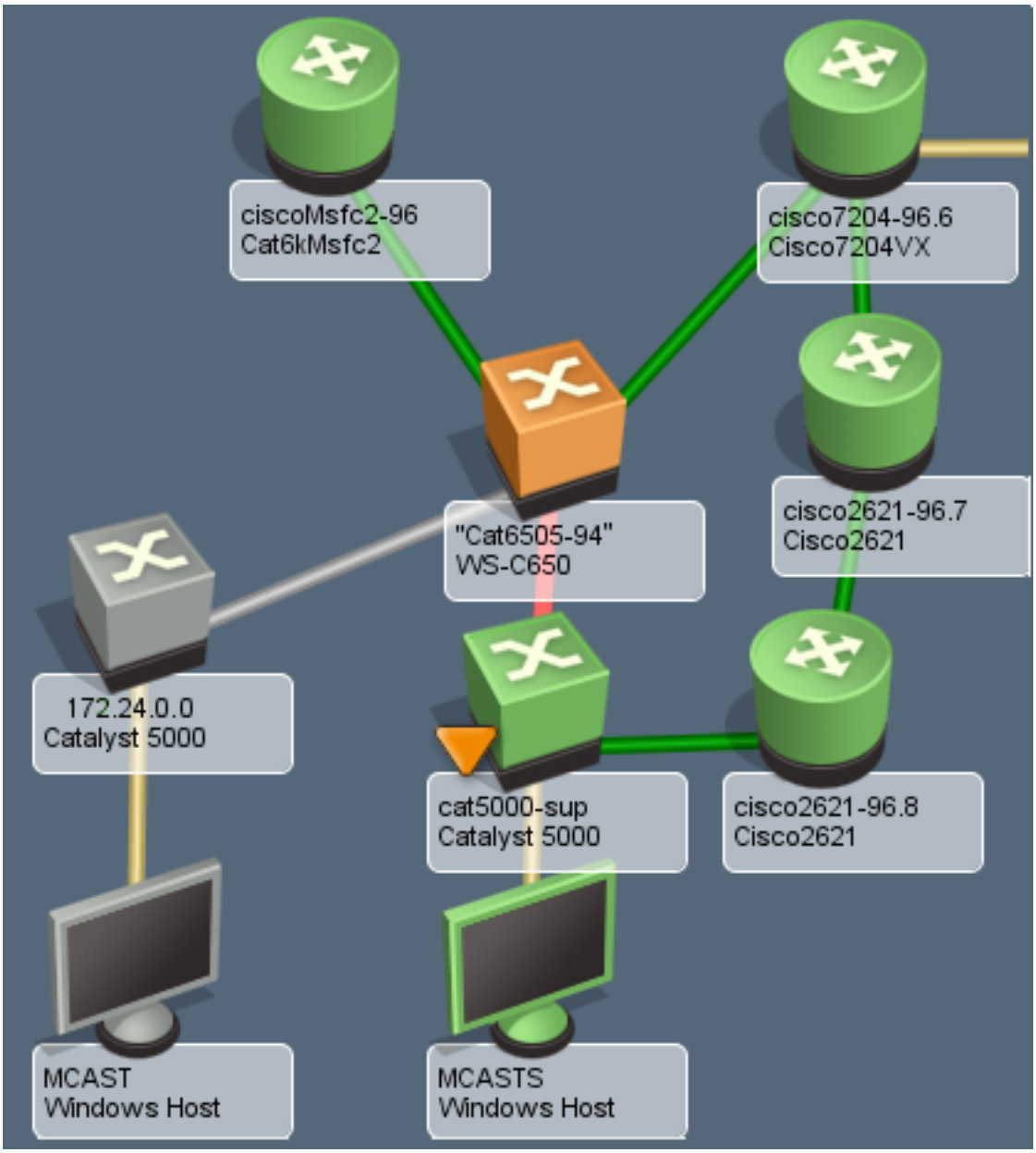


The following illustration shows the Link Information view. The Link Information view shows the root cause of the alarm.



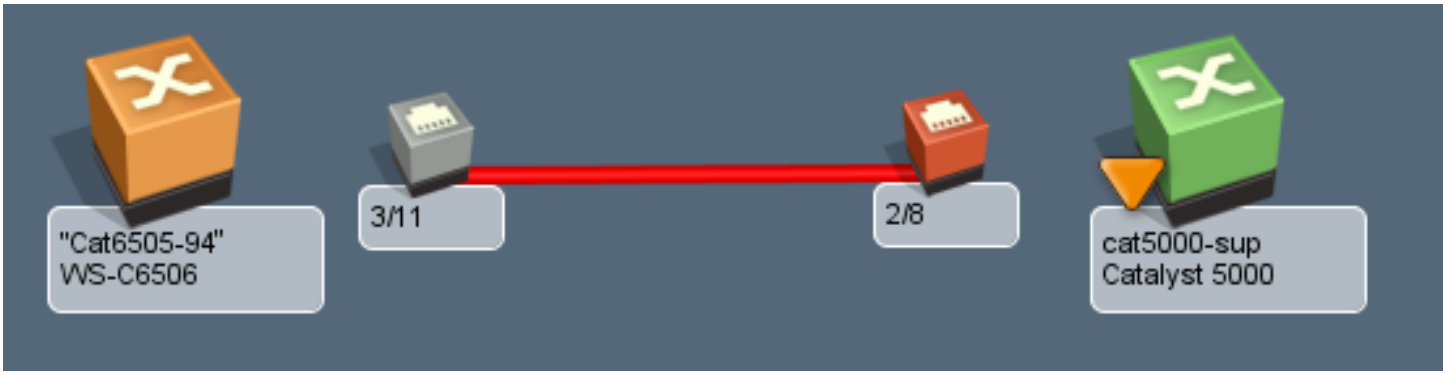
The Catalyst with Downstream Devices with Multiple Management Paths Example

In the following example, the connected devices have Enable Live Links set to TRUE. When the Catalyst board is pulled, the devices that are connected to ports through that board go down. This event triggers DX NetOps Spectrum to determine the cause of the fault. In this example, one downstream switch and host are affected.



NOTE
The switch with the second management path stays contacted and alarms its port.

The following illustration shows the Link Information view. The Link Information view shows the root cause of the alarm.



Cisco Technology Support

This section discusses how DX NetOps Spectrum supports the following Cisco technologies:

- Router Redundancy
- SNMPv3 Device Discovery
- Syslog Trap Support
- Tunnel Interface Modeling
- VLAN Indexing Support
- Virtual PortChannel Support
- Virtual Switching System (VSS) Support

Router Redundancy

The CISCO-HSRP-MIB lets you manage Cisco IOS proprietary Hot Standby Router Protocol (HSRP).

HSRP lets the host appear to be using a single router and maintain the connectivity even if the actual first hop router fails. Multiple routers participate in this protocol. Together, they simulate a single virtual router with a static IP address that is known as the virtual IP address. The end hosts forward their packets to the virtual router.

The router that forwards packets is known as the active router. If the active router fails, the standby router replaces the active router. HSRP provides a mechanism for determining the active and standby routers, using the IP addresses on the participating routers. If an active router fails, the standby router takes over without a major interruption in the connectivity of the host.

HSRP Group Modeling

DX NetOps Spectrum creates models for every Hot Standby Router Protocol (HSRP) group that it discovers. DX NetOps Spectrum identifies them by the virtual IP address. This virtual IP address is added to the Redundancy Excluded Address of the active router of the HSRP group. Each HSRP group model knows the active and standby routers in the HSRP group.

OneClick gives the visibility in the HSRP group membership, using the following router redundancy spotlight methods:

- **Explorer Search**
  Provides a view that highlights HSRP group members with Active and Standby labels, as applicable. You can select a container in the Explorer tab, select the Topology tab in the Contents panel, click the spotlight icon, and select Router Redundancy.
- **Locater Search**

Displays the available searches for HSRP group models. You can open the Router Redundancy directory in the Locater tab. For each model, the Contents panel contains information about the HSRP group model, including Virtual IP, Group ID, and group membership.

HSRP Group Membership

DX NetOps Spectrum monitors each Hot Standby Router Protocol (HSRP) group, looking for state and membership changes. DX NetOps Spectrum polls the HSRP group tables of the active router using the polling interval of the active router device model. DX NetOps Spectrum also responds to the state change traps that the device sends.

If a router fails over, a major alarm is asserted on the HSRP group model, indicating that Router Redundancy has been lost and a standby router is no longer available. DX NetOps Spectrum clears this alarm when a new standby router is detected.

Note: The Information tab for the group model provides a Report Election Change setting. If you enable this setting, DX NetOps Spectrum generates an alarm every time a new active router is selected. DX NetOps Spectrum does not clear this alarm.

Change the State of the HSRPMode Attribute

Limit the volume of SNMP requests to the network devices that are running with the HSRP deployment to prevent a degradation of network performance. You can set the state of the HSRPMode attribute to one of the following three states:

- **Off**
  The HSRP table is not polled.
- **Passive**
  The HSRP table is polled once at the activation. Otherwise, DX NetOps Spectrum relies on updates from traps to update this information.
- **Active**
  The HSRP table is polled every poll interval in addition to the passive processing.

Follow these steps:

1. From the Locater tab, expand Application Models.
2. Select By Name.
   The Search dialog opens.
3. In the Search dialog, type 'CiscoHSRPApp' in the Model Type name text box.
   A list of all of the CiscoHSRPApp devices is displayed.
4. Select all of the devices in the list and right-click to select Utilities, Attribute Editor.
   The Attribute Editor dialog opens.
5. In the left pane, expand User Defined and click the add hyperlink.
   The Attribute Selector dialog opens.
6. Type 'HSRPMode′ in the filter tex box and click OK.
   The attribute HSRPMode is added under User Defined.
7. Select HSRPMode, and click the right arrow to move it to the right pane.

   NOTE
   You can now set the state of the HSRPMode attribute in the right pane.
8. In the left pane, expand SNMP Communication to select *Poll Interval (sec),* and click the right arrow to move it to the right pane.
   You can now set a value for the Poll Interval in the right pane.
9. In the right pane, clear No Change, and set a value for the Poll Interval and set the state of the HSRPMode to Off, Passive, or Active.

You have changed the state of the HSRPMode and set the value for the Poll Interval on all the device models in your landscape.

SNMPv3 Device Discovery

When you discover SNMPv3 devices on the Cisco switches with VLANs, you cannot use the community_string@VLAN_ID format to an index bridging information for each VLAN. Create the contexts instead.

The following configurations are to be done on the Cisco devices. For detailed instructions, refer to the respective Cisco switch configuration guides.

For DX NetOps Spectrum to read the bridging information, create these contexts using the following format:

```
vlan-<VLAN_ID>
```

Example: Create an SNMP v3 User

This example creates an SNMPv3 user context, using the format DX NetOps Spectrum can read:

```
(enable) set snmp user <level1-vlan> nonvolatile

(OUTPUT) Snmp user was set to level1-vlan authProt no-auth  privProt no-priv
```

Example: Create an SNMP Group

This example creates an SNMP group context, using the format DX NetOps Spectrum can read:

```
(enable) set snmp group <v3-level1-vlan> user <level1-vlan> security-model v3 nonvolatile

(OUTPUT) Snmp group was set to v3-level1-vlan user level1-vlan and version v3, nonvolatile.
```

Example: Create an SNMP Access Group

This example creates an SNMP access group context, using the format DX NetOps Spectrum can read:

```
(enable) set snmp access <v3-level1-vlan> security-model v3 noauthentication read <defaultUserView> write
 <defaultUserView> notify <defaultUserView> nonvolatile

(OUTPUT) Snmp access group was set to v3-level1-vlan version v3 level noauthentication, readview
 defaultUserView, writeview defaultUserView, notifyview defaultUserView context match: exact, nonvolatile.

(enable) set snmp access <v3-level1-vlan> security-model v3 noauthentication read <defaultUserView> write
 <defaultUserView> notify <defaultUserView> context <vlan> prefix nonvolatile

(OUTPUT) Snmp access group was set to v3-level1-vlan version v3 level noauthentication, readview
 defaultUserView, writeview defaultUserView, notifyview defaultUserView context: vlan, context match: prefix,
 nonvolatile.
```

Syslog Trap Support

The System Message Log (syslog) protocol lets you send text messages from the Cisco devices to the network management software. The text messages are sent to the DX NetOps Spectrum Event Manager as SNMP traps. Syslog trap support lets the router device identify the text messages and escalate them to alarms as required. Syslog trap support also lets the Cisco Router model icon communicate alarm severity information.

If an alarm occurs as indicated by the Cisco device icon, the DX NetOps Spectrum Alarm Severity and a syslog message appear in the Alarm Log.

The syslog messages are classified based on the severity that ranges from 0 to 7 (most severe to least severe). The alarms display in the Alarm Log. Because these alarms are associated with Cisco device models, the corresponding model icon changes color and flashes, depending on the alarm severity.

The following table lists the severity codes and their descriptions:

Severity	Description
0	Emergency -- System is unusable
1	Alert -- Immediate action required
2	Critical -- Critical condition
3	Error -- Error condition
4	Warning -- Warning condition
5	Notification -- Normal but significant condition
6	Informational -- Informational message only
7	Debugging -- Message that appears during debugging only

The following table maps syslog message severity to the DX NetOps Spectrum alarm severity:

Alarm Severity	Color
0-1	Red
2-3	Orange
4	Yellow

Messages with an alarm severity of 5 through 7 do not generate an alarm because they are of a minor importance. Facility(hardware device, protocol, or a module or system software) lists the messages.

A facility code is an abbreviation of the facility to which the message refers. A facility can be a specific hardware device, a protocol, or software. Within each facility, messages are listed in terms of the severity, from the highest (0) to the lowest (7). A *mnemonic* is an uppercase string that uniquely identifies the message.

An explanation and a recommended action follow each message. Messages appear only when the system remains operational. The following line is an example of a syslog message:

01/01/2001,18:31:15:SYS-5-MOD_INSERT:Module 5 has been inserted.

This message is interpreted as follows:

- 01/01/2001,18:31:15 is the date and time of the error (this information appears if set for system log messaging).
- SYS is the facility type.
- 5 is the severity level, indicating it is a normal but significant condition.
- MOD_INSERT is the mnemonic that uniquely identifies the message.
- "Module 5 has been inserted" is the message text that describes the condition.

The System Message Log (syslog) program saves the system messages in a log file or directs the messages to other devices. Syslog software lets you do the following functions:

- Save logging information for monitoring and troubleshooting
- Select the type and destination of the logging information

By default, the switch logs normal but significant system messages to its internal buffer and sends these messages to the system console. You can specify how system messages must be saved based on the type of facility and the severity level. Messages can be time-stamped to improve real-time debugging and management.

Add Syslog Trap Mappings to DX NetOps Spectrum

DX NetOps Spectrum includes three text files that SpectroSERVER uses to map Cisco syslog traps to DX NetOps Spectrum events.

The following table shows the syslog text files:

Device Syslog Message	Text File
Cisco Router	<$SPECROOT>/SS/CsVendor/Cisco_Router/Rtr.txt
Catalyst Switch	<$SPECROOT>/SS/CsVendor/Ctron_CAT/Switch.txt
Cisco PIX	<$SPECROOT>/SS/CsVendor/CiscoPIX/Pix.txt

Each line of these text files contains information to map syslog messages to DX NetOps Spectrum events. The lines have the following format (for each field, a single space is the delimiter):

```
<facility> <severity> <mnemonic> <event code>
```

Follow these steps:

1. Add a line to the file that contains the previous information.
   For example, to add support for the %SPE-3-SM_DOWNLOAD_FAILED syslog message for Cisco Routers, add the following line to the Rtr.txt file: SPE 3 SM_DOWNLOAD_FAILED 0xffff0001, where 0xffff0001 is an arbitrary code that you select.
2. Create Event Format and the Probable Cause files for the event and alarm.
   In this case, create Eventffff0001 and Probffff0001. You can enter any text in these files. The following variable data can be read from the Event Message and displayed in the Event Format file:
   ```
   {S 1}- Facility
   {T T1_210017 2}- Severity
   {S 3}- Mnemonic
   {S 4} - Message
   ```
3. Add the event-to-alarm mapping. Using the previous example, add the following line:
   0xffff0001 E 50 A 2,0xffff0001

   NOTE
   You must have an EventDisp file in the same directory as the Rtr.txt file.

   An orange alarm is generated if SpectroSERVER receives this syslog trap.

   NOTE
   You can do this configuration while the SpectroSERVER is running. The SpectroSERVER checks for changes to the *.txt files every minute.

Syslog Message Filter

The Cisco Syslog Message Filter OneClick view lets you filter unwanted syslog messages. Filtering syslog messages blocks unwanted alarms or events. SS/CsVendor/SYSLOG contains eight files that correspond to different filter categories. To select the filter category to which a facility belongs, move the facility to the required SS/CsVendor/SYSLOG file.

The following table shows SS/CsVendor/SYSLOG files and corresponding filters:

File	Corresponding Filter
Syslog0	Protocol_Filter
Syslog1	System_Filter
Syslog2	Environment_Filter

Syslog3	Software_Filter
Syslog4	Security_Filter
Syslog5	Hardware_Configuration_Filter
Syslog6	Connection_Configuration_Filter
Syslog7	PIX_Firewall_Filter

NOTE
The facilities are interchangeable with any of the filters.

The filters are as follows:

- **Protocol_Filter**
  Affects the Syslog0 file. Set this filter to True to filter all syslog messages whose facilities deal with protocols. For example, BGP, OSPF, SNMP, SPANTREE.
- **System_Filter**
  Affects the Syslog1 file. Set this filter to True to filter all syslog messages whose facilities deal with the system. For example, CBUS, MEMSCAN.
- **Environment_Filter**
  Affects the contents of the Syslog2 file. Set this filter to True to filter out all syslog messages that deal with environment variables. For example, LCFE, LCGE.
- **Software_Filter**
  Affects the contents of the Syslog3 file. Set this filter to True to filter out all syslog messages that deal with internal software. For example, PARSER, RSP, GRPGE.
- **Security_Filter**
  Affects the contents of the Syslog4 file. Set this filter to True to filter out all syslog messages that deal with the security of the system. For example, RADIUS, SECURITY.
- **Hardware_Configuration_Filter**
  Affects the contents of the Syslog5 file. Set this filter to True to filter out all syslog messages that deal with the hardware configuration of the device. For example, IOCARD, MODEM, DIALSHELF.
- **Connection_Configuration_Filter**
  Affects the contents of the Syslog6 file. Set this filter to True to filter out all syslog messages that deal with connection configuration of the device. For example, MROUTE, ISDN, X25.
- **Pix_Firewall_Filter**
  Affects the contents of the Syslog7 file. Set this filter to True to filter out all syslog messages that deal with the Cisco PIX Firewall device.

Tunnel Interface Modeling

DX NetOps Spectrum supports the Cisco IPSec tunnel interface management for the Cisco devices that support CISCO-IPSEC-FLOW-MONITOR-MIB and CISCO-IPSEC-MIB. These MIBs are available for Cisco firmware versions 12.1 (4) or later.

DX NetOps Spectrum supports the following IPSEC VPN management features:

- The modeling of tunnel interfaces (site-to-site)
- The automatic connectivity mapping
- The interface model identification
- The interface model aging
- Link down trap correlation
- Status monitoring of tunnel interfaces

The following attributes control IPSEC VPN management:

- CreateTunnelLIf
- Interface_Polling_Interval

Configure CreateTunneLIf

The CreateTunnelLIf attribute indicates if tunnel interface models are created for each IPSec tunnel that is defined on the device. If TRUE, it specifies that DX NetOps Spectrum reads the external tables during the interface reconfiguration. These external tables define the tunnel interfaces present. DX NetOps Spectrum creates appropriate tunnel interface models as a subinterface of the related physical interface.

Follow these steps:

1. Navigate to the Locater tab, expand the Application Models folder, and double-click By Device IP Address. The Search dialog opens.
2. Enter the IP address of the Cisco IPSec-capable device you want to configure and click OK. The device appears in the Contents panel.
3. Select the CiscIPSecExtAp device in the Contents panel.
4. Select the Attributes tab in the Component Detail panel.
5. Select CreateTunneLIf in the left pane and click the right arrow button to move it to the right pane.
6. Double-click CreateTunneLIf in the right pane to change its value.

NOTE
Setting CreateTunneLIf to No disables Cisco IPSec tunnel modeling.

Configure Interface_Polling_Interval

The Interface_Polling_Interval attribute defines the tunnel table polling interval in seconds. If set to 0, the table is not polled.

Follow these steps:

1. Navigate to the Locater tab, expand the Application Models folder, and double-click By Device IP Address. The Search dialog opens.
2. Enter the IP address of the Cisco IPSec-capable device you want to configure and click OK. The device appears in the Contents panel.
3. Select the device in the Contents panel.
4. Select the Attributes tab in the Component Detail panel.
5. Select Interface_Polling_Interval in the left pane and click the right arrow button to move it to the right pane.
6. Double-click Interface_Polling_Interval in the right pane to change its value.

VLAN Indexing Support

DX NetOps Spectrum can test whether the VLAN indexing community string is supported on a particular Cisco device. The VLAN indexing community string prevents authentication failure traps.

If a Cisco device supports the VLAN indexing community string, the VLANIndexingSupported (0x4a0037) attribute value is set to Supported 1.

If a device of Cisco does not support the VLAN indexing community string, the VLANIndexingSupported (0x4a0037) attribute value is set to an enumeration NotSupported 0. Further VLAN index reads are not made. This configuration prevents authentication failure traps from being generated.

Test the device, if a Cisco device was not tested due to lack of VLANS information for the device. Perform a Discovery on that device, or to enable the VLAN overlay, set the VLANIndexingSupported (0x4a0037) attribute value to Test 2.

If the configuration of a device changes to support the VLAN indexing community string, change the attribute value to VLANIndexingSupported (0x4a0037) on the Transparnt_App model for that device through the Attribute Editor.

Virtual PortChannel Support

Introduction to Virtual PortChannel Technology

Virtual PortChannels (vPCs) allow links that are physically connected to two different Cisco switches to appear to a third downstream device as coming from a single device and as part of a single port channel. The third device can be a switch, a server, or any other networking device that supports IEEE 802.3ad PortChannels. For more information about vPC, see the Cisco documentation.

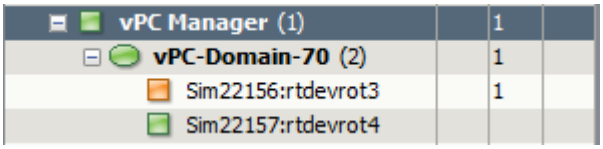
Identification of vPC Configuration

DX NetOps Spectrum monitors vPC domains formed by vPC-enabled Cisco Nexus 7000 series switches with firmware version 6.2.2 that support the CISCO-VPC-MIB. The following vPC parameters are monitored for a particular vPC domain:

- Peer link status
- Peer-keepalive status
- vPC role status
- vPC dual-active status
- vPC PortChannels configured

The monitoring of vPC domains is enabled by creating vPC domain container models (vPC domain model) under vPC Manager node in the Navigation pane of OneClick. Each vPC domain model contains two peer switches that form that vPC domain. All vPC specific alarms and information are displayed at the vPC domain level in OneClick.

The following image shows how a vPC domain model containing two peer switches is arranged in OneClick:



NOTE

As shown in the image, **vPC-Domain-70** is created which contains two peer switches, the value 70 indicates the domain id. The peer switches shown here are simulated, and not real-time switches. If you want to give other meaningful name to a vPC domain model, you can edit its name.

vPC Events, Alarms, and Correlation

The following table lists the events that are generated on a vPC domain:

Events Code	Event Description	Alarm Raised or Cleared
0x00210def	VPC DOMAIN HOST LINK STATUS CHANGE EVENT	Major alarm is raised.
0x00210df1	VPC HOST LINK CLEAR EVENT	Major alarm is cleared.
0x00210df2	VPC DOMAIN ROLE STATUS CHANGE EVENT	Alarm is not raised.
0x00210df3	VPC DOMAIN KEEP ALIVE STATUS CHANGE EVENT	Major alarm is raised.

0x00210df4	VPC PEER KEEP ALIVE CLEAR EVENT	Major alarm is cleared.
0x00210dfa	VPC DOMAIN DUAL ACTIVE STATUS CHANGE EVENT	Major alarm is raised.
0x00210dfb	VPC DOMAIN DUAL ACTIVE CLEAR EVENT	Major alarm is cleared.
0x00210df9	DETECTED ANOTHER DEVICE EVENT	Minor alarm is cleared.
0x00210dfd	PEER DEVICE NOT RESPONDING TO POLLS EVENT	Critical alarm is raised.
0x00210dfe	PEERDEVICE_RESPONDING TO POLLS EVENT	Critical alarm is cleared.

When monitoring your vPC environment, vPC domain models display an alarm state for the following conditions:

Peer Link is Down

The attribute "cVpcStatusPeerLinkStatus" is polled to monitor this condition. A major alarm is raised when the value of this attribute is "down". The alarm is cleared when the value is "up".

Peer-keepalive Link is Disabled or PeerUnreachable or suspendedasdestipunreachable or misconfigured

The attribute "cVpcPeerKeepAliveStatus" is polled to monitor this condition. A major alarm is raised for this condition. The alarm is cleared when the value of this attribute is "Alive".

Dual Active Detected

The attribute "cVpcDualActiveDetectionStatus" is polled to monitor this condition. A critical alarm is raised if the value of this attribute is "True". In this case, the two alarms that are mentioned before are made symptoms to this critical alarm. The alarm is cleared when the value changes to "False".

Third Device found with the same vPC Domain ID

A minor alarm is raised when a third device is found the same domain id.

vPC Peer Switch is Down

This condition applies to the peer switch when it stops responding to polls. In this case, a critical alarm is raised.

> **NOTE**
>
> Non-vPC alarms (like NCM, SPM, so on) are not rolled to vPC domain container models. Alarms are raised only for the vPC conditions that are mentioned here.

vPC Views in OneClick

For each vPC domain model, the **Alarms**, **Events**, and **List** tab in the **Contents** pane displays alarms, events, and details of peer switches. The **Information** tab displays the following information under the **Virtual Port Channel (vPC)** view:

Configured Primary

This field displays the peer switch that is configured as primary.

Configured Secondary

This field displays the peer switch that is configured as secondary.

Current Primary

This field displays the peer switch that is operationally primary.

Host Link status

This field displays the following information for each peer switch:

Host-Link Interfaces Index

This field displays the interface index.

Host-Link Interface Name

This field displays the PortChannel configured on the corresponding interface index.

Host-Link Interface Status

This field displays the status of the interface index.

Spotlight View for vPC Domains

From the **Universe** topology, you can use the **Spotlight** view to directly view the vPC specific details of all vPC domains.

Follow these steps:

1. Select **Universe** container from the Navigation pane.
2. Click **Spotlight View** from Topology tab, and Select **vPC Domains**.
   The vPC Domains List dialog appears.
3. Select the required vPC domain from the list, and select **Hide Icons**.
   You can see the two vPC peer switches of that domain.

Locater Search for vPC Peer Models

You can find vPC peer switches using the **Locater** search from the Navigation pane. The **Locater** search includes the **vPC Manager** folder for this purpose. You can find vPC peer switches using the following two options:

All vPC peer Models using vPC Domain ID

Use this option to find by the Domain ID. If there are multiple vPC domains, click the **List** button to enter Domain IDs using the available delimiters. You can also import a text file that contains a list of Domain IDs (use newline, comma, space, or semi-colon as the delimiter).

All vPC peers

Use this option to find by Landscapes.

Troubleshooting Tips

If vPC domains and their vPC peer models are not created in the Oneclick, consider the following points:

1. Check whether the firmware version that is running on the vPC devices is below 6.2, and upgrade to leverage the vPC support. CISCO-VPC-MIB is supported only from firmaware version 6.2 and above.
2. After an upgrade, you must reconfigure the vPC enabled devices in OneClick using the reconfigure option.

Virtual Switching System (VSS) Support

Overview

VSS devices are support from the 10.0 release. Virtual Switching System (VSS) is a network system virtualization technology, it pools two Cisco Catalyst Switches into one virtual switch. VSS helps in increasing operational efficiency by boosting nonstop communications. VSS simplifies the network by reducing the number of network elements and hiding the complexity of managing redundant switches and links.

One virtual switch member chassis act as an Active virtual switch, while the other member is in Standby state. Even though one member is in Standby mode both chassis act as active and forward the traffic. When one of the virtual switch members fails, there is no convergence of protocols in the network and no disruption occurs to the traffic flowing through the VSS.

DX NetOps Spectrum supports the following Cisco VSS devices:

Device Type Name	System Object ID
Cat 65xx Virtual Switch	1.3.6.1.4.1.9.1.896
Cisco C68xx Virtual Switch	1.3.6.1.4.1.9.1.1934

NOTE

Contact CA Support for information on other Cisco Catalyst VSS device support.

Monitor VSS devices

Once the VSS device is modeled in DX NetOps Spectrum, Chassis container models are created for the VSS device in the Cisco folder under the Chassis Manager. DX NetOps Spectrum detects the role of Chassis container models as Active and Standby. The Active Chassis container model is shown in solid green () color and the Standby chassis container model is shown in light green () color. By expanding the chassis container models you can see the associated modules.

Use the following navigation to view the virtual switches under the Chassis Manager.

OC Navigation -> Chassis manager -> Cisco -> <VSS> ->Chassis containers (Active and Standby)

The Interfaces tab in the Component Detail pane shows the Chassis container models, associated modules and interfaces.

NOTE

After DX NetOps Spectrum upgrade, the VSS Chassis container models of the Cat 65xx Virtual Switch (System Object ID: 1.3.6.1.4.1.9.1.896), which is modeled in the earlier release are not displayed. To view these VSS Chassis container models you must model the Cat 65xx Virtual Switch again after upgrade to the current release.

Cisco Virtual Switch Information view

When you select the VSS device from Explorer tab, the Information tab under the Contents pane shows the Cisco Virtual Switch Information.

You can expand the Cisco Virtual Switch Information to see the following sub views:

- Chassis Information
- VSL Port Statistics
- VSL Statistics
- VSL Connection Information
- Core Switch Configuration

You can expand each sub view for detailed information.

VSS Chassis Information

This view is shown for VSS Chassis containers models. The Information tab under the Component Details pane shows the VSS Chassis Information along with its General Information and Asset Information.

You can expand the VSS Chassis Information to the following details:

- Chassis Switch ID
- Chassis Uptime
- Chassis Role
- Chassis CoreSwitchPriority
- Chassis CoreSwitchPreempt

Locater Search for VSS Models

You can find VSS devices using the **Locater** search from the Navigation pane. Under the Locater tab the Chassis folder contains the sub folder VSS. You can find VSS models using the following two options:

All Chassis Containers

Use this option to find the Chassis container models of VSS devices.

All VSS Devices

Use this option find all VSS Devices.

VSS Alarms and Correlation

When monitoring your VSS environment, VSS models display an alarm state for the following conditions:

VSS CONFIGURATION PROBLEM

It is a critical alarm. Spectrum raises this alarm when it detects a problem in VSS setup. The alarm is cleared when the Spectrum detects the setup is back to normal.

CHASSIS CONTACT LOST

It is a critical alarm. Spectrum raises this alarm when one of the VSS Chassis container model is down. The alarm is cleared when the VSS Chassis is up.

Correlation:

The CHASSIS CONTACT LOST alarm is correlated to VSS CONFIGURATION PROBLEM alarm.

VSS DUAL ACTIVE DETECTED

It is a critical alarm. Spectrum raises this alarm when a dual active detected trap “cvsDualActiveDetectionNotif” is sent by a VSS device. The alarm is automatically cleared when the VSS setup is back to normal.

Correlation:

The VSS DUAL ACTIVE DETECTED alarm is correlated to CHASSIS DOWN alarm.

The CHASSIS CONTACT LOST Critical Alarm and VSS CONFIGURATION PROBLEM critical Alarm is correlated to VSS DUAL ACTIVE DETECTED Critical Alarm.

VSL CONNECTION DOWN

It is a minor alarm. Spectrum raises this alarm when ‘cvsVSLConnectionChangeNotif’ trap is received with ‘cvsVSLConnectOperStatus’ as down. The alarm is cleared when the ‘cvsVSLConnectionChangeNotif’ trap is received with ‘cvsVSLConnectOperStatus’ as up.

Event id: 0x210df5

This event is generated when the VSS Chassis container model changes its role.

Cisco Nexus devices that support Virtual Device Context (VDC)

Overview

The Cisco Nexus devices, with NX-OS software which support Virtual Device Contexts (VDCs), allow the partitioning of a single physical Nexus device into multiple logical devices. This logical separation provides the following benefits:

- Independent Administrative and Management capability
- Change and failure domain isolation from other VDCs
- Isolation of Address, VLAN, VRF, and vPC

The virtual device context allows for the partitioning of a single physical device into more than one logical device. A Cisco Nexus device can support one Admin VDC and multiple non-default VDCs. Each VDC is a virtual entity that can be provisioned, configured and managed like a single physical chassis device.

You need to discover and model each VDC in DX NetOps Spectrum using their individual IP management address. For each device that supports VDC, a container is created and the VDCs within the device will be grouped within the same container in the Universe hierarchy. In 10.2, the CiscoVDC container contains all the VDCs that Spectrum discovers and models, specific to the device whose management IP you have used for discovery and modeling.

If the Cisco VDC Container is initially named after the non-admin/ default VDC context, the container name is updated to reflect the Admin VDC context, once the Admin VDC devices are identified.

From the 10.2.1 release, the container hierarchy creation is no longer created within the Universe hierarchy. A new Virtual Device Manager hierarchy is created in the Explorer view, where all the Cisco Nexus devices that support virtual device context are discovered, modeled and are named based on the Admin VDC context. The existing hierarchy functionality is carried forward from 10.2, while individual devices are still visible in the Universe hierarchy.
This change was made as creating the CiscoVDC container within the Universe hierarchy was blocking the overall topology view of the device along with the connectivity with the neighboring devices of the VDCs. As a result, router connectivity with particular VDC/ devices was not visible in the Universe view and neighboring connections were displayed to the CiscoVDC container rather than the actual VDC, until the user actually drilled down to the VDC container.

> **NOTE**
>
> Till 10.2 the container (CiscoVDC) containing the Cisco VDC devices was created within the Explorer view, from 10.2.1, the Virtual Device Manager container is created and all Cisco nexus devices which support VDC and have a virtual context are discovered and modeled under this Virtual Device Manager hierarchy.

Topology View

In DX NetOps Spectrum, each Cisco Nexus device which supports VDC, is represented by a container. The container will be named on the first modeled VDC. Once the Admin VDC is discovered, the container will be renamed to the Admin VDC context.

If you are upgrading to 10.2.1 from 10.2 and have discovered and modeled Cisco VDC devices with a virtual context in 10.2, after the upgrade is complete, the CiscoVDC container will be removed/ deleted from the Universe hierarchy and all Cisco Nexus devices with a virtual context will be discovered and modeled under the Virtual Device Manager hierarchy in the Explorer view. You can still see individual Cisco Nexus devices under the Universe view.

From 10.2.2, the Topology view for CISCO VDC container is not available, as the contexts are placed under Universe and connectivity details are shown in the Universe topology.

If you have already modeled the Cisco Nexus devices in 10.2.0 and have the Cisco VDC container in Universe hierarchy, after upgrading to 10.2.1. To move the Cisco VDC devices out of VDCContainer into Universe, remove VDCContainer, follow these steps

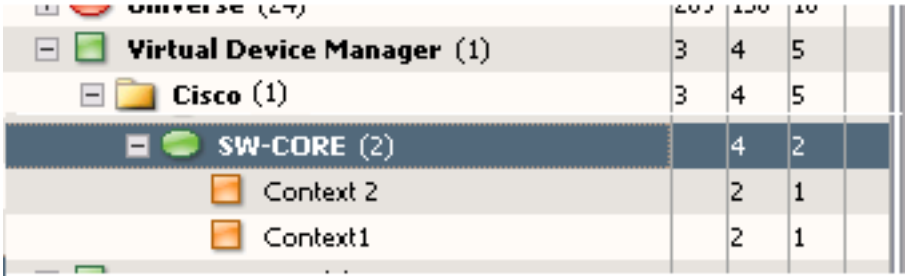
1. Download the following script:
   150
2. Unzip the UpgradeVDC.zip into the <SPECROOT> directory on each SpectroSERVER.
3. Run the script from the command line with no options.
   ```
   ./UpgradeVDCpl.pl
   ```
4. Please run the script on all SpectroSERVERs (that have Cisco Nexus devices modeled.) in a DSS environment. The Cisco VDC container hierarchy is deleted from the Universe view, and the Cisco Nexus devices are now available/ modeled in the Virtual Device Manager hierarchy.

Once you have completed running the script, a summary of actions performed by the script is displayed. See the sample below:

```
####### Summary Report #######
Total CiscoNXOS Models Found:
Actual Models in VDC container:
Total parent associations destroyed:
Total child associations destroyed:
Total VDC App models:
Total VDC App models having Cisco VDC Type null or 0:
Total VDC App models deleted:
Total VDC Container models deleted:
```

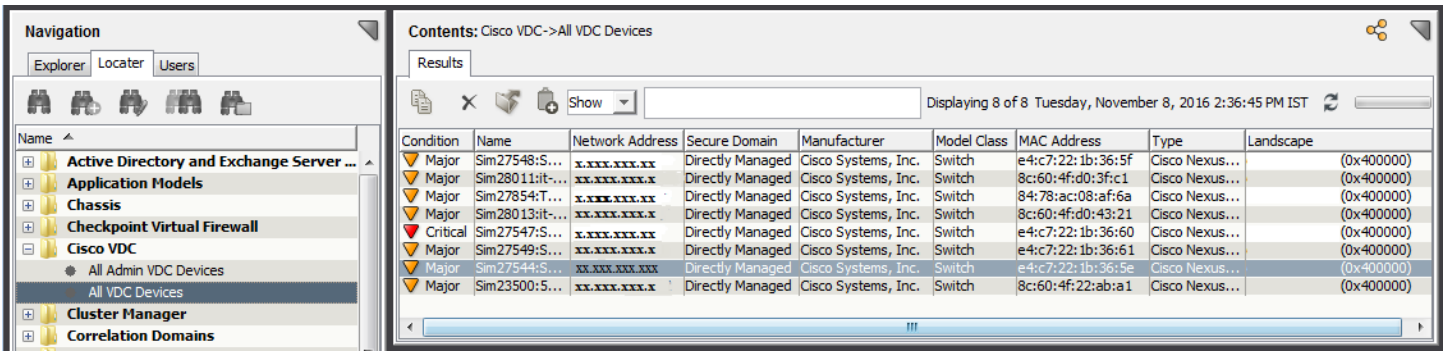
To view the models (related to Cisco Nexus devices that support VDC) in its relevant context, follow these steps:

In the OneClick Console, **Explorer View**, navigate to the **Virtual Device Manager** > select VDC container.



Locater Search for Cisco VDC

You can use pre-configured searches to locate VDC (Virtual Device Context) entities related to Cisco Nexus devices (that support VDC), in the DX NetOps Spectrum database quickly. The searches are grouped under the **Cisco VDC** folder in the **Locater** tab of the **Navigation** panel, as shown below:



The following searches are specific to Cisco VDC models:

Cisco VDC

Locates all devices that are modeled in the DX NetOps Spectrum database that has been identified as serving the specified role in one of the following searches:

- **All Admin VDC Devices**: This search will return all Admin VDC devices
- **All VDC Devices**: This search will return all VDC devices discovered and modeled

Follow these steps, to view VDCs associated to Cisco Nexus 7000 Series devices:

1. Navigate to **Locater** tab, **Cisco VDCs**, and select one of the following:
   – All Admin VDC Devices:
   – All VDC Devices:
2. Select the landscapes you wish to search against, in the Select Landscapes to the Search dialog box.
3. Click OK.
   The results matching your query are displayed in the Contents pane.

Alarm Correlation

After modeling of the logical VDCs is complete, DX NetOps Spectrum monitors the virtual entities normally as independent switch / physical chassis devices. As a result, when any number of VDC's go down within the Cisco Nexus VDC container, the corresponding number of alarms is generated on Spectrum.
Accordingly, if the container is down and assuming that the VDC's are functioning independently as per functionality, all VDCs will go down and raise separate alarms. However, if all the VDCs within the container go down, using its correlation domain capabilities, DX NetOps Spectrum suppresses the alarms for all VDCs and generates a single alarm on the container.

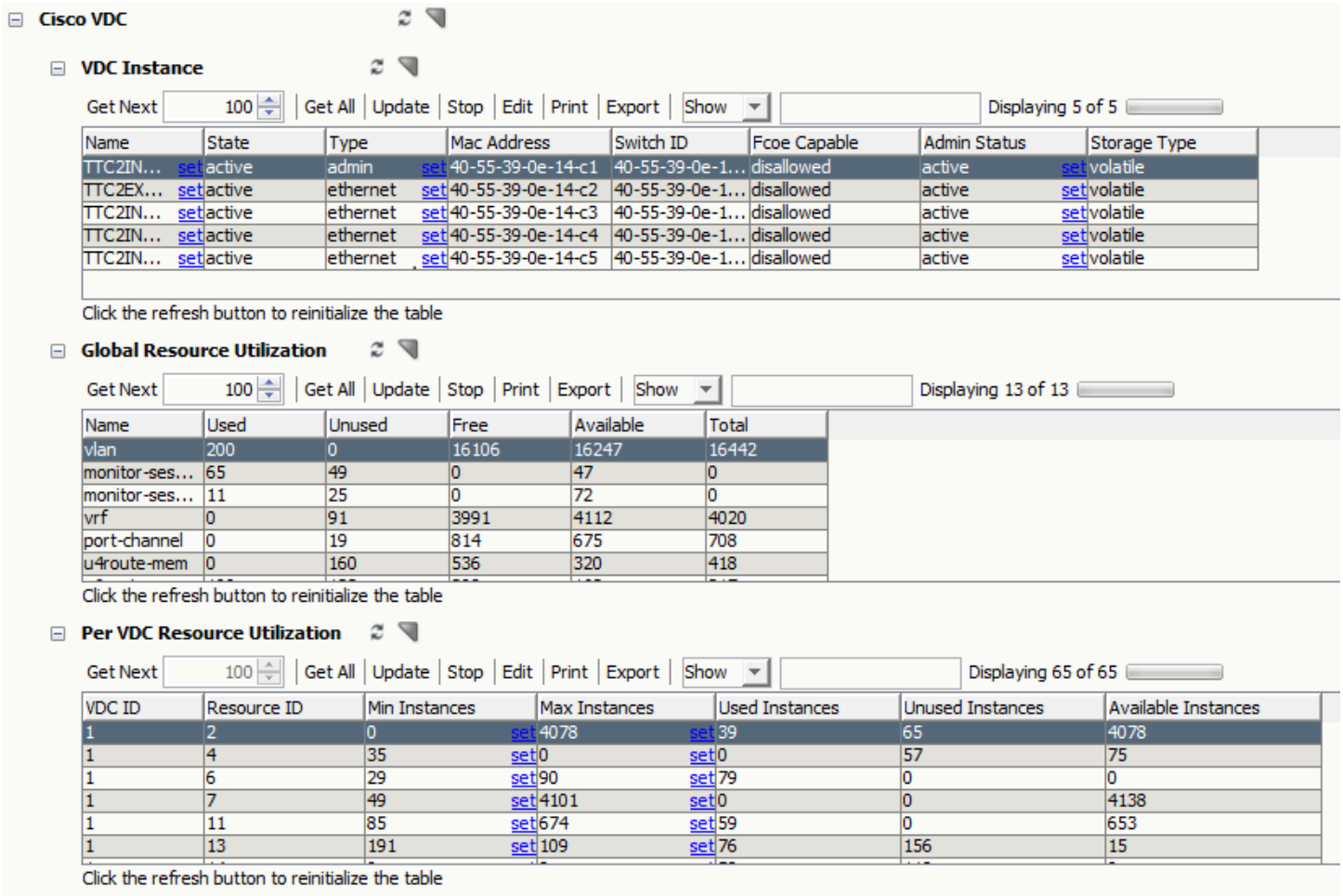
Cisco VDC Information Tab

To view the Cisco VDC Information, navigate to the **VDC context > Information Tab> Cisco VDC** (under the container which represents the physical device in DX NetOps Spectrum.)
Information is displayed under the following categories:

- **VDC Instance**
- **Global Resource Utilization**
- **Per VDC Resource Utilization**

NOTE

The container will be named on the first modeled VDC, which is renamed to the Admin VDC context after the Admin and child contexts are identified/ discovered.



VDC Instance

Field	Description
Name	The human-readable name of the instance. This name uniquely identifies the VDC Instance in the system.
State	This object indicates the current operational state of the virtual device.
Type	This object indicates whether the VDC instance is in Admin or non-admin context.
MACID	This object indicates the router MAC address of the virtual device
SwitchID	This object indicates the MAC address of the device where the virtual device instance is created.
fcoe Capable	This object indicates the FCoE capabilities of the virtual device
Admin Status	This object indicates whether the Admin status for the device is enabled or suspended.
Storage Type	This object specifies the storage type for this conceptual row.

NOTE

The following columnar objects are allowed to be writable when the storageType of this conceptual row is permanent(4):none

Global Resource Utilization

Field	Description
Name	This object indicates the name of the resource on the device.
Used	This object indicates the number of instances of a particular resource that is currently in use.
Unused	This object indicates the number of instances of a particular resource that are reserved, and currently not in use
Free	This object indicates the number of instances of a particular resource that still remain to be used.
Available .	This object indicates the number of instances of a particular resource that are available to be allocated
Total	This object indicates the total number of a particular resource

Per VDC Resource Utilization

Field	Description
VDC ID	A unique value, greater than zero, that uniquely identifies a type of resource.
Resource ID	A unique value, greater than zero, that uniquely identifies a type of resource.Min Instances
Min Instances	This object specifies the minimum number of instances of a particular resource that needs to be allocated to a particular VDC.
Max Instances	This object specifies the maximum number of instances of a particular resource that allows being allocated to a particular VDC.
Used Instances	This object indicates the number of instances of a particular resource that is currently in use for a particular VDC.
Unused Instances	This object indicates the number of instances of a particular resource that are reserved, and currently not in use for a particular VDC.
Available Instances	This object indicates the number of instances of a particular resource that is available to be allocated for a particular VDC.

Cisco ASA (Adaptive Security Appliance) Devices Failover

Cisco Adaptive Security Appliance (ASA) device family delivers enterprise-class firewall capabilities for ASA devices in an array of form factors - standalone appliances, blades, and virtual appliances - for any distributed network environment. ASA software also integrates with other critical security technologies to deliver comprehensive solutions that meet continuously evolving security needs. Cisco ASA devices offers a combination of enterprise-class stateful firewalls with a comprehensive range of next-generation network security services.

With respect to Cisco ASA (Adaptive Security Appliance) devices failover, DX NetOps Spectrum has the following capabilities:

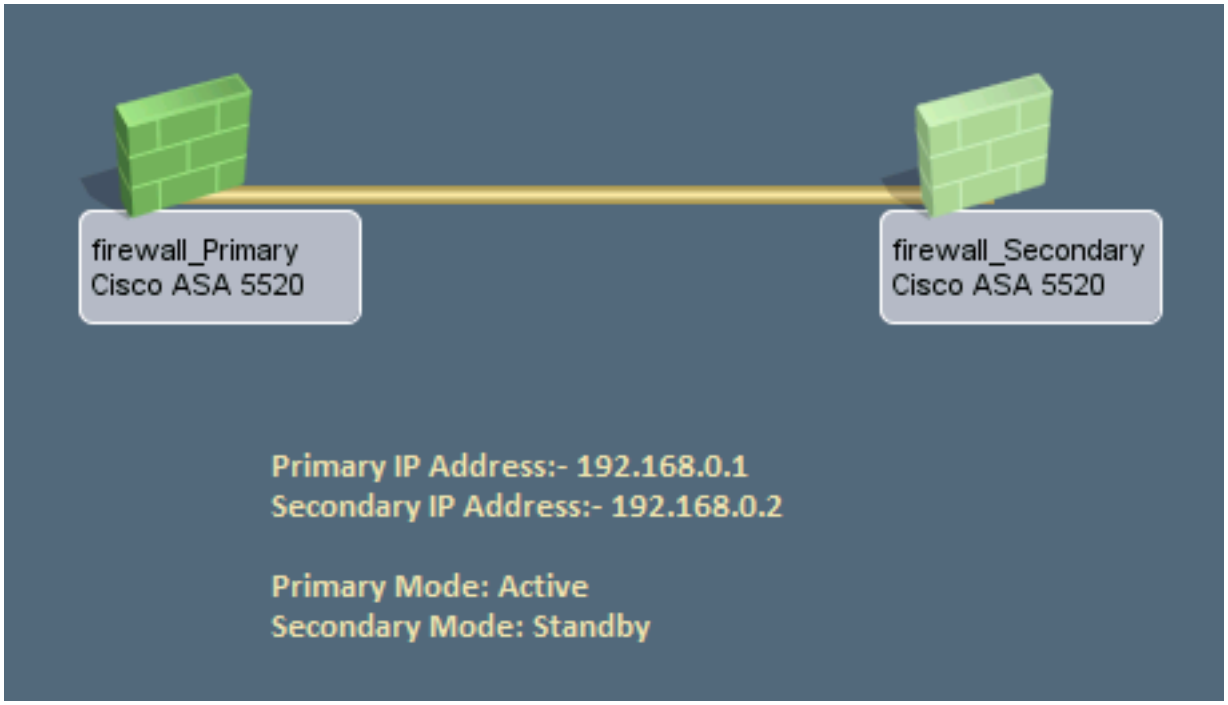
- Identifies Primary and Secondary device and appends the text to model name.
- Generates alarm if the Primary device goes to standby.
- Changes the ASA device status to Active/Standby or Active/Active.
- Polls and discovers changes of Failover States of Cisco ASA Firewalls
- Discover Connections and update the topology, when the failover occurs

NOTE

DX NetOps Spectrum does not support the Network Configuration Manager (NCM) capabilities for Cisco ASA firewall devices.

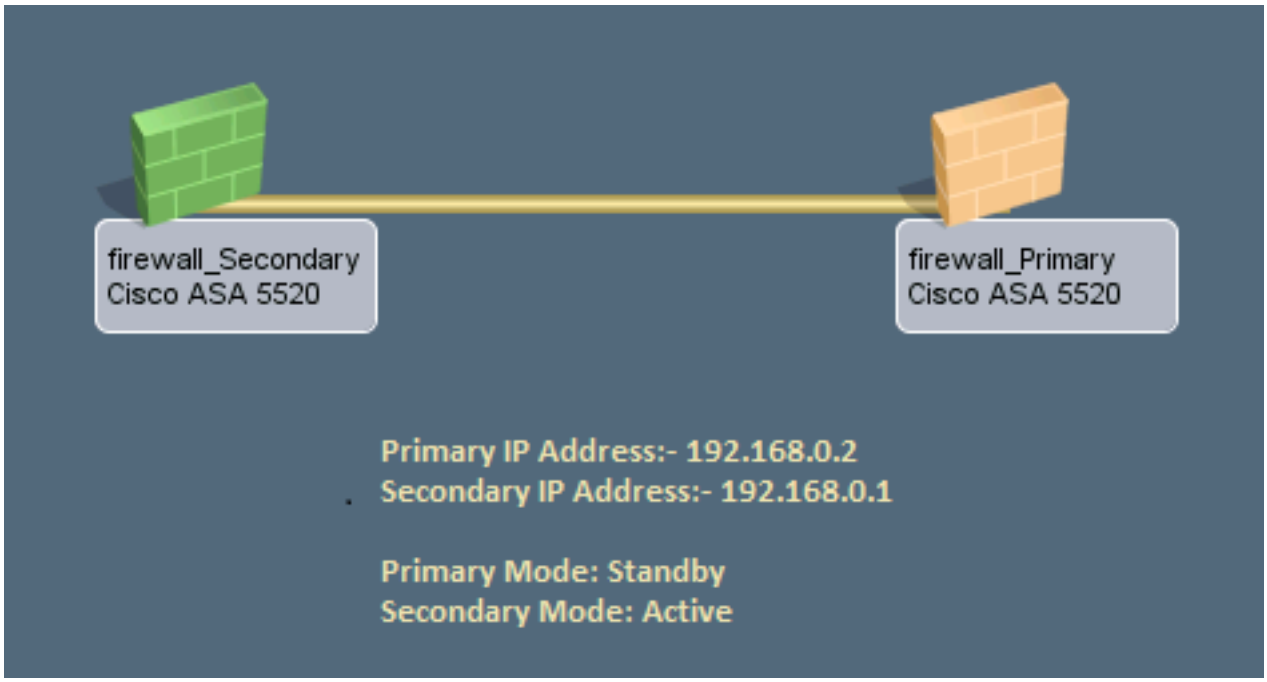
Prerequisites to Cisco ASA Firewall devices Failover Scenario:

- The Spectrum model type must be Cisco ASA. This functionality does not exist on the GnSNMPDev modeltype. To confirm:
  a. Open the Spectrum Modeling Information subview in the Component Detail of your device.
  b. Review the “Model Type Name” field on the right hand side. If it shows as GnSNMPDev (and the device is a Cisco ASA), you need to self certify your system object id from GnSNMPDev to Cisco ASA. Refer to the Self Certification Documentation.
- The two Cisco ASA Firewalls are in a **Primary** and **Secondary** Mode, which ideally means that Primary Firewall is **Active**, whereas the Firewall in secondary mode is in **Standby**.
  See image below for reference: The ASA device on the left is the Primary (Active) device, while the one on the right is the Secondary (Standby) device.



- When a failover happens, the two Firewall devices switch their configuration.
- This means that they switch their Management IP Address as well. This Management IP Address is used to model both devices in Spectrum.
- The MAC Addresses for active interfaces are also switched.

As a result, when such a Failover occurs, the modeling information in Spectrum is no longer accurate because of the IP Address switch.

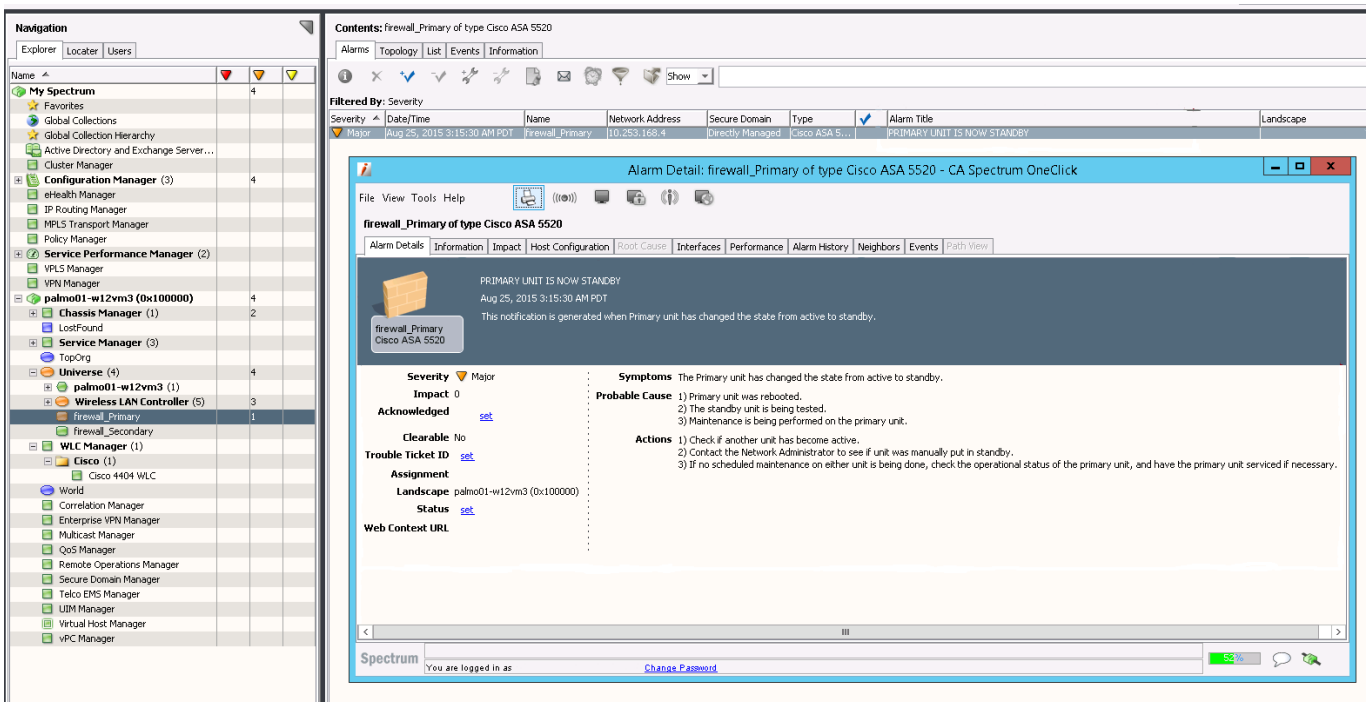


Expected Result:

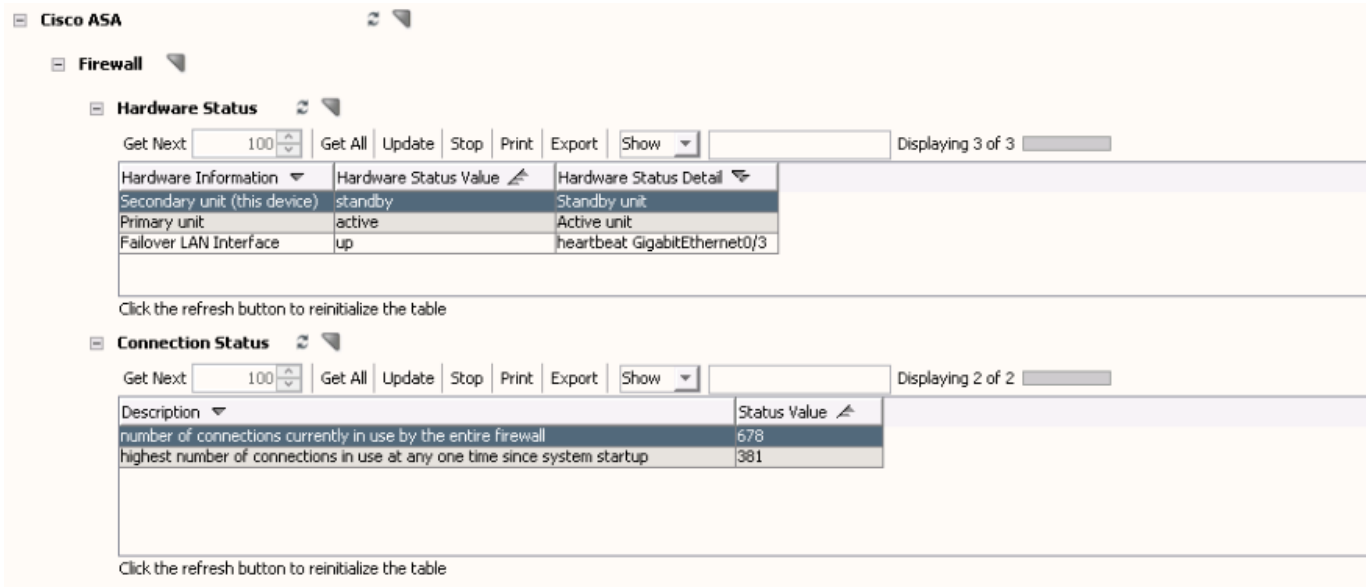
- When failover occurs on the primary device, secondary firewall device should become **Active** and primary firewall device becomes **Standby**.
- The following text: "**_Primary**" & "**_Secondary**" is appended to the model name automatically in the topology view, if the the default value is set to True.

When a Cisco ASA firewall device state goes down or is turned to standby by an Admin user manually, the following sequence of events occurs:

1. The Primary ASA device state turns from **Active** to **Standby**.
2. A major alarm "**PRIMARY UNIT IS NOW STANDBY**" is triggered on the primary device.



3. Once the the Primary ASA device is **UP** again or is brought back to **Active** state, manually.
   The Alarm "**PRIMARY UNIT IS NOW STANDBY**" is cleared on the primary device.
4. To observe Hardware status of the device, navigate to the selected **Primary Cisco ASA device** > **Information Tab** and expand the **Cisco ASA** > **Firewall** sub-view.
   Based on the hardware status we can identify if the device is **Active**/**Standby**.
   The following information is displayed for Cisco ASA devices, in **Component Detail** > **Information Tab** > **Cisco ASA.**



5. Model names are appended (with **_primary** or **_secondary**) to the device name after every polling interval, if the the default value is set to True.

NOTE

Default Polling Interval is changed to 60 seconds from 300 for Cisco ASA devices. If ***Poll Interval (sec)*** attribute value is changed from 60 to 300, the model name changes will only reflect after 300 seconds.

NOTE

The 10.2.2 release introduces an attribute suffixFWModeToModelName/0x00215325. The default value for this attribute is set as TRUE on the device model type CiscoASA, which allows for the existing functionality of the appended firewall mode (_Primary/_Secondary) to the device model name remain same.

If this value is set to FALSE, the appended firewall mode (_Primary/_Secondary) will not be added to the device model name. Therefore if the firewall mode is already added to the device model name, it will be removed from the device model name in the next polling cycle.

SNMP Support for Cisco Meraki Solutions

10.3.1 introduces SNMP support for a range of Meraki solutions including wireless appliances and switches. These include:

- **Cloud Managed Wireless**
  - Access Points
  - Oid 1.3.6.1.4.1.29671.2.1 till 29
  - Controller
    Oid 1.3.6.1.4.1.29671.1
- **Cloud Managed Appliances**
  - Oid 1.3.6.1.4.1.29671.2.100 till 124
- **Cloud Managed Switches**
  - Oid 1.3.6.1.4.1.29671.2.300 till 349
  - Oid 1.3.6.1.4.1.29671.2.356

Condition Correlation

This section discusses Condition Correlation, its various components, and how you can use Condition Correlation in DX NetOps Spectrum to achieve fault management and root-cause analysis.

About Condition Correlation

DX NetOps Spectrum Condition Correlation supports events, troubleshooting, and root-cause analysis. The Condition Correlation component lets you set up a system in DX NetOps Spectrum to determine the root-cause alarm from a heterogeneous group of managed infrastructure resources (models). You can use Condition Correlation to select the criteria that identify a causal problem event. Such events precipitate a specific set of events, which are in turn identified as symptoms. You can select a set of resources (models) for the correlation to consider and define it as the correlation domain.

Condition Correlation provides the following benefits:

- Respond to the real problem efficiently. Spend less time responding to symptomatic problems.
- Track problem trends and interdependencies.
- Respond quickly to changes in the infrastructure. You can manage multiple Condition Correlation implementations from a single landscape.

Condition Correlation Components

You can use Condition Correlation to construct a system of components that define fault indicators. You can use these components to create a process for fault association. Fault indicators specify the resources that are evaluated by the system. The following components are available to you:

- Conditions
- Rules
- Policies
- Correlation Domains

Before you begin configuring Condition Correlation, we recommend reviewing the predefined component settings. You can see these in the Condition Correlation Editor.

Conditions

Conditions are fundamental building blocks of the correlation system. A condition, like a DX NetOps Spectrum alarm, is a transitory occurrence on a resource, such as status change. A condition exists as long as the criteria that produced the condition are met. As with an alarm, a *set* event always initiates a condition, and a *clear* event clears a condition. When you define a condition, you identify the set and clear event types.

A 'set' event creates an alarm that is associated with the condition. Therefore, a condition can be cleared when the associated alarm is destroyed. Similarly, a condition is also cleared when a rule creates the condition (through its 'set' event), and no set of conditions still fulfills the rule. In this case, the condition is cleared automatically. A condition that a rule created through its set event is an *implied condition*.

You can define the conditions that correspond to DX NetOps Spectrum alarms. If the 'set' event of the condition is same as the set event of the alarm, the condition instance is instantiated after the alarm is generated. The alarm itself is linked within the correlation system. This link lets Condition Correlation hide symptomatic alarms from the main alarm list in OneClick and relate symptomatic alarms to root-cause alarms. The symptomatic alarms are listed in the Symptoms list of the root-cause alarm under the Impact tab.

> **WARNING**
> Alarms that are available at startup are not correlated.

You can also define the parameters for a condition that are used to establish correlation criteria when you create correlation rules. A *parameter* can be any event variable data or any model attribute of the model that is associated with the condition. You can create new parameters, or you can create modified versions of existing parameters.

> **NOTE**
> A correlation condition has no relationship with the condition attribute for a DX NetOps Spectrum model.

Rules

A *rule* defines the relationship between two or more conditions when specific criteria are met. You can define a rule to stipulate that one condition is a symptom of, or the cause of, another.

For example, you can associate a symptomatic SPM test threshold violation condition to a root-cause port LinkDown condition. You can apply this rule in a policy, to a set of SPM test and port models in a domain. In addition, you can create a rule to indicate that one or more conditions imply that another exists.

Rule Patterns

You can express rules in any of the following patterns:

- **Caused By**
  Condition Z causes Condition X or a set of conditions.

A correlation is made when all of the symptom conditions exist, the rule criteria apply, and the root cause condition Z exists. If Z is associated with an alarm, all symptomatic alarms are hidden under that alarm. The condition (color) of the model remains the same as before. For example, if one yellow alarm on the model hides another red alarm on the model, the other model remains red with no alarms displayed.

NOTE
When any of the conditions are cleared, the correlation is not broken.

- **Implies**
  Condition X or a set of conditions implies Condition Z.
  When all of the symptoms exist and the rule criteria apply, the root cause Condition Z is created. A 'set' event is therefore created for Condition Z, which can then create an alarm. Condition Z is only cleared if any of the symptoms are subsequently cleared, and if no other set of conditions still supports the rule. But if the condition creates an alarm, the alarm is only cleared if the condition has a 'clear' event, which must clear the alarm. Therefore, the alarm can remain, depending on its configuration.
- **Implied Cause**
  Condition X or a set of conditions is the implied cause of Condition Z.
  The Caused By and Implies pattern combines both of the previous patterns.

WARNING
Correlation using the same condition as the symptom and the cause fails.

You cannot set up a correlation using the same condition (such as implied cause) as both symptom and cause. However, you can create another condition with the same set or clear events and can use the condition as the root cause.

Example

Set up Condition A, and Rule A implies Caused by A on the correlation domain. When Alarm A is created on a device in the domain, you can see that another Alarm A is created on the correlation domain model. However, the Alarm A on the device does not become a symptom of the domain alarm.

To make the alarm a symptom of the domain alarm, you can create a B condition, similar to A, with a rule that Condition A implies Caused by B on the correlation domain.

Other Patterns

Condition Correlation lets you construct more granular rule patterns using more rule criteria that must be met before a correlation is established between two conditions. You can specify the criteria by comparing the parameters of one condition with another or in terms of specific values.

For example, an instance of a LinkDown condition on a port model can be caused by an instance of a BoardPulled condition on a board. This relationship can occur if the slot number of the port is equal to the slot number of the board, and both the port and the board are from the same device.

Policies

A *policy* is a set of one or more rules. You can group any number of rules in a policy. You can apply one or more policies to any number of resource groups (in a domain).

Use policies to simplify the implementation of rules for multiple domains. All implementations of a policy are updated after you add, edit, or remove rules from a policy.

Correlation Domains

A *correlation domain* is a group of resources that is created as a DX NetOps Spectrum container model. Condition Correlation assesses these resources collectively. This assessment is based on the rules in the policies that are applied to it. A domain can include any number of models of various model types and can have any number of policies applied. Therefore, when you select the resources in a domain, you are also deciding what is evaluated by the policy or policies that are applied to it.

You have multiple options for creating a correlation domain and populating it with resources. You can add resources on a per-resource basis. Or you can create a domain from a service or Global Collection model, which are entities that represent collections of resources.

The Condition Correlation Editor

The Condition Correlation Editor window lets you create and manage correlation system components. The window also lists and provides access to all predefined and custom (user-defined) components.

The Condition Correlation Editor window contains the following tabs:

- **Conditions**
  Lists the predefined and custom conditions. Select a condition to see a list of corresponding parameters.

  NOTE
  Not all conditions include parameters.
- **Rules**
  Lists the predefined and custom rules. When you select a rule, the corresponding correlation criteria of that rule are listed in the Rule Criteria tab.
- **Policies**
  Lists the predefined and custom policies. Select a policy to see the corresponding rules of that policy on the Rules tab.
- **Domains**
  Lists the predefined and custom domains. When you select a domain, the corresponding policies of that domain and resources are listed in the Policies tab and the Resources tab respectively.

The Condition Correlation Editor provides buttons to let you create, edit, copy, and delete conditions, rules, policies, or domains.

Use the Filter field to specify the condition, rule, policy, or domain entries to display in the editor window.

Open the Condition Correlation Editor

The Condition Correlation Editor lets you configure all Condition Correlation component settings. You must have OneClick administrative privileges to access the Condition Correlation Editor.

Follow these steps:

1. Log in to OneClick.
2. Select Tools, Utilities, and then Condition Correlation Editor.
   The Condition Correlation Editor window opens. By default, it displays the Conditions tab list and any parameters that are defined for the selected condition.

Condition Correlation Import and Export Features

You can import or export correlation data using the Import or Export features in the Condition Correlation editor. The following options are available in the Condition Correlation editor:

- **Export**
  Exports the correlation data and saves it to an XML file.
- **Import**
  Imports the correlation data from an XML file. The three scenarios that can occur while you import the data are as follows:
  - **SKIP**
    Defines the entry that already exists. The skipped entries are prefixed with [SKIP].
  - **REPLACE**

Replaces the existing entry of the same name and type only if you enable the Replace Existing option. The replaced entries are prefixed with [REPLACE].

– **IMPORT**
Defines all other entries that are not prefixed with SKIP or REPLACE. These entries are prefixed with [IMPORT].

NOTE
The import and export of domains is not supported.

How to Create a Condition Correlation Domain

Deploying a correlation system to a particular group of managed infrastructure resources is synonymous with creating a correlation domain. Once the domain is created on a landscape, the correlation system is in effect.

NOTE
Condition correlations are implemented in a SpectroSERVER or in multiple SpectroSERVERs. Therefore, condition correlations are not affected if you start or stop the OneClick web server.

Verify the following information before configuring the required domain parameters:

- A domain has at least one policy that is applied to it.
- The policy includes at least one rule.
- The rule criteria are logically appropriate for the conditions it evaluates for a correlative association.

WARNING
Attempt to produce a problem to manage and test the correlation system before you deploy the system in a production environment.

Perform the following tasks to create a condition correlation domain:

1. Create a domain and add the resources that you want include in it. In a later step, you can apply one or more correlation policies to the domain.
   Once you have created a domain, you can add resources to it and can remove resources from it at any time.
2. Create one or more conditions that you want to be evaluated by correlation rule criteria.

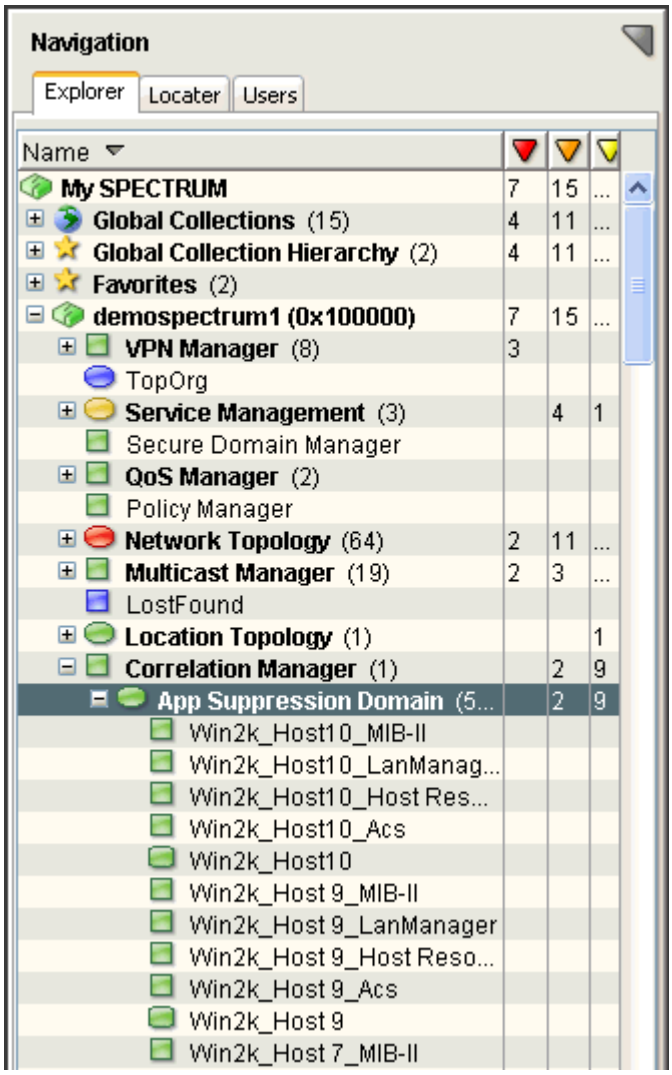
   NOTE
   If you want to use available conditions, skip this step.
3. Create the rule or rules that establish root-cause condition and symptomatic condition associations if criteria specified by the rules are met.

   NOTE
   If you want to use available rules (such as rules that specify predefined conditions), skip this step.

   A rule evaluates two or more conditions. If rule criteria are met, Condition Correlation identifies one condition as the root-cause condition and the other conditions as symptomatic of the root-cause condition. Once you have created a rule, you can modify its criteria or the conditions it evaluates at any time.
4. Create the policy or policies that contain the correlation rules to associate with the domain.
   Note: If you want to use available policies, skip this step. You can add rules to or can remove rules from a policy at any time.
5. Apply one or more policies to the domain.

   NOTE
   Existing correlation domains adjust to policy changes automatically, keeping the correct correlation state.

   The Condition Correlation process is in effect for the resources that are included in the domain. The domain is modeled as a correlation domain container in OneClick.
   The following image is an example for domain container and the resources included in it:



Creating and Managing Conditions

This section explains how to create and manage conditions using the Condition Correlation Editor. Creating a condition involves creating parameters. Managing parameters and conditions involves editing, copying, or adding new and deleting old parameters (values) or conditions.

Create a Condition

Conditions are fundamental building blocks of the correlation system. A Condition, like a DX NetOps Spectrum alarm, is a transitory occurrence on a resource, such as status change. A condition exists as long as the criteria that produced the condition are met. As with an alarm, a set event always initiates a condition, and a clear event clears a condition.

Follow these steps:

1. Open Condition Correlation Editor.
   The Condition Correlation Editor window opens.
2. Click the Conditions tab.
   A list of conditions is displayed.

3. Click the Create icon  .
   The Create Correlation Condition dialog opens.
4. Specify a value for the following condition properties:
   - **Condition Name**
     Defines the condition. For example, supply the names Power_Outage and Battery_On.
   - **Set Event Code**
     Identifies the DX NetOps Spectrum event code that is associated with the condition. When you define a condition, the set and clear event types are identified
5. (Optional, for an advanced correlation only) In the Parameters section, to specify parameters, click the Create icon  .
   The Create Parameter dialog opens. You can create parameters for the conditions as desired. For more information, see Create Parameter.
6. Click OK.

A new condition is created and added to the Conditions tab list. The Author property identifies you as the condition author.

Create a Parameter

A condition *parameter* can be any event variable data or any model attribute of the model that is associated with the condition.

Parameter values are filled in at the time the condition is created, from the event that created it, or from the model where the event was created. These parameters are then available in the Advanced Rule Criteria section.

For count conditions, a *count* parameter is available automatically, after you select that condition type in the rule.

Follow these steps:

1. Click 
   (Create) in the Parameters section.
   The Create Parameter dialog opens.
2. Provide a value for the following parameter properties:
   - **Parameter Name**
     Identifies the parameter. Provide a name that indicates the parameter type.
   - **Parameter Type**
     Specifies the type of parameter. Choose *one* of the following options:
     - Model Attribute: Specifies a model attribute parameter type.
     - Var Bind: Specifies a Var Bind parameter type.
     - Predefined: Specifies a Model, Model Type, or Device Model.
   - **Parameter ID**
     Identifies the type of parameter.
     If you select Model Attribute, click Attribute to open the Attribute Selector dialog and select the appropriate Model Attribute ID.
     If you select Var Bind, enter the Var Bind variable number that is associated with the trap for the model.
     If you select Predefined, select *one* of the following attributes from the adjacent Parameter Type drop-down list:

- Model: Enter the Model_Handle associated with the condition (Attribute ID 0x129fa).
- Model Type: Enter the Model_Type_Handle of the model that is associated with the condition (Attribute ID 0x10001).
- Device Model: Enter the Device_Mdl_Handle of the model that is associated with the condition (Attribute ID 0x10069).

– **Use as discriminator**
Designates the parameter as a discriminator. This setting lets you clear only the 'set' events that include parameter values that match the values in the 'clear' event. You can designate multiple parameters as discriminators. When condition parameters are designated as discriminators, the condition maintains the parameter values that were in place when the set event produced the condition. A condition can only be cleared if the 'clear' event contains parameter values that match the values in the 'set' event.

NOTE
To use different discriminators for special situations, you can use the same condition discriminators that the associated alarm uses. If you use the same condition discriminators as the alarm, the conditions match the alarms and clear accordingly.

3. Click OK.
The parameter is created.

Manage a Parameter

You can edit, copy, and delete the parameter values of all parameters that are listed in the Parameters section.

Follow these steps:

1. Select the parameter in the Parameters section and click

(Edit).
The Edit Parameter dialog opens.
2. To copy a parameter,
click 
(Copy).
The Copy Parameter dialog displays the property conditions of the parameter you selected.

NOTE
The Parameter Name is suffixed with _COPY because the new parameter is copied from an existing parameter and contains a unique name. If the name is already in use, a Name already exists message appears.

3. Edit the properties of the parameter as necessary, and click OK.
4. To delete a parameter, click

(Delete).
The selected parameters are removed from the Condition Parameters list.

NOTE
You cannot delete a parameter that is still in use by a rule.

Manage a Condition

In the Condition Correlation Editor window, you can edit, copy, and delete a condition from the list of predefined (CA-authored) and custom (user-authored) conditions. You can permanently delete user-authored conditions, but you cannot delete predefined conditions. If you or another user has edited and assumed ownership of a predefined condition, you can delete it temporarily. The Condition Correlation Editor restores the predefined condition with its default settings when you restart the OneClick server. You cannot delete a condition that is in use by a rule.

> **WARNING**
> Any changes that you make to existing conditions forces Condition Correlation to drop all current conditions of the same type.

Follow these steps:

1. Click the Conditions tab in the Condition Correlation Editor window.
   A list of conditions is displayed.
2. Select the condition to edit and click
   
   (Edit).
   The Edit Condition dialog opens displaying the property settings of the condition you selected.
3. Edit the values for the following condition properties:
   - **Set Event Code**
     Identifies the DX NetOps Spectrum event code that is associated with the condition.
   - **Clear Event Code**
     (Optional) Identifies the DX NetOps Spectrum clear event code that is associated with the condition.
4. (Optional, for an advanced correlation only) Specify parameters for the selected condition.
   Update one or more parameters that can be used to determine a correlation that is made between instances of the specified condition.
5. To copy a condition, click
   
   (Copy).
   The Copy Condition dialog opens, displaying the property settings for the condition you selected.

   > **NOTE**
   > The Condition Name is suffixed with _COPY because the new condition is copied from an existing condition and contains a unique name. If the name is already in use, a "Name already exists" message appears.
6. To delete a condition, click
   
   (Delete).
   Condition Correlation removes the conditions from the Conditions tab.

Creating and Managing Rules

This section explains how to create and manage rules.

Create a Rule

A rule defines the relationship between two or more conditions when specific criteria are met. You can define a rule to stipulate that one condition is a symptom of, or the cause of, another condition. For example, you can associate a

symptomatic condition with a root-cause condition. You can apply this rule in a policy or to a set of models in a domain. In addition, you can create a rule to indicate that one or more conditions imply that another condition exists.

Follow these steps:

1. Open the Condition Correlation Editor.
   The Conditions tab is displayed by default.
2. Click the Rules tab.
   A list of rules is displayed.
3. Click
   
   (Create).
   The Create Rule dialog opens.
4. Enter a name for the rule in the Rule Name field.
5. (Optional) Click set in the Type column of each item you select, specify the symptom condition to belong to a correlation domain, and select one of the following options:
   - **Exists:** The condition is in the correlation domain.
   - **Not Exists:** The condition is not present in the correlation domain. This option lets you create rules that can only be satisfied if the condition does not exist in the correlation domain.
   - **Counts:** The condition is in the correlation domain, and it enables totals/limits/range comparisons using the Advanced Rule Criteria section of the Create Correlation Rule dialog. This option lets you create rules only if a particular condition exists, reaches a limit, or is in a user-defined range.
     When using a condition for counting, a new parameter is automatically created for that condition named "Condition Count." This count can be used in the Advanced Rule Criteria section, as shown in the following example:
     ```
     TestCondition.Condition Count GREATER_THAN 10.
     ```
     No other parameter can be used for counted conditions. Because multiple copies are present, Condition Correlation cannot determine the condition from which to derive the parameter value.
6. Select one or more symptom conditions in the Symptom Condition(s) list.

   NOTE
   The rule is created based on the selected symptom conditions.
7. Select *one* of the following values from the Relationship drop-down list to specify the relationship between symptomatic conditions and the root cause condition.
   - **Caused By:** The alarm that is associated with the root cause condition caused the associated symptomatic conditions. When the rule is met, OneClick suppresses the alarms for symptomatic conditions and lists them as symptoms under the Alarms view Impact tab in OneClick.
   - **Implies:** The symptomatic conditions suggest the existence of another condition that can be unknown to the management system. When the rule is satisfied, the set event of the implied condition is processed on the target model. This condition can raise an alarm on the target model, but OneClick does not suppress the alarms for symptomatic conditions.
   - **Implied Cause**: This rule incorporates the logic of both the Caused By and Implies rules. The symptomatic conditions are indicative of another condition. The set event of this implied condition is processed on the target model. If this event raises an alarm on the target model, OneClick suppresses the alarms that are associated with the symptomatic conditions. The suppressed alarms are listed as symptoms of the root cause alarm under the Impact tab in OneClick.

     NOTE
     If you select Implies or Implied Cause, the Root Cause Target selection box is displayed on the Correlation Rule dialog. Root Cause Target lets you specify the alarm that can be generated on the correlation domain with which the rule is associated or on one of the symptomatic conditions.

To associate the implied alarm (event) with a model, add the predefined "Model" parameter to the condition that you know is created on the target model. Then select this condition and the "Model" parameter as the root cause target from the Root Cause Target section.
You can imply the condition (event/alarm) on a model where you do not have an alarm and include the model in the correlation. Consider the following examples:

- For a container, select the Model Active condition for that model, and add some rule criteria to identify the correct Model Active condition.
- For a port alarm, add the Device Model parameter to the port condition and add a criterion in the rule that specifies "Model Active.Model EQUAL TO PortCondition.Device Model". The implied condition is created on the desired model. The "Model Active" condition is created once for each model participating in the correlation domain.

NOTE
If you select Implies or Implied Cause, the Clear Symptom condition if Implied Condition is cleared check box is displayed.

8. Select a condition from the Root Cause Condition dialog that caused, or was the implied cause, of the symptomatic conditions.

   NOTE
   You can select only one root cause condition for a rule.
9. (Optional) Select the Clear Symptom condition if Implied Condition is cleared check box.
   The symptom conditions are cleared when the implied condition is cleared. This feature works with a chain of implications similar to the following scenarios:
   – ConditionA implies ConditionB, and ConditionB implies ConditionC
   – You assert ConditionA, and the SpectroSERVER then asserts ConditionB and ConditionC
   – You clear ConditionC, and the SpectroSERVER then clears ConditionB and ConditionA
10. (Optional) Click Show Advanced.
    The Advanced Rule Criteria workspace opens. You can use advanced rule criteria when you have specified condition parameters and you want to establish correlation criteria that are based on parameter or topology values. In addition to the parameter comparison, you can also include topology information (association between models).
11. Click OK.
    A new rule is created and added to the Rules tab list. The Author property identifies you as the author of the rule.

Manage a Rule

In the Condition Correlation Editor, you can edit, copy, and delete rules from the list of predefined and custom rules.

Follow these steps:

1. Click the Rules tab in the Condition Correlation Editor window.
   A list of rules is displayed.
2. Select the rule that you want to edit and click

   (Edit).
   The Edit Rule dialog opens.

   NOTE
   You cannot edit a rule name when the rule is specified in a policy.

3. To copy a rule, click
   
   (Copy).
   The Copy Rule dialog displays the property settings for the rule.

   NOTE
   The suffix _COPY is appended to the Rule Name to provide a unique name for the new rule. A unique name is required.
4. To delete a rule, click
   
   (Delete).
   Condition Correlation removes the selected rules from the list on the Rules tab.
5. Edit the properties of the rule as necessary, and click OK.
   The Condition Correlation Editor saves your changes.

Topology Information

The Advanced Rule Criteria let you specify topology information when you create a rule. Topology rules create associations between models that are used in the correlation rules. You can insert regular parameter criteria, topology criteria, or both. The topology rule criteria, like parameter criteria, are applied to the parameters of conditions. These condition parameters must be the model handles or must be convertible to model handles.

The operator that is used on condition parameters is a relation procedure. The relation procedure lets the rule verify the existence of the relationship between the two models. The operator stands for the type of relationship.

The Topology operator stands for the following relations:

- **Regular Relations**
  Represent regular associations. The rule criteria evaluate to TRUE when the left model is associated with a topology relation to the right model. The following regular relations operators are used:
  – Connects_to
  – HasPart
  – Manages
  – Collects
  – Correlates
- **Special Relation**
  Represents special associations. The rule criteria evaluate to TRUE when the left model is a port of the device model on the right. One special topology relation operator is available: IsPortOf.

Update Topology Operators in Configuration Files

The topology operator is a relation procedure that checks whether the relationship exists between two models. You can use all available topology operators in your correlations. DX NetOps Spectrum also lets you update the topology operators in the configuration files.

Follow these steps:

1. Copy the topology association configuration file, *$SPECROOT*/tomcat/webapps/spectrum/WEB-INF/event/config/topology-criteria- operator-choices.xml, to the *$SPECROOT*/custom/event/config area.

   NOTE
   Create the *$SPECROOT*/custom/event/config directory if it does not exist.

2. (Optional) Update the configuration file to add, delete, or modify a topology operator.
   Note: You can add or remove any number of topology operators from the configuration files.
3. Restart the Tomcat web server.
   The updates take effect when you reopen the Condition Correlation Editor.

Example: Add an 'IsAdjacent_to' topology operator

This example adds the IsAdjacent_to topology operator in the configuration file.

```
<criteria-choice>
 <relation-choice>
    <name>IsAdjacent_to</name>
      <verbose>is adjacent to</verbose>
   <relation-id>0x00010007</relation-id>
  </relation-choice>
</criteria-choice>
```

The code has the following parameters:

- **name**
  Indicates the name of the relation as defined in DX NetOps Spectrum.
- **verbose**
  Indicates the verbose text that is shown for the relation name.
- **relation-id**
  Indicates the relation ID as defined in the database.

  WARNING
  Set the relation-id properly. Otherwise, the correlation does not work.

Creating and Managing Policies

This section discusses how to create and manage policies using the Condition Correlation Editor in OneClick.

Create a Policy

A policy is a set of one or more rules. Create the policy or policies that contain the correlation rules to associate with the domain. Apply the policies to domains to create correlation domains.

Follow these steps:

1. Open Condition Correlation Editor.
   The Conditions tab is displayed by default.
2. Click the Policies tab.
   The Condition Correlation Editor window displays a list of policies.
3. Click
   
   (Create).
   The Create Correlation Policy dialog opens.
4. Supply a value for each of the following policy properties:
   - **Policy Name**
     Defines the policy (such as Power_Outage, DiskPolicy).
   - **Policy Rule(s)**
     Includes the rules for the policy. You can use the arrow buttons to add rules from the Available Rules list to the Policy Rule(s) list, or to remove rules from the Policy Rule(s) list.

5. Click OK.
   A new policy is created and added to the Policies tab list. The Author property identifies you as the author of the policy.

Manage a Policy

In the Condition Correlation Editor window, you can edit, copy, and delete a policy from the list of predefined and custom policies.

Follow these steps:

1. Click the Policies tab in the Condition Correlation Editor window.
   The Condition Correlation Editor window displays a list of policies.
2. Select the policy to modify, and click
   
   (Edit).
   The Edit Policy dialog opens.

   NOTE
   If the policy is applied to a correlation domain, you cannot edit that policy name.
3. To copy a policy, click
   
   (Copy).
   The Copy Policy dialog opens, displaying the property settings for the policy you selected.

   NOTE
   The suffix _COPY is appended to the Policy Name to create a unique name for the new policy. A unique name is required.
4. Edit policy properties as necessary, and click OK.
   The Condition Correlation Editor saves your changes.
5. To delete a policy, click
   
   (Delete).
   A confirmation dialog opens.
6. Click Yes.
   The Condition Correlation Editor removes the selected policies from the Policies tab list.

Creating and Managing Domains

This section discusses the concept of a "Correlation Domain", and discusses how to create and manage a domain using the Condition Correlation Editor in OneClick.

About Correlation Domains

You can create *correlation domains* that contain different correlation policies for various types of managed resources and alarm events. This section describes how to create correlation domains and edit domain settings.

WARNING
The volume of correlation processing that is required for large domains can affect DX NetOps Spectrum performance.

In the Condition Correlation Editor, you can create a domain, or you can copy a domain and modify it.

You can also create a domain from the context of a device, service, or Global Collection model that you want to add to the domain. Use the OneClick ‘Add To’ feature to create a domain in context.

> **NOTE**
> If you plan to add resources to the correlation domain from multiple landscapes, create the domain on the Main Location Server.

Create a Domain in the Condition Correlation Editor

In DX NetOps Spectrum, a *domain* is a group of resources. DX NetOps Spectrum Condition Correlation evaluates these resources collectively. You can apply policies to domains to create correlation domains. The rules in the policies that are applied to the domain are executed on all resources in the domain.

Follow these steps:

1. Open Condition Correlation Editor.
   The Conditions tab is displayed by default.
2. Click the Domains tab.
   The Condition Correlation Editor window displays a list of any domains that users have created. Condition Correlation does not include default domains.
3. Click
   
   (Create) to create a domain.
   The Create Domain dialog opens.
4. Provide a value for each of the following domain properties:
   - **Domain Name**
     Defines the domain (such as Backup_Power, DiskMonitor).
   - **Landscape**
     Defines the landscape for the domain.
5. Move one or more policies from the Available Policies box to the Domain Policies box. If you are creating another version of an existing domain, remove policies as required from the Domain Policies box.
6. Add or remove resources from the domain by taking the following steps:
   a. Click the Resources tab, and click Create.
      The Locate Resources dialog opens.
   b. Search for the resources to add to the domain in the ‘Search using’ panel.
   c. Select the resources to add to the domain from the search list, click ‘Add Selected to Correlation Domain,’ and click Close.
      The resources that you added appear under the Resources tab in the Create Correlation Domain dialog.

      > **NOTE**
      > To add a device model and some port models for the device, add each individual model to the domain. Adding the device model does not add its component models to the domain.
7. Click OK.
   The Condition Correlation Editor saves the new domain to the Domains tab list.

Create a Domain in the OneClick Console

You can use the OneClick 'Add To' feature to create a domain from the context of a device, service, or Global Collection model.

Follow these steps:

1. Select the model in OneClick that you want to use to create a domain.
2. Right-click the model, and select Add To Correlation Domain.
   The Add to Correlation Domain dialog opens.
3. Perform one of the following actions:
   – To create a domain, enter a name for the domain and specify the landscape where you want to create the domain in the 'Create a new correlation domain' section.
   – To include the device, service, or Global Collection model in an existing domain, select the existing domain from the list in the 'Select an existing correlation domain' section.
4. Click OK.
   A domain is created or edited. You can add policies to the domain. For more information, see Create a Domain in the Condition Correlation Editor.

Manage a Domain

In the Condition Correlation Editor, you can copy and modify domains from the list of predefined and custom domains. You can delete domains that were created by any user.

Follow these steps:

1. Click the Domains tab in the Condition Correlation Editor window.
   The Condition Correlation Editor displays a list of domains.
2. Select the domain to edit and click
   
   (Edit).
   The Edit Domain dialog opens.
3. To copy a domain, click
   
   (Copy).
   The Copy Domain dialog opens.
4. Edit the Domain properties as necessary, and click OK.
5. (Optional) Remove resources from a domain that you are copying by taking the following steps:
   a. Select the resources that you want to remove from the Resources tab list in the Edit Correlation Domain dialog.
   b. Click Delete.
      The selected resources are removed from the Resources tab list.
   The Condition Correlation Editor saves your changes.
6. To delete a domain, click
   
   (Delete).
7. Click Yes in the confirmation dialog that opens.
   Condition Correlation Editor removes the selected domains from the Domains tab list.

Testing and Debugging

DX NetOps Spectrum Condition Correlation provides advanced capabilities to enhance the functionality of the base product. When you create or customize the conditions, rules, and policies, that compose a condition correlation system, testing is required. We recommend staging the deployment of each new system to enable testing and debugging.

The topics in this section describe the testing and debugging process in Condition Correlation and recommend some best practices.

How to Develop and Test Correlations

This section describes the design and development process for Condition Correlation. The lifecycle of a condition correlation follows typical software development methodology. Perform the following tasks in this recommended order to create a correlation:

- Create conditions
- Create rules
- Create policies
- Create correlation domains and add models to the domains

> **NOTE**
> Use the same order to develop new correlations using the Condition Correlation Editor.

After creating the condition correlation system, test and debug it. The testing and debugging process for a new correlation includes the following tasks:

1. Simulate the symptom condition on the appropriate model.
2. Verify that the appropriate alarm or event is raised in the OneClick Alarm View.
3. Verify that the condition is recognized by the correlation domain.
4. Simulate the root-cause condition of the appropriate model(s) in the correlation domain.
5. Verify that the condition is recognized by the correlation domain.
6. Verify that the root-cause alarm is raised correctly and that the symptom condition is hidden in the OneClick Alarm View.

Guidelines and Best Practices

Verify the following guidelines and best practices before starting the development process:

- Become familiar with the preconfigured Condition Correlation components, such as the conditions, rules, and policies that are installed with DX NetOps Spectrum. Use the correlation components when required. You can create copies and can edit the preconfigured correlation components. For more information, see Condition Correlation Components.
- Start with simple conditions and rules to build a more complex system.
- Design the rules for easier testing. Use the techniques to test Condition Correlations that are explained in this document.
- Start testing rules and conditions from the bottom of the hierarchy and move up the hierarchy.

Testing a Correlation

After you design and develop a Condition Correlation system, test the system elements before deploying the correlation in a production environment. Multiple methods are available to perform the validation and verification process. The most robust method is to use a live environment. In certain circumstances, this method is not possible. For example, some DX NetOps Spectrum operators and developers cannot bring down infrastructure resources for testing a correlation.

A lab environment can provide a suitable test-bed for this type of verification. However, a lab can lack some of the resources that are required to test the correlation and simulate the scale of the deployment.

The simplest way to test a new Condition Correlation system is to create artificial events on DX NetOps Spectrum models. The following methods are available to test the Condition Correlation system:

- Test the Correlation with the Command Line Interface
- Test the Correlation with the Web Services API

NOTE
The Web Services API method provides more capability with greater complexity.

Test the Correlation with the Command-Line Interface

Use a simple command from the DX NetOps Spectrum Command-Line Interface (CLI) to test a new correlation. All events in DX NetOps Spectrum have an ID that is used to identify the event when it is processed. To know the type of the event that you are creating, an event-type-id is required. The event-type-id can be obtained from any of the following sources:

- The Condition Correlation Editor
  Note: The event is typically defined in an existing condition.
- The Event Configuration tool.
- The EventDisp file that refers the specific event and how it is handled.

Follow these steps:

1. Select Start, Programs, and Command Prompt.
   The DOS prompt appears, ready to accept CLI commands.
2. Start the SpectroSERVER to which you want to connect.
3. Navigate to the following vnmsh directory in the DX NetOps Spectrum installation directory:
   ```
   $ cd $SPECROOT/vnmsh
   ```
4. Open the connection using the following command:
   ```
   $ connect
   ```
   You are connected to the CLI session.

   NOTE
   On a UNIX platform, you can start a CLI session from the shell prompt. You can also start a CLI session from a bash shell prompt on Windows platform. For more information, see the Command Line Interface section.
5. Execute the following CLI command to test the correlation:
   ```
   create event type=event-type-id text=event-text mh=model-handle
   ```
   An event is created on the model with the given mode handle.

Example

To simulate the Chassis Down event on a model (with the model handle of 0x10234), use the following command:

```
create event type=0x10f69 text="Chassis is Down" mh=0x10234
```

NOTE
This command works for some situations. However, it does not let you deliver event variables in the event message. To generate a more complicated event, use the web services method.

Test the Correlation with the Web Services API

Use the DX NetOps Spectrum RESTful Web Services API to test the correlation. The DX NetOps Spectrum RESTful Web Services lets you generate events that include event variables. This method requires a REST client that supports XML input, for example, WizTools RESTclient (on Windows 7).

Follow these steps:

1. Download and install the WizTools REST client.
   For more information, see *http://code.google.com/p/rest-client/.*
2. Determine whether event variables are required to create the event.

3. Verify the syntax in the CsEvFormat file for the relevant event.
   You can view this file through Event Configuration or by accessing the file directly, using bash shell or your preferred text editor.
   For example, to find the file for event type 0x10f96, use the following path:
   ```
   $SPECROOT/SG-Support/CsEvFormat/Event00010f96[language_pack]
   ```
   NOTE
   The language pack extension is used for releases 9.3.0 or later. The extension for US English is '_en_US'.
4. Review the contents of this file. It does not contain event variables. It contains the following text:
   ```
   The <ss> physical Memory has exceeded 2.5 Gigabytes for more than 300 seconds.
   ```
5. If you do not see event variables in the event message, use the template that lacks event variables. For more information, see REST Examples for Correlation Testing.
6. For event type 0x5180302, verify the text for the following message:
   ```
   The BGP Peering session from S 1 to S 2 has been Lost.
   ```
   The event variables in italics are required for the event to be generated correctly. In this example, following event variables are applicable:
   - **S 1**
     Represents the device model name.
   - **S 2**
     Represents the Provider_Cloud model name.

   The context of these parameters is determined by reviewing the actual events that have been generated.
7. Use the template to generate an event with the appropriate event variables. For more information, see REST Examples for Correlation Testing.
   An event is generated.

Verify the Simulated Events

You can verify the simulated events that are generated through the CLI or web services. After the test tools are configured, you can verify the events in the Event View for the target model.

Follow these steps:

1. Open the Condition Correlation Editor.
   The Condition Correlation Editor window opens.
2. Select a DX NetOps Spectrum model in one of the views.
   For example, select a view from the Navigation, List, Topology, or Search Results pane.
3. Select the Event tab for that model.
   All events are displayed for the selected model.
4. Enter text in the event filter dialog to search for events that contain relevant text.
   For example, if you are simulating a Border Gateway Protocol (BGP) backwards transition event, type 'BGP' as the keyword in the filter dialog.
5. Verify that the event is correctly displayed in the event window.
6. Verify that the event variables contain valid values.
   Note: Perform Step 6 only if the correlation requires specific event variables to be set or modified. For more information, see REST Examples.
   The simulated event is verified.

Debugging Correlations

Debugging is an essential component of the process to design, develop, and validate a new condition correlation system. Multiple built-in tools are available to help you debug a condition correlation. This section discusses the development prerequisites and the debugging tools that are available to debug the correlation system.

Debugging Prerequisites

Review the following prerequisites before debugging a Condition Correlation system. In our testing, we have frequently seen preventable errors that are related to these factors:

- The correlation domain has at least one policy that is applied to it.
- The correlation domain contains the models where a correlation is likely to occur. Verify that the model has not been deleted from the correlation domain.
- Rules are set up with the correct relationships.
- Symptom conditions are set up with the correct event types.
- Symptom and root-cause conditions are not reversed.
- Rules or symptoms are set up correctly to suppress the alarms.
- Conditions or policies exist in the correlation domain.

Debugging Tools

Condition Correlation is one of the most complex systems in DX NetOps Spectrum. As a result, various tools and techniques are available to debug a correlation system. The Condition Correlation debugging tools are specific actions that are sent to specific models. The following model support actions are available for the Condition Correlation debugging tools:

- Correlation Domain (custom - possibly multiple per landscape)
- Correlation Manager (predefined - only one model per landscape)

You can send these actions through the DX NetOps Spectrum CLI or Web Services. Both APIs let you create actions on models with optional parameters. A prerequisite for each approach is to find the model handle of the target model. Use the following syntax for CLI-based actions:

```
update action=0xffff0102 mh=CorrelationDomain_mh
```

All debug output, whether it is initiated through the DX NetOps Spectrum CLI or Web Services, appears in the Control Panel message window. The debugging messages are captured in the following VNM.OUT file:

```
$SPECROOT/SS/VNM.OUT
```

> **NOTE**
> Before you attempt the debugging actions, we recommend verifying the events that are produced and checking the configuration of symptoms and rules. Review the debugging prerequisites before you start the process.

Debugging Actions Correlation Domain

The following debug actions can be sent to the Correlation Domain Model through the CLI. You can see the output on the server console. The model handle of the target model is a prerequisite.

Action Code	Outputs	Usage
0xffff0102	List of existing conditions	Verify that expected conditions appear
0xffff0103	Detailed list of existing conditions	Verify that expected condition details match
0xffff0104	Condition definitions	Verify that expected conditions appear active
0xffff0105	Rule definitions	Verify that expected rules appear active
0xffff0106	Detailed Rule definitions	Verify that expected rule details match CCE
0xffff0107	Correlation hierarchy - models in domain	Verify that the target models are in domain
0xffff0202	Count conditions existing in domain	Verify that the count is an expected value
0xffff0203	Details for all count conditions	Verify that the count is an expected value

0xffff0900	Start runtime debug	Enable runtime debugging of domain
0xffff0901	Stop runtime debug	Disable runtime debugging

NOTE
Disable runtime debugging when it is not required to reduce the impact on performance and disk space.

Debugging Actions Correlation Manager

The following debug actions can be sent to the Correlation Manager through the CLI. You can see the output on the server console. The model handle of the target model is a prerequisite.

Action Code	Outputs	Usage
0xffff0100	All condition definitions	Verify that expected condition appears
0xffff0101	All condition definitions - detailed	Verify that expected condition details match
0xffff0110	All rule patterns	Verify that expected rule appears
0xffff0111	All rule patterns - detailed	Verify that expected rule details match
0xffff0120	All policies	Verify that expected policy appears
0xffff0200	All condition registrations	Verify that target model registrations are present
0xffff0300	Condition engine condition table	Verify that active conditions are represented
0xffff0401	Reload shipped condition definitions	Restore initial conditions
0xffff0402	Reload shipped rule definitions	Restore initial rules
0xffff0403	Reload shipped policy definitions	Restore initial policies
0xffff0900	Enable Condition Correlation Mgr debug	Verify initialization, registration, and notification
0xffff0901	Disable Condition Correlation Mgr debug	Disable when not required
0xffff0910	Enable condition engine debug	Verify event and alarm registrations received
0xffff0911	Disable condition engine debug	Disable when not required

NOTE
Disable runtime debug when it is not required to reduce the impact on performance and disk space.

Condition Correlation Examples

This appendix provides a workflow and examples to help you implement Condition Correlation in your environment.

NOTE
All of the fictitious instances of alarms and Condition Correlation components that are referenced in the following examples are enclosed in double quotation marks (“ ”). References to actual events and alarms that are defined in DX NetOps Spectrum are not enclosed in quotation marks.

The following scenarios are discussed as Condition Correlation examples:

- Power Outage
- Disk Full
- WAN Link Failure

How to Configure a Condition Correlation for a Power Outage

DX NetOps Spectrum Condition Correlation can be configured to determine the root-cause alarm and manage trouble-prone segments of your infrastructure. Predefined correlation systems are available in Condition Correlation Editor. However, you can use Condition Correlation to select the criteria that identify a causal problem event. With the help of Condition Correlation components (such as Conditions, Rules, Policies, and Correlation Domains), you can pinpoint the root-cause alarm and can pay less attention to symptomatic alarms.

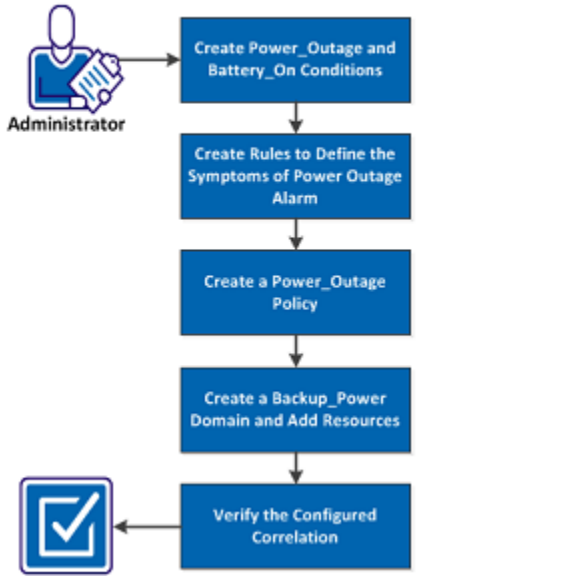
DX NetOps Spectrum includes many predefined correlations. For example, ContactLost_Red (caused by) Chassis Down, LinkDown (caused by) Chassis Down, Dev Module Pulled (caused by) Blade Status Unknown are a few predefined correlations that are available in Condition Correlation Editor.

In a power outage scenario, managed UPS systems generate traps indicating that they have switched to backup battery power. If the backup battery power fades, the systems generate traps that indicate low battery power. When the batteries fail, managed devices that are connected to the UPS systems go down. These devices trigger a flood of events and alarms from the affected area. The volume of events makes it difficult to identify and address the underlying problem.

As an administrator, you can configure a correlation system for a power outage. Create one or more conditions that can be evaluated by correlation rule criteria. If rule criteria are met, Condition Correlation identifies one condition as the root-cause condition and the other condition as symptomatic of the root-cause condition. Create a policy that contains the correlation rules to associate with the domain and apply the policy to the domain. The Condition Correlation process is in effect for the resources that are included in the domain

The following diagram illustrates the process to configure a Condition Correlation for a power outage:

How to Configure a Condition Correlation for a Power Outage



Create Power_Outage and Battery_On Conditions

Conditions are the building blocks of the correlation system. Create Power_Outage and Battery_On conditions in the Condition Correlation editor to configure a correlation system for a power outage. The Power_Outage condition uses the set event code and the clear event code that are associated with the Power Outage alarm. The Battery_On condition uses the set event code and the clear event code that are associated with the UPS trap. You can also use this procedure to create conditions to handle alarms from other root causes.

NOTE
To access Condition Correlation Editor, you require OneClick administrative privileges.

Follow these steps:

1. Open Condition Correlation Editor.
   The Condition Correlation Editor window opens. For more information, see The Condition Correlation Editor.
2. Click the Conditions tab.
   A list of conditions is displayed.
3. Click
   
   (Create).
   The Create Correlation Condition dialog opens.
4. Specify a value for the following condition properties:
   - **Condition Name**
     Defines the condition. For example, supply the names Power_Outage and Battery_On.
   - **Set Event Code**
     Identifies the DX NetOps Spectrum event code that is associated with the condition. For example, use the following set event codes:
     - Set event for Battery_On: 0x0116905a
     - Set event for Power_Outage: 0x01169431
5. Click OK.
   Power_Outage and Battery_On conditions are created.

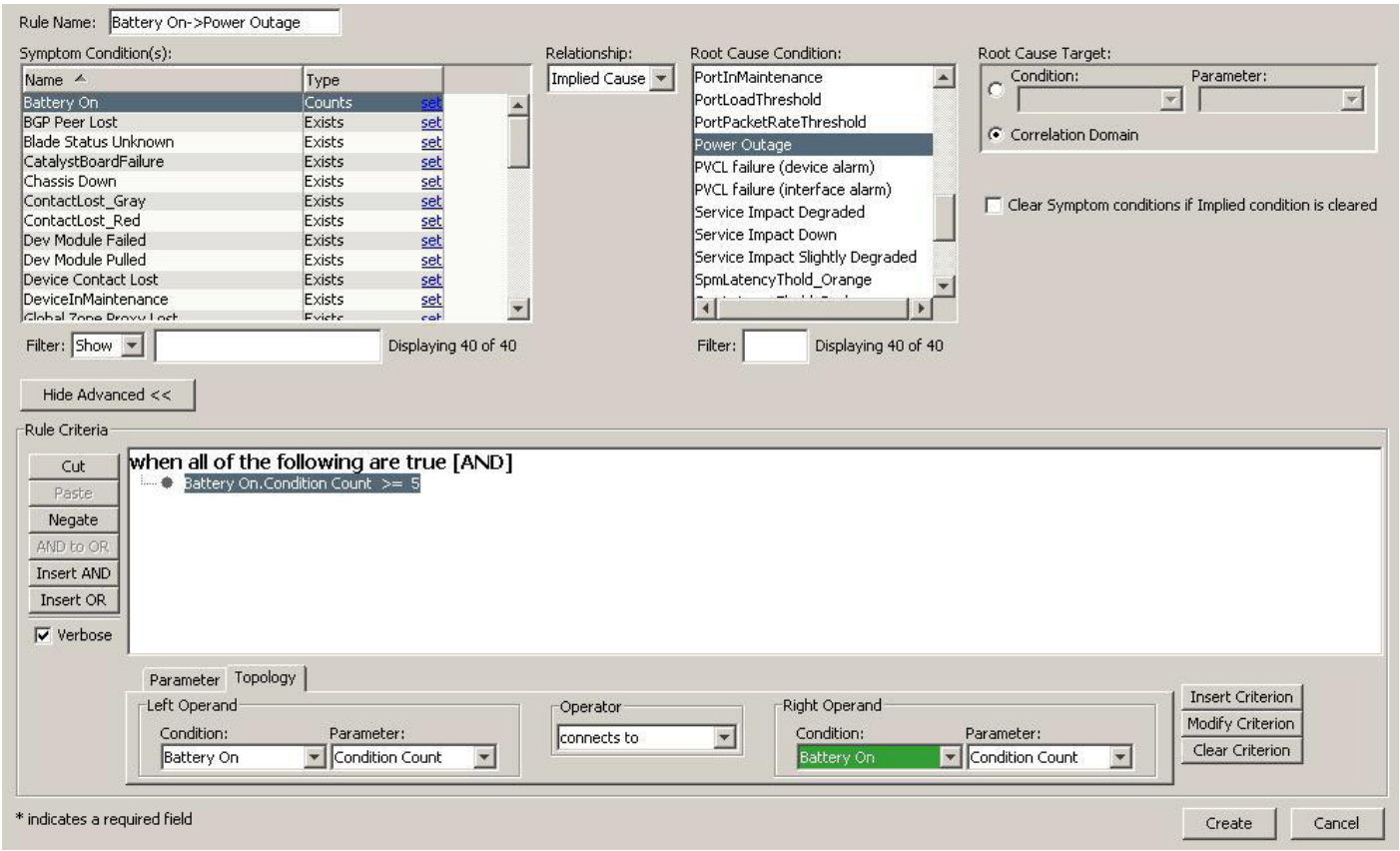
Create Rules to Define the Symptoms of the Power Outage Alarm

A rule is defined to stipulate that one condition is a symptom or a cause of another condition. Create rules to define the symptoms of the Power Outage alarm. You can create the following three rules for Power Outage:

- **Battery On -> Power Outage**
  Specifies that if five or more power systems go on battery power, the Power Outage condition is the implied cause.
- **ContactLost_Red -> Power Outage**
  Specifies that if the ContactLost_Red condition (predefined) is caused by the Power Outage condition, the critical (red) Contact Lost alarm is suppressed as a symptom of the Power Outage alarm.
- **ContactLost_Gray -> Power Outage**
  Specifies that if the ContactLost_Gray condition (predefined) is caused by the Power Outage condition, the (gray) Contact Lost alarm is suppressed as a symptom of the Power Outage alarm.

Follow these steps:

1. Open Condition Correlation Editor.
   The Conditions tab is displayed by default.
2. Click the Rules tab.
   A list of rules is displayed.
3. Click the Create
   icon 
   The Create Rule dialog opens.
4. Enter a name for the rule in the Rule Name field.
   The following image illustrates the configuration in Advanced Rule Criteria:



5. Click Create.
   Rules for Power Outage are created in Correlation Editor.

Create a Power_Outage Policy

A policy is a set of one or more rules. To configure a condition correlation system for a Power Outage, create a Power_Outage policy and add the Power Outage rules to the Policy rules list. All implementations of the policy are updated after you add rules.

Follow these steps:

1. Open Condition Correlation Editor.
   The Conditions tab is displayed by default.
2. Click the Policies tab.
   The Condition Correlation Editor window displays a list of policies.
3. Click

   (Create).
   The Create Correlation Policy dialog opens.
4. Supply a value for each of the following policy properties:
   - **Policy Name**
     Defines the policy. For example, supply the name Power_Outage.
   - **Policy Rule(s)**
     Includes the rules for the policy. You can use the arrow buttons to add rules from the Available Rules list to the Policy Rule(s) list, or to remove rules from the Policy Rule(s) list.

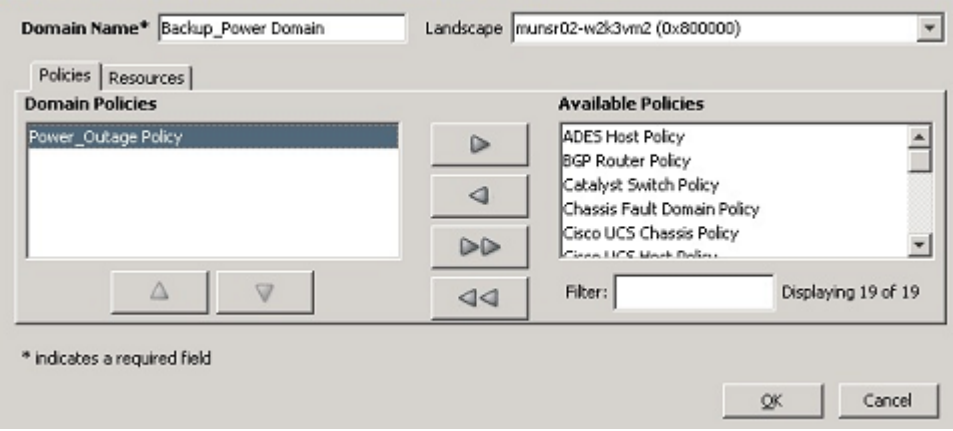
A Power_Outage policy is created that includes the Battery On -> Power Outage, ContactLost_Red -> Power Outage, and ContactLost_Gray -> Power Outage rules.

Create a Backup_Power Domain and Add Resources

A *domain* is a group of resources. You can create a Backup_Power domain for the condition correlation system for a Power Outage. UPS models and the device models (that connect to the power supplies) are the resources of the correlation domain. These resources are added and the Power_Outage policy is applied to the domain. You can include multiple models of various model types and can apply multiple policies.

Follow these steps:

1. Open Condition Correlation Editor.
   The Conditions tab is displayed by default.
2. Click the Domains tab.
   The Condition Correlation Editor window displays a list of any domains that users have created. Condition Correlation does not include default domains.
3. To create a domain, click the Create icon 
   The Create Domain dialog opens.
4. Provide a value for each of the following domain properties:
   - **Domain Name**
     Identifies the domain. For example, supply the name Backup_Power.
   - **Landscape**
     Defines the landscape for the domain.
5. To add the Power_Outage policy to the Domain Policies box, move the Power _Outage policy from Available Policies to the Domain Policies box.
   If you are creating another version of an existing domain, remove policies as required from the Domain Policies box.
   The following image illustrates the Power_Outage policy that is added to the Domain Policies box.



6. Click the Resources tab to add or remove resources from the domain. For more information, see the Creating and Managing Domains section.
7. Click OK.
   The Backup_Power domain is created.

Verify the Correlation

As a best practice, verify the correlation that you configured before you deploy it. When a correlation is correctly configured, the symptom alarms are hidden while the root cause alarm is active.

Follow these steps:

1. Log in to the OneClick console.
2. Click the Alarms tab.
   The Alarms window opens.
3. Verify the status of the symptom alarms (such as ContactLost_Red, ContactLost_Red, Battery_On) and the root cause alarm.
   The symptom alarms are hidden and the root cause alarm is displayed in Alarms tab.
   The correlation is appropriately configured.

Disk Full Scenario

A disk monitor alarm appears on many models multiple times. However, the total number of these alarms is required rather than every instance of each alarm. For an instance, less than five disk monitor alarms are acceptable, but once there are at least five alarms, you want to see a minor alarm. Similarly, if there are more than ten alarms, you want to see a major alarm. If there are more than 15 alarms, then you can see a critical alarm. The following process explains the concept of condition correlation for a Disk Full scenario:

- If any of the DiskFull events are generated on the host devices with different values for the disk (variable binding 4), they are displayed on-screen, as long as their overall number goes up to four. Once a fifth alarm is generated on a model of the correlation domain, the MinorDiskProblemRule instantiates and the MinorDiskProblem alarm is created on the correlation domain. The five DiskFull alarms are hidden as symptoms under the MinorDiskProblem alarm.
- If one or more DiskFull alarms are cleared, the MinorDiskProblem alarm clears, showing the previously hidden other four or fewer DiskFull alarms. By contrast, if more DiskFull alarms are generated and their number reaches ten, the MajorDiskProblem alarm is generated. The minor alarm, which covers 5-9 alarms, disappears. All DiskFull alarms can be the symptoms of the major alarm.
- If the DiskFull alarm total does not reach ten, you can see the MinorDiskProblem alarm. Similarly, if the DiskFull alarm total exceeds 14, you can see the CriticalDiskProblem alarm.

EventDisp Entries

You can use the following EventDisp entries for setting up Condition Correlation. These alarms use variable binding 4 as a discriminator so that multiple alarms can exist on the same device.

test alarm (disk full)

```
0xffff0000 E 50 A 1,0xffff0000,4
0xffff0001 E 50 C 0xffff0000,4
```

5 to 9 test alarms, minor problem with disks

```
0xffff0010 E 50 A 1,0xffff0010
0xffff0011 E 50 C 0xffff0010
```

10 to 14 test alarms, major problem with disks

```
0xffff0020 E 50 A 2,0xffff0020
0xffff0021 E 50 C 0xffff0020
```

more than 15 test alarms, critical problem with disks

```
0xffff0030 E 50 A 3,0xffff0030
0xffff0031 E 50 C 0xffff0030
```

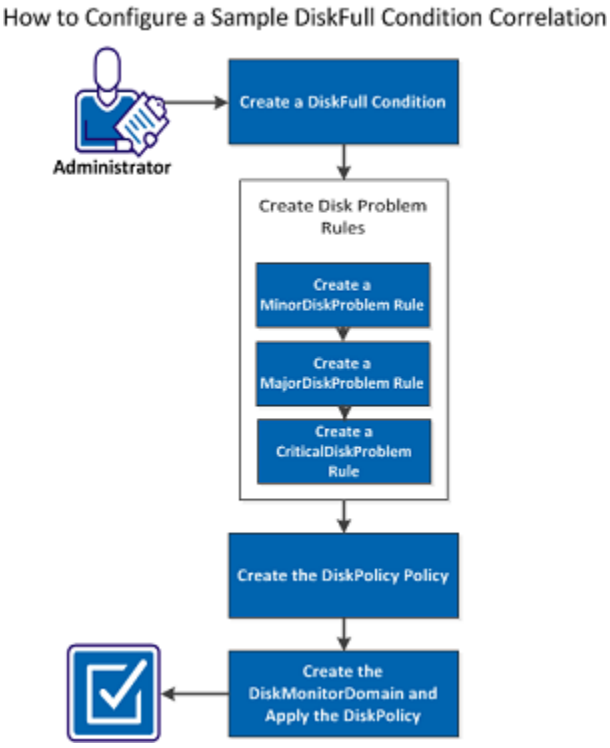
> **NOTE**
> You can create event format and alarm probable cause files. For more information, see the Event Configuration section.

How to Configure the Sample DiskFull Condition Correlation

Condition Correlation determines the root-cause alarm by selecting a criterion to identify a casual problem event. You can use the predefined components to configure a correlation. With the help of Condition Correlation components (such as Conditions, Rules, Policies and Correlation domain), you can pinpoint the root-cause alarm and symptomatic alarms.

As an administrator, you can configure the sample Disk Full Condition Correlation by creating DiskFull condition and Disk Problem rules. After creating the conditions and rules, you can create a DiskPolicy policy and apply it to a DiskMonitor Domain. DiskFull events are generated on the host devices with different values for the disk. Each time the disk problem alarms are cleared, all the existing DiskFull alarms become symptoms of the respective disk problem alarm.

The following diagram illustrates the process to configure a Disk Full Condition Correlation:



Perform the following tasks to configure the sample DiskFull Condition Correlation:

1. Create a DiskFull Condition
2. Create Disk Problem Rules
   a. Create a MinorDiskProblem Rule
   b. Create a MajorDiskProblem Rule
   c. Create a CriticalDiskProblem Rule
3. Create the DiskPolicy Policy
4. Create the DiskMonitorDomain and Apply the DiskPolicy

Create Disk Conditions

You can create disk conditions in Condition Correlation Editor. You can specify the condition name with the set event and clear event codes associated to the condition.

Follow these steps:

1. Open Condition Correlation Editor.

The Condition Correlation Editor window opens.
2. Click the Conditions tab.
   A list of predefined and user-created conditions is displayed.
3. Click Create.
   The Create Correlation Condition dialog opens.
4. Create the following disk conditions:
   – DiskFull Alarm (including Disk parameter):
     • Condition Name: DiskFull
     • Set Event Code: 0xffff0000
     • Clear Event Code: 0xffff0001

     Because a model can have multiple occurrences of these conditions, you must also add a parameter to distinguish them. The alarms are distinguished by variable binding 4. Therefore, use 4 for this parameter as well.
     • Parameter Name: Disk
     • Parameter Type: Var Bind
     • Parameter ID: 4
     • Use as discriminator: Yes
   – Minor Disk Problem:
     • Condition Name: MinorDiskProblem
     • Set Event Code: 0xffff0010
     • Clear Event Code: 0xffff0011
   – Major Disk Problem:
     • Condition Name: MajorDiskProblem
     • Set Event Code: 0xffff0020
     • Clear Event Code: 0xffff0021
   – Critical Disk Problem:
     • Condition Name: CriticalDiskProblem
     • Set Event Code: 0xffff0030
     • Clear Event Code: 0xffff0031
5. Click OK.
   Conditions are created and added to the Conditions tab.

Create Disk Problem Rules

You can create Disk Problem Rules to configure a Disk Full Condition Correlation. Minor, Major, and Critical Disk Problem rules are created in Condition Correlation Editor with specific rule criteria.

You can create the following disk problem rules:

- Create a MinorDiskProblem Rule
- Create a MajorDiskProblem Rule
- Create a CriticalDiskProblem Rule

Create a MinorDiskProblem Rule

You can create a MinorDiskProblem rule in Condition Correlation Editor. You can specify the rule name and the rule criteria for the DiskFull condition.

Follow these steps:

1. Open Condition Correlation Editor.
   The Condition Correlation Editor window opens.
2. Click the Rules tab.

A list of rules is displayed

3. Create a rule using the following properties:
   – Name: MinorDiskProblemRule
   – Symptom Condition(s):
     • Name: DiskFull
     • Type: Counts
   – Relationship: Implied Cause

     NOTE
     The MinorDiskProblem alarm is generated when the rule criteria are satisfied, and causes the rule to hide the DiskFull alarms.
   – Root Cause Condition: MinorDiskProblem
   – Root Cause Target: Select the Correlation Domain option.
4. Click Show Advanced to open the Rule Criteria panel.
5. Create the following rule criteria:
   – 'DiskFull.count GREATER THAN OR EQUAL TO 5':
     • Condition: DiskFull
     • Parameter: Condition Count
     • Operator: GREATER THAN OR EQUAL TO
     • By Value: Yes
     • Value: 5
     • Type: Integer
     • Click Insert Criterion.
   – 'DiskFull.count LESS THAN 10':
     • Condition: DiskFull
     • Parameter: Condition Count
     • Operator: LESS THAN
     • By Value: Yes
     • Value: 10
     • Type: Integer
6. Click Insert Criterion.
7. Click Create.
   The new rule is added to the Condition Correlation Editor Rules tab.

Create a MajorDiskProblem Rule

You can create a MajorDiskProblem rule for the DiskFull condition in Condition Correlation editor. After creating the rule, you can specify the rule criteria.

Follow these steps:

1. Open Condition Correlation Editor.
   The Condition Correlation Editor window opens.
2. Click the Rules tab.
   A list of rules is displayed.
3. Create a rule using the following properties:
   – Name: MajorDiskProblemRule
   – Symptom Condition(s):

    - Name: DiskFull
    - Type: Counts
  - Relationship: Implied Cause
  - Root Cause Condition: MajorDiskProblem
  - Root Cause Target: Select the Correlation Domain option.
4. Click Show Advanced to open the Rule Criteria panel.
5. Create the following rule criteria:
  - 'DiskFull.count GREATER THAN OR EQUAL TO 10':
    - Condition: DiskFull
    - Parameter: Condition Count
    - Operator: GREATER THAN OR EQUAL TO
    - By Value: Yes
    - Value: 10
    - Type: Integer
    - Click Insert Criterion.
  - 'DiskFull.count LESS THAN 15':
    - Condition: DiskFull
    - Parameter: Condition Count
    - Operator: LESS THAN
    - By Value: Yes
    - Value: 15
    - Type: Integer
6. Click Insert Criterion.
7. Click Create.
   The new rule is added to the Condition Correlation Editor Rules tab.

Create a CriticalDiskProblem Rule

You can create the CriticalDiskProblem rule after creating MinorDiskProblem and MajorDiskProblem rules in Condition Correlation Editor.

Follow these steps:

1. Open Condition Correlation Editor.
   The Condition Correlation Editor window opens.
2. Click the Rules tab.
   A list of rules is displayed
3. Create a rule using the following properties:
  - Name: CriticalDiskProblemRule
  - Symptom Condition(s):
    - Name: DiskFull
    - Type: Counts
  - Relationship: Implied Cause
  - Root Cause Condition: CriticalDiskProblem
  - Root Cause Target: Select the Correlation Domain option.
4. Click Show Advanced to open the Rule Criteria panel.
5. Create the following rule criteria:
  - 'DiskFull.count GREATER THAN OR EQUAL TO 15':

- Condition: DiskFull
- Parameter: Condition Count
- Operator: GREATER THAN OR EQUAL TO
- By Value: Yes
- Value: 15
- Type: Integer

6. Click Insert Criterion.
7. Click Create.
   The new rule is added to the list of rules in Condition Correlation Editor.

Create the DiskPolicy Policy

You can create a DiskPolicy policy in Condition Correlation Editor. After creating the policy, you can add the disk problem rules to the policy rules list.

Follow these steps:

1. Open Condition Correlation Editor.
   The Condition Correlation Editor window opens.
2. Click the Policies tab.
   The Condition Correlation Editor window displays a list of policies
3. Create a new policy using the name DiskPolicy.
4. Add the following rules to the Policy Rules list:
   – MinorDiskProblemRule
   – MajorDiskProblemRule
   – CriticalDiskProblemRule
5. Click Create.
   The new policy appears in the list of policies in Condition Correlation Editor.

Create the DiskMonitorDomain and Apply the DiskPolicy

You can create a new correlation domain to accommodate the components. You can apply the DiskPolicy from the policies list to the DiskMonitor Domain.

Follow these steps:

1. Click the Domain tab in Condition Correlation Editor.
2. Click Create.
3. Enter DiskMonitorDomain in the Domain Name text box.
4. Click the Policies tab, select DiskPolicy from the Available Policies list and move it into the Domain Policies list.
5. Click the Resources tab and then select any number of host devices as resources.
6. Click Create.
   The DiskMonitor Domain is added to the list of Domains in the Condition Correlation Editor.

Create a Clear Events Correlation

This section describes some additional functionality that you can implement on this sample DiskFull Condition Correlation.

You can clear the disk problem alarms (such as, major, minor, or critical) from OneClick. However, if you clear the disk problem alarm, all previously hidden disk full alarms reappear. Because the alarm it is correlated to is destroyed. This section describes how you can clear all of these alarms.

Multiple alarms on multiple models can exist. Therefore, the only thing you have is the clear event for one of the disk problem events (minor, major, or critical). You can perform the following tasks to create the clearing events on the correct model:

Create an Additional Parameter for the DiskFull Condition

To add a parameter to the DiskFull condition, you require one additional parameter for the DiskFull condition.

Follow these steps:

1. In the Conditions tab, in the Correlation Editor, select the DiskFull condition and click Edit.
   The Edit Correlation Condition dialog opens.
2. Click Create in the Parameters section.
   The Create Correlation Parameter dialog opens. You need to add the model where the condition (alarm) exists as shown in the following step.
3. From the Parameter Type field, select Predefined.
   The Parameter ID field shows the applicable model handle attribute: 0x129fa.
4. Click Create to add this parameter to the condition.
   This condition parameter can now be used in the clearing rule that you create to assert the clear event on the correct model.

Create an Event Rule to Identify a Cleared Disk Problem Alarm

You can create an event rule to identify when a disk problem alarm is cleared by the user. This rule lets you distinguish between instances when the correlation cleared the alarm (performs automatically when the number of alarms reaches any of the thresholds) and when a user cleared the alarm from the UI, indicating that the user knows about the problem and decides that the problem is resolved.

As you are clearing the alarms from the UI, no direct event code (for example, 0xffff0021) is used. Instead, you can use one of the alarm status events. For example, 0x10706: user has cleared an alarm. In this event, you can find Probable Cause Code of the cleared alarm, in varbind 3. You can use the Probable Cause Code to generate a new event and can use it as a condition to start the clear correlation.

You can create an event rule to generate a event, disk problem alarm has been user-cleared. The 0x10706 event is mapped (by default) in the following file:

```
<$SPECROOT>/SS/CsVendor/Cabletron/EventDisp
```

The syntax to add an event action is as follows:

```
0x00010706 E 50 R CA.EventCondition, \
   " { v 3 } == { H 0xffff0010 } ", 0xffff0100, \
   " { v 3 } == { H 0xffff0020 } ", 0xffff0100, \
   " { v 3 } == { H 0xffff0030 } ", 0xffff0100
```

Log and Add an Event to Clear the DiskFull Alarms

Optionally, you can log the event in the custom EventDisp file using the following syntax:

```
0xffff0100 E 50
```

You can add an event to clear the disk full alarms, regardless of their discriminator value. Use the following clear all (‘A’) alarm clear flag syntax:

```
0xffff0002 E 50 C 0xffff0000, A
```

This event lets you clear all disk full alarms on a model, even if you do not know the values for their discriminator attributes.

Create the Conditions Required for the Clear Correlation

Reload the EventDisp files so that you can set up the clear correlation. The following procedure describes the steps.

Follow these steps:

1. Open Condition Correlation Editor.
   The Condition Correlation Editor window opens.
2. Click the Conditions tab.
   A list of conditions is displayed.
3. Click Create.
   The Create Correlation Condition dialog opens.
4. Create the following condition to start the clear correlation:
   – DiskProblemAlarmUserCleared:
     - Condition Name: DiskProblemAlarmUserCleared
     - Set Event Code: 0xffff0100
     - Clear Event Code: 0xffff0100

       NOTE
       The DiskProblemAlarmUserCleared condition is no longer required after it starts the clear correlation. You can clear this condition after the clear correlation has completed. You can use the same clear event as the set event to generate the condition. This condition is cleared after the completion of clear correlation and is therefore temporary.
5. Create the following condition to clear the DiskFull alarms:
   – DiskFullAlarmClear:
     - Condition Name: DiskProblemAlarmUserCleared
     - Set Event Code: 0xffff0002
       (The Set Event Code indicates that this event can be generated when the condition is generated by an implied rule.)
     - Clear Event Code**:** 0xffff0002
       (The Clear Event Code indicates that the condition is self-clearing, as does the condition in Step 1.)

   DiskProblemAlarmUserCleared condition is created.

Create a Rule to Clear DiskFull Alarms

You can create a rule that clears all DiskFull alarms when a user clears one of the disk problem alarms.

Follow these steps:

1. Open Condition Correlation Editor.
   The Condition Correlation Editor window opens.
2. Click the Rules tab.
   A list of rules is displayed.
3. Create the following rule:
   – Name: DiskFullUserClearRule
   – Symptom Condition(s):

    - DiskProblemAlarmUserCleared
    - DiskFull
  - Type: Exists
  - Relationship: Implies
  - Root Cause Condition: DiskFullAlarmClear
  - Root Cause Target: DiskFull.Model

    NOTE
    This rule ensures that the clear event is generated on each model where a DiskFull alarm exists, enabling the alarm to be cleared.
4. Save the rule.
5. Add the new rule to the “DiskPolicy” policy.
   This rule triggers when any one of the three disk problem alarms is cleared by the user, generating event 0xffff0100.

The setup is complete. Anytime the user clears any one of the three disk problem alarms (minor, major, or critical), all individual DiskFull alarms are cleared. The condition is paired with each DiskFull alarm, and generates the DiskFullAlarmClear condition on the model of the DiskFull alarm. Thus, all of the DiskFull alarms are cleared.

WAN Link Failure Example

The WAN Link Failure example describes the process that Condition Correlation uses to pinpoint the root-cause alarm and symptomatic alarms among a barrage of alarms that are generated for different resources, as a result of a WAN link failure.

WAN Link Scenario

In many WANs, primary connections have a backup. The backup connection typically provides less bandwidth than the primary connection. In this example, a 384K Frame Relay link is backed up by a 128K ISDN link. Also, a DX NetOps Spectrum Service Performance Manager (SPM) test is measuring latency across the WAN link.

When the Frame Relay link goes down, the ISDN link takes over and the SPM test exceeds the latency threshold due to the reduced bandwidth. DX NetOps Spectrum generates two alarms (Critical Alarm - Frame Relay Link Down occurs on the Frame Relay link model; Minor Alarm - SPM Test Exceeded Threshold occurs on the SPM Test model) and one event (ISDN Backup Active occurs on the device).

WAN Link Correlation Strategy

It may not be apparent to network management personnel that all the three conditions are related and they are likely to focus their effort on the critical alarm even though there are other important alarms in the infrastructure.

You can apply the following conditions to the domain including the resources that can be compromised by a primary WAN link failure:

- The two alarms and the ISDN event can be correlated to generate a new primary link down, reduced bandwidth condition that produces a major alarm because the WAN is still working, but with a decreased performance.
- The failed Frame Relay link can be correlated with the Dialup Link Active event and can imply that the primary WAN link is down with reduced bandwidth, if the backup link bandwidth is less than the primary link bandwidth.
- The SPM test can be correlated with the primary WAN link down, reduced bandwidth condition with a rule that stipulates SPM Test Threshold Exceeded is caused by primary WAN link down, reduced bandwidth condition.

This correlation system produces the following alarm and event information: the single major alarm for the WAN link being down. There is also an SPM test threshold exceeded alarm and the ISDN backup active event, but these alarms are hidden under the single major alarm. This lets troubleshooters focus their efforts on the most important alarms. A second rule could be created to produce a minor alarm if the active backup link has the same bandwidth as the failed primary link.

WAN Link Failure Configurations

You can configure the correlation system using the following process:

- In DX NetOps Spectrum, a new “Primary WAN Link Down, Reduced Bandwidth” alarm is created. A set event and clear event for the new alarm is required.
  You can create alarms and can edit event configuration files. For more information, see the Event Configuration section.
- Create the following conditions:
  – A “Primary_WAN_Link_Down_Reduced_Bandwidth” condition using the set and clear event codes from the “Primary WAN Link Down, Reduced Bandwidth” alarm.
  – A “Dialup_Link_Active” condition using set event 0x022ffff6, “Dialup link has been activated,” and clear event 0x022ffffc, “Dialup link is inactive.” This condition is not linked to a DX NetOps Spectrum alarm. However, it infers that the backup, or secondary, link is up and running.
- Create the following rules:
  – A “PrimaryFrameRelay_Red -> LinkDown” rule states that if the “Primary_WAN_Link_Down_Reduced_Bandwidth” and “Dialup_Link_Active” conditions occur, then the implied cause is the “Primary_WAN_Link_Down_Reduced_Bandwidth” condition and the critical (red) Frame Relay Link Down alarm is suppressed by the “Primary WAN Down, Decreased Bandwidth” alarm (orange).
  – An “SPMLatencyThreshold_yellow -> Violated” rule states that the SPM latency threshold violation is caused by the “Primary WAN Link Down, Reduced Bandwidth” condition, and suppressed yellow SPM latency threshold violation alarms.
- A “WAN_Link_Failure” policy is created that includes the “PrimaryFrameRelay_Red -> LinkDown” and “SPMLatencyThreshold_yellow -> Violated” rules.
- A “WAN_Primary_Backup_Links” domain is created. It includes the primary WAN link interfaces, the backup link, and any SPM tests that can be impacted by the lower bandwidth of the backup. The “WAN_Link_Failure” policy is applied to the domain.

Special Topics

This section discusses special topics that are related to Condition Correlation capabilities and implementation.

Condition Correlation and Fault Isolation

If a managed device stops responding to polls, the DX NetOps Spectrum fault isolation algorithm determines whether to create a critical alarm for the device or suppress its alarm state. The unreachable device is the root cause of the alarm. Condition Correlation supports setting up a correlation between a device in the Contact Lost condition and some other condition in your environment. For example, DX NetOps Spectrum receives a trap from a BGP router that is reporting a lost session with a peer router. If the peer router is already in the Contact Lost state in DX NetOps Spectrum, the BGPLost Session alarm can be a symptom of the Contact Lost alarm on the peer router model.

If the peer router in the Contact Lost state has a critical Device Has Stopped Responding to Polls alarm, the correlation is trivial. If the fault state of the peer router is suppressed by the DX NetOps Spectrum fault isolation algorithm, no root cause alarm exists on this model.

You cannot correlate the actual root cause alarm with the Peer Lost alarm without special consideration from Condition Correlation. However, in Condition Correlation, the Device Contact Lost condition receives special consideration. This condition remains in force whenever a device is in the Contact Lost state, regardless of whether the device model is suppressed or has an alarm. If the device model in question is suppressed, the correlation engine locates the isolated alarm and uses it as the root cause for any correlation rules.

About Transfer Rules

DX NetOps Spectrum recognizes a Model Active condition that can be used in a correlation rule. The Model Active condition is used when a port model is added or removed from a correlation domain. This condition can be used for special rules, such as transferring alarms from devices to ports, because the Model Active condition is present for each correlated port. This usage eliminates the requirement for the port to have an alarm to participate in a correlation. Attributes on the condition can then be used to create a rule that transfers alarms to the correct port. The correct port can be identified by using the following parameters from the Model Active condition:

- Component OID of the port.
- Model handle of the parent device of the port.
- Model type of the port.

Condition Correlation provides a default transfer rule: transfer PVCL alarm from the device to interface. It reacts to the PVCL failure condition (alarm 0x210048 - PVCLs Failure Notification) on the device model by extracting the Interface ID of the affected port from the PVCL failure condition. Then it locates the port model by comparing the Interface ID from the PVCL failure condition with the Component_OID parameter of the Model Active condition on the port. A new PVCL failure (0x210c0c - PVCLs Failure Notification) alarm is created on the port. This failure alarm is identified by the Model parameter of the Model Active condition. The PVCLs Failure Notification alarm for the device is made a symptom of the new PVCLs Failure Notification alarm on the port.

Advanced Correlations and Data Type Comparisons

Verify the following information before you configure advanced correlations, which involve comparisons between different data types:

- Condition Correlation converts the right-hand value to the left. This conversion can be problematic; it is unlikely that a real number conversion produces the same text string that you have for a comparison.
- SNMP represents both real text strings (such as, messages, and information), and octet strings (such as, Mac addresses) with no indication of the actual usage. Therefore, in some cases it is impossible for the automatic conversion process to convert to the actual type which you need for a comparison. Because, Condition Correlation does not have the meta-information.
- Condition Correlation does not attempt to convert list types.

REST Examples for Correlation Testing

This section contains examples to help you test and debug custom correlation systems that you have created.

Two examples of simulated events are provided to help you test correlations. In the following XML examples, replace the [model_handle] field with the actual model handle of the target model. You can perform this task through the DX NetOps Spectrum CLI (using show models) or through the attribute browser, by reading the value of the Model_Handle attribute with attribute ID of 0x129fa.

RESTful Web Services XML Example - No Event Variables

The following XML example lets you generate an event for testing purposes without event variables (for example, Event00010220, the model has gone into Maintenance Mode). Use this event template as a framework to develop simulation and testing tools for Condition Correlation testing.

```
<?xml version="1.0" encoding="UTF-8"?>
<rs:event-request throttlesize="10"
xmlns:rs="http://www.ca.com/spectrum/restful/schema/request"
```

```
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.ca.com/spectrum/restful/schema/request ../../../xsd/Request.xsd">
<rs:event>
<!-- target model of event -->
<rs:target-models>
<rs:model mh="0x100000"/>
</rs:target-models>
<!-- event ID -->
<rs:event-type id="0x10220"/>
</rs:event>
</rs:event-request>
```

RESTful Web Services XML Example - with Event Variables

The following sample XML generates an artificial event for testing with event variables. Use this event template as a framework to develop simulation and testing tools for Condition Correlation testing:

```
$SPECROOT/RestfulExamples/xml/Events/CreateEventByModelHandleList.xml
<?xml version="1.0" encoding="UTF-8"?>
<!--
This sample event request will create an event of type
0x10f06 (generates a High Memory Utilization alarm).
->
<rs:event-request throttlesize="10"
xmlns:rs="http://www.ca.com/spectrum/restful/schema/request"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.ca.com/spectrum/restful/schema/request ../../../xsd/Request.xsd">
<rs:event>
<!-- target model of event -->
<rs:target-models>
<rs:model mh="0x100000"/>
</rs:target-models>
<!-- event ID -->
<rs:event-type id="0x10f06"/>
<!-- attributes/varbinds -->
<rs:varbind id="0">75</rs:varbind>
<rs:varbind id="1">99</rs:varbind>
<rs:varbind id="3">mem_instance</rs:varbind>
<rs:varbind id="5">ModelName</rs:varbind>
</rs:event>
</rs:event-request>
```

NOTE
Replace the 0x100000 in the modelmh field with the correct model handle (while retaining the double quotes).

NOTE
You can also edit the attribute varbinds to reflect the number of varbinds (or event variables), their index, and value. This example specifies the following four varbinds:

- Event Variable 0: Value = 75 (memory threshold)
- Event Variable 1: Value = 99 (actual memory utilization)
- Event Variable 3: Value = mem_instance (Memory Instance)
- Event Variable 5: Value = name (Memory Instance Name)

Other events can have a different number of varbinds. But this XML example can be edited as appropriate to have the correct number of varbinds.

Configure WizTools RESTClient

Configure the WizTools REST client to work in a DX NetOps Spectrum environment. The following steps illustrate how the REST client can be configured before sending the XML request to the OneClick web server.

> **NOTE**
> Any REST client can be used to interact with the DX NetOps Spectrum web services. However, in this document, we describe how to configure a specific REST client application. We selected this particular client for its simplicity and ease of use.

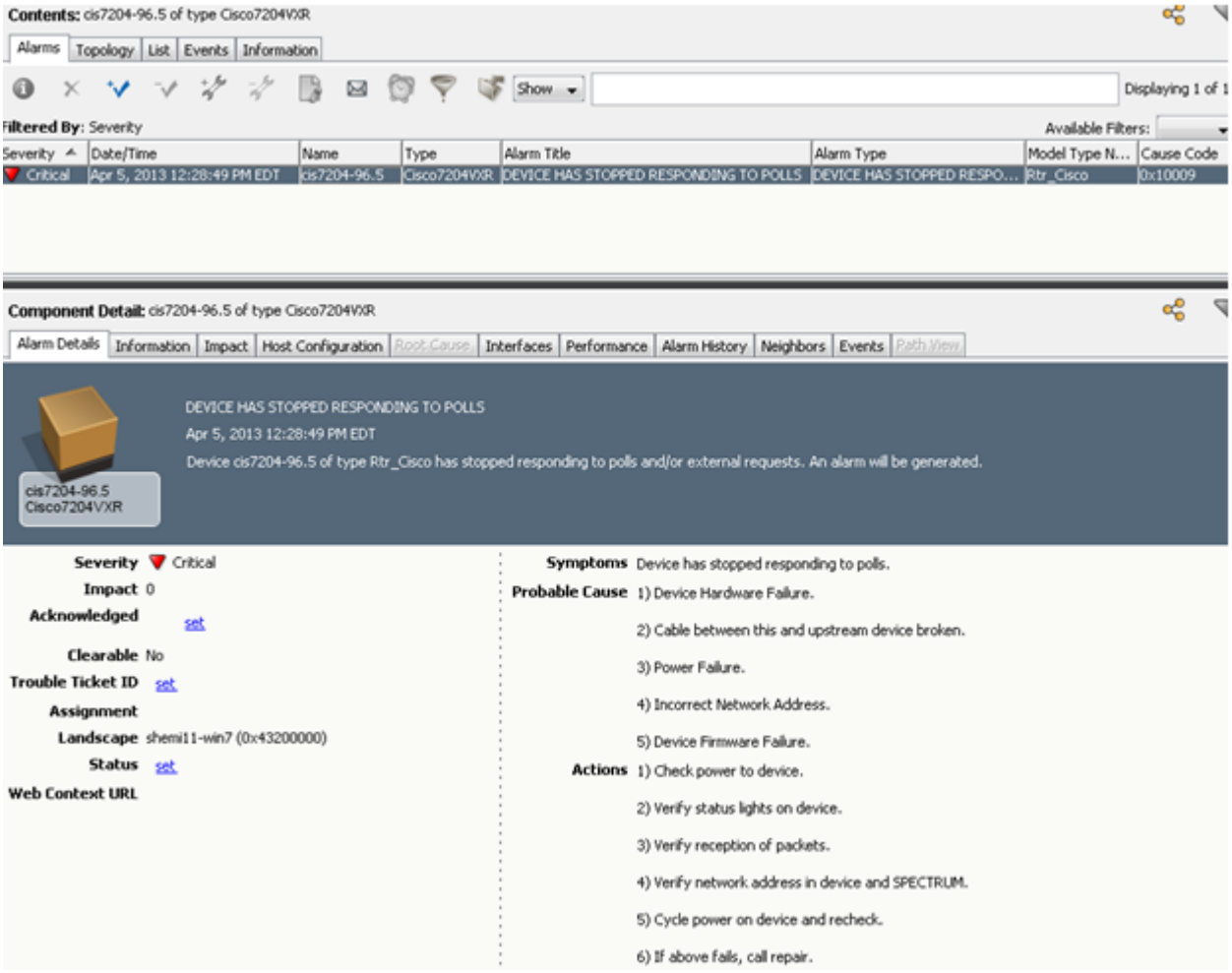
Follow these steps:

1. Add the following string to the URL dialog:
   ```
   http://OneClick web server hostname/spectrum/restful/events
   ```
2. Click the Method tab and select Post.
3. Click the Body tab and select String Body from the list.
   The Body Content-type dialog opens.
4. Supply 'application/xml' for the Content Type.
5. Leave the Charset at 'UTF-8'.
6. Paste the XML contents into the String Body field.
7. Click the Auth tab and select BASIC from the list.
8. Supply values for the following parameters:
   - **Host**
     Specifies the hostname of the OneClick web server.
   - **Realm**
     Identifies the type of authentication.
     Note: Leave this field blank.
   - **Username**
     Indicates the username of the operator who is authorized to access OneClick.
   - **Password**
     Indicates the password.
9. Click Go (>>).
   The WizTools REST client is configured.

Create and View Simulated Alarms An Example

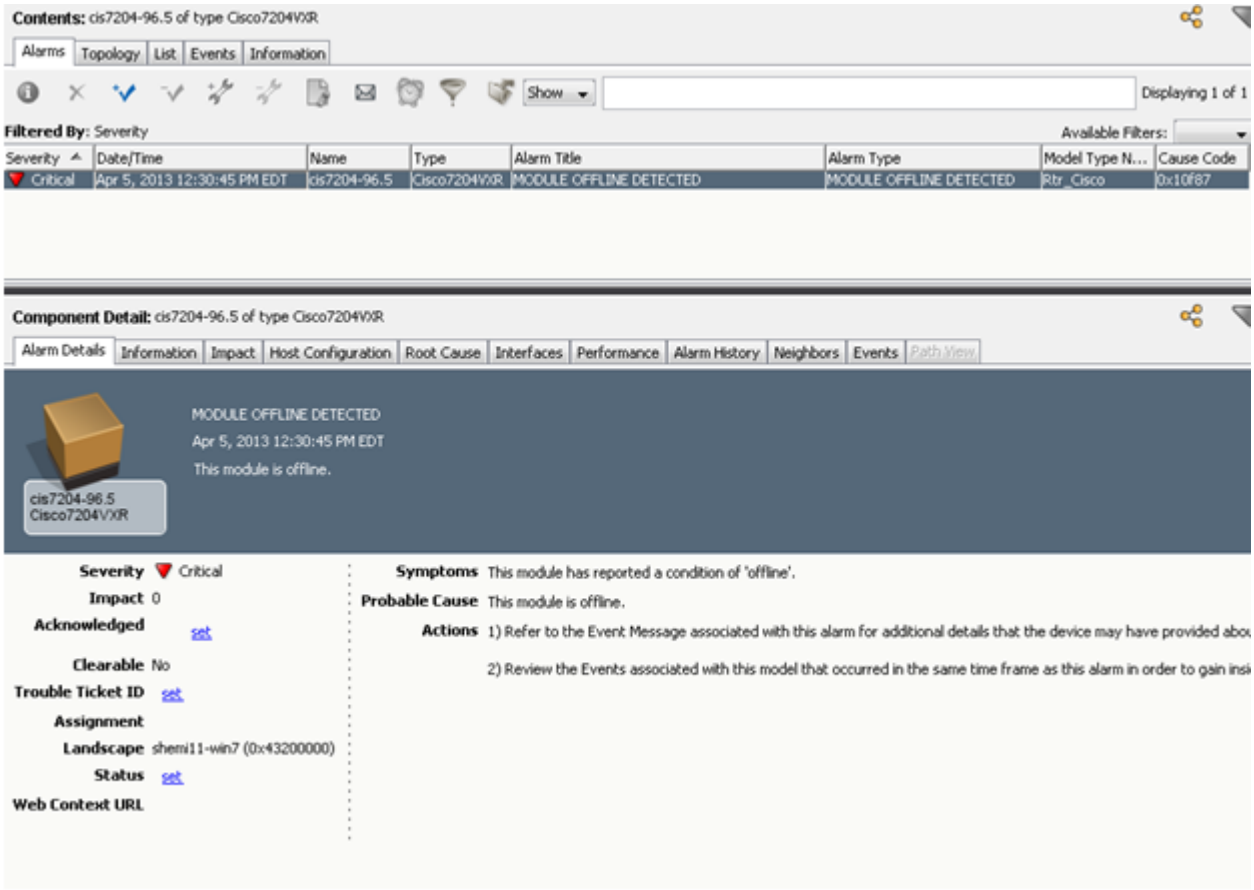
The simulated alarms that you create using the DX NetOps Spectrum CLI or web services can be viewed in the model Alarm View. Create the appropriate events on a managed entity, where DX NetOps Spectrum simulates that an alarm condition exists (although no actual condition exists). DX NetOps Spectrum treats the simulated alarm as if it were an actual alarm on the managed entity. The same intelligence is executed, and the same alarms are displayed in OneClick.

For example, the following image illustrates a simulated Device Contact Lost alarm:



The ModuleOffline event that is created causes the ContactLost alarm to be hidden.

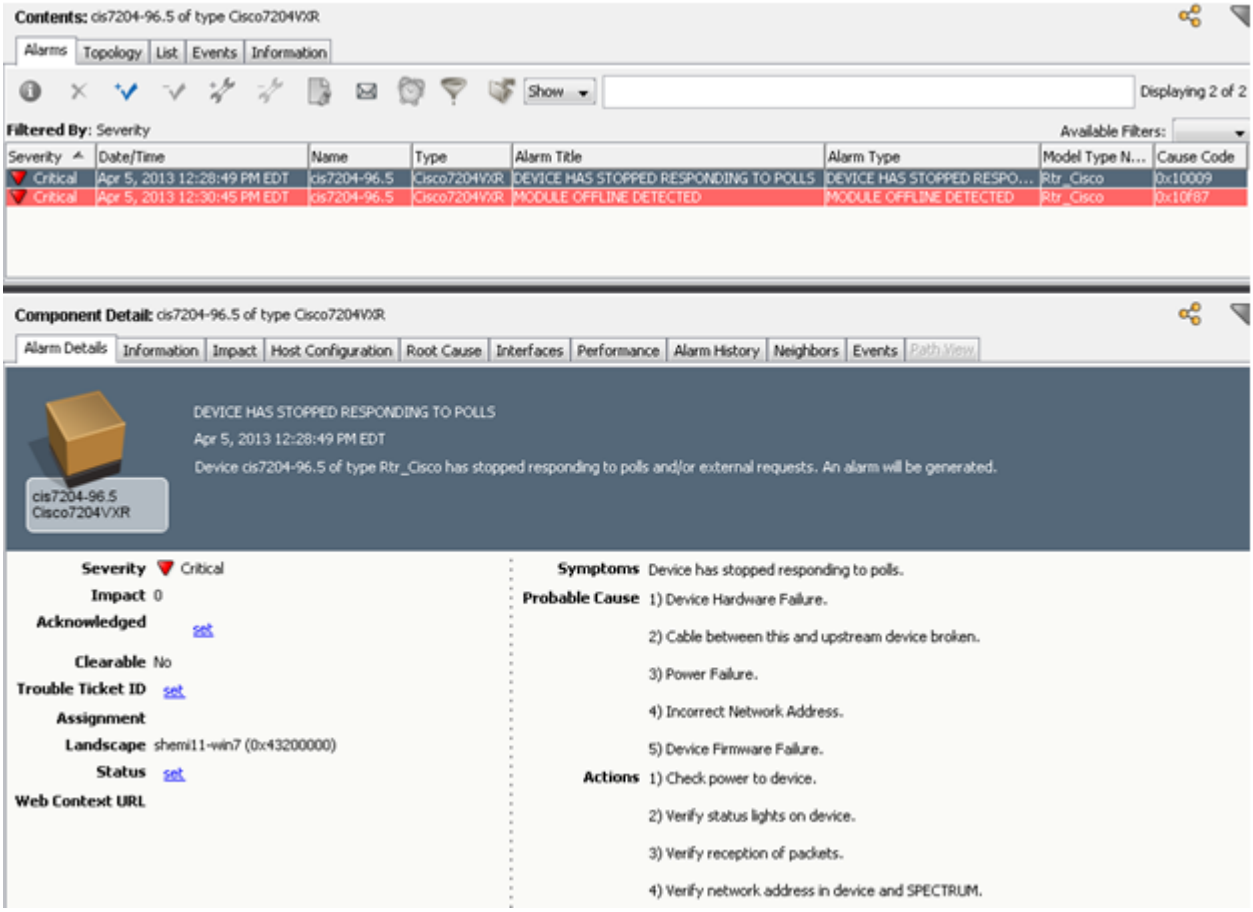
The following image illustrates the ModuleOffline event:



If the Alarm View alarm filter state is changed from its default setting to Show Symptoms, the following alarms are displayed.

- Symptom Alarm: Contact Lost
- Root-Cause Alarm: ModuleOffline

The following image illustrates the Symptom and Root-Cause alarms:



You can clear these alarms by creating the following Clear events:

- ModuleOffline Clear: Event 0x00010f89
- ContactLost Clear: Event 0x00010d30

Device Management Reference

This section introduces DX NetOps Spectrum Device Management documentation for the following devices (presented alphabetically):

AM Communications

This section introduces the DX NetOps Spectrum Device Management documentation for the AM Communications Integration.

Supported Devices

AM Communications develops network management products for non-SNMP broadband components. They monitor RF (Radio Frequency), HFC (Hybrid FIber Coax) components. The NetMentor software package of Cheetah, with optional SNMP Agent Module, converts their proprietary events into SNMPv1/v2 alarms and traps. This management module uses the Generic Southbound Application Gateway integration to provide a place for trap reception and event creation.

This management module supports the Omni2000 Proxy Agent. Omni2000 Proxy Agent is the HFC component monitoring solution of AM Communication.

The DX NetOps Spectrum Model

No specific AM Communications model types are created. The Southbound Gateway provides the model types EventAdmin and EventModel. These model types are used to manage the information that the Omni2000 Proxy Agent sends to DX NetOps Spectrum.

EventAdmin is a container model type that is used to represent the Omni2000 Proxy Agent. EventModels represent unique sources of trap information that the Omni2000 Proxy Agent passes to DX NetOps Spectrum. EventModels are automatically placed in a topology view that you can access by drilling down from the EventAdmin model. These icons do not show any connectivity with one another because they represent an event source, not necessarily a physical device or component.

When the EventAdmin model receives a trap from the Omni2000 Proxy Agent, it maps the trap to a DX NetOps Spectrum event. It then sends the event to the appropriate EventModel for processing. If an EventModel that represents the unique source of trap information does not exist, it is automatically created.

The SouthBound Gateway Toolkit contains instructions for creating an EventAdmin model. Use the EventAdmin model to represent the AM Communications management application.

When you create this model, select a Manager Name of Omni2000.

Traps, Events, and Alarms

This section describes how the AM Communication Integration sends the traps. It also describes how the EventAdmin and EventModels process and manage these traps.

As the Omni2000 Proxy Agent sends traps to DX NetOps Spectrum, the EventAdmin model receives them, and maps them to a DX NetOps Spectrum event. These events are sent to an EventModel that represents the trap source. The value of the NEModelNumber variable binding that is sent in the trap identifies the trap source. This variable binding is from the AMC-MIB**.** If an EventModel representing the trap source does not exist, it automatically is created.

When the EventModel receives the event, it is processed and can be used to create or clear an alarm. The following table displays how each trap is mapped to a DX NetOps Spectrum event and how the event is processed.

trap	alarm	event code	description
NewNEFound		0x3eb0001	HFC Proxy detected new Network Element.
Communicatio-nsStatus		0x3eb0002	HFC Proxy lost or restored communication with Network Element.
Configuration Change	orange	ox3eb0003	Configuration of a single variable of any type was changed (via any interface).
StatusChange		0x3eb0004	An active alarm was cleared.
Alarm	orange	0x3eb0005	An ALARM is detected by a proxy agent.
ToBeSendQue-ueOverflow	orange	0x3eb0006	SNMP agent's TrapToBeSendQueue is full.
NewNELost	orange	0x3eb0007	HFC Proxy Detected New Network Element Lost.

Ceterus Universal

This section describes common deployment scenarios for the Ceterus Universal Transport System devices and how to model them in DX NetOps Spectrum.

Trap Processing

You can configure the Remote Ceterus device to forward traps through its EOC channel to the Local device. In this configuration, the Local device acts as a gateway and forwards these traps. For more information about this feature, see the Ceterus documentation.

You can also configure the Remote device with an SNMP target IP address. In this configuration, the device sends traps through its management port.

If both of these capabilities are configured simultaneously, DX NetOps Spectrum receives duplicate traps. The Ceterus management module is designed to handle this case. It evaluates the incoming Ceterus traps and asserts those traps on the most appropriate DX NetOps Spectrum device model. The management module selects the appropriate model by comparing the Ceterus device community string in the trap with the value of the device sysName.

> **WARNING**
> DX NetOps Spectrum relies on the community name of the device to make this determination. As a result, the community name and the sysName must be synchronized. Polling of sysName occurs every 5 minutes by default. Changing the TID can affect the handling of the trap until sysName has been properly updated through a poll. When an administrator changes the TID (sysName) on a given Ceterus device from “Device A” to “Device B,” the device sends traps to that model. In this case, DX NetOps Spectrum can no longer process the traps. Trap processing does not recommence until the sysName is updated (up to a maximum of 5 minutes by default).

Cheetah Gateway

This section describes DX NetOps Spectrum support for monitoring Cheetah™ network management products.

Supported Devices

Cheetah™ products, including CheetahNet™ (formerly NetMentor™) are network management products for non-SNMP broadband components. They monitor RF (Radio Frequency), and HFC (Hybrid Fiber Coax) components. The CheetahNet/NetMentor software package, with an optional SNMP Agent Module, converts their proprietary events into SNMPv1/v2 alarms and traps. This management module uses the DX NetOps Spectrum Southbound Gateway integration to enable traps to be received and events to be created in DX NetOps Spectrum.

This management module provides an integration between the CheetahNet/NetMentor management application, including the SNMP agent module, and DX NetOps Spectrum. This integration can report events on the following types of HFC devices:

- Power Supply
- Amplifier
- Line Monitor
- Test Point
- Fiber Node
- HEFiber

The DX NetOps Spectrum Model

No specific Cheetah model types are created. The Southbound Gateway provides the EventAdmin and EventModel model types. These model types are used to manage the information that NetMentor sends to DX NetOps Spectrum.

The EventAdmin is a container model type that is used to represent the NetMentor management application. EventModels represent unique sources of trap information that the CheetahNet/NetMentor application passes to DX NetOps Spectrum. EventModels are automatically placed in a topology view that can be accessed by drilling down from the EventAdmin model. These icons do not show any connectivity with one another because they represent an event source, not necessarily a physical device or component.

When the EventAdmin model receives a trap from the CheetahNet/NetMentor application, it maps the trap to a DX NetOps Spectrum event. The EventAdmin also sends the event to the appropriate EventModel for processing. If an EventModel that represents the unique source of trap information does not exist, it is automatically created.

Creating the EventAdmin Model

The SouthBound Gateway Toolkit section contains instructions for creating an EventAdmin Model. Use the EventAdmin model to represent the CheetahNet/NetMentor management application. When you create this model, select a Manager Name of NetMentor.

Traps, Events, and Alarms

This section describes how the EventAdmin and EventModel process and manage traps that are sent by the CheetahNet/NetMentor integration.

As CheetahNet/NetMentor sends traps to DX NetOps Spectrum, the EventAdmin model receives these traps and maps them to a DX NetOps Spectrum event. These events are sent to an EventModel that represents the trap source. The value of the **CNAlarmResource** and the **CNAlarmSubResource** variable bindings that are sent in the trap identify the trap source. Each of these variable bindings is from the CNAlarmsMib (CheetahNet Alarms MIB)**.** If an EventModel representing the trap source does not exist, it is automatically created.

When the EventModel receives the event, it is processed and can be used to create or clear an alarm. The following table describes how each trap is mapped to a DX NetOps Spectrum event and how the event is processed.

Trap OID	Trap Name	Event Generated	Alarm Generated or Cleared	Alarm Severity
1.3.6.1.4.1.1283.10.6.1	Device added	0x3e00001	NA	NA
1.3.6.1.4.1.1283.10.6.2	Device deleted	0x3e00002	NA	NA
1.3.6.1.4.1.1283.10.6.3	Configuration changed	0x3e00003	0x3e00003	Orange
1.3.6.1.4.1.1283.10.6.4	Clear Alarm	0x3e00004	Clears 0x3e00003, 0x3e00005, 0x3e00006, 0x3e00007, 0x3e00008	NA
1.3.6.1.4.1.1283.10.6.5	Warning alarm	0x3e00005		Yellow
1.3.6.1.4.1.1283.10.6.6	Minor alarm	0x3e00006		Yellow
1.3.6.1.4.1.1283.10.6.7	Major alarm	0x3e00007		Orange
1.3.6.1.4.1.1283.10.6.8	Critical alarm	0x3e00008		Red

HP BladeSystem c-Class

This section describes DX NetOps Spectrum support for monitoring the Hewlett-Packard (HP) BladeSystem c-Class device family.

Overview

Support for the HP BladeSystem c-Class device family is available in DX NetOps Spectrum with an enhanced certification. The top-level management uses the model of the HP BladeSystem Onboard Administrator (OA). This device family is modeled and represented in the topology with a OneClick icon representing a chassis.

DX NetOps Spectrum> chassis device management includes the following features:

- Support for the C7000 and C3000 chassis types.
- OA support, represented in DX NetOps Spectrum with a unique model type and chassis icon.
- Automatic blade modeling. After OA modeling, a nontopological module model is created for each occupied chassis slot. These models represent the hardware level view of an occupied slot.
- Automatic chassis identification of previously modeled device models, or managed devices, either pingable or SNMP capable, that are running on a blade.
- Enhanced Interface tab to show the hierarchy view of blades and interfaces for a given chassis. Chassis, managed devices, module models, and interfaces each have a unique icon in the hierarchy.
- Managed devices can be manually associated (or disassociated) with their chassis using a right-click menu option.
- Jump-to navigation from a managed device model to its chassis.
- Jump-to navigation from a module model to its managed device if it exists.
- Support of several chassis-based OneClick views.
- Chassis-based Locater searches.
- Enhanced Fault Isolation capabilities ensure that a single alarm is generated on a chassis-wide failure, eliminating a multiple alarm scenario.

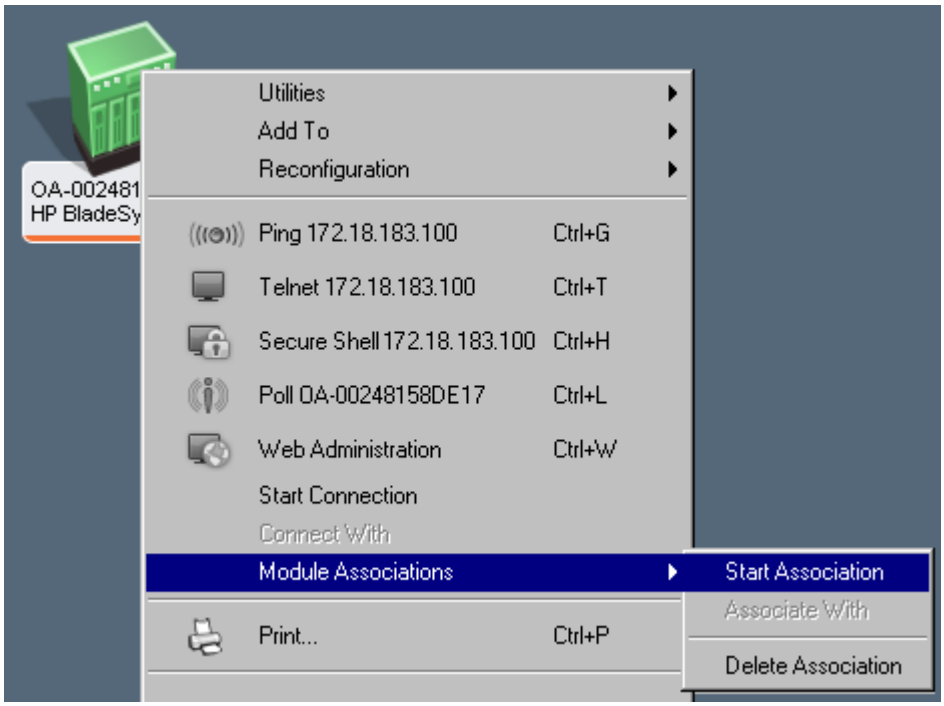
Configuration

Analysis of the chassis modeling environment occurs, by default, every 5 minutes. You can change the polling discovery interval for server and interconnect blades by modifying the *configInterval* attribute. This attribute is located on separate application models that are associated with the HPBladeOnboardAdmin model. For server blades, the relevant application model is HPServerBladeApp. For interconnect blades, the relevant application model is HPNetworkBladeApp.

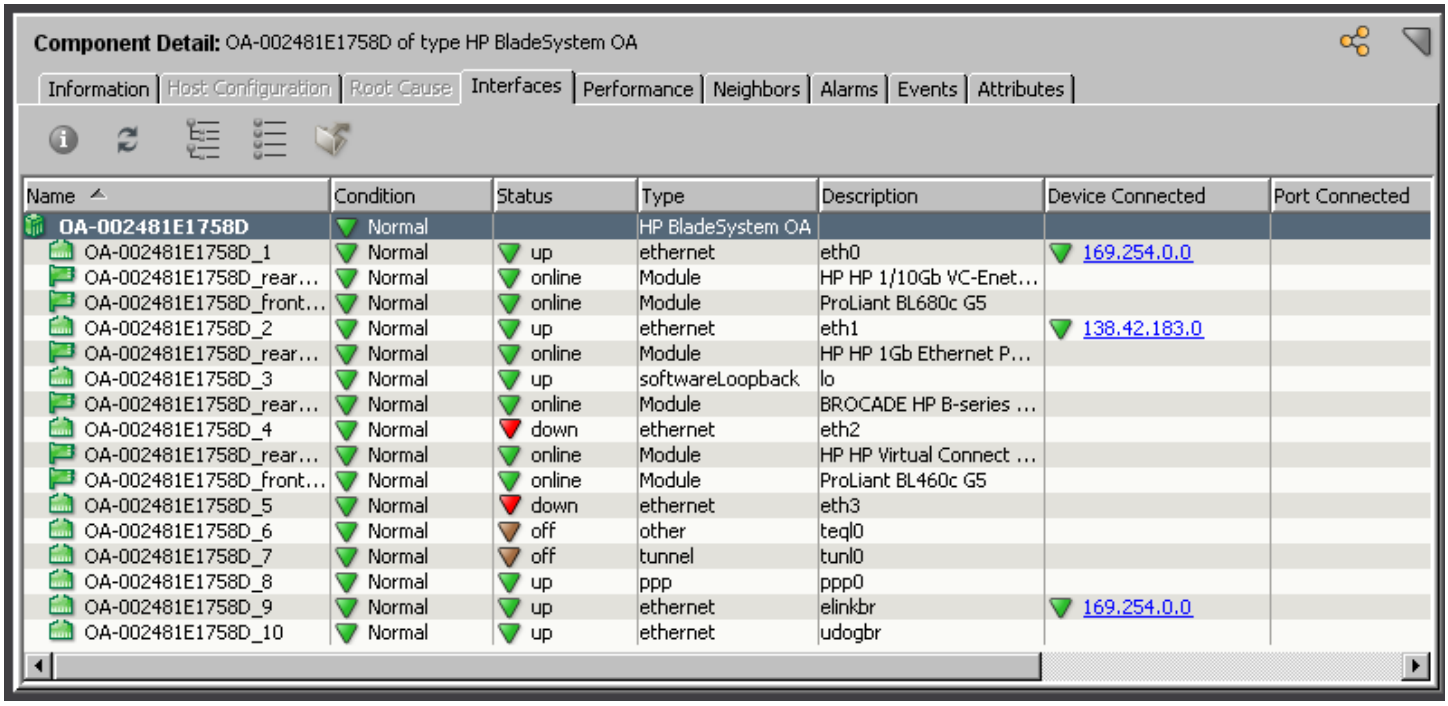
Use the 'By Device IP Address' Locater tab search under Application Models to locate and select the relevant application model. You can modify the *configInterval* attribute using the Attributes tab in the Component Detail panel of OneClick.

Managing Module Associations

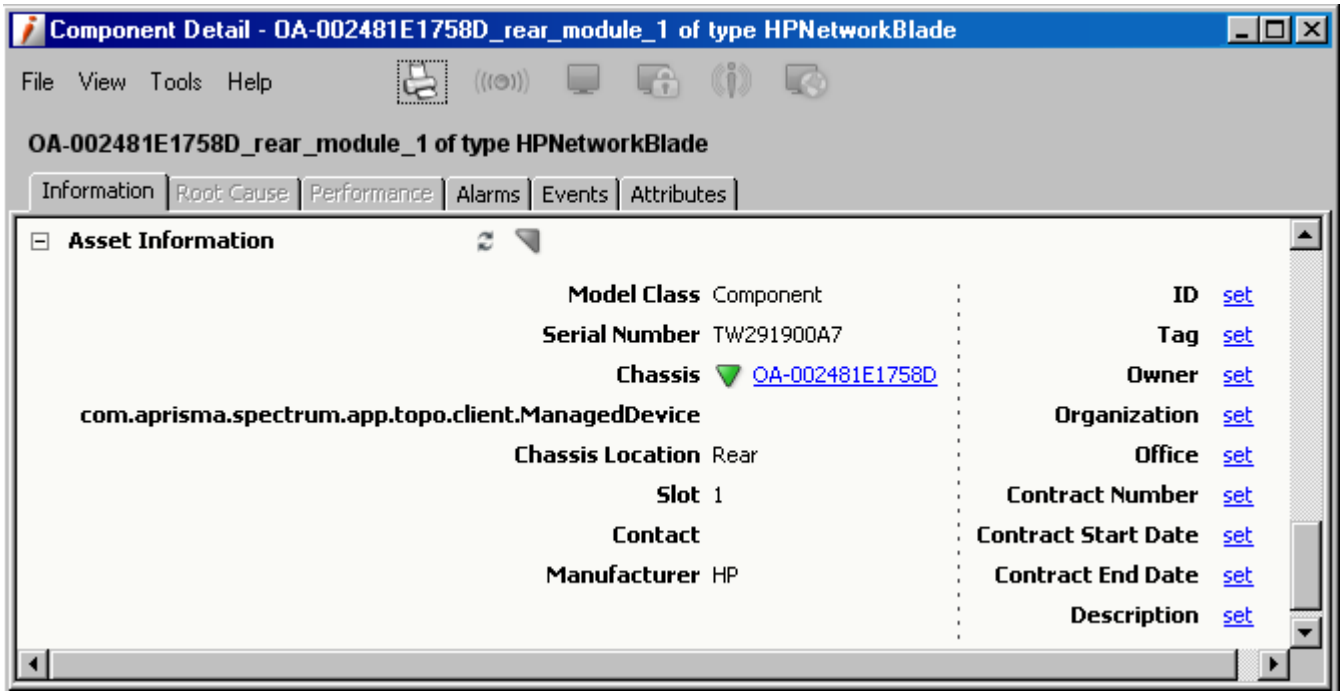
Modeling the OA initiates automated module modeling and creates associations between the modules and the chassis. Existing managed device models that can be identified through a serial number are automatically associated with the chassis. HP Insight Manager Agents do provide the required serial number; they are the recommended configuration. Otherwise, use manual association through the 'Module Associations' menu option. Subsequent Module Association menu options let you manage your association with options such as 'Start Association', 'Associate With', or 'Delete Association'.



You can view the contained modules and their associated interfaces through the OA Interfaces tab. Supported columns provide the chassis location (front or rear), slot number, module type, and description. The module icon helps you identify the type of hardware.

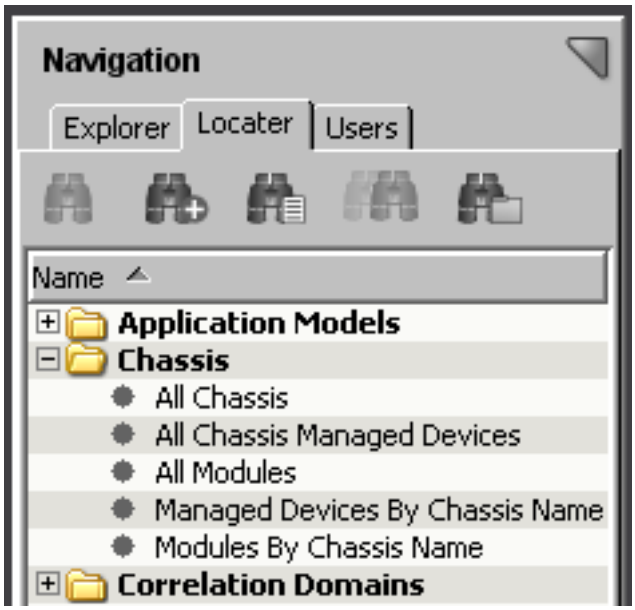


From the perspective of a module model, you can identify the parent chassis using the Chassis navigation link in the Asset Information OneClick view. You can also identify an associated managed device, if one exists, using the Managed Device link in the same view.



Locating Chassis

From the Locater tab of the Navigation panel, you can now see the following Chassis search menu options. This feature assists you in changing the polling discovery interval.

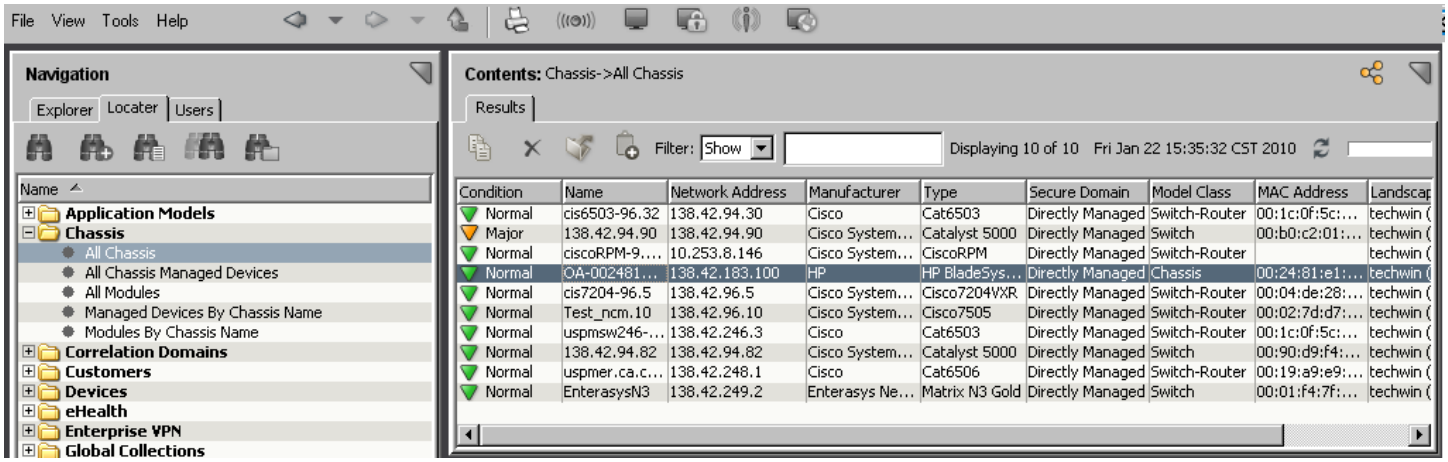


- **All Chassis**
  Displays all chassis models (HP OA model, for example)
- **All Chassis Managed Devices**

Displays all device models that are managed through DX NetOps Spectrum that are running on a blade. This search only includes pingable or SNMP-capable device models. The module models that are created for every occupied slot in a chassis are not included.

- **All Modules**
  Displays all module models, one for every occupied slot in a chassis. The search does not include the managed devices (SNMP- or ICMP-capable). They represent the hardware-level view of an occupied slot.
- **Managed Devices by Chassis Name**
  Displays all device models that DX NetOps Spectrum manages and that are running on a blade on a specified chassis. A subsequent window lets you enter the specific chassis name whose associated devices you want to view.
- **Modules by Chassis Name**
  Displays all module models for the specified chassis. A subsequent window lets you enter the specific chassis name whose associated modules you want to view.

As an example, select the 'All Chassis' chassis search option. The following results appear in the Contents panel:



Juniper M Series

This section describes the Redundant Component Monitoring intelligence available for support of JnprRedundRtr (M20, M40e, and M160) routers in DX NetOps Spectrum.

Redundant Component Monitoring Intelligence

Juniper M20, M40e, and M160 routers support the Redundant Component Architecture. Redundant components include those pieces of hardware that are necessary for proper routing functionality. The following specific components for these routers are passive monitoring and active monitoring.

> **NOTE**
> All Juniper M Series routers can be modeled, without this functionality, by type as a JNPR_Mxxx. This feature provides basic modeling functionality as is described for the JNPR_Mxxx model type.

When either Passive Monitoring or Active Monitoring is invoked, they check for status changes in each type of redundant component. The redundant components of Juniper M Series routers differ based on the following router models:

- **Juniper M20 -** System and Switch Board(s), Routing Engine(s).
- **Juniper M40e** - Routing Engine(s), Miscellaneous Control System(s), System and Forwarding Module(s), PFE Clock Generators.
- **Juniper M160** - Routing Engine(s), Miscellaneous Control System(s), System and Forwarding Module(s), PFE Clock Generators.

Passive Monitoring

Passive Monitoring intelligence reports changes in the status of redundant components only after DX NetOps Spectrum has lost contact with the router. When contact is reestablished with the device, DX NetOps Spectrum queries the device. The query determines whether component status changes have occurred. Passive Monitoring is always on, but it only checks for component status changes in the previously mentioned case.

> **NOTE**
> When contact is reestablished with the device, the components are not always in a “steady” state. The components take a few minutes to reach their steady state. Each router type (M20, M40e, or M160) takes a different amount of time to reach its steady state. As a result, Passive Monitoring waits 60, 90, or 120 seconds before checking the states of the M20, M40e, or M160 components.

Active Monitoring

Active monitoring is used to report changes in the status of redundant components. The value of the Active Polling Interval determines the frequency of active monitoring. This interval determines how often (in seconds) the Active Monitoring intelligence queries the device to find component status changes. The field is read/write. For example, if the Active Polling Interval is set to 60, the device is queried every 60 seconds for component status changes. When Active Monitoring is enabled, it works in addition to the functionality provided by Passive Monitoring.

You have several other options for enabling or disabling this functionality. First, you can set the Active Polling Interval to 0 to disable active polling. Change the value to a value in seconds to enable this functionality when the Active Monitoring attribute is set to True. Changing the Polling Status of the device model to False also disables Active Monitoring.

Setting the Polling Interval of the device model to 0 also disables Active Monitoring. Changing Polling Status to True or changing the Polling Interval to a nonzero value does not reenable Active Monitoring, when Active Monitoring is disabled.

Never set the Active Polling Interval to a value that is less than the "steady" state time for the given router type. For example, the "steady" state time for an M20 is 60 seconds. Set the Active Polling Interval to a value that is greater than 60.

The following attributes control the Active Monitoring intelligence:

- **ActiveMonitor** - Enables or disables the active monitoring intelligence. The default value is disabled.
- **ActivePollInt** - Determines the frequency (in seconds) of Active Monitoring queries to the device for component status changes.

Juniper SRX Devices

Support for Juniper Logical Systems Route

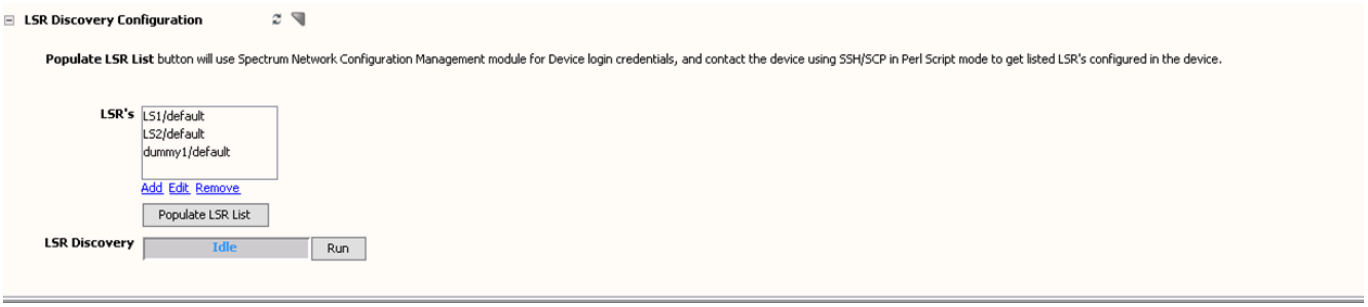
Starting from the 10.2.3 release, DX NetOps Spectrum supports Juniper SRX devices Logical Systems Route. DX NetOps Spectrum uses Network Configuration Management Perl script SSH mode to fetch a list of Logical Systems Routes (LSRs).

DX NetOps Spectrum discovers each LSR as a new Juniper model type. The LSR name is appended with default device name.

After discovery, Spectrum creates an interface for Juniper SRX devices and discovers the connection using the auto-discovery. All LSRs are associated to a container model and placed under the virtual device manager node.

LSR Discovery Configuration

The LSR Discovery Configuration section in the OneClick Interface tab allows you to configure the LSRs in DX NetOps Spectrum.



Using the LSR Discovery Configuration section, you can add, edit, or remove the LSRs in DX NetOps Spectrum.

The 'Populate LSR List' button allows you to get list of LSRs configured in the device using the SSH/SCP in Perl Script mode. It used the Spectrum Network Configuration Management mode for device login credentials.

LSR Discovery

The LSR Discovery option allows you to discover the LSRs configured in a device.

LSR hierarchy in the Explorer tab

After the discovery, a new hierarchy gets created for LSRs under the Virtual Device Manager in the Explorer tab. You can see the discovered devices in the Topology tab.

LSR Locater Search

You can use the Locater Search to search for 'All LSR Devices' and 'All LSR Host Devices'.

Netscreen Firewall

This section describes the Netscreen Tunnel Interface model type (nsTunnelIf) and its functionality.

Tunnel Interfaces

This section describes DX NetOps Spectrum support for monitoring NetScreen Firewall tunnel interfaces.

Model Tunnel Interfaces

Various attributes control whether the site-to-site Tunnel Interfaces are modeled on your Netscreen device. You can model other types of tunnel interfaces by using the following procedure. By default, DX NetOps Spectrum does not model Dialup Tunnels or Tunnels whose monitor state is set to OFF. To enable the modeling of these types of tunnels, use the Model Type Editor.

Follow these steps:

1. Shut down the SpectroSERVER and start the Model Type Editor.
2. To enable modeling of Dialup Tunnels, use the Search text box on the Attributes tab to find the TunnelFilterTypes attribute (0x12a17) of the NSFirewallVPN model type.
3. Remove the value 1 from the list of values for this attribute.
4. To enable modeling of tunnels whose monitor state is OFF, use the Search text box on the Attributes tab to find the TunnelFilterStates attribute (0x12a19) of the NSFirewallVPN model type.
5. Remove the value 0 from the list of values for this attribute.
6. Save your changes in the Model Type Editor, and restart the SpectroSERVER.

7. Reconfigure the Netscreen models using the Manually Poll Device option that is available for each device model. The tunnel interfaces are modeled.

Tunnel Interface “Stacking”

Tunnel interface models are created as subinterfaces of the physical interface whose IP address matches the local address of the tunnel. This behavior is indicated in the VPN-MON.mib. Because NetScreen devices do not support the ifStackTable, this mechanism for determining the lower-layer interface is necessary and effective.

Automatic Connectivity Mapping

A tunnel interface model activates for the first time during initial device modeling or during an interface reconfiguration. Then, DX NetOps Spectrum searches for a tunnel interface model that represents the other end point of the tunnel. If such a model is found, the connection between these two interfaces is modeled. DX NetOps Spectrum uses the local address and remote address that are indicated in the VPN-MON.mib to find the other end point of the tunnel.

Interface Model Identification

You can identify a Tunnel interface model by its local address and remote address, as indicated in the VPN-MON.mib. This identification method lets DX NetOps Spectrum preserve the interface model if the ifIndex of the interface changes.

Status Monitoring of Tunnel Interfaces

On the NetScreen device, the ifOperStatus of a tunnel interface entry is always "UP until it disappears from the ifTable. If a tunnel model becomes "stale", and no link down trap is processed for the tunnel, DX NetOps Spectrum generates a red alarm on the model.

This alarm is suppressed in the following cases:

- If the physical interface is down (the same case in which a link down trap alarm is suppressed).
- If the "Suppress Linked Port Alarms" setting of the Live Pipes model is set to True, and either of the following conditions are met:
  - The connected device is unreachable (by the SpectroSERVER)
  - The "linked" tunnel interface model has an alarm (red)

This status monitoring functionality is only available when Live Links are enabled for the port that is associated with the tunnel interface. For information about enabling Live Links, see the Modeling and Managing Your IT Infrastructure section.

Nortel Contivity VPN Switches

This section describes DX NetOps Spectrum support for monitoring Nortel Contivity VPN switches.

Tunnel Interfaces

This section describes the Tunnel Interface Filter functionality for Nortel Contivity devices.

Tunnel Interface Filtering

The ContivityVPN device populates the **ifTable** with both user and branch VPN tunnel interface entries. However, thousands of user VPN tunnel interfaces can exist. The ContivityVPN interface filtering functionality filters out user tunnel interfaces and prevents unnecessary modeling of these interfaces.

NOTE
Tunnel interface filtering is only available for models of type **ContivityVPN**.

Enabling and Disabling Tunnel IF Filtering

The following steps enable or disable Tunnel IF filtering:

Follow these steps:

1. In the Model Type Editor, set the default list value for the attribute If_Mtype_Map handle 0x011fb4.
2. Look at the list of values, and locate OID instance 131.
3. Set to a value of 0. This setting prevents the interface type from being modeled.
4. To disable the tunnel interface filtering and enable model creation, set this value to 220013.

Modeling of Tunnel Interfaces

The Create Sub-Interface attribute of the Contivity device model controls the creation of models to represent site-to-site or branch tunnel interfaces. No models are ever created to represent "user" tunnels. This behavior is consistent with previous versions.

Tunnel Interface "Stacking"

Tunnel interface models are created as sub-interfaces of the physical interface. The IP address of the physical interface matches the local address of the tunnel as indicated in the Tunnel MIB. The Contivity devices do not support the ifStackTable. As a result, this mechanism of determining the lower-layer interface is necessary and effective.

Automatic Connectivity Mapping

A tunnel interface model activates for the first time during initial device modeling or during an interface reconfiguration. Then DX NetOps Spectrum searches for a tunnel interface model that represents the other end point of the tunnel. If such a model is found, the connection between these two interfaces is modeled. DX NetOps Spectrum uses the local address and remote address that are indicated in the Tunnel MIB (rfc2667) to find the other tunnel end point.

Interface Model Identification

You can identify the Tunnel interface models by their local address and remote address, as indicated in the Tunnel MIB (rfc2667). This identification lets DX NetOps Spectrum preserve the interface model even if the ifIndex of the interface changes.

Interface Model Aging

During an interface reconfiguration, any interface model that is no longer represented in the MIB is marked as "stale" instead of being destroyed. This feature lets DX NetOps Spectrum retain the connectivity modeling between tunnel interfaces and other devices while the tunnel is down. The connectivity information can then be used for the event correlation and fault suppression.

On subsequent reconfigurations, the port age-out time of the device model is compared with the time period that the interface model has been stale. If the interface does not reappear in the MIB, the interface model is destroyed after it ages out. If the interface does reappear in the MIB, the interface model is marked as "current." The port is marked as stale by setting the "isStale" attribute to True. You can set the port age-out time per device Set the "PortAgeOutTime" on the device to a number of minutes. The default age-out time for the Contivity device is two hours (120 minutes).

Link Down Trap Correlation

To avoid sending multiple alarms for a single network outage, link down traps for "tunnel" interface models are correlated with other conditions. The alarms for link down traps are suppressed when the lower layer (that is, the physical interface)

is down. When the "Suppress Linked Port Alarms" setting of the Live Pipes model is set to True, the alarms for the link down traps are suppressed. The alarms are suppressed under the following conditions:

1. The connected device is unreachable (by the SpectroSERVER).
2. The "linked" tunnel interface model has an alarm (is red).

Status Monitoring of Tunnel Interfaces

On the Contivity device, the ifOperStatus of a tunnel interface entry is always "UP" until it disappears from the ifTable. When a tunnel model becomes "stale" and link down traps have not been processed for the tunnel, DX NetOps Spectrum generates a red alarm on the model. The red alarm is suppressed in the same cases in which a link down trap alarm is suppressed. The red alarm is suppressed when the lower layer (that is, physical interface) is down. When the "Suppress Linked Port Alarms" parameter of the Live Pipes model is set to True, this alarm is suppressed.

The alarm is suppressed under the following conditions:

1. The connected device is unreachable (by the SpectroSERVER).
2. The "linked" tunnel interface model has an alarm (red).

Contivity Management Settings

The following Contivity settings are recommended.

Enable Tunnel MIB

We recommend enabling the Tunnel IP MIB on all managed Contivity devices. This setting lets DX NetOps Spectrum create models to represent the tunnel end points on the device. This MIB can be enabled and disabled from the ADMIN->SNMP section of the Contivity web management pages.

Enable Link Up/Down Traps

We recommend enabling link up and link down traps for physical interfaces and for "Nailed-Up" branch tunnels. This setting gives DX NetOps Spectrum immediate notifications of link state changes. Our testing has shown that link traps for "OnDemand" tunnels do not provide much value. The tunnel must be down for approximately 15 minutes before the trap is sent.

Nail-Up Your Monitored Tunnels

We recommend that all tunnels for which connection monitoring is important be "Nailed-Up". DX NetOps Spectrum does not alarm "OnDemand" tunnels when they go down. Specifically, the Alarm on LINK down Trap attribute of the Tunnel_If model determines whether it responds to link down traps or changes to the isStale attribute. A value of Always (1) causes DX NetOps Spectrum to process these events; a value of Never (0) causes DX NetOps Spectrum to ignore them. When DX NetOps Spectrum creates the Tunnel_If models for the Contivity, it sets this attribute to Always for "Nailed-Up" branch tunnels, and Never for "OnDemand" tunnels.

Change the Alarm on LINK down setting from the Configuration tab of the Global Attribute Editor. We recommend leaving it as DX NetOps Spectrum has set it.

Management Settings

The following DX NetOps Spectrum management settings are recommended.

Automatically Reconfigure Interfaces

Set this attribute to True for the Contivity models if you want DX NetOps Spectrum to manage the branch tunnels of the device. For the devices that only support "User" tunnels, set this attribute to False. When set to True, DX NetOps Spectrum reconfigures the interface models whenever the ifNumber object of the SNMP agent changes on the device.

Reconfigure on LINK change

We recommend setting this attribute to False for all Contivity models. When it is set to True, DX NetOps Spectrum performs an interface reconfiguration after every link up or every link down trap is received.

Discovery after Reconfigure

We recommend retaining the default value of False for the Discovery after Reconfigure attribute for all Contivity models. DX NetOps Spectrum models connections between newly found tunnels regardless of this setting. The DX NetOps Spectrum autodiscovery process adds little or no value after most link state changes, especially for Contivity devices. For these devices, most link state changes represent tunnels coming up and going down, and not the configuration of new router or bridge ports.

Create Sub-Interfaces

Set this attribute to True for Contivity models if you want DX NetOps Spectrum to monitor the branch tunnels. If this attribute is set to False, DX NetOps Spectrum does not create models for the tunnel interfaces.

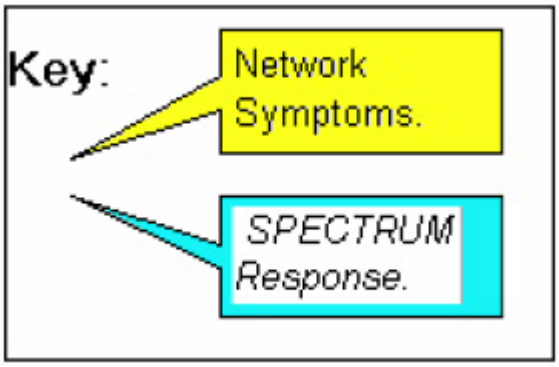
Suppress Linked Port Alarms

We recommend setting this attribute of the Live Pipes model to True. This setting suppresses port alarms when either the connected device is unreachable or the linked port model already has an alarm.

Contivity Fault Scenarios

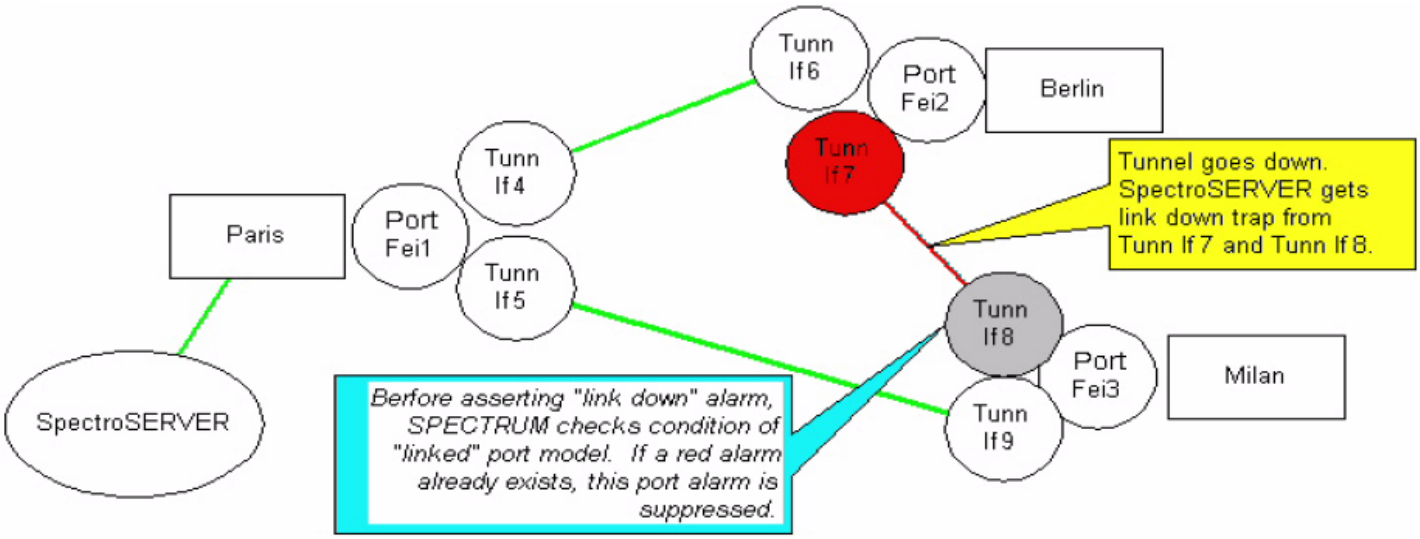
This section describes fault scenarios that are likely in a VPN environment and the DX NetOps Spectrum response to each scenario.

The following key applies to each of the diagrams in this section:



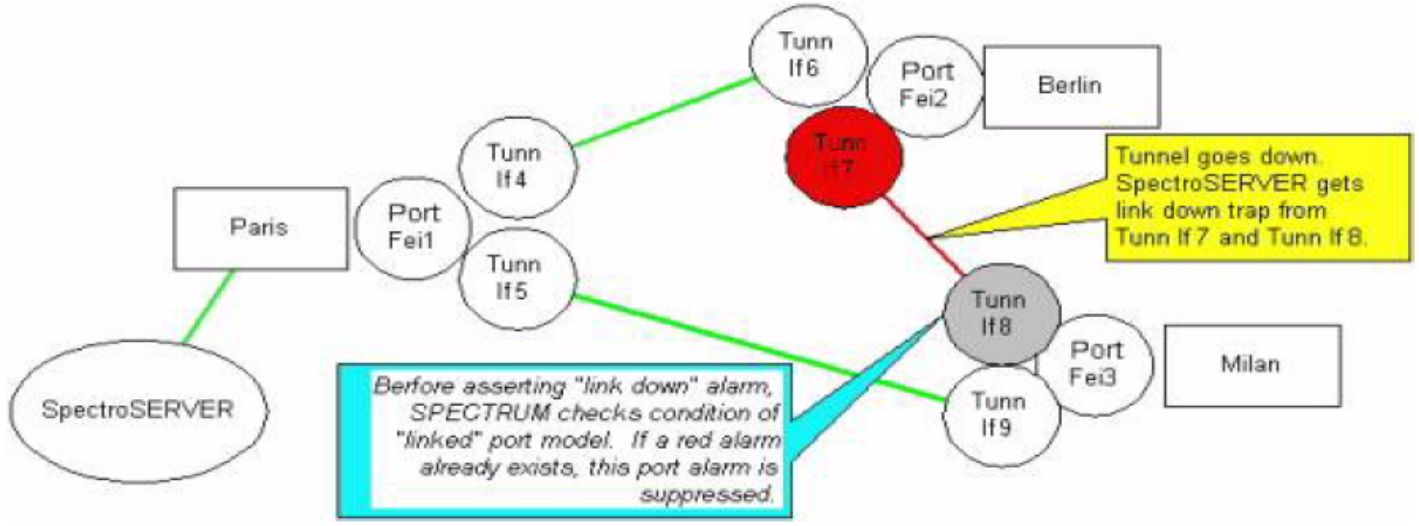
Two Link Down Traps for One Down Tunnel

In the following scenario, the SpectroSERVER retains contact to all managed elements in this meshed environment, but a tunnel between two devices goes down. DX NetOps Spectrum receives two link down traps. One tunnel interface alarms; the other alarm is suppressed.



Loss of Contact and Link Down Trap

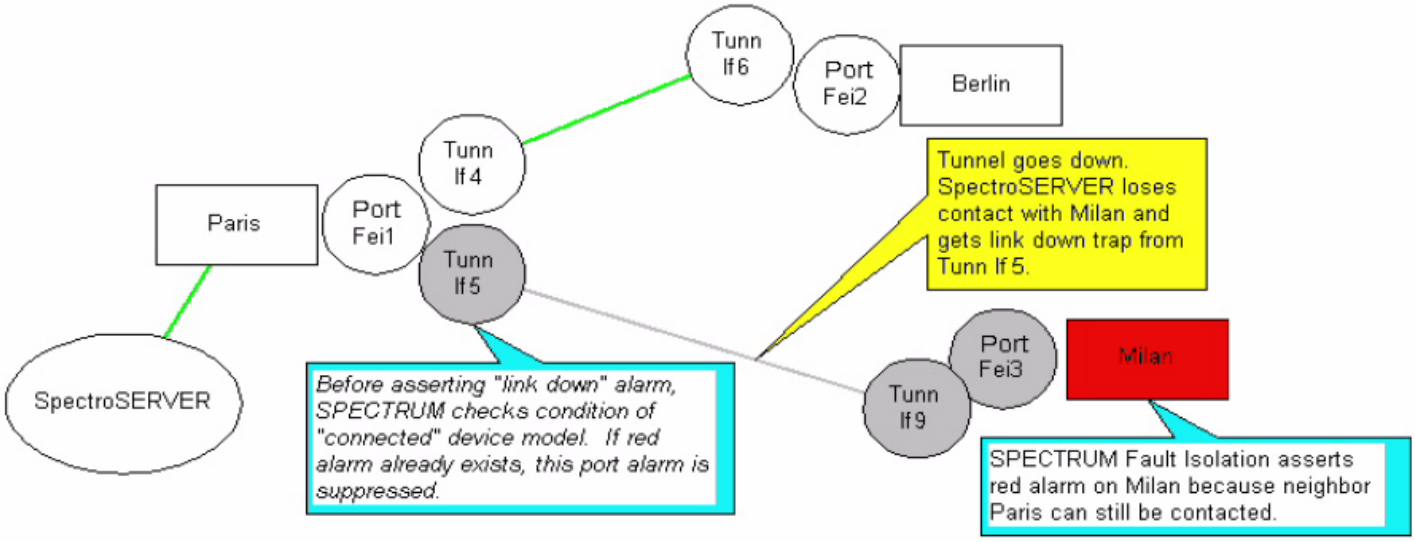
In the following scenario, DX NetOps Spectrum loses contact with a "spoke" Contivity in a hub and spoke network. DX NetOps Spectrum also receives a link down trap from the hub, indicating the tunnel to the lost device. DX NetOps Spectrum sends an alarm for the lost device and suppresses the alarm on the tunnel interface that is indicated by the trap.



Physical Port Down, Loss of Contact, and Link Down Traps

In the following scenario, a physical port of a Contivity goes down or loses its link to the public network. DX NetOps Spectrum gets link down traps for the physical port and tunnels of the Contivity, and loses contact with remote the

Contivity devices. The link down alarms on the tunnel interface models are suppressed, but DX NetOps Spectrum fault isolation creates red alarms on the lost Contivity device models because they have an "up" neighbor.



Known Anomalies - Nortel Contivity

DX NetOps Spectrum contains the following known anomalies.

Sub-Interfaces

When Create Sub-Interfaces is changed from True to False for a Contivity model after tunnel interface models have been created, the tunnel interface models are not destroyed immediately after an interface reconfiguration. Instead, these models go stale and start aging out. To enable tunnel monitoring for a subset of Contivity devices, set the default value of Create Sub-Interfaces to False. Then set Create Sub-Interfaces to True for the individual models of Contivity devices that require tunnel monitoring.

Autodiscovery and Public Address

Generally, the public addresses on the Contivity devices in a VPN are in different subnets because multiple routers separate them. The Contivity devices with public interfaces can be on the same subnet. In this case, autodiscovery can attempt to map the connectivity of the public interfaces. The result would be a LAN container in the same topology view as the Contivity models with pipes to the Contivity models. A fanout model without the LAN would be connected to the public interface models of the Contivity devices.

Port Aging

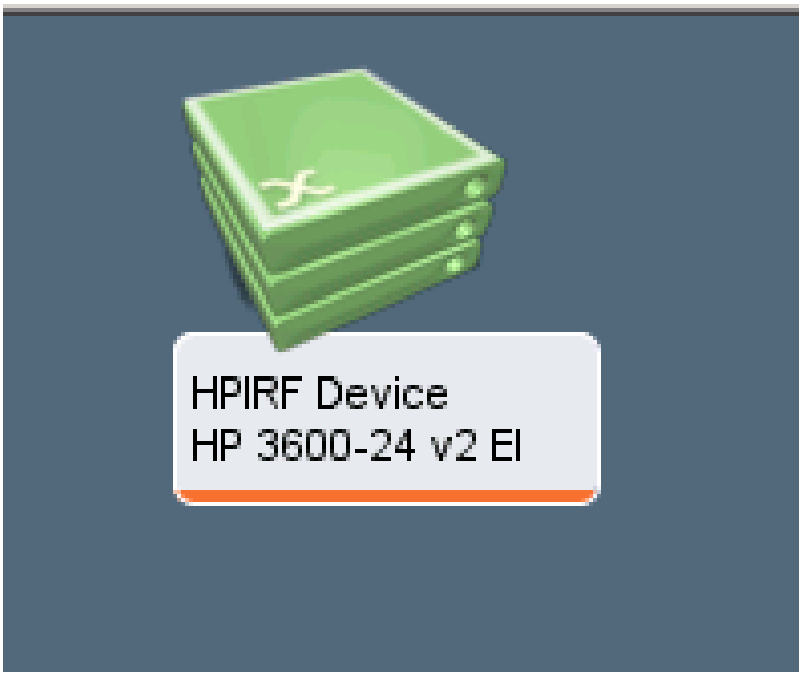
DX NetOps Spectrum port aging is not aggressive. When a tunnel becomes inactive, the tunnel interface model is marked as "Stale". Any future reconfiguration that occurs after the "portAgeOutTime" of the device causes that tunnel model to be destroyed. However, if no future reconfigurations of the device occur, the "Stale" tunnel interface model remains.

For example, consider a polling interval of 5 minutes and a portAgeOutTime of 30 minutes. If a tunnel goes down at 10:27 and DX NetOps Spectrum polls at 10:30, DX NetOps Spectrum detects an ifNumber change and performs and interface reconfiguration. During this process, the tunnel interface is marked as stale. If the tunnel does not come back up, the tunnel interface model is destroyed at 11:00. When ifNumber does not change again for a week, interface reconfiguration cannot run again for a week. This tunnel interface model remains stale for one week and is then destroyed.

HP IRF Device Enhancements

- Modeling of physical devices is successful: After HP IRF device is modeled, double click on the device model to view the physical devices and port level connectivity from the physical devices to neighbor devices.
- EnableChassisTopology option: To show physical devices on the Topology view, enable the attribute 'Enable_Chassis_Topology=Yes'. To enable the attribute to Yes, Navigate to **Topology or List Tab**> **Attribute**> Select the **Enable_Chassis** and set the attribute to '**Yes**'. By default value of Enable_Chassis_Topology is No. This option is applicable only for HPIRF Stack device
- Naming Convention: Appended serial numbers to the master and slave devices in the interface tab, allows for easy differentiation

Icon & device type is displayed: To differentiate the HP IRF stack devices, the icon of devices, now looks as shown in the following screenshot:



There is a change of icon, which is (M) for the master member and (S) for the slave member to help identify the device on the topology.

The master and the slave devices in chassis manager and the interface tab, can now be clearly identified.

NOTE

If a master or slave device is deleted manually then you must reconfigure the model to retrieve the connections.

Limitations:

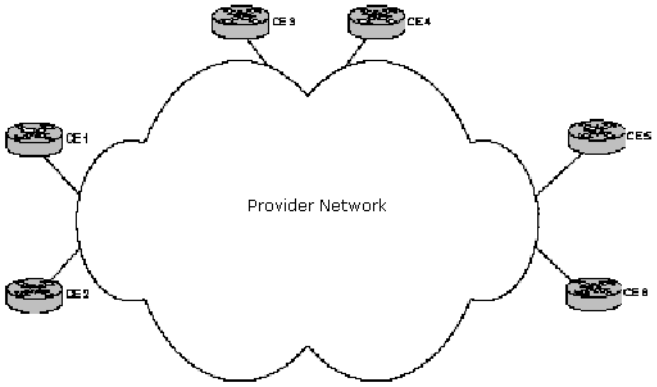
- Changes made to the master and slave devices in OneClick Topology edit mode does not work.
- The interface tab for the HP IRF stack device displays duplicate interfaces (group by IRF stack member and on the stack device).
- The Locater search for stack devices shows a 'no models found' error.
- HP IRF support will not be enabled by default on existing models, after upgrading to 10.3.

Enterprise VPN Manager

About Enterprise VPN Manager

Enterprise VPN Manager is a OneClick application that lets you discover, model, and monitor a provider-provisioned VPN. Your enterprise network ends at your customer edge (CE) devices. When lack access to performance statistics from provider core (P) or provider edge (PE) routers, you can infer service health by monitoring the behavior at the edges of the provider network, where enterprise CE devices exist.

The following illustration shows a typical provider-provisioned VPN from an enterprise customer perspective:



Model the links and devices (the CE routers) connected to a service provider using the Enterprise VPN Manager discovery, manual modeling, or import functionality. The Enterprise VPN Manager component then continuously monitors health and the service that is delivered by a service provider. The Provider_Cloud model represents the service that is provided to you by a service provider. The Outage events are processed and rolled-up into the health of the service provider which is reflected on the Provider_Cloud model. The Outage events are detected by actively polling CE routers for an availability and by polling the status of interfaces on CE routers which are connected to a service provider. Enterprise VPN Manager supports Layer 3 Multiprotocol Label Switching (MPLS) VPN networks for discovery, modeling, and monitoring. In addition, it can also model (manually or by importing the information) and monitor Layer 2 Virtual Private LAN Service (VPLS).

In addition to active polling, Enterprise VPN Manager supports Service Assurance (SA) Ping tests. Set up these tests to monitor service delivery and compliance with response time service-level agreements (SLAs). Ping tests offer the strongest option for service monitoring because they measure end-to-end response time.

NOTE

All elements that are connected to a given service provider must be modeled on a single SpectroSERVER.

Access Enterprise VPN Manager

You can access the Enterprise VPN Manager in the OneClick Navigation panel. Expand the appropriate landscape in the Explorer tab and select Enterprise VPN Manager.

Model information is displayed in the Contents and Component Detail panels.

Discovery and Modeling in Enterprise VPN Manager

Provider_Cloud Model

The Provider_Cloud model represents the service that is offered to you by a service provider. Enterprise VPN Manager offers several methods to model network entities and services, such as Discovery, import, and manual modeling functionality to accommodate the unique needs of your enterprise.

Existing Device Models

If the CE devices with BGP4_App modeled in DX NetOps Spectrum, you can run Enterprise VPN Discovery to detect their connections to the service provider. Otherwise, you can delete and remodel the devices or run Application Reconfiguration on those devices. Once the necessary application models are present, you can run Enterprise VPN Discovery.

Add Autonomous System Numbers (ASNs) to the DX NetOps Spectrum Database

By default, DX NetOps Spectrum includes over two thousand officially registered Service Provider ASNs. You can add additional Service Provider ASNs to the DX NetOps Spectrum database with the Model Type Editor by modifying the ASNamesList attribute of the EntVpnManager model type.

NOTE
For more information about editing model type attributes, see the *Model Type Editor .*

Enterprise VPN Discovery Prerequisites

Model the physical components of your network, before using the Enterprise VPN Manager Discovery functionality.

NOTE
For information about modeling your network, see the *Modeling and Managing Your IT Infrastructure Administrator* and the *Modeling Gateway Toolkit .*

At minimum, verify that your CE routers are modeled in DX NetOps Spectrum.

On CE routers, verify that BGP peering to the service provider is properly configured. If your devices do not support BGP peering, Enterprise VPN Manager supports import from a CSV text file that is based on Autonomous System Numbers (ASN) and manual modeling.

Access Enterprise VPN Discovery Configuration

The Enterprise VPN Discovery subview contains discovery controls and configuration.

Follow these steps:

1. Expand the appropriate landscape in the Explorer tab of the Navigation panel. Select Enterprise VPN Manager. Information and configuration appear in the Information tab of the Contents panel.
2. Expand the Configuration subview, and then expand the Enterprise VPN Discovery subview.
   The Enterprise VPN Manager Discovery options display:
   - Run Discovery
   - Import Config File
   - Provider Name Filter Type
   - Provider Name Filter
   - Discover On Activation
   - Create On Trap
   - Enable Background Discovery
   - Background Discovery Interval (minutes)
   - ASN Mapping

Configure Automatic Discovery on Model Activation

You can configure Enterprise VPN Manager to discover provider networks and CE interfaces automatically using the Discover On Activation option. When Discover On Activation is enabled, an Enterprise VPN Discovery is initiated each time DX NetOps Spectrum activates a device model. This process occurs on initial device model creation or on a SpectroSERVER restart. Determine whether this processing load (during these times) is appropriate in your environment. If not, disable this attribute.

Follow these steps:

1. Expand the Enterprise VPN Discovery subview as described in Access Enterprise VPN Discovery Configurations. Enterprise VPN Discovery options appear.
2. Click set next to Discover On Activation. Select Yes to enable. The option is disabled by default.
   Discover On Activation is set. The value that you selected is displayed next to Discover On Activation.

Configure Enterprise VPN Manager for a bgpEstablished Trap

Several devices that are configured with BGP4_App send a bgpEstablished trap when they establish a connection (a new peering session). You can configure Enterprise VPN Manager to discover provider networks and CE interfaces automatically in response to a bgpEstablished trap. When Create On Trap is enabled, an Enterprise VPN Discovery is initiated each time a bgpEstablished trap is received.

The Create On Trap option offers an alternative to Background discovery. Determine whether this Discovery option is appropriate in your environment. If this behavior is not desired, disable this attribute.

Follow these steps:

1. Expand the Enterprise VPN Discovery subview as described in Access Enterprise VPN Discovery Configurations.
   Enterprise VPN Discovery options appear.
2. Click set next to Create On Trap and select Yes.
   Enterprise VPN Manager runs Discovery when it receives a bgpEstablished trap.

Enable Background Discovery

You can configure Enterprise VPN Manager to discover provider networks and CE interfaces automatically using the Background Discovery option. When Enable Background Discovery is enabled, an Enterprise VPN Discovery is initiated based on the Background Discovery Interval. This process lets you determine the frequency of Discovery in your network. Determine whether this processing load (during these times) is appropriate in your environment.

Follow these steps:

1. Expand the Enterprise VPN Discovery subview as described in Access Enterprise VPN Discovery Configurations.
   Enterprise VPN Discovery options appear.
2. Click set next to Enable Background Discovery and select Yes.
   Background Discovery is enabled and recurs according to the frequency that you set in the Background Discovery Interval.

Configure the Background Discovery Interval

If you enable background Discovery, you can configure the frequency for background Discovery. Enable Background Discovery must be set to Yes to enable the value for the Background Discovery Interval parameter.

Follow these steps:

1. Expand the Enterprise VPN Discovery subview as described in .
   Enterprise VPN Discovery options appear.
2. Click set next to Background Discovery Interval.
3. Enter a value (in minutes).
   The Background Discovery interval is set.

Run an On-Demand Enterprise VPN Discovery

Enterprise VPN Discovery is the simplest method of modeling your network. Meet the prerequisites, before running an on-demand Enterprise VPN Discovery.

Follow these steps:

1. Expand the Enterprise VPN Discovery subview.
   Enterprise VPN Discovery options appear.
2. Click Run.
   The Enterprise VPN Discovery runs, and Discovery status is displayed in the window next to the Run button.

Run Enterprise VPN Discovery on Selected Models

You can configure the Enterprise VPN Network Services Discovery from the OneClick views that display models.

Follow these steps:

1. Select the models.
2. Click Tools, Utilities, Network Services Discoveries, Enterprise VPN Discovery.
   The Discovery process is initiated. You can check the status in the Configuration subview.

Configuring Enterprise VPN Discovery During Modeling

DX NetOps Spectrum lets you configure Network Services Discoveries, including Enterprise VPN Discovery, during modeling. As a part of modeling configuration, you can specify the network service discoveries to run with the modeling process.

NOTE
For more information, see *Modeling and Managing Your IT Infrastructure*.

Filter Service Provider Names During Discovery

The Enterprise VPN Manager lets you filter the Service Provider names during Discovery run.

Follow these steps:

1. Select Enterprise VPN Manager in the Explorer tab.
2. Select the Information tab in the Contents panel.
   The configuration options for Enterprise VPN Manager display.
3. Expand the Enterprise VPN Discovery subview.
   The Discovery options are available, including these options:
   - **Provider Name Filter Type**
     Determines whether the Provider names in the 'Provider Name Filter' field are included or excluded from modeling. Select from the following options:
     - Exclusive
     - Inclusive
   - **Provider Name Filter**
     Lists the Service Provider names to be included or excluded when the Enterprise VPN Discovery is run. This field is used together with the 'Provider Name Filter Type' field.

     NOTE
     Add Service Provider names to the Provider Name Filter field to filter and save them. If the Provider Name Filter Type is Inclusive and the Provider Name Filter is empty, all Provider Names are discovered.

Import Peer/Provider Information

Enterprise VPN Manager enables you to import service provider information from MPLS and VPLS VPNs to create Provider_Cloud models. If BGP peering is not used to communicate with your provider, import lets you associate your Provider_Cloud models with sites. The import file must be in a comma-separated value (CSV) formatted file. You can create a CSV file with a text editor or can export it from another application. Know the service provider ASN and the IP Address for MPLS VPNs or the Interface Model Name for VPLS VPNs of your CE Interfaces.

NOTE
Devices and interfaces must be modeled in DX NetOps Spectrum before importing a CSV-formatted text file.

The following parameters are supported for a line entry in an import file:

ProviderASN,SiteIfIdentifier,ProviderName,Region,SiteName,SitePriority

- **ProviderASN**
  Specifies the Autonomous System Number of the provider. Required parameter.
- **SiteIfIdentifier**
  Specifies the IP Address for MPLS VPNs or the Interface Model Name of the site interface for VPLS VPNs. SiteIfIdentifier is required.
- **ProviderName**
  Specifies the Name of the service provider.
- **Region**
  Lets you define Alarm Domains. This option is helpful when users have regional responsibility. The following values are available:
  - Unavailable = Unavailable
  - Arin = United States and Canada
  - Lacnic = Latin America
  - Ripe = EMEA
  - Afrinic = Africa
  - Apnic = Asia Pacific
- **SiteName**
  Specifies the name of the Provider_Cloud model.
- **SitePriority**
  Specifies an integer from 1 to N where 1 represents the primary connection and 2 through N represent backup connections.

The following text is an example of a CSV import file:

1234,138.42.14.143,ProvName,Lacnic,SiteName,3

Follow these steps:

1. Expand the Enterprise VPN Discovery subview as described in Access Enterprise VPN Discovery Configurations.
   The Enterprise VPN Discovery options display.
2. Click Import.
   The Import File dialog opens.
3. Locate your import file and click Open.
   Your peer or provider information is imported.

Create a Service Provider Model

Enterprise VPN Manager lets you manually model the connections to a service from your provider. Manual modeling is an alternative method of modeling devices that do not support the BGP4_App MIB, which is required to run Enterprise VPN Discovery. Manually modeling the connections to your service provider requires significant time and maintenance.

NOTE
For more information about manual modeling, see the *Modeling and Managing Your IT Infrastructure*.

Follow these steps:

1. In the OneClick Universe Topology tab, in the Topology toolbar,
   click 
   The Select Model Type dialog opens.
2. In the All Model Types tab, select Provider_Cloud and click OK.
   The Create Model of Type dialog opens.
3. Fill in the appropriate information that is requested in the dialog and click OK.
   This model represents the network of the provider.

4. Create models of your CE devices.
5. Select a CE model in the Topology view.
6. Select the Interfaces tab of the Component Detail panel.
7. Locate the interface that is connected to the provider.
8. Right-click the interface and select Start Connection.
9. Return to the Universe Topology view.
10. Right-click the Provider_Cloud model and select Connect with <interfaceName>.
    This operation associates the interface of CE router with the provider.
11. Repeat for each CE interface that must be manually connected to the provider.

Service Monitoring Configuration

Overview

Enterprise VPN Manager can gather information about your provider network by pinging or polling. Ping tests offer the strongest measurement of service health because they measure end to end, while a port polling is focused on a single resource (a device or interface). However, the Ping tests have resource requirements that can affect on your network and equipment. Specifically, the Ping tests require the following additional resources:

- Processing time in DX NetOps Spectrum
- Network bandwidth to set up the tests
- Processing time in the CE routers
- Network bandwidth to execute the tests

Despite the resource requirements of Ping tests, they provide a valuable addition to your management capabilities. We recommend enabling the Ping tests.

Note: An SNMP read or write community name is required to provision SNMP polling.

Polling Configuration

Polling for port status serves as an alternative to Ping testing. Polling does not create as much of a network strain as Ping testing. SNMP secure information is not required to enable polling.

Port Polling

Enterprise VPN Manager uses the port status of the connected interfaces to update the status of the Provider_Cloud model. If port polling is disabled, the status is not updated. We recommend leaving port polling enabled (the default).

NOTE
Port polling includes all BGP sessions (active and inactive).

Follow these steps:

1. Locate the appropriate Enterprise VPN Manager model.
2. In the Contents panel, select the Information tab.
3. Expand the Configuration subview.
4. Expand the Management Configuration subview.
5. Locate Enable Port Polling, click set, and select Yes.
   Port Polling is now enabled.

Peer Session Polling

You can configure Enterprise VPN Manager to poll the status of peer sessions. When Peer Session Polling is enabled, Enterprise VPN Manager looks for changes to PeerState. The total number of operative and inoperative BGP peering sessions is used to compute the percentage of peering failures. This metric is evaluated when Enterprise VPN Manager determines the Provider_Cloud status.

NOTE
For more information about enabling alarms on failed peering sessions, see Modeling and Managing Your IT Infrastructure.

Follow these steps:

1. Locate the appropriate Enterprise VPN Manager model.
2. In the Contents panel, select the Information tab.
3. Expand the Configuration subview.
4. Expand the Management Configuration subview.
5. Locate Enable Peer Session Polling and click set.
6. Select Yes to enable peer session polling.

Ping Test Requirements

Verify that the following requirements are met to enable Ping tests:

- Devices must support Cisco RTTMON MIB or RFC2925.
- Devices must be modeled with a read/write community name.

Configure at least one site with one of the following options:

- Ping from Site.
- Ping to/from Site.

Configure at least one (other) site with one of the following options:

- Ping to Site.
- Ping to/from Site.
- Ping Test Interval, Ping Test Timeout, and Response Time Threshold, set to appropriate values.
- The 'Enable Ping Tests' parameter must be enabled on each Provider_Cloud model that participates in the ping test.
- The 'Enable Ping Tests' parameter must be enabled on the Enterprise VPN Manager model.

Ping Test Scalability

The scalability of Ping tests must be considered in large environments where fully meshed testing is performed. Performance testing has shown that full mesh testing beyond 50 sites greatly increases network traffic. Enterprise VPN Ping tests are therefore disabled by default. The number of Ping tests (and the resource requirements) can be efficiently managed by organizing your Ping tests.

We recommend selecting a relatively small number of important sites to perform Ping testing. When the number of sites (or remote offices) exceeds 50, let larger regional offices test back to corporate headquarters or test among themselves. For example, in an enterprise environment that consists of several regional offices and a corporate headquarters, configure your corporate headquarters as Ping to Site and your larger regional offices as Ping from Site to reduce the network load.

Configure Ping Tests

Configure Ping Tests in Enterprise VPN Manager.

Follow these steps:

1. Access Enterprise VPN Manager.
2. Select the List tab in the Content panel and select the Provider_Cloud model.
3. Select the Information tab in the Component Detail panel and expand the Ping Test Configuration subview.
   The following Ping test configurations are available:
   – **Ping Test Interval (sec)**
   Determines ping test frequency. Raise the value to reduce network traffic.
   Default: 1200 seconds

   NOTE
   Lower values for the Ping Test Interval attribute can cause a severe performance impact.
   – **Ping Test Timeout (sec)**
   Sets the timeout value before an event is generated for Ping tests. An event is generated if the ping response is not received before the timeout.
   Default: 5 seconds
   – **Response Time Threshold (ms)**
   Sets the event threshold for the response time of a successful Ping test.
   Default: 250 milliseconds

Configure Ping Source and Destination

Remote sites typically communicate more efficiently with a central location than with each other. You can select the interfaces that send and receive a ping. By default, interfaces are set to Ping from Site. Therefore, no testing occurs until at least one interface is set to either Ping to Site or Ping to/from Site.

The following settings are available:

- **Ping Disabled**
  Indicates that ping is not enabled for this interface.
- **Ping from Site**
  Indicates that this interface can only originate a ping.
- **Ping to Site**
  Indicates that this interface can only receive a ping.
- **Ping to/from Site**
  Indicates that this interface can originate and receive a ping.

Follow these steps:

1. Select the appropriate Provider_Cloud model to change all relevant interfaces or select an individual Interface model to make an individual change.
2. Open the Attribute Editor.
3. Click Add next to the User Defined folder.
   The Attribute Selector dialog opens.
4. In the left panel of the Attribute Selector, select the Port folder.
5. In the right panel of the Attribute Selector, locate and select the PingTestEnable attribute and click OK.
   PingTestEnable is now displayed in the right panel of the Attribute Editor.
6. Select the appropriate value in the PingTestEnable list.
   Click OK.

   NOTE
   For more information on the Attribute Editor, see Modeling and Managing Your IT Infrastructure.

Collapse BiDirectional Ping

Collapsing the BiDirectional pings reduces network traffic by eliminating potentially redundant ping tests. If one site can receive responses from another, the network between them functions properly. Therefore no test is sent in the opposite direction if BiDirectional tests are collapsed. By default, Collapse BiDirectional Ping is enabled.

Follow these steps:

1. Locate the appropriate Enterprise VPN Manager model.
2. In the Contents panel, select the Information tab.
3. Expand the Configuration subview.
4. Set the value of Collapse BiDirectional Ping to Yes.

Enable or Disable Ping Tests

Ping tests involve additional resource requirements on your network and equipment. As result, these tests are disabled by default. Ping tests are not conducted until:

- Ping source and destination for all Interfaces participate in the ping is configured
- Enable Ping Tests is set to Yes for each Provider_Cloud participating in the Ping test
- Enable Ping Tests is set to Yes on the Enterprise VPN Manager model

To enable or disable ping tests on the Enterprise VPN Manager model

1. Locate the appropriate Enterprise VPN Manager model.
2. In the Contents panel, select the Information tab.
3. Expand the Configuration subview.
4. Expand the Ping subview.
5. Set the value of Enable Ping Tests to Yes to enable ping testing. The default value is No.

To enable or disable ping tests on a Provider_Cloud model

1. Locate the appropriate Enterprise VPN Manager model.
2. In the Contents panel, select the List tab.
3. Select the appropriate Provider_Cloud model.
4. In the Component Detail panel, select the Information tab.
5. Expand the Ping Test Configuration subview.
6. Set the value of Enable Ping Tests to Yes to enable ping testing. The default value is No.

Manage Provider VPN Services

Managing Provider VPN Services

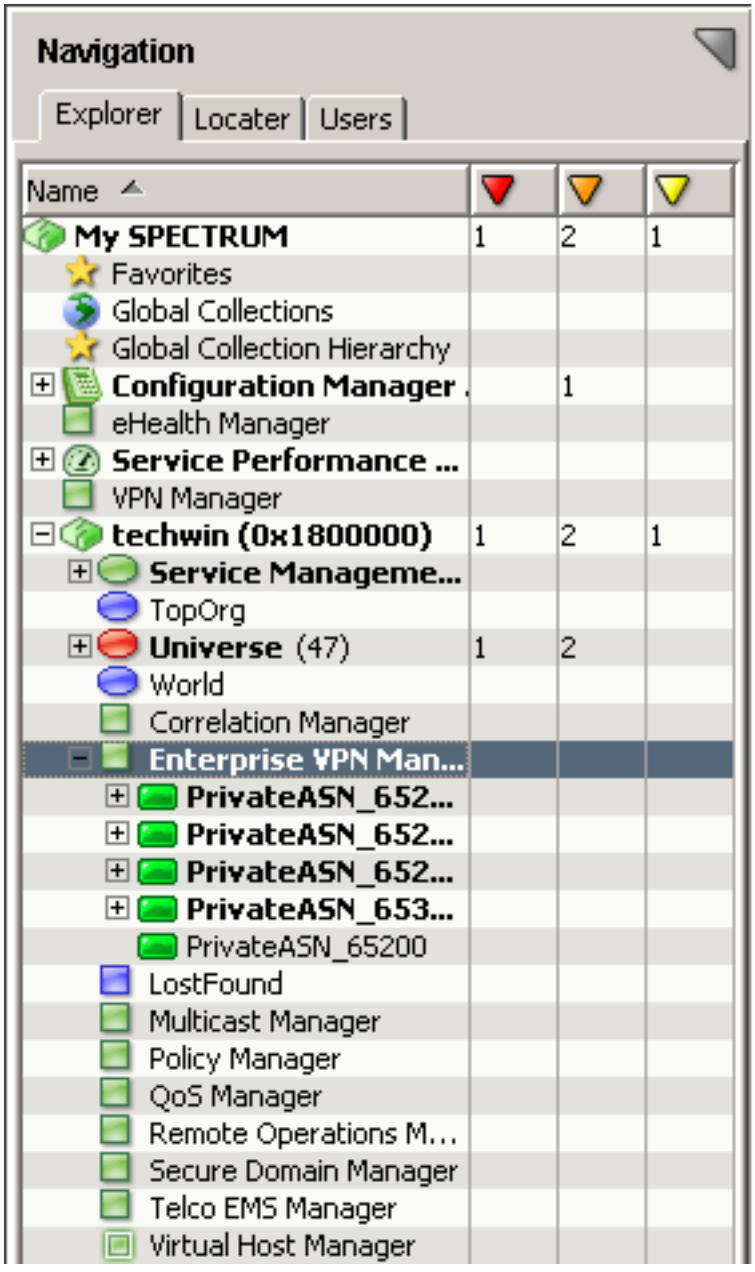
Enterprise VPN Manager Services

Enterprise VPN Manager enables you to continuously monitor the services that are provided to you by your service provider. You can see events and alarms pertaining to the health of various sites and the overall health of your provider. Enterprise VPN Manager lets you monitor the service across the three hierarchal levels (Enterprise VPN Manager model, Provider_Cloud model, and the individual interface/site models) of your provider network.

Enterprise VPN Manager Navigation

The OneClick Navigation panel displays a hierarchal view of your network. The Enterprise VPN Manager model exists within a particular landscape.

Expand the Enterprise VPN Manager model to see your providers and the sites that are connected to your providers, as shown in the following image:



Conduct an Enterprise VPN Search

You can access Enterprise VPN searches through the Locater tab. The Enterprise VPN search results, which appear in the Contents tab, help you access views that present management, performance, and configuration information. The Component Detail panel displays information about the device that is selected in the Contents panel. The following Enterprise VPN searches are available:

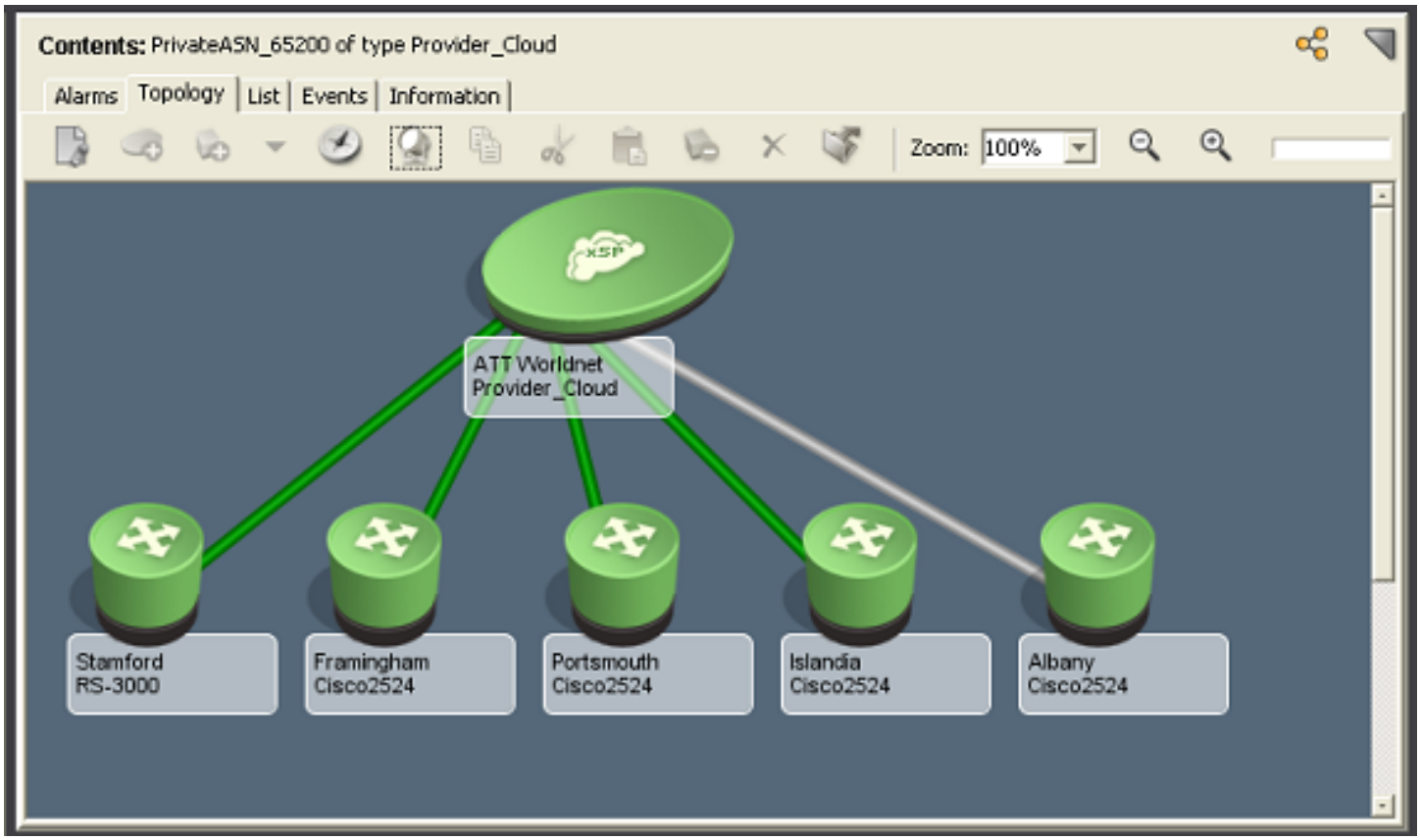
- All CE Devices by Provider
- All CE Interfaces by Provider
- All Enterprise VPN Managers
- All Providers

View Provider_Cloud Topology

The Provider Topology view displays the CE devices that are connected to a Provider_Cloud model. Selecting the Provider_Cloud or an interface model icon displays model information in the Component Detail panel.

Follow these steps:

1. Select the appropriate landscape from the Explorer tab in the OneClick Navigation panel.
2. Expand the Enterprise VPN Manager subview.
3. Select the appropriate Provider_Cloud model.
4. Select the Topology tab in the Contents panel.
   The Provider Topology resembles the following example:



View Events and Alarms

Events and Alarms that are generated on a selected Provider_Cloud model display in the Events tab and Alarms tab of the OneClick Contents panel.

NOTE
For more information about the Contents panel, Using OneClick .

Execute OnDemand Ping Test

Enterprise VPN Manager lets you provision an OnDemand Ping test between two sites. The OnDemand Ping tests are a good way to troubleshoot connectivity between two sites without the resource requirements that are necessary to provision background Ping tests in a large environment.

Follow these steps:

1. Activate the Topology view for the appropriate Provider_Cloud model.
2. Click Select OnDemand Ping Start Point from the right-click menu of the interface model icon from which you would like to initiate the Ping test.
3. Right-click the interface model icon that is the destination of Ping test, and select Ping Test From <source_model_name>. This process starts the on-demand Ping test.
   The results of the Ping test appear in a dialog after the Ping test completes.

   NOTE
   You can execute an on-demand Ping test using interface models in the Navigation panel.

Provider_Cloud Condition

Enterprise VPN Manager provides information about the status or condition of a Provider_Cloud model. The following types of information contribute to status reporting:

- Status of the interfaces that are connected to the Provider_Cloud
- Results of the automated service assurance tests (Ping and Response Time)

WARNING
To enable calculation of the VPN Manager Provider_Cloud condition, verify that the Live Pipes field is enabled. VPN status is not properly updated when Disabling Live Pipes are disabled.

The total number of operative and inoperative BGP peering sessions is used to compute the percent of peering failures. The percent of peering failures is evaluated when Enterprise VPN Manager determines the Provider_Cloud status.

NOTE
For more information on BGP peering sessions, see Modeling and Managing Your IT Infrastructure .

The default thresholds are available in the Information tab of the Provider_Cloud model. The thresholds and their default value are:

- Critical Alarm Threshold % - 5%
- Major Alarm Threshold % - 3%
- Minor Alarm Threshold % - 1%

Provider_Cloud Roll-Up Condition

The aggregate condition of the interfaces contributes to the roll-up condition of the Provider_Cloud. For example, if 100 interfaces (or sites) are connected to a provider and 4 of those interfaces are unreachable by DX NetOps Spectrum, the condition is calculated as follows:

4 of 100 interfaces (4 percent) are unreachable.

This provider has a condition of Major Alarm because the 4 percent outage is above the default Major Alarm Failure threshold of 3 percent.

Service Assurance Test

Automated service assurance tests provide the best indication of provider health. These tests assure not only that the interface and the BGP peering session is operating but that the endpoints are able to pass traffic. An additional test

verifies that the traffic passing through the network of the provider can reach the endpoint within the time thresholds that are specified in the service agreement.

Hide Symptomatic Alarms

Hiding symptomatic alarms reduces the number of alarms that are presented to you. You can configure Enterprise VPN Manager to generate a single alarm when a percentage of your interfaces that are connected to a service provider lose connectivity. When a significant number of sites are experiencing simultaneous problems, the provider is typically the root cause. The Minor Alarm Threshold is a point where Enterprise VPN Manager suppresses multiple site alarms and it instead generates an alarm on the Provider_Cloud model.

For example, you model a network with ten CE devices that are connected to a Provider_Cloud model. The alarm thresholds have the following settings:

- Critical Alarm Threshold %: 35
- Major Alarm Threshold %: 25
- Minor Alarm Threshold %: 15

> **NOTE**
> These settings are usually high and are not recommended for operational use.

When one CE device (10 percent of the devices that are connected to the service provider) becomes unreachable by DX NetOps Spectrum, a red alarm is raised on the device model and the status of the Provider_Cloud model remains green. The status of the Provider_Cloud model remains green because 10 percent falls below the 15 percent Minor Alarm Threshold. When a second CE device (20 percent) becomes unreachable, the Minor Alarm Threshold is violated. The status of the Provider_Cloud model is Minor Alarm and the alarms on the device models are hidden. The status of the two unreachable devices remains critical, but no Contact Lost alarms appear in the alarm log.

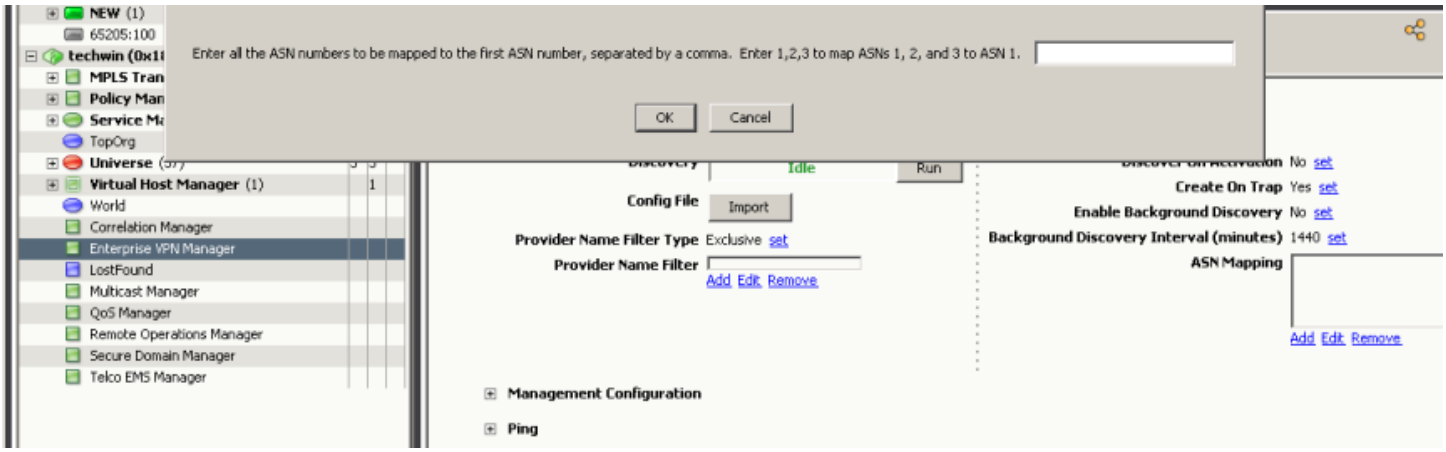
Follow these steps:

1. Locate the appropriate Enterprise VPN Manager model.
2. In the Contents panel, select the List tab.
3. Select the appropriate Provider_Cloud model.
4. In the Component Detail panel, select the Information tab.
5. Expand the Configuration Information subview.
6. Specify the following thresholds:
   - **Critical Alarm Threshold (%)**
     Specifies the threshold for hiding Critical alarms.
     Default: 5
   - **Major Alarm Threshold (%)**
     Specifies the threshold for hiding Major alarms.
     Default: 3
   - **Minor Alarm Threshold (%)**
     Specifies the threshold for hiding Minor alarms.
     Default: 1
7. Select the Impact tab.
8. Expand the Symptoms subview.
   You can view the affected device sites.

Map Multiple Autonomous System Numbers to a Single Provider

The DX NetOps Spectrum MultipleASNLists attribute is now a list rather than a text string. This list lets you easily map multiple Autonomous System Numbers (ASNs) to a single provider.

This option is available when you select Configuration from the Enterprise VPN Manager and then edit the ASN Mapping field. DX NetOps Spectrum displays a List Renderer that lets you easily add, edit, or remove each list of ASNs that are mapped to the first ASN in the list, as shown in the following image:



Enterprise VPN Manager Events

Enterprise VPN Manager Events

Enterprise VPN Manager events enhance management of provider based services. Most of these events mirror the roll-up and service assurance methods of provider status calculation.

Roll-Up Method

The events in the roll-up method approach reflect the health of the service provider, infrastructure, which is modeled using the Provider_Cloud model type. The supported events are listed in the following table:

Event	Event ID	Description
InitialEvent	0x5180400	Provider is Initial
MinorEvent	0x5180401	Provider is Minor (% sites down)
MajorEvent	0x5180402	Provider is Major (% sites down)
CriticalEvent	0x5180403	Provider is Critical (% sites down)
GoodEvent	0x5180404	Provider is Good (all sites up)
MinorAlarmEvent	0x5180405	Provider Minor Alarm (% sites down)
MajorAlarmEvent	0x5180406	Provider Major Alarm (% sites down)
CriticalAlarmEvent	0x5180407	Provider Critical Alarm (% sites down)

Service Assurance Method

Multiple events are generated using the condition calculation method. They can be classified in several ways:

- Scope
  - Single test between a Pair of Sites
  - Tests from a Site to all its Destinations
  - Tests from all Sites to all their Destinations connected to the Provider
- Test Type

  - Connectivity (Did the Ping Succeed)
  - Response Time (Did the Ping Succeed within the threshold)
- Test Phase
  - Test Model Creation
  - Test Setup
  - Ping Test Operation
  - Response Time Threshold

Test Phases

Events occur at each phase of test creation, setup, or execution. The results that are reported attempt to determine the most likely root cause. Events that are caused by other events are typically excluded. An example of this result is shown in the following test phases:

1. Test Model Creation
2. Test Setup
3. Ping Test Operation
4. Response Time Threshold

Each successive test phase builds on the previous one. For example, if the Test Model cannot be created, none of the other phases are attempted. Therefore, you see a Test Creation Event (SingleTestCreateFailed) instead of multiple Ping Failure and Response Time Failure events (and alarms). The same is true for the Ping Test Operation and Response Time Threshold phases. If the Ping connectivity test fails (the timeout is 5 seconds), a Response Time failure is not reported. The default value for critical Response Time threshold is 250 milliseconds. Conversely, the Ping test can succeed but the Response Time threshold fails. The event sequence shows the following events:

Event	Event ID
SinglePingTestGood	0x5180604
SingleRTThreshFailed	0x5180607

Assume that you manage 100 sites and have modified the Ping values of minor, major, and critical thresholds to 5, 10 and 20 percent respectively. Next assume that 21 percent of the Ping tests from a site fail. A critical alarm is raised because it exceeds the value of the critical alarm threshold. There are 79 tests that have succeeded. Of these remaining successful ping tests, there are nine Response Time threshold violations. The calculation is done using 9 of 79 tests leading to a failure rate of 11 percent; this percentage is a major alarm status because it exceeds the major alarm threshold. The event sequence that is displayed in this case is as follows:

Event	Event ID
SiteTotalPingTestsCritical	0x5180621
SiteTotalRTThreshMajor	0x5180624

The example demonstrates how the success of each succeeding phase depends on the results of the previous phase. An event sequence can have the following events:

Event	Event ID
SiteTotalPingTestsGood	0x5180618
SiteTotalRTThreshCritical	0x5180625

Single Test Between Sites

The following events are generated for a single site-to-site test.

Event	Event ID	Description
SingleTestCreateGood	0x5180600	Individual test created successfully
SingleTestCreateFailed	0x5180601	Individual test creation failed
SingleTestSetupGood	0x5180602	Individual test setup succeeded
SingleTestSetupFailed	0x5180603	Individual test setup failed
SinglePingTestGood	0x5180604	Individual ping test succeeded
SinglePingTestFailed	0x5180605	Individual ping test failed
SingleRTThreshGood	0x5180606	Individual RT test succeeded
SingleRTThreshFailed	0x5180607	Individual RT test failed

The following tests are part of each event cycle:

- Ping Connectivity
- Ping Response Time

A Ping cycle can pass the Ping Connectivity test but can fail the Response Time test when the ping response returns outside the specified response time window. In this case, the user sees SinglePingTestGood event followed by a SingleRTThreshFailed event.

Summary from One Site to All its Destinations

The following events are generated for a test from a single site to all of its destinations.

Event	Event ID	Description
SiteTotalCreatesGood	0x5180610	All of the site-to-site test that were created are good
SiteTotalCreatesMajor	0x5180612	Major % of site-to-site tests that were created failed
SiteTotalSetupsGood	0x5180614	All of the site-to-site test setups are good
SiteTotalSetupsMajor	0x5180616	Major % of site-to-site test setups failed
SiteTotalPingTestsGood	0x5180618	All of the site-to-site pings are good
SiteTotalPingTestsMinor	0x5180619	Minor % of site-to-site pings failed
SiteTotalPingTestsMajor	0x5180620	Major % of site-to-site pings failed
SiteTotalPingTestsCritical	0x5180621	Critical % of site-to-site pings failed
SiteTotalRTThreshGood	0x5180622	All of the site-to-site RT thresholds are good
SiteTotalRTThreshMinor	0x5180623	Minor % of site-to-site RT thresholds violated
SiteTotalRTThreshMajor	0x5180624	Major % of site-to-site RT thresholds violated
SiteTotalRTThreshCritical	0x5180625	Critical % of site-to-site RT thresholds violated

Summary for All Sites to All Destinations in Provider

The following events are generated for tests from all sites to all destinations in a provider.

Event	Event ID	Description
TotalTestCreatesGood	0x5180700	All Ping tests for provider were created successfully
TotalTestCreatesMinor	0x5180701	Minor % of tests that were created for provider failed
TotalTestCreatesMajor	0x5180702	Major % of tests that were created for provider failed
TotalTestCreatesCritical	0x5180703	Critical % of tests that were created for provider failed
TotalTestSetupsGood	0x5180704	All Ping tests for provider setup ran successfully
TotalTestSetupsMinor	0x5180705	Minor % of test setups for provider failed
TotalTestSetupsMajor	0x5180706	Major % of test setups for provider failed
TotalTestSetupsCritical	0x5180707	Critical % of test setups for provider failed
TotalPingTestsGood	0x5180708	All Ping tests for provider executed successfully
TotalPingTestsMinor	0x5180709	Minor % of Ping tests for provider failed
TotalPingTestsMajor	0x5180710	Major % of Ping tests for provider failed
TotalPingTestsCritical	0x5180711	Critical % of Ping tests for provider failed
TotalRTThreshGood	0x5180712	All RT tests for provider executed successfully
TotalRTThreshMinor	0x5180713	Minor % of RT Threshold for provider violated
TotalRTThreshMajor	0x5180714	Major % of RT Threshold for provider violated
TotalRTThreshCritical	0x5180715	Critical % of RT Threshold for provider violated
DevTestCreatesGood	0x5180800	All RT tests for provider executed successfully
DevTestCreatesMinor	0x5180801	Minor % of tests created for provider failed
DevTestCreatesMajor	0x5180802	Major % of test created for provider failed
DevTestCreatesCritical	0x5180803	Critical % of tests created for provider failed
DevTestSetupsGood	0x5180804	All Ping tests for provider setup successfully
DevTestSetupsMinor	0x5180805	Minor % of test setups for provider failed
DevTestSetupsMajor	0x5180806	Major % of test setups for provider failed
DevTestSetupsCritical	0x5180807	Critical % of test setups for provider failed
DevPingTestsGood	0x5180808	All Ping tests for provider executed successfully
DevPingTestsMinor	0x5180809	Minor % of Ping tests for provider failed
DevPingTestsMajor	0x5180810	Major % of Ping tests for provider failed
DevPingTestsCritical	0x5180811	Critical % of Ping tests for provider failed

DevRTThreshGood	0x5180812	All RT tests for provider executed successfully
DevRTThreshMinor	0x5180813	Minor % of RT Threshold for provider violated
DevRTThreshMajor	0x5180814	Major % of RT Threshold for provider violated
DevRTThreshCritical	0x5180815	Critical % of RT Threshold for provider violated

Event Configuration

DX NetOps Spectrum Event and Alarm Concepts

This section provides a brief overview of conceptual information about Events and Alarms.

About Alarms and Events

DX NetOps Spectrum is a services and infrastructure management system that notifies you of faults on managed elements within the network infrastructure. DX NetOps Spectrum receives alerts from problem areas within the managed infrastructure. DX NetOps Spectrum converts alerts into events and alarms, which are displayed in OneClick event and alarm views. Alerts, events, and alarms let DX NetOps Spectrum notify you about significant occurrences in your IT infrastructure.

Alerts

An *alert* is an unsolicited message from a managed element on a network. A more specific definition of an alert depends on the management protocol that is used to report the alert. In general, DX NetOps Spectrum uses SNMP as the management protocol to communicate with devices on a network. Alerts that an SNMP-compliant device generates are named *traps*.

You can configure managed elements that have enabled the SNMP traps to direct their traps to the host that is running DX NetOps Spectrum. The host receives a trap DX NetOps Spectrum and identifies the model in the DX NetOps Spectrum database that is associated with the managed element, using the source IP address. Next, DX NetOps Spectrum maps the trap to a DX NetOps Spectrum event. The event is then generated and processed.

DX NetOps Spectrum applies special handling to traps that are not mapped to specific DX NetOps Spectrum events. Traps that occur on managed elements that are not modeled at the time the trap is received are also handled differently. For more information, see Modeling and Managing Your IT Infrastructure .

> **NOTE**
> You map the traps from a managed element to specific DX NetOps Spectrum events using the MIB Tools application in OneClick. Perform these mappings before you can create and modify the events and associated alarms using Event Configuration. For more information, see Certifications .

Events

An *event* is a DX NetOps Spectrum object that indicates that something significant has occurred within DX NetOps Spectrum itself or within the managed environment. Events always occur in relation to a model. When DX NetOps Spectrum receives an alert from a managed element on the network, in response, it generates a DX NetOps Spectrum event for the corresponding model if the received trap is mapped to an event.

DX NetOps Spectrum also generates some events automatically. For example, DX NetOps Spectrum generates an event when models are created or destroyed or when DX NetOps Spectrum connects or disconnects from a device application. DX NetOps Spectrum also generates an event when contact with a managed element is established or lost.

DX NetOps Spectrum uses the configuration of the underlying event to process an event instance. For example, the Archive Manager in the Distributed Data Manager (DDM) database can log an event instance. Or an event instance can clear an alarm or can generate another event using an event rule.

Network operators can view the list of current events in a landscape on the OneClick Events tab. For a specific event, you can also view information such as a description of the event and the time it was created.

To map the traps from a device to specific DX NetOps Spectrum events, use the MIB Tools application in OneClick. Then complete event customization using Event Configuration. In Event Configuration, define event processing rules, create the event message to display to users, and set other parameters.

Event Codes

Every event has a unique event code. The event code is a 4-byte integer that is expressed in hexadecimal format.

An event code has two parts:

- The first 2 bytes contain the developer ID of the developer who created the event
- The last 2 bytes identify the event with a unique number relative to all other event codes for that developer

DX NetOps Spectrum assigns event codes to all events created using MIB Tools or Event Configuration. The next available event code is always used as the default code. In Event Configuration, you have the option of overriding the default code and specifying a different one.

> **NOTE**
> The event code 0x10000 represents a null event. This event cannot be generated. However, the null event can be used in an event rule that requires an event code as a parameter.

Alarms

An *alarm* is a DX NetOps Spectrum object that indicates that a user-actionable, abnormal condition exists in a model. DX NetOps Spectrum generates an alarm when a DX NetOps Spectrum event -- typically generated as a result of a received trap -- specifies an alarm creation. DX NetOps Spectrum can generate an alarm that is based on the results of a watch. DX NetOps Spectrum can also send an event in response to an abnormal situation that did not send an event. (For example, a model loses the connection to its managed element).

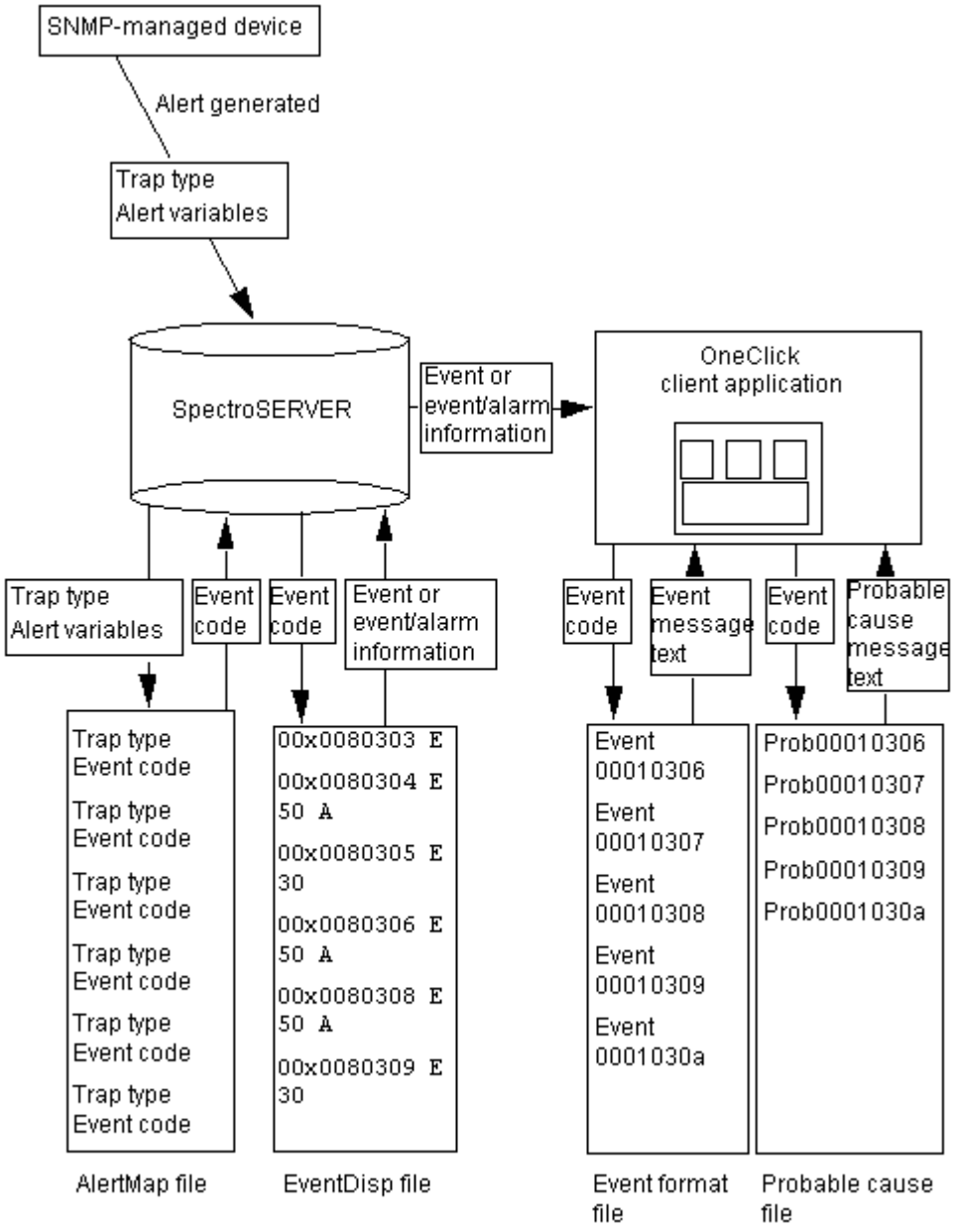
Network operators are alerted to alarms in multiple ways that depend on OneClick configuration. For example, the icon representing the model of the managed element (or a container model for the managed element model) can change color. Or an audio message can announce the new alarm.

Operators can view the current list of alarms in a landscape on the OneClick Alarms tab. For a specific alarm, you can also view detailed information. For example, you can see whether a user has acknowledged the alarm, its symptoms, probable causes, and the recommended corrective actions.

When the abnormal condition that caused the alarm ends, another event automatically clears the corresponding alarm. You can clear the alarm manually and also send any Alarm notifications to external third-party or internal DX NetOps Spectrum applications as appropriate.

You can specify whether an event generates an alarm (and alarm severity) in MIB Tools when you map a trap to an event. However, you use Event Configuration to set more alarm parameters and to change the alarm severity.

The following figure illustrates the flow of alerts, events, and alarms within DX NetOps Spectrum.



Getting Started with Event Configuration

Before You Begin

Before you begin using Event Configuration to create events and alarms, follow any of these steps:

- Obtain a developer ID from CA.
- Map the traps to new DX NetOps Spectrum events using MIB Tools first, if you are adding trap support for a device. (Which is not supported by default in DX NetOps Spectrum).

Obtaining a Developer ID

The first 2 bytes of any event code contains a developer ID. By default, this portion of the event code is the default developer ID that is provided with DX NetOps Spectrum. However, if you are creating events and alarms to support a new device management module, or you are creating a Southbound Gateway integration; we recommend obtaining a unique registered developer ID from CA. A registered developer ID lets you specify event codes for your events that begin with your unique ID. You can easily recognize your custom code in OneClick and avoid potential conflicts with other DX NetOps Spectrum event codes.

To obtain a developer ID, contact CA Support. To be eligible for a developer ID, you must have purchased the Level 1 Toolkit.

NOTE
For information about the toolkit, see Spectrum Integrator. To activate your developer ID, use **SSdbload** with the **-d** option. For more information, see Database Management on loading developer information in.

Mapping Traps to DX NetOps Spectrum Events

When you are creating a management module for a device that is not supported by default in DX NetOps Spectrum. We recommend using MIB Tools to map the traps that the device sends to new DX NetOps Spectrum events. The new events are automatically created when the trap mappings are defined. Follow these steps before you begin using Event Configuration. You can then launch Event Configuration directly from MIB Tools to complete the configuration of the events and associated alarms.

If you create new events using Event Configuration instead of MIB Tools, you cannot use MIB Tools to map traps to them. Instead, you need to manually specify the mappings in the *$SPECROOT*/custom/Events/AlertMap file on each SpectroSERVER in your environment. This rule applies because the process of mapping traps to events using MIB Tools automatically creates new events with unique event codes. You cannot use MIB Tools to map traps to *existing* events that were previously created using Event Configuration.

AlertMap files are ASCII files that store the following Mappings:

- Mappings between the traps that a device sends and DX NetOps Spectrum events
- Mappings between the variable bindings that are sent with a trap and the event variables thatDX NetOps Spectrum generates on a model when the trap is received. Variable bindings can store attribute values in a MIB table, OIDs, or integer bit values.

NOTE
For more information about using MIB Tools to map traps to events, see Certifications .

Preserving Customizations Across Upgrades of DX NetOps Spectrum

Several types of event and alarm configuration files support event and alarm processing in DX NetOps Spectrum, such as: alert mapping files, event disposition files, event format files, event table files, and probable cause files.

The files that are provided with DX NetOps Spectrum to support CA-authored events and alarms are installed in subfolders of the following folders:

```
<$SPECROOT>/SS/CsVendor
<$SPECROOT>/SG-Support
```

When you are customizing CA-authored events and alarms or creating your own events and saving the customizations to one or more landscapes; the event and alarm configuration files that define your customizations are installed in the following folder or in one of its subfolders:

```
<$SPECROOT>/custom/Events
```

Following this procedure ensures that the support files for your custom events and alarms are not overwritten or affected when you upgrade to a newer version of DX NetOps Spectrum.

Starting the Event Configuration Application

You can start Event Configuration from the following locations:

- OneClick Console
- MIB Tools

Starting the Event Configuration Application from the OneClick Console

To load all events that are supported in the landscape (or landscapes in a distributed environment) into the application, start Event Configuration from the OneClick Console.

You can start Event Configuration from the OneClick Console.

Follow these steps:

1. Launch the OneClick Console from the OneClick home page.
2. From the Tools menu, select Utilities, Event Configuration.

Starting the Event Configuration Application from MIB Tools

To add trap support for a device that is not currently supported in DX NetOps Spectrum use MIB Tools, You must map the traps to new DX NetOps Spectrum events and specify those events to generate alarms. After you do the mapping, typically you further customize the events and alarms, which you must do in Event Configuration. For this reason, you can start Event Configuration directly from MIB Tools.

When you start Event Configuration from MIB Tools, only the events that are associated with the traps you select are initially loaded into the application.

You can start Event Configuration from MIB Tools.

Follow these steps:

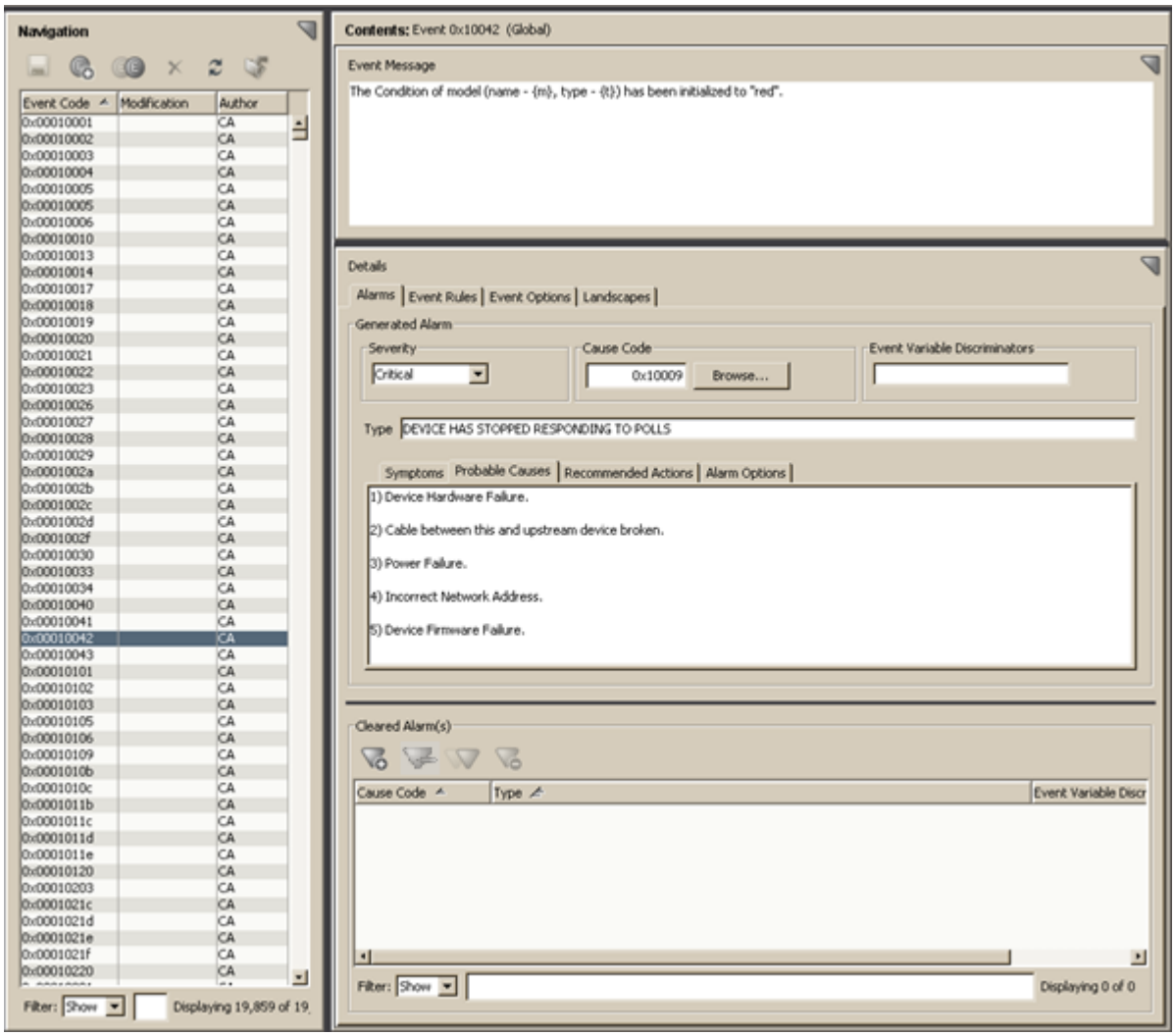
1. In the Navigation panel in MIB Tools, select the MIB containing the traps that have been mapped to the events that you want to configure.

   NOTE
   Map traps to DX NetOps Spectrum events in MIB Tools before you can configure the associated events in Event Configuration. For more information, see information about trap support in Certifications .
2. In the Contents panel, click the Map tab.
3. In the Trap Support table, select the mapped traps for which you want to configure events, and click the edit iconto edit traps for selected items in the trap support table.
   The Event Configuration application is started, and the events that are associated with the traps that you selected display by event code in the Navigation panel.

   NOTE
   Click the Reloads the list of events icon on the Navigation panel, to load all events that are supported in the landscape or landscapes (in a distributed environment).

Overview of the User Interface

You can perform all event and alarm configuration tasks in the Event Configuration main window.



The window is divided into three panels:

- **Navigation:** The Navigation panel lists all of the events that are currently loaded into Event Configuration. The Modified column displays a checkmark next to any event that you have created or modified but not yet saved to a landscape. For more information, see Saving Events to Landscapes.
- **Contents:** The Contents panel displays a customizable event message in the Events tab in OneClick for the event that is selected in the Navigation panel.
- **Details:** The Details panel provides access to the configuration information for the event that is selected in the Navigation panel. If the event generates an alarm, you can also configure the alarm in this panel.

NOTE
You can click the server connection icon to access connection status information for all landscapes in the environment.

Loading All Events from All Landscapes

In order to, load or reload all of the events that are supported in the landscape (or landscapes if the environment is distributed).

On the Navigation panel, to reload the list of events,

click  .

NOTE
Click the Reloads the list of events icon if you start Event Configuration from MIB Tools to work with a limited set of events, and now require access to all events supported in all landscapes.

Save Events to Landscapes

When you save new or modified events (and their disposed actions, such as alarms and event rules) to a landscape, Event Configuration updates the event disposition (EventDisp) files that define them on the SpectroSERVER. The SpectroSERVER flushes all existing event and alarm instances and reloads them using the most recent configuration information.

You can save all events or only selected ones. You can also save the changes to the local landscape only or, in a distributed environment, to some or all of the landscapes.

NOTE
You must save your changes to one or more landscapes before exiting Event Configuration, else the changes you have made are discarded when you exit the application.

WARNING
The save process flushes and reloads all event rules including those event rules that are in the middle of processing. As a consequence, the processing of events by active event rules is aborted, and all associated data (for example, counts for occurrences of contributing events) is lost.

You can save events to the landscape in a single SpectroSERVER environment.

Follow these steps:

1. Do one of the following:
   - To save all modified or created events (and associated alarms) click Save All on the File menu.
   - To save only specific events (and associated alarms) select the events in the table in the Navigation panel and then click Save Selected on the File menu.
2. Click Yes.

You can save events to one or more landscapes in a Distributed SpectroSERVER (DSS) environment.

Follow these steps:

1. Do one of the following:
   - To save all modified or created events (and associated alarms), click Save All on the File menu.
   - To save only specific events (and associated alarms), select the events in the table in the Navigation panel, and then click Save Selected on the File menu.

   If one or more SpectroSERVERs are unavailable. A relevant notification is displayed so you that can cancel the process if wish to.

   NOTE
   If any SpectroSERVERs are unavailable and cannot receive the changes, you can synchronize the events and alarms on all landscapes later.
2. Click Yes to save the events and alarms to available SpectroSERVERs. Alternatively, click No to cancel the process. The Select Landscapes dialog opens. By default, all available SpectroSERVERs are listed in the left list , which means that all available SpectroSERVERs receive the updated events and alarms.
3. If there are available landscapes to which you do not want to save the changes, select them in the left list and click the right-arrow button to move them to the right list.
4. Click OK.

Synchronizing Events in a Distributed Environment

If you are running DX NetOps Spectrum in a distributed environment, update events and alarms using Event Configuration and saving them to landscapes, do any of the following tasks:

- Update the SpectroSERVERs in the environment that do not have the most current events and alarms.
  See Synchronizing Event Disposition Files on SpectroSERVERs.
- Copy the event and alarm support files to all of the OneClick web servers in the environment. See Synchronizing Event and Alarm Support Files on OneClick Web Servers.

Synchronize Event Disposition Files on SpectroSERVERs

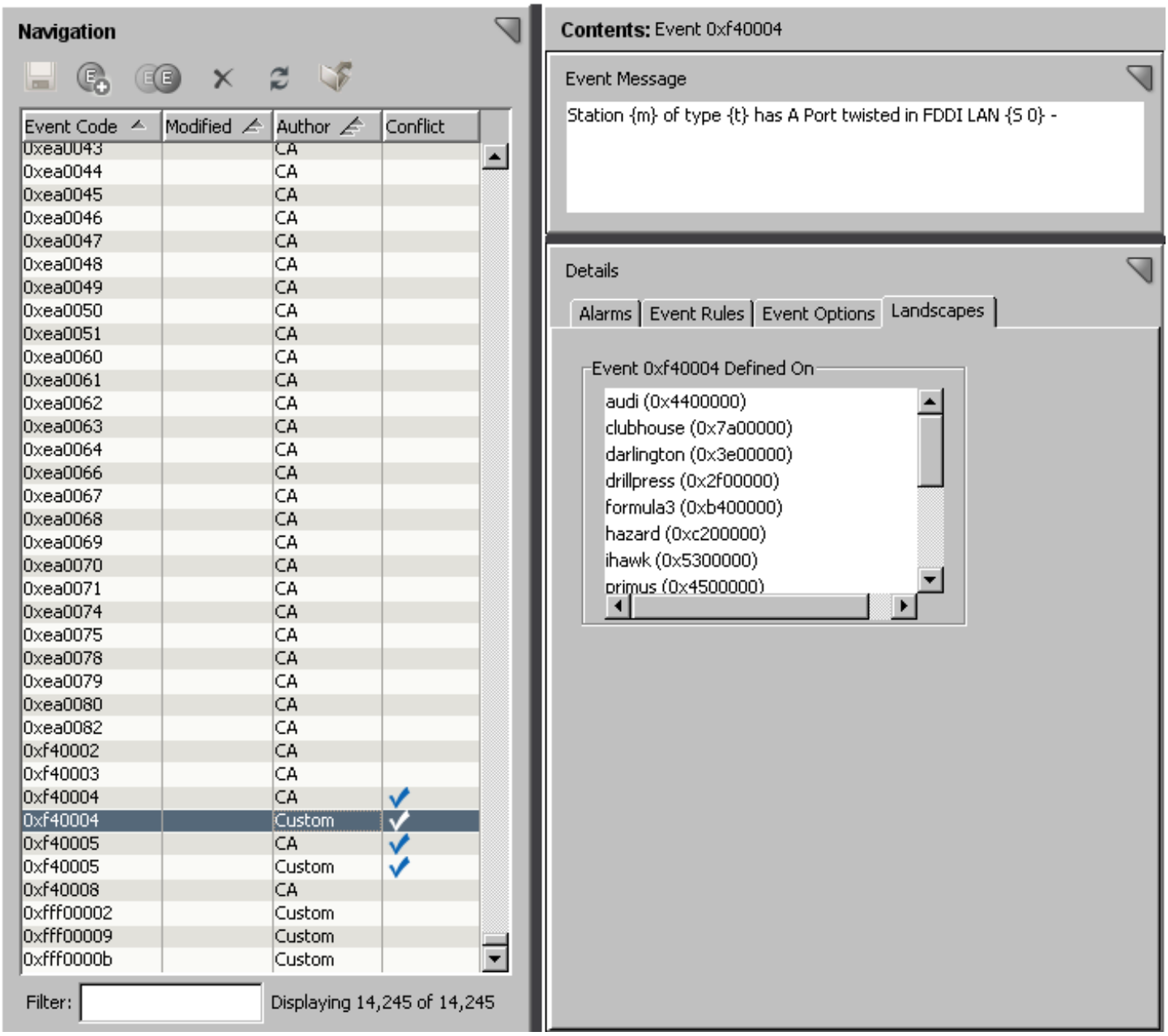
Event and alarm changes are saved to the SpectroSERVERs in a Distributed SpectroSERVER (DSS) environment only when, for example, a server is unavailable. As a result, conflicts in event and alarm configurations arise across the landscapes. Resolve these conflicts by saving the changes to the unavailable servers when they become available. You can perform this task using the synchronization features in Event Configuration.

To begin resolving event (and alarm) conflicts across landscapes, first add the Conflict column to the table of events in the Navigation panel.

Next, examine the Conflict column and note any select marks  .

DX NetOps Spectrum detects an event that is configured differently across two or more landscapes because it finds different event maps for the event in the event disposition files on the SpectroSERVERs. An event with multiple event maps is loaded into Event Configuration once for each unique event map in the DSS environment. The select mark notifies you of each event that has different configurations.

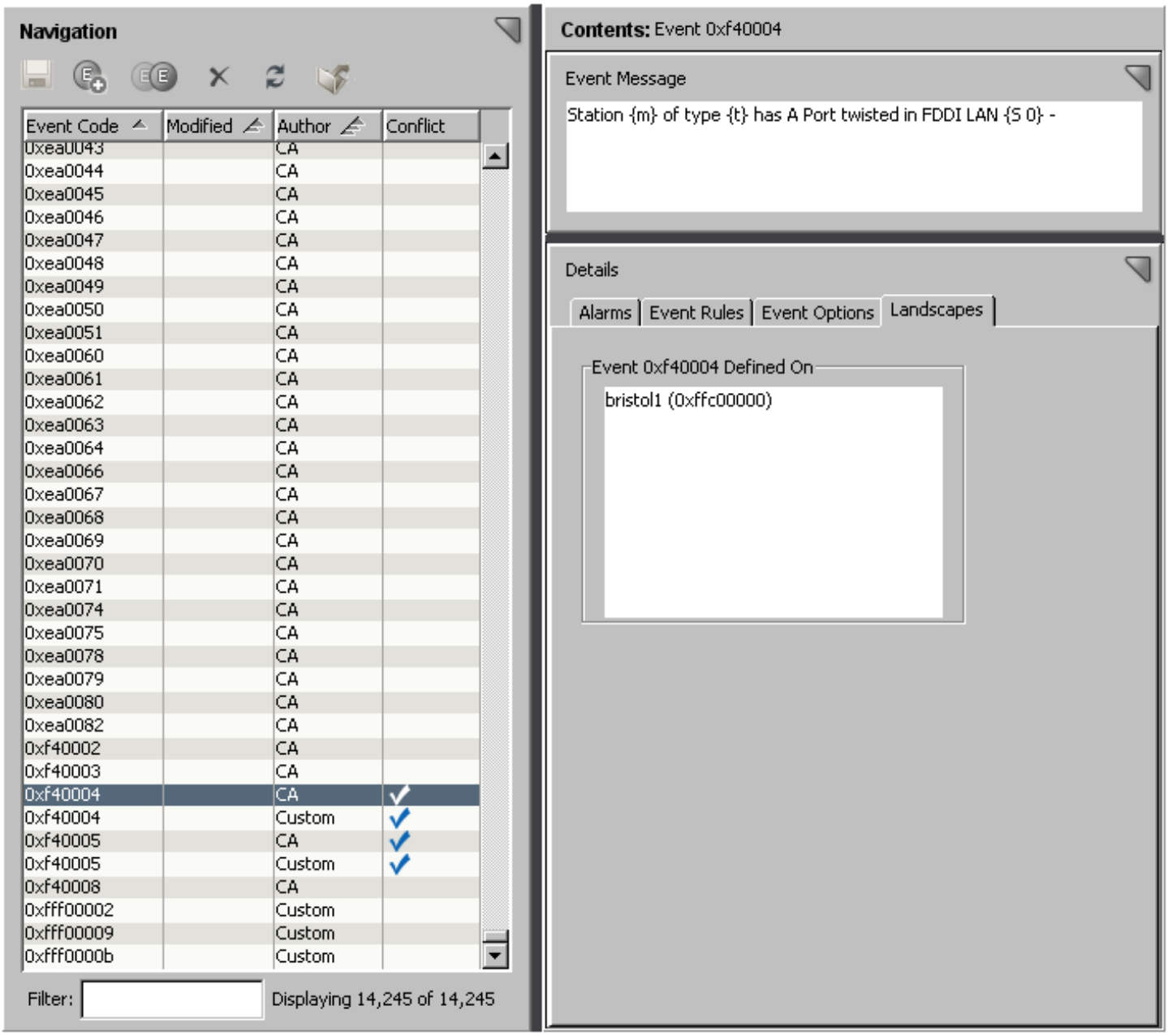
As an example, note the two instances of event 0xf40004 in the following image.



The image shows the result of customizing a predefined event (event 0xf40004) and saving the customization to some but not all landscapes.

You can identify the landscapes to which the custom event has been saved. Select the custom event 0xf40004 and examine the Landscapes tab (displayed automatically in a DSS environment) in the Details panel.

Similarly, you can identify the landscapes to which a predefined event has been saved. Select predefined event 0xf40004 and examine the Landscapes tab. The following image shows the Landscapes tab for the predefined event.



Compare the following information:

- The configurations of the conflicting events
- The landscapes to which the conflicting events have been saved

This information helps you identify the events to save and the landscapes where they are saved to resolve a conflict. Once you have identified these factors, synchronize the landscapes.

Synchronizing the landscapes updates the event disposition (EventDisp) files on one or more SpectroSERVERs. Once updated, they match the event disposition file on the main location server in the DSS environment. Use the Save command if the main location server does not have the event disposition file that you want to use to update the rest of the SpectroSERVERs. For more information, see Saving Events to Landscapes.

Note: You manually designate a main location server when you install DX NetOps Spectrum. For more information about location servers, see Distributed SpectroSERVER Administration .

Follow these steps:

1. Select Synchronize on the File menu.
   The landscapes with event disposition files that differ from the file on the main location server are listed.
2. To synchronize with the main location server, select the landscapes, and click OK.

NOTE
Synchronize the event and alarm support files on the OneClick web servers in the environment.

NOTE
You can also synchronize event and alarm support files between fault-tolerant servers. For more information, see Distributed SpectroSERVER Administration .

Synchronize Event and Alarm Support Files on OneClick Web Servers

Support files are created when you create and configure events and alarms and save them to one or more landscapes using Event Configuration.
The following types of files are created and updated automatically on only the OneClick web server to which you are connected:

- **Event format files**
  Store the event messages that are displayed in OneClick. Every event that DX NetOps Spectrum creates has an event format file.
- **Probable cause files**
  Store the alarm messages that are displayed in OneClick. Every alarm that DX NetOps Spectrum creates and that appears on the Alarms tab in OneClick has a probable cause file.

If you are running multiple OneClick web servers, copy the folders containing the event format files and the probable cause files to the other OneClick servers in your distributed environment. Copy the following folders:

$SPECROOT/custom/Events/CsEvFormat

$SPECROOT/custom/Events/CsPCause

Also copy the contents of these same directories to a custom directory on all of the SpectroSERVERs in your environment in the following circumstances:

- Use the command-line interface (CLI) commands showalarms or showevents.
- Use DX NetOps Spectrum Alarm Notification Manager (SANM).

Updating the Overall Alert and Event System for a Landscape

As described in Saving Events to Landscapes, you can update only the events (and their disposed actions, such as alarms or event rules) on one or more landscapes using the Save commands available on the File menu.

However, to update the overall alert and event system more broadly for a given landscape, in OneClick, click Update Event Configuration on the SpectroSERVER Control subview of the Information tab on the VNM model. This action reloads the following:

- The alert maps that are defined in all custom and predefined AlertMap files.
- The event maps (including event rules) defined in all custom and predefined event disposition files.
- The event procedures that are defined in all custom and predefined event procedure definition files.
- The severity maps defined in all custom and predefined severity mapping files (which are used for alarms that are assigned an alarm severity level of Conditional).
- The event-related resource settings that are defined in the.vnmrc file for the SpectroSERVER.

NOTE
For information about these settings, see Logging Event-Related Errors. For more information about the.vnmrc resource file, see Distributed SpectroSERVER Administration .

- The parse maps defined in all custom and predefined parse map files.

NOTE
For *information about* parse map files, see Host System Resources Management .

WARNING
This update process flushes and reloads all event rules, including those event rules that are in the middle of processing. As a result, the processing of events by active event rules is aborted, and all associated data (for example, counts for occurrences of contributing events) is lost.

Add and Remove Columns from the Events Table

You can modify the event information that is displayed in the table of events in the Navigation panel by adding or removing columns from the table.

For example, if you are working in a Distributed SpectroSERVER (DSS) environment, it can be helpful to add the Conflict column to the table. This information lets you identify whether existing events are configured differently on different landscapes.

NOTE
To filter the events in the table that is based on a specific event property, the corresponding event property column must be displayed. The filtering mechanism checks the text string that you specify against only the text in the *displayed* columns.

Follow these steps:

1. Right-click any column heading.
   The Table Preferences dialog opens.
2. Click the Columns tab, and select the columns that you want to display.

   NOTE
   You can also change the table sort order and font using the controls on the Sort and Font tabs.
3. Click OK.

Logging Event-Related Errors

When you create and configure events and alarms using Event Configuration and then save them to a landscape, the event and alarm processing instructions are written to configuration files referred to as event disposition files.

To help you resolve errors in event disposition files, which result in errors in event processing, you can write errors of different types to log files. You specify which types of errors to log, and the log files to which to write them, using several parameters in the VNM resource file (.vnmrc file) for the SpectroSERVER. see subsections that follow for details on each parameter.

Several types of errors, such as syntax errors, are typically the result of *manual* modifications to event disposition files. To minimize these types of errors, it is recommended that you use Event Configuration, not a manual process, to create and configure events and alarms.

WARNING
If the SpectroSERVER encounters an error in an event map while parsing an event disposition file, that event map is ignored. The event map cannot, therefore, be used to process the associated event.

NOTE
If you modify the event-related.vnmrc parameters that are described in this section, you must reload the parameters on the SpectroSERVER using the Update Event Configuration command on the Information tab

of the VNM model for the changes to take effect. For more information, see Updating the Overall Alert and Event System for a Landscape. Or you can restart the SpectroSERVER, which reloads all of the parameters in the .vnmrc file, not only the event-related ones. For more information about the.vnmrc file, see Distributed SpectroSERVER Administration .

event_disp_error_file

If you set this parameter to the name of a text file, DX NetOps Spectrum writes any syntax errors or other errors that it encounters while parsing an event disposition file to that text file. Use the following syntax:

```
event_disp_error_file=<file name>
```

The text file is created in the *$SPECROOT*/SS folder.

Or you can set the value as follows:

```
event_disp_error_file=stderr
```

This command sends the output to the console window in the DX NetOps Spectrum Control Panel and to the *$SPECROOT*/SS/VNM.OUT file.

NOTE
For more *information about* syntax errors, see Syntax Errors in EventDisp Files.

event_custom_override_warnings

If you set this parameter to TRUE, when DX NetOps Spectrum encounters an event map for an event in a custom event disposition file, and an event map for the same event also exists in an event disposition *file that is provided* with DX NetOps Spectrum, it logs a warning to the text file specified in the event_disp_error_file parameter (if a file is specified). The default value is FALSE.

Setting this parameter to TRUE can be helpful if you must determine which CA-authored events are overridden by your custom events, for example, for troubleshooting purposes. Use the following syntax:

```
event_custom_override_warnings=TRUE
```

NOTE
To set this parameter, you must manually add it to the .vnmrc file.

enable_event_variable_warnings

By default, DX NetOps Spectrum does not log any warnings that are encountered while copying event variable values from one event to another during event rule processing. You can override this default behavior by setting the value of this parameter to TRUE. Use the following syntax:

```
enable_event_variable_warnings=TRUE
```

NOTE
To set this parameter, you must manually add it to the.vnmrc file.

event_duplicate_action_warnings

By default, when DX NetOps Spectrum encounters *identical* (and, therefore, duplicate) event maps for an event within an event disposition file or across multiple event disposition files on a SpectroSERVER, it logs a warning to the text file specified in the event_disp_error_file parameter (if a file is specified).

Typically, this default behavior is desirable. However, you can override it by setting the value of this parameter to FALSE. Use the following syntax:

```
event_duplicate_action_warnings=FALSE
```

NOTE
To set this parameter, you must manually add it to the .vnmrc file.

event_disp_default_log

For the events that you create and manage using Event Configuration, you specify whether they are logged in the Distributed Data Manager (DDM) database using the Store Event in Historical Database event option in Event Configuration. See Specify Event Options for more information.

By default, events that do not have event maps in event disposition files are logged in the DDM database. You can override this default behavior by setting the value of this parameter to FALSE. Use the following syntax:

```
event_disp_default_log=FALSE
```

Typically, most events have event maps in event disposition files. However, there are situations where this behavior is not true. For example, if you mapped a trap to a DX NetOps Spectrum event *manually* instead of using the recommended method of using MIB Tools, and then you inadvertently neglected to define an event map for the event in an event disposition file, the event is still logged in the DDM database depending on the value of this **.vnmrc** parameter.

procedure_error_file

If you set this parameter to the name of a text file, DX NetOps Spectrum writes any errors that it encounters while parsing a procedure file to that text file. Use the following syntax:

```
procedure_error_file=<file name>
```

The text file is created in the *$SPECROOT*/SS folder.

Alternatively, you can set the value as follows:

```
procedure_error_file=stderr
```

This sends the output to the console window in the DX NetOps Spectrum Control Panel and to the *$SPECROOT*/SS/VNM.OUT file.

If the event_disp_error_file parameter is set, the errors that are encountered in procedures that are executed from events are written to the file specified in event_disp_error_file instead of to the file specified in this parameter. Errors that are encountered in other types of procedures, for example, in procedures used for diagnostics, are still written to the file specified in this parameter.

> **NOTE**
> To set this parameter, add it to the.vnmrc file manually.

Event and Alarm Concepts

This section provides a brief overview of conceptual information about Events and Alarms.

About Alarms and Events

DX NetOps Spectrum is a services and infrastructure management system that notifies you of faults on managed elements within the network infrastructure. DX NetOps Spectrum receives alerts from problem areas within the managed infrastructure. DX NetOps Spectrum converts alerts into events and alarms, which are displayed in OneClick event and alarm views. Alerts, events, and alarms let DX NetOps Spectrum notify you about significant occurrences in your IT infrastructure.

Alerts

An *alert* is an unsolicited message from a managed element on a network. A more specific definition of an alert depends on the management protocol that is used to report the alert. In general, DX NetOps Spectrum uses SNMP as the management protocol to communicate with devices on a network. Alerts that an SNMP-compliant device generates are named *traps*.

You can configure managed elements that have enabled the SNMP traps to direct their traps to the host that is running DX NetOps Spectrum. The host receives a trap and identifies the model in the DX NetOps Spectrum database that is associated with the managed element, using the source IP address. Next, DX NetOps Spectrum maps the trap to a DX NetOps Spectrum event. The event is then generated and processed.

DX NetOps Spectrum applies special handling to traps that are not mapped to specific DX NetOps Spectrum events. Traps that occur on managed elements that are not modeled at the time the trap is received are also handled differently. For more information, see Modeling and Managing Your IT Infrastructure .

> **NOTE**
> You map the traps from a managed element to specific DX NetOps Spectrum events using the MIB Tools application in OneClick. Perform these mappings before you can create and modify the events and associated alarms using Event Configuration. For more information, see Certifications .

Events

An *event* is a DX NetOps Spectrum object that indicates that something significant has occurred within DX NetOps Spectrum itself or within the managed environment. Events always occur in relation to a model. When DX NetOps Spectrum receives an alert from a managed element on the network, in response, it generates a DX NetOps Spectrum event for the corresponding model if the received trap is mapped to an event.

DX NetOps Spectrum also generates some events automatically. For example, DX NetOps Spectrum generates an event when models are created or destroyed or when DX NetOps Spectrum connects or disconnects from a device application. DX NetOps Spectrum also generates an event when contact with a managed element is established or lost.

DX NetOps Spectrum uses the configuration of the underlying event to process an event instance. For example, the Archive Manager in the Distributed Data Manager (DDM) database can log an event instance. Or an event instance can clear an alarm or can generate another event using an event rule.

Network operators can view the list of current events in a landscape on the OneClick Events tab. For a specific event, you can also view information such as a description of the event and the time it was created.

To map the traps from a device to specific DX NetOps Spectrum events, use the MIB Tools application in OneClick. Then complete event customization using Event Configuration. In Event Configuration, define event processing rules, create the event message to display to users, and set other parameters.

Event Codes

Every event has a unique event code. The event code is a 4-byte integer that is expressed in hexadecimal format.

An event code has two parts:

- The first 2 bytes contain the developer ID of the developer who created the event
- The last 2 bytes identify the event with a unique number relative to all other event codes for that developer

DX NetOps Spectrum assigns event codes to all events created using MIB Tools or Event Configuration. The next available event code is always used as the default code. In Event Configuration, you have the option of overriding the default code and specifying a different one.

> **NOTE**
> The event code 0x10000 represents a null event. This event cannot be generated. However, the null event can be used in an event rule that requires an event code as a parameter.

Alarms

An *alarm* is a DX NetOps Spectrum object that indicates that a user-actionable, abnormal condition exists in a model. DX NetOps Spectrum generates an alarm when a DX NetOps Spectrum event -- typically generated as a result of a received trap -- specifies an alarm creation. DX NetOps Spectrum can generate an alarm that is based on the results of a

watch. DX NetOps Spectrum can also send an event in response to an abnormal situation that did not send an event. (For example, a model loses the connection to its managed element).

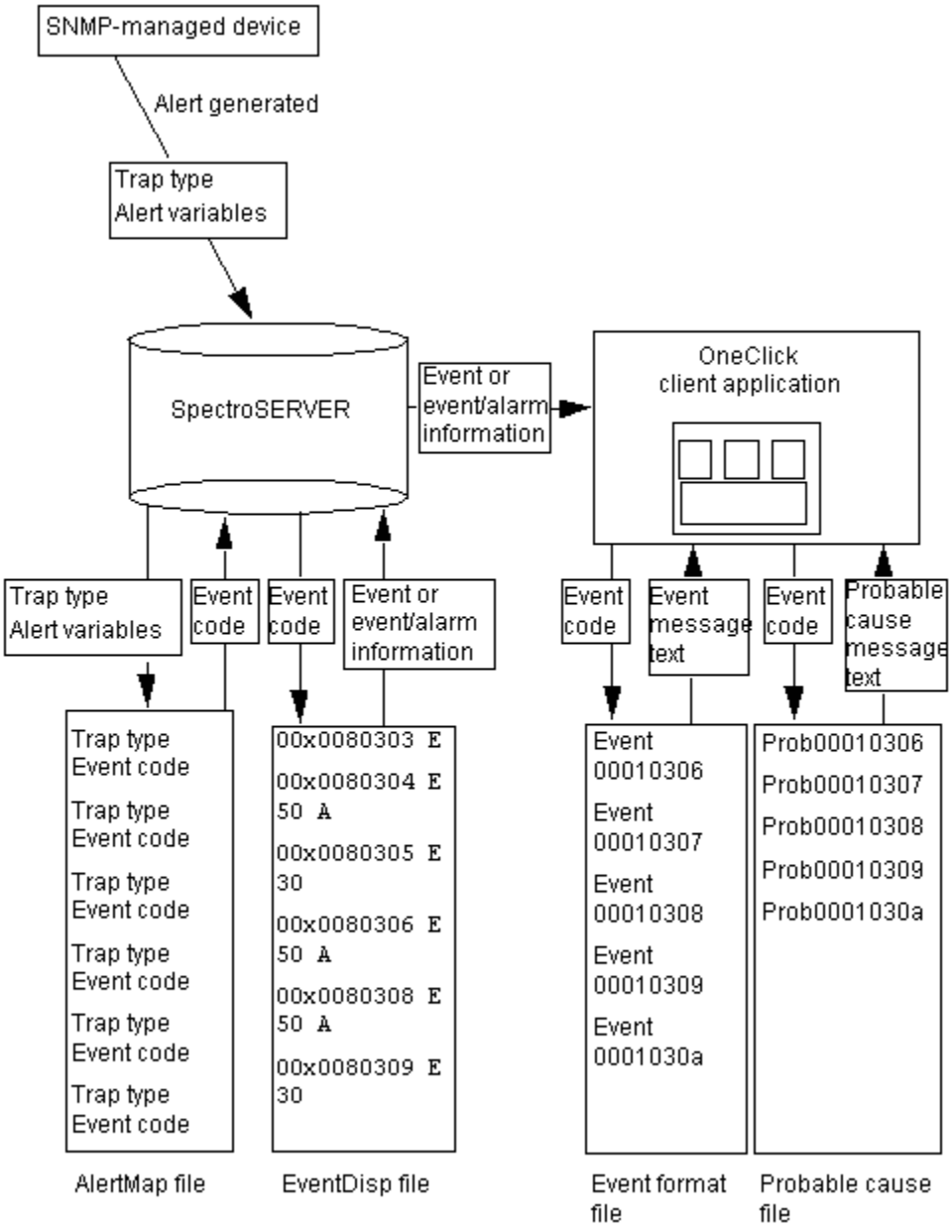
Network operators are alerted to alarms in multiple ways that depend on OneClick configuration. For example, the icon representing the model of the managed element (or a container model for the managed element model) can change color. In addition, an audio message announces the new alarm in both thick client and WebApp.

Operators can view the current list of alarms in a landscape on the OneClick Alarms tab. For a specific alarm, you can also view detailed information. For example, you can see whether a user has acknowledged the alarm, its symptoms, probable causes, and the recommended corrective actions.

When the abnormal condition that caused the alarm ends, another event automatically clears the corresponding alarm. You can clear the alarm manually and also send any Alarm notifications to external third-party or internal DX NetOps Spectrum applications as appropriate.

You can specify whether an event generates an alarm (and alarm severity) in MIB Tools when you map a trap to an event. However, you use Event Configuration to set more alarm parameters and to change the alarm severity.

The following figure illustrates the flow of alerts, events, and alarms within DX NetOps Spectrum.



Working with Events and Alarms

The following sections details all the features and options that you face while working with Events and Alarms in DX NetOps Spectrum.

Finding Events

To find an event, you can filter the list of events in the Navigation panel to include only events with displayed property values that include a specific text string. For example, *if the Type column is displayed*, you can enter “timeout” to filter the

list to include only events that generate alarms that include the word (in uppercase or lowercase) in the alarm type text string (alarm title).

An event that is mapped to a Trap has the Trap Event column checked.

To find events.

1. If necessary, click the Reloads the list of events icon to update the event table in the Navigation panel to include all events in the distributed environment.

   NOTE
   Typically, you must reload the list of events if you started Event Configuration from MIB Tools. Starting Event Configuration in this manner only loads into Event Configuration the specific events you selected in MIB Tools.
2. Verify that the event properties that you want to search against are displayed in the events table. If not, add the appropriate table columns as described in Adding and Removing Columns from the Events Table.

   NOTE
   Only visible table columns are included in the filtering process.
3. In the Filter field, enter the text string to search for in the event table.
   The list of events in the table is filtered to include only those events that have the text string you specified in the displayed property values.

Create Events from Scratch

You can create new events from scratch.

NOTE
Creating a management module for a device that DX NetOps Spectrum does not support by default; use MIB Tools to map traps sent by the device to new DX NetOps Spectrum events before using Event Configuration. (Events are automatically created when you define the trap mappings). You can then launch Event Configuration directly from MIB Tools to configure the events and associated alarms.

NOTE
For more information, see Mapping Traps to DX NetOps Spectrum Events.

To create a new event.

1. In the Navigation panel, click the Creates an event icon
   
   The Create Event dialog appears.
2. Enter an event code or accept the default event code.

   NOTE
   The event code is a 4-byte integer that is expressed in hexadecimal format. The first 2 bytes contain the developer ID, and the last 2 bytes identify the event with a unique number. While the default code is unique, it is recommended that you enter a code beginning with your CA-assigned developer ID. This ID lets you easily recognize your custom code in OneClick and prevents potential conflicts with other DX NetOps Spectrum event codes. The event code 0x10000 represents a null event. This event cannot be generated. However, the null event can be used in an event rule that requires an event code as a parameter.
3. Enter an event message as described in Entering an Event Message. (You can also modify the message after the event is created.)
   Remember that most of the information that a OneClick user receives about an event is through the message text that is affiliated with that event. For this reason, provide as much information about the event as possible in the message.
4. (Optional) Enter the Vendor to specify the developer, vendor, or manufacturer.

NOTE
The Vendor is available in the directory under <$SPECROOT>/custom/Events that contains the EventDisp file. The event options (and other event processing information) are stored in event configuration files referred to as event disposition files.

5. Click OK.
   The new event is added to the table of events in the Navigation panel. The event is displayed in bold. The Modification column displays *New*.

   NOTE
   The event is marked *New*. However, the event is not saved. You can make more updates such as create more events, modify existing events, or delete events, and save all the updates at one time.
6. Configure the event, as described in Configuring Events.
7. (Optional) Add event rules to the event, as described in Creating an Event Rule.
8. Save the changes to one or more landscapes, as described in Saving Events to Landscapes.
   The events appear in the normal font and are not marked in the Modification column.

Create Events from a Copy

You can create new events by copying existing events.

NOTE
Creating a management module for a device that DX NetOps Spectrum does not support by default, use MIB tools to map the traps that are sent by the device to new DX NetOps Spectrum events (automatically created when the trap mappings are defined) before using Event Configuration. You can then launch Event Configuration directly from MIB Tools to configure the events and associated alarms.

NOTE
For more information, see Mapping Traps to DX NetOps Spectrum Events.

To create an event from a copy.

1. In the Navigation panel, select the event that you want to copy, and click the Copies the selected event icon
   
   The Copy Event dialog opens.
2. Enter an event code or accept the default event code.

   NOTE
   The event code is a 4-byte integer that is expressed in hexadecimal format. The first 2 bytes contain the developer ID, and the last 2 bytes identify the event with a unique number. Though the default code is unique (even regarding the event that you are copying), it is recommended that you enter a code beginning with your CA-assigned developer ID. This ID lets you easily recognize your custom code in OneClick and prevents potential conflicts with other DX NetOps Spectrum event codes. The event code 0x10000 represents a null event. This event cannot be generated. However, the null event can be used in an event rule that requires an event code as a parameter.
3. Revise the event message as appropriate for the new event, as described in Entering an Event Message. (You can also modify the message after the event is created.)
   Remember that most of the information that a OneClick user receives about an event is through the message text which is affiliated with that event. For this reason, provide as much information about the event as possible in the message.
4. Click OK.
   The new event is added to the table of events in the Navigation panel. The event displays in bold. The Modification column displays *New*.

NOTE
The event is marked *New*. However, the event is not saved. You can make more updates such as create more events, modify existing events, or delete events, and save all the updates at one time.

5. Configure the event, as described in Configuring Events.
6. (Optional) Add event rules to the event, as described in Creating an Event Rule.
7. Save the changes to one or more landscapes, as described in Saving Events to Landscapes.
   The events appear in the normal font and are not marked in the Modification column.

About Configuring Events

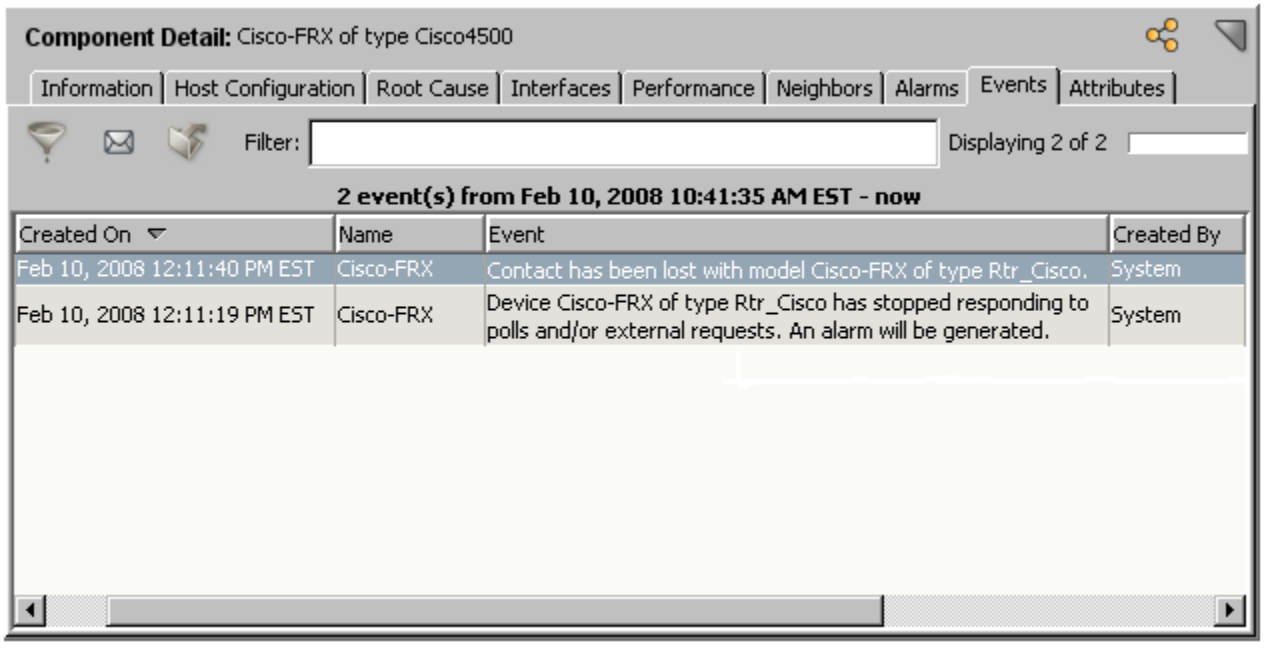
To configure an event, specify the following:

- The message that is displayed to users in OneClick when the event occurs. See Entering an Event Message.
- The scope of the event (global or model type-specific) and whether the event is logged in the Distributed Data Manager (DDM) database by the Archive Manager for historical and reporting purposes. See Specifying Event Options.
- Whether the event generates an alarm. See Configuring Events to Generate Alarms.
- Whether the event clears one or more alarms. See Configuring Events to Clear Alarms.

NOTE
You can also create event rules that are activated (triggered) by an event. For example, some types of events can be tolerated and do not indicate problems if the frequency at which they are generated does not reach a specific threshold, within a specific amount of time. You can create a rule that watches for this scenario, and when it occurs, generates another event (and associated alarm) in response.

Event Messages

The *event message* is the message that is displayed on the Events tab in OneClick when the event occurs.



When you compose an event message, you can use plain text and variables that reference specific data about the generated event. For the descriptions of each variable and the proper syntax to use when including them, see the subsections that follow.

NOTE
The event messages are stored in event configuration files referred to as event format files.

Variable Descriptions and Syntax

This section provides information about the event variables that you can use when you define event messages. When you include an event variable in a message, use the syntax that is defined here.

In the sections that follow, the # sign represents the event variable ID that is mapped to the OID of the variable binding that is sent with the trap. This assignment is made in the OID map in an AlertMap file that MIB Tools automatically creates when you map a trap to an event.

You can construct any type of message using event variable IDs. The only requirement is that the variables that you use in the message are of the proper data type.

NOTE
0x12a63 is a reserved event variable ID that is used for a web context URL.

- **{d “%w- %d %m-, %Y - %T”}**
  This variable is for the date string. This variable must be included in every event message. By including this variable exactly as shown, you tell DX NetOps Spectrum to capture the time and date that the alert is received.

  NOTE
  You do not need to include this variable if you are entering the event message using Event Configuration, as the application automatically inserts the variable in the event format file that it creates when you save the event to a landscape.

  Date/Time specifier options:
  - **%u**
    Use GMT time not local time. Must appear first in format.
  - **%%**
    Writes % to buffer.
  - **%d**
    Writes day-of-month 01..31 to buffer.
  - **%d-**
    Writes day-of-month 1..31 to buffer.
  - **%D**
    Writes date dd/mm/yy to buffer.
  - **%H**
    Writes hour-of-day 00..23 to buffer.
  - **%h**
    Same as %m.
  - **%j**
    Writes day-of-year 1..366 to buffer.
  - **%m+**
    Writes full month name to buffer.
  - **%m-**
    Writes abbreviated month name to buffer.
  - **%m***
    Writes month index 1..12 to buffer.
  - **%m**
    Writes month index 01..12 to buffer.
  - **%M**

Writes minute 00..59 to buffer.
  - **%S**
    Writes second 00..59 to buffer.
  - **%T**
    Writes time hh:mm:ss to buffer.
  - **%w+**
    Writes full weekday name to buffer.
  - **%w-**
    Writes abbreviated weekday name to buffer.
  - **%w**
    Writes weekday index 1..7 to buffer.
  - **%y**
    Writes year 00..99 to buffer.
  - **%Y**
    Writes year 0000..9999 to buffer.
- **{t}**
  This variable inserts the model type name in the message. The (t) variable is defined internally and is not configurable.
- **{m}**
  This variable inserts the model name in the message. The (m) model name variable is defined internally and is not configurable.
- **{e}**
  This variable inserts the event code in the message. The (e) variable is defined internally and is not configurable.
- **{u}**
  This variable inserts the user name in the message. The (u) variable is defined internally and is not configurable.
- **{T <event table file name> #}**
  Inserts a text string that is associated with a MIB table attribute value. The association between the attribute value and the text string is defined in an event table file that MIB Tools creates automatically when you map the trap to the event. For more information, see Referencing Attribute Values in a MIB Table.
- {Y <event table file name> #}
  Inserts a text string that is associated with an OID. The association between the attribute value and the text string is defined in an event table file that MIB Tools creates automatically when you map the trap to the event. For more information, see Referencing OIDs.
- **{Z <event table file name> #}**
  Inserts a text value that is associated with an integer bit value. The association between the integer bit value and the text string is defined in an event table file that MIB Tools creates automatically when you map the trap to the event. For more information, see Referencing Integer Bit Values.
- **{o #}**
  Inserts an object ID.
- **{O #}**
  Inserts an octet string.
- **{X #} or {x #}**
  Inserts an octet string that is displayed in hexadecimal format.
- **{S #}**
  Inserts a text string.
- **{B #}**
  Inserts a Boolean value. Zero denotes false. Any other value denotes true.
- **{I #}**
  Inserts an integer.string that is displayed.
- **{L #}**

Inserts a Counter64 counter.

- **{U #}**
  Inserts an unsigned integer or Counter64 counter.
- **{R #}**
  Inserts a real number in the range: 10E37 to 10E37.
- **{H #}**
  Inserts a 32-bit hex number with a 0x prefix.
- **{K #}**
  Converts a DateAndTime attribute value from an octet string to a text string, and inserts the formatted text string.
- **{G #}**
  Calculates and inserts the device up time that is based on the value of the event variable (#). The value is displayed as days+hours:mins:secs.
- **{D #}**
  Used with an event variable (#), which contains an integer representing the number of seconds from 1969. Converts that value to a string that represents the date and time.
- **{T #}**
  Inserts a date and time.
- **{a #}**
  Inserts an attribute ID.
- **{A #}**
  Inserts an IP address.
- **{V #}**
  Inserts the value of a variable.

Referencing Attribute Values in a MIB Table

If a variable binding that is sent with a trap contains an attribute value from a MIB table, you can use it in an event message. To use it in an event message, apply proper syntax and reference the following information:

- The event variable to which the OID of the variable binding is mapped.
- The event table file that contains the enumerated attribute values and the associated text strings to use in event messages. (MIB Tools automatically creates the Event table files when you map traps to new DX NetOps Spectrum events.)

As an example, assume that you have an event table file that is named BeaconType that associates the following attribute values with corresponding text strings:

```
0x00000001 Reconfiguration
0x00000002 Signal-Loss
0x00000003 Bit-Streaming
0x00000004 Contention-Streaming
0x000000ff None
```

To reference these values in an event message, use the following syntax:

```
{T BeaconType 2}
```

- **T**
  Indicates that you are inserting a MIB table attribute whose values are enumerated in an event table file.
- **BeaconType**
  Specifies the name of the event table file that contains the enumerated values.
- **2**
  Specifies the event variable number that is mapped (in the AlertMap file) to the OID of the variable binding that is sent with the trap. DX NetOps Spectrum takes the value of the event variable and retrieves the corresponding text

string that is defined in the event table file. For example, if the event variable above, 2, stored the value 3, the text “Bit-Streaming” would be rendered in the event message.

Referencing OIDs

If a variable binding that is sent with a trap contains an OID, you can use it in an event message. To use it in an event message,use the proper syntax and reference the following:

- The event variable to which the OID of the variable binding is mapped.
- The event table file that contains the enumerated OID values and the associated text strings to use in event messages. (MIB Tools automatically create Event table files when you map traps to new DX NetOps Spectrum events.)

As an example, assume that you have an event table file that is named NewTable that associates the following attribute values with corresponding text strings:

```
1.3.6.1.4.1.1563.1.2.1.1.3.2.36.2.6 dot6
1.3.6.1.4.1.1563.1.2.1.1.3.2.36.2.5 dot15
1.3.6.1.4.1.1563.1.2.1.1.3.2 dot7
```

To reference these values in an event message, use the following syntax:

```
{Y NewTable 2}
```

- **Y**
  Indicates that you are inserting the value of a variable binding whose possible OID values are enumerated in an event table file.
- **NewTable**
  Specifies the name of the event table file that contains the enumerated values.
- **2**
  Specifies the event variable number that is mapped (in the AlertMap file) to the OID of the variable binding that is sent with the trap. DX NetOps Spectrum takes the value of the event variable and retrieves the corresponding text string that is defined in the Event Table file. For example, if the event variable above, 2, stored the value 1.3.6.1.4.1.1563.1.2.1.1.3.2, the text “dot7” would be rendered in the event message.

Referencing Integer Bit Values

If a variable binding that is sent with a trap contains an OID, you can use it in an event message. To use it in an event message,use the proper syntax and reference the following:

- The event variable to which the OID of the variable binding is mapped.
- The event table file that contains the enumerated integer bit values and the associated text strings to use in event messages.

  NOTE
  When you map traps to new DX NetOps Spectrum events MIB Tools automatically created Event table files.

As an example, assume that you have an event table file that is named NewBitTable that associates the following integer bit values with corresponding text strings:

```
1 dsx1NoAlarm
2 dsx1RcvFarEndLOF
3 dsx1XmtFarEndLOF
4 dsx1RcvAIS
```

To reference these values in an event message, use the following syntax:

```
{Z NewBitTable 2}
```

- **Z**

Indicates that you are inserting the value of a variable binding whose possible integer bit values are enumerated in an event table file.

- **NewBitTable**
  Specifies the name of the event table file that contains the enumerated values.
- **2**
  Specifies the event variable number that is mapped (in the AlertMap file) to the OID of the variable binding that is sent with the trap. DX NetOps Spectrum takes the value of the event variable and retrieves the corresponding text string that is defined in the Event Table file. For example, if the event variable above, 2, stored the value 4, the text “dsx1RcvAIS” would be rendered in the event message.

Example Event Message

The following message is a sample event message:

```
{d "%w- %d %m-, %Y - %T"} A device {m} of type {t} has reported a Firewall trap has occurred. {S 1} contains
 the name of the last trap sent via fw. - (event [{e}])
```

When the message is displayed on the Events tab in OneClick, it is rendered as follows:

- {d "%w- %d %m-, %Y - %T"} is replaced with the date and time
- {m} is replaced with the model name
- {S 1} is replaced with a string value (S for string data type) from a variable binding
- {t} is replaced with the model type
- {e} is replaced with the event code.

Specify Event Options

You can specify the following options for an event:

- Whether the event is logged in the Distributed Data Manager (DDM) database by the Archive Manager for historical and reporting purposes.
  Events for a model that are not logged in the DDM database are displayed on the Events tab in OneClick only if they are generated while the Events tab for that model is displayed.
- Whether the event is global or specific to one or more model types.
  Global events are those that are generated for all models of all model types regardless of the developer who created the model type. Examples of global events include “link down” or “cold start” events.
  If an event is *specific to a model type*, it is generated only for models of specific model types (for example, for a device model type that supports a proprietary MIB).
- The Vendor field appears only for those events that are defined for a vendor. The Vendor field appears as read-only only for events that are defined under a vendor directory. This field is configurable for a new event until the event has been saved, then it appears as read-only.

  NOTE
  The Event options (and other event processing information) are stored in event configuration files referred to as event disposition files.

To specify options for an event.

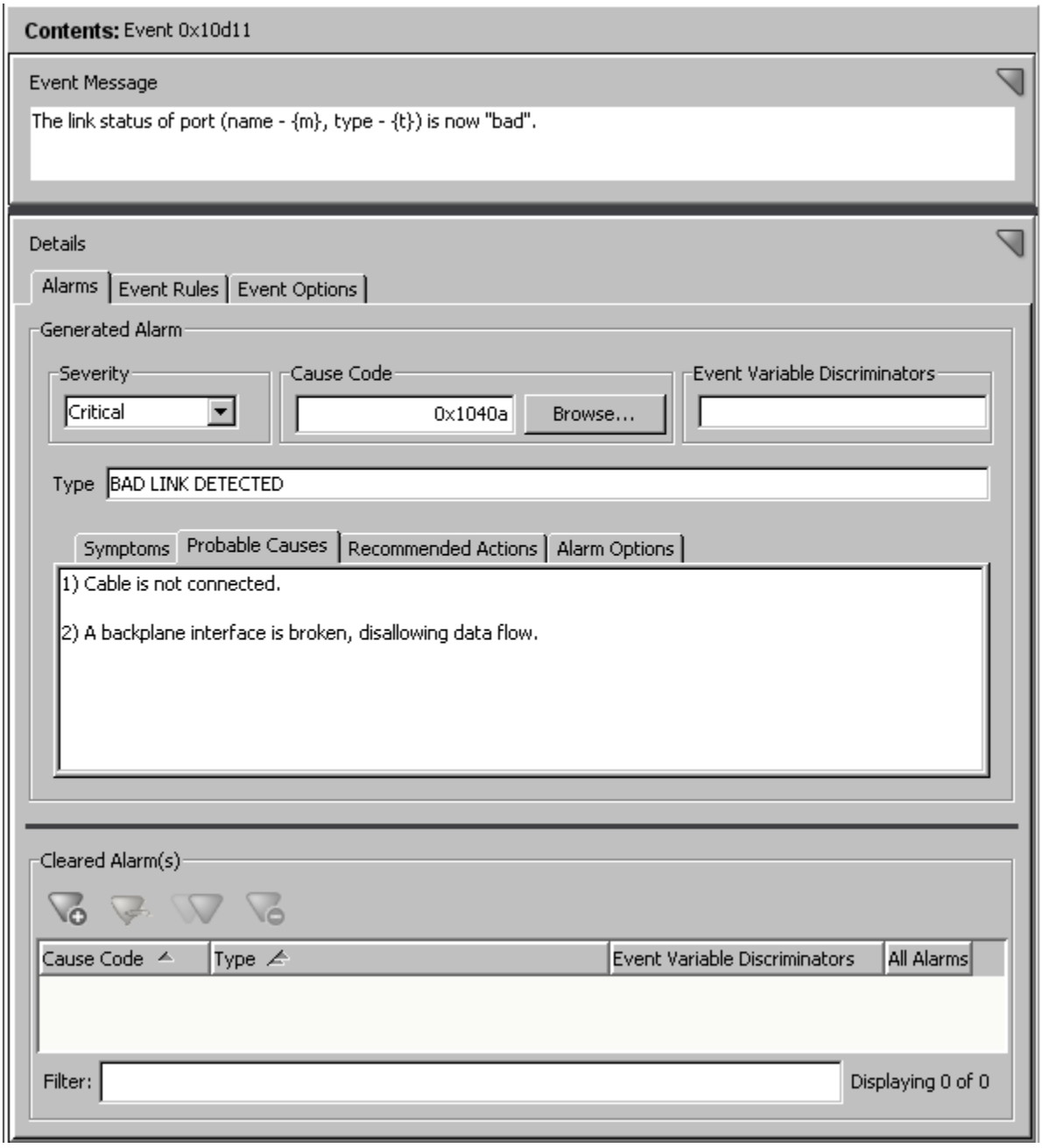
1. Select the event in the Navigation panel.
2. In the Details panel, click the Event Options tab.
3. If you want the event to be logged in the Distributed Data Manager (DDM) database by the Archive Manager, select Store Event in Historical Database.
4. Under Scope, specify the scope of the event:

- If the event is global, select Global.
- If the event is specific to a model type, select Model Type. Then, in the Select Model Type dialog, select the name of one or more model types to which the event applies, and click OK. (Use the CTRL key to select multiple model types.)

  NOTE
  Changing the scope of an event does not modify the event; instead, a duplicate event with the same event code but a different event scope is automatically created. As a result, when you save both events to a landscape, two event maps in two different event disposition files are created. This feature lets you specify event processing for the event and apply those instructions globally, and then override those processing instructions, for the same event, for specific model types. When DX NetOps Spectrumprocesses an event, the event maps in model type-specific event disposition files take precedence over the event maps for the same events in global event disposition files. If you change the scope of an event and you do not require the original event, you can delete it if it is a custom event.

Configure Events to Generate Alarms

You can specify that the currently selected event generates an alarm, and you can configure the alarm itself using the Alarms tab in the Details panel. The Details panel is shown in the following image:



When you configure an event to generate an alarm and then save that change to a landscape, you create a mapping between the event and the alarm in a configuration file, an *event disposition file*.

Follow these steps:

1. Select the event in the Navigation panel, and then click the Alarms tab in the Details panel.

   NOTE
   Under Generated Alarm, the value for Severity is None, indicating that the event does not generate an alarm.
2. Select an alarm severity other than None from the Severity list. For more information, including descriptions of the different severity levels, see Specifying an Alarm Severity.

3. (Optional) For Cause Code, change the alarm cause code (the 8-digit, hexadecimal code that identifies the cause of the alarm). For more information, see Specifying an Alarm Cause Code.

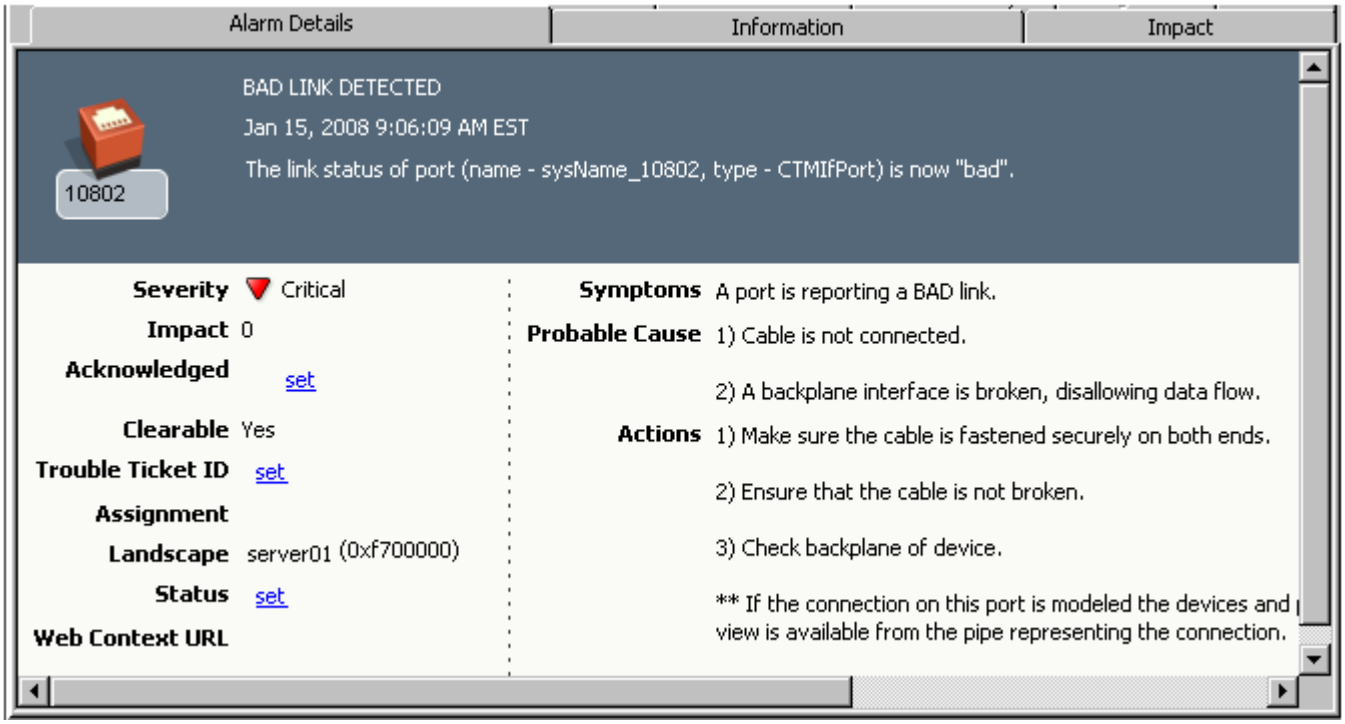
   NOTE
   Events that generate alarms typically use their event codes as alarm cause codes. The event code of the event is, therefore, the default alarm cause code of the alarm.

4. For Event Variable Discriminators, enter a comma-separated list of event variable IDs if you want to use the values of the variables in the event to determine whether to generate the alarm.
   Enter each ID separately; ranges of IDs are not supported.

   NOTE
   By default, DX NetOps Spectrum does not generate a new alarm each time the same event occurs if an alarm already exists for that event on the model. You can use event discriminators or alarm options to change this default behavior.

   For example, if the event generates alarm 0x3b10011, and you enter “1,3” for Event Variable Discriminators, and if an alarm 0x3b10011 already exists on the model, a new alarm is generated ONLY if the values for *both* event variables 1 and 3 are *different* in the new event instance as compared to current alarms on the model generated from the same event.
   For more help with this step, see Using Event Variable Discriminators to Generate Alarms.

5. For Type, enter a text string that identifies the type of the alarm, for example, “BAD LINK DETECTED.”
   The text string that you enter for Type is displayed as the alarm title in OneClick, as shown in the following image. For enhanced readability in OneClick, enter the text string in capital letters.



6. Specify the symptoms, probable causes, and recommended corrective actions for the alarm, respectively, on the Symptoms, Probable Causes, and Recommended Actions tabs. This information is displayed on the Alarm Details tab in OneClick, as shown in the preceding image.

   NOTE
   For more information, see Specifying Symptoms, Causes, and Recommended Actions.

7. Click the Alarm Options tab, and specify advanced options for the alarm. For more information, see Specifying Alarm Options.

8. To configure the selected event to also *clear* one or more alarms, specify the alarms in the Cleared Alarms area of the Contents panel.
   For more information, see Configuring an Event to Clear Alarms.

Specify an Alarm Severity

The following lists the alarm severity levels that are used in DX NetOps Spectrum. Each severity level is associated with a color-coded condition that is displayed on the model. When an alarm of the specified severity is asserted on a model, the condition color is displayed on the model's icon to reflect the alarm status.

- **Normal (0)**
  Color-coded condition: Green
  Indicates that:
  – Contact has been made with the device, and the device is operating typically.
  – A Normal Alarm is generated.
  – If an event generates an alarm but a severity for the alarm is not specified (for instance, if you have created the supporting EventDisp configuration file manually and inadvertently omitted a severity), DX NetOps Spectrum assigns normal severity status to the alarm.
- **Minor (1)**
  Color-coded condition: Yellow
  Indicates that an abnormal situation exists, but no immediate action is required. This level of severity is also used for alarms created only to convey information, such as "Duplicate IP."
- **Major (2)**
  Color-coded condition: Orange
  Indicates that a loss of service has occurred or is impending. Action is required within a short period.
- **Critical (3)**
  Color-coded condition: Red
  Indicates that a loss of service has occurred and immediate action is required.
- **Maintenance (4)**
  Color-coded condition: Brown
  Indicates that the device has been taken offline for maintenance purposes.
- **Suppressed (5)**
  Color-coded condition: Gray
  Indicates that the device cannot be reached due to a known error condition that exists on another device.
- **Initial (6)**
  Color-coded condition: Blue
  Indicates that contact with the device has not yet been established.
- **Variable**
  Color-coded condition: N/A; evaluates to a severity that has a color-coded condition.
  Lets you assign an alarm severity that is based on the value of a variable binding. For example, if the value of the variable binding is 1, then the alarm is assigned a severity level of Minor.
  To use this option, do the following:
  – For Severity, select Variable.
  – For Event Variable, specify the ID of the event variable that stores the variable binding value to use to determine the severity level of the alarm. You must specify a variable whose possible values are enumerated and directly correspond to the numeric severity levels used by DX NetOps Spectrum (identified in the first column in this table).
- **Conditional**
  Color-coded condition: N/A; evaluates to a severity that has a color-coded condition.
  Lets you assign an alarm severity that is based on the value of a variable binding. You can also select the set of enumerated values and corresponding DX NetOps Spectrum alarm severity levels to use. The Color coded condition is useful if, for example, the variable binding defines a set of alarm severity levels that differ from those used in DX NetOps Spectrum.

To use this option, do the following:

- For Severity, select Conditional.
- For Event Variable, specify the ID of the event variable that stores the variable binding value to use to determine the severity level of the alarm. Specify a variable whose possible values are enumerated. The actual value in the event variable is used as the key to look up a corresponding DX NetOps Spectrum alarm severity level in a severity mapping file.
- Below the severity level drop-down list, select the file that contains the user-defined mappings of variable binding values and DX NetOps Spectrum alarm severity levels.

Enable or Disable Alarms of a Severity Type

Alarms use a large number of resources such as memory and processing time. DX NetOps Spectrum lets you disable an alarm of a severity type to reduce the impact on system performance. The disabled alarm is available internally but you cannot view it in the user interface. The disabled alarm does not support alarm attributes like discriminators. By default, alarm types with severity levels Initial and Suppressed are set as disabled. All other alarm types continue to exist as regular alarms.

Note: Normal severity alarms do not support discriminators.

To disable an alarm of a severity type:

1. Select the VNM Model.
2. Click the Information tab and expand the Alarm Management section.
3. Right-click the alarm and select Disable.
   The alarm is disabled and the change takes effect immediately. The existing alarms of the selected severity type continue to be regular or internal alarms until they are cleared.

To enable an alarm of a severity type:

1. Select the VNM Model.
2. Click the Information tab and expand the Alarm Management section.
3. Right-click the alarm and select Enable.
   The alarm is enabled and the change takes effect immediately. The existing alarms of the selected severity type continue to be regular or internal alarms until they are cleared.

- Even after enabling alarms, verify that the client-side filters are adjusted, before you can view the alarm in the client applications.
- We do not recommend enabling the Suppressed alarm type as it can affect performance adversely.

Specify an Alarm Cause Code

An *alarm cause code* is an 8-digit, hexadecimal code that identifies the probable cause of the alarm. As you save the event and associated alarm to a landscape, a mapping between the event code and the associated alarm code is added to the event configuration file that determines how to process the event, referred to as the event disposition file. This mapping is the mechanism by which DX NetOps Spectrum identifies whether or not to generate an alarm for an event, and if so, which one.

As a convention, events that generate alarms typically use their event codes as alarm cause codes, and for this reason, the event code of the event is the default alarm cause code of any alarm. However, you can change the alarm cause code if desired. For example, you want to change the alarm cause code, if you have a more generic existing alarm and want it to be generated whenever the event occurs.

The alarm cause code is also used to name the underlying probable cause file that contains the alarm-related messages (symptoms, causes, and recommended actions) to display in OneClick when the alarm occurs. Each probable cause file is named Prob<alarm_cause_code>, where <alarm_cause_code> is the code you specify on the Alarms tab in Event Configuration.

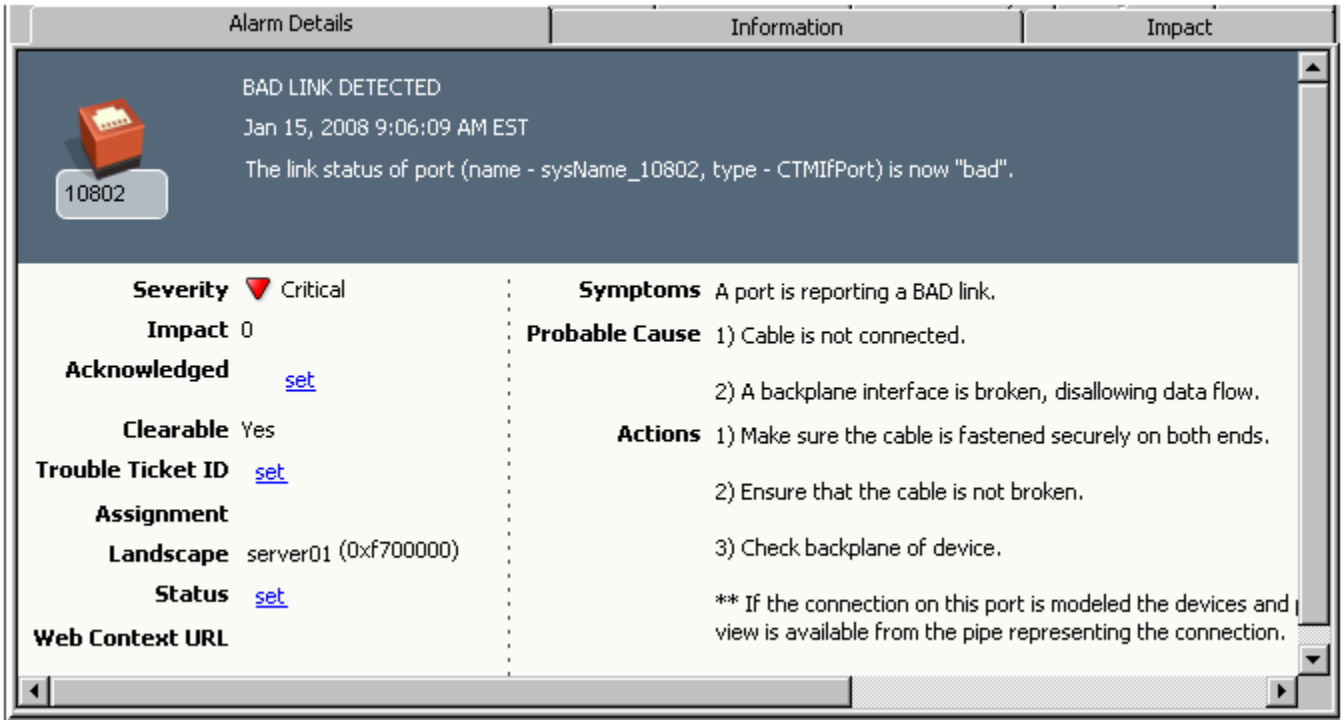
To specify an alarm cause code, do one of the following:

- Accept the default code, which readily identifies the alarm with the event that generates it. If the supporting probable cause file does not exist, it is created automatically by Event Configuration when you save the event and alarm changes to a landscape.
- Click Browse, and in the Select Alarm Cause Code dialog, select an existing code. The displayed list includes all of the alarm cause codes for all loaded events that generate alarms. In this case, the supporting probable cause file is updated automatically when you save the event and alarm changes to a landscape.
- Enter a new 8-digit, hexadecimal value. If the supporting probable cause file does not exist, it is automatically created by Event Configuration while saving the event and alarm changes to a landscape.

NOTE
If you create an alarm-generating event, save it to one or more landscapes and later, change the alarm cause code to a new code and then save the changes to the landscapes. Event Configuration automatically creates a new probable cause file that is named based on the new alarm cause code. However, to remove the probable cause file that is named using the old code (if it is not used by any other alarm-generating events), you must remove it manually.

About Specifying Symptoms, Causes, and Recommended Actions

If an event generates an alarm, you must supply several plain text messages that describe the symptoms, probable causes, and recommended corrective actions for the alarm. These messages are displayed on the Alarm Details tab in OneClick, as shown in the following image.



The text messages that you enter for the symptoms, probable causes, and the recommended corrective actions for an alarm, are stored in an alarm configuration file referred to as a probable cause file. The fields that support traps get auto-populated. For example, when you create a new alarm the trap-specific Alarm Title is auto-populated assuming the MIB in which the trap is defined is found in the MIB database.

Specify Alarm Options

You can specify the following advanced options for an alarm:

- **Alarm is Persistent**
  If this option is selected, the alarm is retained in memory, in case the SpectroSERVER is shut down and restarted.
- **Alarm is User Clearable**
  If this option is selected, users can clear the alarm
- **Generate a Unique Alarm for Each Event**
  By default, DX NetOps Spectrum does not generate a new alarm each time the same event occurs if an alarm already exists for the event on the model.
  If this option is selected, a unique alarm is generated each time the same event occurs.

Using Event Variable Discriminators to Generate Alarms

When an event triggers an alarm in a model, DX NetOps Spectrum does not generate a separate alarm if the same event recurs on the model. This default behavior prevents an event that can occur multiple times due to the same condition. You can, however, configure multiple alarms to occur when the conditions of the event change. You can specify this behavior using event variable discriminators.

Event variable discriminators are numeric values that refer to the IDs of the event variables in an event. In turn, the event variable IDs are mapped (in the AlertMap file) to the OIDs of the variable bindings that are sent with the trap. The discriminators let DX NetOps Spectrum determine to generate alarms for multiple instances of the same event, that is based on the values of variable bindings that are sent with the traps.

You can configure an event to generate an alarm and also specify one or more discriminators. The alarm is generated when the values of the referenced event variables are *different*. You can thus, specify alarms to be generated for distinct event instances (with the same event code) in spite of an existing alarm in the model.

For example, to generate alarm 0x3b10011 you have configured event 0x3b10011, and have specified that event variables 1 and 3 are event variable discriminators. This configuration means that, if an alarm 0x3b10011 already exists on the model, another alarm is not generated.

However, if the following condition is met, a new alarm is generated:

The values for *both* event variables 1 and 3 in the new event instance are *different* compared to current alarms on the model that is generated from the same event.

When you are configuring an alarm, specify the event variable discriminators by entering a comma-separated list of IDs, for example:

```
1,3,5
You must enter each ID; ranges of IDs are not supported.
```

> **NOTE**
> The Event discriminators cannot be specified for normal, maintenance, suppressed, or initial severity alarms.

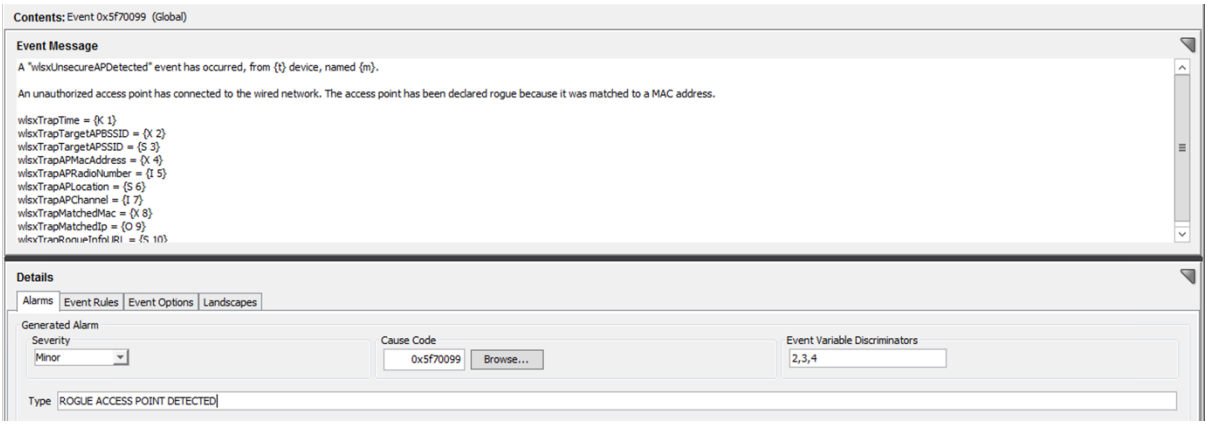
Creating Dynamic Alarm Title

You can modify the alarm to include the event variables in the alarm title. You need not update the event configuration or restart the spectroSERVER after modifying the alarm title. The updated alarm title is applied to the traps that are received after the change. You can only add Event varbinds to alarm title and not DX NetOps Spectrum attribute values. The new alarm title overwrites the dynamic alarm title customizations in EventDisp EventRules.

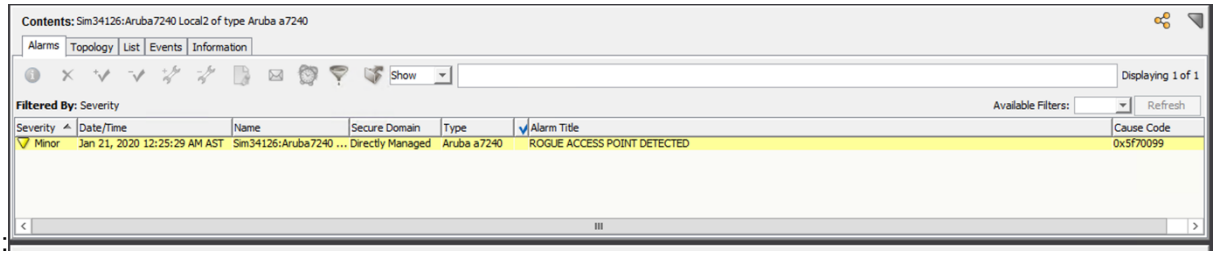
Example: Create a Dynamic Alarm Title

The following example shows how the alarm title looks before and after including trap var binds in the title:

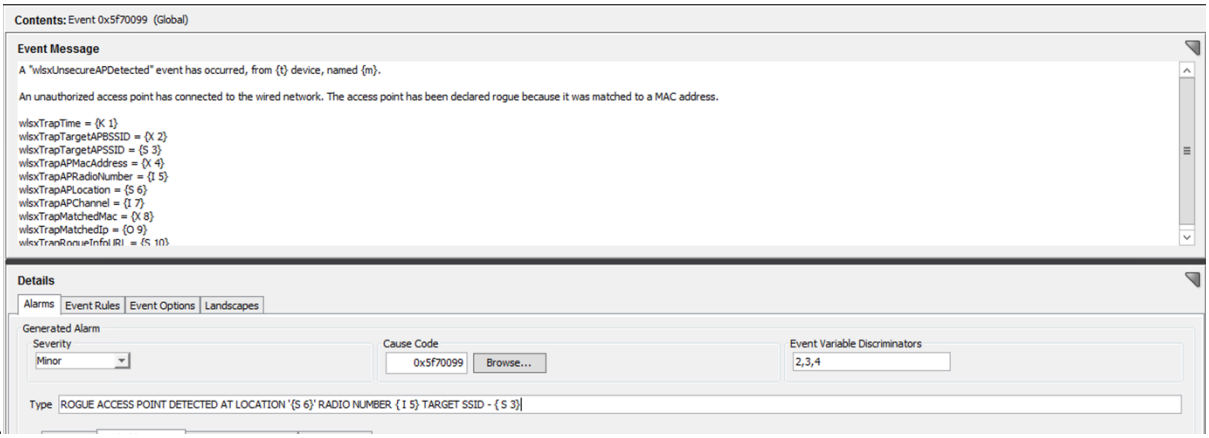
- Default alarm title, before including var binds in the title:



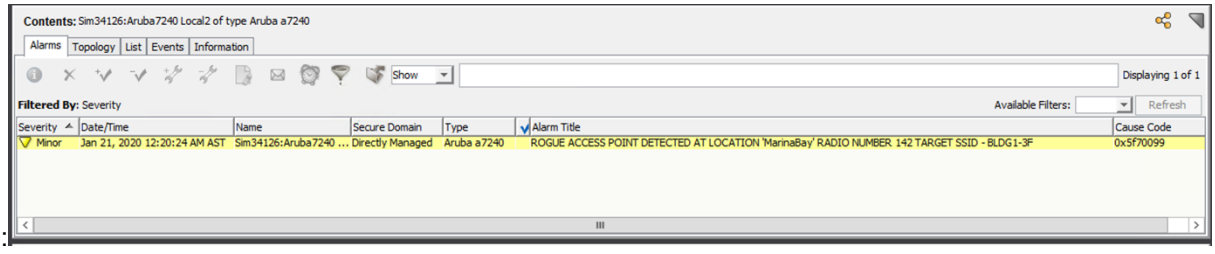
- Alarm title that is created when the trap is received:



- Change the alarm title to include the trap varbinds like ‘SSID’,’ Radio number’ and ‘Location’ in the alarm title. You can include multiple trap varbinds in the alarm title.



- The new alarm title when the new trap is received post-implementation of the change:



Configuring Alarm Severity Mappings for the Conditional Severity Level

A *severity mapping file* is an ASCII file that is used to determine the actual severity of an alarm instance to generate when the alarm configuration specifies an alarm severity of Conditional.

The *Conditional alarm severity* lets you assign the alarm severity that is based on the value of a variable binding that is sent with the trap which triggered the event. It also allows you to select the set of mappings between values and alarm severity levels to use when determining the actual alarm severity.

The severity mapping file defines the mappings between possible variable binding values and DX NetOps Spectrum alarm severity levels. You can create severity mapping files and can customize the ones that are provided by default with DX NetOps Spectrum.

Create Alarm Severity Mappings for the Conditional Severity Level

If you create a severity mapping file, the file is installed in the following location when you save the change to a landscape:

<$SPECROOT>/custom/Events/<vendor_directory>/SeverityMaps/<file name>

Where <vendor_directory> is the directory and <file_name> is the file name that you specified when you created the mapping file using Event Configuration. If you do *not* specify a directory while creating the file, it is installed in the following folder instead:

<$SPECROOT>/custom/Events/CA/SeverityMaps/<file name>

> **WARNING**
> Custom severity mapping files override those Severity files provided with DX NetOps Spectrum, the latter of which are located in <$SPECROOT>/SS/CsVendor/<vendor_directory>/SeverityMaps.

To create an alarm severity mapping file

1. Display the event that generates the alarm, and click the Alarms tab.
2. If you have not already done so, do the following:
   a. For Severity, select Conditional.
   b. For Event Variable, enter or select the event variable that contains the variable binding value to use to determine the alarm severity.
3. Select any severity mapping file and click Configure.
   The Configure Conditional Alarm Severity dialog opens, displaying the contents of the selected severity mapping file.
4. Click
   
   The severity mapping file add dialog opens.
5. Complete the fields as follows:
   – **Directory**
     Specifies the directory in which to save the severity mapping file.
   – **Name**
     Specifies a name for the severity mapping file; ideally the name of the variable binding whose values are enumerated in the file.
6. Add the mappings between the string representations of the variable binding values and DX NetOps Spectrum alarm severity levels (described in Specifying an Alarm Severity) as follows:

   > **NOTE**
   >
   > For Alarm Severity value zero (‘0’), Spectrum generates a normal alarm along with an event. So, if you do not want a normal alarm to be generated for the Severity value zero, do not include the string in the Severity Maps file.

   – To add a mapping, enter a unique string value for String, select a severity for Severity, and click Add.

NOTE
If you want the alarm to be cleared -- instead of generated -- if the corresponding variable binding value is sent, select Clear.

- To change a mapping, select the mapping, change the value for String or for Severity (or both), and click Modify.
- To remove a mapping, select the mapping, and click Remove.

NOTE

DX NetOps Spectrum treats the entries in SeverityMap file as regular expressions. To match the exact variable binding value to a DX NetOps Spectrum alarm severity level, we recommend that you use word boundary characters ('\b') before and after the varbind value string in SeverityMap file.

7. Click OK twice.

Modifying Alarm Severity Mappings for the Conditional Severity Level

You can modify your custom severity mapping files and the mapping files provided by default with DX NetOps Spectrum at any time.

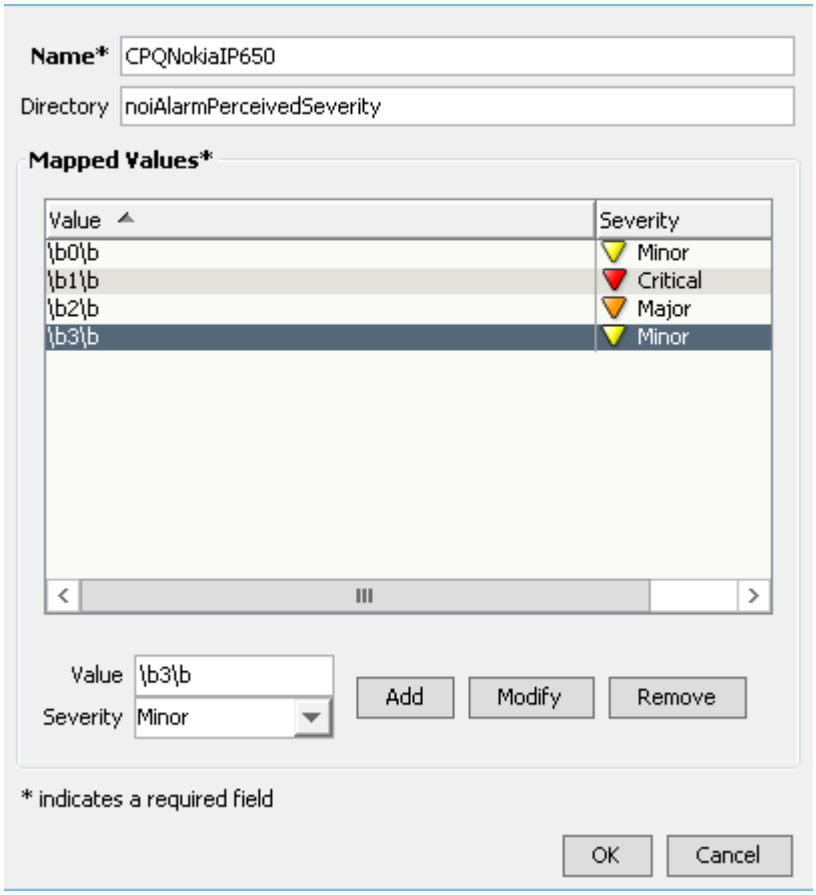
If you modify a severity mapping file provided with DX NetOps Spectrum, a customized version of the file is installed in the following location when you save the change to a landscape:

<$SPECROOT>/custom/Events/<vendor_directory>/SeverityMaps/<file name>

WARNING
Custom severity mapping files override those provided with DX NetOps Spectrum, the latter of which are located in <$SPECROOT>/SS/CsVendor/<vendor_directory>/SeverityMaps.

To modify an alarm severity mapping file, Follow these steps:

1. Display the event that generates the alarm, and click the Alarms tab.
2. If you have not already done so, do the following:
   a. For Severity, select Conditional.
   b. For Event Variable, enter or select the event variable that contains the variable binding value to use to determine the alarm severity.
3. Select the severity mapping file to use to determine the actual severity level of the alarm, and click Configure.
   The Configure Conditional Alarm Severity dialog opens, displaying the contents of the selected severity mapping file.
4. Click
   
   .
   The severity mapping file edit dialog opens.



5. Modify the mappings between the string representations of the variable binding values and DX NetOps Spectrum alarm severity levels as follows:
   - To change a mapping, select the mapping, change the value for String or for Severity (or both), and click Modify.

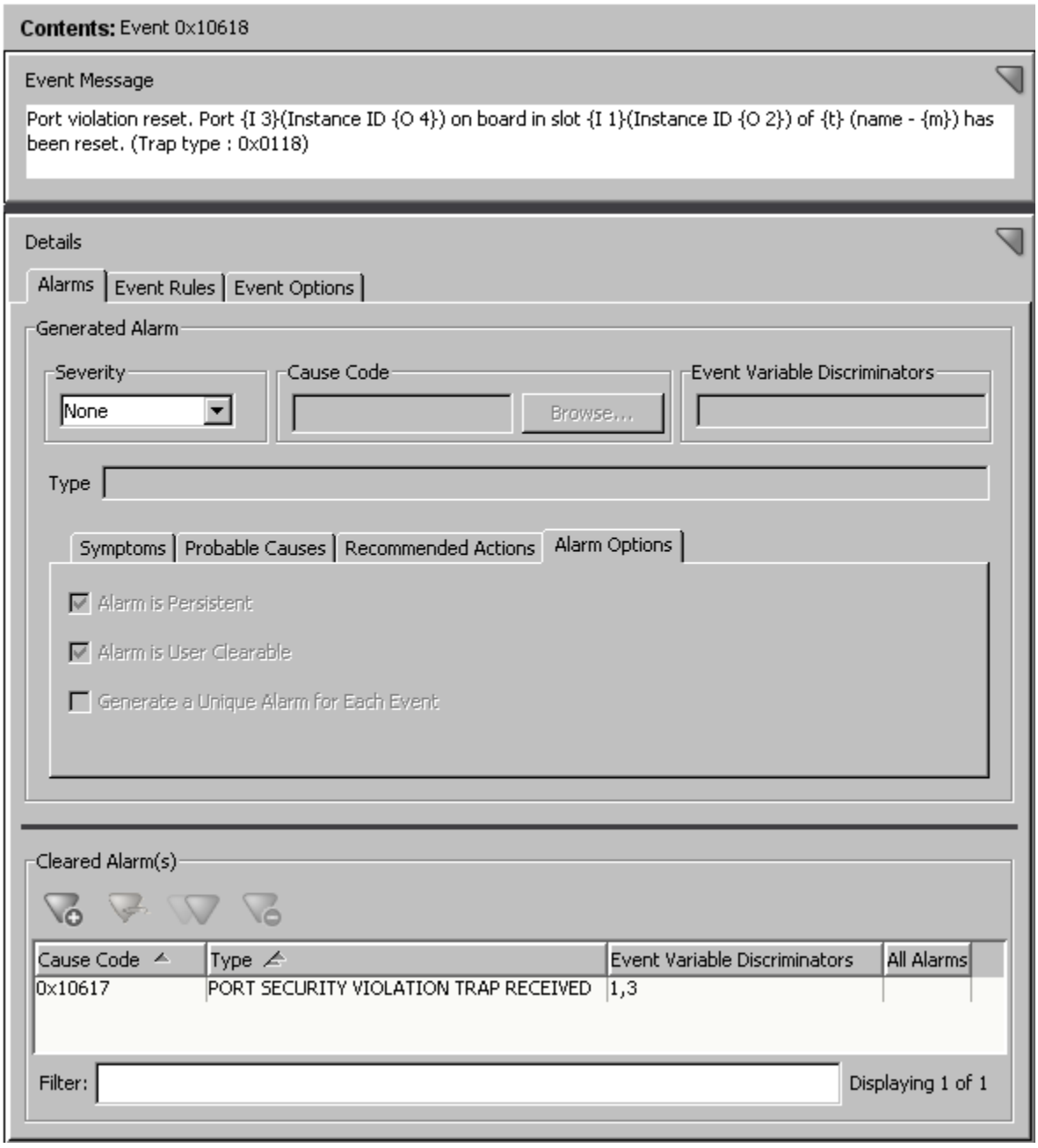
     NOTE
     Select Clear to clear the alarm -- instead of generating it -- if the corresponding variable binding value is sent.
   - To add a mapping, enter a unique string value for String, select a severity for Severity, and click Add.
   - To remove a mapping, select the mapping, and click Remove.

     NOTE

     DX NetOps Spectrum treats the entries in SeverityMap file as regular expressions. To match the exact variable binding value to a DX NetOps Spectrum alarm severity level, we recommend that you use word boundary characters ('\b') before and after the varbind value string in SeverityMap file.
6. Double-click OK.

Configure Events to Clear Alarms

You can specify that the currently selected event clears one or more alarms using the Alarms tab in the Details panel, which is shown in the following image.



When you configure an event to clear alarms and then save that change to a landscape, you create a mapping between the event and the alarms to be cleared in a configuration file referred to as an event disposition file.

Follow these steps:

1. Select the event in the Navigation panel, and then click the Alarms tab in the Details panel.
2. Under Cleared Alarm(s), click the Adds an alarm to the list icon

   The Add Cleared Alarm dialog displays the alarm cause codes for all alarms that are loaded into Event Configuration.
3. Click Browse, and in the Select Alarm Cause Code dialog, select the alarm cause code of the alarm to clear.

To help you identify the desired alarm, enter a text string in the Filter text box. The list is filtered to include only the alarms with displayed properties that contain the text string.

4. Click OK.
5. If the alarm that you want to clear was generated based on event variable discriminators, select Clear Options, and then specify how the alarm is cleared. Select one of the following options:
   - **All Alarms**
     The event clears all existing instances of the alarm, regardless of whether the values in the alarm-clearing event match the values that are stored in the alarm instances.
   - **Event Variable Discriminators**
     The event clears existing instances of the alarm that is based on the values of the alarm variables (which are copied from the alarm-generating event to the alarm when the alarm is generated). Then enter a comma-separated list of event variable IDs (for example: 1,3,5). You must enter each ID separately; ranges of IDs are not supported. If the values in event match the values stored in the alarm, the alarm-clearing event clears the alarm.

     NOTE
     For examples of using event discriminators to clear alarms, see Clearing Alarms.
6. Click OK.
7. (Optional) Repeat the preceding steps to add additional alarms to be cleared.

Modify Events

You cannot delete the events that CA authors and provides with DX NetOps Spectrum. However, you can customize them to meet your requirements simply as you would any other events.

In addition, you can undo any customizations that you make to a CA-authored event by deleting the event. *For CA-authored events only*, this action does not delete the event but instead reverts it to its default configuration.

When you delete (revert) a CA-authored event, the author of the event changes from “Custom” back to “CA.” To identify the author of an event, add the Author column to the table of events in the Navigation panel, as described in Adding and Removing Columns from the Events Table.

To modify events.

1. Select the event in the Navigation panel.
   The event details are displayed in Contents and Details panels.
2. Modify the event details.
   The event is displayed in bold. The Modification column displays *Edited.*

   NOTE
   The event is marked *Edited.* However, the event is not saved. You can make more updates such as create more events, modify existing events, or delete events, and save all the updates at one time.
3. Do *one* of the following:
   a. Go to File, Save All.
      All marked events in the Navigation panel are saved.
   b. Select the desired events in the Navigation panel, and go to File, Save Selected.

      NOTE
      You can select multiple events using the Shift key.

      The selected events are saved.

   The events are modified. The events appear in the normal font and are not marked in the Modification column.

Delete Custom Events

You can delete any event that CA did not author. To identify the author of an event, add the Author column to the table of events in the Navigation panel.

WARNING
Do not delete an event until you are certain it is no longer required. If you delete an event that generates a needed alarm, DX NetOps Spectrum is unable to inform you about a problem in the network infrastructure.

To delete events.

1. To delete events, select the event in the Navigation Panel, and click the delete icon

   The event is displayed in italics. The Modification column displays *Deleted.*

   NOTE
   The event is marked *Deleted.* However, the event is not deleted until you save the changes. You can make more updates such as create more events, modify existing events, or delete events, and save all the updates at one time.
2. Do one of the following:
   a. Go to File, Save All.
      All marked events in the Navigation panel are saved.
   b. Select the desired events in the Navigation panel, and go to File, Save Selected.

      NOTE
      You can select multiple events using the Shift key.

      The selected events are saved.

   The events are deleted. The events appear in the normal font and are not marked in the Modification column.

Event and Alarm Customization

DX NetOps Spectrum events and alarms support extensive customization. The files that support the DX NetOps Spectrum predefined events and alarms are installed in subfolders of the following directories:

```
$SPECROOT/SS/CsVendor
$SPECROOT/SG-Support
```

If you customize predefined events and alarms or create your own, and you then save the customizations to one or more landscapes, the event and alarm configuration files that define your customizations are installed in the following folder or in one of its subfolders:

```
$SPECROOT/custom/Events
```

As a result, the support files for your custom events and alarms are not overwritten or otherwise affected when you upgrade to a new version of DX NetOps Spectrum.

WARNING

Do not customize the following list of event codes as this will adversely affect Spectrum's fault isolation algorithms which leads to undesirable results.

LINK_GOOD = 0x00010d10,
LINK_BAD = 0x00010d11,
LINK_DISABLED = 0x00010d12,
LINK_UNKNOWN = 0x00010d13,
LINK_UNREACHABLE = 0x00010d14,
LINK_INITIAL = 0x00010d15,
LINK_BAD_SUPPRESSED = 0x00010d16,
LINK_LL_IN_MAINTENANCE = 0x00010d2e,
LINKED_PORT_BAD = 0x00010d17,

LINKED_DEVICE_BAD = 0x00010d18,
LINK_STATUS_CLEAR = 0x00010d2a,
LINK_DISABLED_AND_BAD = 0x0001040d,
LINK_ALWAYS_DOWN = 0x00010d2f,
LINK_BAD_WA_LINK_BAD = 0x00010d2d
CS_EVENT_ALARM_SET = 0x00010701
CS_EVENT_ALARM_UPDATED = 0x00010703
CS_EVENT_DEVICE_LOST = 0x00010302
DEVICE_CONTACT_STATUS_LOST = 0x00010d35
DEVICE_GATEWAY_UNREACHABLE = 0x00010d36
MGMT_LOST = 0x00010d00
UNRESOLVED_FAULT = 0x00010d05
APPLICATION_LOST = 0x00010d09
APPLICATION_REACTIVATED = 0x00010d0b
MGMT_REACTIVATED = 0x00010d0c
DEV_MNGMNT_NBORS_DOWN = 0x00010d19
LANDSCAPE_CONTACT_STATUS_CLEAR = 0x00010d20
PORT_LOST = 0x00010d66
PORT_REACTIVATED = 0x00010d67
ALARMING_MGMT_LOST_EVENT = 0x10daa

Customizing the above events would have direct impact on Fault Isolation (FI)/ Internal Port Link Status (IPLS).

Working with Event Rules

Event Rules

Event Rules let you specify the evaluation logic determining how DX NetOps Spectrum processes those events to which you are applying the rules. You can also specify what actions to perform in response to the events.

An event rule stipulates the conditions under which an event condition, a pattern of events, a combination of events in a particular order, or over a specific period, sets the generation of another event.

You can create multiple rules for a single event.

> **NOTE**
> The Event rules are stored in event configuration files referred to as event disposition files.

Event Condition Rules

An *event condition rule* creates an event when specific conditions are satisfied. The input to the rule is a list of conditions and associated events. Each condition is evaluated until one condition evaluates to TRUE, and then the corresponding event is created.

The following parameters are applicable:

- The event that starts the evaluation of the rule
- A conditional expression to evaluate
- If the conditional expression is true, the event to generate

An event condition rule can include multiple conditional expressions and corresponding output events.

Event Pair Rules

In some cases, you expect events to happen in pairs, and if the second event does not occur, it could indicate a problem in the computing infrastructure. Based on such a scenario the *event pair rule* creates an event. If one of two expected events are generated but the second event does not follow the first, a new event is generated in response. You can specify the amount of wait time that can elapse before the new event is generated.

NOTE
Other unrelated events can be generated between the first and the second event. They do not affect the execution of this rule.

Event Rate Rules

Some types of events do not indicate a problem unless the frequency at which they are generated reaches a specific threshold within a specific amount of time. An *event rate rule* creates an event based on this scenario. When a number of events of the same type that is, with the same event code are created within a given time period, a new event is created in response.

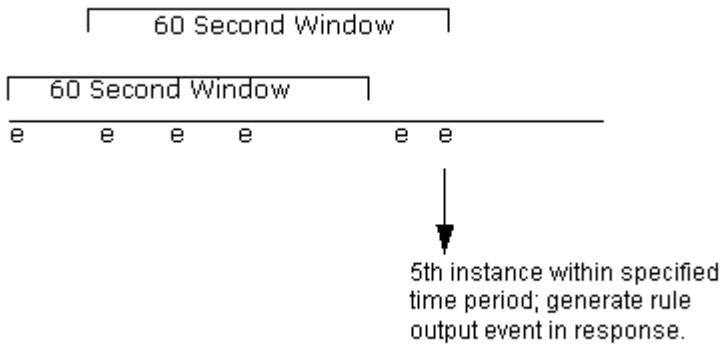
NOTE
Other, unrelated events do not affect rule execution.

Event rate rules never terminate. Once the conditions of the rule are met and a new event is created in response, the rule remains active. However, no additional event is created as long as the frequency at which the evaluated events remains at or above the specified rate in the rule. If the frequency drops below the specified rate, and then subsequently exceeds that rate again, a new event is generated.

An event rate rule can use either of the following methods to define the window of time in which the events must occur:

Sliding Window: With this type of time window, if the specified number of events (or more) ever occurs in any window of the specified time period, the output event is created in response. This type of time window is best suited for accurately detecting a short burst of events.

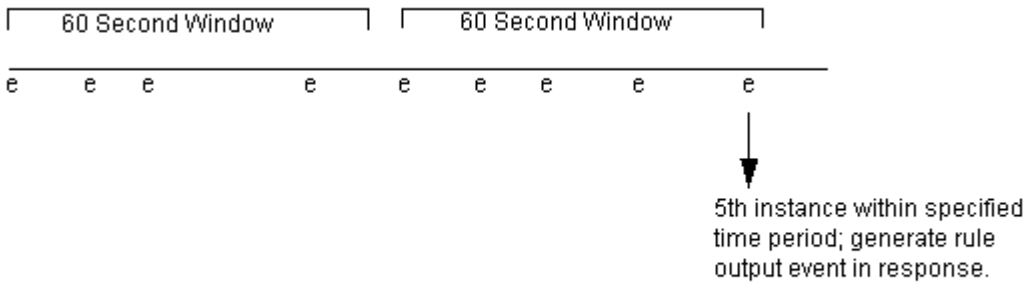
For example, the following illustration shows the sliding time windows that are active for a rule that watches for five instances of a given event (e) within a specified time period.



When a sliding time window is used for a rule, if the rule generates a rule output event, all active time windows are terminated, and a new time window automatically begins.

Sequential Window: When the rule uses this type of time window, non-overlapping time windows are examined, one after another, to determine if the requisite number of events has occurred within the time window. This type of time window is best suited for detecting a long, sustained train of events.

For example, the following illustration shows the sequential time windows that are opened and closed for a rule that watches for five instances of a given event (e) within a specified time period.



If the current time window closes due to time period expiration, or if the rule creates an output event in response, the window is not opened next time until a new event occurrence is detected.

Event Series Rules

An *event series rule* creates an event when a specific event. triggers one or more other events in an ordered or unordered sequence. The combination of events which must occur include any number and type of event. You can specify also the amount of wait time to elapse during which the sequence of events must occur.

Note: Other, unrelated events can be generated during the wait time. They do not affect the execution of a rule.

Event Counter Rules

An *Event Counter rule* counts events. The rule watches for two events, one increasing the count, the other one decreasing it again. An event is generated whenever the count is higher than a threshold, and also once it falls below the set threshold. The Event Counter rule remains instantiated and counts events and does not terminate.

Heartbeat Rules

The *Heartbeat rule* watches for a periodic heartbeat event, and generates a new event when the heartbeat event is missing. You can stop the rule instance using a separate event.

Single Event Rules

The *Single Event rule* watches for a single occurrence of a target event. The Single Event rule reduces an event stream where one event ('up' event) occurs multiple times, before a reset ('down') event is observed. Instead of the multiple 'up' events, a single event which can be reused in other rules, denoting the condition ('up' or 'down')is set. An event is generated the first time the target event is seen. The rule triggers only when the clearing event is seen again. The rule then resets the behavior to the initial state. If needed, a separate event can be generated when the clearing event is seen.

Solo Event Rules

The *Solo Event rule* finds an instance of target event which does not follow or precede any other user-defined event in a defined period.

The time periods are configurable, and there is a separate event to stop the rule.

User-Defined Event Rules

Most of the CA-shipped event rules are supported on the Event Configuration User Interface. However, in some cases, CA can create a special rule or can customize a rule for an individual customer. A customer can also create a rule.

The Event Configuration Editor does not support Event disposition entries using such rules. The displayed event rule is read-only. The rule entry must be edited in a text editor and reloaded in the event disposition files. Such custom event rule entries must have the correct vendor code and rule name, depending on which vendor supplied the rule. For example, a custom rule entry for a rule that is named MyOwnRule' from vendor 'MyOwnVendor' has the following syntax:

0xfff00002 E50 R MyOwnVendor.MyOwnRule, <rule parameters>

NOTE
The xml rule definition file is at $SPECROOT/SS/CsVendor/<Vendor Name>/EventRules/<RuleName>.xml.

Create Event Rules

We recommend that you start the application and examine the rules that CA provides for various events, before creating an event rule.

You can create an event rule in two ways:

- From scratch.
- By copying an existing event rule and modifying the copy. This method is only useful in case the same event triggers the new rule that you want to copy.

NOTE
Event rule definitions are stored in event configuration files referred to as event disposition files.

To create an event rule:

1. In the Navigation panel, select the event that activates (triggers) the event rule.
2. In the Details panel, click the Event Rules tab.
   The list of event rules for the selected event appears.
3. Click the list
   icon  .
4. Select the type of event to create from the drop-down list.

   The configuration dialog for the selected type of event rule opens.
5. Configure the event rule as required.
6. Click OK to save the event rule to the event.

To create an event rule from a copy:

1. In the Navigation panel, select the event that has the event rule that you want to copy.
2. In the Details panel, click the Event Rules tab.
   The list of event rules for the selected event appears.
3. To copy an event rule, select the desired rule and click the copy
   icon  .
   The event rule configuration dialog opens.
4. Modify the configuration of the event rule as required.
5. Click OK to save the event rule to the event.

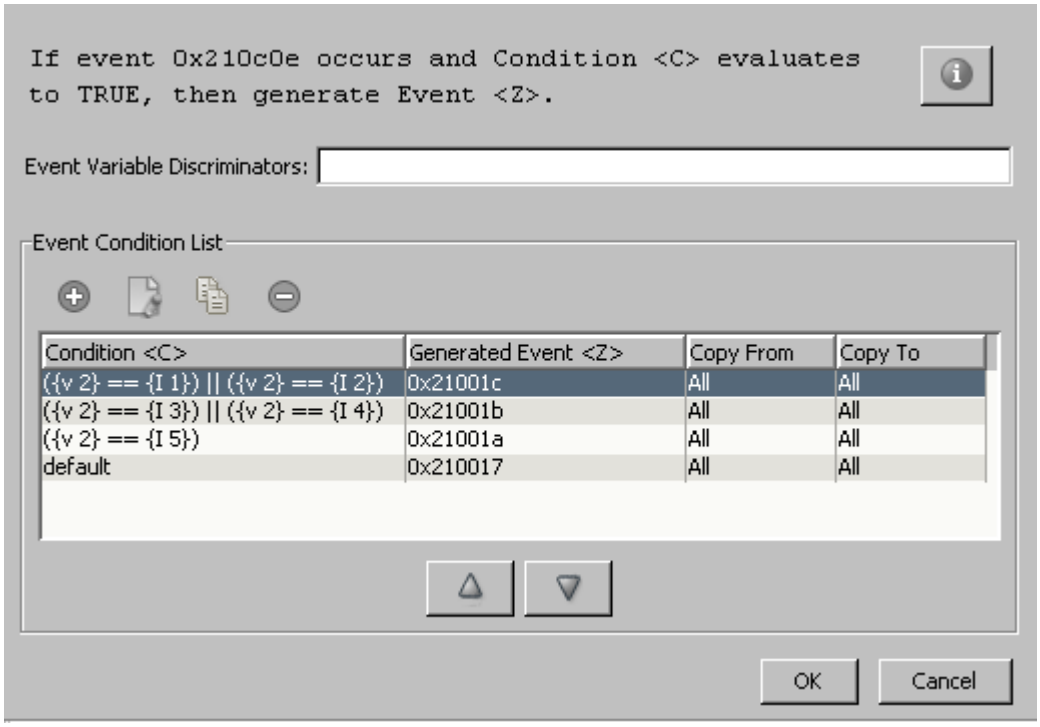
Configuring Event Condition Rule Settings

To create an event condition rule, create one or more conditions to evaluate the event for which you are creating the rule. For each condition, define an event that DX NetOps Spectrum generates if the given condition is met. Optionally, you can also specify that one or more values of the event that triggered the rule is copied to the rule output event.

As an example, assume you want to create the following event condition rule (expressed in pseudo code) for event 0x210c0e:

```
If event 0x210c0e occurs, evaluate the following:
if event variable 2 = 2 or event variable 2 = 2, then generate 0x0021001c,
else if event variable 2 = 3 or event variable 2 = 4, then generate 0x0021001b,
else if event variable 2 = 5, then generate 0x0021001a,
else, generate 0x00210017.
```

This rule requires the four conditional expressions that are shown in the following rule configuration dialog.

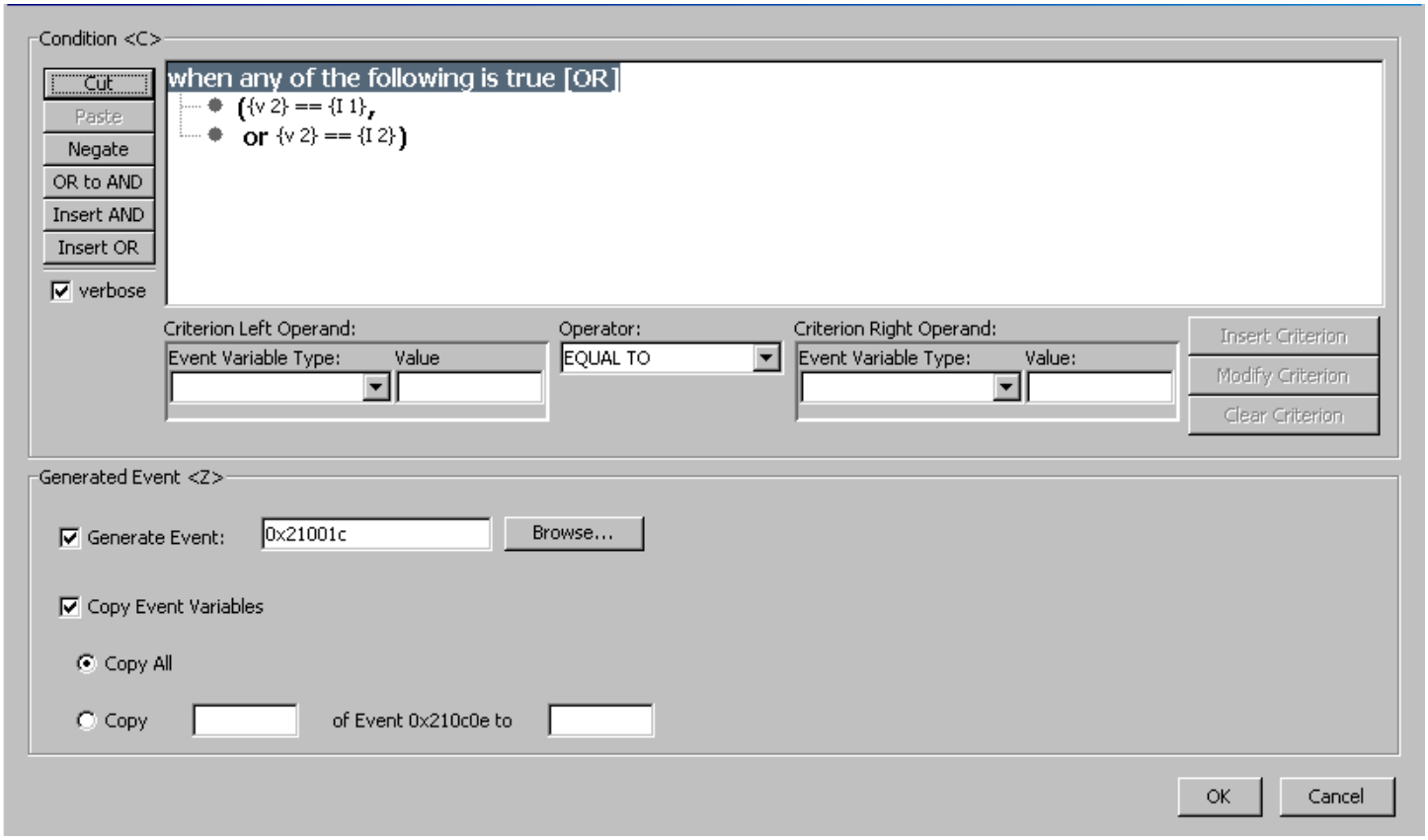


In the dialog, use the buttons above the list of conditions to add new conditions and to modify, copy, and delete a selected condition. To change the sequence in which the conditions are evaluated, use the up and down arrows below the list of conditions.

The rule output event (generated event <Z>) that corresponds to the first condition that evaluates to TRUE is generated.

> **NOTE**
> Event variable discriminators are a general feature available for all types of rules. However, while you can specify discriminators for an event condition rule, a rule of this type currently does not use them during rule processing.

When you add or modify a condition, you define the conditional expression using the following dialog.



The settings in the dialog include the following options:

- **Condition ©**
  The condition to evaluate. To create the expression, use the controls in the top section of the dialog.
  For information about conditional expression syntax, see Event Condition Rule.
- **Generated Event <Z>- Generate Event**
  If the associated condition evaluates to TRUE, the event code of the event to create in response.
  To specify the event code, click Browse, select the event in the Select Event dialog, and then click OK.
- **Generated Event <Z> - Copy Event Variables**
  To copy the values of any event variables in that event which activated the rule to the generated event <Z>, select Copy Event Variables and then do any of the following steps:
  – Select Copy All to copy the values of all of the event variables into the generated event <Z>.
  – Select Copy, and specify a comma-separated list of specific variable IDs whose values must be copied. Ranges of IDs are also supported, for example, “1-4”.

NOTE
For information about using the proper syntax when specifying variable IDs or ranges of IDs using Event Configuration, see Copying Variable Values from Contributing Events to the Rule Output Event.

NOTE
To create a default condition in the condition configuration dialog evaluating to TRUE; select DEFAULT for Operator and click Insert Criterion.

Configuring Event Pair Rule Settings

The Event Pair dialog provides the following settings:

The settings in the dialog include the following:

- **Event Variable Discriminators**
  A comma-separated list of the IDs of the event variables in the event to evaluate as a part of the rule.

  NOTE
  Enter each ID separately; ranges of IDs are not supported.
- **Event <Y> - Event Code**
  The event code of the event that, within the specified time period, follows the event that activates (triggers) the rule. Event <Y> is the second event in the pair of events.
  To specify the event code, click Browse, select the event in the Select Event dialog (which displays all of the events that are loaded into Event Configuration), and click OK.
- **Time <T> - Time Interval**
  The period that begins when the first event is created and during which the second event (event <Y>) occurs.
- **Get Time From Attribute**
  Select this option to use the value of any time attribute whose value is a quantity of time, such as like the polling _interval attribute (0x1134e). The value of the selected time attribute becomes the value of Time <T>. The time unit is seconds.
  The format would resemble the following:
  if Event X occurs and is not followed by Event Y within the time specified in the time- specific Attribute Y, then generate Event Z.
- **Generated Event <Z> - Generate Event**
  The event code of the event to create in response if the second event (event <Y>) does not follow the first event within the specified time interval.
  To specify the event code, click Browse, select the event in the Select Event dialog (which displays all of the events that are loaded into Event Configuration), and click OK.
- **Generated Event <Z> - Copy Event Variables**
  If you want to copy the values of any event variables in the first event that activated the rule to generated event <Z>, select Copy Event Variables, and do one of the following:
  – Select Copy All to copy the values of all of the variables into generated event <Z>.
  – Select Copy, and specify a comma-separated list of specific variable IDs whose values should be copied. Ranges of IDs are also supported, for example, “1-4”.

  NOTE
  For information about the proper syntax to use when specifying variable IDs or ranges of IDs using Event Configuration, see Copying Variable Values from Contributing Events to the Rule Output Event.

Event Variable Discriminators in Event Pair Rules

The use of discriminators not only lets multiple instances of the rule to be generated due to the same event, but also affects rule processing and termination. More specifically, if discriminators are used, the following occurs:

When the event is generated, if there are active instances of the rule, another instance of the rule is created only if the values of the specified variables are different in the new event when compared to the events that activated the existing rule instances. (If you do not specify any discriminators for a rule and if a single instance of the rule is active, then the subsequent instances of the rule are not generated when subsequent instances of the event occur.)

To terminate the rule without generating the rule output event, the values of all of the specified event variables must be the same in both the first event (that triggered the rule) and the second event (for which the rule is watching) in the pair.

As an example, assume you create an event pair rule that watches for a port down event followed by a port up event, and you specify the event variable that stores the interface ID of the port as an event variable discriminator. As a result, if device A has ports 1 and 2, and port 1 goes down, a rule instance for the event is generated. If port 2 subsequently goes down, another rule instance is also generated due to the same event because the interface ID in the second instance

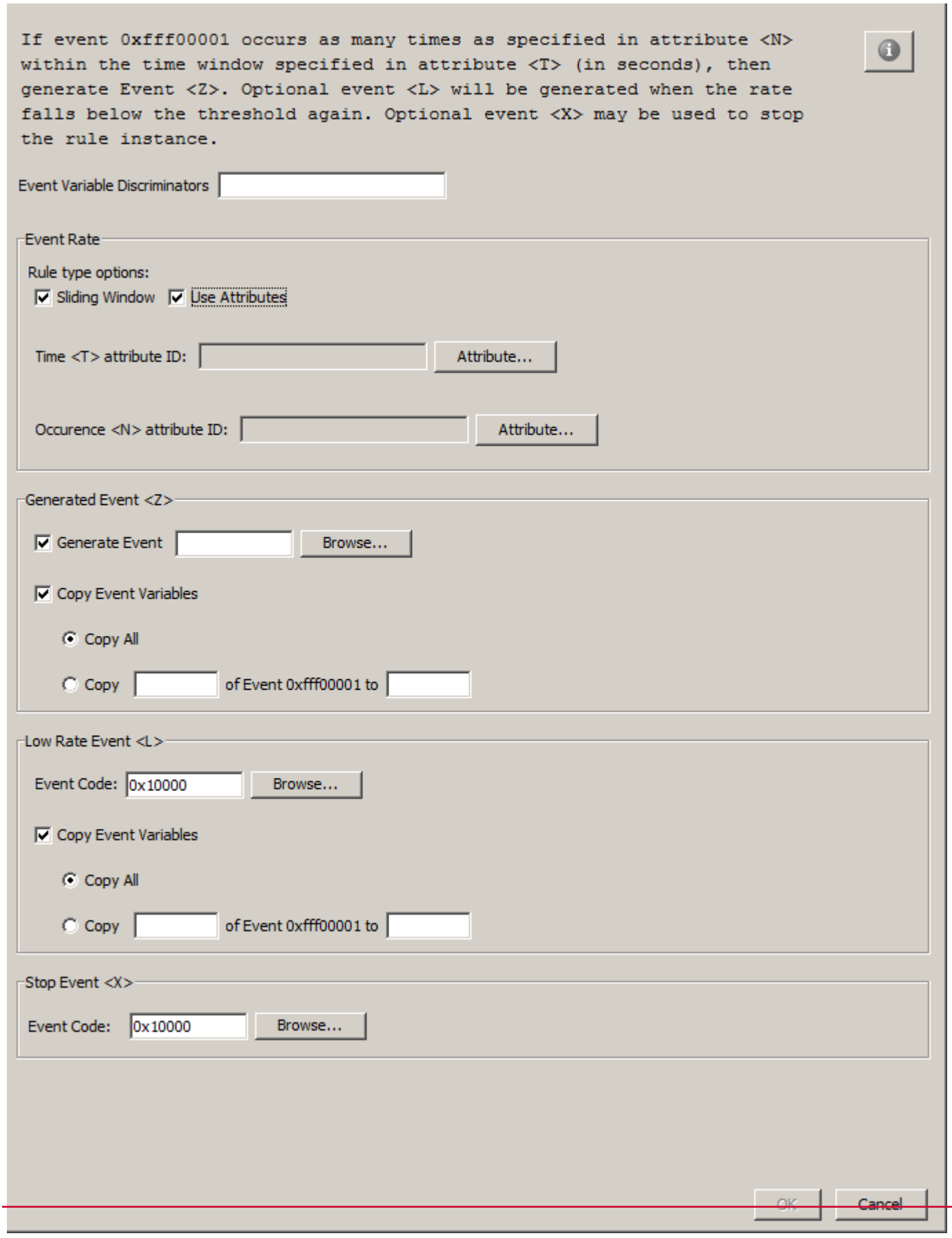
of the event is different. There are now two active instances of the rule due to port down events that are related to two different ports.

To continue the example, if port 2 goes up within the specified time period, the associated rule will immediately terminate without generating the rule output event if the port up event that is generated stores the interface ID of port 2 in the same event variable (that is, the IDs of the event variables in the first event and second event in the pair that store the interface ID are the same). If a similar event is not generated for port 1 within the specified time period, the associated rule terminates by generating the rule output event.

> **NOTE**
> To use event variable discriminators effectively in an event pair rule, the IDs of the event variables in the contributing events must match. That is, the same event variables must store the same variable binding data. If not, you can create an event condition rule that is activated by the first event and copies the data to event variables having the appropriate IDs in the rule output event; you can then use the rule output event as the second event in the pair.

Configuring Event Rate Rule Settings

To configure an event rate rule use the following dialog:


If event 0xfff00001 occurs as many times as specified in attribute <N> within the time window specified in attribute <T> (in seconds), then generate Event <Z>. Optional event <L> will be generated when the rate falls below the threshold again. Optional event <X> may be used to stop the rule instance.
Event Variable Discriminators
Event Rate
Rule type options:
Sliding Window
Use Attributes
Time <T> attribute ID:
Attribute...
Occurence <N> attribute ID:
Attribute...
Generated Event <Z>
Generate Event
Browse...
Copy Event Variables
Copy All
Copy
of Event 0xfff00001 to
Low Rate Event <L>
Event Code:
0x10000
Browse...
Copy Event Variables
Copy All
Copy
of Event 0xfff00001 to
Stop Event <X>
Event Code:
0x10000
Browse...
OK
Cancel

The dialog offers the following settings:

- **Event Variable Discriminators**
  A comma-separated list of the IDs of the event variables that are included in the event. Events in the list are evaluated as a part of the rule.

  NOTE
  Enter individual IDs; ranges of IDs are not supported.

  The use of discriminators not only lets multiple instances of the rule to be generated due to the same event, but also affects rule processing. More specifically, if discriminators are used, the following occurs:
  – When the event is generated, if there are active instances of the rule, another instance of the rule is created only if the values of the specified variables are different in the new event when compared to the events that activated the existing rule instances. (If you do not specify any discriminators for a rule and if a single instance of the rule is active, subsequent instances of the rule are not generated when subsequent instances of the event occur.)
  – To generate the rule output event, the values of the variables must be the same in all instances of the same event (and all other rule conditions must be met).
- **Event Rate - Occurrences <N>**
  The number of instances of the same event that must be created within the specified time period for the rule to generate the output event.
- **Event Rate - Time <T>**
  The period during which <N> occurrences of the same event are required for the rule to generate the output event.
- **Event Rate - Sliding Window**
  The type of time window to use. Select a Sliding Window, or clear Sliding Window to use a *sequential window* of time. For a description of both, see Event Rate Rules.
- **Generated Event <Z> - Generate Event**
  The event code of the event to create if <N> occurrences of the same event occur within the specified period.
  To specify the event code, click Browse, select the event in the Select Event dialog, and click OK.
- **Generated Event - Copy Event Variables**
  To copy the values of any event variables of the event that activated the rule into the generated event <Z>, select Copy Event Variables and do one of the following:
  – Select Copy All to copy the values of all of the event variables into generated event <Z>.
  – Select Copy, and specify a comma-separated list of specific variable IDs whose values are being copied. Ranges of IDs are also supported, for example, “1-4”.

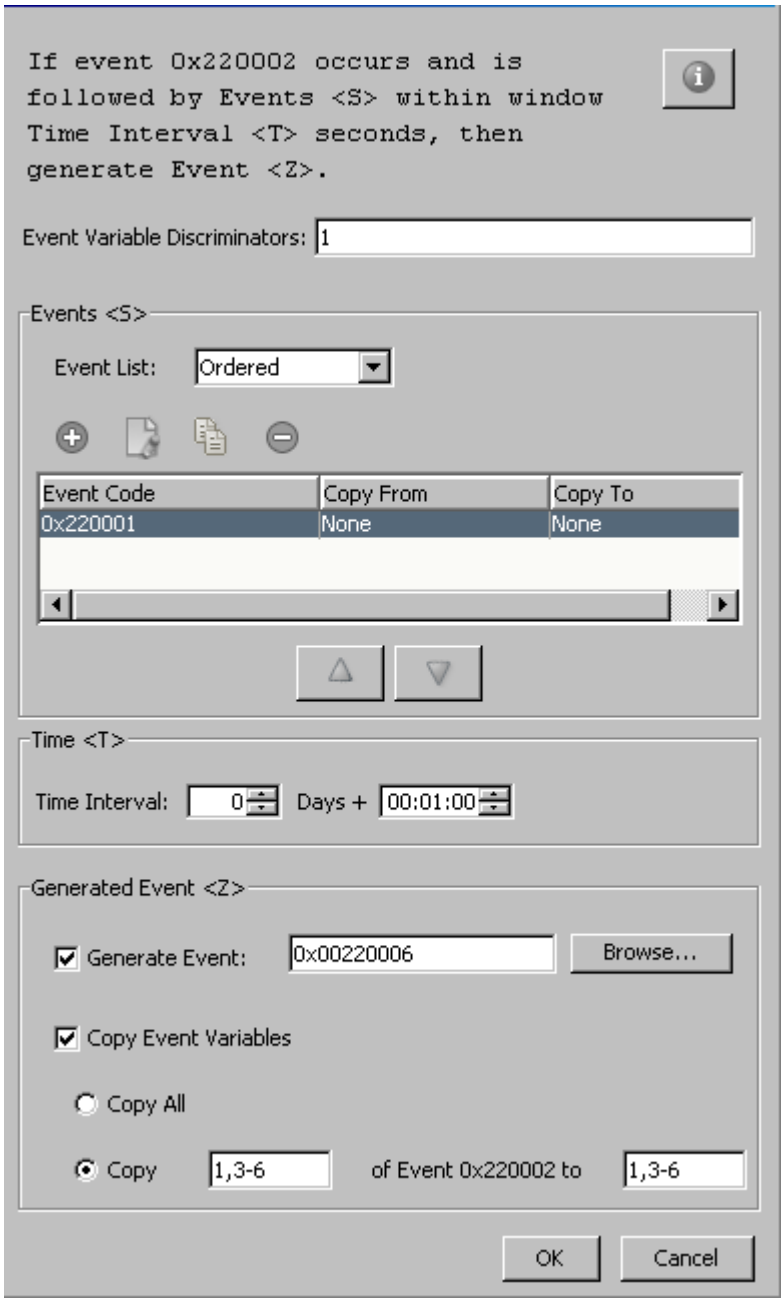
  NOTE
  For information about the proper syntax for specifying variable IDs or ranges of IDs using Event Configuration, see Copying Variable Values from Contributing Events to the Rule Output Event.
- **Low Rate Event**
  (Optional) Generates a low rate event <L> if the frequency drops below the threshold. The rule generates the high rate event again only when the frequency threshold is crossed again.
- **Stop Event**

  NOTE
  (Optional) Stops the rule with an event. The rule runs continuously unless you set the stop event. In many cases, letting it run continuously is the desired behavior.

NOTE
An event rate rule can also generate a different rule output event when the rate of the trigger event drops below the specified threshold. Manually modify the rule in the event disposition file and specify that rule output event. For more information, see EventRateWindow Rule.

Configuring Event Series Rule Settings

You can configure an event series rule using the following dialog.



The settings in the dialog include the following options:

- **Event Variable Discriminators**

  NOTE
  A comma-separated list of IDs of event variables in the event to evaluate as a part of the rule. Enter each ID manually, ranges of IDs are not supported.

  To allow multiple instances of the rule to be generated due to the same event, use discriminators. The use of discriminators also affects processing and termination of rules. The following conditions apply occur if discriminators are used:

  – In case, there are active instances of the rule during event generation, another instance of the rule is created only if the values of the specified variables are different in the new event as compared to the events activating the existing

rule instances. The subsequent instances of the rule are generated when a single instance of the rule is active, only when you specify any discriminators for a rule.

– To generate the rule output event, the values of the variables must be the same in all contributing events (that is, both, in the event that activated the rule, and in all events that are specified in the series).

NOTE
To use event variable discriminators effectively in an event series rule, the IDs of the event variables in the contributing events must match. That is, the same event variables must store the same variable binding data. Alternately, you can create an event condition rule that a given event triggers and copies the data to event variables having the appropriate IDs in the rule output event. You can then use the rule output event in the series.

- **Events <S> - Event List**
  Select Ordered if the list (series) of events must occur in a specific sequence to trigger creation of the response event. Select Not Ordered if the events in the series can occur in any order.
- **Events <S> - Event Code**
  The series of events that must follow the event that activated the rule for the rule to generate the output event. If the events in the series must occur in a specific order, list them in that order.
  To add, modify, copy, and remove events from the list use the buttons above the table.
  To move an event up or down in the list, select the event, and click the up or down arrow.
  You can add an event to the series, and can specify that one or more of the values in its event variables be copied to the rule output event (generated event <Z>). To select the event in the dialog to add to the series, do any of the following actions:
  – Select Copy All to copy the values of all of the event variables to the rule output event.
  – Select Copy, and specify a comma-separated list of specific variable IDs whose values you want to copy. Ranges of IDs are also supported, for example, “1-4”.
- **Time <T> - Time Interval**
  The period during which the series of events must occur for the rule to generate the output event.
- **Generated Event <Z> - Generate Event**
  The event code of the event to create, in response to a series of events occurring within the specified period and in the specified order.
  To specify the event code, click Browse, select the event in the Select Event dialog (displaying all events that are loaded into Event Configuration), and click OK.
- **Generated Event <Z> - Copy Event Variables**

To copy the values of any event variables in that event which first activated the rule to the generated event <Z>, select Copy Event Variables, and perform any of the following steps:

Select Copy All to copy the values of all of the event variables into the generated event <Z>.

- Select Copy, and specify a comma-separated list of specific variable IDs whose values are copied. Ranges of IDs are also supported, for example, “1-4”.

NOTE
For information about the proper syntax to use when specifying variable IDs or ranges of IDs using Event Configuration, see Copying Variable Values from Contributing Events to the Rule Output Event.

To generate an event in response to a specific series of events, where one of several events triggers that same series first. To satisfy this scenario, create the same event series rule for all events that triggers the series.

Alternatively, you want to watch for any combination of a series of events and in any order, inclusive of the event that triggers the series. To satisfy this scenario, we recommend that you create the following rules:

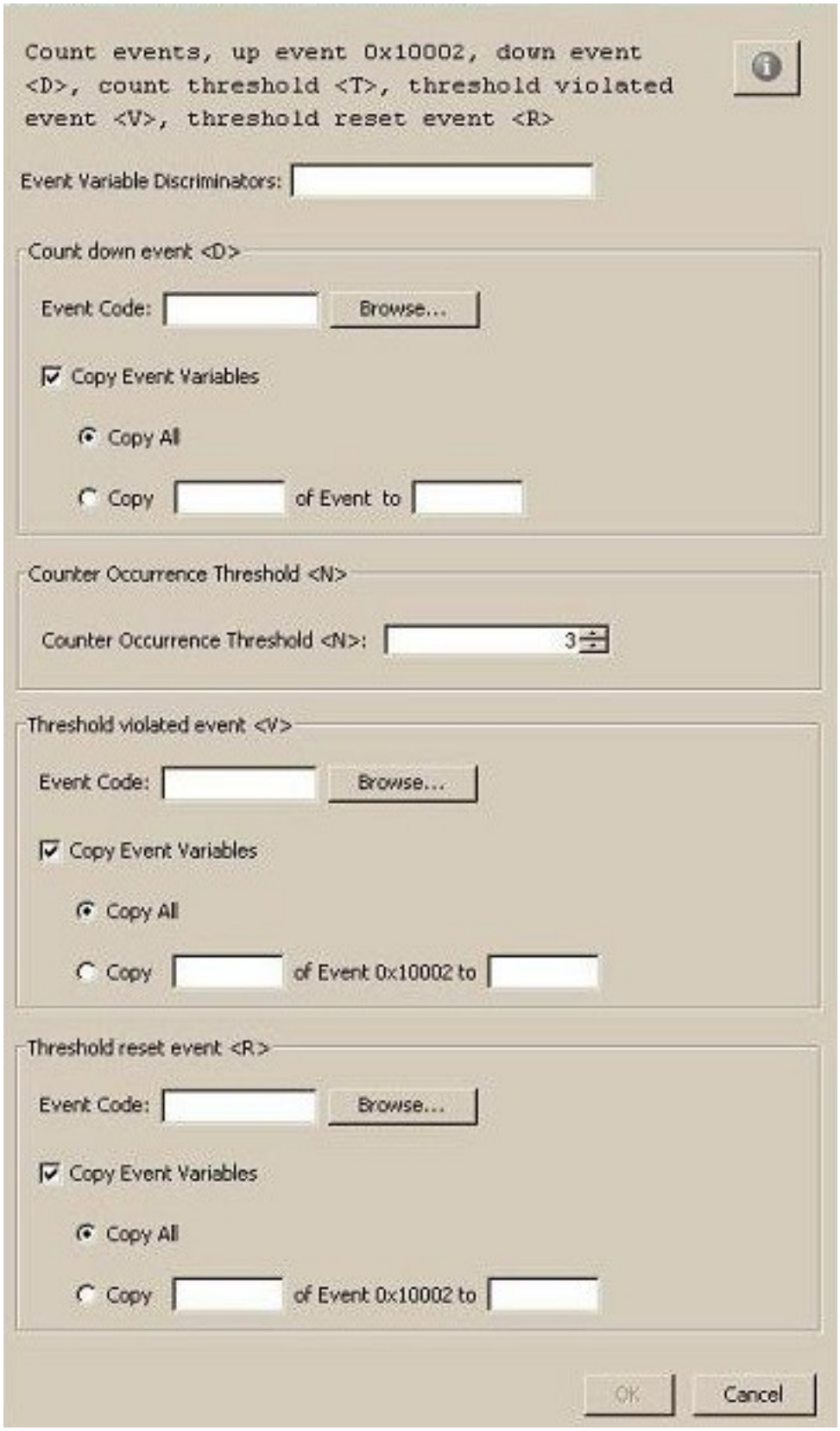
Event that triggers the rule (event to which rule is applied)	Events in the series (not ordered)	Rule output event (generated event <Z>)
0x0001002a	0x0001002b, 0x0001002c	0x0001002f

0x0001002b	0x0001002a, 0x0001002c	0x0001002f
0x0001002c	0x0001002a, 0x0001002b	0x0001002f

Using this set of rules, any combination of events 0x0001002a, 0x0001002b, and 0x0001002c occurring in any order generates event 0x0001002f.

Configuring Event Counter Rule Settings

You configure an event counter rule using the following dialog:


Count events, up event 0x10002, down event <D>, count threshold <T>, threshold violated event <V>, threshold reset event <R>
Event Variable Discriminators:
Count down event <D>
Event Code:
Browse...
Copy Event Variables
Copy All
Copy
of Event to
Counter Occurrence Threshold <N>
Counter Occurrence Threshold <N>:
3
Threshold violated event <V>
Event Code:
Browse...
Copy Event Variables
Copy All
Copy
of Event 0x10002 to
Threshold reset event <R>
Event Code:
Browse...
Copy Event Variables
Copy All
Copy
of Event 0x10002 to
OK
Cancel

The settings in the dialog include the following:

The event for which the rule is defined is the one that counts up the event. The first event initiates a new rule instance and counts (count is 1).

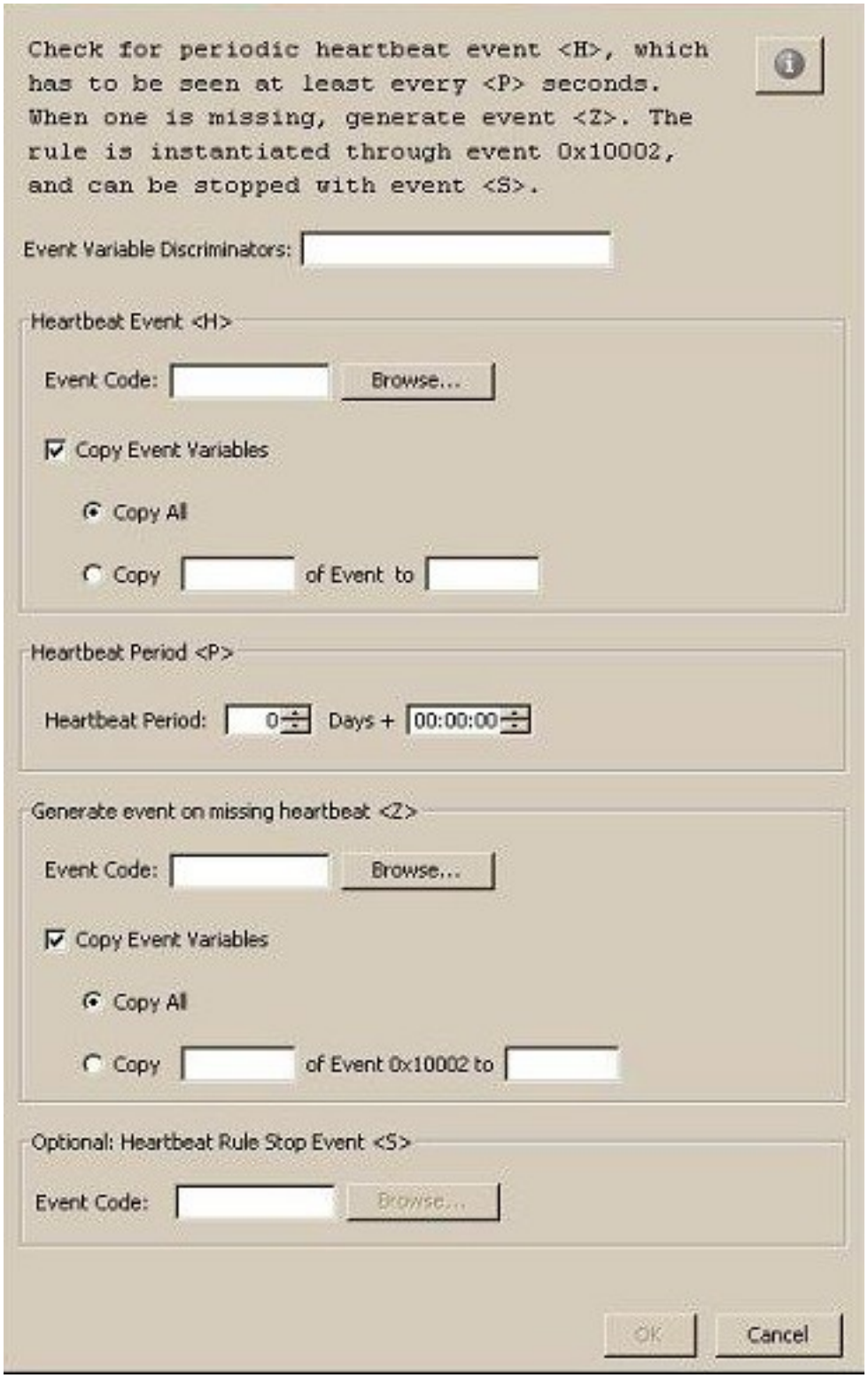
- **Event Variable Discriminators**
  A comma-separated list of the IDs of the event variables in the event to evaluate as a part of the rule.

  NOTE
  Enter each ID separately; ranges of IDs are not supported. Event variable discriminators are a general feature available for all types of rules.
- **Count down event <D> - Event Code**
  Sets the event code that counts down by one. To specify the event code, click Browse, select the event in the Select Event dialog which displays all of the events that are loaded into Event Configuration, and click OK.
- **Count down event <D> - Copy Event Variables**
  If you want to copy the values of any event variables in the first event that activated the rule to generated event <D>, select Copy Event Variables, and do one of the following:
  – Select Copy All to copy the values of all of the variables into Count down event <D>.
  – Select Copy, and specify a comma-separated list of specific variable IDs whose values should be copied. Ranges of IDs are also supported, for example, “1-4”.

  NOTE
  For information about the proper syntax to use when specifying variable IDs or ranges of IDs using Event Configuration, see Copying Variable Values from Contributing Events to the Rule Output Event.
- **Counter Event Threshold <N>**
  Sets a certain threshold that when reached, generates a new event. The rule also generates another event when the count is lower then the threshold again.
- **Threshold violated event <V> - Event Code**
  Is generated when the count threshold is reached. To specify the event code, click Browse, select the event in the Select Event dialog (which displays all of the events that are loaded into Event Configuration), and click OK.
- **Threshold violated event <V> - Copy Event Variables**
  If you want to copy the values of any event variables in the first event that activated the rule to generated event <V>, select Copy Event Variables, and do one of the following:
  – Select Copy All to copy the values of all of the variables into Threshold violated event <V>.
  – Select Copy, and specify a comma-separated list of specific variable IDs whose values you wish to copy. Ranges of IDs are also supported, for example, “1-4”.
- **Threshold reset event Reason: - Event Code**
  Generates when the count is below the threshold again. To specify the event code, click Browse, select the event in the Select Event dialog (which displays all of the events that are loaded into Event Configuration), and click OK.
- **Threshold reset event Reason: - Copy Event Variables**
  If you want to copy the values of any event variables in the first event that activated the rule to generated event Reason:, select Copy Event Variables, and do one of the following:
  – Select Copy All to copy the values of all of the variables into Threshold reset event Reason:.
  – Select Copy, and specify a comma-separated list of specific variable IDs whose values you wish to copy. Ranges of IDs are also supported, for example, “1-4”.

Configuring Heartbeat Rule Settings

You can configure heartbeat rule using the following dialog.



The settings in the dialog include the following:

- **Event Variable Discriminators**
  A comma-separated list of the IDs of the event variables in the event to evaluate as a part of the rule.

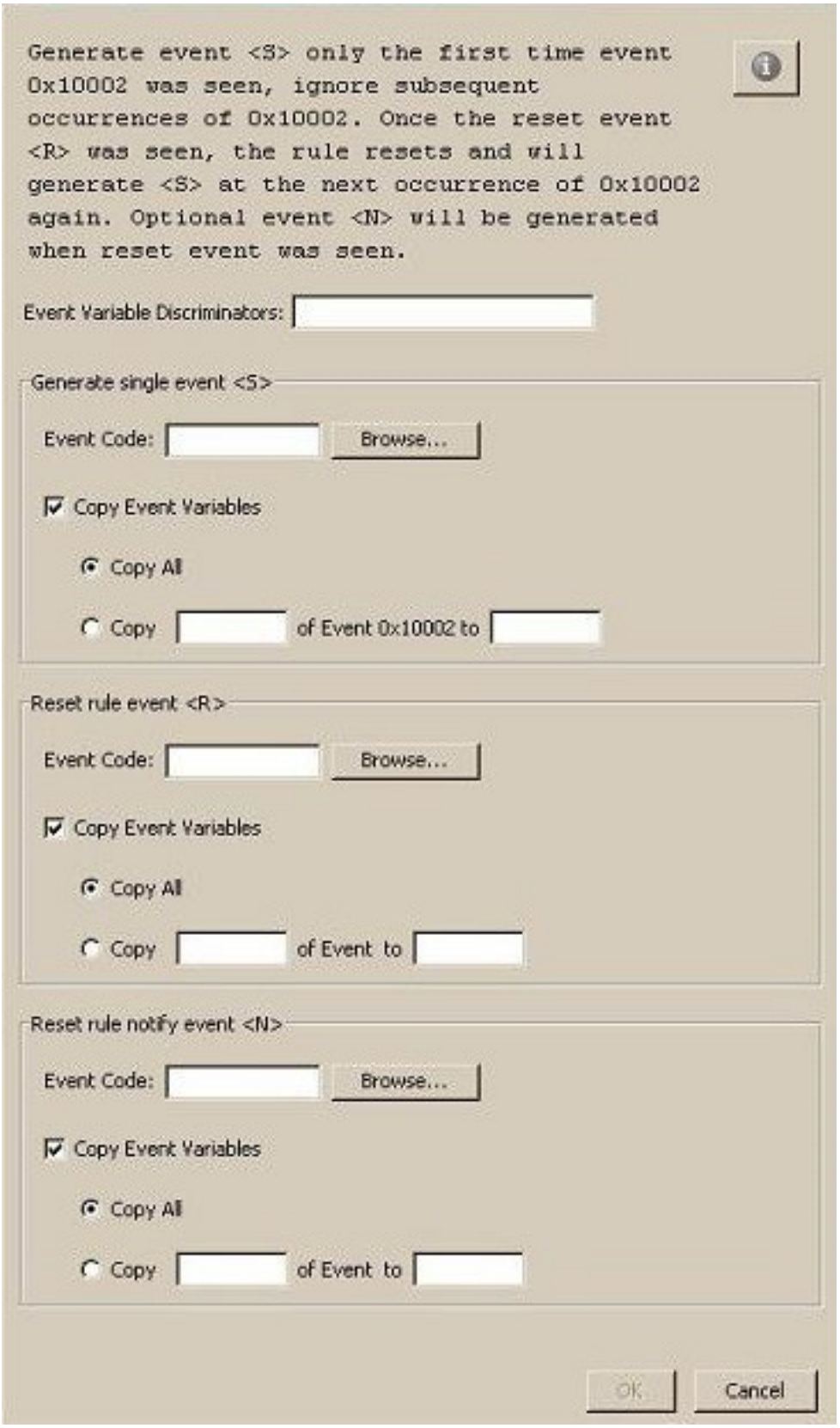
NOTE
You must enter each ID; ranges of IDs are not supported. Event variable discriminators are a general feature available for all types of rules.

- **Heartbeat Event <H> - Event Code**
  Sets the Heartbeat Event code. To specify the event code, click Browse, select the event in the Select Event dialog (which displays all of the events loaded into Event Configuration), and click OK.
- **Heartbeat Event <H> - Copy Event Variables**
  If you want to copy the values of any event variables in the first event that activated the rule to generated event <H>, select Copy Event Variables, and do one of the following:
  – Select Copy All to copy the values of all of the variables into Heartbeat Event <H>.
  – Select Copy, and specify a comma-separated list of specific variable IDs whose values should be copied. Ranges of IDs are also supported, for example, “1-4”.

  NOTE
  For information on the proper syntax to use when specifying variable IDs or ranges of IDs using Event Configuration, see Copying Variable Values from Contributing Events to the Rule Output Event.
- **Heartbeat Period <P>**
  Sets the time gap between individual heartbeats.
- **Generate event on missing heartbeat <Z> - Event Code**
  Generates an event when the heartbeat event is missed. To specify the event code, click Browse, select the event in the Select Event dialog (which displays all of the events loaded into Event Configuration), and click OK.
- **Generate event on missing heartbeat <Z> - Copy Event Variables**
  If you want to copy the values of any event variables in the first event that activated the rule to generated event <Z>, select Copy Event Variables, and do one of the following:
  – Select Copy All to copy the values of all of the variables to Generate event on missing heartbeat <Z>.
  – Select Copy, and specify a comma-separated list of specific variable IDs whose values should be copied. Ranges of IDs are also supported, for example, “1-4”.
- **(Optional) Heartbeat Rule Stop Event <S> - Event Code**
  The Heartbeat Rule Stop Event stops the Heartbeat event rule.To specify the event code, click Browse, select the event in the Select Event dialog (which displays all of the events loaded into Event Configuration), and click OK.

Configuring Single Event Rule Settings

You can configure a single event rule using the following dialog.


Generate event <S> only the first time event 0x10002 was seen, ignore subsequent occurrences of 0x10002. Once the reset event <R> was seen, the rule resets and will generate <S> at the next occurrence of 0x10002 again. Optional event <N> will be generated when reset event was seen.
Event Variable Discriminators:
Generate single event <S>
Event Code:
Browse...
Copy Event Variables
Copy All
Copy
of Event 0x10002 to
Reset rule event <R>
Event Code:
Browse...
Copy Event Variables
Copy All
Copy
of Event to
Reset rule notify event <N>
Event Code:
Browse...
Copy Event Variables
Copy All
Copy
of Event to
OK
Cancel

The settings in the dialog include the following:

- **Event Variable Discriminators**
  A comma-separated list of the IDs of the event variables in the event to evaluate as a part of the rule.

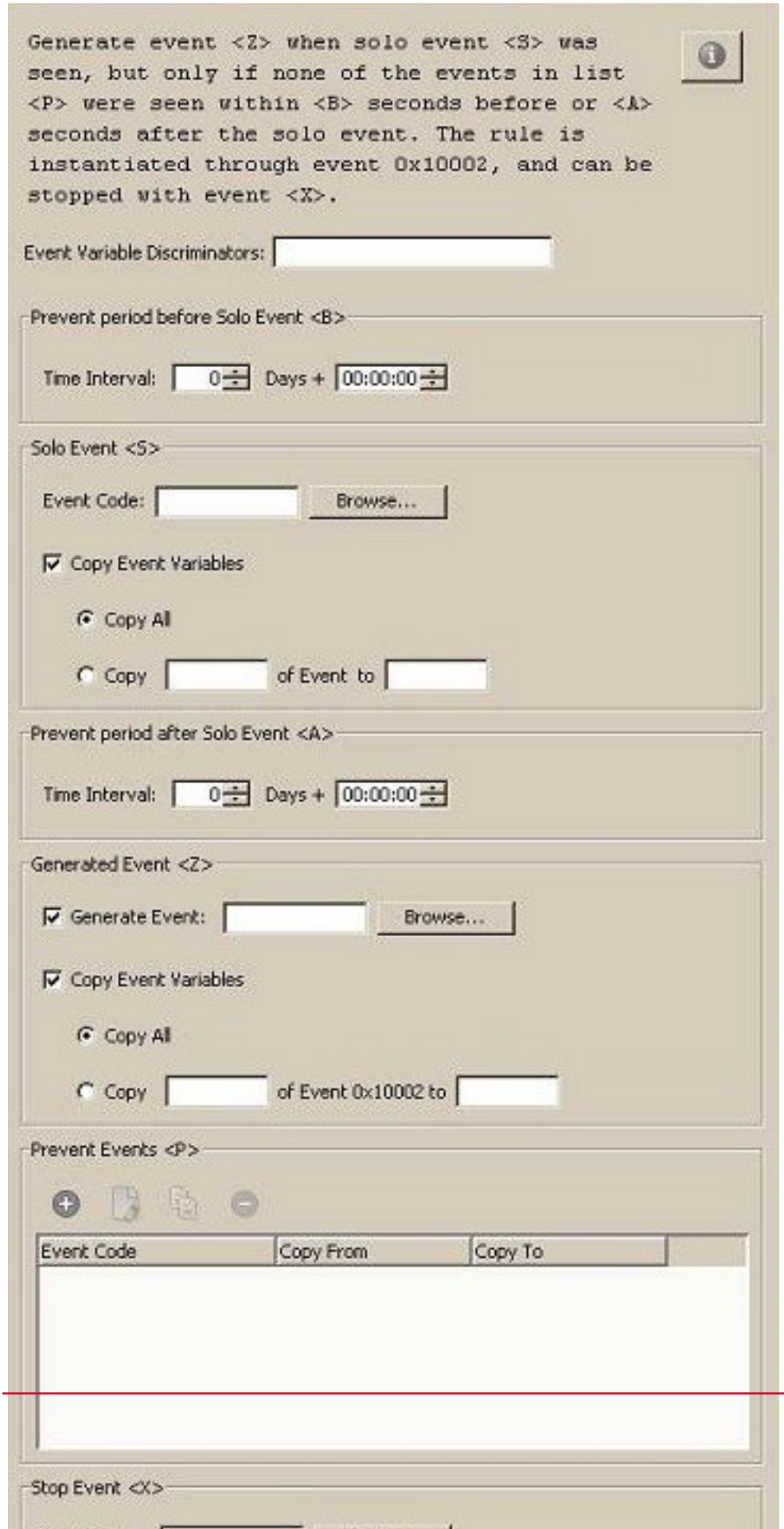
  NOTE
  You must enter each ID; ranges of IDs are not supported. Event variable discriminators are a general feature available for all types of rules.
- **Generate single event <S> - Event Code**
  It is generated the first time the trigger event is seen either when the rule is instantiated, or the first time the trigger event occurs after the reset event is seen. To specify the event code, click Browse, select the event in the Select Event dialog (which displays all of the events loaded into Event Configuration), and click OK.
- **Generate single event <S> - Copy Event Variables**
  If you want to copy the values of any event variables in the first event that activated the rule to generated event <D>, select Copy Event Variables, and do one of the following:
  – Select Copy All to copy the values of all of the variables into Count down event <D>.
  – Select Copy, and specify a comma-separated list of specific variable IDs whose values should be copied. Ranges of IDs are also supported, for example, “1-4”.

  NOTE
  For information on the proper syntax to use when specifying variable IDs or ranges of IDs using Event Configuration, see Copying Variable Values from Contributing Events to the Rule Output Event.
- **Reset rule event Reason: - Event Code**
  Sets the reset event. To specify the event code, click Browse, select the event in the Select Event dialog (which displays all of the events loaded into Event Configuration), and click OK.
- **Reset rule event Reason: - Copy Event Variables**
  If you want to copy the values of any event variables in the first event that activated the rule to generated event <D>, select Copy Event Variables, and do one of the following:
  – Select Copy All to copy the values of all of the variables into Count down event <D>.
  – Select Copy, and specify a comma-separated list of specific variable IDs whose values should be copied. Ranges of IDs are also supported, for example, “1-4”.
- **(Optional) Reset rule notify event <N> - Event Code**
  It is generated when the reset event is seen. To specify the event code, click Browse, select the event in the Select Event dialog (which displays all of the events loaded into Event Configuration), and click OK.
- **Reset rule notify event <N>- Copy Event Variables**
  If you want to copy the values of any event variables in the first event that activated the rule to generated event <D>, select Copy Event Variables, and do one of the following:
  – Select Copy All to copy the values of all of the variables into Count down event <D>.
  – Select Copy, and specify a comma-separated list of specific variable IDs whose values should be copied. Ranges of IDs are also supported, for example, “1-4”.

  WARNING

Configuring Solo Event Rule Settings

You can configure a solo event rule using the following dialog.


Generate event <Z> when solo event <S> was seen, but only if none of the events in list <P> were seen within <B> seconds before or <A> seconds after the solo event. The rule is instantiated through event 0x10002, and can be stopped with event <X>.
Event Variable Discriminators:
Prevent period before Solo Event <B>
Time Interval: 0 Days + 00:00:00
Solo Event <S>
Event Code: Browse...
Copy Event Variables
Copy All
Copy of Event to
Prevent period after Solo Event <A>
Time Interval: 0 Days + 00:00:00
Generated Event <Z>
Generate Event: Browse...
Copy Event Variables
Copy All
Copy of Event 0x10002 to
Prevent Events <P>
Event Code
Copy From
Copy To
Stop Event <X>

The settings in the dialog include the following:

- **Event Variable Discriminators**
  A comma-separated list of the IDs of the event variables in the event to evaluate as a part of the rule.

  NOTE
  You must enter each ID; ranges of IDs are not supported. Event variable discriminators are a general feature available for all types of rules.
- **Prevent period before Solo Event <B>**
  Sets the time period before the solo event where none of the 'prevent' events may occur (in seconds).
- **Solo Event <S>- Event Code**
  Specifies the Solo event. To specify the event code, click Browse, select the event in the Select Event dialog (which displays all of the events loaded into Event Configuration), and click OK.
- **Solo Event <S>- Copy Event Variables**
  If you want to copy the values of any event variables in the first event that activated the rule to generated event <D>, select Copy Event Variables, and do one of the following:
  – Select Copy All to copy the values of all of the variables into Count down event <D>.
  – Select Copy, and specify a comma-separated list of specific variable IDs whose values should be copied. Ranges of IDs are also supported, for example, “1-4”.

  NOTE
  For information on the proper syntax to use when specifying variable IDs or ranges of IDs using Event Configuration, see Copying Variable Values from Contributing Events to the Rule Output Even
- **Prevent period after Solo Event Action:**
  Sets the period after the solo event where none of the 'prevent' events may occur.
- **Generated Event <Z> - Event Code**
  Defines the event that will be generated when the rule triggers (when just the 'solo' event is seen). To specify the event code, click Browse, select the event in the Select Event dialog (which displays all of the events loaded into Event Configuration), and click OK.
- **Generated Event <Z> - Copy Event Variables**
  If you want to copy the values of any event variables in the first event that activated the rule to generated event <D>, select Copy Event Variables, and do one of the following:
  – Select Copy All to copy the values of all of the variables into Count down event <D>.
  – Select Copy, and specify a comma-separated list of specific variable IDs whose values should be copied. Ranges of IDs are also supported, for example, “1-4”
- **Prevent Events <P>**
  Defines a list of 'prevent' events. In the dialog, use the buttons above the list of prevent events to add new events and to modify, copy, and delete a selected event.
- **Stop Event <X> - Event Code**
  The Stop Event stops the Solo Event rule. To specify the event code, click Browse, select the event in the Select Event dialog which displays all of the events loaded into Event Configuration, and click OK.

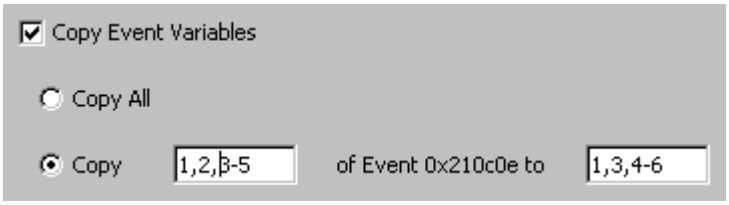
Copy Variable Values from Contributing Events to the Rule Output Event

When you use event rules, the event that is generated is sometimes generated only after multiple contributing events occur or after certain conditions are met.

By default, a rule output event has no event variables. However, you can specify optional processing. The values of the event variables in the events that contribute to the rule processing can be copied to the output event. You can then use event variable discriminators to specify event processing behaviors for the rule output event that is based on those values. And because the values of event variables in events that generate alarms are also stored in those alarms, you can use event discriminators to specify alarm processing (generation or clearing of alarms) based on the values. For an example of this usage, see Event Variable Copying and Event Discriminators.

You can specify that all of the event variables in a contributing event are copied to the rule output event, or you can specify variable IDs. Use the event variable IDs that are mapped (in the AlertMap file) to the OIDs of the variable bindings that are sent with the trap.

To copy only specific variable binding values from a contributing event to the rule output event, enter a comma-separated list of the IDs or ranges of IDs. This configuration is shown in the following image:



Use the first text box to specify the contributing event variables that you want to copy. Use the second text box to specify the event variables in the output event where the values are copied. The first ID in the “source” text box is copied into the first ID in the “target” text box, and so on.

> **NOTE**
> You can copy the value of one source variable into a target variable that has a different ID. For example, the preceding image specifies that source variable 1 is copied into target variable 1, source variable 2 is copied into target variable 3, and source variables 3, 4, and 5 are copied into target variables 4, 5, and 6, respectively.

Modifying Event Rules

You can modify event rules.

To modify event rules

1. In the Navigation panel, select the event that activates (triggers) the event rule.
2. In the Details panel, click the Event Rules tab.
   The list of event rules for the selected event appears.
3. Select the event rule to modify, and click the edit
   icon  .
4. Modify the configuration of the event rule as required.
5. Click OK.

Delete Event Rules

You can delete event rules.

To delete event rules

1. In the Navigation panel, select the event that activates (triggers) the event rule.
2. In the Details panel, click the Event Rules tab.
   The list of event rules for the selected event appears.
3. Select the event rule to delete, and click the delete
   icon  .
4. Click OK to confirm the deletion.

How to Use Procedures in Event Processing

As a DX NetOps Spectrum administrator, you are responsible for configuring DX NetOps Spectrum events. You want to configure selected event types to launch automatic actions, such as creating other events in response to selected events on selected models.
DX NetOps Spectrum includes many procedures that accomplish common tasks. The procedure files are located in *$SPECROOT*/SS/CsVendor/CA/Procedures. Each procedure XML file defines the procedure (input parameters, return value, and other options) and provides documentation about its function.
Using DX NetOps Spectrum procedures for event processing involves the following tasks:

- Understand Event Disposition Files
- Add a Procedure to an Event Map

Event Disposition Files

When DX NetOps Spectrum receives an SNMP trap and is mapped to a DX NetOps Spectrum event that is generated for the model, an *event disposition file* is used to determine how to process the event. The processing instructions in an event disposition file (an ASCII text file) can include any of the following information:

- Whether the event is logged
- The severity of the event
- Whether the event generates an alarm of a specific severity
- Whether the event clears one or more alarms
- Whether the event triggers an event rule (a series of events that are monitored and that trigger another event when they occur in a specific pattern or time frame)

DX NetOps Spectrum predefined events all have processing instructions that are defined in global and model type-specific event disposition files. In addition, whenever you create a new, custom event (either using MIB Tools, to map a trap to a new event, or later, using Event Configuration), a custom event disposition file is automatically created. Typically manual creation of the event disposition file is not required.

> **WARNING**
>
> A few types of modification, such as adding event-processing procedures to event maps, must be done in a text editor. For most other event modifications, we recommend specifying the processing instructions using Event Configuration. The Event Configuration utility can save your customizations to one or more landscapes.

If you are adding trap support for a new device, first use MIB Tools to map the traps to new DX NetOps Spectrum events and specify basic event settings. Then launch Event Configuration (it can be launched directly from MIB Tools) and can complete the event configuration. This workflow avoids most manual modifications to event disposition files. However, we have provided reference information about the proper syntax to use if manual modifications are required.

More information:
Location of Event Disposition Files
File Syntax of Event Disposition Files

File Syntax of Event Disposition Files

Each line in an event disposition (EventDisp) file is called an *event map.* Each event map can specify one or more event processing behaviors, such as whether the event is logged, or whether it generates an event. The following syntax is used for an event map:

```
eventcode E eventseverity processing parameters P procedure S sbgw processing flag
```

eventcode

Specifies the event code of the DX NetOps Spectrum event for which the processing behavior is being defined. The event code is defined in the AlertMap file that maps the trap to the event, or it is specified within the code that generates the event. The event code must be specified for DX NetOps Spectrum to process the event.

E
(Optional) Indicates that the Archive Manager logs the event data to the Distributed Data Manager (DDM) database. If the E flag is not used, the event data is temporarily logged, but it is not preserved when the SpectroSERVER is shut down and then restarted.

eventseverity
Specifies the relative severity of the event on a scale of 0 to 100, where 0 is the least severe and 100 is the most severe. If a value for event severity is not specified, DX NetOps Spectrum uses a default value of 0.

> **NOTE**
>
> The event severity parameter is not currently used by the event management system. However, if you are logging the event, you are advised to assign an event severity value. The use of this parameter can be incorporated into event processing in the future. If you are not logging the event, do not assign an event severity value.

processing_parameters
Specifies additional event processing behaviors, such as whether the event generates an alarm, clears one or more alarms, or executes one or more event rules.

P
Specifies the action code that indicates that a procedure follows.

procedure
Specifies the procedure to execute. To specify multiple procedures, use multiple instances of P *procedure*.

S *sbgw processing flag*
(Optional) Specifies whether an event should be registered for Southbound Gateway processing. It applies to modeltype specific entries only and to only those modeltypes that are derived from the southbound modeltype fragment.

An event does not require an event map in an EventDisp file, but most events have one. If an event does *not* have an event map in an EventDisp, the event is logged in the DDM database by default (which means that it is preserved when the SpectroSERVER is shut down and then restarted). However, no additional processing takes place. If an event has an event map, the event is processed according to the event map.

Note the following about EventDisp files:

- Empty lines in the EventDisp file are ignored.
- If a single event map spans multiple lines, include a backslash '\' at the end of each line so that DX NetOps Spectrum considers the next line as a part of the event map.

Location of Event Disposition Files

The name of every event disposition (EventDisp) file is always EventDisp.

> **NOTE**
>
> In the Windows environment, no file extension is used.

The EventDisp files that support the events provided with DX NetOps Spectrum are installed in the following directories:

- *$SPECROOT*/SS/CsVendor/*vendor_directory*

An EventDisp file in this location defines the processing for all events created by the developer that are *global* in scope. The *vendor_directory* variable is the name of the developer, vendor, or manufacturer for which the EventDisp file is used.

NOTE

In some instances, CA uses something other than the developer name for the developer-specific directory; for example, the IETF directory contains the EventDisp file that specifies the processing for events that result from standard RFC traps.

- *$SPECROOT*/SS/CSVendor/*vendor_directory*/*model_type_name*

An EventDisp file in this location defines the processing for all events that are created by the developer whose scope is *limited to the model type* represented by *model_type_name*. In this case, *vendor_directory* is the name of the developer, vendor, or manufacturer for which the model type was created (for example, Cisco).

NOTE

If an event map entry exists in both a global EventDisp file and a model type-specific EventDisp file, the event map entry in the model type-specific file takes precedence.

If you customize the event processing for the DX NetOps Spectrum predefined events, or if you create new custom events using MIB Tools or Event Configuration, a custom EventDisp file is created in the following directory or in one of its subdirectories:
$SPECROOT/custom/Events.

NOTE

The event map entries in a custom EventDisp file override those entries in the predefined DX NetOps Spectrum EventDisp files.

Add a Procedure to an Event Map

To specify that one or more procedures are executed when an event is processed, add the procedures to the relevant event map. The event map is included in the event disposition file. When the event is generated for a given model, the SpectroSERVER processes the event and executes the procedures.

DX NetOps Spectrum procedures can only be created using a text editor, not in the Event Configuration interface. DX NetOps Spectrum Event Configuration lets you view, but not edit, any procedures that you have associated with an event.

Associate an event procedure with an event by creating an event disposition action. Start the action with the 'P 'action code, and then contain the procedure code within double-quotation marks.

NOTE

Both the current event and the current model are required for proper procedure execution. The SpectroSERVER sets both of these parameters automatically.

Follow these steps:

1. Log in to the SpectroSERVER as an administrator.
2. Locate the event disposition files in the following directory: SS/CsVendor/*vendor_directory*
3. Find the procedure XML file that you want to use in the following directory: *$SPECROOT*/SS/CsVendor/CA/Procedures
4. Add the selected procedure to an event map using the following syntax: *event_code* E *event_severity processing_parameters* P *procedure*

***event_code* S**
Specifies the event code of the event.

E
Identifies the action code that indicates the event data is logged by the Archive Manager in the Distribute Data Manager (DDM) database.

event_severity
Specifies the relative severity of the event on a scale of 0 to 100.

processing_parameters
Specifies additional event processing behaviors, such as whether the event generates an alarm, clears one or more alarms, or executes one or more event rules.

P
Specifies the action code that indicates that a procedure follows.

procedure
Specifies the procedure to execute. To specify multiple procedures, use multiple instances of P *procedure*.
The server sets the correct environment for the procedure evaluation, with the current model and event preset.
Event disposition procedures are parsed once during server startup, producing an executable procedure object. When it receives the event for the procedure, the server evaluates the procedure object and performs the specified operations.

Example
The following event map executes a procedure that creates event 0xabcd0002 on the device model of a port, which is retrieved by reading the Device_Mdl_Handle attribute (0x10069). Event 0xabcd0002 has copies of all of the event variables in the original event.

```
0xabcd0000 E 50                                          \
    P "CreateEventWithVariables(                         \
        ReadAttribute( { C CURRENT_MODEL }, { H 0x10069 } ),  \
        { H 0xabcd0002 },                                \
        GetEventVariableList()                           \
)"
```

In the example, CURRENT_MODEL is a predefined constant that refers to the current model. The model is automatically set in the environment by the SpectroSERVER when the event is processed.
As shown in the example, a procedure must be fully enclosed in double quotes. The procedure must have a procedure name and can take parameters. Parameters follow the procedure name. Both values and constants (direct values) must be enclosed in braces and separated by commas. For example, the following syntax is valid:

```
{ U 101 } (unsigned integer with value of
 101)
{ C CURRENT_MODEL } (a constant that is defaulting to the current model handle)
```

A procedure call can be inserted by using the procedure name with its parameters. Add arguments in braces. For example:

```
ReadAttribute( { C CURRENT_MODEL }, { H 0x1006e } )
```

If a procedure spans multiple lines, use the line continuation character to indicate that each line continues a single event map.
You can use any number of procedures in an event map. No limit on the procedures that you can use from an event disposition file is enforced.

> **NOTE**
>
> The custom (user-defined) event procedures are read-only in the Event-Configuration editor.

Input Parameters

A procedure accepts the following types of input parameters:

- A value. Values are specified by type and value.
- A constant. Constants are specified by the letter C followed by the reserved word for the constant.
- Another procedure, which is evaluated when the parameter is needed and whose return value is used as the actual input parameter for the calling procedure.

Parameter Types

When you specify a direct value as a parameter, you must specify its type. Value types have a derivation hierarchy, which is shown in the following table. You can use a more specialized type for a more general type. For example, you can use a model handle in place of an unsigned integer, and you can use either of these types in place of a number.

When a parameter has a type that is not the same as or a derivative of the expected type, if the type is similar, most procedures attempt a conversion to the expected type. If a conversion fails or is not performed, an error occurs.

NOTE

You can use the ToUInteger procedure (provided with DX NetOps Spectrum) to convert a parameter value to an unsigned integer. Many times, this can help to avoid errors due to expected versus actual parameter types.

To specify a parameter type in a procedure, you specify a letter (a short symbol) for the type. For example, both of the parameters in the following procedure are handles (IDs), which are indicated by the letter 'H.' The first handle specifies the model for which to generate the event, and the second handle specifies the event to generate.
CreateEvent({ H 0x29c00003 }, { H 0xffff0000 })

The table that follows provides a short list of parameter types, their derivation hierarchy, and their associated short symbols. For the complete list of short symbols, see EventCondition Rule.

Parameter Type	Description	Short Symbol
Object	Derivation root	N/A
List	List root	N/A
AttributeList	List of attribute values	N/A
EventVariableList	List of event variables	N/A
Attribute Value	Any possible attribute value	N/A
Number	Number root	N/A
Integer	An integer value	I
Unsigned Integer	An unsigned integer value	U
DateTime	A time value	T
ModelHandle	A model handle	H
ModelType	A model type handle	H
RelationHandle	A relation handle	H
Attribute ID	An attribute ID	a
Unsigned Long	An unsigned long (64-bit) value	L
Boolean	A Boolean value	B
Double	A real value	R
String	String root	N/A
Text String	A String value	S
Octet String	An octet String value	O
Tagged Octet String	A hexadecimal tagged octet String	X
Object ID	An object ID value	o

IP Address	An IP address	A
Variable	The value of a variable	V

Constants

You specify a constant using the letter C followed by the reserved word for the constant. For example, the following procedure creates event 0xffff0000 for the current model:
CreateEvent({C CURRENT_MODEL}, { H 0xffff0000 })

You can use the following list of constants in procedures in event maps:

CURRENT_EVENT
The current event, which is set in the environment automatically by the SpectroSERVER.

CURRENT_MODEL
The current model, which is set in the environment automatically by the SpectroSERVER.

RELATION_SIDE_LEFT
The model on the left side of an association.

RELATION_SIDE_RIGHT
The model on the right side of an association.

RELATION_SIDE_EITHER
The model on either side of an association.

Return Values

Every procedure returns a value whose type depends on the function that the procedure performs, for example, a list of model handles or a Boolean value of TRUE or FALSE.

The possible types of return values are the same as the possible types of input parameters for procedures. This lets you use procedures as input parameters because they ultimately evaluate to permissible input values.

Examples of Procedures in Event Disposition Files

The following examples show how procedures can be used in event maps to implement event processing. Assume that each procedure is wrapped inside double quotes.

Add together two integer numbers (4 and 16):

```
Add( { I 16 }, { I 4 } )
```

Add together a direct value of 4 and the value in event variable 2 in the current event:

```
Add( { U 4 }, GetEventVariable( { U 2 } ) )
```

Create event 0xffff0000 for model 0x29c0003:

```
CreateEvent( { H 0x29c00003 }, { H 0xffff0000 } )
```

Create event 0xffff0000 if event variable 1 in the current event exceeds 100:

```
If( Less( { U 100 }, GetEventVariable( { U 1 } ) ),    \
 CreateEvent( { H 0x29c00003 }, { H 0xffff0000 } ), \
 Nil() )              \
)
```

Create event 0x10002 with a list of event variables. The list will be a copy of the variables in the current event. In the new event, also set event variable 2 to 1965:

```
CreateEventWithVariables( { H 0x29c00003 },       \
 { H 0x10002 },            \
 SetEventVariable( GetEventVariableList( { C CURRENT_EVENT } ),  \
 { U 2 }, \ { U 1965 }            \
 )
)
```

Create a list and add several elements. The list will contain the following elements in this order: 5,3,2,1,4,6:

```
AddTail(          \
 AddHead(         \
  AddTail(        \
   AddHead(       \
    AddHead(      \
     AddTail(     \
      CreateList(),  \
      { U 1 } ),    \
     { U 2 } ),     \
    { U 3 } ),      \
   { U 4 } ),       \
  { U 5 } ),        \
 { U 6 } )
```

Create event 0x10002 for model 0x29c00003 if the model has IP address 191.168.102.25:

```
If ( HasIPAddress( { H 0x29c00003 }, { A 192.168.102.25 } ),     \
 CreateEvent( { H 0x29c00003 }, { H 0x10002 } ),              \
Nil()                                                          \
)
```

Create event 0xffff0000 for model 0x29c00003 if attribute 0xffff0000 of the current model has the same value as event variable 1 in the current event:

```
If( Equals( ReadAttribute( { C CURRENT_MODEL }, { H 0xffff0000 }),  \
 GetEventVariable( { U 1 } )                                  \
 ),                                                            \
 CreateEvent( { H 0x29c00003 }, { H 0xffff0000 } ),              \
Nil()                                                           \
)
```

Return TRUE if the current time is 8 a.m. or later:

```
GreaterOrEqual( GetHour( GetCurrentTime() ), { U 8 } )
```

Retrieve the value of event variable 1 in the current event and write it to attribute 0xffff0001 in the current model:

```
WriteAttribute( { C CURRENT_MODEL }, { H 0xffff0001 }, GetEventVariable( { U 1 } ) )
```

Create event 0xffff0000 for model 0x29c00003 if exactly 3 event variables in the current event's event variable list have a value of 4. The procedure iterates over the list, assigning each element in the list to the loop variable 'X' in turn and checking its value. If the value of the element is 4, the return variable 'ret' is incremented. After iterating through the list, the return value is checked, and the event is generated if its value is 3:

```
If( Equals( ForEach( GetEventVariableList(),                \
  { Variable X },                                  \
```

```
   { Variable ret },                                  \
   { U 0 },                                          \
   If( Equals( { Variable X },                       \
    { U 4 } ),                                  \
    Assign( { Variable ret },                   \
     Add( { Variable ret },                \
      { U 1 } )                            \
    ),                                          \
    Nil()                                       \
   )                                                 \
  ),                                                       \
  { U 3 } ),                                               \
  CreateEvent( { H 0x29c00003 }, { H 0xffff0000 } ),       \
  Nil()                                                    \
 )
```

Parse a Single Varbind into Multiple Event Variables

In this example, a device produces a trap that contains a single varbind. That varbind contains multiple components that are required for the correct processing of the alarm. A procedure is required to extract the components and place them into event variables, where they can be used as event discriminators.
We have analyzed the trap and know that the varbind contains a string with a well-defined pattern: ERROR TEXT:SEVERITY. We also know that the ERROR TEXT could be any arbitrary string, but that the SEVERITY is limited to a set of values: Critical, Major, Minor, Clear, None, or Special.
Therefore, we must create a pattern of (.*):(Critical|Major|Minor|Clear) to match our varbind. We use the left-hand side as a discriminator, and the right side as the severity. We also want a unique alarm instance that is created for each ERROR TEXT. And we want the Clear trap to clear only matching ERROR TEXT alarms.
We want to create a single event that is capable of handling many different error code and severity combinations. Using a procedure, we instruct the event to create multiple event variables. But first, we must create an entry in the AlertMap file for the trap.
Follow these steps:

1. Log in to the SpectroSERVER as an administrator.
2. Locate the AlertMap file in the following directory: SS/CsVendor/*vendor_directory*

   NOTE

   An AlertMap file maps an alert to a DX NetOps Spectrum event. AlertMap files are automatically created when you map a trap to an event using MIB Tools.
3. Open the file in a text editor.
4. Use the following syntax for the new entry: *Alert Code Event Code OID Map*where the OID Map includes the OID, the value variable ID, and the instance variable ID.
   For example, the following syntax provides a fictional example:
   1.3.6.1.4.1.17844.1.2.6.0 0x05a91900 1.3.6.1.4.1.17844.1.1.1(1,0)
5. Create a regexp pattern to extract the error code into one variable, and the severity into another variable.
   In this example, we create a procedure that parses any number of tokens so that we can reuse it for other, similar requirements:
   Pseudo Code:

   ```
   Given a text string and a pattern
   Set the starting index
   If the text string matches the pattern then for each () in the pattern,
   extract the text enclosed and put it into an event variable at the current index; increment the index
   else throw an error message
   ```

```
When all the () are exhausted, create an event with the new event variable list
```

Example from an EventDisp File:

```
0x05a91000 P " \
SetVariable({V pattern},{S \"(.*):(Critical|Major|Minor|Clear|None|Special)\"}, \
 SetVariable({V counter},{U 500}, \
If(Regexp(GetEventVariable({U 1}),{V pattern}), \
CreateEventWithVariables({C CURRENT_MODEL},{H 0x05a91002}, \ ForEach(GetRegexpList(\
GetEventVariable({U 1}),{V pattern}),{Variable X}, \
{Variable retVal},GetEventVariableList(), \
Prog2(Assign({V retVal},SetEventVariable({V retVal},{V counter},{Variable X})), \
 Assign({V counter},Add({V counter},{U 1})))))), \
CreateEventWithVariables({C CURRENT_MODEL},{H 0x05a91001},
 GetEventVariableList())))))"
```

NOTE

Validation occurs during the processing. If the input string does not match the pattern (for example, the syntax ERROR TEXT:BadSeverity does not match), an alarm is created. The alarm details indicate that the processing did not complete because the input was not formatted correctly. Including validation is extremely helpful in troubleshooting. For a complete walkthrough of this code sample, see "Code Walkthrough," below.

6. Add the procedure to an event map using the syntax that is described in Add a Procedure to an Event Map.
   The server sets the correct environment for the procedure evaluation, with the current model and event preset.

Code Walkthrough

```
0x05a91000 P " \
SetVariable({V pattern},{S \"(.*):(Critical|Major|Minor|Clear|None|Special)\"}, \ SetVariable({V counter},{U
 500}, \
If(Regexp(GetEventVariable({U 1}),{V pattern}), \
CreateEventWithVariables({C CURRENT_MODEL},{H 0x05a91002}, \ ForEach(GetRegexpList(\
GetEventVariable({U 1}),{V pattern}),{Variable X}, \
{Variable retVal},GetEventVariableList(), \
Prog2(Assign({V retVal},SetEventVariable({V retVal},{V counter},{Variable X})), \ Assign({V counter},Add({V
 counter},{U 1})))))), \
CreateEventWithVariables({C CURRENT_MODEL},{H 0x05a91001}, GetEventVariableList())))))"
```

{V pattern} is the extraction pattern. If you wanted 3 tokens you would only have to change this pattern. The following example shows an extraction pattern:

```
(.*):(.*):(Something|Else)
The (.*) brown (fox|duck) (jumped|slid) over the lazy (.*) => {S 501}=quick {S 502}=fox
{S503}=jumped {S 504}=dog
```

{V counter} is the starting index. The GetRegexpList function determines that the entire string is contained at the counter, while the first extracted token is at counter + 1.

GetEventVariable({U 1}) returns the value of varbind 1 (you might know it as {S 1} or {I 1} in event messages). This instruction effectively means, "Get the event variable at index 1." If your variable is in varbind 100, use U 100, for example.
ForEach says, "For each item in the list returned by GetRegexpList, do something."
GetRegexpList returns a list of extracted tokens, but they cannot solely be used as event variables. They must be written to the EventVariableList.
Prog2 is like a code block that lets two independent actions execute sequentially. In this case, the code instructs, "Assign the current token to the event variable list at the current index, then increment the index counter."

Logging Errors in Event Procedures

You can log the errors that the SpectroSERVER encounters while parsing a procedure XML file or while executing a procedure in an event map to an error log file.
Errors that are found while parsing a procedure file are written to the file that is specified in the procedure_error_file parameter in the SpectroSERVER .vnmrc resource file. Errors that are found while executing a procedure that is used in an event map are written to the event disposition error file. The event disposition filename is specified in the event_disp_error_file parameter. A procedure-specific error file can also be used. Its filename is specified in the procedure_error_file parameter. A filename of "stderr" for these parameters redirects the output to the server console output.

Troubleshooting Event Procedures

Using DebugValue procedure

You can use the DebugValue procedure to assist you in troubleshooting procedures used in event processing that are not performing as you intend. The DebugValue procedure prints the value of a parameter to a log file. The error log file is specified in the event_disp_error_file parameter in the SpectroSERVER .vnmrc file.

> **NOTE**
>
> For information on how to set the event_disp_error_file parameter in the .vnmrc file in the Input Parameters section

The DebugValue procedure accepts the following input parameters:

- A text string that is printed before the value is printed. This parameter is especially helpful for differentiating values when you are using more than one DebugValue procedure in a given event procedure.
- Any value (value, constant, or another procedure).

The DebugValue procedure returns the value that is passed in as the second input parameter (which is unchanged). As a result, you can use it as an input parameter in any other procedure without affecting the execution of that procedure.
Example
The following DebugValue procedure prints a text string, followed by the value of a read operation on an attribute of the current model. It returns the value of the read operation, which is then used as the first input parameter in the Equals procedure.

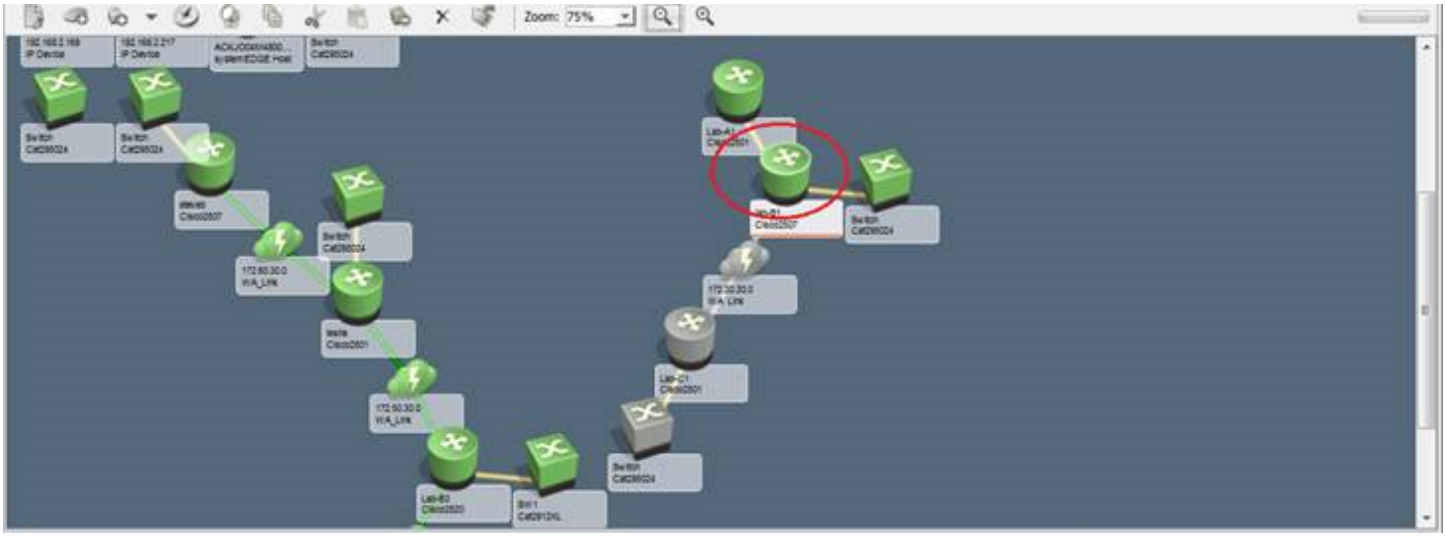
```
If( Equals( DebugValue( { S \"Attribute value is:\" },           \
  ReadAttribute( { C CURRENT_MODEL }, { H 1 } ),        \
  GetEventVariable( { U 1 } )                            \
 ),                                                     \
 CreateEvent( { H 0x29c00003 }, { H 0xffff0000 } ),           \
Nil()                                                    \
)
```

Spectrum Fault Isolation not suppressing or asserting alarms as expected

Symptom:

During a major outage, we did not receive an expected critical alarm on a model. No alarm was asserted on the model as expected. Spectrum Fault Isolation not suppressing or asserting alarms as expected.

In the following screen shot, the circled router model was down and should have alarmed Critical with the "DEVICE HAS STOPPED RESPONDING TO POLLS" alarm. However, Spectrum did not assert a Critical alarm. The model stayed green but could not be contacted by Spectrum.



Cause:

The 0x10d35 event associated with the "DEVICE HAS STOPPED RESPONDING TO POLLS" alarm was modified as follows:

0x00010d35 R CA.EventPair, 0x10d30, "0xfff0000e -:-", 600 R CA.EventCombo, "0xfff00030 -:-", 300, "0x10d30 -:-"

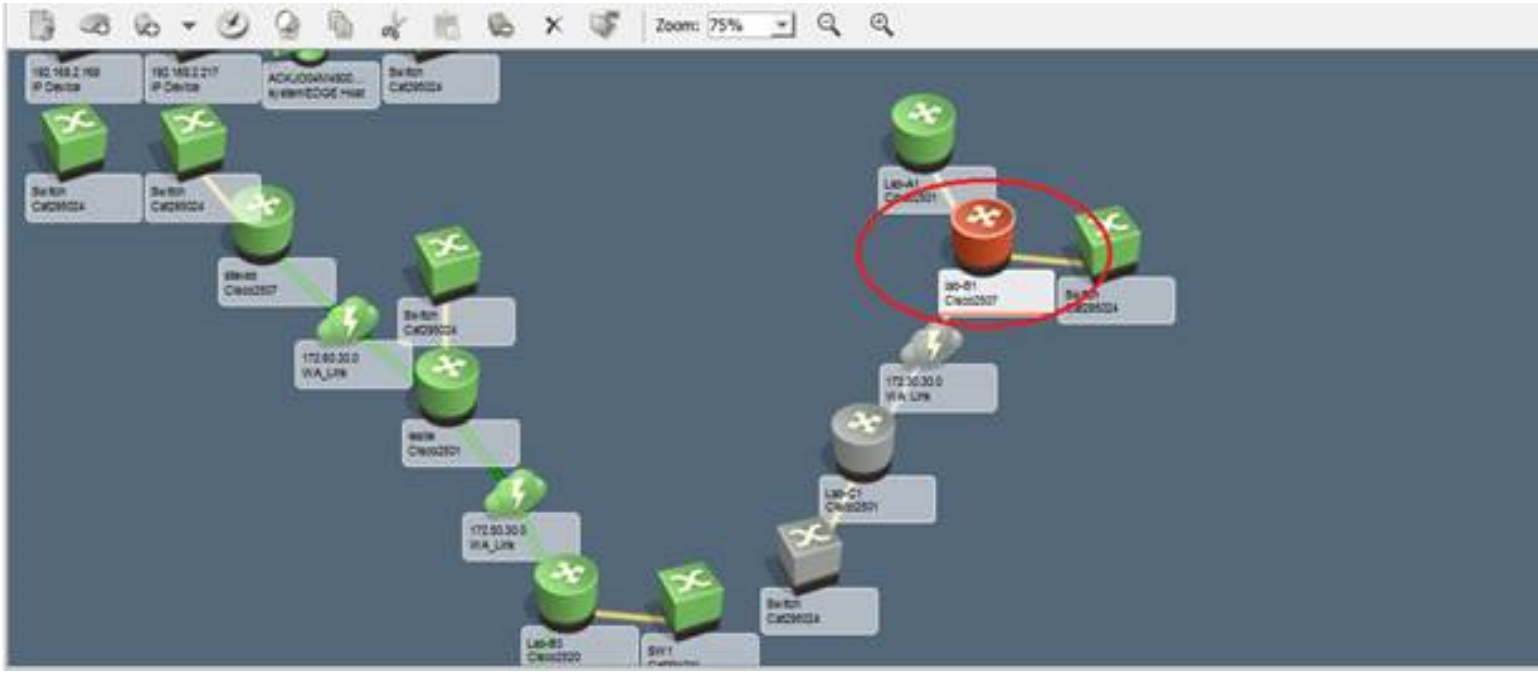
Out of the box, the 0x10d35 event is defined in the $SPECROOT/SS/CsVendor/Cabletron/EventDisp file as follows:

0x00010d35 E 75 A 3, 0x00010009,N

Resolution:

Modifying the out of the box events may have undesirable results. The underlying Fault Isolation, Impact Analysis and Fault Suppression code is looking for specific events with specific conditions and specific probable cause codes in order to function properly. Changing any one of these from the default may have undesirable results.

In the above scenario, after changing the 0x10d35 event back to the out of the box definition and replicating the outage, Spectrum alarmed as expected:



AlertMap Files

SNMP Trap Overview

An SNMP trap is sent out as a trap PDU (Protocol Data Unit). The PDU contains the following pieces of information:

- **Enterprise OID**
  Identifies the company responsible for a device sending the trap. For example, 1.3.6.1.4.1.X is an enterprise OID where X identifies the enterprise (for example, Sun). The numbers preceding the X represent a hierarchy of global bodies responsible for the management of information. The Internet Assigned Numbers Authority (IANA) allocates the enterprise level numbers that identify companies and their management MIBs on the MIB tree. For more information, see their website, http://www.iana.org.
- **Network Address**
  Specifies the network address of the managed element initiating the trap.
- **Generic Trap Identifier**
  This can be a value from 0 through 6. There are six standard industry traps within the SNMP protocol. These traps have a generic trap identifier of 0-5. The number 6 indicates that the trap is an enterprise-specific trap. Enterprise specific traps are proprietary traps that are created for developer-specific types of managed nodes. Proprietary MIBs define these traps.
- **Specific Trap**
  Specifies the specific trap number as listed in the trap definition of the trap MIB.
- **Time Stamp**
  Specifies the time at which the trap was created.
- **Variable Bindings**
  Specifies the variables and values that are defined in the trap. These variables are generally pointers to other MIB objects.

For example, when a redundant power supply fails in a Cisco router or switch, the following information is sent in the trap PDU:

- Network address of the device
- Timestamp of the trap
- Enterprise OID: 1.3.6.1.4.1.9
- Generic Trap ID: 6
- Specific Trap ID: 5
- Variable Binding(s): 1.3.6.1.4.1.9.9.13.1.5.1.2, 1.3.6.1.4.1.9.9.13.1.5.1.3

The network address and the timestamp information vary depending on the device and the time that the trap was sent.

The enterprise OID identifies the vendor company using the last two digits. In this example, 9 indicates a trap that is generated by a Cisco device.

The generic trap ID of 6 indicates that this trap is an enterprise-specific trap that is defined by a proprietary MIB. The specific trap ID of 5 is the number that is assigned to the trap in the referenced Cisco MIB. The following portion of the Cisco environment monitoring MIB (CISCO-ENVMON-MIB) defines this trap.

```
ciscoEnvMonRedundantSupplyNotification TRAP-TYPE
-- Reverse mappable trap
    ENTERPRISE ciscoEnvMonMIBNotificationPrefix
    VARIABLES {
        ciscoEnvMonSupplyStatusDescr, ciscoEnvMonSupplyState }
-- Status
-- mandatory
    DESCRIPTION
        "A ciscoEnvMonRedundantSupplyNotification is sent if the
        redundant power supply (where extant) fails. Since such a
        notification is usually generated before the shutdown state is
        reached, it can convey more data and has a better chance of being
        sent than does the ciscoEnvMonShutdownNotification."
    ::= 5
```

This trap definition has two variables (ciscoEnvMonSupplyStatusDescr and ciscoEnvMonSupplyState) that are sent as variable bindings. Such variables are actually references to other managed objects defined by the MIB. The OIDs sent in the variable bindings represent these managed objects.

The following excerpts from the MIB define ciscoEnvMonSupplyStatusDescr and ciscoEnvMonSupplyState:

```
ciscoEnvMonSupplyStatusDescr OBJECT-TYPE
    SYNTAX DisplayString(SIZE(0..32))
--         Rsyntax OCTET STRING(SIZE(0..32))
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Textual description of the power supply being instrumented.
        This description is a short textual label, suitable as a human-
        sensible identification for the rest of the information in the
        entry."
    ::= { ciscoEnvMonSupplyStatusEntry 2 }
ciscoEnvMonSupplyState OBJECT-TYPE
    SYNTAX CiscoEnvMonState
--         Rsyntax INTEGER {
--             normal(1),
--             warning(2),
--             critical(3),
```

```
--             shutdown(4),
--             notPresent(5),
--             notFunctioning(6)
--             }
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The current state of the power supply being instrumented."
    ::= { ciscoEnvMonSupplyStatusEntry 3 }
```

The ciscoEnvMonSupplyStatusDescr variable has a string value describing the power supply, and the ciscoEnvMonSupplyState variable can have a value of 1, 2, 3, 4, 5 or 6 depending on its status. The current value for each variable is detected using the variable bindings in the trap PDU.

About Mapping a Trap to a DX NetOps Spectrum Event

When DX NetOps Spectrum receives a trap, it must be associated with a specific DX NetOps Spectrum event for DX NetOps Spectrum to use the trap information. This association is made in an AlertMap file.

NOTE
DX NetOps Spectrum can receive SNMP traps from devices that have been modeled using the Pingable model type. To enable this association, you must set the enable_traps_for_pingables variable in the DX NetOps Spectrum .vnmrc file to TRUE. For information about how to do this, see the Distributed SprectroSERVER Administration section. The AlertMap file for the Pingable model type is located in the <*$SPECROOT*>\SS\CsVendor\Cabletron\Pingable directory.

About Processing Alerts with AlertMap Files

The IP address of the managed element issuing the trap is sent as a part of the trap information. DX NetOps Spectrum uses the IP address to identify the model that represents the managed element issuing the trap. From this information, DX NetOps Spectrum is able to identify the model type that is associated with the managed element.

To process the trap correctly, DX NetOps Spectrum refers to the AlertMap file. Depending on where they reside, AlertMap files can be applied globally (that is, to all DX NetOps Spectrum model types) or only to models of a specific model type.

Thus, if you want to change the trap mapping for a particular model type only, you must modify the AlertMap file for that model type. If an AlertMap file at the model type level does not exist, you must create one and must make your changes there.

If you want to change how an alert is mapped globally (that is, for all model types), you need to find all AlertMap files that specify a mapping for that trap, and make your changes to each one of these files.

NOTE
Global AlertMap files apply to all vendor model types, not just those models types created by the vendor who created the file.

AlertMap File Location

NOTE
Default trap mappings are located in the <*$SPECROOT*>/SS/CsVendor directory while MIB Tools customized trap mappings are located in the <*$SPECROOT*>/custom/Events/AlertMap directory. MIB Tools' customized trap mappings are preserved when you upgrade DX NetOps Spectrum, and they take precedence over any existing default trap mappings. For more information about working with custom trap mappings and default trap mappings, see Map Tab.

Global AlertMap files are located in <*$SPECROOT*>/SS/CsVendor/<*developername*>/AlertMap, where <*developername*> is usually the name of the vendor, manufacturer, or developer that is associated with the model type.

In some instances, CA uses something other than the developer name for the directory. For example, the IETF directory contains the AlertMap file that maps standard RFC traps.

If the AlertMap file is specific to the model type, it is located in <*$SPECROOT*>*/SS/CSVendor/* <*developername*>/ <*model_type_name*>AlertMap, where:

<*developername*> is the name of the vendor, manufacturer, or developer that is associated with the model type (for example, Compaq). If you are developing your own management module, your developer name is used here.

<*model_type_name*> is the name of the model type.

If a mapping for a trap exists in both a model type AlertMap file and a global AlertMap file, the model type AlertMap file takes precedence for that particular model type.

When you map traps using MIB Tools, entries are generated in the following file on all SpectroSERVERs in your DSS environment:

<*$SPECROOT*>/custom/Events/AlertMap

The mapping information for a trap in these files overrides any mapping information that previously existed for that same trap in other files or directories on the SpectroSERVER.

NOTE
The name of every AlertMap file is always “AlertMap.” No file extension is used in the Windows environment.

NOTE

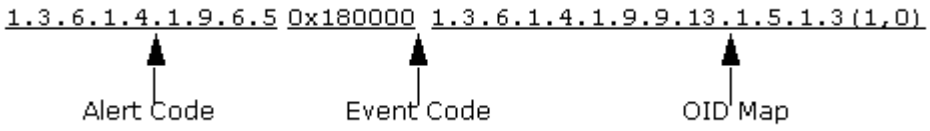
For more information about managing MIBs and traps, see the Certifications section.

NOTE

Trap mappings for the two traps, jnxVpnIfUp and jnxVpnIfDown are removed from the AlertMap file located at <$SPECROOT>/SS/CSVendor/junpr_rtr/ directory. The VPN Manager Explorer view does not list the Device models under VPN Site models. So, the JnxVPN traps assert alarms on VPN Manager model instead on the device model.

AlertMap File Syntax

If the appropriate AlertMap file is found, DX NetOps Spectrum looks for an entry matching the trap. Each entry in the Alert Map file has three components: the alert code, the event code, and the OID map.



Alert Code

The alert code consists of three pieces of information from the trap: the Enterprise OID string, the generic trap identifier, and the specific trap identifier.

1.3.6.1.4.1.9: Enterprise OID (in this case indicating a Cisco device).

6: The Generic Trap Identifier (in this case the 6 indicates an Enterprise specific trap).

5: Specific Trap Identifier (in this case 5).

Event Code

The event code is a 4-byte integer that is expressed in hexadecimal format. Each event code has two parts: the first 2 bytes contain the developer ID of the developer who created the file, and the last 2 bytes identify the event with a unique number relative to all other event codes for that particular developer.

If the event code is zero, an event is not generated for that particular alert.

If the event code is non-zero, DX NetOps Spectrum maps the alert variables to event variables using the OID map.

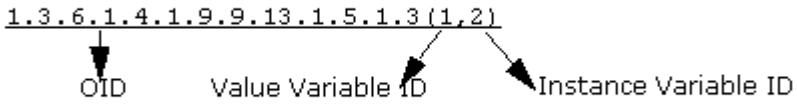
OID Map

The trap that you are mapping can include one or more pieces of variable information, which is known as variable bindings. Each of these variable bindings provides information about the trap. You can select to map these bindings to event variables so that the information is used by DX NetOps Spectrum.

Variable bindings and the information about how DX NetOps Spectrum maps them are specified in the OID map section of the alert map entry. A single alert map entry can have multiple OID maps that are associated with it indicating that there were many variable bindings that are sent with the SNMP trap. These OID maps are separated by the backslash (\) character and a new line as in the following example.

```
1.3.6.1.4.1.9.6.5 0x180000 1.3.6.1.4.1.9.9.13.1.5.1.3(1,2)\
1.3.6.1.4.1.9.9.13.1.5.1.2(4,0)
```

As shown in the following example, the OID map can be broken down into three parts: the OID, the value variable ID, and the instance variable ID.



The OID identifies the specific variable being sent with the trap. The previous OID references the ciscoEnvMonSupplyState variable in the Cisco Environment monitoring MIB (CISCO-ENVMON_MIB).

The value variable ID stores the value of the variable that is sent in the variable binding. Any integer value can be used here; however, it must be different from the integer that is used for the instance variable ID. A value of zero indicates that you do not want to store the value of the variable binding.

The instance variable ID stores the instance portion of the OID. If your variable binding identifies a particular object from a table variable within the trap MIB, it is likely to include an instance ID. Any integer value can be used here; however, it must be different from the integer that is used for the value variable ID. A value of zero indicates that you do not want to store the value of the instance variable.

> **WARNING**
> Do not use spaces between the OID, the value variable ID, and the instance variable ID.

Comments

There are two comment identifiers that allow you to add comments to the AlertMap file: # or //. Text that is entered after one of these identifiers and before the start of the next line is ignored when processing the AlertMap file.

For example:

```
#Comment
0.0 0x10306 #Comment
1.0 0x10307 //Comment
```

```
2.0 0x220001  1.3.6.1.2.1.2.2.1.1(1,2)
3.0 0x220002  1.3.6.1.2.1.2.2.1.1(1,2)\
              1.3.6.1.2.1.2.2.1.3(4,0)
//Comment
//Comment
4.0 0x1030a
5.0 0x1030b
1.3.6.1.4.1.45.6.271 0x1060f 1.3.6.1.4.1.45.1.2.1.9.2.1.2(3,4) #Comment
```

As shown in OID Map, the backslash (\) character is used as a line continuation character. You can also use comments when using the line continuation character.

For example:

```
3.0 0x220002  1.3.6.1.2.1.2.2.1.1(1,2)\ #Comment
              1.3.6.1.2.1.2.2.1.2(3,0)\
                //Comment
              1.3.6.1.2.1.2.2.1.3(4,0)
```

NOTE
Inline and multiline comments are not supported.

How DX NetOps Spectrum Maps Alert Variables to Event Variables

DX NetOps Spectrum maps alert variables to event variables as follows:

1. DX NetOps Spectrum scans the AlertMap file entry to find an OID Map whose OID either exactly matches the trap variable's OID, or is a prefix of the trap variable's OID.
2. If more than one OID map has an OID that is a prefix of the trap variable's OID, DX NetOps Spectrum chooses the OID with the longest prefix (best match).
3. If an OID map is found, DX NetOps Spectrum examines its value variable ID to decide whether to translate the trap variable's value into an event variable. If an OID map is not found, DX NetOps Spectrum ignores the alert variable.
4. If the value variable ID is zero, DX NetOps Spectrum ignores the trap variable's value. Otherwise, the type, length, and value of the event variable are obtained from the trap variable, and an event variable is constructed. The value variable ID can then be used to represent the value of the variable binding in an Event Format file.
5. DX NetOps Spectrum examines the OID Map to decide whether to translate the trap variable's instance ID into an event variable. If the OID in the OID Map exactly matches the trap variable's OID, there is no instance ID to translate. If the instance variable ID in the parameter is zero, the instance OID is ignored. If the instance ID exists and the instance variable ID is non-zero, an event variable is constructed. The instance variable ID can then be used to represent the value of the OID instance in an Event Format file.

Error Messages

If DX NetOps Spectrum is unable to identify the model for a given IP address, an event is generated on behalf of the VNM model indicating that a trap was received from an unknown SNMP device (event 0x00010802). Contained within this event are details about the trap, including the agent IP address, enterprise OID, trap code, community name, and variable binding data.

If DX NetOps Spectrum can properly identify the model for the trap source, but it cannot find an entry for the specific trap code in the AlertMap file, an event indicating that an unknown alert was received (event 0x00010801) is generated on behalf of that model. Contained within the event are details about the trap, including the agent IP address, device type, device time, trap type, and variable binding data.

SNMPv2 Support

DX NetOps Spectrum supports the receipt and processing of SNMPv2 format InformRequests and traps as defined by RFC 2576, “Coexistence between Version 1, Version 2, and Version 3 of the Internet-Standard Network Management Framework.”

InformRequest Support

When DX NetOps Spectrum receives an SNMPv2 InformRequest, it is decoded and converted to SNMPv1 format as specified in RFC 2576. A response to the InformRequest is generated and sent back to the inform originator.

How an SNMPv2 Trap is Mapped to a DX NetOps Spectrum Event

If you are mapping SNMPv2 traps to DX NetOps Spectrum events, you need to reference the translated trap in the AlertMap file.

The following is an overview of how an SNMPv2 trap is translated. For complete information about this process, see RFC 2576. Section 3.2 of the RFC explains how an SNMPv2 trap is translated into an SNMPv1 trap so that it can be used in an SNMPv1 environment.

1. The SNMPv1 enterprise OID for the trap is determined as follows:
   a. If the SNMPv2 snmpTrapOID parameter is one of the standard traps (defined in RFC 1907), then set the value of the SNMPv1 enterprise parameter to the value of the snmpTrapEnterprise.0 variable binding if it exists. If it does not exist, set the value to snmpTraps.
   b. If the SNMPv2 snmpTrapOID parameter is not a standard trap, the following criteria is used:
      If the next-to-last sub-identifier of the snmpTrapOID is 0, then find the enterprise number by removing the last two sub-identifiers.
      snmpTrapOID = 1.3.6.1.4.1.1916.4.1.0.1
      Enterprise number = 1.3.6.1.4.1.1916.4.1
      If the next-to-last number is not 0, then find the enterprise number by removing the last sub-identifier.
      snmpTrapOID = 1.3.6.1.4.1.5486.1.3.8
      Enterprise number = 1.3.6.1.4.1.5486.1.3
2. The SNMPv1 generic trap identifier for the trap is determined as follows:
   a. If the SNMPv2 snmpTrapOID parameter is a standard trap as defined in RFC 1907, use the standard generic trap parameter for that trap (0-5).
      standard trap = 1.3.6.1.6.3.1.1.5.1 (cold start)
      generic trap identifier = 0
   b. If the SNMPv2 snmpTrapOID parameter is not a standard trap, set the generic-trap parameter to 6.
      snmpTrapOID = 1.3.6.1.4.1.5486.1.3.8
      generic trap identifier = 6
3. The SNMPv1 specific trap parameter is determined as follows:
   a. If the SNMPv2 snmpTrapOID parameter is a standard trap as defined in RFC 1907, set the specific-trap parameter to 0.
      standard trap = 1.3.6.1.6.3.1.1.5.1 (cold start)
      specific trap parameter = 0
   b. If the SNMPv2 snmpTrapOID parameter is not a standard trap, set the specific-trap parameter to the last sub-identifier of the SNMPv2 snmpTrapOID parameter.
      snmpTrapOID = 1.3.6.1.4.1.5486.1.3.8
      specific trap parameter = 8
4. The SNMPv2 variable bindings are converted directly to SNMPv1 bindings. As noted in RFC 2576, variable bindings of type Counter64 cannot be translated.

Example:

If you are sending the following SNMPv2 snmpTrapOID to the SpectroSERVER: 1.3.6.1.4.1.5486.1.3.8

Use the following OID as the alert code in the AlertMap file: 1.3.6.1.4.1.5486.1.3.6.8.

Enterprise OID: 1.3.6.1.4.1.5486.1.3

Generic Trap Number: 6

Specific Trap Number: 8

About Event Disposition Files

When DX NetOps Spectrum receives an SNMP trap and is mapped to a DX NetOps Spectrum event that is generated for the model, an *event disposition file* is used to determine how to process the event. The processing instructions in an event disposition file (an ASCII text file) can include any of the following information:

- Whether the event is logged
- The severity of the event
- Whether the event generates an alarm of a specific severity
- Whether the event clears one or more alarms
- Whether the event triggers an event rule (a series of events that are monitored and that trigger another event when they occur in a specific pattern or time frame)

DX NetOps Spectrum predefined events all have processing instructions that are defined in global and model type-specific event disposition files. In addition, whenever you create a new, custom event (either using MIB Tools, to map a trap to a new event, or later, using Event Configuration), a custom event disposition file is automatically created. Typically manual creation of the event disposition file is not required.

> **WARNING**
> A few types of modification, such as adding event-processing procedures to event maps, must be done in a text editor. For most other event modifications, we recommend specifying the processing instructions using Event Configuration. The Event Configuration utility can save your customizations to one or more landscapes.

If you are adding trap support for a new device, first use MIB Tools to map the traps to new DX NetOps Spectrum events and specify basic event settings. Then launch Event Configuration (it can be launched directly from MIB Tools) and can complete the event configuration. This workflow avoids most manual modifications to event disposition files. However, we have provided reference information about the proper syntax to use if manual modifications are required.

Location of Event Disposition Files

The name of every event disposition (EventDisp) file is always EventDisp.

> **NOTE**
> In the Windows environment, no file extension is used.

The EventDisp files that support the events provided with DX NetOps Spectrum are installed in the following directories:

- *$SPECROOT*/SS/CsVendor/*vendor_directory*
  An EventDisp file in this location defines the processing for all events created by the developer that are *global* in scope. The *vendor_directory* variable is the name of the developer, vendor, or manufacturer for which the EventDisp file is used.

  > **NOTE**
  > In some instances, CA uses something other than the developer name for the developer-specific directory; for example, the IETF directory contains the EventDisp file that specifies the processing for events that result from standard RFC traps.
- *$SPECROOT*/SS/CSVendor/*vendor_directory*/*model_type_name*
  An EventDisp file in this location defines the processing for all events that are created by the developer whose scope is *limited to the model type* represented by *model_type_name*. In this case, *vendor_directory* is the name of the developer, vendor, or manufacturer for which the model type was created (for example, Cisco).

NOTE
If an event map entry exists in both a global EventDisp file and a model type-specific EventDisp file, the event map entry in the model type-specific file takes precedence.

If you customize the event processing for the DX NetOps Spectrum predefined events, or if you create new custom events using MIB Tools or Event Configuration, a custom EventDisp file is created in the following directory or in one of its subdirectories:

$SPECROOT/custom/Events

NOTE
The event map entries in a custom EventDisp file override those entries in the predefined DX NetOps Spectrum EventDisp files.

File Syntax of Event Disposition Files

Each line in an event disposition (EventDisp) file is called an *event map*, and each event map can specify one or more event processing behaviors, such as whether the event should be logged or whether it should generate an event. The following syntax is used for an event map:

```
<eventcode> E <eventseverity> <processing parameters> S <sbgw processing flag>
```

- **<eventcode>**
  Specifies the event code of the DX NetOps Spectrum event for which the processing behavior is being defined. The event code is defined in the AlertMap file that maps the trap to the event, or it is specified within the code that generates the event. The event code must be specified for DX NetOps Spectrum to process the event.
- **E**
  (Optional) Indicates that the event data should be logged by the Archive Manager to the Distributed Data Manager (DDM) database. If the E flag is not used, the event data is temporarily logged, but it is not preserved when the SpectroSERVER is shut down and then restarted.
- **<eventseverity>**
  Specifies the relative severity of the event on a scale of 0 to 100, where 0 is the least severe and 100 is the most severe. If a value for event severity is not specified, DX NetOps Spectrum uses a default value of 0.

  NOTE
  The event severity parameter is not currently used by the event management system. However, if you are logging the event, you are advised to assign an event severity value since use of this parameter may be incorporated into event processing in the future. If you are not logging the event, an event severity value should not be assigned.
- **<processing parameters>**
  Specifies additional event processing behaviors, such as whether the event generates an alarm or clears one or more alarms. The topics in this section provide detailed information on the proper syntax for various event processing behaviors.
- **S <sbgw processing flag>**
  (Optional) Specifies whether an event should be registered for Southbound Gateway processing or not. It applies to modeltype specific entries only and only those modeltypes derived from the southbound modeltype fragment. The S is followed by either of the following sbgw processing flags:
  - + (process for Southbound Gateway: S+)
  - - (do not process for Southbound Gateway: S-)

Because the default sbgw processing flag setting is '+', an entry without the 'S' flag present will always be processed by the Southbound Gateway. Setting any entry to 'S-' adds a modeltype specific EventDisp action without it being processed by the Southbound Gateway.

While an event does not require an event map in an EventDisp file, most events have one. If an event does *not* have an event map in an EventDisp, the event is logged in the DDM database by default (which means it is preserved when the SpectroSERVER is shut down and then restarted), but no additional processing takes place. If an event *does* have an event map, the event is processed according to the event map.

Note the following about EventDisp files:

- Empty lines in the EventDisp file are ignored.
- If a single event map spans more than one line, a backslash '\' at the end of each line must be used to help ensure that DX NetOps Spectrum considers the next line as a part of the event map.

Generating Alarms

The following syntax is used to generate an alarm:

```
<eventcode> E <eventseverity> A <alarmseverity>,<alarmcause>
```

- **A**
  Indicates that an alarm is generated when the specified event occurs.
- **<alarmseverity>**
  Specifies a number between 0 and 6 that identifies the severity of the alarm:
  – 0 (Normal)
  – 1 (Minor)
  – 2 (Major)
  – 3 (Critical)
  – 4 (Maintenance)
  – 5 (Suppressed)
  – 6 (Initial)
  Each severity in the preceding list is associated with a color-coded condition. When an alarm is asserted on a model, the associated condition color is displayed on the model's icon to reflect the alarm status.
  You can also specify an alarm severity of Variable or Conditional, each of which evaluates to one of the numeric severity levels. The following syntax is used to indicate a severity level of Variable:
  ```
  { v <event_variable_ID> }
  ```
  – **<event_variable_ID>**
    Specifies the ID of the event variable in the event to evaluate to determine the alarm severity level.
  Similarly, the following syntax is used to indicate a severity level of Conditional:
  ```
  { v <event_variable_ID> <folder_name>.<file_name> }
  ```
  – **<event_variable_ID>**
    Specifies the ID of the event variable in the event to evaluate.
  – **<folder_name>**
    Specifies the severity mapping file to use to determine the alarm severity level.
  – **<file_name>**
    Specifies the severity mapping file to use to determine the alarm severity level.

    NOTE
    If a severity is not specified in the event map, DX NetOps Spectrum uses a default value of 0 (Normal).
- **<alarmcause>**
  Specifies a number that is used to identify the probable cause file that contains the messages associated with the alarm.

Example:

The following event map serves as an example:

```
0x3e00002 E  50  A  2,0x3e00003
```

The example specifies the following about an event with an event code of 0x3e00002:

- It is logged in the Distributed Data Manager (DDM) database by the Archive Manager for historical and reporting purposes.

  NOTE
  Events for a model that are not logged in the DDM database are displayed on the Events tab in OneClick only if they are generated while the Events tab for that model is displayed.
- It has a severity level of 50.
- It generates a major (orange) alarm whose alarm cause code is 0xe00003.

Generating Alarms for Events Based on the Values of Event Variables

Event discriminators are references to event variables that let you to generate alarms for events based on the values of the variables.

The following syntax is used to generate an alarm using event discriminators:

```
<eventcode> E <eventseverity> A <alarmseverity>,<alarmcause>,<eventdiscriminators>
```

where <eventdiscriminators> is a comma-separated list of one or more event variable IDs that indicate the event variables to examine to determine whether to generate an alarm. These are the event variable IDs that are mapped (in the AlertMap file) to the OIDs of the variable bindings sent with the trap.

The following event map serves as an example:

```
0x3b10011 E 70 A 1,0x3b10011,1,3
```

This example specifies that when an event with an event code of 0x3b10011 is generated, if an existing alarm with an alarm cause code of 0x3b10011 already exists on the model, another alarm for the same event is generated only if the values for *both* event variables 1 and 3 are *different* in the new event when compared to current alarms on the model generated from the same event.

NOTE
The values of event variables are also stored in the alarms that are generated based on events. This means you can also use event discriminators to differentiate multiple occurrences of an alarm and clear alarms based on the values of the variables.

Generating Alarms Unconditionally for Each Event

By default, if an alarm exists on a model for a given event, DX NetOps Spectrum does not generate a new alarm each time the same event occurs. However, the U flag can be used to change the default behavior and generate a new alarm unconditionally each time the same event occurs. The syntax is as follows:

```
<eventcode> E <eventseverity> A <alarmseverity>,<alarmcause>,U
```

This following event map serves as an example:

```
0x3dc0000 E 50 A 1,0x3dc0000,U
```

Generating Alarms That Users Cannot Clear

By default, users can clear alarms. However, the N flag can be used to change the default behavior so that users cannot clear an alarm. The syntax is as follows:

```
<eventcode> E <eventseverity> A <alarmseverity>,<alarmcause>,N
```

The following event map serves as an example:

```
0x3dc0000 E 50 A 1,0x3dc0000,N
```

Generating Alarms That Are Not Persistent

By default, alarms are persistent, that is, they are retained in memory if the SpectroSERVER is shut down and restarted. However, the T flag can be used to change the default behavior so that an alarm is not persistent. The syntax is as follows:

```
<eventcode> E <eventseverity> A <alarmseverity>,<alarmcause>,T
```

The following event map serves as an example:

```
0x3dc0000 E  50  A  1,0x3dc0000,T
```

Combining the U, N, and T Flags

Any combination of the U, N, and T flags can be used within an event map.

As an example, the following event map generates a unique alarm for each event that has an event code of 0x3dc0000. The alarm cannot be cleared by users, and it is not persistent.

```
0x3dc0000 E 50 A 1,0x3dc0000,U,N,T
```

The N flag and the T flag can be used with event discriminators as shown in the following example, which is a variation of the event map described in Generating Alarms for Events Based on the Values of Event Variables.

```
0x3b10011 E 70 A 1,0x3b10011,1,3,N,T
```

Specify an Event Frequency

The F flag can be used to detect if an event is occurring with abnormal frequency within a specified period of time. Use event rate rules to generate events for this purpose. The syntax described in this section is supported only to preserve compatibility with EventDisp files created for versions prior to version 6.5.

The syntax for using the F flag in an event map is as follows:

```
<EventCode> E <EventSeverity> F <OccurenceLimit> <Duration> <FrequencyEventCode>
```

- **F**
  Indicates how frequently the event is evaluated.
- **<OccurenceLimit>**
  Specifies the number of times that the event must occur.
- **<Duration>**
  Specifies the amount of time in seconds in which the number of events specified in <OccurenceLimit> must occur to generate an event.
- **<FrequencyEventCode>**
  Specifies the event code of the event to generate if the specified number of events occurs within the specified period of time.

Specify an Event Duration

You can use the D flag to detect that a second event in what was expected to be a pair of events did not occur within a specified period of time.

> **NOTE**
> Use event pair rules to generate events for this purpose. The syntax described in this section is supported only to preserve compatibility with EventDisp files created for versions prior to version 6.5.

The syntax for using the D flag in an event map is as follows:

```
<FirstEventCode> E <EventSeverity> D <SecondEventCode> <Duration> <DurationEventCode>
```

- **D**
  Indicates that the duration of the time between events is evaluated.
- **<SecondEventCode>**
  Specifies the event code of the second event that should occur.
- **<Duration>**
  Specifies the amount of time in seconds to wait for the second event to occur.
- **<DurationEventCode>**
  Specifies the event code of the event to generate if the second event does not occur within the specified period of time.

Clearing Alarms

You can specify that an event can clear any of the following types of alarms:

- An alarm with a specific alarm cause code that was created without event discriminators.
- All alarms with a specific alarm cause code that were created based on event discriminators if their variable values *match* the variables in the alarm-clearing event.
- All alarms with a specific alarm cause code that were created based on event discriminators *regardless* of whether their variable values match those in the alarm-clearing event.

Clear Alarms Created Without Event Discriminators

The following syntax is used to clear an alarm that was not created with event discriminators (which means there is only one instance of the alarm, and it does not contain copies of the event variables):

```
<eventcode> E <eventseverity> C <alarmtobecleared>
```

- **<eventcode>**
  Specifies the event code of the event.
- **E**
  Indicates the Archive Manager logs the event data to the Distributed Data Manager (DDM) database.
- **<eventseverity>**
  Specifies the relative severity of the event on a scale of 0 to 100, where 0 is the least severe and 100 is the most severe.
- **C**
  Indicates that an alarm be cleared when the specified event occurs.
- **<alarmtobecleared>**
  Specifies the alarm cause code of the alarm to clear.

Example:

```
0xfffff0000 A 1, 0xffff0000
0xfffff0001 C 0xffff0000
```

The first event map generates alarm 0xffff0000 when event 0xfffff0000 occurs. The second event map clears alarm 0xffff0000 when event 0xfffff0001 occurs.

Clear Alarms Based on Event Discriminator Values

In the context of clearing alarms, event discriminators are the IDs of the event variables to examine in the alarm-clearing event and an alarm to determine whether to clear the alarm as a result of the event.

Recall that the event variable IDs are mapped (in the AlertMap file) to the OIDs of the variable bindings that are sent with the trap, and their values are copied from the alarm-generating event to the alarm if the alarm is generated as a result of examination of their values.

Because the values of event variables are also stored in the alarms that are generated based on events, you can use event discriminators to differentiate multiple occurrences of an alarm and clear alarms that are based on the values of the variables. More specifically, the alarm can be cleared if the values in the alarm-clearing event *match* the values that are stored in the alarm.

The following syntax is used to clear an alarm that is based on event discriminator values:

```
<eventcode> E <eventseverity> C <alarmtobecleared>,<eventdiscriminators>
```

- **<eventcode>**
  Specifies the event code of the event.
- **E**
  Indicates that the Archive Manager logs the event data to the Distributed Data Manager (DDM) database.
- **<eventseverity>**
  Specifies the relative severity of the event on a scale of 0 to 100, where 0 is the least severe and 100 is the most severe.
- **C**
  Indicates that an alarm is cleared when the specified event occurs.
- **<alarmtobecleared>**
  Specifies the alarm cause code of the alarm to clear.
- **<eventdiscriminators>**
  Specifies a comma-separated list of one or more event variable IDs that indicate the variables to examine to determine whether to clear the alarm.

Example:

```
0xfffff0000 A 1, 0xffff0000, 1, 2
0xfffff0001 C 0xffff0000, 1, 2
```

Alarm 0xffff0000 is cleared only if the values of event variables 1 and 2 in the alarm-clearing event (0xffff0001) have the same values as those stored in the alarm, which are copied from the alarm-generating event (0xffff0000) when the alarm is generated. If the alarm-clearing event does *not* contain these event variables, the alarm is not cleared.

You can also use the following syntax to specify that a single event clears multiple types of alarms:

```
<eventcode> C <alarmtobecleared>,<eventdiscriminators> C <alarmtobecleared>,<eventdiscriminators>,...
```

Examples of Event Maps That Clear Alarms

In the following example, the first event map specifies that event 0x3dc0004 generates alarm 0x3dc0001. The second event map specifies that event 0x3dc0002 clears alarm 0x3dc0001.

```
0x3dc0004  E  50  A  2,0x3dc0001
0x3dc0002  C  0x3dc0001
```

The following two event maps are the same as those above except both use event discriminators. The first event map specifies that a new alarm is generated for each 0x3dc0004 event that has unique values for *both* event variables 1 and 3. The second event map clears all alarms with the alarm cause code of 0x3dc0001 if the values in the alarm for variables 1 and 3 *match* the values for the same variables in event 0x3dc0002.

```
0x3dc0004  E  50  A  1,0x3dc0001,1,3
0x3dc0002  C  0x3dc0001,1,3
```

NOTE
If the event map that creates the alarm uses event discriminators, but the event map that clears the alarm does not, the event discriminators are still considered when clearing the alarm.

The following event maps use the same logic as the example above even though the event discriminators are not explicitly stated in the event map that clears the alarm. Collectively, the event maps specify that all alarms with an alarm cause code of 0x3dc0001 should be cleared if the values in the alarm for variables 1 and 3 *match* the values for the same variables in event 0x3dc0002.

```
0x3dc0004  E  50  A  1,0x3dc0001,1,3
0x3dc0002  C  0x3dc0001
```

The following event map uses the U flag to generate a unique alarm each time that event 0x3dc0004 occurs. The event that clears the alarm, event 0x3dc0002, clears all alarms that have an alarm cause code of 0x3dc0001.

```
0x3dc0004  E  50  A  1,0x3dc0001,U
0x3dc0002  C  0x3dc0001
```

In the following example, the first two event maps generate alarms. The third event map uses event 0x3dc0009 to clear alarms with different alarm cause codes. All alarms with an alarm cause code of 0x3dc00010 are cleared, and all alarms with an alarm cause code of 0x3dc00011 are cleared if their values for variables 1 and 3 match the values for the same variables in event 0x3dc0009.

```
0x3dc0006 E 50 A 2,0x3dc00010
0x3dc0007 E 50 A 1,0x3dc00011,1,3
0x3dc0009 C 0x3dc00010 C 0x3dc00011,1,3
```

Clearing Alarms Regardless of Event Discriminator Values

Use the following syntax to clear all instances of an alarm that were generated based on event discriminators regardless of whether the values in the alarm-clearing event match the values that are stored in the alarm instances:

```
<eventcode> E <eventseverity> C <alarmtobecleared>,A
```

- **<eventcode>**
  Specifies the event code of the event.
- **E**
  Indicates that the Archive Manager logs the event data to the Distributed Data Manager (DDM) database.
- **<eventseverity>**
  Specifies the relative severity of the event on a scale of 0 to 100, where 0 is the least severe and 100 is the most severe.
- **C**
  Indicates that an alarm be cleared when the specified event occurs.
- **<alarmtobecleared>**
  Specifies the alarm cause code of the alarm to clear.
- **A**
  Specifies that all alarm instances be cleared regardless of the values of the variables that are stored in the events.

Example:

```
0xffff0000 E 50 A 1,0xffff0000,1,2,3
0xffff0002 E 50 C 0xffff0000,A
```

In this example, all instances of alarm 0xffff0000 are cleared when event 0xffff0002 occurs regardless of whether event 0xffff0002 contains event variables 1, 2, and 3, and, even if it does, regardless of whether their values match those stored in the alarms.

About Defining Event Rules

Event rules let you specify a complex decision-making system to indicate how an event is processed. An event rule looks for a series of events to occur on a model in a certain pattern or time frame. If the events occur as the rule specifies, another event is generated for the given model, and that new event must be defined in the EventDisp file so that DX NetOps Spectrum processes them appropriately.

> **WARNING**
> While you can create event rules manually, we recommend that you use the Event Configuration application to do so. This section is provided merely as a reference of the underlying syntax that is used to define rules in EventDisp files.

Event Rule Syntax

Event rules are implemented via event maps using the following syntax:

```
<EventCode> E <EventSeverity> R {<event_discriminators>} <event_rule_name>,
 <event_rule_parameter1>, ...<event_rule_parameterN>
```

- **R**
  Indicates that an event rule is used.
- **{<event_discriminators>}**
  (Optional) Comma-separated list of one or more event variable IDs that indicate the variables to examine to determine whether to generate the rule output event. These are the event variable IDs that are mapped (in the AlertMap file) to the OIDs of the variable bindings sent with the trap.

  > **NOTE**
  > Event discriminators apply to each contributing event in the event rule. For examples of using event discriminators with event rules, see EventRateCounter Rule and EventCombo Rule.
- **<event_rule_name>**

Is expressed using the following syntax:

CA.<RuleName>

<RuleName> specifies one of the following types of rules:

- EventPair
- EventRateWindow
- EventRateCounter
- EventSequence
- EventCombo
- EventCondition
- EventCounter
- Heartbeat
- SingleEvent
- SoloEvent

<eventruleparameter>

Varies depending on which type of rule is being used.

EventPair Rule

The syntax of an event pair rule is as follows:

```
<FirstEventCode> R CA.EventPair, <SecondEventCode>, <GeneratedEventCode>, <time>
```

For example, the following event pair rule generates event 0x0001002f when event 0x0001002a occurs but is not followed by event 0x0001002b within 60 seconds:

```
0x0001002a  R CA.Eventpair, 0x0001002b, 0x0001002f, 60
```

To also log event 0x0001002a and assign it a severity level of 50, the following syntax is used:

```
0x0001002a E 50 R CA.Eventpair, 0x0001002b, 0x0001002f, 60
```

EventPairTimeAttr Rule

The EventPairTimeAttr Rule is similar to EventPair Rule. The difference is that it the Attribute ID is the third parameter instead of the Time window.

The syntax of an event pair rule is as follows:

```
<FirstEventCode> R CA.EventPairTimeAttr, <SecondEventCode>, <GeneratedEventCode>, <Attribute ID>
```

- **Attribute ID**
  Specifies an attribute on the current model (the one where the event was generated one), which holds the time value for the pair rule. It can be used to set individual, model-specific time values for the rule instead of globally defined hardcoded ones.

EventRateWindow Rule

> **NOTE**
> An EventRateWindow rule is an event rate rule that uses a sliding window of time.

The syntax of an EventRateWindow rule is as follows:

```
<TriggerEventCode> R CA.EventRateWindow, <NumberofOccurrences>, <time>, <GeneratedHighRateEventCode>,
 <GeneratedLowRateEventCode>
```

where <GeneratedHighRateEventCode> is the event to generate if the trigger event occurs at the specified frequency in the specified time frame. If you also specify an event for <GeneratedLowRateEventCode>, which is an *optional* parameter, if the frequency drops below the threshold, the rule generates the low rate event. The rule generates the high rate event again only when the frequency threshold is crossed again.

As an example, the following EventRateWindow rule generates event 0x0001002f if five events of type 0x0001002a occur within 60 seconds:

```
0x0001002a R CA.EventRateWindow, 5, 60, 0x0001002f, 0x000100030
```

Once the frequency threshold is crossed and the high rate event (0x001002f) is generated, another rule output event is not generated until the frequency drops below 5 events within 60 seconds. At this point, the low rate event (0x000100030) is generated. The high rate event is not generated again until the frequency threshold is crossed again.

EventRateWindowAttrParams Rule

> **NOTE**
> An EventRateWindowAttrParams rule is an event rate rule that uses a sliding window of time.

The syntax of an EventRateWindowAttrParams rule is as follows:

```
<TriggerEventCode> R CA.EventRateWindow, <Attribute containing NumberofOccurrences>, <Attribute containing
 time window>, <GeneratedHighRateEventCode>, <GeneratedLowRateEventCode>, <GeneratedStopEventCode>
```

where <GeneratedStopEventCode> is the event that is generated to stop the rule. Update the attributes for the model and then generate this stop event to stop the rule instance on that model. The next rate window event starts a new rule instance, which reads and uses the new attribute values.

The EventRateWindowAttrParams rule is similar to the EventRateWindow rule. However, this rule accepts attribute ids instead of direct values for the event occurrence parameter and the time window parameter. These attributes must be present on the model where the event is generated and should contain integer values.

The EventRateWindowAttrParams rule is most useful for detecting an event that is not significant if it happens occasionally, but is significant if it happens frequently. If the event occurs above a certain rate, this rule generates another event. No additional events are generated as long as the rate stays at or above the threshold. But if the rate drops below the threshold and then subsequently exceeds the threshold, another event is generated.

An event can also be generated when the rate drops below the threshold.

> **NOTE**
> See EventRateWindow Rule for descriptions of the <GeneratedHighRateEventCode> and <GeneratedLowRateEventCode> events.

EventRateCounter Rule

> **NOTE**
> An EventRateCounter rule is an event rate rule that uses a sequential window of time.

The syntax of an EventRateCounter rule is as follows:

```
<TriggerEventCode> R CA.EventRateCounter, <NumberofOccurrences>, <time>, <GeneratedEventCode>
```

As an example, the following EventRateCounter rule generates event 0x0001002f if 5 events of type 0x0001002a occur within 60 seconds:

```
0x0001002a R CA.EventRateCounter, 5, 60, 0x0001002f
```

To also log event 0x0001002a and assign it a severity level of 50, the following syntax is used:

```
0x0001002a E 50 R CA.EventRateCounter, 5, 60, 0x0001002f
```

Example: Using Event Discriminators with an EventRateCounter Rule

The following example shows two event discriminators used in an EventRateCounter rule:

```
0x10001 E 50 R {1,2} CA.EventRateCounter, 3, 60, 0xffff0000
```

The event discriminator list, {1,2}, contains variable IDs 1 and 2. Therefore, in order for event 0xffff000 to be generated, event 0x10001 must occur 3 times within 60 seconds, and all 3 instances must contain the same values for variable IDs 1 and 2.

For example, if event 0x10001 occurred 3 times in 60 seconds, and each time variable ID 1 had a value of 10.253.40.57 and variable ID 2 had a value of 65, then event 0xffff0000 would be generated. However, if event 0x10001 occurred 3 times in 60 seconds but the first 2 times variable ID 1 had a value of 10.253.30.57 and the third time it had a value of 10.253.89.60, then 0xffff0000 would not be generated.

EventSequence Rule

An EventSequence rule is an event series rule that requires the series of events to occur in a specific sequence.

The syntax for the EventSequence rule is as follows:

```
<FirstEventCode> R CA.EventSequence, <GeneratedEventCode>, <time>, <SecondEventCode>,
 <ThirdEventCode>,...<NthEventCode>
```

As an example, the following EventSequence rule generates event 0x0001002f when events 0x00010002a, 0x0001002b and 0x0001002c occur, in that order, within 60 seconds.

```
0x0001002a R CA.EventSequence, 0x0001002f, 60, 0x0001002b, 0x0001002c
```

To also log event 0x0001002a and assign it a severity level of 50, the following syntax is used:

```
0x0001002a E 50 R CA.EventSequence, 0x0001002f, 60, 0x0001002b, 0x0001002c
```

EventCombo Rule

NOTE
An EventCombo rule is an event series rule that requires the series of events to occur but the order of occurrence does not matter.

The syntax of an EventCombo rule is as follows:

```
<FirstEventCode> R CA.EventCombo, <GeneratedEventCode>, <time>, <EventCodeA>, <EventCodeB>, ...<EventCodeN>
```

As an example, the following EventCombo rule generates event 0x0001002f if event 0x0001002a occurs, and it is followed by at least one instance each of event 0x0001002b and event 0x0001002c within 60 seconds and in any order.

```
0x0001002a R CA.EventCombo, 0x0001002f, 60, 0x0001002b, 0x0001002c
```

To specify that the new event should be generated but one of several events can first trigger the rule, create a series of n rules where n is the number of events in the combination, and where each rule uses a different event for the trigger. For example, consider the following three rules:

```
0x0001002a R CA.EventCombo, 0x0001002f, 60, 0x0001002b, 0x0001002c
0x0001002b R CA.EventCombo, 0x0001002f, 60, 0x0001002a, 0x0001002c
0x0001002c R CA.EventCombo, 0x0001002f, 60, 0x0001002a, 0x0001002b
```

Using this set of rules, any combination of events 0x0001002a, 0x0001002b, and 0x0001002c occurring at least once within 60 seconds and in any order would generate event 0x0001002f.

To also log event 0x0001002a and assign it a severity level of 50, the following syntax can be used:

```
0x0001002a E 50 R CA.EventCombo, 0x0001002f, 60, 0x0001002b, 0x0001002c
```

Example: Using Event Discriminators with the EventCombo Rule

The following example shows two event discriminators used in an EventCombo rule:

```
0x10001 E 50 R {1} CA.EventCombo, 0xffff0000, 60, 0x10002
```

The event discriminator list, {1}, contains variable ID 1. Therefore, event 0x10001 must occur, and then event 0x10002 must occur within 60 seconds, and each must contain the same values for variable ID 1 in order for event 0xffff000 to be generated.

For example, if event 0x10001 occurred and had 10.253.40.57 as a value for variable ID 1, and event 0x10002 occurred 45 seconds later and had a value of 10.253.40.57 for variable ID 1, then event 0xffff0000 would be generated. However, if event 0x10001 occurred and had 10.253.40.50 as a value for variable ID 1, and event 0x10002 occurred 45 seconds later and had a value of 10.253.40.57 for variable ID 1, then event 0xffff0000 would not be generated.

EventComboInclusive Rule

The EventComboInclusive rule is similar to the EventCombo rule. The difference is that it registers all the combo events, and not just the one which the rule is defined for.

The syntax of an EventComboInclusive rule is as follows:

```
<FirstEventCode> R CA.EventComboInclusive, <GeneratedEventCode>, <time>, <EventCodeA>,
 <EventCodeB>, ...<EventCodeN>
```

As an example, the following EventComboInclusive rule generates event 0x0001002f if either of event 0x0001002a, 00x0001002b or event 0x0001002c occurs:

```
0x0001002a R CA.EventComboInclusive, 0x0001002f, 60, 0x0001002b, 0x0001002c
```

EventCondition Rule

The conditional expressions in an EventCondition rule can compare a variable binding value or a DX NetOps Spectrum attribute value to a user-specified value using standard comparison operators as follows:

```
<FirstEventCode> R CA.EventCondition, "<conditional expression 1>", <event to generate when 1 is
 TRUE>,"<conditional expression n>", <event to generate when n is TRUE>, "default", <default event>
```

- **“<conditional expression x>”**
  Consists of one or more expressions comparing a variable binding value or a DX NetOps Spectrum attribute value to a user-defined value (x represents any value from 1 to n).
- **<event to generate when x is TRUE>**
  Specifies the event that is generated if the conditional expression evaluates to TRUE.
- **“default”, <default event>**
  (Optional) Specifies a default event that is generated if none of the conditions are met. For example, if the following syntax was included at the end of the rule, event 0xffff1234 would be generated if none of the other conditions expressed in the rule were met:
  ```
  "default", 0xffff1234
  ```

  NOTE
  Default can be expressed as “DEFAULT”, “default”, or “Default”.

Conditional expressions are evaluated from left to right and, with some simplification, follow C programming-style evaluations. If the whole condition evaluates to TRUE, then the event is generated.

Condition Syntax

The conditional text is always enclosed in quotation marks as follows:

“condition”

A simple condition is made up of data or objects and comparison operators or methods.

Data or Objects

Each data element or object to be compared in the condition must be contained within curly brackets and must have both a type and a value:

```
{ TYPE VALUE }
```

For example, a comparison that involves an integer value of 2 is expressed as { I 2 }.

The following lists the supported types and their meanings:

Short Symbol	Alternate Names	Meaning	Type Values	Examples
A	Addr ADDR address IP Addr IP Address IP_ADDRESS	Contains an IP address	XXX.XXX.XXX.XXX where each XXX subterm is a number from 0 to 255, and the whole term forms a valid IP address	{ Addr 192.168.1.1 }

a	attr ATTR attribute ATTRIBUTE	References a model attribute of the model for which the event rule is being processed. When evaluated, the attribute's current value is read. The attribute's type is used to determine if comparison is valid.	An attribute ID, specified as a hex number. The leading 0x is optional. The letters a-f may be lower- or uppercase. The reading of table attributes using an object ID or variable data as an index is also supported (see examples).	{ attr 0x11564 } { Attribute A00044 } { attr 0xffff0001 obj 1.1.6.8.0.1 } would read table attribute 0xffff0001, with "1.1.6.8.0.1" as the OID suffix (index). { attr 0xffff0001 VARDATA 2 } would read table attribute 0xffff0001 using the object id contained in the second variable binding as the OID suffix (index).
B	BOOL Bool Boolean boolean	A boolean value	False, true This value is not case-sensitive	{ B True } { boolean false }
H	HEX Hex Hex_ID HEX Id	A hex attribute ID value (this is just a value, not an attribute reference, use 'a' for that purpose)	An attribute ID, specified as hex number. The leading 0x is optional. The letters a-f may be lowercase or uppercase.	{ Hexid 0xffff0123 } { H ABCD } { HEX 0X91 }
I	Int integer INTEGER INT	An integer value	Any number within the range of -214783648 to 214783647	{ Int 10 } { I 98765 } { integer -300 }
L	Unsigned long int LONG long UNSIGNED_LONG_INTEGER LongInt	An unsigned long 64 bit integer	Any number within the range of 0 to 18446744073709551615	{ L 0 } { LongInt 123456789098}
o	Object ID obj obj_id OBJECT	An object ID	X X.X X.X.X and so on where X is an unsigned integer (>= 0)	{ o 1.2.3.4.5.6 } { object_id 1.3.6.1.2.1 } { OBJ 100000.4.5 }
O	Octet Octet String Decimal Octet String DEC Oct_Str Dec_OCTET_str	A tagged octet string comprised of decimal values	#.#.#.... where # is any number between 0 and 255	{ Oct Str 1.2.3.4 } { OCTET_STR 255.0.1.20 } { OCTSTR 10.20.30.40.50 }
R	REAL real Real	A real number (double)	Any number containing a '.' , and possibly followed by an exponent: E|e +|- EXP EXP any number	{ R 1. } { Real 3.1415 } { REAL -2.843E-17 } { R .00001e+20 }

S	String str Str STRING	A string	Any characters enclosed in double quotes	{ S \" a string \" } { String \"another string\" } { str \"12345a@b?c*\" } { S \"\" }
U	unsigned unsigned long unsigned long int ULONG U INT U_long_integer	An unsigned long integer	Any number in the range from 0 to 4294967295	{ uint 1234567890 } { UNSIGNED INT 0 }
v	variable data VARDATA Var_DATA EventAttr event_attr	References an event attribute (variable data) from the current event. Like a model attribute reference, this is evaluated when needed, and the event attribute type is used to check if the comparison is valid	A unsigned integer (> 0)	{v 1 } { VARDATA 2 } { event attr 3 }
X	Hex octet string HEXOCT_STR Hex_Octet HEXOctet	A tagged hexadecimal octet string	XX.XX.XX..... Where XX is a hex number from 0 to FF	{ X 12.01.AB.EF } { HEXOCT AB.CD.EF.01 } { Hex octet string 2.3 } { X a.b.c }

Using the Escape Character

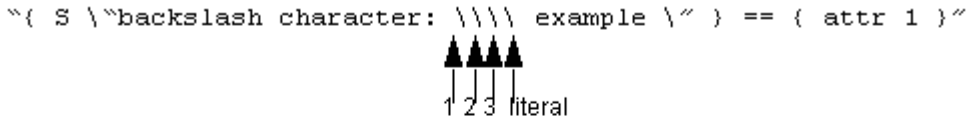
In addition to enclosing the entire condition within quotation marks, you must also enclose a string in quotations marks. Also, to help ensure that the entire condition is parsed properly, the opening and closing quotation marks enclosing the string both must be preceded by a backslash. The following condition, which includes string xyz, serves as an example:

```
"{ S \"xyz\" } == { v 1 }"
```

If you want to include a backslash '\' or a double quote in the string itself, you need 3 backslashes before each of these characters, as shown in the following example where:

- Backslash 3 represents the escape for the literal backslash or quote
- Backslash 2 represents the escape character needed to escape backslash 3 within the string
- Backslash 1 represents the escape character needed to escape backslash 2

This same logic holds true for a double quote example.



Regular expressions also use the backslash ‘ \’ as an escape character. Because the regular expression can be used in a condition, the backslash used within the regular expression as the escape character must be preceded by several backslashes.

Comparing Strings

The strcmp method is used to compare characters in a string (for example, in strings, octet strings, IP Addresses, and so on). Note that the implementation of this method differs slightly from the standard C implementation of strcmp. This method returns TRUE if the two strings are equal, and it returns FALSE if the strings are not equal.

To make use of this method, use the following format:

```
"strcmp({ TYPE <string> }, { TYPE <string> })"
```

For example, the following condition compares an IP Address with a DX NetOps Spectrum attribute whose value is an IP Address. If the two strings are the same, the condition returns TRUE.

```
"strcmp({ A 179.82.253.01 }, { attr 0x00011aec })"
```

Note the following in the example:

- Each type/string pair to be compared is enclosed in curly brackets
- The two type/string pairs are separated by a comma and enclosed in parenthesis
- The name of the method, strcmp, is placed on the left hand side of the parenthesis
- The entire condition is enclosed in quotes

Regular Expressions

The regexp or REGEXP method is used to compare a string to an input pattern. To do this, regular expressions use a series of meta-characters that let you express the pattern of characters that you are looking for. The method returns TRUE if the regular expression input pattern matches the input string; otherwise, it returns FALSE.

To make use of this method, use the following format:

```
"regexp({ TYPE <string> }, { TYPE <input pattern> })"
```

The EventCondition rule supports the syntax of the Perl Compatible Regular Expression (PCRE) package. The following are some of the basic meta-characters supported by the PCRE package and examples of their usage:

Meta-character	Meaning	Example
^	Indicates the beginning of a line.	The following example searches for the string “CA” occurring at the beginning of a line in event variable 1: “regexp({ VARDATA 1 }, { S \“^CA\” })”
$	Indicates the end of a line.	The following example searches for lines ending with the string “CA” in event variable 1: “regexp({ VARDATA 1 }, { S \“CA$\” })”
[]	Encloses a character class. A character class shows some literal text that you would like to let at a certain point within the string.	This example searches for the string “port” followed by a value of 1, 2 or 3 in event variable 1: “regexp({ VARDATA 1 }, { S \“port [1-3] \” })”
*	Indicates zero or more of the specified preceding characters.	This example searches for zero or more occurrences of 172 in the value of the attribute 0x00011aec: “regexp({ attr 0x00011aec }, { S \“(172)*\” })”

+	Indicates one or more of the specified preceding characters.	This example searches for one or more occurrences of 172 in the value of the attribute 0x00011aec: “regexp({ attr 0x00011aec }, { S \“(172)+ \” })”
.	Represents any single character except for a new line character.	This example searches for an occurrence of the word “port” followed by a space and then any single character within event variable 1: “regexp({ VARDATA 1 }, { S \“port . \” })”
|	Separates alternative patterns.	“regexp({ VARDATA 1 }, { S\“interface|port \” })” This example searches for either the word interface or the word port within the event variable 1.
\	Used as a general escape character letting you to use the literal meaning of a meta-character. When you use the escape character within the regular expression, you must be sure to also consider the necessary escape characters to be used within the context of the string and the condition..	This example searches for the string “172.55” within event variable 1: “regexp({ VARDATA 1 }, { S\“172\\\\.55\” })” Because the . character is usually treated as a meta-character within a regular expression, it is necessary to use the escape character (a backslash) to indicate that you would like the . to be treated literally. At the regular expression level, this yields the following syntax: 172\.55. However, since you are using this regular expression within a string, you must precede the backslash with an additional backslash. Reading the expression from left to right, the first backslash represents the escape character needed to escape the second backslash within the string. This yields the following syntax: 172\\.55. Additionally, you are using the string within the context of the condition, therefore each existing \ must have a corresponding backslash to be used as an escape. This yields the following syntax: 172\\\\.55.

Comparison Operators

The following comparison operators are supported for numeric or boolean values:

- **==**
  equal to
- **!=**
  not equal to
- **>**
  greater than
- **<**
  less than
- **<=**
  less than or equal to
- **>=**

greater than or equal to

Most numeric values can be compared with each other even if they are not of the same type. For example, the following condition compares the integer 2 to the value of a DX NetOps Spectrum attribute:

```
"{ I 2 } = = { attr 0x000117dc }"
```

Exists Operator

You can use the exists operator to check for the existence of a variable binding value or other value:

```
exists( <expression> )
```

This can be useful, for example, when you want to evaluate a value if it exists, and exit the event condition rule if it does not. The following spelling variants are supported:

exists

Exists

EXISTS

An exists conditional expression returns TRUE when the expression is valid and contains a value. For example, “exists({ v 1 })” returns TRUE when event variable 1 exists and contains any value.

To terminate an event condition rule without action when an exists conditional expression returns FALSE, you can use the “no action” action (which sends the 0x00010000 null event). The following spelling variants are supported in either all lower case, all upper case, or initial capital letters:

no action

no-action

no_action

As an example, the following event condition rule checks whether event variable 1 exists in event 0xffff0000. If it does not exist, no action is taken. If it does exist, event 0xffff0001 is generated.

```
0xffff0000 E 50 R CA.EventCondition,           \
     " ! Exists( { v 1 } )", "No-Action",      \
     " { v 1 } == { I 1 } ", 0xffff0001
```

The Logical NOT (!)

You can use the logical NOT (!) operator to reverse the logical value of an expression. Apply the logical NOT (!) in the same way as you would when writing C++ code. For example, the following condition compares variable binding 1 and variable binding 2. The logical NOT (!) is applied to the outcome of the comparison. Thus, if variable binding 1 is equal to variable binding 2, the entire expression evaluates to false.

```
"! ({ v 1 } = = { v 2 })"
```

Complex Conditions

More complex conditions can be created which use logical operators and parenthesis to combine simple conditions. Valid logical operators are the following:

&&, which represents AND

||, which represents OR

The following condition includes several subconditions enclosed in braces and linked together using logical operators.

```
"({ I 2 } = = { I 2 }) &&  ({ I 2 } != { I 3 })"
```

For this condition to evaluate to TRUE, both of the subconditions on either side of the && must evaluate to TRUE. Since an integer value of 2 is equal to an integer value of 2, the left side of the condition is TRUE. Since an integer value of 2 is not equal to an integer value of 3, the right side of the condition also evaluates to TRUE. This means that the whole condition evaluates to TRUE.

```
"({ I 3 } = = { I 4 } ) || ({ I 4 } > { I 2 })"
```

For this condition to evaluate to TRUE, the subcondition on the left hand side of the || or the subcondition on the right hand side of the || must evaluate to TRUE. Since 3 is not equal to 4, but 4 is greater than 2, the entire expression evaluates to TRUE.

Multiple subconditions can be used to create the necessary expression. For example, the following condition evaluates to TRUE since 4 is greater than 2, and 3 is less than 8.

```
"({ I 3 } = = { I 2 } || { I 4 } > { I 2 }) && ({ I 3 } < { I 8 } || { I 2 } = = { I 4 })"
```

Nested Conditions

Although simple conditions can be combined together using logical operators to create complex conditions, you cannot use simple conditions as a part of other expressions. A simple condition can have only one comparison operator in it.

For example, the following syntax is not supported. The result of a strcmp() cannot be used as the argument for an equals (==) operator:

```
"strcmp ( { S \"a\" }, {S \"a" } ) = = { B TRUE }"
```

Example: Basic EventCondition Rule

The following example shows a basic EventCondition rule that uses some of the condition syntax described in the previous sections. The first condition that evaluates to TRUE is used, and the event code immediately following that condition is generated.

```
0x00045678 R CA.EventCondition,                                    \
    "regexp({ VARDATA 1 }, { S \"port [1-3] \" })",                        \
    0x00012345,"{ VARDATA 2 } == { attr0x000117dc }",                      \
    0x00012344, "strcmp({ A 179.82.253.01 }, \{ attr 0x00011aec }    \
            )",                     \
0x00122334
```

Example: (Complex EventCondition Rule) Generating an Event Based on a Variable Binding Value

The following example shows an EventCondition rule that generates an alarm based on the value of one of the variable bindings sent with the trap. The event condition rule generates a second event, 0xffff0000, if the value of variable binding 1 is equal to any of the following values: DAT0005, DAT0006, DAT0007, DAT0008, DAT0011, DAT0012, DAT0013, DAT0014, DAT0021, DAT0022, or DAT0023.

```
0x1030f E 50 R CA.EventCondition,                    \
  "regexp( { v 1 },                                      \
           { S \"DAT00(05|06|07|08|11|12|13|14|21|22|23) \" } \
         )",                                             \
  "0xffff0000 -:-"
```

> **NOTE**
> There is a space at the end of the regular expression, which indicates that each pattern must end with a space.

The regular expression first looks for a match between variable binding 1 and DAT00 combined with any of the choices in the brackets (05 , 06 , 07 , etc.).

There is a -:- at the end of the event condition. This symbol specifies that all of the variable binding values from the originating event (0x1030f) should be copied to the new event (0xffff0000).

To generate an alarm when the event condition evaluates to true, an event map for the new event (0xffff0000) that specifies to generate an alarm is needed in the EventDisp file.

EventCounter Rule

The EventCounter rule implements a counter, counting up for 'up' and down for 'down' events. The rule creates an event if a certain threshold is reached.

The syntax of EventCounter rule is as follows:

```
<CountUpEventCode> R CA.EventCounter, <CountDownEventCode>, <threshold>, <ThresholdBreachedEventCode>,
 <ThresholdResetEventCode>
```

- **CountUpEventCode**
  Counts up by one. The first one also initiates the counter rule.
- **CountDownEventCode**
  Counts down by one.
- **threshold**
  The counter threshold (integer number). Once the count reaches (equals) that threshold, the target event is generated
- **ThreaholdBreachedEventCode**
  Generates when the count threshold is reached
- **ThresholdResetEventCode**
  Generates when the count is below the threshold again

Example: EventCounter rule

```
0x0f420001 E 50 R CA.EventCounter, 0x0f420002, 3, 0x0f420003, 0x0f420004
```

Use events 0x0f420001 to count up, and events 0x0f420002 to count down. When the counter is greater or equal to 3, event 0x0f420003 will be generated. Once the count falls below 3 again, event 0x0f420004 will be generated.

Heartbeat Rule

The Heartbeat rule is set to watch a 'heartbeat' event. The event is seen at a regular interval. In case any instance of the heartbeat is found missing, the rule creates an event.

```
<TriggerEventCode> R CA.Heartbeat, <HeartbeatEventCode>, <MissingHeartbeatEventCode>, <timeout>,
 <StopEventCode>
```

- **TriggerEventCode**
  Instantiates the heartbeat event rule
- **HeartbeatEventCode**
  The heartbeat event code
- **MissingHeartbeatEventCode**
  The event generated in case a heartbeat is missing
- **timeout**
  The time gap between individual heartbeats (in seconds)
- **StopEventCode**
  (Optional) The event code that stops the rule. The rule instance is running forever, looking for a heartbeat until stopped

Example: Heartbeat Rule

```
0x0f440001 E 50 R CA.Heartbeat, 0x0f440002, 0x0f440003, 10, 0x0f440004
```

Event 0x0f440001 instantiates the heartbeat rule. Next, it will look for event 0x0f440002 to occur at least once every 10 seconds. If a heartbeat event is found missing, the rule creates event 0x0f440003. Use event 0x0f440004 to stop the heartbeat rule instance.

SoloEvent Rule

The SoloEvent rule finds an instance of the target event that is not followed by or preceded by any other event in a defined time window. The events that are in the prevent list may not occur. Other events will not affect the Solo event.

As an example, you may want to have a rule triggered when event A occurred, but only none of events B, C or D (the 'prevent' events) occurred within five minutes before or 10 minutes after it.

The syntax of the SoloEvent Rule is as follows:

```
<TriggerEventCode> R CA.SoloEvent, <StopEventCode>, <SoloEventCode>, <prePreventPeriod>, <postPreventPeriod>,
 <TargetEventCode>, <PreventCode 1>, <PreventCode 2 (optional), ... , <PreventCode N (optional)>
```

- **TriggerEventCode**
  Initiates the 'solo' event rule.
- **StopEventCode**
  Stops the rule. The rule will run endlessly, unless you set the stop event.
- **SoloEventCode**
  Sets the solo event.
- **prePreventPeriod**
  Sets the time period before the solo event where none of the 'prevent' events may occur (in seconds).
- **postPreventPeriod**
  Sets the time period after the solo event where none of the 'prevent' events may occur.
- **TargetEventCode**
  Defines the event that will generate when the rule triggers (when just the 'solo' event was seen).
- **PreventCode 1 to PreventCode N**
  Defines a list of 'prevent' events.

Example: SoloEvent Rule

The following example will create event code 0x0f400005 if the solo event 0x0f400003 was seen, and none of the events 0x0f400006, 0x0f400007, 0x0f400008 or 0x0f400009 were seen within 20 seconds before or after it. The first event 0x0f400001 will have to be generated to instantiate the rule. Event 0x0f400002 is available to stop the rule.

```
0x0f400001 E 40 R CA.SoloEvent, 0x0f400002, 0x0f400003, 20, 20, 0x0f400005, 0x0f400006, 0x0f400007,
 0x0f400008, 0x0f400009
```

SingleEvent Rule

The SingleEvent rule reduces an event stream where one event ('up' event) may occur multiple times, before a reset ('down') event is seen. Instead of the multiple 'up' events, a single event is set that can be reused in other rules, denoting the condition ('up' or 'down').

The syntax of SingleEvent rule is as follows:

```
<TriggerEventCode> R CA.SingleEvent, <ResetEventCode>, <SingleTargetEventCode>, <SingleResetTargetEventCode>
```

- **TriggerEventCode**
  Occurs multiple times and should be converted into a 'single' occurrence. This will also trigger the rule to be instantiated when it occurs the first time.
- **ResetEventCode**
  Sets the reset event. When this event is seen, the rule is ready to create another single event again.
- **SingleTargetEvent**
  Generates the first time the trigger event is seen either when the rule is instantiated, or the first time the trigger event occurs after reset event is seen.
- **SingleResetTargetEvent**
  (Optional) Generates when the reset event is seen.

Example: SingleEvent Rule

```
0x0f430001 E 50 R CA.SingleEvent, 0x0f430002, 0x0f430003, 0x0f430004
```

Generates event 0x0f40003 every time event 0x0f430001 is seen for the first time. The SingleEvent is instantiated either at initial rule creation, each time after the trigger is seen the first time or after the reset event 0x0f430002 was seen. Event 0x0f430004 will also be generated when the reset event has been seen.

Using Multiple Event Rules in a Single EventDisp Entry

You can specify that a single event be processed using multiple event rules. For example, the following event disposition entry specifies that event 0x0001002a be processed using both the EventSequence rule and the EventPair rule:

```
0x0001002a R CA.EventSequence, 0x0001002c, 60, 0x0001002d,0x0001002e \
    R CA.EventPair, 0x0001002b, 0x0001002f, 60
```

This entry specifies that event 0x0001002c will be generated when events 0x0001002a, 0x0001002d, and 0x0001002e occur in that order within 60 seconds, and event 0x0001002f will be generated if event 0x0001002a occurs, but is not followed by event 0x0001002b within 60 seconds.

The backslash character is used at the end of the line to show that the event disposition entry continues onto the next line.

> **NOTE**
> Multiple rules must be specified within a single event disposition entry. If you were to create two separate event disposition entries for an event, only the first event disposition entry would be processed.

Copy Event Variables from One Event to Another

You can trigger events using event rules. However, sometimes the event is generated only after multiple contributing events occur or certain complex conditions are met.

By default, a rule output event does not have any event variables. You can however specify that, the values of the event variables in the events that contribute to the processing of the rule, be copied to the rule output event. Copying the event variables lets you specify event processing behaviors for the rule output event based those values, which you can do using event variable discriminators. Moreover, because the values of event variables in events that generate alarms are also stored in those alarms, you can also use event discriminators to specify alarm processing (generation or clearing of alarms) based on the values of the copied variables.

This section provides reference information about event variable copying syntax.

Event Variable Copying Syntax

Consider the following EventCombo rule which generates event 0xa000f, when events 0x10002, 0x10003, and 0x10004 are all received within 10 seconds of event 0x10001:

```
0x10001 R CA.EventCombo, 0xa000f, 10, 0x10002, 0x10003, 0x10004
```

Now assume that the contributing events have variable bindings that have values, some of which you want to copy to event 0xa000f as follows:

Contributing Event	Event Variables to Copy	Event Variables in 0xa000f to Receive the Copies
0x10001	2, 3	1, 2
0x10002	1, 5	3, 4
0x10003	none	none
0x10004	2, 3, 4, and 5	5, 6, 7, and 8

NOTE
An event generated by an event rule does not have any event variables unless they are copied from contributing events.

To copy the event variables as specified in the preceding table, the following event variable copying syntax is used:

```
0x10001 R CA.EventCombo, "0xa000f 2-3:1-2",    \
            10,                  \
            "0x10002 1:3, 5:4",   \
            0x10003,             \
            "0x10004 2-5:5-8"
```

This syntax copies event variables 2 and 3 in 0x10001 to event variables in 0xa000f, copies event variables 1 and 5 in 0x10002 to event variables 3 and 4 in 0xa000f, and copies event variables 2, 3, 4, and 5 in 0x10004, respectively, to event variables 5, 6, 7, and 8 in 0xa000f.

Event Parameters and Variable IDs

Event parameters specify the event variables to copy from a contributing event to the event generated by the rule. This information is associated with the contributing event, and it is entered after the event ID. To specify that the event copy information is part of the event parameter and not the next rule parameter, the event ID and the copy information need to be enclosed in double quotes, as shown:

```
"0x10002  1:3"
```

Event Variable Copy Information

The event variable copy information section of an event parameter can consist of several parts, each separated by a comma. Each part specifies a source variable ID or range of IDs. followed by a target variable ID or range of IDs. A colon separates the source IDs from the target IDs, as shown:

```
"0x10001  1:1, 2:3"
```

NOTE
If the source IDs and the target IDs are the same, either can be left empty.

Variable IDs

A single event variable is identified by its ID, which typically is a small number.

To copy a specific source event variable, enter its ID in the source position (left of the colon) in the copy information section. Similarly, to copy to a specific event variable, enter its ID in the target position (right of the colon).

For example, the following syntax copies variable 1 in event 0x10001 to variable 1 in the rule-generated event:

```
"0x10001  1:1"
```

Similarly, the following syntax copies variables 1 through 5 in event 0x10003 to variables with the same IDs in the rule-generated event:

```
"0x10003 1:1, 2:2, 3:3, 4:4, 5:5"
```

Variable IDs Using Ranges

Ranges of source and target variable IDs can be specified using a start ID followed by the dash ('-') character and then a stop ID. The start and stop IDs are included in the range.

The start ID, the stop ID, or both IDs can be left out. An entry with no start ID copies all of the IDs in the range from 1 to the stop ID, inclusive. An entry with no stop ID copies all of the variables with IDs equal to or greater than the start ID. An entry without either the start ID or the stop ID copies all of the variables (the dash can be left out too).

The number of variables in the source and target ranges need to be the same. If the numbers do not match, the number of IDs in the smaller range is used to copy variables, and a warning is generated.

As examples, the following 3 rule fragments are all equivalent. Each copies variables 1 through 73 in event 0x10004 to variables with the same IDs in the rule-generated event:

```
"0x10004  1 - 73 : 1 - 73"
"0x10004  1-73 :"
"0x10004 :1-73"
```

To copy variables 1 through 3 in event 0x10005 to the rule-generated event using new IDs 6 through 8:

```
"0x10005 1-3:6-8"
```

To copy all variables up to and including variable 5 to the rule-generated event using new IDs beginning with 1:

```
"0x10006 -5:1-"
```

To copy variables 2 through 5 to new IDs 1 through 4, and to also copy all remaining variables beginning with ID 7 using the same target IDs in the rule-generated event:

```
"0x10007 2-5:1-4, 7-:7-"
```

To copy all of the variables in event 0x10008 to the rule-generated event, use any of the following:

```
"0x10008 -:-"
"0x10008 -:"
"0x10008 :-"
"0x10008 :"
"0x10008 1-:1-"
```

The following example copies variables 3 to 5 from event 0x10009 to the rule-generated event where the source event has fewer variable entries then the target event. This example will succeed, but a warning will be generated:

```
"0x10009 3-5:3-"
```

Copying Variables from the Initial Event

Copying variables from the initial event (the event that the event disposition entry is for) is a special case. In this case, the copy information for the initial event is attached to the event to be generated rather than the source event.

As an example, consider the following EventCombo rule and EventRate rule, both of which watch for event 0x10001. If event 0x10001 occurs in combination with event 0x10002, its event variable 1 is important; if it occurs frequently, its event variable 3 is of interest:

```
0x10001        R CA.EventCombo, "0xa000f 1:1", 10, 0x10002 \
        R CA.EventRateCounter, 10, 3600, "0xa000e 3:1"
```

NOTE
Different types of rules specify the event to generate in different parameter positions. In the preceding example, the EventCombo rule uses the first variable position, and the EventRateCounter rule uses the third variable position.

Multiple Event Occurrence

When two event rules watch for multiple occurrences of an event (for example, an EventRateWindow rule and an EventRateCounter rule), the final occurrence of the event is used to copy the event variables, since they are the most current.

As an example, assume that event 0x10001 is a security alert from an intrusion detection system that holds information about the severity of a potential intrusion. Also assume that the following event rate rule exists, which generates event 0xa000f when event 0x10001 is received at least 5 times during a 100 second interval:

```
0x10001 R CA.EventRateCounter, 5, 100, "0xa000f  1:1"
```

A possible sequence of contributing events might be as follows:

NOTE
For readability, the severity event variable is shown as a text string.

0x10001 Warning

0x10001 High

0x10001 Warning

0x10001 Critical

0x10001 Critical

where the fifth 0x10001 event received triggers the generation of the rule output event, creating event 0xa000f with its variable 1 set to “Critical.”

As an alternative and more specific approach, the following EventCondition rule could be used so that the rule output event is only generated in response to five critical 0x10001 events (instead of in response to five 0x10001 events of any severity):

```
0x10001 R CA.EventCondition, \
    "regexp ( { variable 1 } , { S \"Critical\" } )", "0xa0001 1:1"
0xa0001 R CA.EventRateCounter, 5, 100, "0xa000f"
```

Event Variable Copy Example

Having reviewed Event Variable Copying Syntax for information about how to copy event variables from contributing events to rule-generated events, examine again the following event variable copy example:

```
0x10001 R CA.EventCombo, "0xa000f 2-3:1-2",    \
            10,                  \
            "0x10002 1:3, 5:4",  \
            0x10003,             \
            "0x10004 2-5:5-8"
```

When this EventCombo rule generates event 0xa000f, the event variables in the contributing events are copied as shown in the following table:

Source Event	SourceVariable ID	0xa000f (Target)Variable ID
0x10001	2	1
0x10001	3	2
0x10002	1	3
0x10002	5	4
0x10004	2	5
0x10004	3	6
0x10004	4	7
0x10004	5	8

Event Variable Copying and Event Discriminators

The event copying syntax can be used in conjunction with event discriminators to differentiate events that generate alarms. For example, assume alarm 0xffff0000 should be generated when event 0x10001 is received, and event variable 1 contains the index of the board that failed. Event variable 1 can be used as the alarm discriminator. Also assume that the device sometimes falsely reports errors, and the alarm should be cleared if supporting event 0x10002 is not received within 10 seconds. The following event maps satisfy these requirements:

```
0x10001 E 50 \
    A 2, 0xffff0000, 1 \
    R CA.EventPair, 0x10002, "0xa000f 1:1", 10
0xa000f C 0xffff0000, 1
```

The EventPair rule is used to watch for the second event and issue the clearing event containing the correct variable needed to clear the alarm. Note the use of the event discriminator to determine whether to generate and to clear the alarm, as well as the copy information contained in the rule.

Syntax Errors in EventDisp Files

All of an event's processing syntax must exist as a single event map. An EventDisp file that lists the same event code twice as the first entry on a line is considered to be improperly formatted. In this situation, the first line that applies to a particular event code is processed, and any additional entries are discarded.

The following example illustrates *incorrect* syntax:

```
0x3e00002  E  50
0x3e00002  A  2,0x3e00002
```

The following example illustrates *correct* syntax:

```
0x3e00003  E  50  A  2,0x3e00003
```

In the incorrect example, only the first event map is processed. Thus, when event 0x3e00002 is received, it is logged with an event severity of 50. However, no alarm is generated.

The correct example shows how to specify that DX NetOps Spectrum should log an event, assign an event severity to it, and generate an alarm as a result of the event.

Add Comments in EventDisp Files

You can add comments to an EventDisp file using the # identifier. Any text that follows this identifier on a line is ignored when the EventDisp file is processed.

You must enter the # identifier as the first character on the line. Do not enter a space and then #, as this produces an error when the EventDisp file is processed. If the comment must span multiple lines, enter the # identifier as the first character on each line.

The following example shows proper usage:

```
# This is a valid comment.
# This is a valid comment.
0x3dc0004  E  50  A  2,0x3dc0001
0x3dc0002  C  0x3dc0001
```

Event Format Files

About Event Format Files

An *event format file* contains the message about the event that is displayed to users on the Events tab in OneClick when the event occurs. The message can contain references to the event variables that hold data retrieved from the variable bindings of the trap. There exists an event format file for each event generated by DX NetOps Spectrum.

The events provided with DX NetOps Spectrum all have event format files that define appropriate event messages. In addition, whenever you create a new, custom event (either using MIB Tools when you are mapping a trap to the new event or later using Event Configuration) the associated event format file is automatically created. This means that typically you should not need to manually create an event format file.

> **WARNING**
> It is recommended that you create and modify all event messages using Event Configuration, which updates the appropriate event format files on (only) the OneClick web server to which you are connected when you save the changes to a landscape. However, this appendix provides reference information on event format files if manual modifications are ever required.

Location of Event Format Files

The event format files that support the events provided with DX NetOps Spectrum are installed in the following folder:

<$SPECROOT>/SG-Support/CsEvFormat

Custom event format files that are created by MIB Tools or Event Configuration are installed in the following folder:

<$SPECROOT>/custom/Events/CsEvFormat

Each event format file is named Event<event_code>, where <event_code> is the 4-byte, hexadecimal event code assigned to the event. For example, an event with an event code of 0x12345678 has an event format file named Event12345678.

Contents of an Event Format File

If you modify an event format file manually, note the following:

- As you compose the event message, keep in mind that most of the information that a OneClick user receives about an event is via the message text that is associated with the event. For this reason, provide as much information about the event as possible in the message.
- An event message can consist of plain text and variables that reference specifics about the instance of the individual event. For information on the correct syntax for including variables, see Variable Descriptions and Syntax.
- If there exists an event format file for an event, but no event map for the event exists in an event disposition file, the contents of the event format file are still displayed on the Events tab in OneClick when the event occurs.
- If no event format file exists for an event, a default message indicating this is displayed on the Events tab in OneClick.

About Event Table Files

An *event table file* does the following:

- Enumerates the possible values of a variable binding that is sent with a trap. The values can be attribute values in MIB tables, OID values, and integer bit values.
- Provides corresponding text values for the enumerated values.

When a trap is mapped to a DX NetOps Spectrum event, the variable bindings sent with the trap are mapped to event variables. This means that by referencing an event variable and the event table file for the appropriate variable binding you can define event messages that include the text values for the variable binding values. In turn, this means that the event message that is displayed to users in OneClick contains data that is specific to the trap that is sent.

You do not need to manually create event table files. All of the events provided with DX NetOps Spectrum that are mapped to traps that contain variable bindings with enumerated definitions in the MIB have supporting event table files. In addition, whenever you add trap support for a device that is not supported by default in DX NetOps Spectrum, and you map the traps to new, custom events using MIB Tools, an event table file is automatically created for each variable binding that meets this same criterion.

> **NOTE**
> This appendix provides reference information on the proper syntax for an event table file. However, typically you should not need to modify these files.

Location of Event Table Files

The event table files that support the events provided with DX NetOps Spectrum are installed in the following folder:

<$SPECROOT>/SG-Support/CsEvFormat/EventTables

Custom event table files created by MIB Tools are installed in the following folder:

<$SPECROOT>/custom/Events/CsEvFormat/EventTable

Each event table file is named based on the associated device and variable binding.

Contents of an Event Table File

Event table files are used to enumerate the possible attribute values, OID values, or integer bit values in a variable binding sent with a trap, and to associate those values with corresponding text values that can be used in event messages.

Associate the Attribute Values in a MIB Table with Text Values

To associate an attribute value in a MIB table with a text value, the file must iterate each possible value in hexadecimal format and its associated text value, as shown in the following example:

```
0x00000001 Reconfiguration
0x00000002 Signal-Loss
0x00000003 Bit-Streaming
0x00000004 Contention-Streaming
0x000000ff None
```

Associate OID Values with Text Values

To associate an OID value with a text value, the file must iterate each possible OID value and its associated text value, as shown in the following example:

```
1.3.6.1.4.1.1563.1.2.1.1.3.2.36.2.6 dot6
1.3.6.1.4.1.1563.1.2.1.1.3.2.36.2.5 dot15
1.3.6.1.4.1.1563.1.2.1.1.3.2 dot7
```

Associate Integer Bit Values with Text Values

To associate an integer bit value with a text value, the file must iterate each possible integer bit value and its associated text value, as shown in the following example:

```
1 dsx1NoAlarm
2 dsx1RcvFarEndLOF
3 dsx1XmtFarEndLOF
4 dsx1RcvAIS
```

This example works only with OneClick. Change the integer values to its equivalent 8 byte HEX values to make the event table file work with AlarmNotifier, as shown in the following example:

```
0x00000001 dsx1NoAlarm
```

```
0x00000002 dsx1RcvFarEndLOF
0x00000003 dsx1XmtFarEndLOF
0x00000004 dsx1RcvAIS
```

Probable Cause Files

About Probable Cause Files

A *probable cause file* is an ASCII text file that defines the symptoms, probable causes, and recommended corrective actions for an alarm. When an alarm is generated as a result of an event, the text in the associated probable cause file is displayed on the Alarm Details tab in OneClick. In this way, OneClick users are provided with information that can assist in resolving the abnormal condition. There exists a probable cause file for each alarm generated due to a DX NetOps Spectrum event.

The events provided with DX NetOps Spectrum that generate alarms all have probable cause files that define appropriate alarm-related messages. In addition, whenever you create a new, custom event that generates an alarm (either using MIB Tools when you are mapping a trap to the new event or later using Event Configuration) the associated probable cause file for the alarm is automatically created. This means that typically you should not need to manually create a probable cause file.

> **WARNING**
> It is recommended that you create and modify all alarm-related messages using Event Configuration, which updates the appropriate probable cause files on (only) the OneClick web server to which you are connected when you save the changes to a landscape. However, this appendix provides reference information on the proper syntax to use if manual modifications are ever required.

Location of Probable Cause Files

The probable cause files that support the events and alarms provided with DX NetOps Spectrum are installed in the following folder:

<$SPECROOT>/SG-Support/CsPCause

Custom probable cause files that are created by MIB Tools or Event Configuration are installed in the following folder:

<$SPECROOT>/custom/Events/CsPCause

Each probable cause file is named Prob<alarm_cause_code>, where <alarm_cause_code> is an 8-digit (including leading zeros), hexadecimal code that identifies the probable cause of the alarm.

> **NOTE**
> As a convention, events that generate alarms typically use their event codes as alarm cause codes.

Contents of a Probable Cause File

The contents of a probable cause file should include text only. On the first line, specify the cause of the alarm in all capital letters; this information is displayed as the alarm type (or title) on the Alarm Details tab. Also provide one section each on the following:

- The symptoms of the alarm
- The probable causes
- Recommended corrective actions

If there are multiple items under a category, enter the information in numbered list format.

Example: Probable Cause File Using Appropriate Syntax

```
UNKNOWN USER
SYMPTOMS:
The user's SMTP mail transaction failed with error code 550 - unknown user.
PROBABLE CAUSES:
1) The user may have entered an invalid SMTP mail login.
2) The SMTP server login account information may be incorrect.
RECOMMENDED ACTIONS:
1) Have the user check their username and try again.
2) If the username is correct, check the SMTP server login account information.
```

Host System Resources Management

Introducing the Host System Resources Manager

Host resources monitoring is a DX NetOps Spectrum mechanism that defines host resource conditions and thresholds that, when met or violated, generate events and alarms. The goal of resource monitoring is to alert network administrators about significant resource events that could affect host performance and Service Level Agreements.

To help you monitor resources, DX NetOps Spectrum provides management support for the following resource monitoring agents:

- **CA SystemEDGE Agent**
- **CA Unicenter NSM System Agent**
- **Dell OpenManage**
- **Fujitsu ServerView Agent (for PRIMERGY servers)**
- **HP Systems Insight Manager**
- **iAgent**
- **IBM Director**
- **Net-SNMP (UC Davis)**
- **Sun Management Center**

This support for the monitoring agents lets you view and evaluate relevant, up-to-date information about the status of resources on host systems in the network.

Host System Resources Management Concepts

Host System Resources Management Concepts

The following terms and concepts are key to understanding and working with <sp> host system resources management.

Alarm Condition
An alarm condition refers to process thresholds that you specify in an RFC 2790 monitoring rule.

Configuration Threshold
A configuration threshold refers to process thresholds that you specify in an NSM Agent monitoring rule.

File System
A file system is any data storage system on a host.

Host
A host is any computer system that communicates with other systems in the network. In this section, a host refers to any device that is modeled in <sp> and that supports the RFC 2790 host resources MIB, NSM Agent proprietary MIBs, or log file monitoring.

Host Resources
Host resources are the processes, file systems, processors, memory, and other host elements that can be monitored.

Log File
A log file is any file that includes status information about a host or a host application.

Monitor Rule
A monitor rule in <oc> lets you associate <sp> alarms with resource state changes and resource activity thresholds.

Process
A process is any application that runs on a host.

Monitoring Tasks Overview

Monitoring Tasks Overview

This section provides instructions for completing the following tasks in OneClick:

- Create and manage process monitoring rules
- Create and manage the file system monitoring rules
- Create the file system monitoring rule sets that are applied to DX NetOps Spectrum Global Collection containers to automate the creation of monitoring rules
- Create a log file monitor

Creating Process and File System Monitoring Rules

When you create a process or file system monitoring rule for a host model, you specify conditions that cause DX NetOps Spectrum to generate alarms. You can specify multiple available conditions when you create a monitoring rule. You can also specify whether DX NetOps Spectrum generates alarms for the monitoring rule model or the host model.

RFC 2790 Host Resources MIB Monitoring Rule Alarm Conditions and Thresholds

A process monitoring rule for a host that supports the RFC 2790 host resources MIB includes the following alarm conditions:

- Process start
- Process stop
- Process instance count exceeds a certain number
- Process instance count falls below a certain number

A file system monitoring rule includes the following alarm conditions:

- File system utilization threshold is met
- File system goes offline

For more information about RFC 2790 host resources monitoring rules, see Configure RFC 2790 Process Monitoring Rule Parameters.

NSM Agent Monitoring Rule Thresholds

The following table shows the configuration thresholds that you can specify for an NSM Agent process monitoring rule. The available thresholds depend on both the host type (UNIX or Windows) and the version (3.1 or r11) of the agent on the host.

For more information, see NSM Agent Process Monitoring Rule Parameters.

	Platforms and NSM Agent Versions	Platforms and NSM Agent Versions	Platforms and NSM Agent Versions	Platforms and NSM Agent Versions
Configuration Thresholds	**Win r11**	**UNIX r11**	**Win 3.1**	**UNIX 3.1**
Children	X	X	X	X
CPU Usage	X	X	X	X
CPU Usage Long-term		X		
Handles	X			
Instances	X	X	X	X
Restart	X	X		
Runtime	X			
Size	X	X	X	X
Threads	X	X	X	

Using Rule Sets to Automate Monitoring Rule Creation

A rule set is a collection of monitoring rules. You can apply one or more rule sets to a Global Collections container to automate monitoring rule creation for models in the container. When a model that supports the RFC 2790 MIB or the NSM Agent is added to the collection, monitoring rules are automatically configured on the model. Rules are configured for any of the processes or file systems to which rules in the rule set apply.

For example, a rule set that includes a monitoring rule for the svchost.exe process is applied to a Global Collection. The collection is configured to add Windows hosts as the hosts are modeled in DX NetOps Spectrum. The monitoring rule for svchost.exe is configured on all host models that are added to the collection. Conversely, when the hosts are removed from the collection the monitoring rule is removed from the hosts.

Modifications that you make to a rule in a rule set that is associated with a Global Collection apply to all instances of that rule. This type of rule has an indicator that it belongs to (or is "owned" by) a rule set. You can check rule set ownership in the rule set name. The name appears in the Rule Owner field in all monitored process tables and monitored file system tables in OneClick.

Suppose you want to change an alarm condition for svchost.exe monitoring. In the svchost.exe rule, change the maximum process count threshold from 10 to 12. The change then applies to all svchost.exe monitoring rule instances in the collection.

For more information, see Create a Rule Set.

About Creating a Log File Monitor

Agents that support log file monitoring use regular expressions to find the log file text. Typically, you monitor log files to find information about system or application error conditions. Discovery of a text match results in DX NetOps Spectrum generating an alarm on the device where the log file entry originated.

For more information, see Log File Monitoring.

Host Resources Monitoring and Service Level Agreements

Host resource monitoring lets you monitor host resources that can affect the network services that are defined in a Service Level Agreement (SLA). For example, a process monitoring rule can determine whether a virus protection process has

stopped unexpectedly, or whether a malicious process has started on a host. A file system monitoring rule can determine whether a disk drive or physical RAM on a host has reached or is nearing capacity. The viability of a business service can depend on whether processes are running on a host, or whether the host provides adequate data storage capacity.

NOTE
For more information about setting up a service management system and SLAs, see the Service Manager section.

Host Resource Events and Alarms Reporting

The DX NetOps Spectrum Report Manager application lets you generate reports on events and alarms for host models. Alarms and reports are generated for threshold violations for monitored processes and the file systems. Alarms are also generated from error messages that are parsed from log files.

NOTE
For more information, see the Report Manager section.

Getting Started with Managing Host System Resources in OneClick

This section describes how to invoke workspaces where you configure monitoring rules, rule sets, and views of monitored host resource information.

NOTE
For more information about the OneClick Console interface elements, see the Using OneClick section.

Access the Workspace for Creating and Managing Monitoring Rules

Create and manage monitoring rules from the context of a host model that supports a monitoring agent.

Follow these steps:

1. Select the host for which you want to create a monitoring rule from the Contents panel.
2. Expand the System Resources option under the Information tab in the Component Detail panel.
   The Running and Monitored Processes section lets you create and manage process monitoring rules. For more information, see Process Monitoring.
   The Monitored Logs and Process Logs section lets you create log file monitoring rules. For more information, see Log File Monitoring.
   The File Systems section lets you create file system monitoring rules. For more information, see File System Monitoring.

Access the Workspace for Creating and Managing Rule Sets

Unlike monitoring rules that you create for a particular host, DX NetOps Spectrum creates different rules for Global Collections. For any host that is included in a Global Collection to which the rule set has been applied, DX NetOps Spectrum creates rules that you specify in a rule set. This feature automates the process of creating monitoring rules for multiple, different host types.

Manage rule sets in the Contents panel.

Follow these steps:

- Select Locater, System & Application Monitoring, All Monitoring Rules.
  The Contents panel lists any rule sets that have been created.
  Default rules are not set. See Working with Monitoring Rule Sets for details about creating and managing rule sets and applying them to Global Collections.

View Monitoring Rule Information

OneClick lets you view comprehensive information about monitored processes and file systems in the Component Detail panel.

To view information about a process monitoring rule:

- Select Locater, System & Application Monitoring, All Monitored Processes.

> **NOTE**
> Because process models are not created for rules for SystemEDGE hosts, monitoring rules for SystemEDGE hosts do not appear in this view.

To view information about a file system monitoring rule:

- Select Locater, System & Application Monitoring, All Monitored File Systems.

The view provides information about the selected host and the monitoring configuration on the host. The monitoring agent that is associated with the rule determines the information that the view provides.

Process Monitoring

A process monitoring rule specifies the criteria that, when met, cause DX NetOps Spectrum to generate alarms. This section describes how to set up process monitoring rules for host models with process monitoring agents. See Working with Monitoring Rule Sets for information about setting up an automated method for creating process monitoring rules for models included in Global Collection containers.

Create a Process Monitoring Rule

You can create a process monitoring rule for a host model regardless of whether the process is running on the host.

> **NOTE**
> Only the users with the appropriate privileges can create process monitoring rules. For more information, see System and Application Monitoring Privileges.

Follow these steps:

1. In the Contents panel, select the host model for which you want to create a monitoring rule.
   Information for this host device appears in the Component Detail panel.
2. In the Component Detail panel, on the Information tab, expand System Resources, Running, and Monitored Processes.
   The available process options for this host type appear.

   > **NOTE**
   > RFC 2790 indicates a host that supports the RFC 2790 host resources MIB.
3. Expand both Running Processes and Monitored Processes.
   The Running Processes table lists running processes for the selected host model.
   The Monitored Processes table lists process monitoring rules that have been created for the selected host model.
4. To create a process monitoring rule for the selected host model, use *one* of the following methods:
   - If the process is running, right-click the process in the Running Processes table and select 'Monitor this process.'
   - If the process is not running, it is excluded from the Running Processes table. Click Add above the Monitored Processes table. You can then specify process monitoring rules for processes that run periodically but are not currently running that you want to know about when they start. For example, you want to know when virus scan and system maintenance processes run.

NOTE
For NSM Agent monitoring, use this method when you want to create a monitoring rule that watches multiple, different processes that the match criteria specify. For more information, see NSM Agent Process Monitoring Rule Parameters.

A dialog opens, depending on the host type. If you selected a process from the Running Processes table, the dialog includes the process name and other information. If you invoked the dialog using the Add option, you are prompted to provide all process information.

5. Configure process monitoring rule settings:
   – For agents that support the RFC 2790 host resources MIB, see RFC 2790 Process Monitoring Rule Parameters.
   – For agents that support NSM Agent versions 3.1 or r11, see NSM Agent Process Monitoring Rule Parameters.
   – For SystemEDGE host agents, see SystemEDGE Host Process Monitoring Rule Parameters.
6. Click OK.
   The following events occur:
   – The process monitoring rule is added to the Monitored Processes table. The table columns represent predefined process identifier information specific to the monitoring agent type on the selected host. The rule applies to all identical instances of the process that satisfy the process match selection criteria.
   – A process model is created for RFC 2790 and NSM Agent rules.

     NOTE
     Local ownership in a monitoring rule indicates that the rule has been created explicitly for a particular host. As a result,it is not part of a rule set. For more information about rule sets, see Working with Monitoring Rule Sets.
7. Specify the alarm generation and agent polling options, which are located above the Monitored Processes table, depending on host type:
   – **Watch For New Processes Every (seconds)**
     Specify the frequency with which DX NetOps Spectrum inspects the Running Processes table for new instances of processes that a monitoring rule is watching. DX NetOps Spectrum updates the 'Number Running' value in the Monitored Processes table for a monitored process when it detects that a new instance of the process is running.
   – **Generate Alarm On**
     Select a destination for alarms resulting from rule violations. You can specify that DX NetOps Spectrum create alarms on the process monitoring rule model or the host model.
   – **Agent Poll Interval (seconds)**
     Specify the frequency with which the agent collects process information from the host device. The minimum value is 30 seconds.
   – **Agent Poll Method**
     Specify how and when the agent collects process data:
     - **disabled**
       The agent does not retrieve process information (by polling or by get request), and it sets all status indications for alarm conditions to passive or ok.
     - **poll-interval-and-query**
       The agent retrieves process information both by polling and by the get request.
     - **poll-interval-only**
       The agent retrieves process information by polling only.
     - **query-only**
       The agent retrieves process information by get request only.

Differentiating Processes

At any time, a host can run multiple instances of a particular process. The svchost.exe process on Windows hosts and the nfsd process on Linux and UNIX hosts are typical examples. You can create a process monitoring rule that applies to all process instances, to some process instances, or to a single process instance. For example, if you decide to monitor all instances of svchost.exe, do not differentiate them by parameters or names.

For DX NetOps Spectrum, the alarm conditions and thresholds that are specified in the svchost.exe process monitoring rule apply to all instances of the process. Assuming that the rule specifies an alarm for process starts and stops, DX NetOps Spectrum generates an alarm for each start and stop, for each instance. In other words, DX NetOps Spectrum applies the rule to each entry in the Running Processes table that matches an entry (by process name) in the Monitored Processes table.

You can create a rule for an instance or a group of identical instances of a process. In this case, you must differentiate the instance or group of instances from the instances that you do not want to monitor. You can use a unique name, parameters, or both to distinguish them. The differentiation options let you make many different types of distinctions between process instances.

Process Monitoring Rule Parameters

The section describes the process monitoring rule parameters for the following host types:

- RFC 2790
- NSM Agent
- SystemEDGE Host

RFC 2790 Process Monitoring Rule Parameters

You can specify the following parameters when you create process monitoring rules for hosts that support RFC 2790 monitoring:

- Process identifiers, including a process name and process differentiator
- Process start/stop and process count alarm conditions
- Polling of the Running Processes table for new instances of processes with associated monitoring rules

Monitor Information

You can selectively monitor all instances of a process or specific instances of a process. Use the following parameters in the monitoring rule:

- **Process Name**
  Identifies the process on the host model. You can differentiate a process instance with this setting , or you can also use the Match Parameters field to provide more precise differentiation.
  For hosts that support RFC 2790 monitoring, the value that is entered in this field is case-insensitive. It converts to lowercase, as displayed in the Monitored Processes (RFC 2790) table. Also, duplicate entries are not allowed. If a new entry is created with the same Process Name (and Match Parameters value, if specified), the new entry replaces the existing entry. Any configuration settings that were changed are updated.
- **Match Parameters**
  Specifies one or more process parameters that differentiate identically named instances of the same process. You can add parameters or can modify the parameters that are included with a process before you save the configuration. This setting is used along with the Process Name to differentiate a process instance.
- **Descriptive Name**
  Identifies a nickname for the process. We recommend supplying a descriptive name that more clearly conveys the purpose or function of a process than its proper name (for example, "java runtime" for the javaw.exe process). This setting does not serve as a process differentiator.

Alarm Configuration for RFC 2790

You can specify the following alarm conditions in an RFC 2790 monitoring rule:

- **Process Count Less Than**
  Specifies whether DX NetOps Spectrum generates an alarm when a process instance count is less than a particular value. DX NetOps Spectrum clears the alarm when the process count is equal to or greater than the value.
- **Process Count Greater Than**
  Specifies whether DX NetOps Spectrum generates an alarm when a process instance count is greater than a particular value. DX NetOps Spectrum clears the alarm when the process count is equal to or less than the value.
- **Process Start**
  Specifies whether DX NetOps Spectrum generates an alarm whenever the process is started. DX NetOps Spectrum clears the process-start alarm when the process stops.
- **Process Stop**
  Specifies whether DX NetOps Spectrum generates an alarm whenever the process is stopped. DX NetOps Spectrum clears the process-stop alarm when the process starts.

NSM Agent Process Monitoring Rule Parameters

Process monitoring rules are defined in the Add Monitored Process dialog, as described in Create a Process Monitoring Rule. When you create a process monitoring rule for a host that supports NSM Agent monitoring, you can specify the following parameters:

- Process monitoring rule identifiers
- Process match criteria
- Configuration threshold monitoring options
- Configuration threshold values
- Advanced options, such as aggregate status evaluation policy, resource cluster group, and aggregate violation threshold

> **NOTE**
> Your NSM Agent version and agent host platform determine your access to all of these settings and to the options that are described in this section.

You can specify the agent polling interval and method for all NSM Agent versions on all platforms. For more information, see Create a Process Monitoring Rule.

Monitor Information

The Add Monitored Process dialog includes the following process monitoring rule identifiers. Available identifiers depend on the NSM Agent version and agent host platform:

- **Monitor Name**
  Identifies the name of the monitoring rule. DX NetOps Spectrum distinguishes identical monitoring rule configurations by the monitor name. This name must be unique.
- **Descriptive Name**
  Identifies a monitoring rule nickname or brief descriptive term.

The following table describes the attributes, or fields, that uniquely identify the process monitor for each agent type:

Version	Monitor Identification Fields
Win r11	Monitor Name* Descriptive Name (optional)
UNIX r11	Monitor Name* Descriptive Name (optional)

Win 3.1	Descriptive Name (optional) Process Name* Path* User*
UNIX 3.1	Process Name* Parameters * Path * User *

* Uniquely identifies the process monitor.

Process Match Criteria

Before you implement a process monitoring rule on an NSM agent, identify the processes that you want DX NetOps Spectrum to evaluate according to the threshold criteria. You can use regular expressions and string comparisons to identify processes.

WARNING
The r11 agent supports regex for match criteria, but the 3.1 agent supports wildcard (*) use only.

The following table describes the attributes, or fields, that are used as process matching criteria for each type of NSM agent.

NOTE
For r11 NSM Agents, Match Type applies to the combination of all the other match criteria attributes. It defines how the combinations of the other process match fields are evaluated.

Version	Monitor Identification Fields
Win r11	Process Name Match Type Path User
UNIX r11	Process Name Match Type Parameters Path User
Win 3.1	Process Name Path User
UNIX 3.1	Process Name Parameters Path User

The Add Monitored Process dialog includes the following fields and options, depending on the NSM Agent version and agent host platform you are working with:

- **Process Name**
  Identifies the process or processes text pattern to match. You can use literal string identifiers or regular expressions to specify a text search pattern.

NOTE
If no other process match criteria is specified, all processes matching the name in the Process Name field are monitored.

- **Match Type**
  Lets you specify the process or processes that match or do not match the process match criteria.

  NOTE
  Process Name match criteria are case-insensitive.

  Options include:
  - **positive-regular-expression**
    The agent searches for processes that match the process name as a regular expression.
  - **negative-regular-expression**
    The agent searches for processes that do not match the process name as a regular expression.
  - **positive-string-compare**
    The agent searches for processes that match the process name as a string comparison.
  - **negative-string-compare**
    The agent searches for processes that do not match the process name as a string comparison.
- **Parameters**
  Identifies the process arguments to match. You can specify parameters as a literal string or a regular expression depending on the version of NSM and the platform you are using.
- **Path**
  Identifies the path name of the process or processes to match. You can specify paths as a literal string or a regular expression.
- **User**
  Identifies the user name of the process account to match. You can specify user names as a literal string or a regular expression depending on the version of NSM and the platform you are using.

Threshold Configuration for NSM Agent

Threshold configuration defines what is watched by the monitor. You can specify multiple thresholds when you create a monitoring rule. For example, you can instruct the monitor to watch only the amount of CPU time that a process consumes. Or you can instruct the monitor to watch CPU usage and process children, threads, and handles, and also how often a process restarts.

DX NetOps Spectrum generates Major (Orange) alarms for violations of warning thresholds and Critical (Red) alarms for violations of critical thresholds. Alarm generation depends on the overall status of the monitoring rule.

The thresholds that you can specify depend on the host platform (Windows or UNIX) and the NSM Agent version (3.1 or r11) running on the host.

The following table describes the threshold and monitoring options that are available for each NSM agent:

Threshold	Monitoring Options (Platform and Agent Version)	Monitoring Options (Platform and Agent Version)	Monitoring Options (Platform and Agent Version)	Monitoring Options (Platform and Agent Version)
	Win r11	**UNIX r11**	**Win 3.1**	**UNIX 3.1**
Children	do-not-monitor down-warning down-critical	do-not-monitor down-warning down-critical	do-not-monitor down-warning down-critical	do-not-monitor monitor

CPU Usage	do-not-monitor warning-only critical-only minimum-only maximum-only all	do-not-monitor warning-only critical-only minimum-only maximum-only all	do-not-monitor warning-only critical-only both	do-not-monitor warning-only critical-only both
CPU Usage Long-Term	N/A	do-not-monitor warning-only critical-only minimum-only maximum-only all	N/A	N/A
Handles	do-not-monitor down-warning down-critical	N/A	N/A	N/A
Instances	do-not-monitor down-warning down-critical	do-not-monitor down-warning down-critical	do-not-monitor down-warning down-critical	do-not-monitor monitor
Restart	do-not-monitor down-warning down-critical	do-not-monitor down-warning down-critical	N/A	N/A
Runtime	do-not-monitor down-warning down-critical	N/A	N/A	N/A
Size	do-not-monitor warning-only critical-only minimum-only maximum-only all	do-not-monitor down-warning down-critical	do-not-monitor warning-only critical-only both	do-not-monitor monitor
Threads	do-not-monitor down-warning down-critical	do-not-monitor down-warning down-critical	do-not-monitor down-warning down-critical	N/A

NOTE
Specify the value '-1' for a particular minimum or maximum value threshold to disable the threshold. You can selectively specify that the monitor watches, for example, a minimum threshold but not a maximum threshold, or the reverse.

- **Children**
  Specifies whether the monitor watches the process children count.

  NOTE
  For version r11 on Windows, this option is in the Resources, Type drop-down list.
- **CPU Usage/CPU Shortterm Usage/CPU Longterm Usage**
  Specifies whether the monitor watches the amount of CPU time that a process uses.
  Some of the available options include the following:
  - **Warning Threshold**
    This value can be between one (1) and ninety-nine (99) percent, but it must fall below the critical threshold percent value. For multiple process instances, the maximum of all instances is compared with this value.
  - **Critical Threshold**

This value can be between two (2) and one hundred (100) percent, but it must exceed the warning threshold percent value. For multiple process instances, the maximum of all instances is compared with this value.

CPU Interval

This value defines the total value in seconds to use as the base to calculate the CPU value. Specifically, the CPU usage of a process, in seconds, refers to this interval. You can set the value to any value greater than zero (0) or -1.

- If set to -1, the CPU value is calculated as the CPU usage, in seconds, used up to the current time since the start of the agent or the creation of the process monitoring rule.
- If the CPU interval is set to a value greater than the current agent polling interval and this time has not elapsed for the first time, the CPU value is extrapolated.
- If the CPU interval is set to a value smaller than the current agent polling interval, the CPU value is calculated as the appropriate fraction of the value for the last agent polling interval.
- If the CPU interval is set to a value greater than the current agent polling interval and this time has already elapsed, the CPU value is calculated as the sliding sum (the sum of the value for the current poll interval and the value calculated at the last poll) weighted according to its fraction of the CPU interval.
- If the interval is set to -1, any overloading (%) used for the thresholds are ignored.

- **Min/Max Units**
  The unit of measure, in seconds or as a percentage, used for CPU usage thresholds.
- **Instances**
  Specifies whether the monitor watches the process instance count.
- **Resources**
  Specifies whether the monitor watches one of the following resource types:
  - **threads**
    Specifies the process thread count.
  - **handles**
    Specifies the total number of handles currently opened by each thread in the process.
  - **children**
    Specifies the process children count.
  - **runtime**
    Specifies the time, in seconds, that the process has been running since it was created.
- **Restart**
  Specifies whether the monitor watches the process restart count. Determines the policy that the agent uses to determine when to set the status of the restart alarm condition to down for a threshold violation.
  - **none-should-stop-or-start**
    Sets the status to down if any process stops or starts.
  - **none-should-stop**
    Sets the status to down if any process stops.
  - **none-should-start**
    Sets the status to down if any process starts.
  - **some-should-continue**
    Sets the status to down if all processes stop.
- **Size**
  Specifies whether the monitor watches the amount of memory (in kilobytes) that a process consumes.
- **Threads**
  Specifies whether the monitor watches the process thread count.

NOTE
For version r11 on Windows, this option is in the Resources, Type drop-down list.

Monitoring Options

A monitoring option specifies whether the NSM Agent watches a particular configuration threshold and which threshold types (warning or critical, minimum, or maximum values) to watch.

Monitor drop-down lists in the Add Monitored Process dialog contain the following options depending on the host platform (Windows or UNIX), the NSM Agent version (3.1 or r11), and the particular alarm condition you are configuring:

- **do-not-monitor**
  No alarm. The agent disregards threshold settings.
- **monitor**
  Critical alarm. The agent monitors minimum and maximum values for all thresholds.
- **warning-only**
  Major alarm. The agent evaluates only the warning thresholds (both minimum and maximum) to determine the status of the process.
- **critical-only**
  Critical alarm. The agent evaluates only the critical thresholds (both minimum and maximum) to determine the status of the process.
- **minimum-only**
  Major (warning) and Critical (critical) alarms. The agent evaluates only the minimum thresholds (both warning and critical) to determine the status of the process.
- **maximum-only**
  Major (warning) and Critical (critical) alarms. The agent evaluates only the maximum thresholds (both warning and critical) to determine the status of the process.
- **all**
  Major (warning) and Critical (critical) alarms. The agent evaluates all thresholds.
- **down-warning**
  Major alarm. When the resource is in a bad condition the agent uses a warning severity. This lets you designate a threshold violation as less crucial than a down-critical violation.
- **down-critical**
  Critical alarm. When the resource is in bad condition the agent uses a critical severity. This lets you designate a threshold violation as more crucial than a down-critical violation.
- **both**
  Major (warning) and Critical (critical) alarms. The agent evaluates both warning and critical thresholds to determine the status of the process.

Advanced Options

Advanced options let you specify an evaluation policy for configuration threshold violations when the monitor watches two or more processes, a process resource cluster group, and an aggregate alarm condition violation threshold that when met degrades the status of a process and triggers DX NetOps Spectrum alarm generation.

> **NOTE**
> The advanced options available depend on which host platform (Windows or UNIX) and NSM Agent version (3.1 or r11) you are configuring.

- **Evaluation Policy (r11 only)**
  Specifies how the agent calculates values that it compares to alarm condition thresholds for a monitor that watches multiple, different processes. It also specifies which other processes are included in the threshold violation culprits list.

  > **NOTE**
  > NSM Agent version 3.1 compares the worst values (the individual policy) from all watched process instances to alarm condition thresholds to determine threshold compliance.

  Evaluation Policy options include:

  - **individual (default)**
    Specifies that the agent compares the worst values (lowest and/or highest) of all process instances to alarm condition threshold values. If a value violates a threshold condition, the culprits list includes all instances individually violating the most severe threshold.
  - **min**
    Specifies that the agent compares the lowest values (minimum) of all process instances to alarm condition threshold values. If a value violates a threshold condition, the culprits list includes all instances with the same minimum value.
  - **max**
    Specifies that the agent compares the highest values (maximum) of all process instances to alarm condition threshold values. If a value violates a threshold condition, the culprits list includes all instances with the same maximum value.
  - **sum**
    Specifies that the agent compares the cumulative values (sum) of all process instances to alarm condition threshold values. If a value violates a threshold condition, the culprits list includes all instances.
  - **avg**
    Specifies that the agent compares the average values of all process instances to alarm condition threshold values. If a value violates the threshold condition, the culprits list includes all instances individually violating the most severe threshold.
- **Cluster Resource Group (r11 only)**
  Identifies the cluster resource group.
- **Aggregate Violation Threshold**
  This option specifies the consecutive number of agent polling cycles for which any threshold is required to be in a less-than-ok state before the aggregate status for the monitor changes. This value must be greater than 0. The Aggregate Violation Threshold field is not available for UNIX 3.1.
  The Status field in the Monitored Processes table for the selected host model indicates the aggregate status condition.

If the NSM Agent Fails to Retrieve Process Information

If the NSM Agent subagent that is responsible for retrieving process monitoring information goes down, DX NetOps Spectrum responds as follows:

- Generates an NSM PROCESS MONITORING AGENT LOST alarm on the host model
- Asserts a suppressed APPLICATION_LOST alarm condition on the process models

When the process monitoring subagent restarts, DX NetOps Spectrum clears the NSM PROCESS MONITORING AGENT LOST alarm on the host model and clears the APPLICATION_LOST alarms on the associated process models.

Status Indications for NSM Agent Process Monitoring Rules

The Status field in the Monitored Processes table for the selected host model indicates the aggregate status condition of the monitor. The status field represents the worst-case aggregate for the status values of each threshold that is defined on the monitor.

The aggregate status enters a suboptimal state when any threshold is in a violated state over a particular number of consecutive agent polling cycles. The Aggregate Violation Threshold field defines the number of consecutive times that any threshold is in a violated state before the aggregate status value changes. DX NetOps Spectrum does not generate alarms for violated thresholds until the aggregate status is in a suboptimal state.

SystemEDGE Host Process Monitoring Rule Parameters

Process monitoring rules are defined in the Add Process Monitor Table Entry dialog. For more information, see Create a Process Monitoring Rule.

When you create a process monitoring rule for a SystemEDGE host, you can specify the following parameters:

- Process monitoring rule identifiers
- Configuration threshold monitoring options
- Configuration threshold values
- Advanced options, such as sending traps and monitoring a parent process or Windows service

NOTE
When a rule is created for a SystemEDGE host, a process model is not created. As a result, when you search for and view rules in the Locater tab, the monitoring rule does not appear.

Monitor Information

The Add Process Monitor Table Entry dialog includes the following process monitoring rule identifiers:

- **Index**
  Specifies an integer value that uniquely identifies the process monitor entry. If this field is left blank or set to 0 when creating an entry, an unused index is automatically selected.
- **Process Name**
  Identifies the process text pattern to match. You can use literal string identifiers or regular expressions to specify a text search pattern.
- **Match Parameters**
  Indicates whether to match both the process name and the parameters or simply the process name.
- **Description**
  Identifies a monitoring rule nickname or brief descriptive term.

Threshold Configuration

Threshold configuration defines the attributes and metrics that the monitor watches. Depending on the SystemEDGE host version, you can specify applicable thresholds when you create a monitoring rule.

The following parameters are available:

- **Attribute**
  Is the process attribute to monitor.
- **Operator**
  Is the Boolean operator that is used to compare the current value to the threshold value. 'No Operation' only tracks the current value; it does not compare against the threshold value.
- **Threshold Value**
  Is the threshold value against which the agent compares the current value. This parameter works with the Operator parameter.
- **Interval**
  Is the time (in seconds) between successive samples by the agent. Values range from 30 to MAXINT and must be a multiple of 30.
- **Sample Type**
  Is the type of sampling to perform on the monitored object.
  - **absolute**
    Measures the actual value (for example, a gauge).
  - **delta**
    Measures a change in value (for example, a counter).
- **Severity**
  Is the severity to use for the object state model.

  NOTE
  This threshold value is not available for all SystemEDGE host versions.
- **Object Class**

Is the object class to use for the object state model.

NOTE
This threshold value is not available for all SystemEDGE host versions.

- **Object Attribute**
  Is the object attribute to use for the object state model.

  NOTE
  This threshold value is not available for all SystemEDGE host versions.
- **Object Instance**
  Is the object instance to use for the object state model.
  Note: This threshold value is not available for all SystemEDGE host versions.
- **Execute Action**
  Specifies the command that is executed if a threshold is crossed (a string, up to 4096 characters). The action script must be present on the host.
- **Send Arguments**
  Indicates whether to send default arguments to action scripts or programs (for example, trap type or a description field).

Advanced Options

Advanced options let you specify actions to perform during the monitoring process.

- **Send SNMP Traps**
  Indicates whether to send SNMP traps.
- **Send Process Start Traps**
  Indicates whether to send process start traps.
- **Handle Process Start Traps**
  Indicates whether to execute actions, log events, and send SNMP traps when a process start trap occurs. Acts as a convenience flag for setting the three individual flags at the same time.
- **Send Not-Ready Trap**
  Indicates whether to send not-ready traps.
  - **Single**
    A single not-ready trap is sent.
  - **Continuous**
    A continuous not-ready trap is sent.
- **Send Process Clear Traps**
  Indicates whether to send process clear traps.
- **Monitor Parent Process**
  Indicates whether to monitor the parent process.
- **Monitor Windows Service**
  Indicates whether to monitor the Windows service.
- **Reinitialize Entry**
  Indicates whether to reinitialize the entry.
- **Log Events**
  Indicates whether to log events.
- **Monitor For *x* Processes**
  Indicates whether to monitor for the specified number of processes.
- **Breach After *x* Consecutive Events**
  Indicates whether to send a trap after the specified number of consecutive events.
- **Allow For *x* Consecutive Breach Traps**
  Indicates whether to allow for the specified number of consecutive breach traps.

Creation of SystemEDGE Process Models

For granular monitoring of services and processes running on SystemEDGE host, you can enable the creation of process models of all the monitored processes.

This functionality is enabled by adding the "enable_sysedge_process_modeling_support=true" configuration to the ".vnmrc" file. When this functionality is enabled, you can see the list of process models in the "Locater, System and Application Monitoring, All Monitored Processes".

When you configure the alarms to be generated on these process models, the alarms are mapped to the process models, and not to the SystemEDGE. As a result, only the service monitoring a process which is down is shown effected.

Edit a Process Monitoring Rule

You can edit local process monitoring rules. You can also edit rules that are owned by rule sets in the context of a host model. In the latter case, the modification transforms the ownership of the rule from the rule set to the model (Rule Owner value converts to Local).

> **NOTE**
> To edit a rule, you must have a user model in all landscapes where the rule was created.

Follow these steps:

1. In the Contents panel, select the model with the process monitoring rule that you want to edit.
   Information for this host device appears in the Component Detail panel.
2. In the Component Detail panel, in the Information tab, expand System Resources, Running, and Monitored Processes, Monitored Processes.
   The Monitored Processes table lists process monitoring rules for the selected model.
3. Select the process monitoring rule that you want to edit, and click Edit.
   The Edit Process Monitor Table Entry dialog opens.
4. Modify the settings as required, and click OK.
   Changes to the process monitoring rule for the selected model take effect immediately.

Delete a Process Monitoring Rule

You can delete local process monitoring rules and rules that are owned by rule sets for a host model. In the former case, monitoring stops for the process. In the latter case, the deletion also stops monitoring for the particular model by the rule from the rule set. However, the deletion of a rule set rule is temporary. Process monitoring that is specified by the rule is reestablished the next time the rule set is updated. See Deleting a Rule Outside of a Rule Set for more information.

When you delete a process monitoring rule, DX NetOps Spectrum and the process monitoring agent stop monitoring all identical (non-differentiated) instances of the process that is specified in the rule. In addition, the rule is removed from the agent MIB.

Follow these steps:

1. In the Contents panel, select the model with the process monitoring rule that you want to delete.
   Information for this host device appears in the Component Detail panel.
2. In the Component Detail panel, expand System Resources, Running, and Monitored Processes, Monitored Processes.
   The Monitored Processes table lists process monitoring rules for the selected model.
3. Select the process monitoring rule that you want to delete, and click Delete.
   You are prompted to confirm the deletion.
4. Confirm the deletion.
   The process monitoring rule is deleted.
   Process monitoring that is specified by the rule for the selected model stops immediately.

Maintenance Mode

When a process monitor is in maintenance mode, the process is not monitored. Any events or alarms that are related to monitoring of that process are not generated.

Placing a process monitor into maintenance mode can be useful when a single application on a host where several critical applications are running is upgraded. You can place only the process that is associated with that particular application into maintenance mode while the upgrade is taking place. Monitoring of the other applications can continue.

Maintenance mode can also be scheduled, which allows you the ability to specify what time of day to alarm on processes.

Maintenance mode is only supported for RFC 2790 and NSM Agent process monitoring.

NOTE
When a host device is in maintenance, process monitoring for that device is automatically suspended.

Place Process Monitor in Maintenance Mode

A process monitor can be placed into maintenance mode at any time. This procedure describes how to place a process monitor into maintenance mode immediately.

Follow these steps:

1. In the Contents panel, select the host model for which you want to place a process monitor into maintenance mode.

   NOTE
   Maintenance mode is only supported for RFC 2790 and NSM Agent process monitoring.
2. In the Component Detail panel, in the Information tab, expand System Resources, Running and Monitored Processes, and RFC 2790, if applicable.
3. Perform *one* of the following steps from the Monitored Processes or Monitored Processes (RFC 2790) table to place a process monitor into maintenance mode:
   - Select the process monitor to place into maintenance mode, and click the Maintenance button above the table.
   - Right-click the process monitor to place into maintenance mode, and select 'Toggle Maintenance Mode.'

   The process monitor is now in maintenance mode, and its icon changes to brown. The mode is reflected in the Condition column of the Monitored Processes table. If the icon does not change immediately, click Refresh.

   NOTE
   You can use this same procedure to take a process monitor out of maintenance mode.

Schedule Maintenance Mode for Process Monitor

You can schedule the times when a process monitor is in maintenance mode by applying a maintenance schedule. You can apply an existing schedule, or you can create a new one. You can apply multiple schedules to a process monitor.

Follow these steps:

1. On the Locater tab, select System & Application Monitoring, All Monitored Processes.
2. Select the process monitor in the Contents panel to which you want to apply a maintenance schedule.

   NOTE
   Maintenance mode is only supported for RFC 2790 and NSM Agent process monitoring.
3. In the Component Detail panel, expand the Process Monitor Details subview, locate 'In Maintenance,' and click Schedule.
   The Add/Remove Schedules dialog opens. Any maintenance schedules that are applied to the process monitor appear in the Current Schedules column.
4. (Optional) Apply an existing schedule. Select a schedule from the Available Schedules column, and click the left arrow to move it to the Current Schedules column.
5. Click Create.

The Create Schedule dialog opens.

6. Select a Start Date, a Start Time, and either an End Time or Duration for the schedule.
7. Select a Recurrence factor.

   NOTE
   Leave the Recurrence set to None to create a one-time maintenance mode window.
8. Supply a Description to identify the schedule.
9. Click OK.
   The Create Schedule dialog closes. The new schedule appears in the Current Schedules column in the Add/Remove Schedules dialog.
10. Click OK.
    The Add/Remove Schedules dialog closes. The maintenance mode scheduling changes are applied to the process monitor. The changes appear in the Assigned Maintenance Schedules list.

Roll Down Maintenance Alarms from the Device Model

When a device is placed in maintenance mode, the maintenance alarms that are generated on the device can be rolled down to the associated process models. Enable this propagation by setting the rollMMAlarmToApp attribute to true. When this option is enabled, the alarms also roll down to the application models that are associated with the device.

NOTE
For information about placing a device into maintenance mode, see the Using OneClick section. For information about modifying model attributes, see the Modeling and Managing Your IT Infrastructure Administration section.

Process Model Internal Condition

DX NetOps Spectrum can maintain the condition of process models without having the process monitoring events generate alarms. This functionality can be useful when incorporating multiple monitored process models within a service or resource monitor. Rather than having alarms generated on the device or process models each time a process monitoring rule is violated, you can have a single alarm on the service model when the service policy is violated.

The functionality is disabled by default. Enable it by using the Attribute Editor to set the value of the EnableInternalCondition attribute to Yes. This attribute is on the device model for NSM Process Monitoring and on the rfc2790App application model for RFC 2790 Process Monitoring. When the functionality is either enabled or disabled, any existing process monitoring alarms are cleared on the associated process models, and their InternalCondition attribute is set to Normal.

While the functionality is enabled and the 'Generate Alarm On' option is set to 'Process Model', process monitoring events do not generate alarms. Instead, the InternalCondition attribute of the process model is set to reflect the condition of the process model. The value of this attribute is displayed on the Internal Condition column of the System & Application Monitoring, All Monitored Processes table on the Locater tab. The value can also be found on the Attributes tab of the process model.

While the Internal Condition functionality is enabled, do not map log-file monitors to any process models. The log-file monitoring events continue to generate alarms.

For hosts that support RFC 2790 monitoring:

- When the functionality is enabled or disabled:
  - Manually clear any process monitoring alarms that exist on the affected device model.
  - Process count conditions are reasserted; however, the process start and process stop conditions are not reasserted.
- If a SpectroSERVER is restarted while the Internal Condition functionality is enabled on a device in its landscape, you must disable the functionality and then reenable it on the device. These steps ensure that the Internal Condition of the process models accurately synchronizes with the actual condition of the process monitor.

File System Monitoring

A file system monitoring rule (RFC 2790) specifies file system alarm conditions that cause DX NetOps Spectrum to generate alarms. Alarms are generated when the conditions occur on a host model for which the rule is created:

- File system utilization
- File system goes offline

This section describes how to set up file system monitoring for particular host models. See Working with Monitoring Rule Sets for information about automating the creation of file system monitoring rules for models in Global Collection containers.

Create a File System Monitoring Rule

When you create a file system monitoring rule, you can specify any file system, online or offline. DX NetOps Spectrum creates a model for the rule.

During file system monitoring rule configuration, you define the alarm conditions that cause DX NetOps Spectrum to generate alarms. Examples of such alarm conditions include system utilization thresholds or a file system that goes offline.

Note: Only users with the appropriate privileges can create file system monitoring rules. For more information, see System and Application Monitoring Privileges.

Follow these steps:

1. In the Contents panel, select the model with the file system that you want to monitor.
   Information for this host device appears in the Component Detail panel.
2. In the Component Detail panel, expand System Resources, File Systems.
   The available file system monitoring options for this host type appear.
3. Expand File Systems (RFC 2790) and Monitored File Systems (RFC 2790).
   The File Systems (RFC 2790) table lists file systems for the selected model. The Monitored File Systems (RFC 2790) table lists file system monitoring rules that have been created for the selected model.
4. Use *one* of the following methods to create a file system monitoring rule for the selected model:
   - If the file system you want to monitor is available, right-click the file system in the File Systems (RFC 2790) table and select Monitor this File System.
   - If the file system is not available and therefore not included in the File Systems (RFC 2790) table, click Add on the Monitored File Systems (RFC 2790) table. This lets you specify, for example, a file system that is offline that you want to know about and monitor when it does come online.

   The Add File System Monitor dialog opens. If you selected a file system from the File Systems (RFC 2790) table, the box includes the file system name.
5. Configure the settings. The available settings include the following:
   - **File System Name**
     Specifies the file system. If you added a file system to monitor that is not currently available, type the name. If you added an available file system, the name is entered automatically.
     For hosts that support RFC 2790 monitoring, the value that you enter in this field is case-insensitive. This field converts to lowercase, as displayed in the Monitored File Systems (RFC 2790) table. Duplicate entries are not allowed. If a new entry is created with the same File System Name, the new entry replaces the previous one, updating any configuration settings that were changed.
   - **Description**
     Specifies a nickname, or alias, for the file system.
   - **Threshold Value Type**

Specifies whether to monitor file system utilization thresholds in terms of capacity percentage or unit of storage (Bytes, Kbytes, Mbytes, Gbytes, Tbytes).

- **Utilization Thresholds**
  Specifies thresholds for events, minor alarms, major alarms, and critical alarms. DX NetOps Spectrum clears threshold alarms when metrics no longer exceed thresholds.
- **Alarm if Offline**
  Specifies whether DX NetOps Spectrum generates an alarm when the file system goes offline. DX NetOps Spectrum clears the alarm when the file system comes back online.

6. Click OK.
   The file system monitoring rule is added to the Monitored File Systems (RFC 2790) table. DX NetOps Spectrum generates alarms in response to the alarm condition threshold violations specified in the rule.

   NOTE
   A value of "Local" in the Rule Owner field of a monitoring rule indicates that the rule has been created explicitly for a particular host and is therefore not part of a rule set. For more information about rule sets, see Working with Monitoring Rule Sets.
7. Select a destination for alarms resulting from rule violations from the Generate Alarm On drop-down list. You can specify that DX NetOps Spectrum create alarms on the monitoring rule model or the host model.

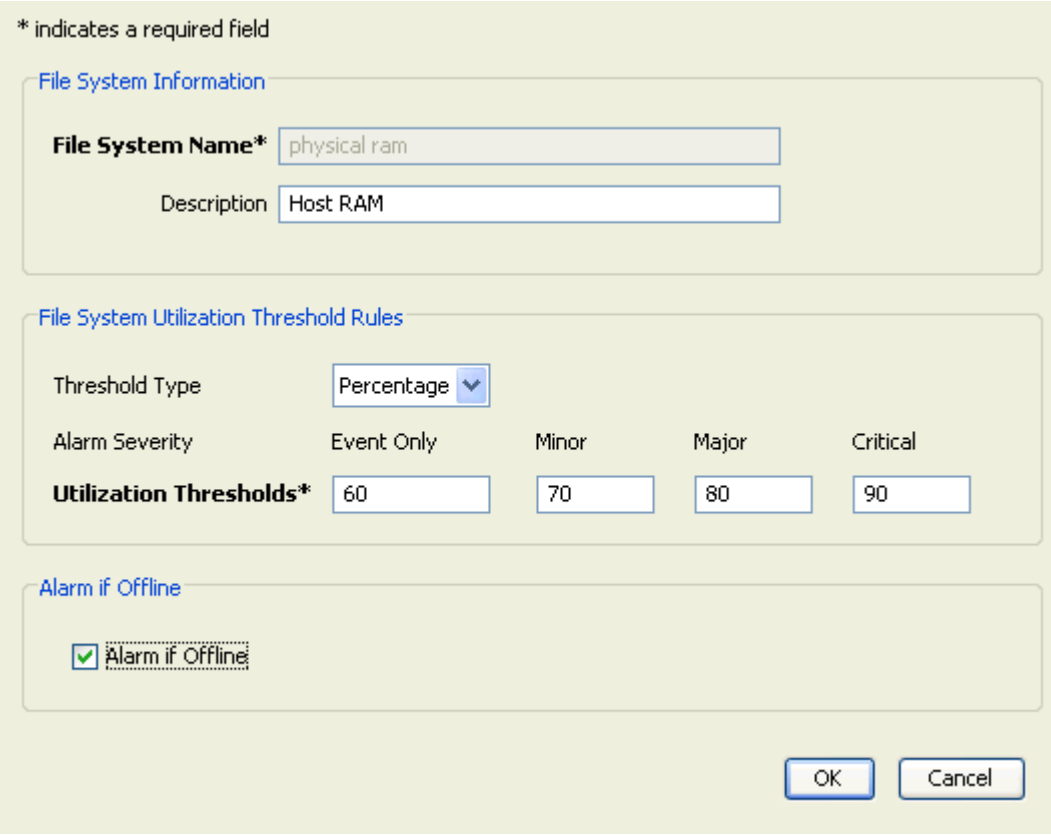
Edit a File System Monitoring Rule

You can edit local file system monitoring rules and rules that are owned by rule sets for a host model. In the latter case, the modification transforms the ownership of the rule from the rule set to the model (Rule Owner value converts to Local). However, the changes and the ownership conversion are temporary because the original rule specifications and ownership are reestablished the next time the rule set is updated. See Editing a Rule Outside of a Rule Set for more information.

NOTE
To edit a rule, you must have a user model in all landscapes where the rule was created.

Follow these steps:

1. In the Contents panel, select the model with the file system monitoring rule that you want to edit.
   Information for this host device appears in the Component Detail panel.
2. In the Component Detail panel, expand System Resources, File Systems, Monitored File Systems (RFC 2790).
   The Monitored File Systems (RFC 2790) table lists file system monitoring rules.
3. Select the file system rule that you want to edit, and then click Edit.
   The Edit File System Monitor dialog opens. Read-only settings are grayed out.



4. Modify settings as required, and click OK.
   Changes to the file system monitoring rule for the selected model take effect immediately.

Delete a File System Monitoring Rule

You can delete local file system monitoring rules and rules that are owned by rule sets for a host model. In the former case, monitoring stops for the file system. In the latter case, the deletion also stops monitoring for the particular model by the rule from the rule set. However, deletion of a rule set rule is temporary because file system monitoring specified by the rule is reestablished the next time the rule set is updated. See Deleting a Rule Outside of a Rule Set for more information.

Follow these steps:

1. In the Contents panel, select the model with the file system monitoring rule you want to delete.
   Information for this host device appears in the Component Detail panel.
2. In the Component Detail panel, expand System Resources, File Systems, Monitored File Systems (RFC 2790).
   The Monitored File Systems (RFC 2790) table lists file system monitoring rules.
3. Select the file system monitoring rule that you want to delete, and then click Delete.
   You are prompted to confirm the deletion.
4. Confirm the deletion.
   File system monitoring that is specified by the rule for the selected model stops immediately.

Working with Monitoring Rule Sets

A rule set is a collection of monitoring rules for processes and file systems you can apply to a Global Collection. Rule set automates the process of setting up and managing monitoring for hosts modeled in DX NetOps Spectrum. When

you create a process or file system monitoring rule for a particular host model, that rule applies only to that host model. If you want to apply the same rule to other host models, create the same rule again and again for each host model. If you want to edit the rule for all models, modify each instance of the rule for each host model. This task is obviously a tedious and inefficient way to manage host monitoring for numerous host models. By applying rule sets to Global Collections, you leverage a more efficient method of managing IT infrastructure resources. When host models are added to a collection, DX NetOps Spectrum creates monitoring rules that reference the processes or file systems for those models. Furthermore, when monitoring rules in rule sets are modified, the modifications apply to all host models in the collection. When host models are removed from a collection, all monitoring rules for the models are removed too.

Create a Rule Set

You can create rule sets that contain multiple monitoring rules for both host processes and file systems, or you can create rule sets that include one or the other. You can apply as many rule sets as you want to a Global Collection. You can also apply the same rule set to multiple collections.

> **WARNING**
> Plan your rule set implementation carefully to avoid duplicating rules or implementing conflicting rules. Duplicate or conflicting rules can cause unexpected results and make troubleshooting difficult. Also verify that the Global Collections to which you apply rule sets include host models that are appropriate for the monitoring rules in those rule sets.

As you create rule sets, keep the following points in mind:

- Rule sets must have unique names.
- Rules that are included in rule sets do not override identically named local monitoring rules for host models included in Global Collections. If a local monitoring rule has been created for a particular host model and the model is included in a Global Collection that has a rule set applied to it that contains an identically named rule, the local rule is preserved and remains in effect for the model.

> **NOTE**
> Only users with the appropriate privileges can create monitoring rule sets. For more information, see System and Application Monitoring Privileges.

Follow these steps:

1. Select Locater, System & Application Monitoring, All Monitoring Rules.
   The Contents panel lists any rule sets that have been created.
   Default rule sets are not present.
2. Click Create a New Rule Set by
   Type 
3. Select one of the following options, depending on the agent you are working with:
   - RFC2790
   - NSM Agent:
     - r11 Windows
     - r11 UNIX
     - 3.1 Windows
     - 3.1 UNIX

   The 'New Rule Set' dialog appears.
4. Type a name for the rule set in the Rule Set Name field, and then click OK.
   The new rule set appears in the list. You can now add process monitoring and file system monitoring configuration rules to the rule set. And you can apply the rule set to a Global Collection container.

Add a Monitoring Rule to a Rule Set

You can add monitoring rules to a rule set before or after you apply the rule set to a Global Collection.

Follow these steps:

1. Select Locater, System & Application Monitoring, All Monitoring Rules.
   The Contents panel lists rule sets.

   NOTE
   If no rule sets are listed, create a rule set for the rule, as described in Create a Rule Set.
2. Select the rule set to which you want to add the monitoring rule.
   The Component Detail panel displays information about the rule set.
3. On the Information tab, specify the type of rule to add to the rule set:
   – To add a process monitoring rule, expand Process Monitoring Rules.
   – To add a file system monitoring rule, expand File System Monitoring Rules.

   NOTE
   The NSM rule sets do not support file system monitoring rules.

   Each rules table lists rules that have been added to the rule set.
4. Click Add for the type of rule to add to the rule set.
   Either the Add Monitored Process dialog or the Add File System Monitor dialog opens.
5. Configure settings.
   – See Process Monitoring Rule Parameters for information about configuring a process monitoring rule.
   – See Create a File System Monitoring Rule for information about creating a file system monitoring rule.
6. Click OK.
   The rule is added to the rule set.

Apply a Rule Set to a Global Collection

Applying a rule set to a Global Collection automates the process of creating monitoring rules. DX NetOps Spectrum automatically creates monitoring rules for all models in the Global Collection.

As you apply rule sets to Global Collections, consider the following facts:

- If you remove models from the Global Collection, all monitoring rules that are specified by the rule set are removed from the models.
- If you edit a rule from a rule set for a particular model that is included in a Global Collection, the ownership of a rule changes to local ownership. The rule is no longer associated with the rule in the rule set and applies only to that particular model.
- If you delete a rule set that is associated with a Global Collection or vice versa, the rules that are specified by the rule set are removed from the models in the collection.

Follow these steps:

1. Select Locater, System & Application Monitoring, All Monitoring Rules.
   The Contents panel lists rule sets.

   NOTE
   If no rule sets are listed, create a rule set as described in Create a Rule Set.
2. Right-click the rule set or sets that you want to apply to a Global Collection, and select Apply/Remove Global Collection(s).
   The 'Apply and Remove Collection(s) to/from the Rule Set' dialog appears.
   All Global Collections that are listed in the left side of the dialog are currently applied to the selected rule set. Global Collections that are listed on the right side have not been applied.
3. In the Not Applied To list, double-click the Global Collection to which you want to apply the rule set.

The selected Global Collection moves to the Applied To list.

NOTE
You cannot apply Global Collections to multiple rule sets simultaneously.

4. (Optional) Select the Reapply check box to reapply the Global Collection or Collections that are already applied to the rule set when you click OK in the dialog.
5. Click OK to apply your changes.

NOTE
Only the changes that you made in the dialog are applied. A Global Collection that already appears in the Applied To list is not reapplied unless you have selected the Reapply check box.

The Applied Global Collections List in the Information tab of the selected rule set shows the Global Collections to which it is applied.

Remove a Rule Set from a Global Collection

When you remove a rule set from a Global Collection, DX NetOps Spectrum removes monitoring rules in the rule set from all models in the Global Collection.

Follow these steps:

1. Select Locater, System & Application Monitoring, All Monitoring Rules.
   The Contents panel lists rule sets.
2. Right-click the rule set or sets from which you want to remove a Global Collection and select Apply/Remove Global Collection(s).
   The 'Apply and Remove Collection(s) to/from the Rule Set' dialog appears.

NOTE
You can also click the icon in the Results tab toolbar to launch this dialog.

All Global Collections that are listed in the left side of the dialog are currently applied to the selected rule set. Global Collections that are listed on the right side have not been applied.

3. In the Applied To list, double-click the Global Collection that you want to remove from the rule set.
   The selected Global Collection is moved to the Not Applied To list.

NOTE
You cannot remove Global Collections from multiple rule sets simultaneously.

4. (Optional) Select the Reapply check box to reapply the Global Collection or Collections that are already applied to the rule set when you click OK in the dialog.
5. Click OK to apply your changes.

NOTE
Only the changes you made in the dialog are applied. A Global Collection that already appears in the Applied To list is not reapplied unless you have selected the Reapply check box.

The Applied Global Collections List in the selected Information tab of the rule set is updated. The Global Collection or Collections that you removed are no longer displayed.

Edit a Rule in a Rule Set

When you edit a rule in a rule set that is applied to a Global Collection, the revised rule settings extend to all models in the Global Collection.

NOTE
To edit a rule, you must have a user model in all landscapes where the rule was created.

Follow these steps:

1. Select Locater, System & Application Monitoring, All Monitoring Rules.
   The Contents panel lists rule sets.
2. Select the rule set with the rule that you want to edit.
   The Component Detail panel displays information about the rule set.
3. In the Component Detail panel, specify the type of rule to edit, either a process monitoring rule or a file system monitoring rule.
   Each rule type table lists rules that have been included in the rule set.
4. Select a rule, and click Edit.
   The Edit dialog opens.

   NOTE
   Some settings are unavailable for edit.
5. Edit settings, and then click OK.
   The modified settings take effect immediately.

Editing a Rule Outside of a Rule Set

Under some circumstances, you may want to modify a monitoring rule for a particular model in a Global Collection even though the rule belongs to a rule set that has been applied to a Global Collection. You might not want the modification to apply to the rule in the rule set because the modification would then apply to all models in the Global Collection. But you still want to keep the model in the collection.

In this case, convert the rule to a local version for the model. You can modify this rule from the context of the particular model outside of the rule set to achieve this result.

Delete a Rule from a Rule Set

When you delete a rule from a rule set that is applied to a Global Collection, the rule is removed from all models in the Global Collection.

Follow these steps:

1. Select Locater, System & Application Monitoring, All Monitoring Rules.
   The Contents panel lists rule sets.
2. Select the rule set with the rule that you want to delete.
   The Component Detail panel displays information about the rule set.
3. In the Component Detail panel, specify the type of rule to delete from the rule set, either a process monitoring rule or a file system monitoring rule.
   Each rule table lists rules that have been included in the rule set.
4. Select the rule, and click Delete.
   The rule is removed from the rule set and from its rule table.

Deleting a Rule Outside of a Rule Set

Under some circumstances, you may want to delete a rule for a particular model in a Global Collection even though that rule belongs to a rule set that is applied to a Global Collection. You might not want to delete the rule in the rule set and thus delete it for all models in the Global Collection. But you still want to keep the model in the collection.

In this case, delete this rule from the context of the particular model outside of the rule set. When the association between the rule set and the Global Collection is updated, however, the deleted rule is recreated for the model.

Delete a Rule Set

When you delete a rule set that is applied to a Global Collection, all rules in the rule set are removed from the models in the collection.

Follow these steps:

1. Select Locater, System & Application Monitoring, All Monitoring Rules.
   The Contents panel lists all available rule sets.
2. Select the rule set to delete, and click Delete.

Log File Monitoring

This chapter describes how to set up log file monitoring in OneClick for the following agents:

- iAgent
- CA SystemEDGE Agent
- CA Unicenter NSM Agent

This chapter also describes how to configure iAgent syslog server monitoring and trap forwarding to DX NetOps Spectrum.

Setting up log file monitoring entails the following tasks:

- Specifying the criteria that initiate the trap and the event generation. The traps and events are generated when the type of information that you specify is detected in a log file.
- (Optional) Specifying an association between a log file and a monitored process model. Events are then generated in response to a log file entry for the process model. The events are generated on the process model rather than on the process host model.

About the Log File Monitoring Process

Various devices on your network can be configured to send data to log files on an iAgent, SystemEDGE agent, or NSM server. Or the applications on one of these servers can send data to a log file. In either case, these agents can be configured to monitor these log files and generate SNMP traps based on the content in log file entries.

Log file monitoring involves setting up a text pattern matching system that detects and parses log files for the type of information that you specify. The monitoring agent then sends a trap to DX NetOps Spectrum that contains data about the parsed text. This data is then mapped to a DX NetOps Spectrum event and an alarm is asserted on the agent model or the device or process to which it pertains. You can also use an event condition rule to configure DX NetOps Spectrum to create a more granular event, and optionally an alarm, from the "text match in log file" event. As a result, you receive notifications of events that have occurred in your infrastructure and that indicate potential or actual problems. For more information, see Event Configuration .

The syntax of the log file that you are monitoring depends on the type of log file and the data that is sent to it. Because application log files are matched directly to the applications that you are monitoring, no special log file syntax is required. However, DX NetOps Spectrum processes other log files that gather data from other devices differently. Therefore, these log file entries must conform to certain syntax criteria.

Regardless of the type of log file that you want to monitor, whether an application log file or a syslog file containing entries for multiple applications from multiple devices, you must define a regular expression (regex) that identifies, or parses, the type of information that you want to monitor. The regex syntax must be compatible with the type of agent. When matching text is found, the monitoring agent sends a trap to DX NetOps Spectrum that contains the matching text. DX NetOps Spectrum associates the trap to an event that is asserted on the host model.

NOTE
For more information about defining regular expressions, see Event Configuration .

Configuring DX NetOps Spectrum to Process Syslog File Matches explains how to configure an agent to monitor log files for strings of information that generate a trap.

NOTE
iAgent can only monitor log files that exist on the iAgent server. It cannot monitor log files on a mapped network drive.

Log File Syntax

You can monitor application logs or log files that receive data from other devices, such as Syslog files. No special syntax is required for log files that monitor application logs. However, for DX NetOps Spectrum to assert the trap information about the appropriate device model, log files that receive information from devices on the network must have the following format, which is based on the BSD Syslog and Cisco IOS format:

```
<MessagePrefix>%<MessageHeader><Additional_Information>
```

- ***<MessagePrefix>***
  Contains the date and time of the message and the IP address or the host name of the source of the information contained in the entry. There can be other information that is interspersed within the prefix, but it must contain these two pieces of information.

  NOTE
  If a host name is used to identify the source, it can be of the form myhost.ca.com or myhost.
- ***<MessageHeader>***
  Must have the format < *A*>-< *B*>-
  - ***< A>***
    Contains any number of uppercase alpha characters, underscores, or the string "Aprisma."
  - ***< B>***
    Contains any number of uppercase alpha characters, numeric characters, or underscores.
  - ***< C>***
    Contains any number of uppercase alpha characters, underscores, or dashes.
- ***<Additional_Information>***
  Can contain any data.

In general, this syntax can be found in the following types of log files:

- Kiwi syslog file entries from a Cisco or Riverstone device.
- Kiwi syslog file entries from another type of device that uses the *<MessageHeader>* format described previously.
- CA log files.

NOTE
For information about configuring DX NetOps Spectrum to process iAgent traps, see Configuring DX NetOps Spectrum to Process Syslog File Matches.

Create Log File Monitors for iAgent Hosts

The following procedure describes how to set up log file monitoring for iAgent host agents.

Follow these steps:

1. In the Contents panel, select the model with the log file you want to monitor.
   Information for this host device appears in the Component Detail panel.

2. In the Component Detail panel, in the Information tab, expand System Resources, Monitored Logs and Process Logs, Monitored Logs.
   The Monitored Logs list is displayed.

   NOTE
   Some list fields are agent-specific.
3. Click Add in the Monitored Logs list.
   The Add Log File Monitor dialog for the agent opens.
4. Configure log file monitor settings as needed. Pay particular attention to the following mandatory and optional settings:
   - **Log File Name**
     Identifies the monitored log file.
   - **Regular Expression**
     Identifies the text patterns to parse in the log file.

     NOTE
     For more information about defining regular expressions, see the Event Configuration section.
   - **Description**
     Indicates the purpose of the monitor to other users.
   - **Send Trap on Match/Send Trap**
     Specifies whether the agent sends a trap to DX NetOps Spectrum when the regular expression detects a matching text pattern.
5. Click OK.
   The monitoring configuration is added to the Monitored Logs list, and monitoring begins immediately.

Log File Monitors for NSM Agents

You can set up log file monitoring and file monitoring for NSM Agents in OneClick. The following definitions describe how these two monitors differ:

- **NSM Log File Monitor**
  An NSM Log File Monitor watches contents of a file for specific patterns.
- **NSM File Monitor**
  An NSM File Monitor simply watches for the existence or absence of a file.

NSM Agent log file monitoring lets you perform the following tasks.

- Edit and view file monitors for NSM Agents
- Edit and view log monitors for NSM Agents
- View status changes for file and log monitors for NSM Agents

Set Up a Log File Monitor for NSM Agents Using OneClick

You can use OneClick to set up log file monitoring for NSM host agents.

Follow these steps:

1. In the Contents panel, select the model with the log file that you want to monitor.
   Information for this host device appears in the Component Detail panel.
2. In the Component Detail panel, in the Information tab, expand System Resources, Monitored Logs and Files, Monitored Logs.
   The Monitored Logs list appears.
3. Click Add in the Monitored Logs list.
   The Add Log File Monitor dialog opens.
4. Configure log file monitor settings as needed. The following options are available:

- **Monitor Name**
  Identifies the name of this log file monitor.
- **File Name**
  Identifies the full path and file name (or wildcarded file name) of the log file to monitor.
- **Positive Pattern**
  Places the monitor in a DOWN state if the specified regular expression is found in the file.
- **Negative Pattern**
  Places the monitor in a DOWN state if the specified regular expression is *not* found in the file.
- **Toggle Positive Pattern**
  Places the monitor in an UP state if the specified regular expression is found in the file. This field is only available when the Status Policy is toggled or toggledEOF.
- **Toggle Negative Pattern**
  Places the monitor in an UP state if the specified regular expression is *not* found in the file. This field is only available when the Status Policy is toggled or toggledEOF.
- **Start**
  Is the starting character position.
- **End**
  Is the ending character position.
- **Status Policy**

Defines how the monitor handles files. The following options are available:

- **poll**
  a. Sets monitor status to UP at the beginning of each poll. If a match is made, the state changes to DOWN. The file is scanned from the last read location unless it is a new monitor, in which case the entire file is read.
- **historical**
  a. Sets monitor status to DOWN when a match occurs, and status remains DOWN for the life of the log file. Therefore, the log file is recreated. The file is scanned from the last read location unless it is a new monitor, in which case the entire file is read.
- **startFromPreviousRead**
  a. Sets monitor status to DOWN when a match occurs, and status remains DOWN until you reset it. The file is scanned from the last read location.
- **toggled**
  a. Lets you specify a DOWN pattern, as with the historical attribute, and also an UP pattern (formed with the toggle positive and negative pattern attributes), which is compared for changing the state back to UP. The file is scanned from the last read location.
- **firstLineOnly**
  a. Reads only the first line of a file. The monitor status is set to UP at the beginning of each poll. If a match is found, the state changes to DOWN.
- **pollEOF**
  a. Sets the monitor status to UP at the beginning of each poll. If a match is found, the state changes to DOWN. The file is scanned from the last read location unless it is a new monitor, in which case reading starts at the end of the file.
- **startFromPreviousReadEOF**
  a. Sets the monitor status to DOWN when a match occurs, and status remains DOWN until you reset it. The file is scanned from the last read location unless it is a new monitor, in which case reading starts at the end of the file.
- **toggledEOF**
  a. Lets you specify a down pattern, as with the historical attribute, and also an up pattern (formed with the toggle positive and negative pattern attributes) which is compared for changing the state back to UP. The file is scanned from the last read location unless it is a new monitor, in which case reading starts at the end of the file.
- **rescan**
  a. Rescans the file from the beginning if the file length has increased. If the file exceeds 10 KB, sets the monitor to UNKNOWN.
- **Monitor Status**
- Lets you disable the status side of the monitor without matching trap sending. The following options are available:
- **downCritical**
  a. Indicates that the state change works as configured and a critical alert is raised.
- **doNotMonitor**
  a. Indicates that the log file is monitored, but the state is always UP.
- **downWarning**
  a. Indicates that the state change works as configured and a warning alert is raised.
- **Trap Send Policy**

Defines the policy that is applied to the monitor status traps. The following options are available:

- **never**
  a. Indicates that the state change never causes traps to be sent.
- **once**
  a. Indicates that the state change trap is sent only when the monitor state changes.
- **perPoll**
  a. Indicates that the state change trap is sent every poll, even if the state does not change but a match condition occurred since the last poll.
- **each**
  a. Indicates that the state change trap is sent for each match that the agent finds. For toggle attributes, when the monitor goes down, the toggle pattern is the next match that is looked for. As a result, subsequent down patterns are not matched.
- **Match Trap Policy**
  Defines the policy that is applied to the match traps. The following options are available:
- **send**
  a. Sends a trap for each match that is found. For toggle attributes, when the monitor goes down, the toggle pattern is the next match that is looked for. As a result, subsequent down patterns are not matched unless status monitoring is switched off.
- **doNotSend**
  a. Does not send a trap for each match found.
- **History Policy**
  Defines whether trap details are stored in the history table. The following options are available:
- **generateHistory**
  a. Indicates that status traps are recorded in the history table.
- **noGenerateHistory**
  a. Indicates that status traps are not recorded in the history table.

5. Click OK.
   The monitoring configuration is added to the Monitored Logs list. Monitoring begins immediately.

Set Up a File Monitor for NSM Agents Using OneClick

You can use OneClick to set up log file monitoring for NSM host agents.

Follow these steps:

1. In the Contents panel, select the model with the file that you want to monitor.
   Information for this host device appears in the Component Detail panel.
2. In the Component Detail panel, in the Information tab, expand System Resources, Monitored Logs and Files, Monitored Files.
   The Monitored Files list is displayed.
3. Click Add in the Monitored Files list.
   The Add File Monitor dialog opens.
4. Configure file monitor settings. Pay particular attention to the following mandatory and optional settings:
   - **Monitor Name**
     Identifies the name of the file monitor.
   - **File Name**
     Identifies the name of the file that this monitor watches.
   - **File Exists**
     Indicates whether the file exists.
   - **Trap Send Policy**

Specifies the frequency with which the NSM Agent sends traps. The following settings are available:

- **never**
  a. Never sends a trap.
- **once**
  a. Sends a trap only when the state has changed.
- **perPoll**
  a. Sends a status trap at each poll if the state is DOWN.
- **History Policy**
  For details about History Policy settings, see Set Up a Log File Monitor for NSM Agents Using OneClick.

5. Click OK.
   The monitoring configuration is added to the Monitored Files list, and monitoring begins immediately.

Create Log File Monitors for SystemEDGE Hosts

You can use OneClick to set up log file monitoring for a SystemEDGE host agent.

Follow these steps:

1. In the Contents panel, select the model with the log file that you want to monitor.
   Information for this host device appears in the Component Detail panel.
2. In the Component Detail panel, in the Information tab, expand System Resources, Monitored Logs and Process Logs, Monitored Logs.
   The Monitored Logs list appears.
3. Click Add in the Monitored Logs list.
   The Add Log File Monitor dialog for the agent opens.
4. Configure the log file monitor settings as needed:
   - **Log File Name or Directory Name**
     Identifies the monitored log file or directory, depending on the Monitor Type.
   - **Monitor Type**
     Identifies whether to monitor a log file or a directory.
     - **File**
       Indicates that a log file is monitored.
     - **Directory**
       Indicates that a directory is monitored. You can also specify whether to Monitor Recursively and whether to Follow Symbolic Links.
   - **Description**
     (Optional) Is a brief description to indicate the purpose of the monitor, for example.
   - **Interval**
     Is the interval, in minutes, between successive scans of the log file.
   - **Severity**
     Is the severity to use for this monitor entry.
   - **Parse File**
     Is a regular expression to match in the log file (up to 256 characters). The value must be a valid string, as defined in ed(1).
   - **Does Not Match**
     Indicates whether to apply a logical NOT operator when parsing the log file.
   - **Execute Action**
     Is a string that specifies the command that is executed after finding a match (up to 4096 characters). The action script must be present on the host.
   - **Send Arguments**

Indicates whether to send default arguments to action scripts or programs (for example, trap type, a description field).

- **Send SNMP Traps**
  Specifies whether the agent sends a trap to DX NetOps Spectrum when the regular expression detects a matching text pattern.
- **Send Not-Ready Trap**
  Indicates whether to send not-ready traps.
  - **Single**
    A single not-ready trap is sent.
  - **Continuous**
    A continuous not-ready trap is sent.
- **Reinitialize Entry**
  Indicates whether to reinitialize the entry.
- **Breach After *x* Consecutive Matches**
  Indicates whether to send a trap after the specified number of consecutive matches have occurred.
- **Log Events**
  Specifies whether to log events.

5. Click OK.
   The monitoring configuration is added to the Monitored Logs list, and monitoring begins immediately.

Log-to-Process Mapping

DX NetOps Spectrum can generate an event on the process that a parsed log file entry references rather than on the host model. To configure such events, verify that the process monitoring rule for the host model references the process. And associate the process with the log file that includes the entry that is related to the process. See Process Monitoring for more information about process monitoring rules.

Specify a Mapping for RFC 2790 Agents and SystemEDGE Hosts

You can use OneClick to specify a mapping of log to process for RFC 2790 Agents.

Follow these steps:

1. In the Contents panel, select the model with the log files that you want to monitor.
   Information for this host device appears in the Component Detail panel.
2. In the Component Detail panel, expand System Resources, Monitored Logs and Process Logs, Monitored Process Log File Mappings.
   The Monitored Process Log File Mappings list appears.
3. Click Add in the Monitored Process Log File Mappings list.
   The Add Log to Process Mapping dialog appears.
4. Enter the following data:
   - **Log File Name**
     Is the log file to monitor.
   - **Process Name**
     Is the process that is specified in the process monitoring rule.
5. Click OK.
   The mapping is added to the Monitored Process Log File Mappings list. DX NetOps Spectrum generates events on the monitored process model whenever text about the process is parsed from a log file.

Mapping for NSM r11 Agents

You can use OneClick to specify a mapping of log to process for NSM r11 Agents.

NOTE
You cannot create mappings for NSM 3.1 Agents.

Follow these steps:

1. In the Contents panel, select the model with the log files that you want to monitor.
   Information for this host device appears in the Component Detail panel.
2. In the Component Detail panel, expand System Resources, Monitored Logs and Files, Monitored Process Log File Mappings.
   The Monitored Process Log File Mappings list appears.
3. Click Add in the Monitored Process Log File Mappings list.
   The Add Log to Process Mapping dialog opens.
4. Enter the following data:
   – **Log File Name**
     Name of the log file.
   – **Monitor Name**
     Name of the process monitor. This value likely differs from the name that is specified for the process in the process monitoring rule.
5. Click OK.
   The mapping is added to the Monitored Process Log File Mappings list. DX NetOps Spectrum generates events on the monitored process model whenever text about the process is parsed from a log file.

Managing Monitored Log and Process Log Mapping Settings

You can use OneClick to edit and delete monitored log and process log file mapping settings.

- To edit monitored log and process log file mapping settings, select a configuration entry that you want to modify, click Edit on the configuration entry list, edit the entry, and then click OK.

  NOTE
  You cannot edit an active monitor entry. To edit a monitor entry that is in an active state, change the status of the entry to notInService(2) or notReady(3). You can perform this task in MIB Tools, using the SET command.
- To delete monitored log and process log file mapping settings, select a configuration entry that you want to remove, click Delete on the configuration entry list, and then click OK.

Configuring the Product to Process Syslog File Matches

You can configure DX NetOps Spectrum to process a Syslog file matches from iAgent, SystemEDGE, and NSM Agents.

Trap Processing Overview

Each trap that an agent generates based on the content of a log file entry has an OID. This OID generates the DX NetOps Spectrum event 0x3e00009 based on the trap mapping in the AlertMap file of the agent. This event is asserted on the model.

The matched line of each log file entry (up to 255 characters) and the log file name that generated the trap is sent as part of the trap information. DX NetOps Spectrum parses the trap data to determine the original source of the log file entry. The source can be an IP Address, hostname, DX NetOps Spectrum model handle, or application log file name.

Processing Traps That Contain an IP Address, Host Name, or Model Handle

If an IP address, hostname, or a model handle has been extracted as the source of the log file entry, DX NetOps Spectrum can find the device model that matches the IP address, hostname or model handle and can assert the event

on this model. If the log file entry conforms to the syntax described in Log File Syntax, to make the event asserted on the device model meaningful, you can create a ParseMap file to customize the event and its contents.

NOTE
DX NetOps Spectrum contains many ParseMap files. You do not always have to create one.

If no ParseMap file is created, the event that is routed to the device model is the same event asserted on the mode of the agent.

Disable DNS Lookup for Syslog Traps

If a hostname is received as part of the Syslog trap, DX NetOps Spectrum (until 10.2.3) uses DNS Lookup to resolve the hostname to an IP address and locates the associated model. Or, DX NetOps Spectrum checks for the host entry in the /etc/host file.

In your environment, if the DNS Server is not accessible or you do not want to contact DNS Server or /etc/hosts file to resolve hostnames, you can use the ‘preventDNSlookup’ attribute (introduced in 10.3) to stop DNS Lookup.

From 10.4.2.2, when the 'preventDNSlookup' attribute is enabled, instead of resolving the hostname to IP address, DX NetOps Spectrum fetches the model handles of the models from DX NetOps Spectrum Database that has the model name same as hostname received in the Syslog message and asserting Syslog trap on those models. If there are multiple models whose model name matches the hostname from the Syslog message, the trap is asserted on all the matched models.

To enable the ‘preventDNSlookup’ attribute follow these steps:

1. Select the VNM model of a landscape in the Topology tab
2. Go to Component details->Attributes tab
3. Search for ‘preventDNSlookup’ attribute and double click it to edit
4. Change the value to Yes. The 'preventDNSlookup’ attribute is enabled for the selected VNM of the landscape. If you want to enable “preventDNSlookup” attribute on VNM models of other landscapes, repeat all the above steps.

NOTE

The default value for ‘preventDNSlookup’ attribute is ‘No’. When the value is ‘No’ DX NetOps Spectrum uses ‘DNSlookup’ to resolve the hostnames.

Create ParseMap Files

ParseMap files specify the event that is associated with the information in the incoming trap. In addition, ParseMap files allow you to specify that portions of the log file entry text be used as event variables. You can use these variables in conjunction with an Event Rule to process the event.

NOTE
For information about event processing and Event Rules, see theEvent Configuration *section*.

As described in Log File Syntax, a log file entry contains the following components:

```
<MessagePrefix>%<MessageHeader><Additional_Information>
```

DX NetOps Spectrum identifies the ParseMap file that processes the trap by finding the ParseMap file whose name matches the text of the *<MessageHeader>* from the log file entry. The following log is an example of a log file entry:

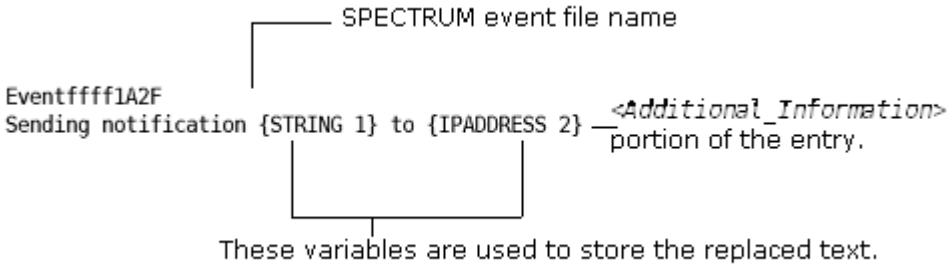
```
2004-2-19 11:19:14 Local7.Info 172.19.38.36 Feb 19 09:14:50
%SNMP-I-SENT_TRAP, Sending notification linkUp to 192.168.32.44
```

The *<MessageHeader>* portion of the entry is SNMP-I-SENT_TRAP. As a result, DX NetOps Spectrum looks for a ParseMap file named SNMP-I-SENT_TRAP. Create a ParseMap file for each log entry with a unique *<MessageHeader>* that you configure to generate a trap.

NOTE
Many ParseMap files are available for use in DX NetOps Spectrum. You can find them in the following directory: *<$SPECROOT>*/SS/CsVendor/ParseMaps.

Follow these steps:

1. Create new text file using any text editor.
   A text file is ready for editing.
2. In the first line of the text file, type the new event file name for the event that you want to generate. The event file name must begin with Eventfffff and end with xxxx where x is any valid hexadecimal number.
   For example, Eventfffff1A2F and Eventfffff1234 are the valid event file names; Event012za8b is not.
3. Add a new line in the text file (press the Enter key).
4. Use this line as the *<Additional_Information>* portion of the log file entry. You can specify portions of this text as event variables, which can be used to process the event with an Event Rule.
   Specify variables using a data type and an integer. Valid data types are STRING, STRINGNOWS, INTEGER, and IPADDRESS. See STRING Data Type Usage Guidelines for important information.
   The following image shows a valid ParseMap file for the log entry shown in the previous section. The variable 1 stores Uplink as a String. The Variable 2 stores 192.168.32.44 as an IP Address.



5. Save the text file in the *<$SPECROOT>*/SS/CsVendor/ParseMaps directory. The name of this text file must match the *<MessageHeader>* portion of the log file entry. In this example, the filename would be SNMP-I-SENT_TRAP.

   NOTE
   Do not include a file extension in the filename.

   Only the first two lines of the ParseMap file are processed. The information that you include on subsequent lines is not processed but can be included for informational purposes.

ParseMap File Example

The following sequence of lines is an example of a ParseMap file that is provided with DX NetOps Spectrum named SYS-0-MOD_TEMPMAJORFAIL. The ParseMap file can be found in the following directory: *<$SPECROOT>*/SS/CsVendor/ParseMaps.

```
Event04bd1522
Module {STRING 1} major temperature threshold exceeded
%SYS-0-MOD_TEMPMAJORFAIL: Module {STRING} major temperature threshold exceeded
```

This instructs a matched syslog file:

```
Jul 28 10:56:42 [10.253.9.11.2.45] 7931: *Jul 28 10:50:47.271:
%SYS-0-MOD_TEMPMAJORFAIL: Module 100 major temperature threshold exceeded
```

This causes the event Event04bd1522 to be generated on the model with the IP address 10.253.9.11, even though the agent generated the trap.

STRING Data Type Usage Guidelines

This section provides important information about using STRING data types in your ParseMap files.

Valid STRING Data Types

As mentioned in "To create a ParseMap file," the following data types are valid for use in variables.

- **STRING**
  Matches all string characters up to the next literal, data type, or to the end of a string.
- **STRINGNOWS**
  Matches all string characters up to the next space, literal, data type, or the end of a string.
- **INTEGER**
  Matches any positive integer value.
- **IPADDRESS**
  Matches any valid IPv4 address.

Whitespace in STRING Variables

Because whitespace is a valid character in the definition of the STRING variable, always separate multiple STRING tokens with recognizable patterns.

For example, The following ParseMap are *valid* entries:

```
{STRING 1}, {STRING 2}
{STRING 1} {IPADDRESS 2} {STRING 3}
{STRING 1} literal text {STRING 2}
{STRINGNOWS 1} {STRING 2}
```

However, do not have these entries because the resulting regular expression becomes ambiguous:

```
{STRING 1}{STRING 2}
{STRING 2} {STRING 2}
```

Create an Event Format File

Each event code that you specify in a ParseMap file must have a separate Event Format file. When an event is asserted, the text of the Event Format file appears in the Event Log. When creating the Event Format file, keep in mind that most of the information the troubleshooter receives about an event comes from the event message text.

Create the Event Format file using a text editor and place the file in the following directory: *<$SPECROOT>*/SG-Support/CsEvFormat. The Event Format file must be named Event*<xxxxxxxx>* where *<xxxxxxxx>* is the event code that is given to the event in ParseMap file. For example, if you have an event with an event code of 0xffff1A2F, DX NetOps Spectrum uses the Event Format file named Eventffff1A2F.

To make the text of the event message meaningful, you can use the variables assigned in the ParseMap file of the event and the built-in variable available for all Event Format files.

> **NOTE**
> For complete instructions on creating an Event Format file, including the built-in variables that are available, see the Event Configuration section.

Example: Event Format File

Use the following Event format file for the event generated by the ParseMap file.

The IP Address variable is inserted using the data type O (octet) and the variable that is assigned from the ParseMap file, 1. The device name variable is inserted using the data type s (string) and the variable assigned from the ParseMap file, 2. The built-in variables {d "%w- %d %m-, %Y - %T"}, {m}, {t}, and {e} show the date of the event, model name, model type name, and event ID.

```
{d "%w- %d %m-, %Y - %T"} A device {m} of type {t} has reported a problem.
Its ip address is {S 1} and the device name is {S 2}. - (event [{e}])
```

Generating an Alarm Based on the Event

You can specify further processing on the event created in the ParseMap file. You can generate an alarm based on the event, or can use the event as part of an Event Rule. To do this, determine all of the model types that this event could be asserted on and could specify the appropriate event processing in each model type's EventDisp file. If you want the event to be processed the same way for each model type, you can specify the event processing in a global EventDisp file.

If you have specified that an alarm is created based on an event, create a probable cause file that are displayed in the OneClick Console when the alarm is asserted.

> **NOTE**
> For more information about EventDisp and probable cause files, see the Event Configuration section.

Apply the Changes to the SpectroSERVER

To activate the new or updated Event Format and ParseMap files, apply the changes to the SpectroSERVER. This can be done using the Update command found in the Event Configuration Editor, using the command line interface, or by stopping and restarting the SpectroSERVER. See the Event Configuration *section* for more information about each of these methods.

Enable Event Forwarding for Agent Models

You can configure the model of an agent to forward events to models on remote landscapes. Set the SBG_AlertForwardingEnabled (0x3dc000c) attribute for the model to TRUE.

SystemEDGE Application Insight Modules (AIMs)

The SystemEDGE agent provides a plug-in architecture through which it can load Application Insight Modules (AIMs) when it initializes. These AIMs provide an extensible and flexible approach to supporting application-specific semantic knowledge.

DX NetOps Spectrum supports the following AIMs:

- Apache Web Server
- Microsoft IIS
- CA Insight DPM for DB2, Oracle, SQL Server, and Sybase

> **NOTE**
> The SystemEDGE AIMs are available from the Information tab in the Component Detail panel for a selected SystemEDGE agent.

In addition, DX NetOps Spectrum reports alarms which are sent through traps by the Insight AIMs. Each Insight AIM sends out a trap unique to its type, which lets you differentiate between the Insight AIM agent types. Detailed per-alarm information also includes the database name, the alarm type, and the alarm description.

The Insight AIM alarm types vary between agent types and cover a wide range of notifiable conditions. These AIM alarms are no different from other alarms in DX NetOps Spectrum and appear in the same tables and offer the same functionality.

Apache Web Server

The AIM for Apache lets you monitor the health and availability of the Apache web server.

This module works with the SystemEDGE agent to provide the following information:

- The number of "hits" that your web server is receiving. You can track daily volume and set monitor points to watch for unusual traffic levels or denial of service attacks.
- The amount of space that your web log file and web server file are consuming.
- Idle services and active processes. You can gauge how effectively the Apache web server processes monitor idle services, see a warning when the number of idle services is too low, and can monitor the number of active processes.
- The percentage of system resources (CPU and memory) that the Apache web server is using.
- Whether bottlenecks on your web server are related to the CPU, memory, disk, or network.

Microsoft IIS

The AIM for Microsoft IIS provides you with the information you required to monitor the Microsoft IIS application and its use of your system resources.

This module works with the SystemEDGE agent to let you do the following tasks:

- Monitor the availability of Microsoft IIS and its services (Web, FTP, SMTP, and NNTP).
- Automatically restart any service that fails.
- Determine if Microsoft IIS starts to consume significant levels of system resources, including central processing unit (CPU) usage, disk space, and memory.
- Monitor logs for security, system, and application events across the Web, FTP, SMTP, and NNTP services.
- Detect error statistics across the Active Server Pages (ASP), Common Gateway Interface (CGI), and Internet Server Application Program Interface (ISAPI) application extension pages, including Web 404 (page-not-found) errors and ASP script errors.

CA Insight DPM

The Insight AIM provides important information about performance, configuration, availability, and health of DBMS type, for real-time management and long term trending and capacity planning.

The Insight AIM implements a management information base (MIB) that includes variables that are specific to each supported DBMS type. The following DBMS types are supported:

- DB2
- Oracle
- SQL Server
- Sybase

View NSM Agent Information

DX NetOps Spectrum OneClick provides visibility into information that the NSM System agents gather. You can configure process, log file, and file monitoring in the System Resource subview section. Other views provide read-only information that is available from the proprietary MIB values.

You can access NSM agent information in OneClick. This procedure assumes that you have already modeled the NSM agents in your network, either by Discovery or by modeling them manually.

Follow these steps:

1. Select a modeled NSM agent device icon in the Topology tab.
   The Component Detail panel displays the Information tab for the selected NSM agent model.
2. Expand the System Resources subview.
   NSM agent-specific information is displayed.

Trap-to-Alarm Mapping

The CA Unicenter NSM agent management module integrates NSM agent traps into the DX NetOps Spectrum event and alarm processing.

DX NetOps Spectrum processes traps that are sent by NSM agents including System and Performance agents. For each NSM System or Performance agent trap with a state of Warning or Critical received, DX NetOps Spectrum generates an alarm as shown in the following table. When DX NetOps Spectrum receives the related OK trap, DX NetOps Spectrum clears the corresponding alarm.

NSM Trap Received by DX NetOps Spectrum	DX NetOps Spectrum Alarm Generated
Warning Trap	Minor alarm
Critical Trap	Major alarm

Trap processing is based on the NSM agent model types. Each model type processes traps for several agents as outlined in the following table.

Model Type	Processes Traps on Behalf of These Agents
Host_NSMSysUnix	caiUxsA2 caiLogA2 hpxAgent
Host_NSMSysWin	caiWinA3 caiLogA2 caiAdsA2 hpxAgent
Host_NSMv3SysUnix	caiUxOs caiLogA2 hpxAgent
Host_NSMv3SysWin	caiW2kOs caiLogA2 caiAdsA2 hpxAgent

Event Code and Probable Cause File ID Ranges

The following table lists event codes and probable cause file IDs for NSM Agent MIBs.

NSM Agent MIBs	Range of Associated DX NetOps Spectrum Event Codes and Probable Cause Files
caiUxsA2	Event04ef0000 - Event04ef00e9 Prob04ef0002 - Prob04ef00e3
caiWinA3	Event04ef1000 - Event04ef10c7 Prob04ef1002 - Event04ef10c1
caiLogA2	Event04ef2000 - Event04ef2010 Prob04ef2002 - Prob04ef200e
caiAdsA2	Event04ef3000 - Event04ef3042 Prob04ef3002 - Prob04ef303e

hpxAgent	Event04ef4000 - Event04ef4008 Prob04ef4002 - Prob04ef4006
caiUxOs	Event04ef5000 - Event04ef5069 Prob04ef5002 - Prob04ef5067
caiW2kOs	Event04ef6000 - Event04ef6099 Prob04ef6002 - Prob04ef6095

System and Application Monitoring Privileges

This section lists privileges that are related to system and application monitoring for OneClick users.

NOTE
For more information about configuring privileges, see OneClick Administration .

- **System & Application Monitoring**
  Controls access to the System & Application Monitoring privileges. Deselecting this privilege automatically deselects the following privileges:
  - **Manage Rule Sets**
    Allows the user to create a monitoring rule set.
  - **Monitor File Systems**
    Allows the user to create a file system monitoring rule.
  - **Monitor Processes**
    Allows the user to create a process monitoring rule.

IP Routing Manager

The information about the IP Routing Manager is classified down into the following sections.

Introducing IP Routing Manager

Introducing IP Routing Manager

Provides an overview of IPRM Features, Concepts and Architecture, as well as the Route Explorer overview.

IP Routing Manager and Features

DX NetOps Spectrum IP Routing Manager (IPRM) was created as a tool to proactively monitor the state of IP routing protocols. IPRM also assists with troubleshooting failures and performance degradation impacting service delivery. The status of IP routing protocols is critical to the overall health of any environment's network. Additionally, IP Routing Manager helps you monitor and visualize the IP routed path(s) between critical endpoints in the network to ensure data flows over the most desirable and high-performing paths. Understanding the path that data takes is necessary to correlate service assurance alarms to their root cause.

DX NetOps Spectrum IP Routing Manager allows you to discover and view a network's topology by integrating with Route Explorer (REX). Route Explorer (REX) is an appliance-based route analytics solution developed and marketed by Packet Design.

DX NetOps Spectrum IP Routing Manager features include:

- Discovering and visualization of the Layer 3 network
- Providing visualization of Autonomous System (AS) and OSPF-specific hierarchies

  NOTE

  Visualization of BGP devices in IPRM hierarchies is not supported.
- Discovering real-time Layer 3 paths through the network
- Providing dynamic updates to Layer 3 topology and paths
- Performing dynamic path monitoring:
  - Path hop details
  - Forward and reverse path support
  - ECMP support
  - Path change events/alarms
- Monitoring REX trap events/alarms
- Providing visualization of IP Subnets
- Providing bulk modeling of unmanaged devices
- Performing visualization of Layer 3 link information and custom icons
- Integrating Layer 3 topology view with other DX NetOps Spectrum's topology views (for example Universe)

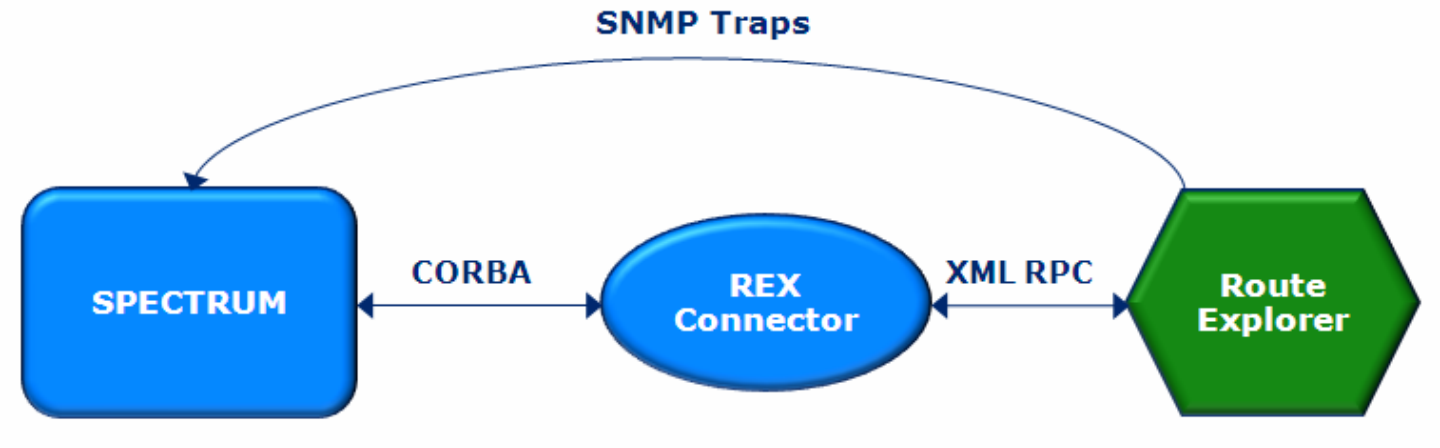
The product supports both fault tolerant and distributed DX NetOps Spectrum deployments, with all user configuration and interaction being performed on the main location server (MLS) system.

In addition to REX appliance, IP Routing Manager will also work with IPRM Route Recorder and OEMed versions of REX - RAMS from Hewlett Packard and Route Insight Manager from Juniper. IPRM Route Recorder (IPRM-RR) is a special version of REX that does not expose its GUI, but acts as instrumentation for IPRM delivering real-time routing information via an XML API. Packet Design will make this product available to DX NetOps Spectrum customers wishing to take advantage of IPRM.

IP Routing Manager Architecture and Concepts

IP Routing Manager utilizes the following architecture. The REX Connector is a separate process, and is the connection between DX NetOps Spectrum and Route Explorer. REX Connector utilizes the REX API to communicate with Route Explorer.

The Route Explorer software can be configured to notify clients of routing events via SNMP traps. Each one indicates some sort of Layer 3 topology change or information. These traps sent from Route Explorer will be translated into DX NetOps Spectrum events (and alarms when appropriate) and will be generated on the corresponding device model.

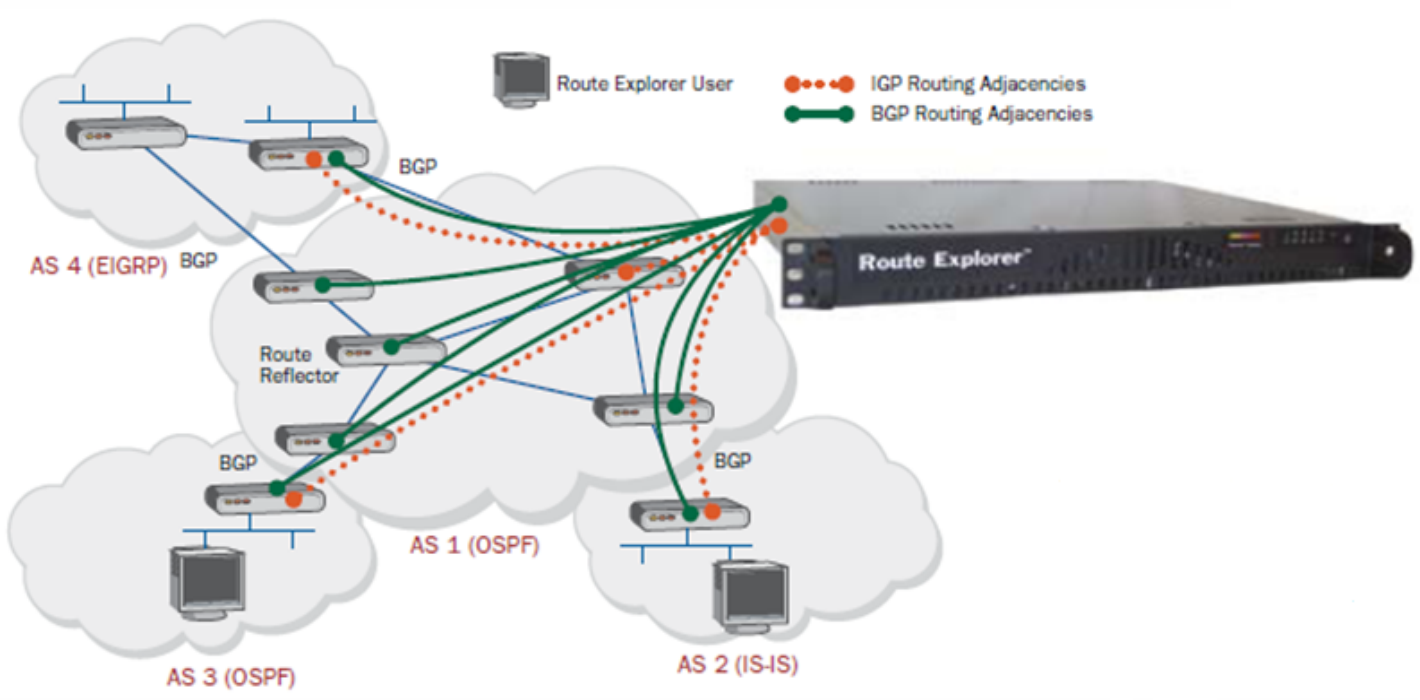


Some important concepts to understand to utilize IPRM fully:

- **IP Subnet** - An IP Subnet (REX refers to them as prefixes). IPRM distinguishes between two logical types of subnets:
  - **User Subnet** - IP Subnet attached to at least one router (also known as a gateway) and typically serving a number of hosts and devices. IPRM defaults to hiding these icons from the topology view.
  - **Infrastructure Subnet** - IP Subnet interconnecting two or more routers (REX shows them in its topology GUI as pseudo-nodes). In cases when only two routers are attached to such a subnet, IPRM can be configured to represent these as links in topology view, otherwise these are represented as IP Subnet icons.
- **PointToPoint Link** - IPRM's representation of Layer 3 Point-to-Point Links between two routers.
- **Managed Path** - End-to-end Layer 3 path across managed infrastructure configured for proactive assurance monitoring.
- **Autonomous System (AS)** - Part of the Layer 3 topology (a collection of routers and IP subnets), organized by REX into an Administrative Domain. IPRM also arranges OSPF protocol models within corresponding AS models.

Route Explorer Overview

A single Route Explorer can concurrently monitor and analyze complex IP networks which may have multiple routing protocols (OSPF, IS-IS, EIGRP, and BGP) and span multiple autonomous systems. A distributed architecture involving multiple appliances may be employed to enhance management continuity in the event of a network failure while supporting multi-tiered or regionalized management domains.



The routing analytics appliance is developed and marketed by CA's Technology Partner - Packet Design. This appliance operates by passively monitoring the routing protocols exchanges between routers. It 'announces' itself as if it was a router, but does not advertise any prefixes, so no traffic flows to/through it. Therefore, it's neither a bottleneck nor a failure point.

IP addresses are assigned and adjacencies are set up with a small number of routers:

- REX requires an adjacency with one router in each OSPF area or IS-IS level. Once adjacencies are up, REX begins monitoring the protocols and is able to present a complete, network-wide map within minutes.
- Each adjacency may be over a physical connection (local) or GRE tunnel/VLAN (remote)

The only messages Route Explorer sends out to its neighbors are periodic 'Hello' messages to maintain adjacencies. Therefore, virtually zero load is placed on the real routers or the network, both during discovery and ongoing monitoring. REX can scale to manage the largest networks in the world (7,000 routers/8 million routes with a single box). In 99%+ of cases a single appliance is adequate.

REX has two interfaces, a web-based user interface for REX administration purposes and a graphical user interface for the end-user. IPRM-RR will have have the REX GUI disabled, to allow for easy DX NetOps Spectrum OneClick utilization.

Route Explorer Administration

The following guidelines are helpful to Route Explorer Administration:

- For DX NetOps Spectrum and REX firmware version compatibility, see the Integration Compatibility section.

  NOTE
  Additional future versions of REX firmware may be supported by IP Routing Manager v10.1 or higher, based upon API compatibility.
- In a distributed REX installation, for example, a modeling engine and one or more pure route recorder(s); you should configure IP Routing Manager's REX integration settings to connect to the modeling engine.
- Only one REX connection is supported.

REX Connector Process

IP Routing Manager communicates with REX via a process called the REX Connector. This process is managed by process daemon and started up when the SpectroSERVER is started. Because this process can use large amounts of CPU processing when Layer 3 topology discoveries are done, it is ideal to run DX NetOps Spectrum on a multi-CPU system.

- Integrates DX NetOps Spectrum with Route Explorer
- External Java-based application
- Implements a new CORBA interface allowing DX NetOps Spectrum to request Layer 3 topology and path information
- Translates DX NetOps Spectrum's CORBA-based requests into Java/XML-Remote Procedure Calls to access REX API
- Performs topology and path polling for change detection, notifies clients of changes via callbacks
- Supports configuration of Route Explorer alerts in the OneClick client
- Utilizes ports 14002 and 14006 to communicate with the SpectroSERVER and Naming Service
- Utilizes port 2000 to communicate with Route Explorer

  NOTE
  The REX Connector should run on separate CPU from SpectroSERVER if possible.

  WARNING
  IP Routing Manager communicates with the REX API via XML-RPC over a TCP connection. This TCP connection is not encrypted. This should be realized for data passing over the tunnel. However, the REX API password is encrypted in DX NetOps Spectrum.

Autonomous Systems

Router Explorer organizes protocol instances (for example BGP and OSPF) into groups called 'Administrative Domains'. You can create a hierarchical structure of Administrative Domains to arrange the protocol instances in a way that is coherent with your IT infrastructure. The IP Routing Manager data model provides the Autonomous System model as a

logical container for protocol models. To provide continuity, as well as synchronization between the two data models, IP Routing Manager uses the Administrative Domain path as the name of the corresponding Autonomous System model.

For example, the following text is a sample topology name as provided by the REX API:

```
CA.MySubDomain.OSPF/Backbone
```

IP Routing Manager takes the text that precedes the protocol name, in this case OSPF, and uses it as the name of the Autonomous System model: 'CA.MySubDomain'.

You can set the model name on the Autonomous System model according to your needs.

Using the REX Administration Features

Refer to Route Explorer Administration section for instructions on how to utilize the REX Administration web-based user interface.

Installing and Configuring IP Routing Manager

Installing IP Routing Manager

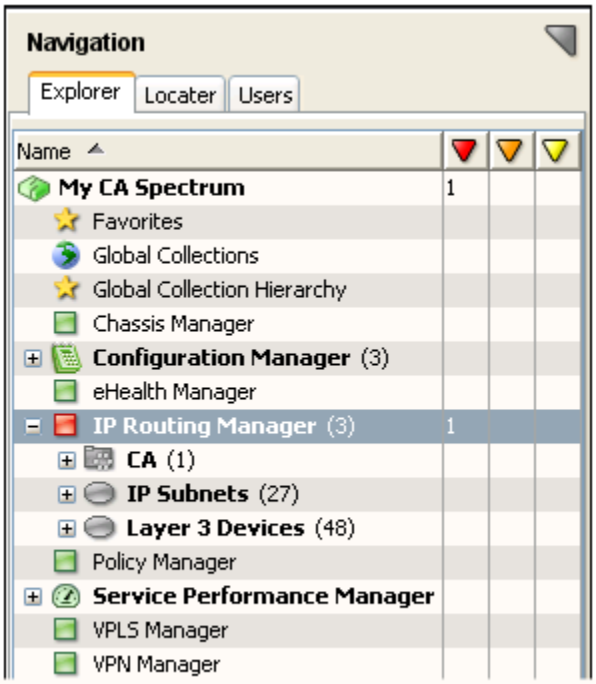
To install IP Routing Manager

1. Run the DX NetOps Spectrum installation program to install DX NetOps Spectrum and OneClick. See Install DX NetOps Spectrum for details.
2. Bring up the DX NetOps Spectrum Control Panel, click the Start SpectroSERVER button.

After installing this product, you will see an IP Routing Manager entry at the 'global' level in the Navigation Panel of the OneClick Console; the IP Routing Manager model of the Main Location Server (MLS) will be shown. This IP Routing Manager view is the central point for all Layer 3 topology-related activities: viewing Layer 3 topology and paths, configuration tasks, and general information gathering.

When you have discovered the Layer 3 topology, all Layer 3 topology and path data models representing the logical protocol hierarchy will be listed underneath the IP Routing Manager icon in the Navigation Panel. All Layer 3 paths you have created will be listed underneath the IP Routing Manager icon in the corresponding folder icon.

This panel also displays the entire Layer 3 topology across *all* DX NetOps Spectrum landscapes. It will also display all critical paths across DX NetOps Spectrum landscapes.



NOTE
IP Routing Manager *can* generate events and alarms even before it is setup or configured. This may happen because you previously configured Route Explorer to send traps to DX NetOps Spectrum; when DX NetOps Spectrum receives these traps, IP Routing Manager will generate the appropriate events/alarms regardless of whether or not it is setup and connected to Route Explorer.

Connecting to Route Explorer

To connect to Route Explorer

1. Navigate to the Component Details Panel of the IP Routing Manager model.
2. Select the Information Tab, and open the Route Explorer Integration sub-view.
3. Enter the Route Explorer machine's host name, the query password, and the Administrative Domain name you'd like to use.
   You can also choose to enter a new heartbeat value. The heartbeat value determines how often IP Routing Manager pings Route Explorer to make sure it's still up and functioning.
4. Once these values are configured, press the Connect button.
   IP Routing Manager will then attempt to connect to Route Explorer, and you will be notified of the success/failure of the operation.

   NOTE
   You will not be able to connect unless you authorize the SpectroSERVER to be able to connect with the REX machine through the 'Queries' Admin Page in the REX Web UI.

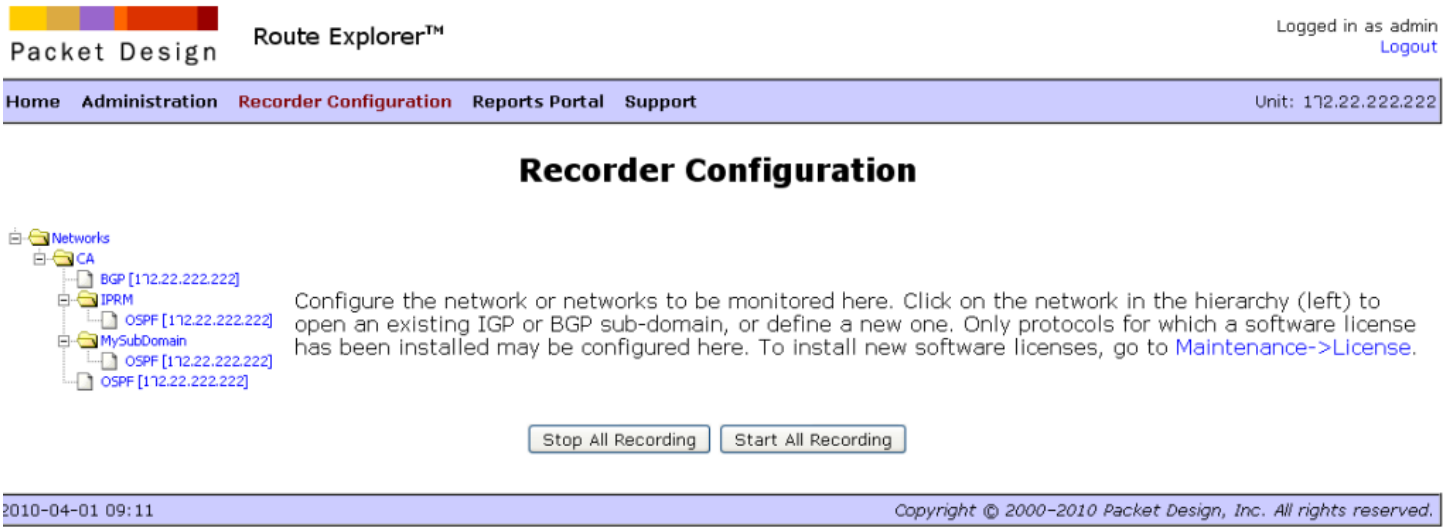
   The Connection Status field displays the status of IP Routing Manager's connection to Route Explorer. Once connected, the Connect button will become a Disconnect button, which you can use to disconnect from Route Explorer when desired.

   NOTE
   To change any of these REX configuration settings, first disconnect from REX, make the required changes, and then re-connect for the changes to take effect.

WARNING
If Route Explorer stops recording information on particular protocols, IP Routing Manager will notify you through an event and alarm.

Route Explorer Configuration Considerations

When configuring the 'Administrative Domain' setting in the REX Configuration subview; you should refer to the 'Recorder Configuration' in the Route Explorer web interface. The 'Recorder Configuration' is accessed by logging into the web interface and clicking on the 'Recorder Configuration' in the menu at the top. In the case of a distributed REX deployment (where there are multiple Route Explorer units all replicating their data to a single Modeling Engine unit), refer to the 'Recorder Configuration' on the Modeling Engine's web interface.



The 'Recorder Configuration' page displays a hierarchy containing the Administrative Domains (represented as folders) and the protocol instances (represented as sheets of paper). The 'Networks' folder at the top of the hierarchy is the root folder and cannot be used when configuring IPRM. The next folder down is the top-level administrative domain. Enter the name of the top-level administrative domain into the Route Explorer Configuration sub-view in IPRM.

Upgrading IP Routing Manager

For sites who have configured/used IP Routing Manager in Spectrum 9.4 and are upgrading to a newer release (such as Spectrum 9.4.1), you will need to perform the following steps after the upgrade has finished installing.

To upgrade IP Routing Manager:

1. From the OneClick, Explorer option, select 'IP Routing Manager'.
2. Select the Contents, Information tab.
3. Expand the 'Route Explorer Integration' option and select Configuration.
4. For the 'Query Password' field, select the 'set' link and set the appropriate password value; and select Save when finished.

   WARNING
   This step needs to be performed even if the ‘Query Password' has not changed. This will encrypt the password value and is *required* in order for IP Routing Manager to be able to connect to the Packet Design appliance.

1. If a fault tolerant SpectroSERVER is being used, then an Online Backup should be performed after re-setting the query password so that the attribute change is propagated to the backup SpectroSERVER.

2. After setting the Query Password, you can verify a connection from IP Routing Manager to Route Explorer by selecting Connect.
3. After successfully connecting to Route Explorer, perform a layer 3 topology discovery by selecting the 'Discover' option in the IP Routing Manager configuration subview. You do not need to clear the layer 3 topology first. (This step ensures that certain attributes are updated to support some of the new features introduced in IPRM 9.2.1.)

Alarm and Event Configuration

DX NetOps Spectrum has the ability to manage the alert configurations on the Packet Design appliance via a subview in OneClick. Expanding the Route Explorer Integration, Alerts subview reveals the alert configuration settings. Modifying the alert configuration settings in this view will automatically create the appropriate corresponding alert configurations on the Packet Design appliance to which IP Routing Manager is connected.

> **NOTE**
> In order to configure alerts, IP Routing Manager must be connected to an appliance and 'Topology and Managed Path Monitoring' must be set to either 'Initiated by Traps Only' or 'Traps and Scheduled Polling'.

Traps Sent to MLS

Since the IP Routing Manager product will span all SpectroSERVERs in a DSS environment, and Route Explorer will only be communicating with the MLS landscape, the MLS will forward traps sent from Route Explorer to the correct SS in a DSS environment so events and alarms are generated on the correct device models.

In DX NetOps Spectrum, you must use the Route Explorer GUI in order to configure alerts. Using the IPRM you can configure four types of alerts (adjacency state, router state, peering state and prefix state) through OneClick.

Note: See the Route Explorer Administration for assistance in configuring the traps.

Topology Change Trap Details

IP Routing Manager supports the following *Router State Change Traps*:

- **Router Connected**
  This trap is sent when a router begins participating in the routing protocol. The router may have just come up. This is a notification that a full adjacency has been established between the router and one of its neighbors.
- **Router Isolated**
  This trap is sent when a router becomes isolated from the network. The router may not actually be down, but isolated because of another outage.
- **Router State Flap**
  This trap is sent when the router's connected/isolated flap count exceeds a threshold over a given duration. DX NetOps Spectrum will include the threshold values in the event generated.

IP Routing Manager supports the following *Adjacency State Change Traps*:

- **Adjacency State Up**
  This trap is sent when a IP Subnet protocol adjacency comes up. This adjacency can be between two routers (and each one's interfaces), or between a router and a pseudo-node (this is how REX refers to a IP Subnet LAN).
- **Adjacency State Down**
  Same information as above, but the adjacency goes down.
- **Adjacency State Flap**
  This trap is sent to indicate adjacency flapping. DX NetOps Spectrum will include the threshold values in the event generated.

IP Routing Manager supports the following *Path Change Traps*:

- **Path Change**
  This trap is sent when REX detects any change in the IP Subnet path(s) between the given end points, including number of hops, path cost (metric), intermediate hops, etc. This trap does not directly result in an event or alarm being generated. DX NetOps Spectrum uses this trap as a way to detect topology/path changes, and the needed event or alarm will be generated when the topology/path changes are modeled.

IP Routing Manager supports the following *Prefix State Change Traps*:

- **Prefix State Up**
  This trap is sent when a router interface is turned on (enabled). This trap announces that the interface's prefix has come online. This trap is also sent for routes that are redistributed from another routing protocol via a router.
- **Prefix State Down**
  This trap is sent when a router goes away, and all of its prefixes are removed. A router may also remove a prefix that it is advertising (premature withdrawal).
- **Prefix State Flap**
  Same handling as for other 'flap' traps.

Path Change

You can configure each monitored path to generate an event or an alarm of a specified severity when the path changes. Changes detected include: path cost, number of ECMPs, and actual path hops traversed.

To accomplish this, select a specific Layer 3 Path model's Information Tab, and configure the relevant settings.

WARNING
Once a path change alarm is created for a given managed path, any subsequent changes will not result in new alarms but will instead be appended as events to the existing outstanding alarm.

NOTE
You can utilize the Layer 3 Path History feature to help debug why the path may have changed or been lost.

Path Loss

You can configure each monitored path to generate an event or an alarm of a specified severity when the path is lost. A path is determined to be 'lost' if there does not exist at least one complete forward and return path, if applicable, where both the source and destination nodes are matched with the requested source and destinations.

To accomplish this, select a specific Layer 3 Task model's Information Tab, and configure the relevant settings.

Self Monitoring

If IP Routing Manager loses the connection to REX, or there is a configuration problem with the integration, a red alarm is asserted on the IP Routing Manager model notifying the user of the situation.

Fault Tolerance Support

IP Routing Manager fully supports fault-tolerant SpectroSERVER setup. The REX Connector runs on primary and secondary landscapes, both connected to REX. Secondary landscapes periodically poll Route Explorer to verify that it is still available, but they do not detect any topology or path changes. On failover, secondary SpectroSERVER resumes normal REX communication and polling.

In order for REX traps to be handled correctly, you need to configure REX to send traps to both the primary and secondary SpectroSERVERs. Optionally, a tool like TrapExploder could be used which allows you to configure REX to send traps to a single IP address, and the TrapExploder can be configured to replicate the traps it receives to multiple recipients.

When a secondary SpectroSERVER becomes active it automatically configures alerts on the Route Explorer appliance so that the desired traps are sent to the secondary SpectroSERVER. In order for this to function properly, you must perform an OnlineBackup after you have configured the alerts on the primary SpectroSERVER.

NOTE
If the secondary SpectroSERVER is in 'warm' or 'cold' standby mode it will not process any traps; if it is in 'hot' standby mode it will process traps.

Using IP Routing Manager

Here are some of the ways of using the IP Routing Manager:

- Initiating Layer 3 Topology Discovery
- Additional Discovery Explanation
- Layer 3 Topology Navigation
- Using the Layer 3 Path Discovery Feature

Initiating Layer 3 Topology Discovery

A topological model of your Layer 3 topology will be built and maintained inside the SpectroSERVER. The Layer 3 topology, including Autonomous System and OSPF protocol details, will be graphically displayed to you using traditional DX NetOps Spectrum device icons, new LAN icons, and pipes. All routers from all landscapes in the DSS will be displayed in this single view.

The entire Layer 3 routing hierarchy will be displayed in the OneClick Navigation Panel under the IP Routing Manager entry. Sub-hierarchies will include an entry for each Autonomous System (AS) that is being managed. Each AS will be further broken down into OSPF protocol details (if applicable). You will be able to expand the hierarchy to drill down and get detailed information about the IP Subnet topology.

WARNING
IP Routing Manager prevents creating duplicate Layer 3 paths. Duplicates are determined by source IPs, destination IPs, and path names.

To initiate Layer 3 topology discovery

1. Navigate to the Component Details Panel of the IP Routing Manager model.
2. Select the Information Tab, and open the Configuration sub-view.
   The Layer 3 Topology view contains buttons to discover and model the Layer 3 topology.
   Use the Topology and Managed Path Monitoring field to configure how IP Routing Manager should monitor the Layer 3 topology and paths. The various values supported include:
   - Traps and Scheduled Polling - REX Connector polls for changes, and SS responds to REX change traps
   - Initiated by Traps Only - No REX Connector polling, but SS does respond to REX change traps
   - Off - No changes are detected. No REX Connector polling and no trap handling. You must manually initiate re-discovery.

   If the Traps and Scheduled Polling option is used, you should set the value of Scheduled Polling Interval (s) to the desired value. When changes are detected, IP Routing Manager will dynamically update the Layer 3 topology with new and removed routers and links.
   See the '*Without GUI component, how will I be able to configure IPRM-RR traps?*' question within the Frequently Asked Questions section for additional information.
3. Click the Discover button to begin the Layer 3 topology discovery and modeling process.

Depending on the size of the Route Explorer database you've connected to, this process could take many minutes to complete. Once finished, the complete and current Layer 3 topology will be saved in the DX NetOps Spectrum database.

NOTE
The Last Topology Update field will also contain the date and time of the discovery. The Last Topology Update field is updated every time DX NetOps Spectrum models a topology change.

NOTE
You can click the Discover button at any time to have IP Routing Manager update (or re-discover) the complete Layer 3 topology, and thus stay in sync with Route Explorer. You can also press the Clear button to completely remove all Layer 3 topology modeling (including Layer 3 paths that may have been discovered).

User subnets are discovered and modeled when you invoke a Layer 3 discovery or on dynamic updates. They are differentiated from Infrastructure subnets by a new attribute field in the models' Information Tab. You can choose whether or not User subnets are displayed in the Layer 3 topology view by setting the 'Display User IP Subnets' value in the IP Routing Manager Configuration sub-view.

Additional Discovery Explanation

The following items are important to consider in OneClick Integration:

- IP Routing Manager gathers data from all the distributed landscapes that you have configured and runs at a global level, above any landscape, in the Navigation Panel.
- When you deploy DX NetOps Spectrum in a distributed environment, you have multiple landscapes. You must assign one as a main location server. This is where all Layer 3 modeling is stored and then displayed at a global level. All the other landscapes communicate with this main landscape location.

Security Impact

The user privileges can be found in the Component Details panel of the selected user model in the Users tab. IP Routing Manager's privilege group contains the following values:

- **IPRM hierarchy**
  Allows a user to see the IPRM hierarchy.
- **Manage IP Subnet Paths**
  Enables/disables the path creation menu items and the Layer 3 Path configuration sub-view.
- **Configure IPRM**
  Shows/hides the Configuration sub-view on the IPRM model.
- **Configure Route Explorer Integrations**
  Shows/hides Route Explorer Integrations sub-view on the IPRM model.

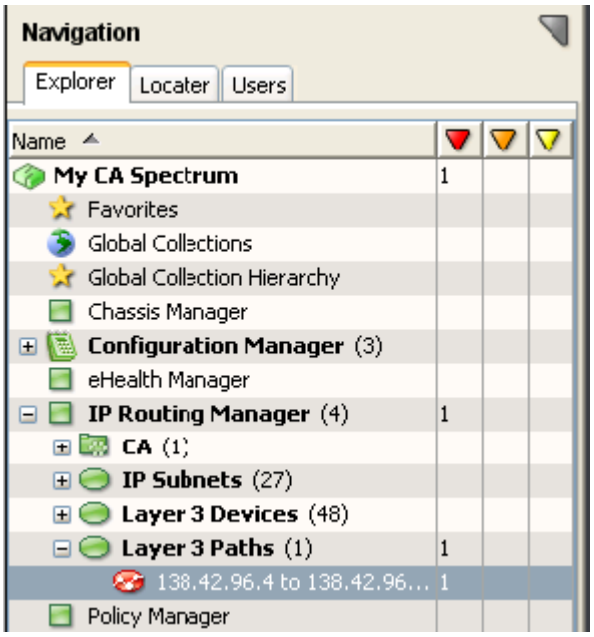
Layer 3 Topology Navigation

The following areas are important when utilizing Layer 3 Topology Navigation.

Navigation Panel

The entire Layer 3 routing hierarchy is displayed in the OneClick Navigation Panel under the IP Routing Manager entry. Sub-hierarchies include an entry for each Autonomous System that is being managed. Each Autonomous System is further broken down into OSPF protocol details (if applicable). You are able to expand the hierarchy to drill down and get detailed information about the Layer 3 topology.

A Layer 3 Devices container and an IP Subnets container exist in the Navigation Panel, allowing you to quickly find and spotlight any IP Subnet device or LAN currently shown in the topology (e.g. user LANs and LANs connected to only two routers are hidden by default, and are not listed in the IP Subnet container).



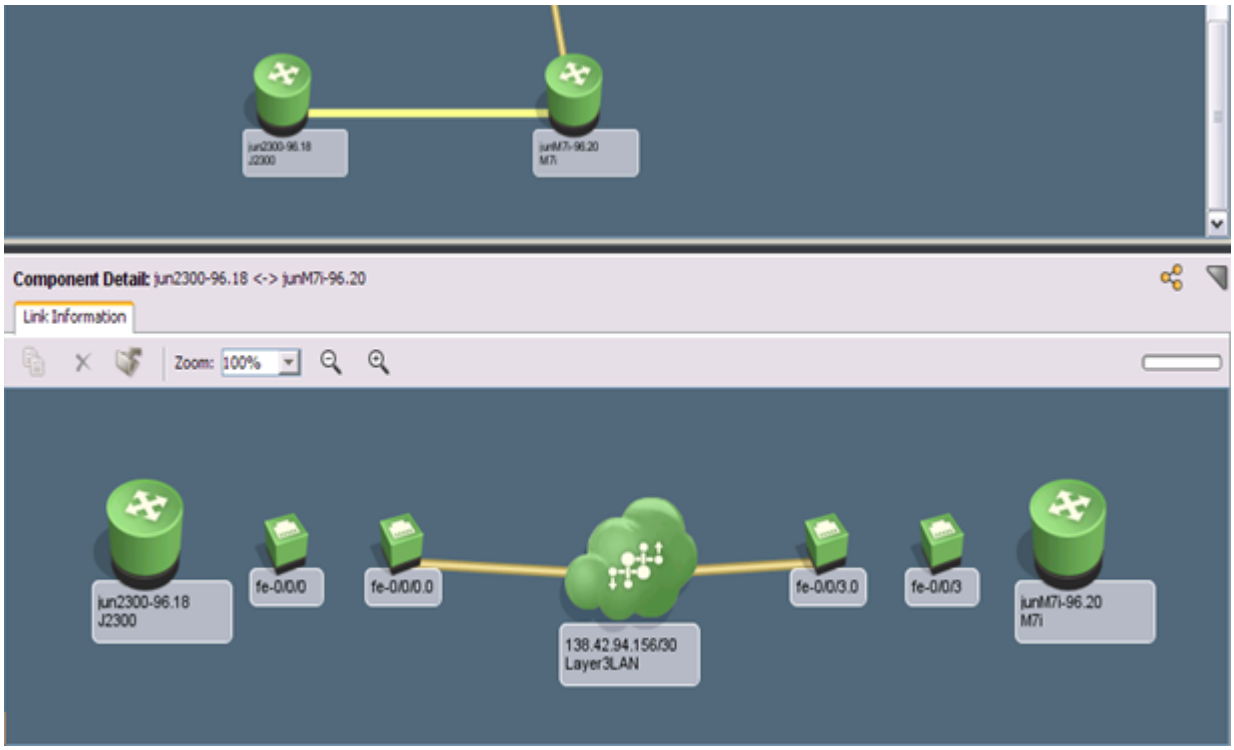
Topology Tab

A topological model of your environment's Layer 3 topology is built and maintained inside the SpectroSERVER. When the Topology Tab is selected, the Layer 3 topology, including Autonomous System and OSPF protocol details, is graphically displayed using traditional DX NetOps Spectrum device icons, new IP Subnet icons, and pipes. All routers from all landscapes in the DSS are displayed in a single topology view. A standard information view is provided for IP Subnets.

Additionally, a Link Information view is provided to show the port-level connectivity between routers and subnets, as well as show the hidden IPSubnet or PointToPointLink model for a given connection.

The Link Information tab has been enhanced with the concept of a middle model. If you select a link between two routers (a link which represents a IPSubnet or a PointToPointLink) in the Layer 3 topology, the Link Information view will show the port-level connections (if available) between the two routers, as well as the (hidden) IPSubnet or PointToPointLink model between the two routers. If you select a link between a router/UnmanagedDevice and a IPSubnet, the Link Information view will show the port-level connections between the router and that LAN.

WARNING
Interface connectivity will not be available if the connected router is currently modeled as an 'UnmanagedDevice'.



By default, the Layer 3 topology view will only show an actual icon for the IPSubnet when there are more than two routers connected to it. IP Routing Manager compresses the LAN icons connected to only two routers into a simple pipe. You could optionally turn on display of all IPSubnet icons (via the 'Display IP Subnet Between Two Routers' configuration option on the IP Routing Manager model). The ability to display IPSubnet icons allows you to refine the Layer 3 topology view to best suit your needs. PointToPointLink models are always represented as a simple pipe between two routers.

Mouse-over pop-ups are supported. Any time the mouse hovers over an IP Subnet icon, or any pipe connected to an IP Subnet, simple information about the IP Subnets and connected interfaces will be displayed.

List Tab

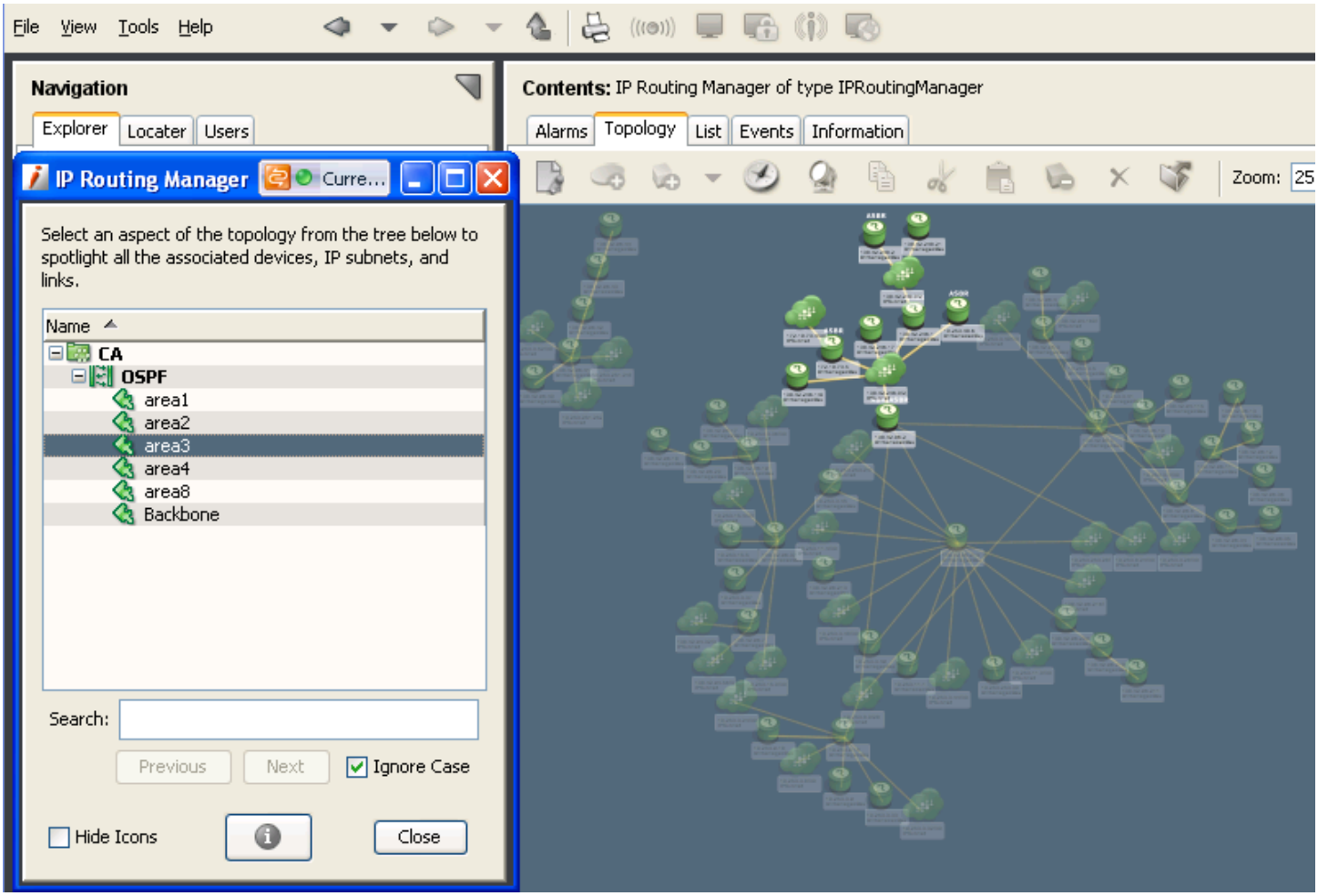
The List Tab of the IP Routing Manager model contains information about all routers and subnets that exist in the IP Subnet topology. The contents of the List Tab will be filtered based on what is selected in the Navigation Panel. Let's say you select an OSPF area. In the Topology Tab, all routers, subnets, and pipes involved in the OSPF area are spotlighted, and the rest are grayed out. The List Tab will then only display the models that are spotlighted. The List Tab contains a new Layer 3 Node Type column which displays the role each router is playing in the Layer 3 topology. Highlighting each model in the List Tab allows you to explore the Component Details of each model. IPSubnet and PointToPointLink models display AS and OSPF-specific data, as well as the designated router IP.

Spotlighting

A new navigation-based spotlighting capability has been introduced. Selecting (highlighting) a Layer 3 path model in the Navigation Panel will cause that particular path to be spotlighted in the IP Routing Manager's topology view. This Navigation Panel feature can be thought of as a built-in spotlighting feature that you can access *without* clicking the spotlight icon in the top menu and opening a separate dialog box. It's built right in.

This product also supports the traditional toolbar-based spotlighting dialog box, similar to VRRP and VLANs, so that you can un-dock the topology view and still use the spotlighting feature. When opened, the dialog box will contain a Layer 3 hierarchy exactly like the one displayed in the Navigation Panel. Toolbar-based spotlighting takes precedence over Navigation-based spotlighting.

For example, when you select an OSPF area in the Navigation Panel, all routers, subnets, and links which are members of the selected OSPF area will be spotlighted. The rest of the topology view will be grayed out. Spotlighted router icons will display protocol-specific annotations such as ABR and ASBR and the like.



Unmanaged Devices

All routers that Route Explorer knows about may not be modeled in the DX NetOps Spectrum DSS environment. When this happens, an UnmanagedDevice model is created, and its NetworkAddress attribute is filled in with the IP address of the router. UnmanagedDevice models are displayed in the Layer 3 topology just like 'full' DX NetOps Spectrum router models.

To model an UnmanagedDevice as a 'full' DX NetOps Spectrum router model, you have two options:

1. Right-click the icon and select 'Manage selected device…'. The Create by IP dialog opens, which lets you discover the device and replace the new model in all DX NetOps Spectrum views where the current UnmanagedDevice model exists.
2. Select multiple UnmanagedDevice models in the IP Routing Manager List Tab. Right-click and select 'Manage selected devices ...' to launch an AutoDiscovery. Auto discovery automatically populates the IP range list with the IP addresses of all selected models.

Both methods of discovery allow you to select which landscape (if in a DSS) to place the newly discovered device models.

In addition to the Model Unmanaged Device menu option, you can use the Discovery application to model your unmanaged devices. Every time AutoDiscovery is run, it will always attempt to replace any UnmanagedDevice models with a newly discovered device in its results set.

The existing context-launch functionality of Discovery is leveraged to allow users to select one or more UnmanagedDevice models in the topology tab and the list tab, and then launch Discovery from the toolbar or right-click menu. Discovery uses the IP information from the selected UnmanagedDevice models to create a new Discovery configuration. The Discovery configuration can then be used to discover the unmanaged devices and create device-specific models to replace the UnmanagedDevice models.

During the final stages of a modeling process (within the Discovery app), the DX NetOps Spectrum searches all landscapes for UnmanagedDevice models. It then compares the IP addresses of the newly modeled devices with the IP addresses of the UnmangedDevice models. If an address from a new model matches the address from the UnmanagedDevice model, the UnmanagedDevice model is replaced with the new device-specific model (even if the new model is a Pingable). In the IPRM topology view, the UnmanagedDevice icon is replaced with the icon of the new model.

Jump to Feature

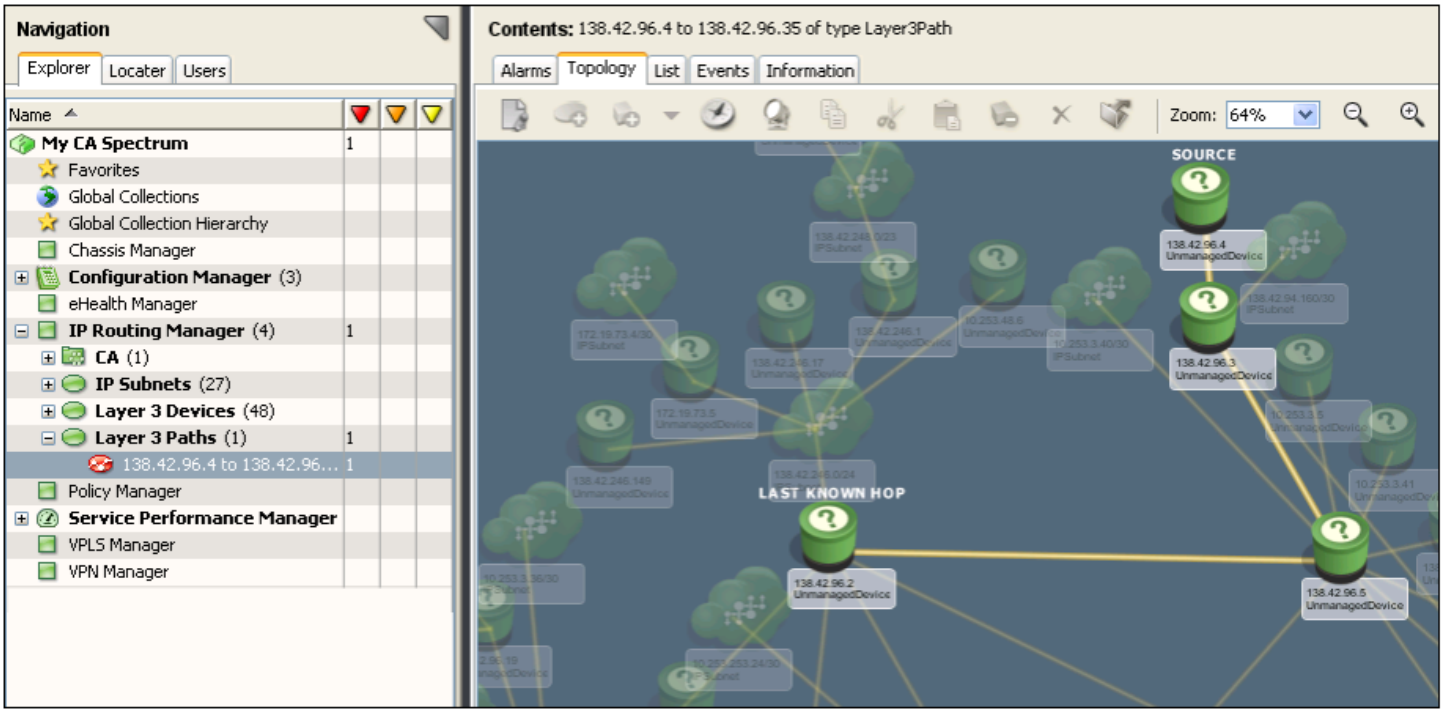
You are able to right-click on any router in the Layer 3 topology and select Location, Universe. This enables you to view the traditional DX NetOps Spectrum topology/universe view of the container model in which the router model exists. If the router model exists in a remote landscape in a DSS environment, it supports this as well. The router model is highlighted in the view. This feature is the key to integrating the new Layer 3 topology with the traditional DX NetOps Spectrum modeling and viewing hierarchy (Layer 2).

You are also able to right-click on any router in a traditional DX NetOps Spectrum topology or any other OneClick view and select Location, IP Routing Manager. This enables you to jump to the new Layer 3 topology view of the MLS and highlights the router in the view.

> **WARNING**
> The most efficient way to search when you have a long list of models in the Explorer is to find the model using a Locater search, select the found model, right click and select Location, IP Routing Manager.

Providing End to End Layer 3 Path Visualization

IP Routing Manager provides end-to-end Layer 3 path visualization. This provides you with hop-by-hop paths and alerting on critical paths.



Using the Layer 3 Path Discovery Feature

You have the ability to view and monitor the current Layer 3 path(s) between any two IP-based endpoints within REX's visibility on demand using OneClick.

Using the Layer 3 Path Discovery Feature

1. Select the start point in the IP Routing Manager Topology View; right-click on the node and select 'Utilities, Select As Layer 3 Path Source'.
2. Select the end point in the IP Routing Manager Topology View; right-click on the node and select 'Utilities, Create Layer 3 Path'.
   This will bring up up a new dialog for you to enter additional information.
   A name for the Layer 3 path is automatically created by DX NetOps Spectrum, but you can assign any name in the dialog box. When manually entering path destination IP address information, you may choose a single device/port IP address, or a subnet IP/mask pair. The path source must be a single IP address of a router. When entering an IP subnet for the path destination, dotted-decimal or CIDR suffix notation may be used for the mask.

   NOTE
   Duplicate Layer 3 paths cannot be created. Duplicates are determined by source/destination IPs, as well as path names.

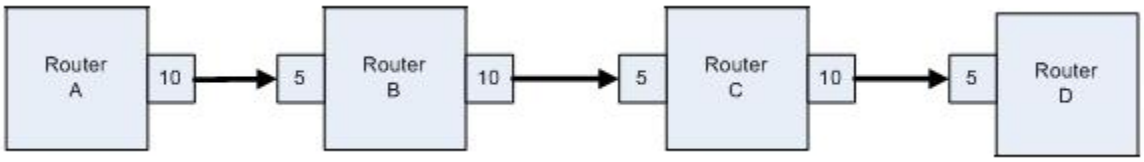
   You also have the option to discover only forward paths between the source and destination, or both forward and return paths. Discovering return paths will help you discover asymmetric paths, as well as add to IP Routing Manager's ability to detect when a path is down.
3. Click the OK button in the Create Layer 3 Path dialog to initiate path discovery and modeling.
   When finished, the new Layer 3 path model is displayed in the Navigation Panel under a new Layer 3 Paths container in the IP Routing Manager hierarchy.

Important Considerations

There is no separate monitoring mode for paths. All paths will be monitored for changes according to the global Topology and Managed Path Monitoring value.

When you trace a path to a loopback IP on a router, there's a good chance that REX will include the last internal hop in its cost calculation and hops returned. Since IP Routing Manager has no way to graphically display this internal hop in the topology view, the REX Connector prunes it from the data returned, and IP Routing Manager doesn't show it in the topology view or the path hop details table. The cost of the internal hop is included in the overall path cost, however.

The cost of a path is determined by adding the routing metrics assigned to each of the egress interfaces of each hop in the path. See the attached picture for an example. The path is from the loopback IP address of router A to the loopback IP address of router D. The path data REX sends will contain a path cost of 31. This is the sum of all egress interface metrics, plus the internal hop of router D. Since IPRM cannot visualize the internal hop in router D, it doesn't include that hop in the Path Hop Details table.



NOTE
IP Routing Manager currently only supports creation of Layer 3 paths whose endpoints are within REX's visibility. If you enter an IP/subnet that is outside this visibility, then the path may not be complete, and a path lost alarm could be generated.

Layer 3 Path Viewing

Selecting a Layer 3 path model in the Navigation Panel allows you to view data about the path. The Component Detail panel contains source and destination IP information, the cost of the path, the number of hops, and the date/time it was discovered or last updated.

If the path has more than one equal-cost multi-path (ECMP), then DX NetOps Spectrum creates separate Layer 3 ECMP models, and displays them as children of the Layer3Path model in the Navigation Panel. When you highlight a Layer 3 Path in the Navigation panel, the Layer 3 topology will spotlight all of the routers, subnets and connections which are included by all of the associated ECMP's. When you highlight a specific Layer 3 ECMP in the hierarchy, the topology will spotlight only the routers, subnets and connections which represent that particular ECMP.

NOTE
Alarms are never generated on ECMP models, only the Layer 3 Path model. Thus, an ECMP model will always be green.

Select a Layer 3 Path or ECMP to view the Path Hop Details table, which contains dynamic, comprehensive information for each hop in the path. Details include source/destination routers, ingress/egress interfaces, link type and prefix, hop cost, and protocol.

NOTE
Path Hop Details are dynamic only when 'Topology and Managed Path Monitoring' is set to 'Traps and Scheduled Polling' or 'Initiated by Traps Only'. For more information, see Initiating Layer 3 Topology Discovery.

You can destroy Layer 3 Path models by simply right-clicking on them in the OneClick Navigation Panel and selecting Delete.

In REX, pseudonodes represent IP Subnet IP subnets/LANs. The REX GUI displays them as nodes in the topology with routers connected to them. REX considers pseudonodes as distinct hops in a Layer 3 path. For example, for a path between 3 routers (A, B, and C), there will be two IP subnets (X and Y) with one between each router. REX would consider this path to have 4 hops: A -> X, X -> B, B -> Y, Y -> C.

IPRM uses pseudonodes to create IPSubnet models, and displays them similarly to how REX displays them. But in Layer 3 Paths, IPRM treats pseudonodes (IPSubnets) as something that a hop transitions through. So, for the path above, IPRM says the path has 2 hops: A -> X -> B, B -> Y -> C. A subnet is not really a hop endpoint, but a portion of the network that the data travels through to get to the next hop.

> **NOTE**
> In Route Explorer version 9.3.16, pseudonodes are no longer recognized as distinct hops in a path. Instead, pseudonodes are now treated similarly to how IP Routing Manager recognizes them.

Layer 3 Path History

The Layer 3 Path history feature allows the user to view changes in the Layer 3 path between two end points over a specified period of time.

Using the Layer 3 Path History Feature

1. Select the start point in the IP Routing Manager Topology View; right-click on the node and select 'Utilities, Select As Layer 3 Path Source'.
2. Select the end point in the IP Routing Manager Topology View; right-click on the node and select 'Utilities, View Layer 3 Path Change History'.
   This will bring up a new dialog.
3. In the dialog, enter the time range for historical data you want to view.
4. Click the View History button.
   The Path Change History table appears.

> **NOTE**
> The content of the Path Change History table is static. For dynamically updated information on the path, use the Path Hop Details table as described in Layer 3 Path Viewing.

Viewing the Path History of a Monitored Path

For any Layer 3 Path model you have created you have the ability to see the path change history for that monitored path.

- Right click on an existing Layer 3 Path model in the explorer view and select 'Utilities, View Layer 3 Path History' top launch the path history view.
  This automatically fills in the Source and Destination IPs for the dialog.

Viewing the Path History from an Alarm

If you receive a Layer 3 Path Lost or a Layer 3 Path Change alarm you can view the path change history.

- Right click on the alarm in the Contents Panel and selecting 'Utilities, View Layer 3 Path History'.

Important Considerations

> **NOTE**
> If IP Routing Manager is connected to a Packet Design appliance that is running the Packet Design/Route Explorer software, the Path Change History's 'Link Prefix' column may contain missing values.

The path change history functionality works by taking samples of the Layer 3 topology at specified times. By default IPRM take 100 samples over the time range. This is the maximum allowed; the minimum is 5. If you experience performance problems or the path history takes a long time to render you can reduce this value by setting a parameter in the $SPECROOT/IPRM/REX/config.xml file:

```
<path_history_sample_count>10</path_history_sample_count>
```

> **NOTE**
> DX NetOps Spectrum recommends setting this value to 30 for 2x series Packet Design appliances running REX software below version 8.0. For 3x series appliances running REX software 8.0 or higher, you can experiment setting the value as high as 100 (which is the default) depending on the size of your Layer 3 topology.

Frequently Asked Questions and Useful Tips

Frequently Asked Questions

Integration Overview

- **Question: Does the REX appliance perform the layer 3 topology discovery and DX NetOps Spectrum IPRM imports that information from REX? Or does DX NetOps Spectrum perform its own native layer 3 topology discovery?**
  Answer: REX performs the layer 3 topology discovery and passes this information to IPRM.
- **Question: Does IPRM keep track of routing changes in real-time?**
  Answer: IPRM Topology is updated in real time and spotlighting can be used to view up to date hop-by-hop path composition. Also, IPRM supports related alarm conditions based on the updates from the REX appliance.
- **Question: How long does it take to remove the disconnected router icon from the topology view?**
  Answer: Disconnected routers age-out of REX database slowly. The time for the disconnected routers to age out of the REX database is based on the routing protocol being used. In the case of OSPF, it is approximately one hour.
- **Question: Why would IPv4 prefix count shown in REX GUI differ from IP Subnet count shown by IPRM?**
  Answer: There could be several reasons for this:
  - By default (user configurable) IPRM hides the icons representing User Subnets - IP subnets with one (gateway) or more routers and a number of end hosts and devices served. In this case these models are completely hidden in OneClick UI (including Topology Tab, List Tab and in the Navigation Tree).
  - By default (user configurable) IPRM hides the icons representing IP Subnets used solely to connect two routers. In this case the corresponding IP Subnet models are hidden in OneClick UI.
  - REX's count of prefixes is really count of routes. For example, if the same prefix is announced by two routers it counts it twice. IPRM does not create multiple IP Subnet models for the same subnet.
  - In the case of distributed Route Explorer configuration, it's important to make sure IPRM is connected to the REX appliance acting as the 'Modeling Engine' and the replication is turned on to ensure it can have the complete topology picture.
- **Question: Does IPRM connect to REX via the REX API or directly with the database?**
  Answer: IPRM uses REX's XML RPC API.
- **Question: How secure is data transmitted between REX and IPRM?**
  Answer: The API password (Query Password) is transmitted encrypted.
- **Question: Which ports are used between the REX, IPRM Connector and SpectroSERVER?**
  Answer: Communication between SpectroSERVER and REX Connector uses ports 14002 and 14006 for CORBA, as well as a few dynamically-chosen ports as connection sources from the REX Connector to the Naming Service and to the SpectroSERVER.
  REX Connector expects open TCP connection to port 2000 on the Route Explorer appliance.
- **Question: How secure is REX appliance's communication with the routers?**
  Answer: REX only uses read-only access and supports both RADIUS and TACACS secure access methods. It also supports MD5 authentication for OSPF and BGP peerings.

User Interface

- **Question: What kind of UI does REX offer?**
  Answer: REX UI has two components: Administration & Configuration is available via the Web and all the product features are available via X-GUI. Any standard browser (Mozilla, IE) can be used to access the Web UI while the X-GUI can be accessed either via a X-Manager or a VNC client.
  X-GUI is disabled in IPRM-RR, since it is expected you use DX NetOps Spectrum's OneClick GUI.
- **Question: Without GUI component, how will I be able to configure IPRM-RR traps?**
  Answer: IPRM provides configuration of path change, router state, adjacency state, prefix state, and peering state alerts via OneClick GUI.
- **Question: Is there a context sensitive integration between IPRM and REX GUI?**

Answer: No, DX NetOps Spectrum OneClick is the only GUI part of the solution (with the exception of web-based REX administration console used during the installation).

Routing Protocols

- **Question: What routing protocols does IPRM support?**
  Answer: REX appliance supports all routing protocols, including the following standard protocols as defined by the IETF RFCs: IS-IS, OSPF, BGP and MP-BGP as well as EIGRP (Cisco proprietary) and static routes.
  IPRM shows the full routed topology as discovered by REX appliance. Initial version of IPRM in 9.2.0 offered navigation-based spotlighting and icon labeling for OSPF protocol and AS groups. As a result of customer feedback, in Spectrum r9.2.1 navigation-based spotlighting was replaced by individually customizable topology maps. Users can still leverage the spotlighting dialog to highlight different routing protocol aspects in their Layer 3 topology view.
  Other visualization techniques under consideration for future IPRM releases include display of EIGRP session establishment status, spotlighting for IS-IS levels, display of attributes from BGP packet headers as observed by REX, etc.
  In addition to visualization techniques, protocol support includes display of routing protocol attributes. Here DX NetOps Spectrum already offers support for BGPv4 (RFC 1269), RIPv2 (RFC 1724) and OSPF (RFC 1253) SNMP MIBs.
- **Question: Is monitoring of BGP prefixes supported?**

For example, I have configured BGP and are able to see BGP UP/DOWN alarms of most of the devices monitored by DX NetOps Spectrum. In case BGP is up on one router, but the number of BGP prefixes received from its peer is 0 (zero), the device will not have complete routing table and as result it will not route the traffic. I would like to know if DX NetOps Spectrum can generate an alarm in such case.

Answer: REX appliance can generate necessary alert and forward to IPRM which will then be able to create an alarm from this.

For loss of Peering, between REX and a router, the 'Peering State' alert can be used. For loss of prefixes, the 'BGP Prefix Flood/Drought' alert can be used.

The BGP Prefix Flood/Drought alarms rely on the baseline of BGP prefixes, which is done automatically by the IPRM Route Recorder (no user intervention necessary).

Let the recording go on for 24 hours; IPRM RR would have then established a baseline (containing prefixes expected from each BGP peer). Configure BGP prefix Flood and Drought alert with the following parameters:

1. A watch list (via a router group) containing the peers that need to be watched.
2. Configure prefix drought alert with a threshold of 100%. This will issue an alert when all baselined prefixes being heard from this peer are withdrawn.

For a greater sensitivity (i.e., cause alerts to be fired when some number of prefixes, less than 100%, are withdrawn), change the threshold parameter as required (say 90%).

- **Question: How can modifying my IPRM Alert Configuration Defaults affect DX NetOps Spectrum's performance?**
  Answer: IPRM Alert Settings *can* be used to tailor the amount of events/alarms you see in IPRM.
  For example, if you are monitoring adjacencies, and a link was flapping constantly (coming up and going down), you would see 5 events for that link from each router (both ends of the adjacency) in 60 seconds. For prefix alert, you would see 5 events for that prefix being pulled in 60 seconds. Therefore, with a constantly flapping interface, you would probably see 15 events for that 60 second period, and then 15 more for the next 60 second period. With IPRM's current default settings, users could see up to 15 events/minute for a flapping link. If 'x' links are flapping, then you'd get 'x' times 15 events/minute. This may raise the concern that with a high rate of events was that these events would trigger excessive device polling. However, that is not a problem because IPRM does not poll devices as a result of these events; instead, these events trigger IPRM to re-discover the layer-3 topology so that it can determine what has changed in the topology. Additionally, IPRM leverages a 'rolling timer' to prevent topology re-discovery from happening too frequently. IPRM doesn't re-discover for every event that occurs, it will only re-discover at most once every minute

as long as these events are occurring. Since the alerts are all disabled by default in IPRM, it's not until a client site makes changes to the alert configuration that they can expect to start seeing events/alarms in IPRM.

- **Question: Is multi-protocol BGP supported?**
  Answer: No, multi-protocol BGP is used for VPN services over Layer 3 and as such is outside the scope of this product.
- **Question: Does IPRM support route summarization boundaries for EIGRP?**
  Answer: To correctly model EIGRP networks, REX needs to establish an adjacency behind the summarization boundaries. So knowledge of where the summarization boundaries exist is important for the correct and complete REX configuration.
- **Question: Are static routes supported?**
  Answer: REX appliance normally doesn't poll devices, but it can optionally collect static route information (along with other data not distributed in the routing protocols such as interface names, router models, etc) via SNMP polling. This is completely configurable by the customer. The static nature of such data requires very low frequency polling (e.g. once a day or week at most). Note also that REX only polls (potentially only those selected by customers) routers, not every network device.
- **Question: Protocols that support some form of adjacency setup, such as OSPF or IS-IS, may be used to bootstrap a BFD session. These protocols may then use BFD (Bidirectional Forwarding Detection) to receive faster notification of failing links than would normally be possible using the protocol's own 'KeepAlive' mechanism. Does IPRM support BFD events?**
  Answer: BFD protocols are not supported.
- **Question: Is policy based routing supported?**
  Answer: No, the information about policy based routing is not advertised in routing protocols, the primary source of information for REX.
- **Question: How are routing loops supported?**
  Answer: Unfortunately, REX does not track routing loops. Routing loops are transient issues in the network. Most routing loops occur as protocols converge and hence are short-lived (milliseconds to seconds). For example, a routing loop may occur because of reordering of the protocol packets received by a router. REX does not (in most cases cannot) track such transient issues. REX assumes that all protocols have converged and accordingly compute the Routing Information Bases at different routers.
  There could be some long lived routing loops that can be analyzed in REX. If you try to find a path and a routing loop exists along the way, REX will show that depending on the root-cause of the routing loop. However, it is virtually impossible for REX to dynamically track and alert on this condition as this would require REX to track all possible paths in the network (even for a medium size network this could mean tens of thousands of paths).
- **Question: Does IPRM support IPv6 addressing?**
  Answer: No, the first release of IPRM doesn't support IPv6 addressing. REX appliance does support this and IPRM will be extended to support this as well in one of the upcoming upgrade releases.

Managed Path

- **Question: Can I create managed path between routers from different OSPF areas?**
  Answer: Yes, you can monitor paths from one OSPF area to another. It supports paths from one AS to another.
- **Question: Can I create managed path between user IP subnets?**
  Answer: The source of the managed path has to be a router; the destination can be IP Subnet. Currently, REX restricts managed path source to routers only, however this restriction may be lifted in the future and IPRM updated accordingly.
- **Question: Can IPRM display path change history between any two end points?**
  Answer: Yes, IPRM can display path change history between any two end points.
- **Question: Why does the hop count for managed path differ between IPRM and REX? (This question is ONLY applicable for 9.2 and REX 8.5.x.)**
  Answer: In REX, pseudo-nodes represent IP Subnets, in line with IS-IS and OSPF protocols modeling.
  The REX GUI distinctly displays pseudo-nodes in its topology view and IPRM follows the suit representing them as IP Subnet models in its own topology view

When it comes to Layer 3 path modeling, REX continues treating pseudo-nodes as distinct hops. For example, for a path between 3 routers (A, B, and C), there will be two IP Subnets (X and Y) with one between each router. REX would consider this path to have 4 hops:
1. A -> X
2. X -> B
3. B -> Y
4. Y -> C

IPRM takes a different approach and treats pseudo-nodes as something that data simply travels through to get to the next hop. So, for the path above, IPRM shows the path having only 2 hops:
1. A -> X -> B
2. B -> Y -> C

- **Question: Does IPRM support routing changes simulation?**
  Answer: No. This capability is supported by Route Explorer's GUI which is not directly leveraged by IPRM. If you already have a full REX appliance, you will continue to be able to benefit from this capability.
- **Question: Does IPRM expose any other Packet Design add-on capabilities such as MPLS LDP LSPs and traffic monitoring?**
  Answer: No, IPRM integrates DX NetOps Spectrum with REX only at this stage.
- **Question: Why is the path cost value different from the sum of the metrics in the Path Hop Details table?**
  Answer: The path may traverse an internal router hop, for which the metric value will not be displayed in the Path Hop Details table (or the Path View tab).

IPRM Deployment

- **Question: What versions of REX appliance are supported?**
  Answer: CA Spectrum r10.0 IPRM supports REX firmware version 11.0.47.1.
- **Question: How many REX appliances will I need?**
  Answer: One REX appliance is sufficient (2600 unit with 8 GB of RAM). If you have multiple REX appliances, IPRM connects to the master modeling engine that aggregates all of the data from multiple REX appliances.
- **Question: I already have HP RAMS deployed. Can I use it with IPRM?**
  Answer: Yes, if you are considering moving from HP NNM to DX NetOps Spectrum, you can take advantage of any investment they have made in HP RAMS. IPRM is compatible with HP RAMS 8.11 and newer.
- **Question: I have a distributed Route Explorer installation, i.e. a modeling engine and one or more route recorders. The route recorder is responsible for collecting routing information, and the modeling engine provides GUI and a central database. How is DX NetOps Spectrum IPRM integrated in this case?**
  Answer: In this case, you should configure IPRM's REX integration settings to connect to the Modeling Engine and enable replication so that all data is copied to the Modeling Engine.
- **Question: What's the performance impact from adding IPRM to the SpectroSERVER?**
  Answer: There is no precise measurement available. A lot will depend on the amount of IPRM polling configured by the operator (for example, none vs. every 60 seconds).
- **Question: What is the performance impact on the network devices when REX performs discovery and monitoring?**
  Answer: Route Explorer operates by passively monitoring the routing protocols exchanges between routers. To do this, Route Explorer 'announces' itself as if it were a router, but doesn't advertise any prefixes, so no traffic flows to/through it (its neither a bottleneck nor a failure point).
  When Route Explorer is installed, it is given an IP address and adjacencies are set up with a small number of routers (e.g. REX requires an adjacency with one router in each OSPF area or IS-IS level). If the router is local, this can be a physical connection. Otherwise, REX can connect to remote routers either via GRE tunnels or VLANs. Once these adjacencies are up, REX begins monitoring the protocols and is able to present a complete, network-wide map within minutes.
  The only messages Route Explorer sends out to its neighbors are periodic 'KeepAlive' messages to maintain these adjacencies. Thus, there is virtually zero load placed on the real routers or the network, both during discovery and ongoing monitoring.
- **Question: How does REX establish routing adjacency with remote autonomous systems?**

Answer: In these cases GRE tunnels or VLANs can be configured to establish secure links to remotely located routers. In cases where the network policy, architecture or devices don’t allow or support this, you can deploy multiple REX appliances (Route Recorders) at the appropriate places in the network to establish direct adjacencies/peerings with the routers. In this distributed architecture, routing information collected by multiple Route Recorders is combined into a single topology view at the 'master' REX appliance (Modeling Engine). IPRM accesses the Modeling Engine to get the complete topology view of the network.

- **Question: Does REX appliance always work in passive mode?**
  Answer: In the majority of cases routing topology and management information is obtained from listening to routing protocol exchanges. However one routing protocol works in a different way from others - Cisco EIGRP. In this case only the change in distance to a given subnet can be determined. Based on this information alone REX uses its sophisticated algorithms to work out the routing topology and path state changes, and then uses some non-privileged CLI commands (and screen scraping) to validate its calculations.
  Optionally, REX also uses SNMP or CLI commands to collect information about Static routes; since static routes don’t change frequently, SNMP polling is done at very low frequency (typically, once a day; this is configurable).
- **Question: How does REX achieve scalability?**
  Answer: Packet Design have done quite a bit of scalability testing, both in-house and based on real world usage by their customers, and they can provide more details. That said, in 99%+ of the cases, a single appliance will be more than adequate. The reason Packet Design is so confident is due to the fact that under recommended deployment conditions REX will not have any users (i.e. GUI sessions).
  The limitation for scalability has to do with memory since REX requires a separate copy of the topology model for each user session (as well as another copy for monitoring/alerting). Large networks (# of routes and routers) can require very large models, and therefore take up a significant amount of memory. But without the need to support any direct users, Packet Design believes they have not seen a network that would not easily fit within a single appliance. And they have production deployments at some of the largest routed networks in the world. To put some parameters around that, Packet Design is already supporting networks comprised of over 7,000 routers and more than 8 million routes with a single box.

  NOTE
  Only routers are counted as nodes, whereas the number of devices monitored by a large DX NetOps Spectrum deployment includes many non-routing devices.
- **Question: Does REX support NAT'ed environment?**
  Answer: No, REX does not support this environment type.
- **Question: Does IPRM support subnets configured with overlapping IP addresses?**
  Answer: Route Explorer and DX NetOps Spectrum take different approaches to supporting this network configuration, and as such the initial release of IPRM does not support it. However, in future IPRM may be extended to support this deployment scenario.
- **Question: Does IPRM support fault tolerant DX NetOps Spectrum deployment?**
  Answer: Yes, secondary SpectroSERVER takes over communication with REX in case of failover, and pass control back to primary when it goes back online.
- **Question: Does REX support fault tolerant deployment?**
  Answer: Yes, primary REX appliance can be configured with secondary 'warm' standby REX appliance. In case of primary REX failure the switchover is to be done manually, but the secondary REX will have complete routing history from prior to the switchover event.
- **Question: Does IPRM support distributed DX NetOps Spectrum deployment?**
  Answer: Yes, all user configuration and interaction is performed on the main location server (MLS).

Useful Tips

Useful tips for IP Routing Manager:

- REX Connector should run on separate CPU from SpectroSERVER if possible to reduce impact on SpectroSERVER. This is because when Layer 3 discovery is performed, the REX Connector causes additional CPU usage. This is

primarily applicable to Windows, which can be adjusted by using Task Manager's 'Set Affinity' option. UNIX/Linux applications handle the load balancing better.
- You must disconnect from REX to change REX configuration settings (for example, REX administrative domain), and then re-connect for the changes to take effect.
- When using Monitoring Mode = 'Off', you must manually configure topology and path-related change notifications (traps) within REX GUI beforehand.
- Disconnected routers age-out of REX database slowly. The time for the disconnected routers to age out of the REX database is based on the routing protocol being used. In the case of OSPF, it is approximately one hour.
- Use 'Clear Layer 3 Topology' button to start over. This deletes all IPRM modeling in MLS, including subnets, paths, and Unmanaged Device models.
- An Autonomous System in IPRM is equivalent to an 'Administrative Domain' in REX.
- Source and destination of Layer 3 paths must be within REX's visibility.
- When tracing a path to a router loopback IP, the last internal hop is used in cost calculation and number of hops, but not included in path hop details nor topology view. This last internal hop is not represented visually in topology view.
- In OneClick, the 'View Layer 3 Path Change History', 'Create Layer 3 Path' and 'Select as Layer 3 Path Source' menu items may be disabled in the Tools, Utilities menu. These items are enabled/disabled based on what is currently selected in the Explorer tab.
  For example, if you select a router model under the 'Layer 3 Devices' container in the Explorer tab, all three of these menu items will be enabled under Tools, Utilities.

IP Routing Manager Troubleshooting

Known Anomalies

DX NetOps Spectrum's IP Routing Manager contains the following known anomalies:

- Modeling your Packet Design appliance in DX NetOps Spectrum will prevent IPRM-related traps and alarms from working properly.
- LANs/subnets cannot be used as the source for IP Subnet paths. This is a limitation of Route Explorer. In some cases, multiple routers route to the same subnet so REX is unable to determine the source router for the path.
  Therefore, IPRM cannot discover forward-and-return paths when a LAN/subnet is used as the path destination.
- Some interface information may appear 'missing' from the Path Hop Details table. In some cases, particularly with IS-IS, REX cannot determine the interface IP address for a hop in the path.
- Paths that have an internal router as the final path hop do not show this hop in the Path Hop Details table or in the path topology view. An internal router hop cannot be visualized in the topology and is therefore omitted from the topology view and the Path Hop Details table, but the cost of the internal hop is included in the path's total cost calculation.
- IPv6 is not supported.
- Overlapping IP address and AS ranges are not supported.
- If IP Routing Manager is connected to a Packet Design appliance that is running version 9.2 or greater of the Packet Design/Route Explorer software, the Path Change History's 'Link Prefix' column may be missing values.
- **Problem:**
  If a particular IP address is configured on the interface of more than one router and these routers are modeled in DX NetOps Spectrum, but at least one of the routers is modeled as a Pingable; IP Routing Manager may create links to the wrong model. In particular, there may be a link connecting one router model to another when that link should instead be connecting the Pingable model (which represents a different router) to another router model. A side-effect of this issue is a broken Layer 3 path view, where the path through the network is contiguous from source to destination, but the IP Routing Manager topology view displays a disjointed view of the path.
  Solution:
  Replacing the Pingable model with a model of a more appropriate, SNMP-managed model type (for example, Rtr_Cisco) and then re-discovering the Layer 3 topology in IP Routing Manager can resolve the issue.

IP Routing Manager Traceability Events

IP Routing Manager creates DX NetOps Spectrum events when any of the following occurs in IPRM so that users have visibility into what led to the current state of the IPRM. All events will be generated on the IP Routing Manager model.

- **DISCOVER_BUTTON_PRESSED_EVENT (0x564001e)**
  This event will be generated whenever the user presses the Discover button in OneClick. The name of the user will be included in the event.
- **CLEAR_BUTTON_PRESSED_EVENT (0x564001f)**
  This event will be generated whenever the user presses the Clear button in OneClick. The name of the user will be included in the event.
- **CONNECT_TO_REX_BUTTON_PRESSED_EVENT (0x5640020)**
  This event will be generated whenever the user presses the Connect button in OneClick. The name of the user will be included in the event.
- **DISCONNECT_FROM_REX_BUTTON_PRESSED_EVENT (0x5640021)**
  This event will be generated whenever the user presses the Disconnect button in OneClick. The name of the user will be included in the event.
- **REX_HOSTNAME_CHANGED_EVENT (0x5640022)**
- **REX_QUERY_PASSWORD_CHANGED_EVENT (0x5640023)**
- **REX_ADMINISTRATIVE_DOMAIN_NAME_CHANGED_EVENT (0x5640024)**
- **REX_HEARTBEAT_INTERVAL_CHANGED_EVENT (0x5640025)**
  The above events will be generated whenever the user changes one of the corresponding REX Configuration values OneClick. The name of the user will be included in the event.
- **TRAP_BASED_TOPOLOGY_POLL_EVENT (0x5640026)**
  This event will be generated whenever IPRM performs a Layer 3 topology poll based on receiving a trap from REX.
- **STARTUP_BASED_TOPOLOGY_UPDATE_EVENT (0x5640027)**
  This event will be generated whenever IPRM performs a Layer 3 topology update when the SpectroSERVER is restarted, or when regaining contact with REX after a communication problem.
- **MON_MODE_CHANGE_BASED_TOPO_UPDATE_EVENT (0x5640028)**
  This event will be generated whenever IPRM performs a Layer 3 topology update based on changes to the Topology Monitoring Mode in OneClick.
- **IF_RECONFIG_BASED_TOPOLOGY_UPDATE_EVENT (0x5640029)**
  This event will be generated whenever IPRM performs a Layer 3 topology update based on DX NetOps Spectrum router model interface reconfigurations. This is done to keep the Layer 3 topology in sync with DX NetOps Spectrum device modeling.
- **TOPOLOGY_UPDATED_EVENT (0x564002a)**
  This event will be generated whenever a Layer 3 topology update has been completed. Situations in which this is done outside of other events include topology changes detected by the REX Connector in Active mode.
- **TOPOLOGY_MONITORING_MODE_CHANGE_EVENT (0x564002b)**
  This event will be generated whenever the user changes the Topology Monitoring Mode in OneClick. The name of the user will be included in the event.
- **TOPOLOGY_POLLING_INTERVAL_CHANGE_EVENT (0x564002c)**
  This event will be generated whenever the user changes the Topology Polling Interval in OneClick. The name of the user will be included in the event.

IP Routing Manager Debugging

You have the ability to turn on debugging output for IPRM intelligence running inside the SpectroSERVER. An attribute called DebugLogEnabled exists on the IP Routing Manager model which controls the generation of log files. You can access it via the Attribute Tab of the IP Routing Manager model's Component Details panel.

Upon SpectroSERVER startup, it checks to see if DebugLogEnabled is TRUE. If so, then it opens a new log file in the IPRM/logs/debug directory and begins outputting data when necessary. When you set DebugLogEnabled to

FALSE, the log file is closed. When you set DebugLogEnabled back to TRUE, a new log file is created. After that, every night at midnight, the current log file is closed and a new one is opened. The log file names are of the form 'Layer3ModelingDebugLog.<datetime>'.

All Layer 3 topology and path modeling functionality will output modeling details to the debug log. This includes manually-triggered discoveries, discoveries triggered by changes detected, path creation requests, etc. The modeling details we output include data on each router and subnet that exists in the Layer 3 topology, links that were created or removed, UnmanagedDevice models created, attribute value updates, path creation details including hop details, path change information, and so on.

All debug output is time-stamped so the user can easily cross-reference it with debug output generated by the REX Connector.

Steps to Take if the REX Connector is Not Running

Steps to Take if the REX Connector Is Not Running

1. Check the following files for errors:
   ```
   $SPECROOT/IPRM/REX/REXCON.OUT
   $SPECROOT/lib/SDPM/processd_log for errors
   ```
2. Use DX NetOps Spectrum CLI to manually start the REX Connector:
   ```
   update  action=0x5640002 mh=<mh_of_IPRM>
   ```

Steps to Take if Unable to Connect to Route Explorer

Steps to Take if Unable to Connect to Route Explorer

- The password encryption for API requests may be broken if the Query Password had been set and then a Route Explorer software update had been performed. To alleviate this problem, set the Query Password on each of the Packet Design appliances (via the REX web interface) to a temporary value, save the changes, and then set the Query Password back to the original password and save the changes.
- Verify that 'Queries' are enabled in REX.
- Login to REX web interface and go to Application, 'Administration -> Application -> Queries'.
- Ensure the following are enabled 'XML-RPC Query Server' and 'Enable remote access'.
- Set the password to the same value entered in the IPRM configuration.
- Ensure REX Connector is able to access tcp port 2000 on Route Explorer system.

Layer 3 Topology Discovery Fails

Error: Unable to retrieve Layer 3 topology

Symptom:

When the DX NetOps Spectrum is integrated with Route Explorer (REX), the discovery of layer 3 topology fails.

Description:

When you initiate the Discovery to retrieve layer 3 topology from REX, if the response is huge the discovery fails with the following heap memory error:

```
Java.lang.OutOfMemoryError.Java heap space
```

Solution:

Starting from the 10.1 release, the DX NetOps Spectrum retrieves Layer 3 topology in chunks. The default chunk size is configured as 5000. So, DX NetOps Spectrum retrieves 5000 entries per chunk. You can configure the default chunk size in the config.xml file, which is at $SPECROOT\IPRM\REX. The changes are reflected in the REXCON.OUT file after 30 seconds.

Set the chuck size to a less number to receive the Layer 3 topology data.

Example: <chunk_size>2000</chunk_size>

REX Connector Debug Logging

To Set REX Connector Debug Logging

1. Edit IPRM/REX/config.xml file.

   NOTE
   The REX Connector checks config.xml for changes every 30 seconds, there is no need to restart the connector.
2. Set the <debugging> tag to one or more of these values (comma-separated, case insensitive):
   - **Off**
     No output at all.
   - **Trace**
     Minimal REX Connector startup, shutdown, administrative tasks.
   - **Comm**
     Includes basic information about communication with REX.
   - **Debug**
     Includes useful, human-readable topology and path data that REX provides. Technical Support can use this to cross-reference with modeling debug provided by the SpectroSERVER.
   - **Max**
     Includes raw XML from REX API requests and responses.

   Output is written to:

   ```
   IPRM/REX/logs/debug/debug.log.<timestamp>
   ```

REX Connector Debug Client

A command line utility 'rexcli' simulates SpectroSERVER-based requests to the REX Connector. The 'rexcli' utility provides a picture of the topology or a path at a specified date/time.

To Edit the 'rexcli' Utility

1. Navigate to IPRM/REX and edit 'rexcli' utility.
2. Specify the following supported parameters (if needed):
   - **-topo <date>**
     Retrieves the topology at the given time.
   - **-path <srcIP> <dstPrefix> [forward_and_return] <date>**
     Retrieves the path at the given time.
   - The 'dstPrefix' is specified in 'slash notation'
     (for example, 10.253.5.2/32 for a router destination).
   - The 'forward_and_return' (optional) retrieves both the forward and return paths
     (when possible).
   - The 'date' is the date/time for which you want to retrieve the IP Subnet topology or path, and is specified as: 2011/07/31T14:56:42; which represents: July 31, 2011 2:56:42 pm.

   The output is written to the shell; you can redirect to a file.

By enabling debugging in the REX Connector's config.xml file, you can correlate the output from 'rexcli' utility with the debug output from the REX Connector.

Initiating Layer 3 Topology Discovery Using an Offline Database

The REX Connector can also use 'offline' Route Explorer databases. 'Offline' Route Explorer databases are generally historical databases which have been archived on the Route Explorer unit. This feature allows IPRM to discover the Layer 3 topology stored in an offline database; this is useful for testing different types of networks and for customer support.

You can add an additional XML configuration option to the REX Connector's `config.xml` file. This additional option is a boolean named 'allow_offline_dbs'. When this option is set to 'true', users can configure the REX integration (via OneClick) to use an offline database.

Follow these steps:

1. Navigate to the Component Details Panel of the IP Routing Manager model, and then open the Configuration sub-view.
2. Clear any existing Layer 3 topology data by clicking **Clear**.
3. Set REX Connector options in `IPRM/REX/config.xml` .
4. Add <use_offline_dbs>TRUE</use_offline_dbs> to the file to access offline databases.
5. Click **Discover** in the IP Routing Manager model's Configuration sub-view.

Modeling and Managing Your IT Infrastructure Administrator

Network Modeling in DX NetOps Spectrum

Network modeling in DX NetOps Spectrum is the act of graphically representing network entities and their connections. Icons that are created, placed, and connected within the OneClick topology views represent various aspects of a modeled network.

Using the modeling features of the OneClick client, you can easily create and maintain accurate software models of your network. These intelligent network models enable DX NetOps Spectrum to determine actual points of failure and to suppress superfluous alarms.

DX NetOps Spectrum network representation is based on logical relationships and rules and appears different from your network diagrams. Discovery uses address tables and ICMP ping tests to identify subnet address ranges and devices within those ranges. Once discovered, DX NetOps Spectrum models those devices and subnets.

OneClick Topologies

You can use four core topologies to model your IT infrastructure in DX NetOps Spectrum:

- Universe Topology
- Global Collections Topology
- World Topology
- TopOrg Topology

All four of these topologies are available from the Navigation panel.

> **NOTE**
> We recommend that you begin modeling with the Universe topology. After you have established one or more modeled elements in the Universe topology, you can reuse these modeled elements to define the other topologies.

To navigate through the model views of any topology, click the view control icons in the toolbar. In some cases, you can click an aggregate icon container to view its content.

Universe Topology

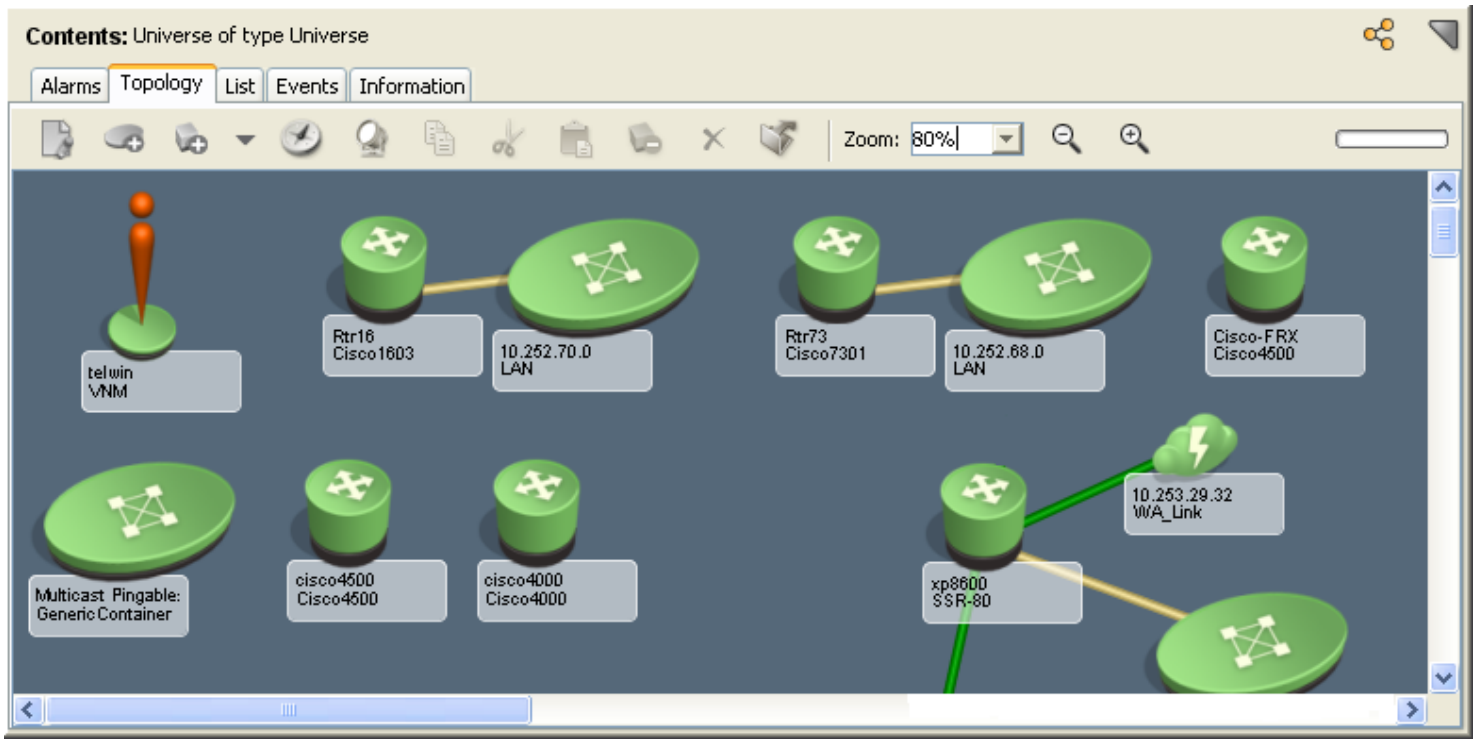
The Universe topology helps you organize an enterprise network view of your infrastructure. Most often, it provides a view of:

- A top-level topology view of OSI Layer 3 devices and their connections
- A drill-down topology view of OSI Layer 2 devices and their connections
- A Component Detail view, showing the attributes that are associated with a modeled entity

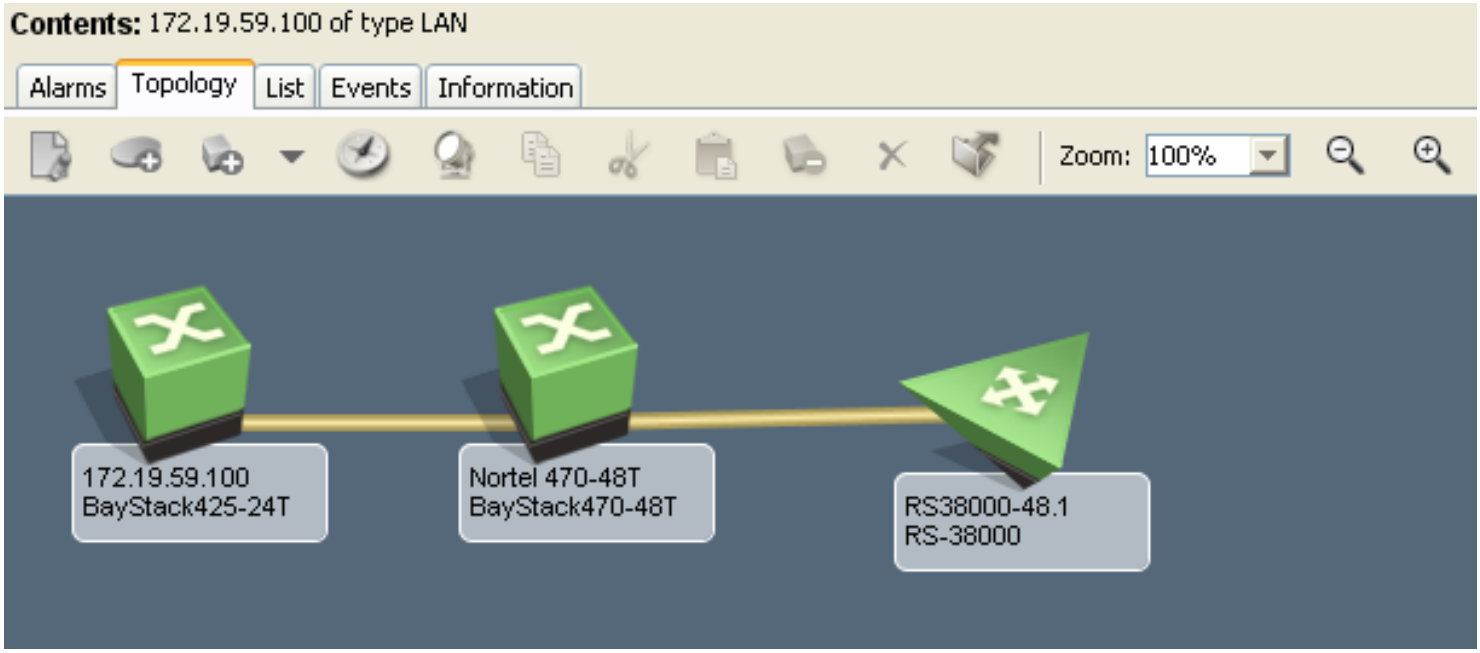
The top view in the Universe topology typically includes:

- The DX NetOps Spectrum Virtual Network Machine (VNM)
- Network groupings
- Network segments
- OSI Layer 3 devices and their connections

The following illustration shows a typical top view:

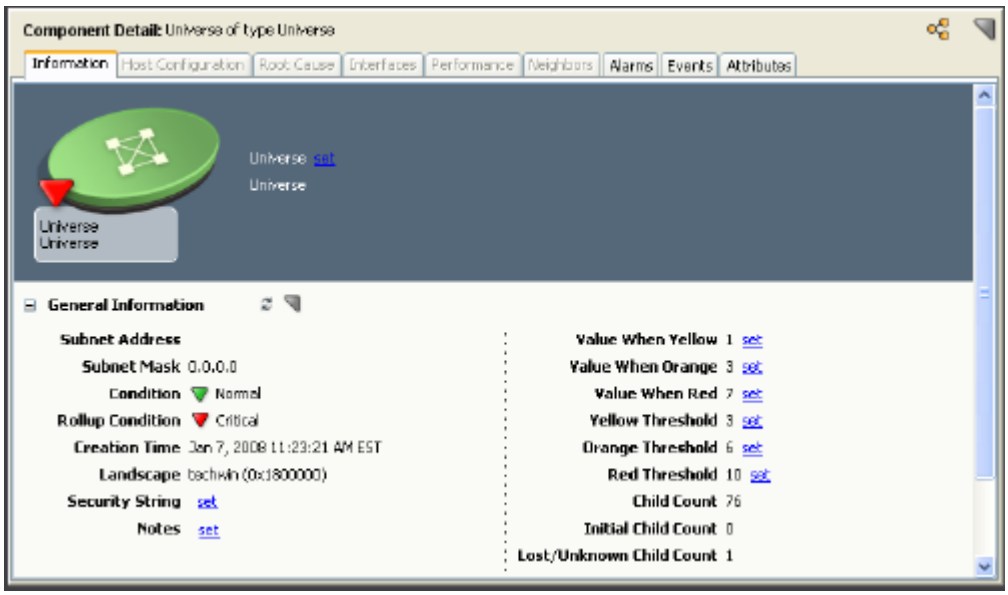


The following illustration shows a drill-down view of a LAN container that is selected from the top view. A drill-down view in the Universe topology most often includes all OSI Layer 2 devices and their connections. The drill-down view also shows off-page references to devices modeled in other views, as shown in the following illustration:



Component Detail Panel

The Component Detail panel within the Universe topology identifies the attributes that are associated with a modeled network entity. Attributes can include the device interfaces, alarms and events, and other pertinent device information.



You can view the device attributes and possibly change their settings by clicking the Component Detail panel tabs. Depending on the context of the Contents panel, you can use the Component Detail panel to:

- View current alarms in the Alarm Details tab.
- View and modify general device settings in the Information tab. For example, grant or deny access to a modeled device by providing or possibly removing a security string.
- View root cause analysis data in the Root Cause tab.
- View CPU and memory utilization information in the Performance tab.
- View device interface information in the Interfaces tab.
- View neighboring routers in the Neighbors tab.
- View historical events in the Events tab.
- View attribute information from the Attributes tab. This information appears only under these conditions:
  - You select an entity from the Explorer tab.
  - You are in either the Topology, List, or Events tab in the Contents panel.

Define Models in the Universe Topology

You can define models in the Universe topology using the OneClick Discovery feature, which automates the modeling process for you. You can also manually define new models or edit existing models in the Universe topology by using the modeling tools that are provided with OneClick.

The Universe topology view represents a true connectivity view of your infrastructure. Therefore, we strongly recommend that you reuse modeled elements from this view when creating other views. Therefore, as a best practice, copy model elements from the Universe topology view to create Global Collections, World, or TopOrg views. This approach helps to ensure accurate fault isolation of your network within the OneClick environment.

Global Collections Topology

A *landscape* is the network domain that is managed by a single SpectroSERVER. In OneClick, a landscape is the network view of one SpectroSERVER. To organize entity-based network views that span one or more landscapes, use Global Collections. Global collections enable you to monitor all aspects of your IT infrastructure from any perspective.

As an administrator, you can use Global Collections to create and track collections of network entities, organizations, or services that make up your infrastructure. For example, you can create and maintain collections that identify and track:

- Response teams within an organization responsible for maintaining equipment
- Devices supporting various services in your organization
- Customers receiving services from your organization

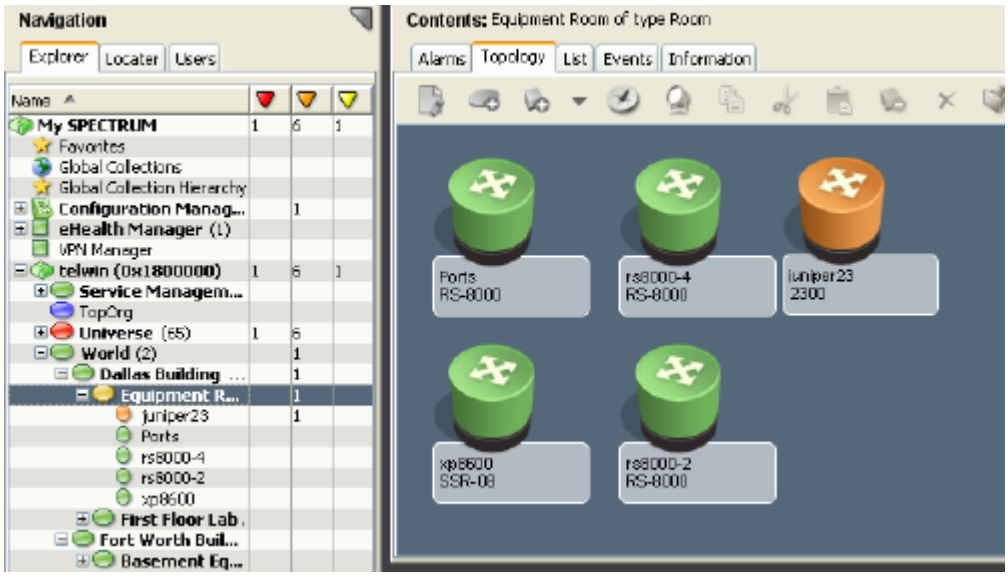
View or Modify Modeled Devices

You can view and change modeled device attributes or settings in the Component Detail panel within a Global Collections topology. Click the tabs in this panel for general information about a device, its interfaces, alarms, events, and other pertinent information.

World Topology

The World topology helps organize your network geographically in OneClick. In this topology, you can represent device models of network locations from a national or regional level all the way down to a wiring closet.

The following example illustrates a drill-down view of an equipment room that is at a fictitious North Dallas location.



In the World topology, you can create several layers of views that represent locations of your network devices. For example, you can have views for Texas regional offices, Dallas office, and Dallas Equipment room. Additionally, the Component Detail panel lets you view and sometimes change the attributes that are associated with a modeled device in any World topology view. For instance, clicking the Component Detail tabs for a modeled device lets you view device information, interfaces, alarms, events, and other pertinent device information.

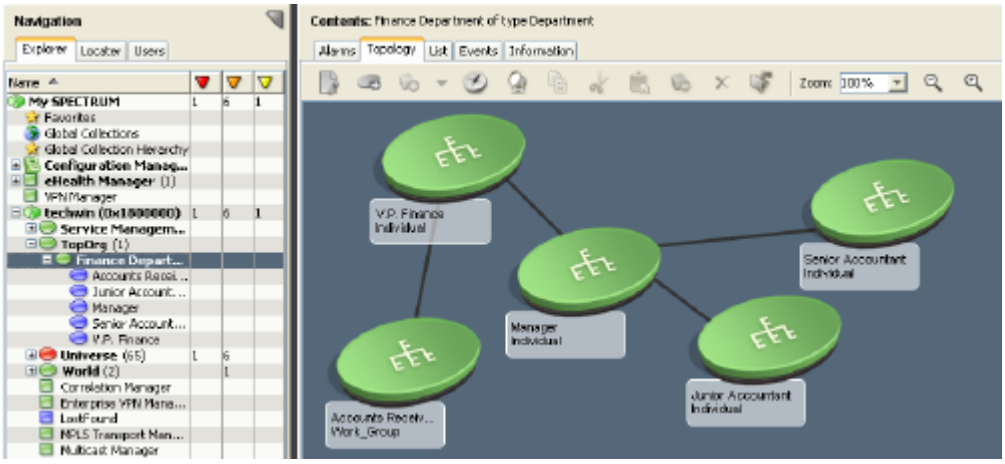
NOTE

- When populating the World topology views with modeled devices, we highly recommend that you **copy and paste modeled elements from Universe topology views**. Universe topology views represent the true connectivity views of your infrastructure, helping to ensure accurate fault isolation of your network within the OneClick environment.
- The Cut, Copy, Paste and Delete functionality for VHM models is disabled by default. If required, you can enable the option by following the steps explained in the 'Cut, Copy, Paste and Delete Functionality for VHM Models' section.

TopOrg Topology

The TopOrg topology allows you to represent your network organizationally. In this topology, you can group subnets and device models by services, responsibilities, departments, or by other organizational considerations.

The following illustration displays an organizational view that identifies individuals and groups within a fictitious Finance Department. This type of view is useful when identifying how a network failure or a reconfiguration affects an organizational unit.



NOTE
If you purchased the Service Manager module, you can use this module with the TopOrg topology to model business services and applications. Further, the Service Manager module also tracks the performance of the service against a contract or Service Level Agreement (SLA).

NOTE
For more information about using the Service Manager module, see Service Manager .

In the TopOrg topology, you can create several layers of views that represent various levels of your network devices. For example, you can have views for Enterprise ownership, Department ownership, supporting devices, and supporting services. Additionally, the Component Detail panel lets you view and possibly change the attributes that are associated with a modeled device in any TopOrg topology view.

NOTE
When populating the TopOrg topology views with modeled devices, we recommend that you copy and paste modeled elements from Universe topology views. Universe topology views represent the true connectivity views of your infrastructure. Therefore, using them as your base for all other views helps ensure more accurate fault isolation of your network within the OneClick environment.

Topology Toolbar

The following table describes some of the buttons available in the Topology tab toolbar for working with topologies.

Icon	Description
	**Edit mode:** Click to put the current topology view into Edit mode.
	**Create new model by type:** Click to create a new model by type and add it to the topology view.
	**Create new model by IP:** Click to create a new model by IP address or Host Name. Click the down arrow and select one of the following options: Create By IP Create By Host Name
	**Discovery:** Click to create a new discovery based on the selected model or models.

	**Spotlight View:** Click to highlight all models related to a VPN, a VLAN, or router redundancy in the Topology view. Spotlighting allows you to easily determine relationships between these items and your network, and between these items and other models on your network. When you click the Spotlight view button, a menu appears containing the following options: Router Redundancy VLAN List VPN List
	**Remove Model:** Click to remove the selected model from the Topology view.

Icons in Topology Views

Icons that appear in OneClick topology views are graphical representations of network entities. Some network entities include:

- Individual devices
- Groups of devices
- Geographic locations
- Physical connections

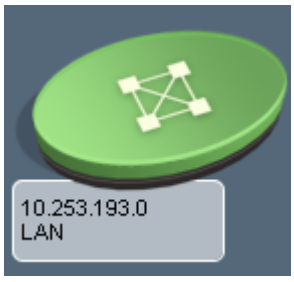
An icon is simply the image with which you interact when you manipulate and configure a modeled element. When monitoring the condition of your network, an icon represents the current status of a device, network group, device location, or a physical link.

DX NetOps Spectrum offers both aggregate and individual icons for representing entities in your infrastructure.

Aggregate Icons

An IP or physical address does not manage an aggregate icon. However, you can configure aggregate icons to display the device IP address that a container represents. Or, you can configure them to display the subnet address of the devices that the container represents. Aggregate icons primarily act as containers, or placeholders, in a topology view.

An aggregate icon often represents a network group. Some examples of network groups are LAN, LAN_802.x, FDDI, ATM_Network, WA_Link, and Dialup_Link. The following image is an example of an aggregate icon:

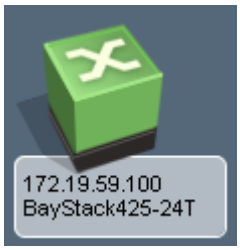


DX NetOps Spectrum offers many types of aggregate icons. The appearance of an aggregate icon is always based on the entity it represents in your network.

Individual Icons

Individual icons are typically associated with an IP address or a physical address. DX NetOps Spectrum can communicate directly with the devices the individual icons represent, so long as the entities they represent are SNMP and ICMP entities.

Individual icons often represent network devices. Some examples of individual icons are those icons that represent a router, switch, or host. The following image is an example of an individual icon:



DX NetOps Spectrum offers many types of individual icons. The appearance of an individual icon is always based on the entity it represents in your network.

Icon Types by Device model category

You model devices in your network through Discovery or manual modeling. During modeling, DX NetOps Spectrum automatically determines the functionality of each device and selects the appropriate icon shape and symbol for that device.

Icons come in various shapes and sizes. Icon symbols vary by the model class that is represented and by the topology in which the icon is located.

Icon types include:

- VNM icon
- Network group icon
- Device icon
- Off-page reference icon
- Segment icon
- Live pipes (links)

VNM

The Virtual Network Machine (VNM) icon typically appears in the top-level view above the network group icons. The background color of the VNM icon changes to indicate the current condition of the SpectroSERVER. For example, the VNM icon turns red when SpectroSERVER disk space reaches 90 percent capacity.

Network Group Icon

The network group icons represent network groupings, for example, cable groups, LANs, and IP Class A, B, C networks. The following icon shows an example of a LAN network group icon:

Device Icon

A device icon represents an individual device. The device icon color changes to indicate the current condition of the modeled device. For example, the device icon changes to red whenever DX NetOps Spectrum detects a serious condition requiring attention. The following icon shows an example of a device icon:

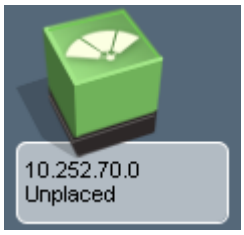
Off-Page Reference Icon

The off-page reference icon is a special purpose topology icon. The off-page reference icon represents a device that is directly connected to a device in the current view but which is modeled in another layer. The following icon shows an example of an off-page reference icon:

NOTE
You can globally suppress off-page references in topology views. For more information about suppressing off-page references, see Using OneClick .

Segment Icon

Segment icons represent conceptual elements of a network. Examples of segment icons can include a coax segment, a wa_segment, a fanout, an unplaced icon, and a pingable. The following icon shows an example of a segment icon:



Live Pipes (Links)

Live pipes represent the connection status between network devices. The links change color to indicate the current condition. A gold pipe represents a resolved connection or indicates that live pipes have not been enabled for the connection. A silver pipe represents an unresolved connection.

Live pipes are not enabled by default. To monitor the connection status between devices, enable a live pipe. The following image shows an example of a live pipe which has been enabled:

Icon Color and Condition

All icons change color to indicate the condition of the device or devices they represent. For instance, a device icon changes color when an alarm condition for that device occurs. A rollup triangle on a device icon or container icon changes color when an alarm condition occurs on one or more of its components. The components include devices or interfaces.

Rollup Condition Colors

The rollup triangle that is associated with the network container icon indicates that one or more components of the container has an alarm condition. In this example, a device within the network container has a critical alarm.

The rollup triangle that is associated with the device icon indicates that a component of that device has an alarm condition. In this example, an interface on a Cisco router has a minor alarm.

Icon Condition Colors

Logical links (or pipes) change color to indicate the condition of the connection:

- Disabled or maintenance conditions = brown
- Good conditions = green
- Initial conditions = blue
- Suppressed or unknown conditions = silver
- Poor conditions = red

Proxy Models

To easily identify proxy models in a topology view you will see a PROXY text watermark show up overlaid over the model icon. You can see various types of highlighted Proxy Models below:

Do the following to enable or disable this view:

- You need to select the relevant model and set the **"Is a Proxy Model"** attribute to **Yes**, to enable the view.
- You can also set the **"Is a Proxy Model"** attribute to **No**, to disable the view.

> **NOTE**
>
> You can use **Attribute Editor** to set the **"Is a Proxy Model"** attribute by editing the **isEventCreationEnabled (0x129f8)** attribute.
> The **isEventCreationEnabled (0x129f8)** attribute= **false** only indicates that the model is a proxy model and nothing more.

Wireless Network Devices

Wireless Devices models in the Topology View has the Wireless sign on the icon to distinguish them from non-wireless device models.
The icon colors of wireless device models will depend on the state or Icon condition as explained in the Icon Condition Colors above.

Wireless Controller Device icons

Wireless Access Points icons

Topology View using new Accessibility Icon theme

To provide a better user experience and in adherence to VPAT/508 standards, we have enabled an Accessibility theme for the OneClick topology view.
This theme provides new indicators denoting model severity/icon conditions on top of our existing topology view. This will enable model severity view, which is currently color coded, for users who are unable to distinguish between colors.
The accessibility icons will have additional indicators so that users can easily identify the difference in the icon model's severity.

The OneClick Console has two icon themes, Accessibility and OneClick. By default, the OneClick icon theme is selected.

To enable the Accessibility theme, follow these steps:

1. Click **Start Console** from the OneClick Administration page.
   The OneClick Console user interface is launched.
2. Navigate to the menu options, **View** > **Icons Theme**, and then select **Accessibility**.
   The OneClick Console automatically refreshes and the Accessibility indicators are displayed in the topology view.

The Icons theme selection is persistent across sessions. Once you have selected an icon theme for your topology, you will view the same theme in the OneClick console every time you logon.

Mapping the Accessibility Icons/ indicators with model severity status:

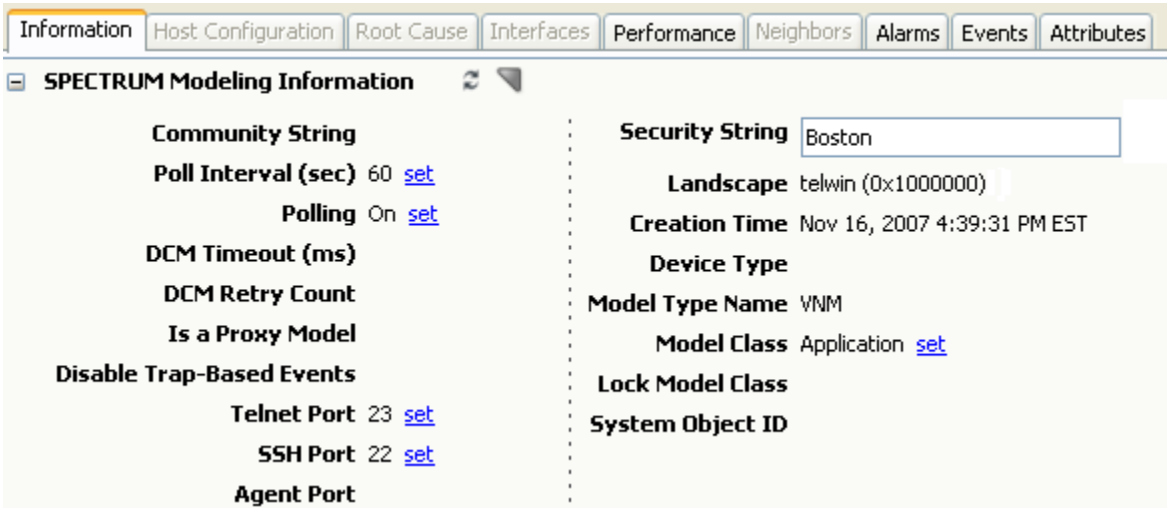
To revert to the (default) OneClick icon theme, follow these steps:

1. Navigate to the **OneClick Console** menu options, **View** > **Icons Theme**, and then select **OneClick.** The OneClick Console automatically refreshes and the default theme is displayed in the topology view.

Provision Access to Modeled Elements

As an administrator you can secure access to models by applying a security string. A security string establishes permission to various modeled elements in a OneClick topology view such as a modeled device.

After a security string is applied to a modeled device, all subcomponent models (or views) of that device inherit the security string. The security string field for implementing this model security appears in the Component Detail panel, as shown in the following example:



As shown, the security string ‘Boston’ prevents any user that does not have an Access Group of ‘Boston’ from accessing this modeled element.

NOTE
Any user with an Access Group of ‘Admin’ overrides model security; such users can access all model elements regardless of the security string implemented.

NOTE
For more information about creating or renaming Access Groups that are associated with individual users or user groups, see OneClick Administration .

Discovering and Modeling Your Network

This section details how you can discover devices and model your network topology by the following means:

Discovery

Discovery is a utility that you can run to find devices in your network and to model them automatically in the Universe topology. Discovery uses a set of configuration parameters that you can modify to determine which network entities to discover and model. You can reuse any set of previously saved configuration parameters and you can also rename, duplicate, or delete configurations.

A configuration determines the focus and scope of a Discovery or modeling session. You define the configuration by selecting parameters on the Discovery Console Configuration tab. After you create a configuration, you can choose when to activate it:

- You can activate the configuration immediately.
- You can schedule the activation, including scheduling it to recur.
- You can save the configuration and activate it later.

Depending on your user privileges, you can use the automated Discovery and modeling features together or you can use them separately. For example, here are some ways you can use Discovery:

- **To perform network inventories:** With read/write privileges to the Discovery parameters, you can use Discovery to identify assets within your network. And, you can export, as needed, the results describing those assets to a desired file format for further review and distribution.
- **To model network entities that you want to manage:** With read/write privileges to both Discovery and modeling parameters, you can use Discovery to:
  a. Determine which elements in your network you want to model.
  b. Identify how you would like DX NetOps Spectrum to model these elements for you automatically.

  Specifying modeling parameters with the Discovery parameters lets you easily create accurate software models of your infrastructure with less time and effort than modeling manually.

Separate Configurations

Creating separate Discovery and modeling configurations offers you more flexibility for customizing the Discovery and modeling process. By providing separate configurations, you can:

- Discover limited portions of your network by performing several smaller Discovery operations instead of performing one large Discovery operation.
- Model the results of a Discovery operation using different modeling options.
- Filter and export the results of a Discovery session in different ways.
- Filter and export the results of a modeling session in different ways.

Discovery Console

The Discovery Console consists of two panels: the **Navigation** panel on the left, and the **Contents** panel on the right.

Navigation Panel

The **Discovery Navigation** panel contains the **Landscape** drop-down list, a toolbar, and a list of configurations and folders available on the selected landscape. From the toolbar you can create, copy, delete, import, or export configurations and create new folders for the configurations. In the **Name** column, you can select a configuration to open it and view its details in the **Contents** panel.

Contents Panel

The Discovery Console groups the parameters you use to define **Discovery** and modeling configurations into four tabs in the **Discovery Contents** panel:

- Configuration Tab
- Discovery Tab
- Modeling Tab
- History Tab

Configuration Tab

The Discovery Configuration tab lists all the required and optional parameters you can set to create a configuration.

This tab contains the following settings:

- **Seed Routers**
  Specifies the IP addresses or hostnames of your network seed routers that act as an initial communication point for discovering the network topology. For a hostname, DX NetOps Spectrum attempts to resolve the host name to an IP address when you start Discovery. If DX NetOps Spectrum cannot resolve the host name to an IP address, an error message occurs. The error message displays in the Discovery status panel in the lower section of the Discovery tab. If you start Discovery in the context of a network element, the device IP address is populated in the Seed Router field. You can add more seed router IP addresses as needed.
- **IP/Host Name Boundary List**
  Specifies IP ranges, IP addresses, host names, or any combination of this information that DX NetOps Spectrum can use to define the boundaries for the configuration. For a host name, DX NetOps Spectrum attempts to resolve the host name to an IP address when you start Discovery. If DX NetOps Spectrum cannot resolve the host name to an IP address, an error message occurs. The error message displays in the Discovery status panel in the lower section of the Discovery tab. You can start Discovery in the context of a container in the Topology tab or the Explorer tab. In this situation, DX NetOps Spectrum populates the IP/Host Name Boundary List with an address range. The address range is determined using the IP address and network mask of the selected device. You can specify more IP addresses, IP address ranges, or host names.

  NOTE
  The IP/Host Name Boundary List also accepts single IPv6 addresses; however, IPv6 ranges are not supported.
- **SNMP Information**
  Specifies SNMP community strings and profiles for SNMPv1, SNMPv2c, and SNMPv3 communication.
- **Modeling Options**
  Specifies whether to perform a Discover only operation or a combined Discover and model operation. To review and accept the modeling defaults or to edit them, click the Modeling Options button.
- **Advanced Options**
  Contains the Advanced Options button which opens the Advanced Options dialog. In the Advanced Options dialog you can review, accept, or redefine the following options:
  - **SNMP Ports**
    Specifies the default SNMP port and any additional ports. This feature is most often used for managed node environments that use port numbers other than the default port number of 161.
  - **IP Exclusion List**
    Create, delete, modify, or import an IP exclusion list. This list instructs the Discovery session to exclude the devices in a defined IP address range.
  - **Discovery Options**
    Specifies whether the Discovery process uses ICMP and Route Tables. For the Route Tables option, you can set a Throttle level to control how often the server sends SNMP requests.
  - **Auto Export**

Specifies whether to export the Discovery session results automatically and the preferred format for exporting them: comma-delimited, tab-delimited, or web page.

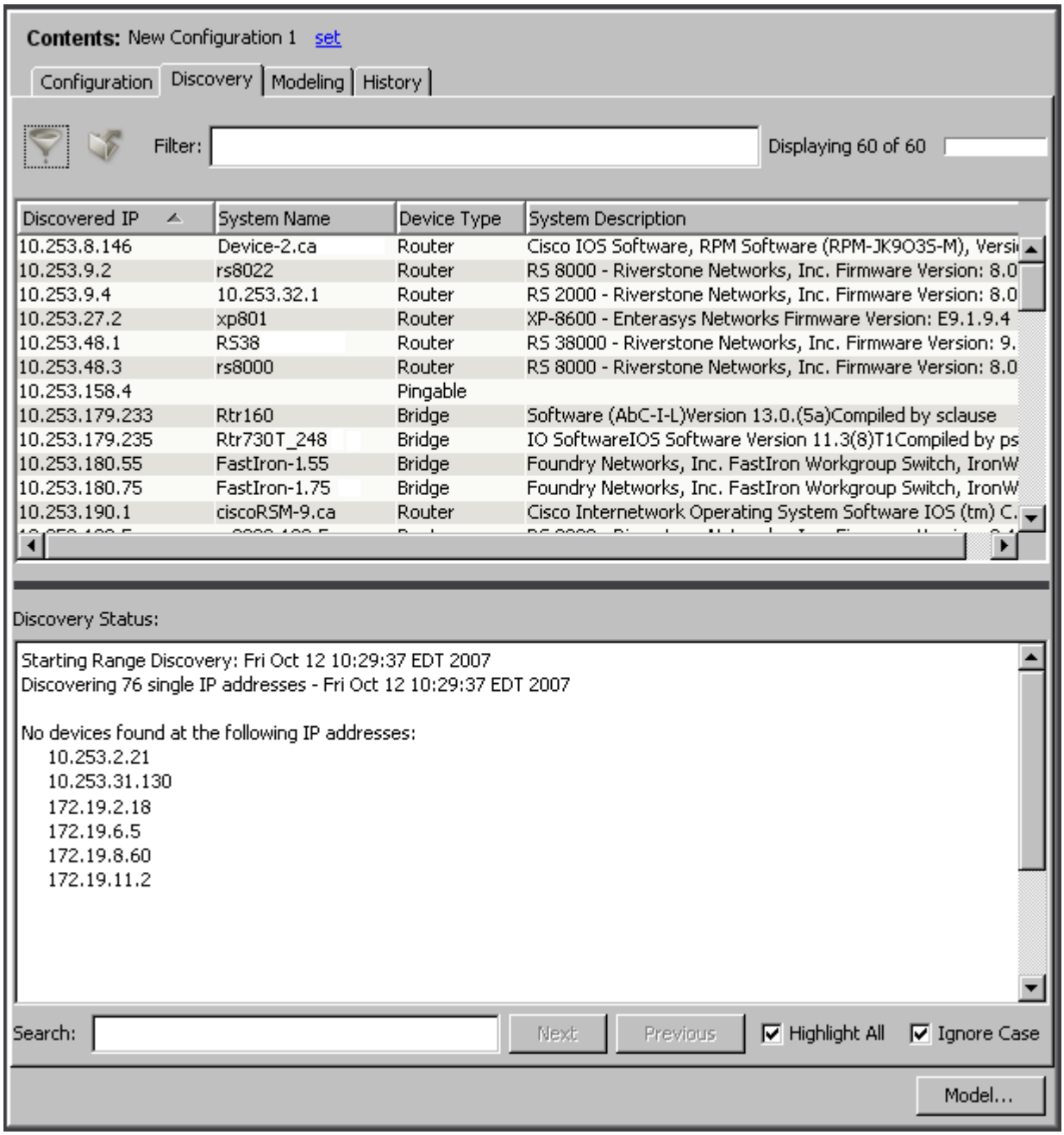
- **Scheduling Options**
  Specifies whether to activate certain configurations regularly using a schedule.

  NOTE
  In a DSS environment that spans multiple time zones, the local time of each SpectroSERVER is used for scheduling. For more information about OneClick schedules, see OneClick Schedules in a DSS environment .
- **Save options as default**
  Specifies whether to save the current configuration settings as the default configuration. For example, by default the ‘Discover only’ option is enabled on the Configuration tab. To change the default setting to the ‘Discover and automatically model to DX NetOps Spectrum’ option, select that option. Then, select the ‘Save options as default’ check box, and select File, Save.
- **Discover**
  Activates the Discovery session as it is defined in the Configuration tab.

Discovery Tab

The Discovery tab displays the results and status of the most recent Discovery session for the configuration that is selected in the Navigation panel. The discovered devices appear at the top of the Discovery tab. In the lower section of the Discovery tab, the status and error messages that are generated display in the Discovery Status panel. All users with Discovery privileges can access this tab. The Discovery tab initially appears disabled for new Discovery configurations and becomes enabled after an initial Discovery session is generated from the Configuration tab.

The Discovery results table includes the following columns by default: Discovered IP, System Name, Device Type, and System Description. To display the Table Preferences dialog, right-click one of the column headers. From this dialog, you can select more columns to display. The Model State column tells you whether the device is modeled in OneClick. This information is useful when identifying devices that are discovered on your network that require modeling in OneClick.



Filter, Sort, Export, Search, and Model Discovery Results

The Discovery tab provides the following options to help you review, filter, export, and model Discovery results:

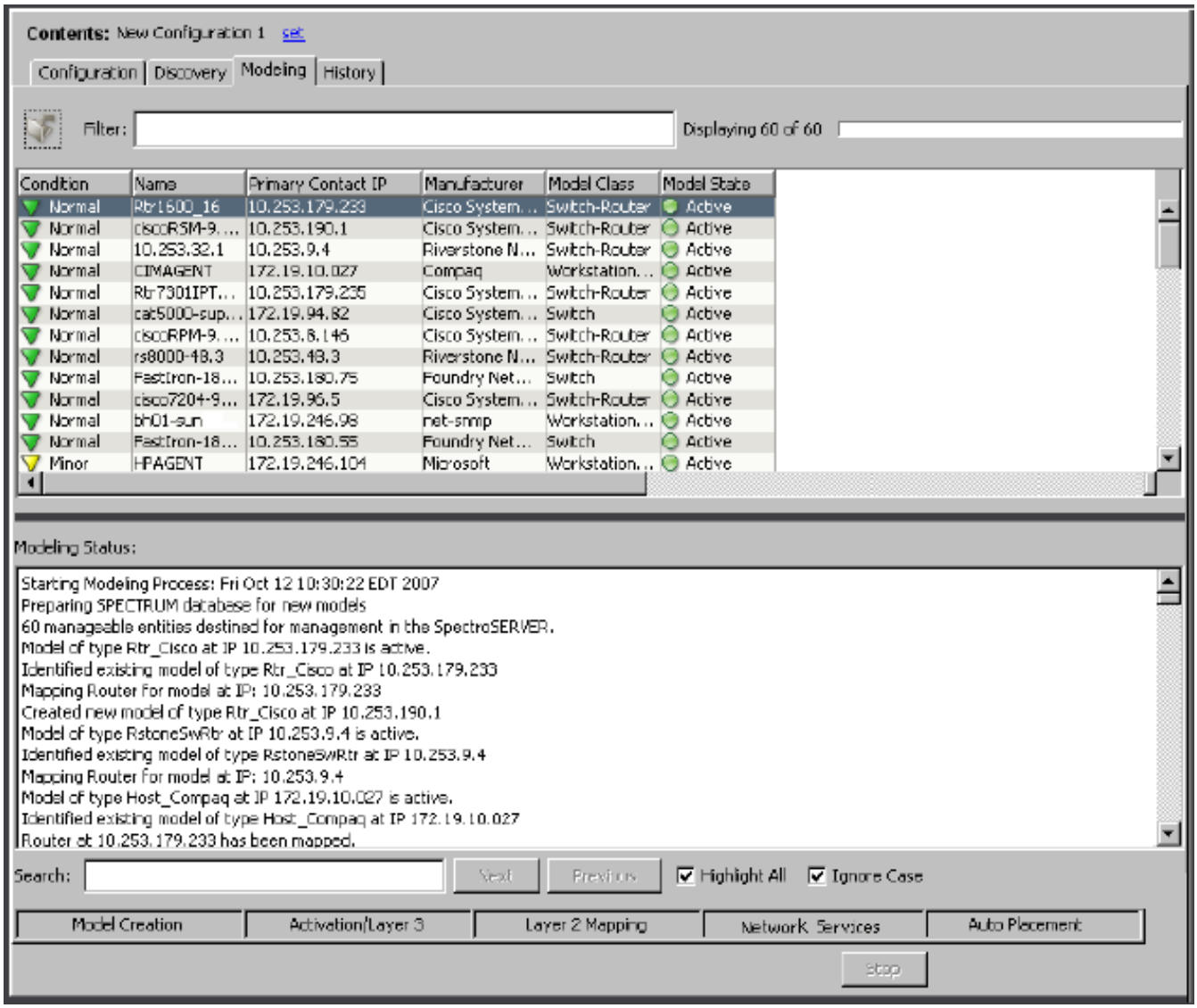
- **Filter:** The Filter text box lets you quickly filter the devices in the results list. For example, to develop, model, and export a list of Cisco devices from your results list, complete the following procedure:
  a. Type **Cisco** in the Filter text box.
     This filters the results list by Cisco devices.
  b. Click Model.
     This models only the Cisco devices.
  c. Click Export.

This exports the Cisco device results list.

- **Advanced Filter:** To apply more filter criteria, click the Advanced Filter button and creating one or more expressions. These expressions let you set more filters on the Discovery results list.
- **Exclude:** You can exclude one or multiple entries in the Discovery results list by right-clicking the entries that you want to exclude and selecting Exclude. You can also exclude these devices from the Discovery configuration. Select 'Save options as default' in the Configuration tab. Exclude one or more devices, then save the configuration. Discovery excludes those devices when it runs.
- **Export:** Click Export to open the Export table data to file dialog. In this dialog, you can specify a file format and location to export the Discovery results.
- **Status Search:** Enter character strings that you want to search for in the Discovery or the Model Status sections into the Search text box. To see all search matches in the Status panel, select the ‘Highlight All’ check box. To make the search case-insensitive, select the ‘Ignore Case’ checkbox. Use Next and Previous to navigate through the search matches in the Discovery Status panel.
- **Model:** Click to open the Modeling Configuration dialog. In this dialog, you can review or modify the default modeling options provided. When you click OK, DX NetOps Spectrum models only the devices appearing in the Discovery results list.

Modeling Tab

The Modeling tab displays the results and status of the last modeling session. The top portion of the tab shows the modeled devices. The Modeling Status section at the bottom displays the status and error messages that are generated during the last modeling session.



All users with modeling privileges can view the Modeling tab. This tab initially appears disabled for new Discovery configurations and becomes enabled only after an initial modeling session is activated from the Discovery tab.

NOTE
The Modeling tab provides the same options that the Discovery tab does to help you review, filter, export, and search modeling results.

Modeling Session Status Bar

The Modeling Status section displays a status bar at the bottom. The status activates immediately after clicking the Model button on the Discovery tab. This status bar divides the modeling process into four operation phases:

- **Model Creation**
  Phase 1 -- The label for this phase turns green while the SpectroSERVER processes the data to model.
- **Activation/Layer 3**
  Phase 2 -- The label for this phase turns green while the SpectroSERVER maps Layer 3 devices.
- **Layer 2 Mapping**

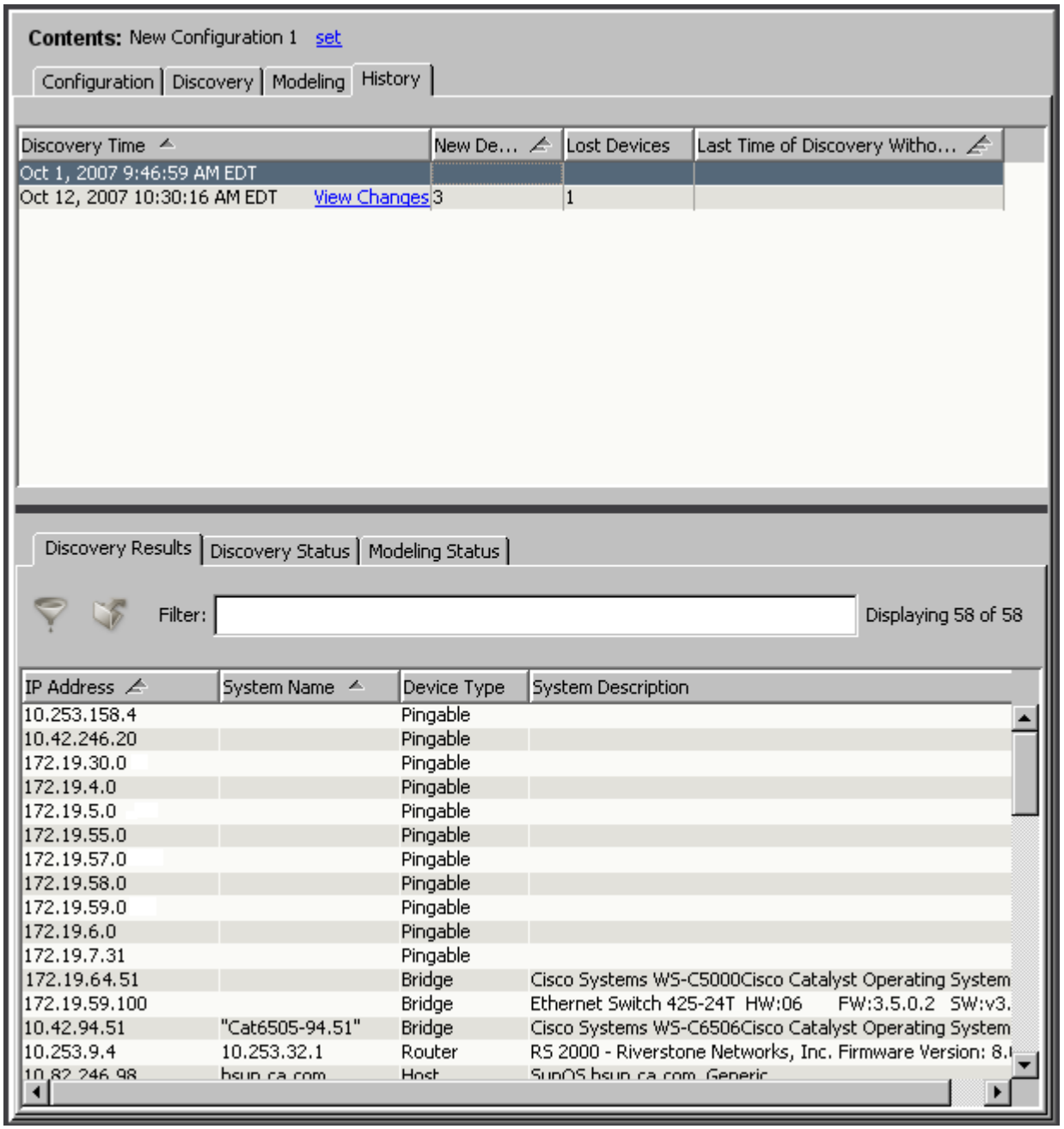
Phase 3 -- The label for this phase turns green while the SpectroSERVER waits for model activation and maps Layer 2 devices.

- **Network Services**
  Phase 4 -- The label for this phase turns green while the SpectroSERVER processes the running status for each network service.
- **Autoplacement**
  Phase 5 -- The label for this phase turns green while the SpectroSERVER places the models appearing in the modeling result list.

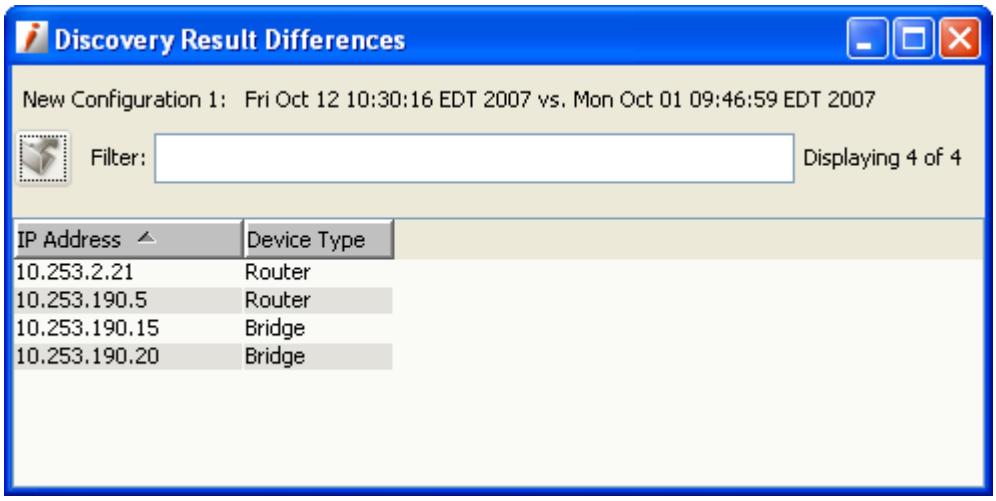
  NOTE
  On the Discovery Console, the Status box at the bottom displays the status and error messages that are related to each of these phases.

History Tab

The History tab displays information about the configuration that is selected in the Discovery Navigation tab. The following image shows an example of the History tab.



- **Discovery Time**
  Displays the time and date that a Discovery session occurred.
- **View Changes**
  Opens the Discovery Result Differences dialog. The dialog lists all devices that were either found or lost during the selected Discovery session when compared to the previous Discovery session. You can filter the information and can export it to a file.



- **New Devices**
  Displays the number of new devices that are found since the previous Discovery session for the selected configuration.
- **Lost Devices**
  Displays the number of devices that are lost between a previous Discovery session and the next Discovery session for the selected configuration.
- **Last Time of Discovery Without Changes**
  Displays the time and date information for when the selected configuration last ran without any changes. No time and date information are displayed when changes have occurred each time that the configuration has run.

The History tab also displays the Discovery Results, Discovery Status, and the Modeling Status tabs in the lower panel.

Discovery Connection Status

To display the Discovery Connection Status dialog, click the Connection status icon
( ).
In the Discovery Console, this icon is located in the Status bar.

When displayed from the Discovery Console, the Connection Status dialog shows:

- SNMP Service (SpectroSERVER) connection status
- Web server connection status
- Landscape (SpectroSERVER) connection status

NOTE
For more information about the Connection Status dialog and the Status bar, see Connection Status Indicator.

Open the Discovery Console

When creating a Discovery configuration, you can open the Discovery Console from the context of a selected model. You can also open the Discovery Console without this context.

To open the Discovery Console without specific model context, do *one* of the following steps:

- Click Tools, Utilities, Discovery Console. The Discovery Console opens.
- Click

  (Discovery) in the Topology tab toolbar without selecting a device in the Navigation panel or Topology tab. The Discovery Console opens.
- Click

  (Discovery) in the List tab toolbar without selecting a device in the Navigation panel or Topology tab. The Discovery Console opens.

Discovery can also be launched with context. Launching in context automatically generates a Discovery configuration that is based on the models that are selected in the OneClick Console. Discovery retrieves IP/subnet mask information from the selected routers, switch-routers, and LAN models and automatically creates a Discovery configuration that is based on this information.

To open the Discovery Console with context, do *one* of the following steps:

- Right-click a selected model in the Universe Navigation panel and select Utilities, Discovery Console.
- Right-click a selected model in the Contents panel, Topology tab. Select Utilities, Discovery Console.
- Select a model in the Topology tab and click

  (Discovery) in the toolbar.
- Right-click a selected model in the Contents panel, List tab. Select Utilities, Discovery Console.
- Select a model in the List tab and click

  (Discovery) in the toolbar.

The Discovery Console opens. If you select certain types of containers, a router, or a switch-router, the Configuration dialog also opens from the context of the selected model.

Discovery derives the context differently, based on how you create the Discovery configuration, as follows:

- From the Discovery button (Topology or List tab) - Discovery derives the context using your selection in the Contents panel.
- From the Tools menu - Discovery gets its context from your most recent selection in the Navigation panel.

Define a Discovery Configuration

A Discovery configuration determines which network devices to discover. When creating a Discovery configuration, you can open the Discovery Console from the context of a network element or container. You can also open the Discovery Console without this context.

Without context: Follow these steps:

1. Open the Discovery Console.
2. Specify the Discovery configuration settings.

NOTE
The IP/Host Name Boundary List and SNMP Information sections are mandatory for a successful Discovery configuration.

3. Do *one* of the following steps:
   – Click Discover to activate a Discovery session for the configuration you created.
   The results of the Discovery session appear on the Discovery Console, Discovery tab.
   – Click File, Save.
   The configuration that you created is saved.

Note: To apply the most recent changes that are made to any Discovery configuration, select the 'Save options as default' check box.

You can create a Discovery configuration with seed router context. As expected, Discovery discovers the selected router model. Plus, depending on other configuration parameters, Discovery discovers the LANs and routers that are connected to the seed router.

A Discovery configuration with container model context finds all of the network devices that reside within that container IP range.

With context: Follow these steps:

1. Open the Discovery Console.
2. Do *one* of the following steps:
   – If the Configuration dialog opens, enter a name for the configuration and click OK.

   NOTE
   The Configuration dialog opens when Discovery cannot find an existing Discovery configuration for the selected device.

   – If the Use Existing dialog opens, do *one* of the following steps:
   • Select an existing configuration that you want to use and click OK.
   • Click Create, enter a name for the new configuration, and click OK.
   If no existing configurations include the IP address of the selected device, the Configuration dialog opens. Use the provided name for the new configuration (which is based on the device type), or enter another name.
   – If the Configuration dialog does not open,
   click Discovery 
   Then, do *one* of the following steps:
   • Select an existing configuration that you want to use and click OK.
   • Enter a name for the new configuration and click OK.
   The Configuration dialog closes and the Discovery Console is now fully visible. The Seed Router section of the Configuration tab contains an IP address entry for the selected device. If you select 'LAN containers' in the OneClick Console before launching Discovery, the IP/Host Name Boundary List section can contain one or more IP ranges.
3. Specify the Discovery configuration fields in the Configuration tab.
   The Discovery configuration is defined.

How to Set Discovery Configuration Parameters

To set the parameters in the Discovery Configuration tab, do the following steps:

- Populate the Seed Routers List
- Specify Host Names and IP Addresses
- Specify SNMP Information
- Specify Modeling Options
- Configure Advanced Options

Seed Routers

The Seed Routers section is optional, but recommended for large Discovery operations. Seed routers are a core list of routers that Discovery uses as a starting point when determining the routed subnets. All routers that are discovered within the IP/Host Name Boundary List are treated as seed routers.

If you start Discovery in the context of a selected device, the IP address for the device appears in the Seed Routers field.

Populate the Seed Routers List

You can populate the Seed Routers list to have Discovery determine the routed subnets from this list.

Follow these steps:

1. In the Seed Routers field, enter addresses or host names to build a list of one or more seed routers and click Add.
2. From the Discovery Type drop-down list, choose *one*:
   - **Routers only**
     Discovers only the routing devices within the IP range or host name.
   - **Routers and only local LANs in IP/Host Name Boundary List**
     Discovers only the routed subnets within the IP range or host name.
   - **Routers and all local LANs**
     Discovers all subnets that are routed by the routers that are discovered in the IP range or host name.
3. Depending on how you want DX NetOps Spectrum to discover subnets, choose *one* of the following options:
   - Select the 'Scan Subnets Using ICMP/SNMP Sweep' option and then select the maximum subnet size that you want to discover.
   - Select the 'Discover Subnets Using:' option and then select either 'ARP tables,' 'Cisco CDP tables,' or both.
   - Continue setting configuration parameters in the other view sections, as needed.

   Seed Routers information has now been added to the Discovery configuration.

IP/Host Name Boundary

The IP/Host Name Boundary List section is required for all Discovery configurations. The IP/Host Name Boundary List field populates with an IP range when you start Discovery with a container selected from the Topology tab. The IP range is determined using the network address and network address mask for the container. However, you can specify more IP ranges, individual IPs, or one or more host names, as needed.

- For IPv4, the boundary list accepts single IPs, IP ranges, and host names. Wildcards can also be applied.
- For IPv6, the boundary list accepts single addresses and host names. IPv6 ranges are not valid input. Also, wildcards cannot be applied to IPv6 addresses.
- If an IPv6 address is entered into the first IP address field, the second IP address field is automatically disabled. In this case, you cannot enter an IPv6 range.

Specify Host Names and IP Addresses

You can specify which host names, IP addresses, or IP address ranges you want DX NetOps Spectrum to discover using these three methods:

- Manual entry
- Import statically
- Import dynamically from a specified file location

Manual: Follow these steps:

1. Enter in the IP/Host Name Boundary List section (first text box) any *one* of the following values:

- A host name
- A single address
- The lowest address in the IP range

NOTE
You can use a wildcard character to input individual IP addresses. For example, entering 10.10.*.1 could discover: 10.10.0.1; 10.10.16.1; 10.10.32.1; and so on.

2. Enter in the second text box the same single host name or address, or the highest address in the IP range. Click Add.

WARNING
Attempting to process a large range of IP addresses or several sparsely populated subnets can lead to unwanted results.

3. Repeat Step 1 and Step 2 for each host name, IP address, or IP range of addresses that you want Discovery to contact.
You can also import a list of host names, IP addresses, or IP address ranges for DX NetOps Spectrum to discover.

Import statically: Follow these steps:

1. Click Import in the Configuration tab.
2. Select Local Host from the 'Import file location' drop-down list.
3. Select the text file containing the host names or IP addresses. Click Open.
OneClick Discovery reads the host names or IP addresses information from the selected file.

Import dynamically: Follow these steps:

1. Click Import in the Configuration tab.
2. Select ‘One Click web server host’ from the 'Import file location' drop-down list.
3. Enter the path to the text file containing the host names or IP addresses.
The file must be on the OneClick web server host. The file path must use the native format of the OneClick web server operating system (OS):
   - If the web server is running a Microsoft Windows OS, the format of the path must be:
     C:\Program Files\Spectrum\IP_Files\core_network_ips
   - If the web server is running a Linux OS, the format of the path must be:
     /usr/Spectrum/IP_Files/core_network_ips

NOTE
By default, the installation path for the OneClick web server appears in the Import file path field.

4. Click Open.
OneClick reads the file and imports the host names or IP addresses during each configuration activation. The host names or IP addresses in the text file can be updated regularly, and the updates are reflected in each activation. Importing dynamically lets you maintain current host names or IP addresses automatically when activating a configuration on a scheduled basis.

IP Address Considerations

When specifying IP addresses in the IP/Host Name Boundary List fields, consider the following situations, which can lead to unwanted results:

- Attempting to process a large range of IP addresses. Consider the inclusive range when entering IP address boundaries, and be as specific as possible. You get better results by entering multiple smaller and more pertinent ranges than a single large range. For example, do *not* attempt to run a Discovery with a single IP range of 0.0.0.1 to 255.255.255.255.
- Attempting to discover several sparsely populated subnets. This situation takes significant time because of the timeouts and retries for each unused address. Although many threads are involved in this process, a sparsely

populated subnet can quickly exhaust all the available threads. This situation causes the discovery process to take a long time. In this case, seed router discovery can be a better choice.

SNMP Information

The SNMP Information section is mandatory for Discovery configurations. Here you can review, edit, or remove SNMPv1 and v2c SNMP community strings and v3 security parameters that are currently applied to the configuration. You can also add new strings and profiles manually or you can import them from a text file.

> **NOTE**
> You cannot create a profile name using Import. Instead, first create any desired profile names using the Edit SNMP v3 Profiles dialog before importing the text file. When importing a text file, DX NetOps Spectrum compares the SNMPv3 profile names to the existing profiles. These existing profiles were created manually using the Edit SNMP v3 Profiles dialog. If a profile name included in the text file being imported does not exist in DX NetOps Spectrum, an error message displays, and the import action fails.

Specify SNMP Information

The SNMP Information section is mandatory for all Discovery configurations. You can manually specify an ordered list of SNMP community strings and profiles for SNMPv1, v2c, and v3. Or, you can import a list of strings. By default, Discovery uses ‘public’ if no other SNMP community strings are specified.

> **NOTE**
> For SNMPv3 communication, use profiles.

Manual: Follow these steps:

1. Select either the SNMP v1 option or the SNMP v2c option in the SNMP Information section.
2. Type the SNMP community string name for the devices you want discovered in the SNMP Community String field and click Add.
   The SNMP community string is inserted into the available SNMP community strings and profile names list.

Import: Follow these steps:

1. Create and save a text file containing the SNMP community strings that you want to use for SNMP. Be sure to use the correct syntax.
2. Click Import in the SNMP Information section.
3. Select the text file that contains the SNMP community strings you want to import.
   Valid SNMP community strings and profile names that are imported are added to the available SNMP community strings and profile names list.

Syntax for Imported SNMP Communities and Profiles

You can use this syntax to create a text file that contains SNMP community strings for importing into Discovery configurations.

This syntax has the following format:

```
<name>,v<SNMP_version>
```

> **NOTE**
> The text file must list each SNMP community string and profile on a separate line.

- ***<name>***
  Defines the SNMP community string.
- ***<SNMP_version>***
  Defines the applicable SNMP version, either 1, 2, or 3.

NOTE
The version number for SNMPv1 community strings is optional.

Examples: SNMPv1

```
public
public,v1
```

Example: SNMPv2

```
public,v2
```

Example: SNMPv3

```
public,v3
```

Modeling Options

Modeling configuration settings determine how Discovery models discovered devices. By default, OneClick provides modeling configuration parameters that you can use and modify. Access these settings by clicking the Modeling Options button on the Discovery Configuration tab. Or click the Model button on the Discovery tab.

This Modeling Configuration dialog contains the following settings for configuring discovered models:

- **Destination Container**
  Specifies the topology view container where Discovery places discovered device models. You can select a container, such as a LAN container or the New Devices container.
  Default: Universe container
- **Modeling Layout**
  Specifies how Discovery places and arranges models in the Universe topology view.
  - **Placement**
    Specifies where models appear in the topology:
    - **Flat:** Discovery places all devices, including Layer 1 and Layer 2 in the Destination Container; no LAN containers are created.
    - **Hierarchical:** (Default) Discovery places all Layer 3 devices, LAN containers, and Wide Area Links in the Destination Container. Layer 1 and Layer 2 devices are placed in the proper LAN container (based on IP address) under the Destination Container. If Discovery cannot find the appropriate LAN container for Layer 1 or Layer 2 devices, these devices are placed in the Destination Container.
  - **Arrangement**
    Specifies how models are arranged in the topology:
    - Grid
    - Radial (Default)
    - Tree
- **Modeling Options**
  Specifies how Discovery models discovered devices:
  - **Create Wide Area Link Models**
    Determines whether Discovery creates a WA_Link model between the wide area linked interfaces of two routers. When this option is disabled, Discovery directly connects the linked interfaces.
    Default: Enabled
  - **Create LANs (IP Subnets)**
    Determines whether Discovery uses a LAN container when representing an IP Subnet. Discovery creates the LAN container during the Layer 3 mapping process for any router interface that routes to a local LAN.
    Default: Enabled
  - **Remove Empty LANs**

Determines whether Discovery destroys any empty LAN containers that the Create LANs (IP Subnets) option created.
Default: Disabled

– **Create “802.3” (Fanout)**
Determines whether Discovery models an 802.3 Fanout segment when DX NetOps Spectrum cannot make an accurate connection among three or more interfaces. This model represents the ambiguous connections among these interfaces. However, DX NetOps Spectrum uses network traffic data (IfInOctet and ifOutOctect statistics) when the Traffic Resolution protocol option is enabled. This protocol determines connections between interfaces and, in many cases, eliminates the need to model a Fanout.
Default: Disabled

Note: If you have 50 or more connections to a single Fanout model, consider changing this model to a Shared Media Link. The Shared Media Links must be modeled manually. These models can provide more control over fault management behavior when multiple connections are monitored. Unlike a Fanout model, Shared Media Links provide configurable thresholds for handling downstream connections that report problems. For example, a Fanout model reports a problem only when *all* downstream connections are down. However, a Shared Media Link can report the problem sooner, as when 60 percent of the downstream connections are down.

Create Physical Addresses

Determines whether to create a physical address model for a MAC address that is heard by a switch but not associated with any modeled device. The layer 2 mapper attempts to find a connection for each address found. If a connection is found, a Fanout is created and the physical address is associated to it through Connects_To. If no connection is found, the model is placed in Lost and Found. This option is not recommended.

Default: Disabled

- **New Devices in Maintenance Mode**
  Determines whether DX NetOps Spectrum places the newly discovered devices directly into maintenance mode.
  Default: Disabled
- **Activation Timeout (in minutes)**
  Determines the number of minutes that Discovery waits for new models to activate before mapping their connectivity. When the timeout expires without any new devices activating, connectivity is established to the extent possible. Connectivity occurs regardless of whether all connections to discovered devices have activated. The minimum activation time is 5 minutes and the maximum time is 15 minutes.

Protocol Options

Lets you configure options for mapping the connectivity between models. By default, Discovery enables several protocol options that are based on best practices. You can disable the default settings or can enable other protocol settings.

NOTE
IPv6 MIB data is not used for connection mapping.

Network Services Options

Let you specify the network services to run during the modeling process. The supported options include VPN, Enterprise VPN, QoS, Multicast, VPLS, and MPLS Transport. Options are available depending on the components that you have installed.

Filter

Opens the Advanced Filter dialog, where you can exclude selected discovered devices from modeling. You can also click Show Advanced to create a complex filter criterion that includes a combination of AND/OR clauses. The ‘Hints’ link on the Advanced Filter dialog provides more information.

Auto Export

Specifies whether and how you want to export modeling results automatically. Also lets you select the format for exporting them: comma-delimited, tab-delimited, or web page.

Reset Defaults

Instructs Discovery to use the default modeling settings that are provided with DX NetOps Spectrum.

Define Modeling Options

The modeling configuration settings determine how Discovery models the devices it discovers. By default, OneClick provides modeling configuration parameters that you can use or modify. At any time, you can review or change the modeling configuration. Change the configuration using one of these methods:

- Click the Modeling Options button on the Discovery Configuration tab.
- Click the Model button on the Discovery tab.

NOTE
Before you can define a modeling configuration, define a Discovery configuration.

Follow these steps:

1. In the Discovery Console, with the current Discovery configuration selected in the Navigation panel, do *one* of the following steps:
   – Define a combined Discovery and modeling session by selecting 'Discover and automatically model to DX NetOps Spectrum.' To review or modify the modeling configuration, click the Modeling Options button in the Configuration tab.
   – Define a Modeling session after activating a Discovery session by clicking the Model button in the Discovery tab.

   The Modeling Configuration dialog opens.
2. In the Modeling Configuration dialog, review or modify any of the fields as needed.
3. How you accessed the Modeling Configuration dialog in Step 1 determines which of these Discovery options you can select:
   – If you clicked Modeling Options on the Discovery Configuration tab to access the Modeling Configuration dialog, click OK. All of your changes are saved, and the Modeling Configuration dialog closes.
   Upon activating the next modeling session, Discovery uses the last saved modeling parameters.
   – If you clicked Model on the Discovery tab to access the Modeling Configuration dialog, click OK. The currently specified modeling configuration parameters are saved, modeling session is activated, and the Modeling Configuration dialog closes.

   Discovery starts a modeling session that is based on the parameters you have specified.

Advanced Options

Configure Advanced Options if you want to perform any of these Discovery configuration procedures:

- Define the SNMP ports in addition to the default port (161).
- Exclude certain IP addresses from the Discovery.
- Modify the default settings for ICMP, route tables, throttle, time-out, and retries.
- Enable or disable the automatic export of Discovery results.

Follow these steps:

1. Click Advanced Options.
   The Advanced Options dialog opens.
2. (Optional) Enter a new port number in the SNMP Ports text box and click Add.
   The new SNMP port appears in the list of ports Discovery uses for this configuration.
3. (Optional) Enter an IP address or range to the IP Exclusion List text box.
   The IP addresses you enter are excluded from this Discovery.
4. Modify the default Discovery Options settings as needed. These options are as follows:
   – **Use ICMP first, then SNMP**

When this option is enabled, Discovery uses ICMP when discovering devices. If ICMP is enabled, Discovery pings the devices in the ranges/subnets first. The devices that responded to ICMP are then queried using SNMP. This option can help reduce the number of SNMP requests, especially when multiple SNMP community strings are being used.
Default: Enabled

- **Use Route Tables**
  Use this option only if seed routers are specified in the Discovery configuration. When this option is enabled, Discovery finds neighbor routers and the routed subnets from the IP route tables.
  Default: Enabled
- **Require discovered IP entry in device's IP Address Table**
  When this option is enabled, Discovery includes only those devices which have the discovered IP address present in their IP address table.
  To discover devices that do not have their discovered IP address in the IP address table (e.g. devices using NAT addresses) this option must be disabled. Disabling this option ignores IP address table checking for range discovery.
  Default: Enabled
- **Throttle**
  Most often this option applies to networks with routing tables containing more than 1,000 entries. If you have networks with routing tables containing over 1,000 entries, you can specify a throttle value (Low, Medium, or High) to stagger the processing workload by having DX NetOps Spectrum pause for one second after reading every 50 entries (High), 100 entries (Medium), or 250 entries (Low).
  Default: None
- **Timeout (in seconds)**
  Specifies the number of seconds that Discovery waits for a device response.
  Default: 3
- **Retries**
  Specifies the number of attempts that Discovery makes after the first attempt times out before establishing contact.
  Default: 3
- **ICMP Payload Size**
  Specifies the payload size in bytes. This option is only available if you have selected 'Use ICMP, then SNMP.'
  Default: 8
- **Secure Domain**
  This option is only available if you have the Secure Domain Manager installed.

  NOTE
  For information about the Secure Domain option, see Secure Domain Manager.

5. Do *one* of the following steps in the Auto Export section:
   - To disable Auto Export, select 'Do not export results' from the Auto Export drop-down list.
     Auto Export is not enabled for this Discovery and results are not exported automatically.
   - To enable Auto Export, select *one* of the following options from the Auto Export drop-down list:
     - Export results as CSV (Comma delimited)
     - Export results as text (Tab delimited)
     - Export results as a web page

     Auto Export is enabled for this Discovery. The Discovery results are sent to the location identified on the dialog and in the format that you selected.
6. Click Close.
   The Advanced Options dialog closes and your settings are saved to this Discovery configuration.

Scheduling Options

The scheduling options are as follows:

- Select an existing schedule to run the configuration (click the Select button in the Scheduling Options section).
- Create a schedule.

If the configuration is scheduled, the schedule name displays next in the Scheduling Options section.

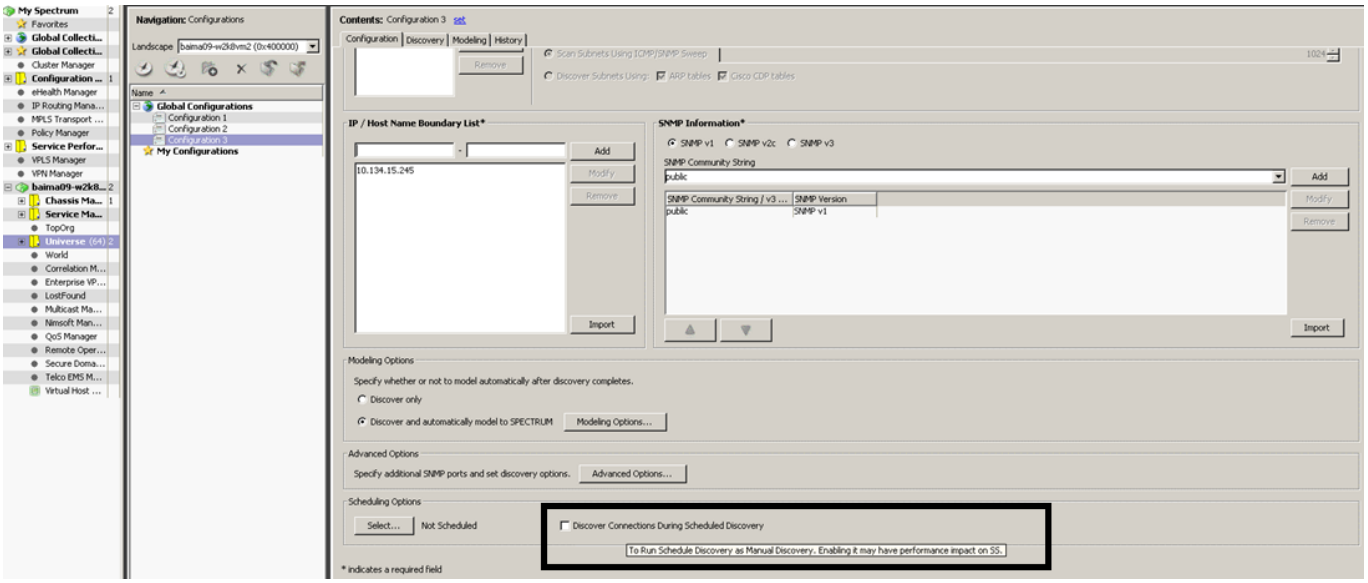
NOTE
For more information about setting schedules, see Using OneClick .

Discover Connections During Scheduled Discovery

Use the Discover Connections During Scheduled Discovery option to run the scheduled discovery as a manual discovery. Unlike normal schedule discovery, this option discovers and models all the devices (with their connections) that are specified in the respective configuration IP range, irrespective of the existing discovered devices. You can access this option from the OneClick Discovery Console.

NOTE
By default, the Discover Connections During Scheduled Discovery option is not selected. If you want to run the schedule discovery as manual discovery, select this option. You may experience a performance impact on SpectroSERVER.

The following image displays the Discover Connections During Scheduled Discovery option that is available in the Discovery Console:



Activate a Discovery Session

Activating a Discovery Session

If you have Discovery operations privileges, you can activate a Discovery session by clicking Discover in the Configuration tab. In addition, if you have privileges to modeling operations, you can click Model to activate a Discovery session from the Modeling tab. The following procedure provides instructions for activating a Discovery session for an existing Discovery configuration using the Discovery Console.

NOTE
If you do not have a Discovery configuration that is ready to activate, Define a Discovery Configuration before activating a Discovery session. We recommend that you also review and make any necessary changes to the Discovery configuration before using this procedure to activate a Discovery session.

Follow these steps:

1. Click Tools, Utilities, Discovery Console.
   The Discovery Console opens.
2. Click the name of the Discovery configuration for which you want to activate a Discovery session in the Discovery Navigation panel.
3. In the Discovery Configuration tab, click Discover to do either one of the following tasks:
   – To activate a Discovery session and/or combined Discovery and Modeling session.
   Discovery activates a Discovery session or a combined Discovery and modeling session that is based on the parameters that are specified in the Configuration tab. The results of the Discovery session appear in the Discovery tab. The results of the modeling session appear in the Modeling tab.
   – To rediscover an existing discovered or modeled configuration.
   All newly discovered results appear in the results list on the Discovery tab.

   NOTE
   Results from this new Discovery session overwrite the results of the previous Discovery session.

How to Discover Devices in Your Network

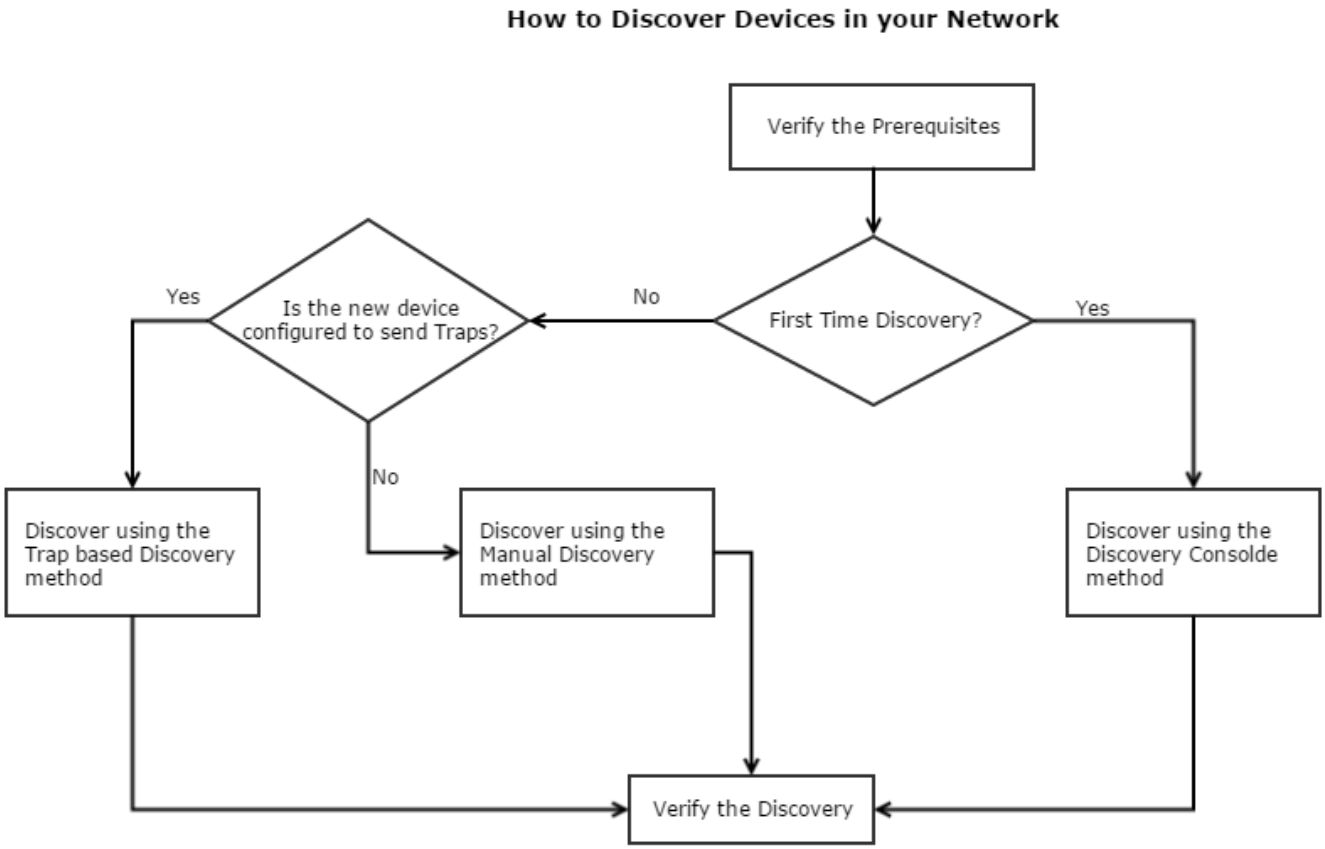
As a DX NetOps Spectrum administrator, you are responsible for discovering all of the devices in a network. Your goal is to create a DX NetOps Spectrum landscape. A landscape is all the data that is specific to any one virtual network machine (VNM) in a single network. The term also identifies the network domain that is managed by a single SpectroSERVER. In the OneClick console, a landscape is the network view of one SpectroSERVER. DX NetOps Spectrum gathers the MAC address, IP address, and other important information about the devices in your landscape when you discover them. Discovery helps you perform the following tasks:

- Update device configurations
- Detect the root cause of a network breakdown
- Troubleshoot faulty devices

After devices are discovered in a network, DX NetOps Spectrum models them in real-time on the OneClick console, a web client application to view and manage network devices. The real-time modeling of devices gives you a full view of a network, and lets you track conditions within the computing infrastructure.

The following diagram shows how to discover devices in your network so that you can monitor and manage them:

Figure 75: Discover devices in network



Running discovery the devices involves the following tasks:

- Verify the Prerequisites.
- Determine the Discovery Method.
- Discover Using the Full Discovery Method.
- Discover Using the Automatic Discovery Method.
- Discover Using the Manual Discovery Method.
- Verify the Discovery.

Verify the Prerequisites

Before you discover devices in your network, verify the following prerequisites:

- DX NetOps Spectrum is installed on your host.
- You are logged in to the OneClick console with discovery privileges.
- The OneClick console displays your DX NetOps Spectrum landscape with your host name on the Navigation panel.
- The Universe container exists in your landscape.
  The Universe container helps you organize an enterprise network view of your infrastructure.
- You know how to configure your network devices to send traps to DX NetOps Spectrum.

Determine the Discovery Method

Your goal is to discover and model devices in your network. Select one of the following discovery methods depending on whether you are discovering devices for the first time, or you are discovering new devices in your network:

Discovery Console Method
Select this method when you want to discover a new network and all of its existing devices for the first time. Use the Discovery Console method after installing DX NetOps Spectrum. This method takes a long time to discover all the existing devices depending on the size of the network.

Trap Based Discovery Method
Select this method when new devices are configured to send traps to DX NetOps Spectrum. As soon as you add the device to the network, the device sends a trap and DX NetOps Spectrum automatically discovers the device.

Manual Discovery Method
Select this method when you want to manually discover existing devices in a network that are not configured to send traps. Use the manual discovery method when you know the IP address or the model type of the new device. This method is best for discovering a new device quickly.

Discover Using the Discovery Console Method

Use the Discovery Console method to discover a network and its existing devices for the first time. In this method, you specify the discovery parameters (for example, IP Address range, SNMP version) in a configuration file to discover the network and its devices.

> **NOTE**
>
> The devices that are configured with SNMP v3 profiles are discovered as unknown device types. To discover devices for which SNMP v3 profiles are configured, select the SNMP v3 option. Create an SNMP v3 profile with parameters or select an existing profile. The devices that are configured with SNMP v3 profiles are discovered only when the DX NetOps Spectrum and the device SNMP v3 profiles match.

Follow these steps:

1. Log in to the OneClick console.
2. Select Explorer from the Navigation pane, and perform the following actions:
   a. Expand your landscape.
   b. Click Universe.
3. Select Topology, and click Discovery on the toolbar.
   The Discovery Console opens.
4. Click Create a new folder.
5. Perform the following actions to create and save a configuration file in this folder:
   a. Expand Global Configurations.
   b. Right-click the new folder, and select Create.
   c. Enter the configuration file name.
      Retain the default value of the Landscape field, and click OK.
   The configuration file is created, and the Configuration tab on the Contents pane of the Discovery Console is activated.
6. Perform the following actions to configure the discovery parameters:
   a. Enter the IP address ranges of all the subnets in the IP/Host name boundary List pane.
   b. Select the SNMP Communication option from the SNMP Information pane.
   c. Add all the SNMP Community Strings in the SNMP Information pane.
   d. Select the required Discovery Type from the Seed Routers pane.
   e. Select 'Discover and automatically model to Spectrum' as the Modeling Option.
7. To discover SNMP v3 configured devices, perform the following actions:
   a. Select SNMP v3 and click Profiles.

The Edit SNMP v3 Profiles dialog opens.
   b. Choose an existing profile from the list or create a profile.
   c. To create a profile, enter the Profile Name, User ID, Authentication Type, and click Add.
8. Click Discover.

DX NetOps Spectrum discovers and models the complete network and its existing devices in the OneClick console.

Discover Using the Trap-Based Discovery Method

After you perform a full discovery using the full discovery method, you can discover new devices that are added to the network using the automatic discovery method. With this method, you enable DX NetOps Spectrum to process unmanaged traps coming from the devices and the traps are used to discover the devices automatically.

Follow these steps:

1. Log in to the OneClick console.
2. Select Explorer from the Navigation pane and perform the following actions:
   a. Expand your landscape.
   b. Click Universe.
3. Select Topology, and click the VNM model type.
   The VNM model on the OneClick console represents the host computer running DX NetOps Spectrum.
4. Perform the following actions in the Component Detail pane:
   a. Select Information.
   b. Expand AutoDiscovery Control.
   c. Expand Trap Based Continuous Discovery.
   d. Enable Unmanaged Trap Discovery.

The new devices are discovered and modeled. You can view them in the OneClick console.

NOTE

Traps from the already discovered and modeled devices are not considered for discovery.

Discover Using the Manual Discovery Method

When a new device is not configured to send a trap to DX NetOps Spectrum, you can discover the device using the manual discovery method. In this method, you specify the IP address, model type, or the host name to discover the device.

NOTE

To discover devices on which SNMP v3 profiles are configured, select the SNMP v3 option. The devices that are configured with SNMP v3 profiles do not respond to the ordinary discovery. Create an SNMP v3 profile with parameters or select an existing profile. The devices that are configured with SNMP v3 profiles are discovered only when the DX NetOps Spectrum and the device SNMP v3 profiles match.

Follow these steps:

1. Log in to the OneClick console.
2. Select Explorer from the Navigation pane and perform the following actions:
   a. Expand your landscape.
   b. Click Universe.
3. Select Topology from the Contents pane.
4. On the toolbar, click Create new model by type, Create new model by IP, or Create By Host Name.
   The Select Model Type or the Create Model By IP Address dialog opens.

5. Perform the following actions to configure the discovery parameters:
   a. Enter the name, IP address, and SNMP Community String.
   b. Set the values for the Poll Interval (sec), Agent Port, and Secure Domain.
   c. Select the SNMP Communication Option.
6. To discover SNMP v3 configured devices, perform the following actions:
   a. Select SNMP v3 and click Profiles.
      The Edit SNMP v3 Profiles dialog opens.
   b. Choose an existing profile from the list or create a profile.
   c. To create a profile, enter the Profile Name, User ID, Authentication Type, and click Add.
7. Click OK.

DX NetOps Spectrum discovers and models the device. You can view them in the OneClick console.

Verify the Discovery

You verify the discovery process by checking the model and MAC address of the discovered devices using the OneClick console.

Follow these steps:

1. Log in to the OneClick console.
2. From the Contents pane, click Topology.
3. Verify that the discovered devices are modeled in real time.
4. Click a device model
5. From the Component Detail pane, select Information.
6. Expand General Information.
7. Verify that the MAC address is present for the device model.
8. Click all the device models that are created in order, and verify their MAC addresses.

You have discovered devices in your network and verified the discovery. You can now manage and monitor your network and its devices.

Activate a Modeling Session

You can model your Discovery results during the Discovery session, or you can save the configuration and model the results later. You can specify how DX NetOps Spectrum models discovered devices that appear in the Discovery tab results list. In the Modeling Configuration dialog, you can accept the default modeling options, or you can change them to meet your requirements.

You can activate a modeling session in the following two ways:

- Select the ‘Discover and automatically model to DX NetOps Spectrum’ option in the Configuration tab before Discovery runs. A modeling session runs automatically with this option selected.
- Click Model in the Discovery tab after a ‘Discover only’ session has run for that configuration.

Prerequisites to Activating a Modeling Session

- Activate at least one Discovery session of an existing Discovery configuration.
- Review and make any necessary changes to the modeling configuration.
- (Optional) Exclude certain devices from being modeled.
- Ensure that you have sufficient privileges for activating a modeling session.

Follow these steps:

1. In the Discovery Console, click the name of the Discovery configuration for which you want to activate a modeling session in the Navigation panel.
2. Do *one* of the following tasks:
   – In the Configuration tab, select the 'Discover and automatically model to DX NetOps Spectrum' option, and click Discover.
   Discovery activates a Discovery session and then automatically models the discovered devices appearing in the Results tab of the Discovery tab.
   – In the Discovery tab, click Model to activate a modeling session and model the last set of discovered devices. The last set of discovered devices appear in the Results list of the Discovery tab.

Run a Network Services Discovery

You can discover devices for the following network services: VPN, Enterprise VPN, QoS, Multicast, VPLS, and MPLS Transport. Options are available depending upon the products you have installed.

> **NOTE**
> You can only run one Network Services Discovery at a time.

Follow these steps:

1. Select the models that you want to run a Network Services Discovery for:
   a. Expand any subnet, folder, and so on, in the Explorer tab or in the List tab, that contains the models you want to select.
   b. Click the List tab in the Contents panel and select the models by using CTRL+click.
2. Click Tools, Utilities, Network Services Discoveries, and then select the Network Services Discovery that you want to run.
   The Discovery executes for the selected models. A pop-up window appears, indicating success or failure of starting the Discovery. The Network Services Discovery fails only when a Discovery is already running for the selected product.
   The Discovery process can run for some time. You can check the status of the Discovery process:
   a. Select the product in the Navigation panel.
   b. Click the Information tab in the Contents panel.
   c. Expand the Configuration subview.
   d. Expand the Discovery subview.

> **NOTE**
> For more information about running a specific Network Services Discovery, see VPN Manager , Enterprise VPN Manager , QoS Manager , Multicast Manager or MPLS Transport Manager . For more information about VPLS, see Enterprise VPN Manager .

Create Discovery Configuration Folders

Discovery configurations are stored in folders. You can create a folder for a new configuration or a copy of an existing configuration.

Follow these steps:

1. Click Tools, Utilities, Discovery Console.
   The Discovery Console opens.

2. Click
   
   (Creates a new folder).
   The New Folder dialog opens.
3. Type a name for this Discovery configuration.
   The new folder is displayed in the Navigation panel.

Reorganize Discovery Configurations

You can move configurations and folders to different folders using the drag-and-drop method.

Follow these steps:

1. Click Tools, Utilities, Discovery Console.
   The Discovery Console opens.
2. Select the configuration or folder you want to move and drag and drop it to the desired location.
   The configuration or folder is moved to the desired location.

Rename Discovery Configurations or Folders

You can change the original name of a Discovery configuration after a Discovery session has ended and you can rename Discovery configuration folders.

Follow these steps:

1. Click Tools, Utilities, Discovery Console.
   The Discovery Console opens.
2. Right-click the configuration or folder you want to rename and select the Rename node option.
3. Type the new name and click OK.
   The configuration or folder you selected displays its new name.

> **NOTE**
> You can also rename configurations by clicking ‘set’ beside the current configuration name in the Contents panel. However, you cannot rename folders from the Contents panel.

VLAN Discovery

Discovering Virtual Local Area Network

To discover Virtual Local Area Networks (VLANs) successfully, create a root container for each VLAN domain (in a VLAN domain, each VLAN ID is unique). All the devices in a VLAN domain must be placed in a unique root container.

If the devices from different VLAN domains are placed in the same container, VLAN Discovery on that container might not work properly. Also, the VLAN spotlighting might not be able to distinguish the VLANs and devices from the different VLAN domains.

View, Filter, and Export Results Lists

Each time that you activate a Discovery or modeling session, Discovery automatically places the results in the Discovery tab or the Modeling tab. You can use these results to select the devices you want to model or export. For more information, see the following sections:

- Filter, Sort, Export, Search, and Model Discovery Results
- Export a Results List
- Set a Modeling Results List to Export Automatically
- Filter Results Using Advanced Filter

Export a Results List

You can export a results list by clicking the Export button that appears on the Discovery or Modeling tab. The Export feature accesses the Export table data to file dialog where you can identify:

- Location to save the exported data file
- Name for the exported data file
- Type of file to use to export the data
- Type of file format to save the file

Follow these steps:

1. In the Discovery Console, click the Discovery tab or the Modeling tab.
2. Click the Export button.
   The 'Export table data to file' dialog opens.
3. Complete the following information:
   - **Save in**
     Specifies the location to save the exported data file.
   - **Save as type**
     Specifies the file type that you want to use when saving the exported data.
   - **File name**
     Defines the name for the exported data file.
   - **Files of type**
     Specifies the type of file format to use.
4. Click Save.
   The data is exported to the specified location, file name, and file format.

Set a Modeling Results List to Export Automatically

You can automatically export your Discovery or modeling results and status to a web server location.

Follow these steps:

1. In the Discovery Console, select the Discovery configuration for which you want to automate exporting of modeling results.
2. In the Configuration tab, select the 'Discover and automatically model to DX NetOps Spectrum' option.
3. Click the Modeling Options button.
   The Modeling Configuration dialog opens.
4. In the Auto Export section, do these steps:
   a. To export modeling results tables, select the 'results table' check box.
   b. Choose the exported files format from the results tables drop-down list: CSV (Comma delimited), text (Tab delimited), and web page.
   c. To export status data, select the 'status information (plain text only)' check box.
5. Click OK.

Filter Results Using Advanced Filter

Using the Advanced Filter dialog, you can create filters with compound clauses to exclude certain entries from appearing in the Discovery or modeling results list. If you have privileges to both Discovery and modeling operations, you can access

the Advanced Filter dialog before activating a combined Discovery and modeling session. If you have privileges to the Discovery operations only, you can access the Advanced Filter dialog after initiating a Discovery session.

NOTE
Discovery uses the results list on the Discovery tab to determine which devices to model or export.

Follow these steps:

1. Before you perform a Discovery, do these steps:
   a. In the Discovery Console, in the Configuration tab, select the Discover and automatically model to DX NetOps Spectrum option, and then click Modeling Options.
   The Modeling Configuration dialog opens.
   b. Click the Filter Options button.
   The Advanced Filter dialog opens.
   c. Go to Step 3.
2. After a Discovery session, do these steps:
   a. In the Discovery Console, click the Discovery tab.
   b. In the Discovery tab, click the Advanced Filter button.
   The Advanced Filter dialog opens.
   c. Go to Step 3.
3. In the Advanced Filter dialog, complete the fields as follows to create a single expression filter.
   – **Attribute/Ignore Case**
   Select a device attribute to filter.

   NOTE
   If you choose an alphabetic attribute value, you can either clear (ignore) or select (include) the Ignore Case check box.
   – **Comparison Type**
   Select the type of comparison to make against the value that is specified in the Attribute field.
   – **Attribute Value**
   Enter or select the desired attribute value to filter.
4. To filter a single expression, click OK.
   The Advanced Filter option excludes entities in the results list using the filter parameters that you specified.
   Build a compound clause:
5. Click Show Advanced.
   The compound expression box and logical operator buttons appear.
6. Click Add to move the single expression (created in Step 2) to the compound expression box.
7. Click *one* of the following logical operator buttons to build a compound expression: New AND; New OR; or AND/OR.
   – The compound expression is represented in a tree structure that is grouped by logical operators (AND/OR). Each logical operator in the tree can include any number of attribute criteria nodes and logical operator nodes. For more information, click Hints.
   – Alternatively, you can create advanced search expressions using prefix notation.
8. Repeat Step 5 and Step 6 for each compound expression you want to build.
9. Click OK after building the expressions.
   The Advanced Filter mechanism excludes the entities in the results list using the compound filter expressions applied.

Importing and Exporting Discovery Configurations

Import Discovery Configurations

You can import multiple Discovery configurations from your computer, in XML format, into DX NetOps Spectrum.

Follow these steps:

1. Click Tools, Utilities, Discovery Console.
   The Discovery Console opens.
2. Click
   
   (Import).
   The Open dialog opens.
3. Browse for the Discovery configuration you want to import from your local computer and click Open.
   The Discovery configuration is imported.

Export Discovery Configurations

You can export multiple Discovery configurations, in XML format, to your computer.

Follow these steps:

1. Click Tools, Utilities, Discovery Console.
   The Discovery Console opens.
2. Expand the folder where the configuration you want to export is located in the Navigation panel and select the configuration.
3. The configuration information appears in the Contents panel.

   NOTE
   You can select multiple configurations. If you select a folder with multiple configurations, all of those configurations are exported. However, the folder hierarchy is not exported.
4. Click
   
   (Export).
   The Export to file dialog opens.
5. Select the export location on your local computer, enter a configuration name in the File name field, and click Save.
   The Export Results from Export dialog opens.
6. Click OK.
   The Discovery configuration is exported.

After Discovering and Modeling

DX NetOps Spectrum does not support all possible network devices with model types and management modules. To get the DX NetOps Spectrum environment you need for managing your network, you can modify the Discovery and Modeling results.

After you successfully discovered and modeled your network, you can examine and enhance the results, for example:

- Modify the device type names to reflect the devices on your network accurately.

  NOTE
  For more information about device types, see Certifications.
- Modify the attributes for device models and model types using the Attribute Editor or the Attributes tab.
- Create model types for devices that the DX NetOps Spectrum model types and management modules do not directly support.

NOTE
For more information, see Certifications.

- Import MIBs using the OneClick MIB Tools utility to get updated MIBs for devices, and for features that the provided DX NetOps Spectrum MIBs do not support.

  NOTE
  For more information about using MIB Tools, see Certifications.
- Modify the names of the existing Discovery configurations.

VNM AutoDiscovery Control Settings

AutoDiscovery control settings available on the VNM Information tab affect some of the actions that occur during Discovery and Modeling sessions.

If you have a Distributed SpectroSERVER (DSS) environment with multiple SpectroSERVERs, apply all settings changes to *all* of your SpectroSERVERs.

Access VNM AutoDiscovery Control

To access the VNM AutoDiscovery Control subview, select the VNM in the Explorer tab or in a Universe topology view.

Then select the Information tab in the Component Detail panel.

Loopback Interfaces and Discovery

You can set DX NetOps Spectrum to use a loopback interface as a primary agent address.

You can also specify which loopback interface to use when modeling devices.

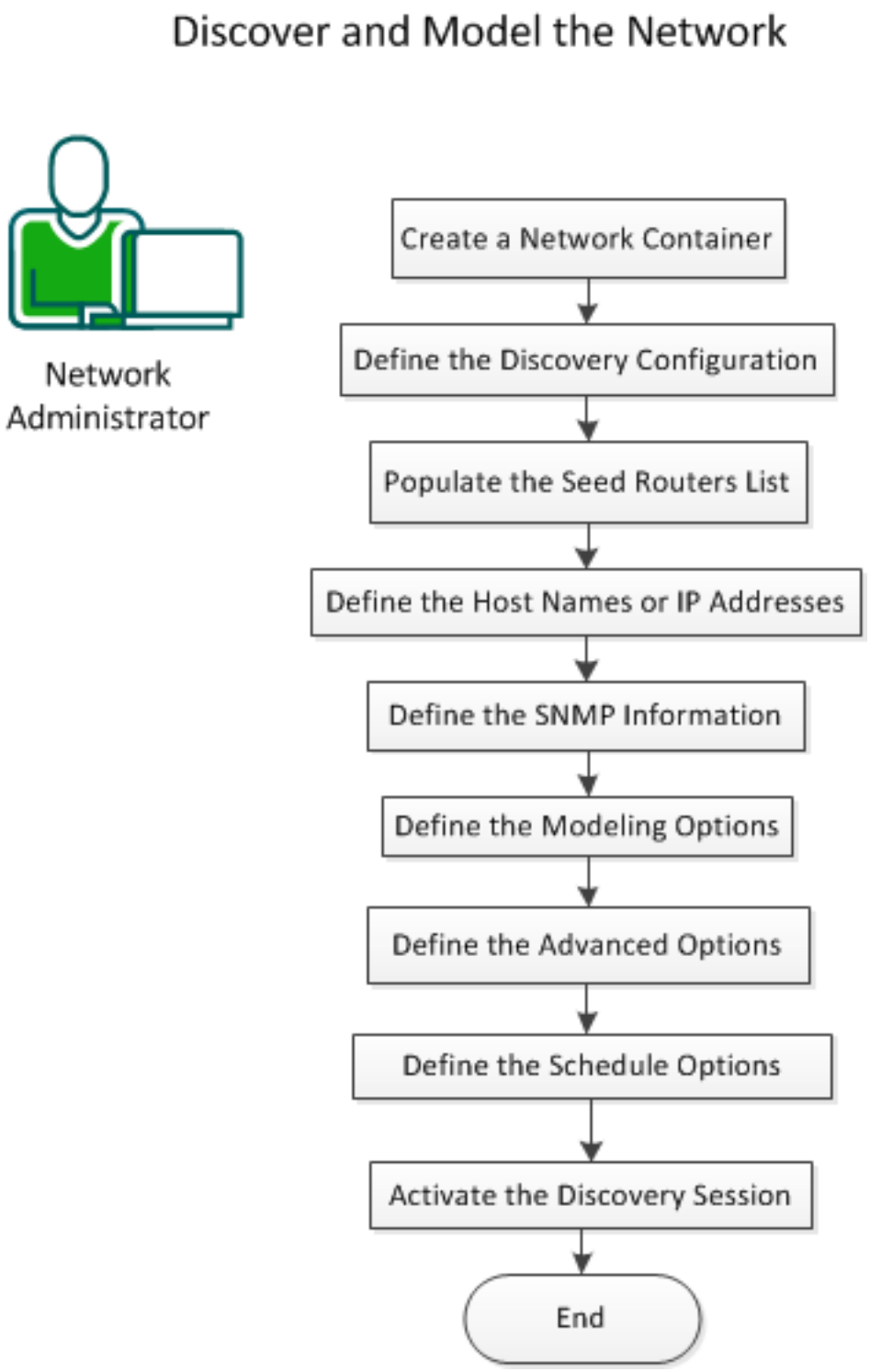
How to Discover and Model the Network

DX NetOps Spectrum is a services and infrastructure management system that monitors the state of managed elements including devices, applications, host systems, and connections. DX NetOps Spectrum includes some client applications. OneClick is the client application that provides the graphical user interface to monitor the network and launch other client applications. You can create and maintain the software models of a network by using the modeling features of OneClick.

This scenario describes how a network administrator uses the Discovery utility of OneClick to define and discover models that represent entities in the IT infrastructure.

The OneClick client application provides the graphical user interface to monitor the network and launch other client applications. Run the Discovery utility of OneClick to define and discover models that represent entities in the IT infrastructure.

The following diagram describes how to use Discovery:



Create a Network Container

A container represents the group of devices that you want to model. Containers help you monitor and manage the health of the devices that they represent. Create a container in the Universe topology.

Follow these steps:

1. Expand <*usxxyyzz*> on the **Explorer** tab in the **OneClick Navigation** pane and select **Universe.**
   The <*usxxyyzz*> hierarchy identifies the local **SpectroSERVER** landscape.
2. Select **Topology** in the **Contents** pane.
3. Select the **Model by Type** icon, select **Network,** and select **OK.**
4. Complete the following fields in the **Create Model of Type Network** dialog:
   Name - Defines the unique host name for the device that you are modeling.(Optional)
   Network Address - Specifies the IPv4 or IPv6 address for the device that lets DX NetOps Spectrum communicate with the device.
   (**Optional) Security String** - Specifies the security for the device. Adding a security string prevents certain users from viewing the model.
   (**Optional) Subnet Mask** - Specifies the device subnet address that the container represents. When specified, the subnet address label appears whenever a user points the mouse to a container icon in a topology view.
5. Select **OK.**

Define the Discovery Configuration

Discovery uses a set of configuration parameters that you can define to determine the network entities to discover and model.

Follow these steps:

1. Select the network container on the Topology tab.
2. Right-click on the selected container and select Utilities, Discovery Console.
3. Type a name for the configuration.
4. Select a landscape.
5. Specify the folder in which the new configuration must be created.
6. Select OK.
   The procedures that follow let you configure the parameters.

Populate the Seed Routers List

> **NOTE**
> The Seed Routers section is optional. We recommend that you populate the seed routers list for large Discovery operations.

Seed routers are a list of core routers that Discovery uses as a starting point when determining the routed subnets. All routers that are discovered within the IP / Host Name Boundary List are treated as seed routers. Add the seed routers list to let Discovery determine the routed subnets from the list.

Follow these steps:

1. Type an IP address or host name in the **Seed Routers** section and select **Add.**
2. (Optional) To create a range of IP addresses or host names, repeat Step 1.
3. Select one of the following options from **Discovery Type**:
   Routers only - Discovers the routing devices within the IP address range or host names. Routers and only local LANs in IP/Host Name Boundary ListDiscovers the routed subnets within the IP range or host name.
   Routers and all local LANs - Discovers the subnets that are routed by the routers that are discovered within the IP range or host name.
4. If you selected '**Routers and only local LANs in IP/Host Name Boundary List**' or '**Routers and all local LANs**', select one of the following options:

- – **Scan Subnets Using ICMP/SNMP Sweep** - Scans the subnets that use ICMP or SNMP sweep. If you select this option, set the maximum subnet size in the counter.

- – **Discover Subnets Using** - Discovers the subnets that use the following tables:
- – • **ARP tables** - Specifies that the subnet uses the Address Resolution Protocol (ARP) tables.
  • **Cisco CDP tables** - Specifies that the subnet uses the Cisco Discovery Protocol (CDP) tables.

Define the Host Names and IP Addresses

The IP / Host Name Boundary List field populates with an IP range when you start Discovery with a container that is selected from the Topology tab. Import the host names, IP addresses, or IP address ranges from a file on the OneClick web server.

Follow these steps:

1. Select **Import** in the **IP / Host Name Boundary List** section.
2. Select **'One Click web server host'** from **'Import file location'**.
3. Type the path to the text file containing the host names or IP addresses in one of the following formats:
   Microsoft Windows: C:\Program Files\Spectrum\IP_Files\core_network_ips
   Linux: /usr/Spectrum/IP_Files/core_network_ips
4. Select **OK.**

Define the SNMP Information

Import an ordered list of SNMP community strings for SNMP v1, SNMP v2c, or profiles for SNMPv3.

Follow these steps:

1. Create a text file that lists each SNMP community string and profile in the following format:
   `<name>v<SNMP_version>`
   Defines the SNMP community string. Defines the applicable SNMP version.
   Note: The version number for SNMPv1 community strings is optional.
   Examples: SNMPv1publicpublic, v1Example: SNMPv2public, v2Example: SNMPv3public, v3
2. Select Import in the SNMP Information section.
3. Select the text file that you created in Step 1.
4. Select Open.

Define the Modeling Options

Define the modeling configuration settings to combine the Discovery and modeling sessions.

Follow these steps:

1. Select '**Discover and automatically model to SPECTRUM**'.
2. Select **Modeling Options**.
3. Specify the destination container by performing the following steps:
   a. Select **Destination Container**.
   b. Select the topology view container to place the discovered models.
   c. Select **OK.**
4. Configure the modeling layout by performing the following steps:
   a. Select one of the following options from Placement to specify where models appear in the topology:
      Flat Discovery places all devices, including Layer 1 and Layer 2 in the Destination Container; no LAN containers are created.
      Hierarchical Discovery places all Layer 3 devices, LAN containers, and Wide Area Links in the Destination Container. Layer 1 and Layer 2 devices are placed in the proper LAN container (based on IP address) under the

Destination Container. If Discovery cannot find the appropriate LAN container for Layer 1 or Layer 2 devices, these devices are placed in the Destination Container.

b. Select one of the following options from Arrangement to specify the arrangement of models in the topology:
Grid arranges the models in a grid pattern.
Radial arranges the models in a radial pattern.
Tree arranges the models in a tree structure.

5. Configure the modeling options by selecting the required options in the **Modeling Options** section:

- **Create Wide Area Link Models**

Determines whether Discovery creates a WA_Link model between the wide area linked interfaces of two routers. When this option is disabled, Discovery directly connects the linked interfaces.

- **Create LANs (IP Subnets)**

Determines whether Discovery uses a LAN container when representing an IP Subnet. Discovery creates the LAN container during the Layer 3 mapping process to connect a router to a local LAN.

- **Remove Empty LANs**

Determines whether Discovery removes any empty LAN containers that the Create LANs (IP Subnets) option creates.

- **Create “802.3” (Fanout)**

Determines whether Discovery models an 802.3 Fanout segment when DX NetOps Spectrum cannot make an accurate connection among three or more interfaces. This model represents the ambiguous connections among these interfaces. However, DX NetOps Spectrum uses network traffic data **(IfInOctet** and **ifOutOctect** statistics) when the Traffic Resolution protocol option is enabled. This protocol determines connections between interfaces and, often, eliminates the need to model a Fanout.

NOTE

- Because DX NetOps Spectrum cannot identify the actual traffic flown path link between a device and fanout, when you try to delete one of the links, Spectrum deletes all the links that are connected to fanout.
- If you have 50 or more connections to a single Fanout model, consider changing the model to a Shared Media Link. The Shared Media Links must be modeled manually. These models can provide more control over fault management behavior when multiple connections are monitored. Unlike a Fanout model, Shared Media Links provide configurable thresholds for handling downstream connections that report problems. For example, a Fanout model reports a problem only when all downstream connections are down. However, a Shared Media Link can report the problem sooner, as when 60 percent of the downstream connections are down.

- **Create Physical Addresses**

Determines whether to create a physical address model for a MAC address that a switch hears but not associates with any modeled device. The layer 2 mapper attempts to find a connection for each address found. If a connection is found, a Fanout is created and the physical address is associated to it through Connects_To. If no connection is found, the model is placed in Lost and Found.

WARNING
Important! We recommend that you do not enable this option to avoid performance issues.

- **New Devices in Maintenance Mode**

Determines whether DX NetOps Spectrum places the newly discovered devices directly into maintenance mode.

- **Create Wireless Access Points**

Determines whether the Discovery creates models of Wireless Access Points (**WAP**s) to a particular controller given in the Discovery list. SNMP disabled APs and APs which are not mentioned in the discovery list are modeled as **AccessPoint** models (**AccessPoint models**).

- **Reconfigure Interfaces**

Determines whether the Discovery/Schedule Discovery re-configures the interfaces of the devices which are already modeled in the network for the given IP range. Select this check box to reconfigure the interfaces of the existing devices with the discovery of new devices.

- **Activation Timeout (in minutes)**

Determines the number of minutes that Discovery waits for new models to activate before mapping their connectivity. When the timeout expires without any new devices activating, connectivity is established to the possible extent. Connectivity occurs regardless of whether all connections to discovered devices have activated.Limits: 5-15 minutes.

1. Specify the protocol that is used in connectivity-mapping process by performing the following steps:
   a. Select **Protocol Options**.
   b. Select the required options:
      IP Address Tables - Determines whether to use IP address tables when mapping. When disabled, Discovery disables Layer 3 mapping and maps only the Layer 2 connections. Additionally, Discovery disables the following options:
      - IP Route Tables Protocol
      - Create Wide Area Link Model
      - Create LANs (IPSubnets)
      - Remove Empty LANs.

      IP Route Tables - determines whether to use the IP Address Table when mapping routers. By default, this option is not selected because these tables can be large and can take time to read. When this option is enabled, DX NetOps Spectrum cannot map unnumbered IP interfaces (0.0.0.0).
      Source Address Tables - Determines whether Discovery uses the device Source Address table when mapping layer 2 connectivity.
      Discovery Protocol Tables - Determines whether Discovery uses the Discovery Protocol tables when mapping device connectivity. Supported discovery protocols include Cisco, Nortel, Cabletron Switch, Extreme, Alcatel, Foundry, and Link Layer.
      ARP Tables for Pingables - Determines whether Discovery uses the ARP table when determining MAC addresses that can be used for the connectivity mapping.
      Spanning Tree - Determines whether Discovery uses the device Spanning Tree Address table (SAT) when mapping Layer 2 connectivity information about the device.
      Traffic Resolution - Determines whether Discovery uses network traffic data when determining connections between interfaces. Often, letting Discovery to use the traffic data eliminates the need to model Fanout segments.
      ATM Protocols - Determines whether the ATM Discovery runs against all ATM switches in the SpectroSERVER database.
      Note: For more information about ATM Protocols, see ATM Circuit Manager .
   c. Select OK in the **Protocol Options** dialog.
2. Configure the **Network Services Options** by performing the following actions:
   a. Select **Network Services Options**.
   b. Select the options to specify the network services to run during the modeling process.
      Note: OneClick displays the options that are based on the installed components.
   c. Select OK.
3. Configure the filtering options by performing the following actions:
   a. Click **Filter Options**.
   b. Specify the parameters to exclude selected discovered devices from modeling.

c. (Optional) select **Show Advanced** and create a complex filter criterion that includes a combination of the AND/OR clauses.
d. (Optional) select the ‘Hints’ link for more information.
e. Select OK in the **Advanced Filter** dialog.

4. (Optional) select 'results table' in the Auto Export section and specify the file format in the drop-down to export modeling results automatically.
5. (Optional) select 'status information (plain text only)' in the Auto Export section to automatically export the status information of the discovered devices.
6. Select OK in the **Modeling Configuration** dialog.

(Optional) Define the Advanced Options

Define Advanced Options to perform *one* of the following Discovery configuration procedures:

- Define the SNMP ports in addition to the default port (161).
- Exclude a specific IP address from Discovery.
- Modify the default settings for ICMP, route tables, throttle, time-out, and retries.
- Manage the export of Discovery results.

Follow these steps:

1. Select **Advanced Options**.
2. Type a new port number in the **SNMP Ports** section and select **Add.**
3. Type an IP address or range in the **IP Exclusion List** section and select **Add.**
4. Modify the following settings in the **Discovery Options** section:
   Use ICMP first, then SNMPUses ICMP when discovering devices. If ICMP is enabled, Discovery pings the devices in the ranges or subnets first. The devices that responded to ICMP are then queried using SNMP. This option reduces the number of SNMP requests, especially when multiple SNMP community strings are being used. Use Route Tables identifies the neighbor routers and the routed subnets from the IP route tables. Use this option only if seed routers are specified in the Discovery configuration.ThrottleStaggers the processing workload of Discovery by pausing for one second after reading the routing table entries, if there are more than 1000 entries. Select High, Medium, or Low to pause after reading 50, 100, or 250 entries, respectively. Timeout (in seconds) Specifies the number of seconds that Discovery waits for the response from a device.RetriesSpecifies the number of attempts that Discovery makes after the first attempt times out before establishing contact.ICMP Payload Size Specifies the payload size in bytes. This option is only available if you have selected ‘Use ICMP, then SNMP.’Secure Domain specifies the secured domain.

   NOTE
   This option is available only if you have the Secure Domain Manager installed. For more information, see the Secure Domain Manager section.
5. Do one of the following steps in the **Auto Export** section:
   – To disable auto export, select ‘**Do not export results’**.
   – To enable auto export, select the file format that you want to export in, followed by the type of results you want to export.
6. Select **OK.**

(Optional) Define the Schedule Options

Configure the scheduling options to run the configuration. If the configuration is scheduled, the schedule name is displayed in the Scheduling Options section.

Follow these steps:

1. Click Select in the **Scheduling Options** section.
2. Create a schedule.

Activate the Discovery Session

Activate the Discovery session after defining all parameters.

Follow these steps:

1. (Optional) select '**Save options as default**' to save the current configuration settings as the default configuration.
2. Select **Discover** to activate the Discovery session.
3. Select **File, Close** when the modeling process is complete.
   The network is discovered and modeled.

The network entities and their connections are represented graphically in the Topology tab. Icons that are created, placed, and connected within the OneClick topology views represent various aspects of a modeled network.

Modeling Your Network Manually

This section contains information about how to monitor your network manually.

When to Model Manually in OneClick

When to Model Manually in OneClick

You most often perform manual modeling tasks when you want to represent one or more previously modeled Universe topology devices in other OneClick topologies. For example, OneClick topologies such as Global Collections, World, or TopOrg.

You can also model a network device in DX NetOps Spectrum manually after using Discovery in the Universe topology. For example, the Discovery feature is unable to discover new devices in your network that are temporarily offline or blocking management communication. To resolve this situation, you could rediscover these new devices later using Discovery, or you could manually add them to the Universe topology.

In the Universe topology, you can also manually:

- Add devices and annotations to existing models.
- Change device configuration information.
- Improve the readability of models by keeping the layers within the Universe topology simple.

DX NetOps Spectrum can model Fanouts automatically, but consider manually changing these Fanouts to a Shared Media Link model when the Fanout has more than 50 connections. The Shared Media Link models use configurable thresholds that can provide more control over fault management behavior.

To keep the layers within the Universe topology simple, consider placing routers near the top and grouping devices logically by IP domains.

Additionally, you can manually model one or more network connections between modeled devices appearing in a Universe or Global Collections view. Finally, manually model all container icons within a World or TopOrg topology view.

How to Model Manually in the Universe Topology

This section describes the steps to manually model in the OneClick Universe topology.

Create Model Dialog

The Create Model dialog includes the Create Model by Type, Create Model by IP, and Create Model by Host Name dialogs. These dialogs contain settings that depend on the model:

- **Name**
  Specifies the unique host name for the device that you are modeling.

  NOTE
  Model by Host Name supports host names that resolve to either an IPv4 or an IPv6 address.
- **Network Address**
  Specifies the IPv4 or IPv6 address for this device to let DX NetOps Spectrum communicate with it.
- **SNMP Community String**
  Specifies the SNMP community string for this device to let DX NetOps Spectrum communicate with it.
  Note: You can create a model using Create Model by IP or Create Model by Host Name. If you do not specify a value for SNMP Community String or Agent Port, DX NetOps Spectrum uses the predefined SNMP credentials. You configure these SNMP credentials in the VNM model's Information tab in OneClick. Navigate to the Modeling and Protocol Options section of the AutoDiscovery Control subview. If the device cannot be contacted using each of the SNMP credentials but it can be contacted using ICMP, a Pingable model is created.
- **Serial Number**
  Specifies the serial number for the device that you are modeling.
- **Security String**
  Specifies the security for the device. Adding a security string prevents selected users from viewing this model.
- **Subnet Mask**
  Specifies the device subnet address that this container represents. The subnet address label then appears when a user points to a container icon in a topology view.
- **Poll Interval (sec)**
  Specifies the frequency with which this device is polled. By default, DX NetOps Spectrum polls modeled devices for status updates every 60 seconds (or for some model types every 300 seconds).
  Longer polling intervals use less bandwidth for management traffic, but you receive fewer device status updates. Consider whether to use the default polling interval (60 seconds) for critical devices and to use 600 seconds for less important devices.
- **Log Ratio**
  Defines how often DX NetOps Spectrum polls devices for updates before logging the results.
  Default: 10
- **Creation Author**
  Specifies the name of the user who is modeling this managed device.
- **Manufacturer**
  Specifies the manufacturer name of the managed device that you are modeling.
- **Southbound Gateway-specific**
  For more information about the Southbound Gateway settings, see the Southbound Gateway Toolkit section.
- **Unique ID**
  The Unique Identifier is a composite of up to six variable data items (1-6). The final unique identified string is composed as follows:
  <1>_<2>_<3>_<4>_<5>_<6>
  If one of the unique identifier components is not provided, it is not included within the composite unique identifier.
- **Manager Name**
  If the name of the third-party application does not apply in the list, choose Default. When this attribute is set on the EventAdmin, all EventModels contained with this EventAdmin inherit this attribute.
- **Event Model Prefix**

This field is prepended to the EventModel name for the Event Models that this EventAdmin contains. This behavior provides consistent naming prefixes for all EventModels that are associated with a particular EventAdmin. This prefix is useful when sorting or filtering various DX NetOps Spectrum applications.

- **Dialup Link Type**
  Specifies the functional type of the Dialup_Link. Possible types are Dial Backup Link, Primary on Demand Link, and Bandwidth on Demand Link.

  NOTE
  For more information about the Dialup_Link settings, see the Non-Persistent Connections Manager User *section*.
- **Dialup Protocol Type**
  Specifies the protocol type to use on the Dialup_Link. Possible protocol types include Analog, Switch-56, ISDN, and Frame_Relay.
- **Activation Grace Period (Min)**
  Specifies the time (in minutes) for the secondary link to become active after a primary link failure. If this grace period expires before the secondary link is active, a red alarm is generated. Only the Dialup_Link models use this field.
  Default: 3 minutes
- **Deactivation Grace Period (Min)**
  Specifies the time (in minutes) for an active secondary link to deactivate after the failed primary link reactivates. If the secondary link is still active after this grace period expires, then a yellow alarm is generated. Only the Dialup_Link models use this field.
  Default: 3 minutes
- **Active Time Until Yellow (Hours)**
  Specifies the number of hours that a backup link can be active before a yellow alarm is generated.
- **Active Time Until Orange (Hours)**
  Specifies the number of hours that a backup link can be active before an orange alarm is generated.
- **Active Time Until Red (Hours)**
  Specifies the number of hours that a backup link can be active before a red alarm is generated.
- **Device Symbol**
  Specifies the type of icon to use for this device in the Topology view.
- **DCM Timeout (ms)**
  Specifies how long the SpectroSERVER waits for a device response.
  Default: 3000 milliseconds
- **DCM Retry Count**
  Specifies the number of times that the SpectroSERVER tries to establish device communication after the DCM timeout value expires.
  Default: 3
- **Agent Port**
  Specifies the SNMP agent port.
  Default: 161

  NOTE
  You can create a model using Create Model by IP or Create Model by Host Name. If you do not specify a value for SNMP Community String or Agent Port, DX NetOps Spectrum uses the predefined SNMP credentials. Configure these SNMP credentials in the OneClick Information tab for the VNM model. Navigate to the Modeling and Protocol Options section of the AutoDiscovery Control subview. If the device cannot be contacted using each of the SNMP credentials but can be contacted using ICMP, a Pingable model is created.
- **Secure Domain**
  Specifies a secure domain for this device. Select the applicable domain from the drop-down list.
- **SNMP Communications Options**

Specifies that SNMP protocol that this device supports: SNMP v1, SNMP v2c, or SNMP v3. DX NetOps Spectrum uses the protocol that you specify here to discover and map this device.

  - **Profiles**
    Opens the Edit SNMP v3 Profiles dialog, where you can create profiles for SNMP communication.

- **Discover Connections**
  Specifies DX NetOps Spectrum Discovery behavior. When enabled, DX NetOps Spectrum discovers the linked connections (pipes) between this device and other devices.

Add Containers to Universe Topology Views

You can create a container or can use an existing container to represent the group of devices you want to model. You can create containers at any topology level to help reduce the complexity of your topology views. Containers can effectively help you monitor and manage the health of the devices they represent.

You can manually add a container to a Universe topology view by using the Select Model Type dialog. Some examples of containers you can add include LAN, FDDI, Network, or WAN.

Follow these steps:

1. Select the Universe topology view where you want to add a container.
   The selected topology view appears in the Topology tab, Contents panel.
2. Click
   
   (Creates a new model by type) in the Topology tab toolbar.
   The Select Model Type dialog opens.
3. Click the Containers tab, select the type of container you want to add, and click OK.

   NOTE
   You can use the Filter text box to filter the containers list in the Containers tab. For example, enter **LAN** in the Filter text box to filter the container list to show the LAN container types only.

   The Create Model of Type dialog opens.
4. Complete the fields in the dialog.
5. Click OK.
   The Create Model of Type dialog closes, and OneClick places the newly created network container in the selected Universe topology view.

Add Existing Devices to a Container

To add modeled network devices to a container, double-click the container icon and copy and paste modeled devices into the selected container. These modeled devices can come from other Universe topology views, list views, or the Explorer tab. Or, model new devices in this container by using these topology toolbar functions:

- Create a model by model type.
- Create a model by IP address.
- Create a model by host name.

Add Network Devices to Universe Topology Views

You can manually add network devices to a Universe topology view using the Topology toolbar ‘Create model by’ functions. You can add one or more devices to a container using these buttons.

The best practice for manually adding a device to the OneClick environment is to use one of these functions:

- Create a model by IP address.
- Create a model by host name.

Add a Device Using Create Model by Model Type

Creating Model by Model Type

The ‘Create model by model type’ function is considered an advanced function. This function requires that you have an understanding of how network devices are categorized in the SpectroSERVER modeling catalog.

Follow these steps:

1. Select the Universe topology view where you want the new device to appear.
   The selected Universe topology view appears in the Topology tab, Contents panel.
2. If you want to place the new device inside a container, double-click the container icon to display the topology view for that container.
3. Click
   
   (Creates a new model by type) in the Topology tab toolbar.
   The Select Model Type dialog opens.
4. Click the All Model Types tab.
   A list of model types appears.
5. (Optional) Enter text in the Filter field to filter the list.
   As you type characters in the Filter field, only the model type names that contain the same string of characters are shown in the list.
6. Select the model type of the device you are adding and click OK.
7. The Create Model Of Type dialog opens.
8. Complete the fields in the dialog.
9. Click OK.
   The Create Model of Type dialog closes and OneClick places the newly created device icon in the selected Universe topology view.

Add a Device Using Create Model by IP Address or Create Model by Host Name

You can add a device using the ‘Create Model by IP’ or ‘Create Model by Host Name’ function.

Follow these steps:

1. Select the Universe topology view where you want the new device to appear.
   The selected Universe topology view appears in the Topology tab, Contents panel.
2. To place the new device inside a network group container, double-click the container icon to display its topology view.
3. Click the Create Model by IP down arrow
   ()
   and select one of these options in the Topology tab toolbar:
   – Create by IP
   – Create by Host Name
   The Create Model dialog opens.
4. Complete the fields in the dialog.
5. Click OK.

The Create Model dialog closes, the Creating Model dialog indicates that the request is processing. When it closes, the newly created device icon is placed into the selected Universe topology view.

Manual Modeling Tips

To move or enhance the appearance of the recently modeled device icon, click **Edit mode** in the Topology tab toolbar.

To cut/copy/paste the modeled device icon to another Universe topology view, list view, or the Explorer tab, use one of the following cut/copy/paste functions:

- The Topology tab toolbar
- The List tab toolbar
- The right-click menu in the Explorer tab

To change configuration parameters of a modeled device, select the modeled device and change the appropriate settings in the Component Detail panel, Information tab. For example, configuration parameters can include the SNMP community string, polling interval, logging interval, and security string.

Model By Type Preference

Specify which model types are available when users manually create models by model type in the Topology tab using the Model By Type preference. When users click **Creates a new model by type** () in the Topology tab toolbar, the Select Model Type dialog opens. The Select Model Type dialog contains the **All Model Types** tab. This tab lists all the model types available for them when they create a model. You can prevent model types from appearing in this list by modifying the Model By Type preference.

Configure Availability of Model Types for Manual Modeling

As an administrator, you can configure the Model By Type preference to exclude or include certain model types. This feature controls which model types that users can see when they manually add models in the Topology tab.

> **NOTE**
> For general information about setting preferences and using the Set Preferences dialog, see the *Operator section*. For advanced information about setting user preferences, locking preferences, and administrating preferences, see the Administrator *section.*

Follow these steps:

1. Click View, Preferences.
   The Set Preferences dialog opens.
2. Expand the Topology Tab folder in the Name column and click Model By Type.
   The Set Preferences dialog displays the Include Model Types list on the left and the Exclude Model Types list on the right.

   > **NOTE**
   > By default, the Include Model Types list includes all the available model types when you first access the Model By Type preference. You have not yet excluded any model types.
3. (Optional) Enter text in the appropriate Filter field to filter the desired list.
   As you type characters in the Filter field, only the model type names that contain the same string of characters are shown in the list.
4. Do *one*:
   – From the Include Model Types list, select the model type that you want to *exclude*. Click the right arrow button to move it to the Exclude Model Types list.

NOTE
To select several model types at once, press the CTRL key and click each model type.

After you click Apply, the selected model type is moved to the Exclude Model Types list. This model type no longer appears in the Select Model Type dialog.

– From the Exclude Model Types list, select the model type that you want to *include*. Click the left arrow button to move it to the Include Model Types list.
After you click Apply, the selected model type is moved to the Include Model Types list.This model type now appears in the Select Model Type dialog.

5. Click Apply.
The model type changes you made are applied.
6. Click OK.
The Set Preferences dialog closes.

Polling Interval Changes

You can change the polling interval for any device. To change the interval, enter a new value in the Poll Interval (sec) field in the DX NetOps Spectrum Modeling Information subview for the device model. You can also use the Attribute Editor.

NOTE
Polling intervals also apply to application models, many of which have an initial setting of zero, which in effect disables polling. However, the preferred method for disabling polling for any model is to set the Polling Status attribute to Off.

Disable Polling for a Model or a Model Type

To conserve bandwidth, you can increase the default polling intervals for selected models. Additionally, you may decide that the status of certain devices is not worth any polling bandwidth, even at longer intervals. For example, some network administrators choose to disable modeling endpoint devices such as workstations. The reason is that they do not need to receive the alarms that occur each time that these devices are powered down. Therefore, to model the endpoints but conserve bandwidth from network polling traffic, you can disable polling to these models (or to any models). You disable polling by changing the value of the Polling Status attribute to FALSE.

Polling Status Changes

You can change the polling status for any model by turning on or off polling. To change polling status, use the Polling setting in the DX NetOps Spectrum Modeling Information subview for the selected model. You can also use the Attribute Editor.

NOTE
The Polling Status value takes precedence over the Polling Interval value in terms of enabling/disabling various periodic external requests to a model. Setting the Polling Interval to zero automatically changes the Polling Status to Off. If you reset the Polling Status to On, the DX NetOps Spectrum inference handlers could still generate requests, even though the Polling Interval is set to zero. However, to enable normal DX NetOps Spectrum polling for fault isolation purposes, the Polling Interval would have to be manually reset to a non-zero value.

Poll Devices That Are Down

When contact with a device has been lost, DX NetOps Spectrum uses two methods to continue polling the device for a status change. First, DX NetOps Spectrum pings the device once every 60 seconds. Second, DX NetOps Spectrum sends SNMP requests every third polling interval by default. For example, if the device polling interval is set to 60 seconds, DX NetOps Spectrum polls the down device once every 180 seconds.

To change the default interval at which DX NetOps Spectrum polls a down device, insert the following syntax into the **<$SPECROOT>/SS/.vnmrc** file:

```
down_device_poll_interval_multiplier=<user_defined_multiplier>
```

For example, if the **<user_defined_multiplier>=2** and the device polling interval is 60 seconds, then DX NetOps Spectrum polls the down device once every 120 seconds (2*60=120).

Connections (Pipes) Between Modeled Devices

You can depict the physical connections (pipes) between the modeled devices using the **Start** and the **End Connection** options in the Topology right-click menu. In OneClick, you can manually create three types of connections between modeled devices:

- **Resolved connection:** (For a fully resolved connection.) A resolved connection occurs when two devices are connected at the port level. For example, Port-A of device one is connected to Port-B of device two.
- **Partially resolved connection:** A partially resolved connection occurs when only one port is known between two devices. You typically create this type of connection when you know only the port of one modeled device. For example, device-one is connected to Port-A of device-two.
  When you manually model a partially resolved connection, DX NetOps Spectrum attempts to resolve the port connection of the other device. If DX NetOps Spectrum succeeds, it represents the connection as a fully resolved connection. You can later determine whether the connection is fully resolved. To check for a resolved connection, click the link within that view and view the Link Information tab in the **Component Detail** panel.
- **Unresolved connection:** An unresolved connection occurs when two modeled devices (or containers) are not connected in any way at the port level. For example, container-A is connected to container-B.

Make sure that the Live Pipes attribute for each VNM is set to **Enable** when these conditions exist:

- You have a DSS (distributed SpectroSERVER) environment.
- Any of the connections or pipes that you are creating span between two or more devices that different SpectroSERVERs manage.

Dynamic Link Status Partial or Fully Resolved Connections

After you create a partial or fully resolved connection between two modeled elements, you can monitor the status of that connection. You monitor the status by enabling the connection as a Live Pipe. The color of live pipes in the Universe topology views indicates status information about the connection. For example, good connection conditions are green, bad connection conditions are red, and disabled live-pipe connections are gold.

A live pipe shows a combined status condition for fully resolved connections (two ports). The connection having the most severe condition determines the color of the pipe. A live pipe can generate an alarm when one or both the links it represents goes down.

> **NOTE**
> Initially after modeling a connection, the color of the connection is gold or silver. Gold appears for resolved or partially resolved connections; silver appears for all unresolved connections.

DX NetOps Spectrum monitors the Border Gateway Protocol (BGP) peer session under these circumstances:

- Live pipes are enabled on a connection between two routers, or on a connection between a router and a provider cloud.
- The ports that are involved with the connection are participating in a BGP peer session.

Remove Connections from a Universe Topology View

You can delete a connection between two modeled elements in the Universe topology view by right-clicking the pipe and selecting Delete. When a pipe is deleted, DX NetOps Spectrum removes all of its associations. If the pipe represents more than a single port connection, DX NetOps Spectrum prompts you to confirm the deletion.

Automatic Recreation of Pipes

DX NetOps Spectrum automatically recreates pipes when you copy and paste a set of previously connected modeled icons to another Universe topology view or list view. Also, pipes are automatically recreated when you copy and paste the models in the Explorer tab. If you delete one of the connected modeled icons for a view, the pipe is erased. Later, you could copy that device from the Lost and Found view to the original topology view or list view. In this case, OneClick automatically recreates the connection between the two modeled devices.

Create an Unresolved Connection Between Modeled Elements

When you do not know the port connections between two modeled elements that you want to connect, you can create an unresolved connection. When you create an unresolved connection between two modeled elements in the Universe topology view, the pipe representing the connection is silver. You are prevented from enabling that connection as a Live Pipe. However, after you create an unresolved connection between two modeled elements in a Universe topology view, DX NetOps Spectrum automatically attempts to resolve the connection between them. If DX NetOps Spectrum succeeds in resolving the connection, the pipe representing the connection is gold and it behaves as a resolved connection. You can then proactively monitor the status of that resolved connection by right-clicking the pipe and selecting Live Pipe.

If DX NetOps Spectrum is unable to detect at least one port-level connection between two devices, the pipe in the topology view remains as an unresolved connection (silver). You cannot enable Live Pipe on an unresolved connection.

Follow these steps:

1. In a Universe topology view, right-click any modeled element (device or container) and select Start Connection to designate the starting point of a connection.
2. In a Universe topology view, right-click the modeled element (device or element) and select Connect with <*starting point address*> to designate the endpoint of the connection.
   DX NetOps Spectrum models an unresolved silver-colored pipe between the two modeled devices that are specified. If the connection between the modeled elements spans two separate views, an off-page reference icon appears in the view.

Create a Resolved Connection

When you know the ports of both modeled devices, you can create a port-to-port resolved connection.

Follow these steps:

1. Designate the starting point of a connection in a Universe topology as follows:
   a. Select a modeled device (such as a switch or router) that contains port interfaces.
   b. Click the Interfaces tab in the Component Detail panel.
   c. Right-click a port row in the Interfaces tab and select Start Connection.
2. Designate the endpoint of the connection in a Universe topology as follows:
   a. Select a modeled device (such as a switch or router) that contains port interfaces.
   b. Click the Interfaces tab in the Component Detail panel.
   c. Right-click a port description in the Interfaces tab and select Connect with <*starting point port address*>.
   DX NetOps Spectrum creates a resolved (gold-colored) pipe between the two modeled icons. If the modeled devices are in separate views, an off-page reference icon appears in the view.
3. To monitor the link status of this connection, right-click the connection and select Enable/Disable Live Links.

Create a Partially Resolved Connection

Sometimes, you know only the device port of one of the two modeled devices that you want to connect. In this case, you can create a partially resolved connection.

Follow these steps:

1. Designate the starting point of a connection by following these steps:
   a. In a **Universe** topology view, select a modeled device (such as a switch or router) that contains port interfaces.
   b. In the **Component Detail** panel, click the **Interfaces** tab.
   c. In the **Interfaces** tab, right-click a port row and select **Start Connection**.
2. Designate the endpoint of the connection. In a **Universe** topology view, right-click any modeled element (device or container) with an unknown port address and select **Connect with** *<starting point modeled port address>*.
   DX NetOps Spectrum models a partially resolved (gold) pipe between these two modeled devices. If the connection between the modeled devices spans two separate views, an off-page reference icon appears in the view.
3. (Optional) Monitor the link status of this connection.

> **NOTE**
> OneClick automatically attempts to locate the unknown device port. You can verify whether DX NetOps Spectrum locates this device port by clicking the link and viewing the Information tab in the **Component Detail** panel.

Resolve Unresolved Connections

You can resolve unresolved and partially resolved connections from the Interfaces tab.

Follow these steps:

1. Select the device model for which you want to resolve a connection.
2. Click the Interfaces tab in the Component Detail panel.
   A warning appears in the Interfaces tab toolbar specifying how many unresolved connections this device has.
3. Right-click an unused interface and select Resolve Connection To, *<model at other end of connection>*.
   The warning in the Interfaces toolbar changes to show the revised number of unresolved connections. For example, if you had only one unresolved connection, the warning now disappears. If you had two unresolved connections, the warning now indicates that you have only one unresolved connection.

Lock and Unlock Resolved Connections

In OneClick, you can preserve a resolved connection between two modeled devices by locking that connection. When you lock a connection, Discovery does not delete the connection.

Follow these steps:

1. In a Universe topology view, right-click the resolved connection that you want to lock/unlock and select Lock/Unlock Connection.
   The Lock/Unlock Connection dialog opens.
2. Select the connections that you want to lock/unlock and click OK.
   The resolved connection is locked/unlocked according to your selection.

Enable/Disable Modeling of a Connection Between Two Ports

10.4.2 allows you to enable or disable modeling of a connection between two ports. It provides a new attribute (LockPort) on the port model type. This attribute helps you decide whether you want to create connections between two ports. By default, the value of the attribute is set to *No*. This implies that there is no change in the existing behavior while performing the discovery or establishing a connection between two ports. However, if you set the value to *Yes* and again perform the discovery, then no connection is established on the port for which LockPort is set to Yes. The LockPort value acts only when there is no connection between ports. If a connection already exists and you change the port's LockPort value to Yes and again perform the discovery, then no change happens to the existing connection.

Understand the Working

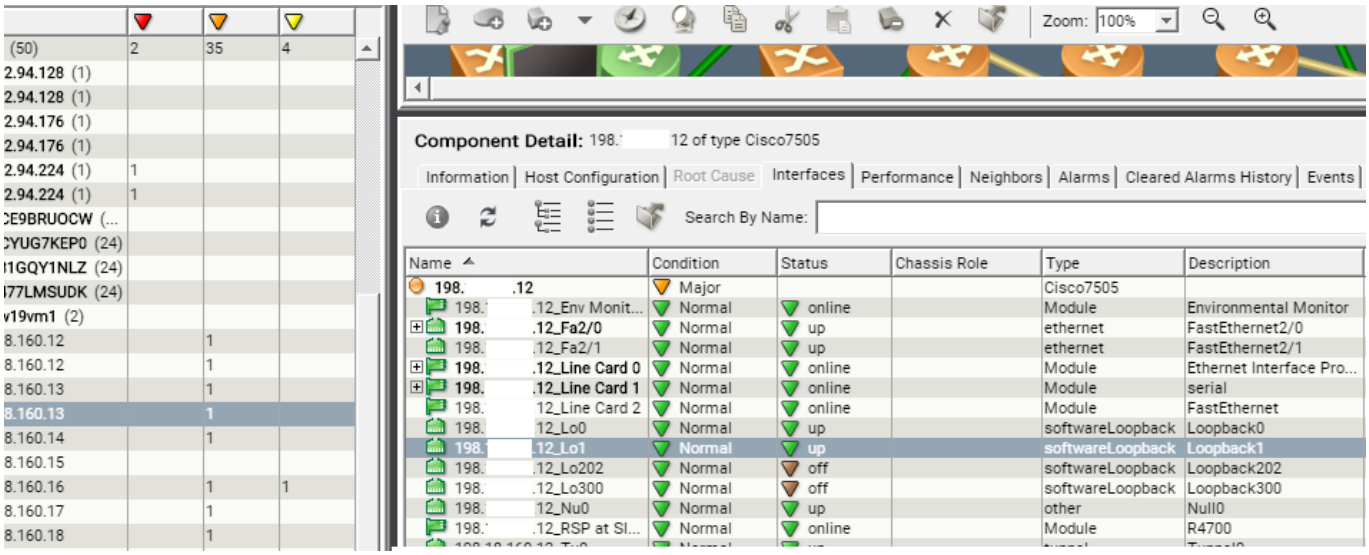
During the initial or new discovery, all the devices and the connections between them are created in the usual manner. At a later stage, if you do not want to create a connection between any ports, select the required port, locate the LockPort attribute, and change the value to Yes. Now, you can remove the unwanted connections for that port. From this point onward, for all kinds of discoveries, no connection is made on this port. If that is device is removed, the LockPort attribute value is reset to No.

Configure the Value

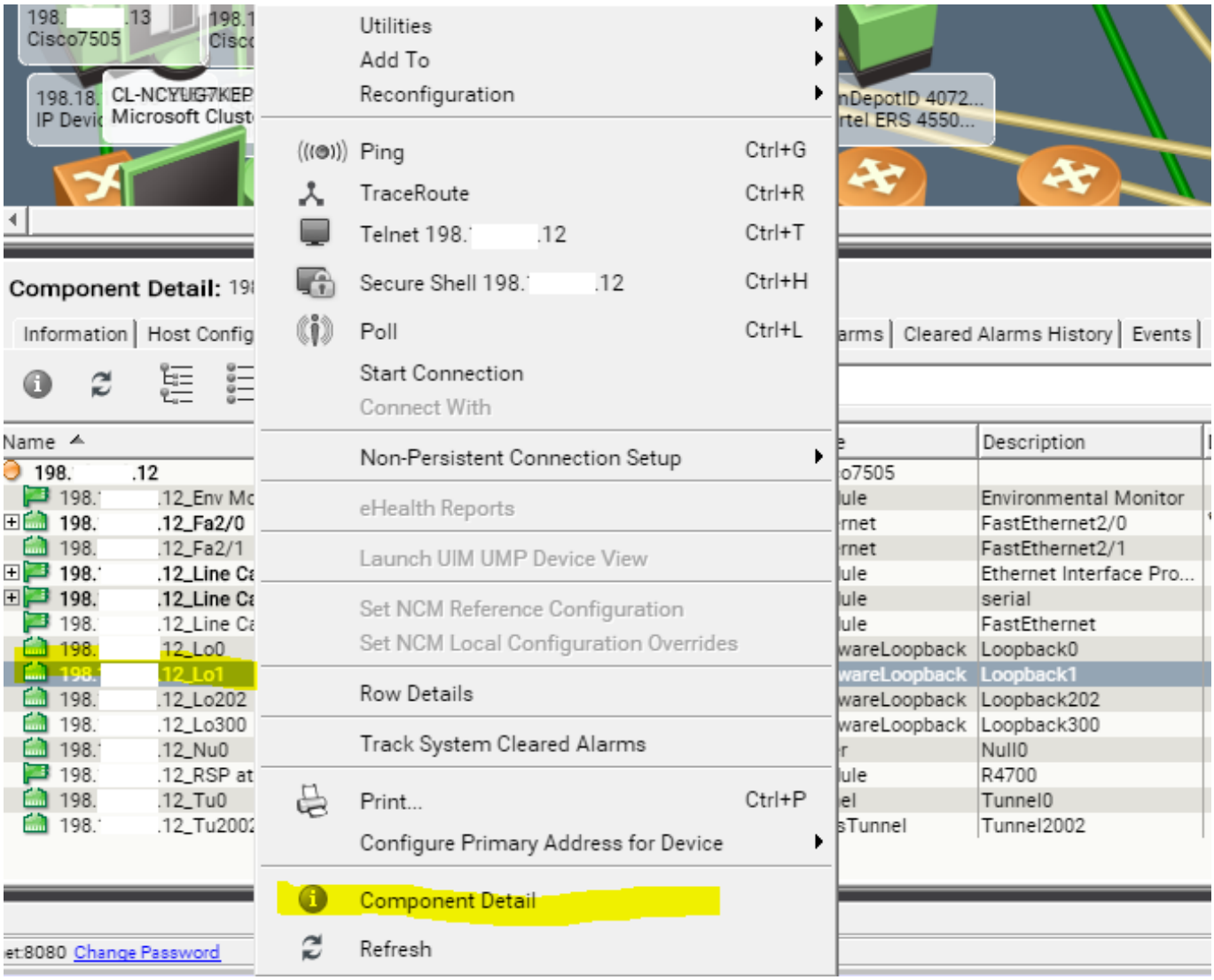
You can configure the value of the LockPort attribute.

Follow these steps:

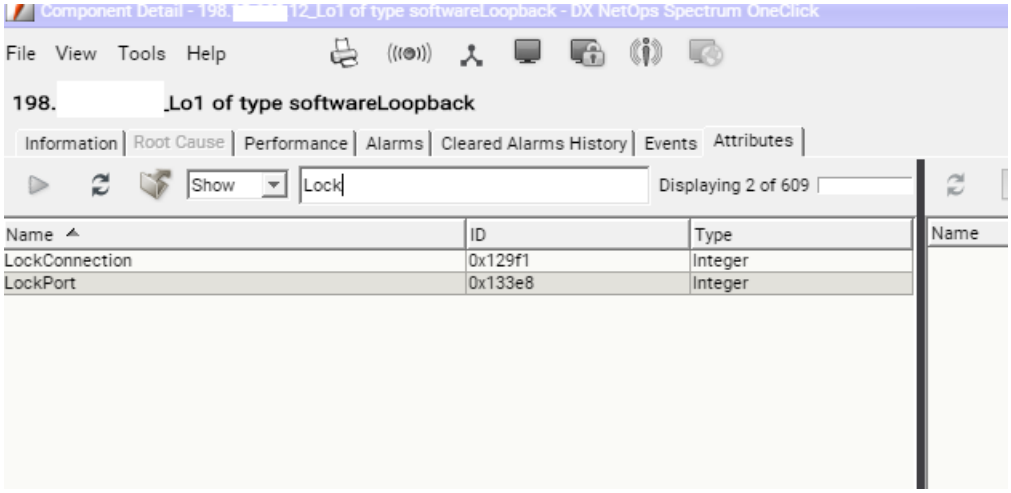
1. Access the OneClick UI.
2. Select any device that has interfaces and then select the **Interfaces** tab under the **Component Detail** pane.



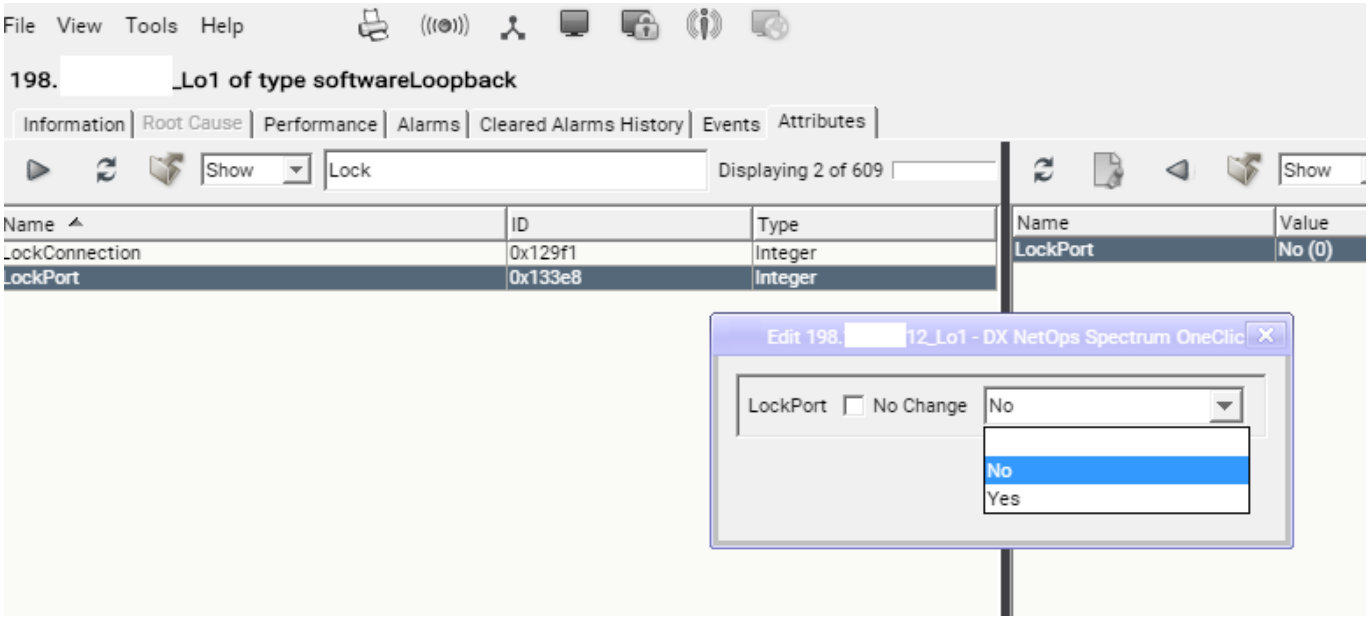
3. Select any interface (for example, Loopback, Ethernet, or Tunnel Interface).
4. Right-click and select the **Component Detail** option.



5. Select the **Attributes** tab and search for the LockPort parameter.



6. Change the value of the parameter as required.



You have successfully changed the value of the attribute.

> **NOTE**
> This functionality is not applicable for DMVPN discovery and Manual connection.

Enable or Disable a Live Link

You can monitor the link status of any resolved connection that is depicted in the Universe topology views by enabling Live Links. For live links on a partial or fully resolved connection, you can monitor the port connections at either end of the modeled devices.

The status of a live link displays through colors (red for critical, green for good, and so on). In addition, if a port connection within a live link goes down, you can view alarm information about that connection in the Alarms tab of the Contents panel.

You can enable and disable a live link for a partial or fully resolved connection.

> **NOTE**
> You cannot enable live links on a silver (unresolved) connection.

Follow these steps:

1. Right-click the connection that you want to enable/disable as a live link and select Enable/Disable Live Link.
2. Select the connections that you want to change and click OK.
   If you enable a connection, the color changes to green (good condition) or red (bad condition). The color of a disabled connection is gold.

Enable All Live Links

To enable all live links at once you need to:

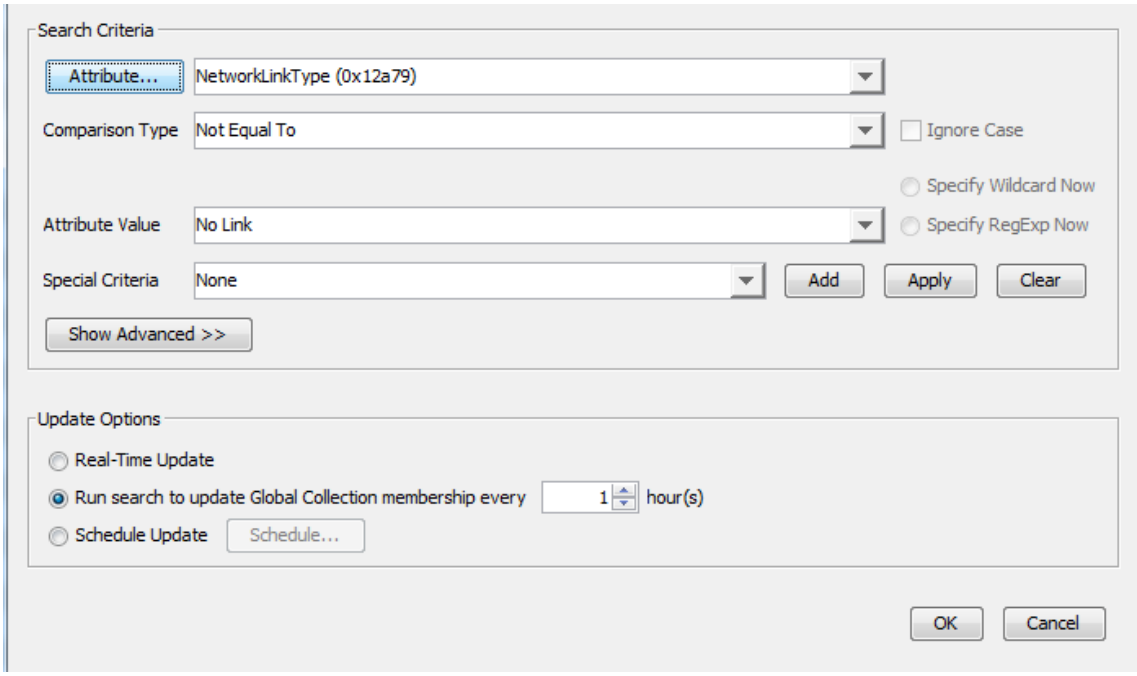
- Create a **Global Collection** for all live links in the Navigation pane.
- Apply the **ok_to_poll** attribute on that Global Collection using Attribute Editor.

Follow these steps to create the Global Collection:

1. Right-click **Global Collections** and select **Create Global Collection**.

The **Create Global Collection** dialog appears.

2. Enter the **Name**, **Owner**, **Description**, **Security String** for the new Global Collection.
3. Click **Landscapes** to select the required landscapes, and click OK.
4. Click **Search options** to create a Global Collection for all connected port models (live links).
   The **Search Options** dialog appears.
5. Configure the Search Options dialog as shown in the following image:



6. Click OK.

The Global Collection for all connected port models is created.

Follow these steps to enable all the live links using ok_to_poll attribute:

1. Select the new Global Collection of all connected port models (live links) in the **Navigation** pane.
2. Click List, and select all the port models in the **Contents** pane.
3. Right-click, and Select **Utilities**, **Attribute Editor.**
   The **Attribute Editor** dialog appears.
4. Click **add** in the left pane to add a user defined attribute.
   The Attribute Selector dialog appears.
5. Select **Port** in the left pane, and then select the **ok_to_poll** attribute in the right pane and Click OK.
   You are back to **Attribute Editor** dialog.
6. Set this attribute to Yes, and then Click Apply.
   Now this Attribute value is loaded on all the live links of the global Collection. After loading, click Close.
7. Click OK to close the **Attribute Editor**.

All the live links are enabled in the **Universe**, you can see that all enabled healthy live links in green. The color changes based on the condition of the links.

To disable all the live links at once, set the **ok_to_poll** attribute to "No" on the Global Collection, or delete the Global Collection.

BGP Peer Session Monitoring

Border Gateway Protocol (BGP) peer session monitoring polls the status of the peer session between two BGP devices.

From 10.4.2, DX NetOps Spectrum uses the Cisco BGP MIB to monitor the ipv4 and ipv6 BGP peer sessions for cisco devices. If the device does not support Cisco BGP MIB, then DX NetOps Spectrum uses the BGP standard MIB.

NOTE
When you upgrade DX NetOps Spectrum to 10.4.2, reconfigure the devices to monitor the peer session using the Cisco BGP MIB.

DX NetOps Spectrum monitors the BGP port peer session status at the polling interval of the port model's Polling_Interval attribute value under these conditions:

- The BGP peer session monitoring is enabled.
- The live pipe on a BGP peer session port is turned on.

When a monitored BGP peer session is no longer found in the bgpPeerTable MIB table, the result is:

- A "BGP peer session removed" event is generated.
- The session is no longer monitored.

Consider the following information about alarms and BGP peer sessions:

- Set the Polling_Interval attribute from 0 to 120 (or some other positive value). For more information, see Not Receiving BGP Alerts After Enabling BGP Peer Session Monitoring in the VNM Model.
- When WA_Link is connected to the ports in a BGP peer session, a BGP peer session down alarm is generated on one of the port models. The session is not generated on WA_Link.
- If the loss of a BGP peer session is the root cause of a downstream outage, a critical BGP alarm is generated. This alarm hides the lost contact alarm of the downstream device.
- If a monitored BGP peer session goes down, a single alarm is generated on the BGP peer session port model. If the monitored BGP peer session is between two directly connected routers, the alarm is asserted on the port model on which the outage is first detected.
- If a BGP peer session port is administratively disabled or operationally down, the BGP peer session alarm on the port model becomes symptomatic alarm of the link condition alarm.
- When a backward transition trap is received, the BGP MIB is polled to verify that the peer session is established. If the peer session is not established, an alarm is generated on the peer session port model.

NOTE
For more information about using MIB Tools, see the Certifications section.

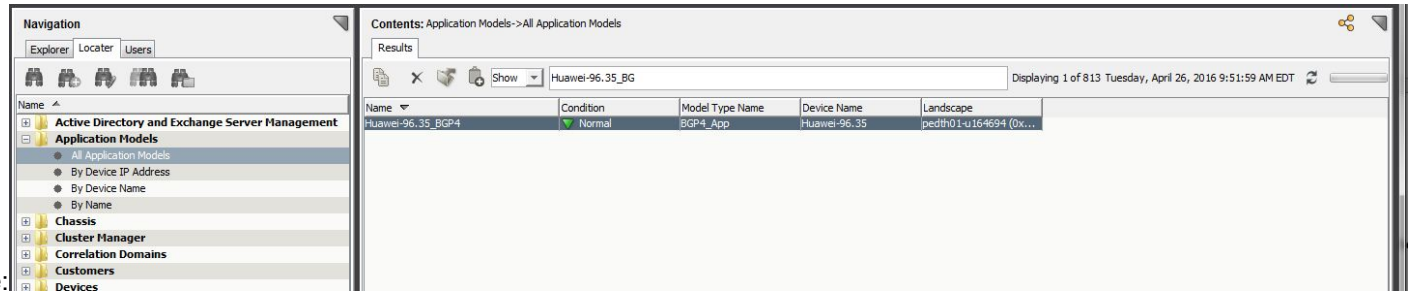
Troubleshooting BGP Peer Monitoring

This section discusses the common errors that occur while configuring BGP Peer Monitoring and their solution.

Not Receiving BGP Alerts After Enabling BGP Peer Session Monitoring in the VNM Model

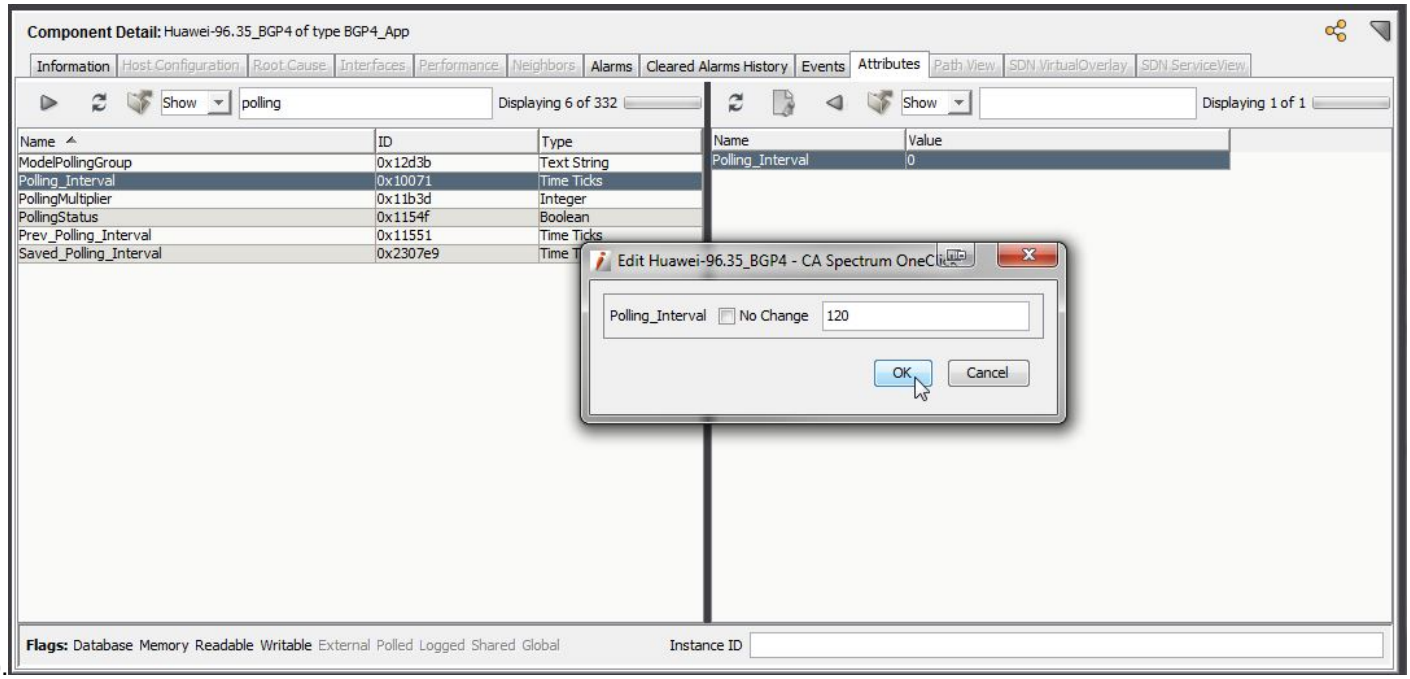
You may not receive BGP alerts if the polling interval is not set on the BGP Application Model.

1. Run a **Locater** search for **Application Models,** filtering on the BGP4_App for the device:



2. Under **Component Detail** for the BG4_App, click on the **Attributes** tab.

3. Search for the **Polling_Interval**.
4. Set the **Polling_Interval** from 0 to 120 (or some other positive value).



5. Save the changes.
   After this change, you should start seeing expected BGP alerts.

Remove Modeled Elements from the Universe Topology View

The Universe Topology and Explorer view allows you to remove modeled elements.

> **WARNING**
>
> Due to high possibility of models getting corrupted in the process, it is recommended that you do not use this option, unless the model exists in a static Global Collection or in the World view.

To remove modeled elements from the Universe Topology/ Explorer View,

Follow these steps:

1. In the **Universe** topology/explorer view, right-click the modeled element, and select **Remove.**
   The **Confirm Removal** dialog opens.
2. Click **Yes.**
   The element is removed from the topology/explorer view.
   Models are placed in the Lost and Found (**LostFound**) subview.

Delete Modeled Elements from the Universe Topology View

You can delete modeled elements from the Universe topology view.

Follow these steps:

1. In the Universe topology view, right-click the modeled element, and select Delete.
   The Confirm Delete dialog opens.
2. Click Yes.

The model is deleted permanently from the system. When deleting containers, models within the containers are placed in the Lost and Found. If those models exist in other topologies like World View and Global Collection, they continue to exist there. In this case, they are not moved to Lost and Found.

Cut Modeled Elements from the Topology View or List View

You can cut a modeled element from a topology view, list view, or using the right-click menu in the Explorer tab. In this case, OneClick removes the model from the view and places it into the Lost and Found. If desired, you can remove the modeled element from the Lost and Found view as well.

To cut a modeled element from a view, right-click the modeled element and select Cut.

The cut modeled element moves to the Lost and Found view.

Follow these steps:

1. Select **LostFound** in the Explorer tab in the OneClick Navigation panel.
2. Click the List tab in the Contents panel.
3. Select the elements that you want to remove in the List tab.
4. Right-click and select Delete.

Enhance Topology Views

You can put the current Topology view in Edit mode. Then, you can enhance the Topology view using the tools that are provided in the Edit mode toolbar.

Export a Topology View

You can export any topology view to a PNG file format.

Follow these steps:

1. Navigate to the topology view that you want to export.
2. Click the Export button in the Topology tab toolbar.
   The Save As dialog opens.
3. Specify a name and location for the file and click OK.
   The topology view is saved as a PNG file.

> **NOTE**
> For more information about exporting views and table data, see the Using OneClick section.

Modeling Manually in a Global Collections Topology

If you have the Manage Global Collections privilege, you can create a collection in the Global Collections topology. Create a collection from any modeled elements that were previously modeled in one or more Universe topology views.

When you create a collection within the Global Collections topology, provide a name and an owner for that collection and define its members. The owner field is used to indicate who is responsible for the Global Collection. This field is initially set to the OneClick user who created the collection. Select members for a collection either by specifying search criteria or by using the copy and paste feature.

Dynamic Membership

When you use search criteria to define the members of a Global Collection, the members of that Global Collection are considered *dynamic*. They remain in the Global Collection only as long as they meet the specified search criteria.

DX NetOps Spectrum automatically removes a modeled element from the Global Collection that no longer meets the original search criteria, using the method that you choose:

- The next time the Global Collection is manually updated
- Periodically, at the next scheduled interval
- Dynamically, as soon as the model no longer meets the Global Collection search criteria
- Scheduled, according to the assigned schedule

The default period for automatic scheduled collection updates is 24 hours. At any time, you can redefine the dynamic members in a Global Collection by editing the specified search criteria.

> **NOTE**
> Schedules can be assigned only after the initial creation of a Global Collection. Schedules are intended for Global Collections with search criteria that have the capacity to result in degraded product performance.

Static Membership

When you use copy/paste or add functions to define members of a collection, the members of that collection are considered static. The static members always remain in a collection until you decide to remove them manually.

Connections Between Modeled Elements (Members)

The connections between modeled elements in a Global Collection topology view and in a Universe topology view are similar. In both views, connections exhibit the same behavior and functionality. In the Global Collections view, you can create partial and fully resolved connections, or unresolved connections (links, pipes). You can monitor the status of any resolved connection using Live Links.

Updating Modeled Elements in Global Collections

DX NetOps Spectrum updates all modeled elements in a Global Collection view, using the method that you choose:

- The next time the Global Collection is manually updated

  > **NOTE**
  > You must have the Update Global Collection Membership privilege to update a Global Collection manually.
- Periodically, at the next scheduled interval
  The default period for automatic scheduled Global Collection updates is 24 hours. At any time, you can redefine the dynamic members in a Global Collection by editing the specified search criteria.
- Dynamically, as soon as the model changes to meet or no longer meet the Global Collection search criteria
- Scheduled, according to the assigned schedule

Note: Scheduled updates can only be applied if you have the Schedule Global Collection Updates privilege. Scheduled updates are intended for Global Collections with search criteria that have the capacity to result in degraded product performance. Scheduled updates are not available during the initial creation of a Global Collection. Without the Schedule Global Collection Updates privilege, you cannot change the Global Collection Update Options when a schedule is applied to a Global Collection.

Generate Reports on Collections

You can generate reports on global collections using the Report Manager module. By using the Report Manager module with the OneClick Global Collections topology, you can at any time generate a single report about any one collection. For more information about running reports, see the Report Manager section.

NOTE
The Report Manager module is not included in the DX NetOps Spectrum core product line. This module must be purchased separately.

How to Define and Manage Global Collections

Defining and Managing Global Collections

When defining a collection of modeled elements in the Global Collection topology, consider following this process:

1. Create a Global Collection.

   NOTE
   A new global collection remains empty until you define its members.
2. Define Dynamic members or, when necessary, Define static members.
3. Edit the members in a global collection as needed. For dynamic members in a global collection, you can redefine the search criteria. For static members in a global collection, you can delete, or copy/paste members in a global collection, or can add members to a global collection.
4. Create a Global Collection Hierarchy if you want to organize global collections using folders and subfolders.
5. Delete global collections as needed. The modeled elements within a global collection represent copies of modeled elements from the Universe topology. Therefore, Delete removes only the specified global collection and the copies of the modeled elements that global collection represents.

Create an Empty Global Collection

You can create an empty global collection when you are unsure of the type of global collection you want to create. To create an empty global collection, provide only the name of the global collection. Save your empty global collection until you are ready to add static members, dynamic members, or both.

Follow these steps:

1. In the Explorer tab of the Navigation panel, right-click Global Collections and select Create Global Collection. The Create Global Collection dialog opens.
2. Complete the following fields and click OK:

- **Name**
  Specifies the name for the global collection.

  NOTE
  If your global collection name matches an existing global collection name, a warning appears. Click Yes to continue naming the global collection, or click No to enter another name for the global collection.

- **Owner**

Specifies the user who is responsible for the global collection.

- **Description**
  (Optional) Specifies a description for the global collection.
- **Security String**

(Optional) Specifies a security string expression to prevent certain users from viewing the contents of this global collection.

NOTE
For information about security string expressions, see the Administration section.

The global collection is created and appears in the Navigation panel under Global Collections.

Create a Global Collection of Dynamic Members

You can create a dynamic collection of dynamic members.

Follow these steps:

1. In the Explorer tab of the Navigation panel, right-click Global Collections and select Create Global Collection.
   The Create Global Collection dialog opens.
2. Enter relevant values for the following fields.
   - **Name**
     Specifies the name for the global collection.

     NOTE
     If your global collection name matches an existing global collection name, a warning appears. Click Yes to continue naming the global collection, or click No to enter another name for the global collection.
   - **Owner** Specifies the user who is responsible for the global collection.
   - **Description**
     (Optional) Specifies a description for the global collection.
   - **Security String**
     (Optional) Specifies a security string expression to prevent certain users from viewing the contents of this global collection.

     NOTE
     For information about security string expressions, see the Administrating section.

3. Click Search Options to provide the search settings for the global collection.
   Complete any of the following fields to create a single search expression:
   - **Attribute**
     Specifies the attribute of a device to filter. From the drop-down list of commonly used attributes, select the attribute that you want to use. The predefined list might not include the attribute that you want. In this case, click Attribute to specify the model type (device, port, or other) and its associated attribute that you want to find.

     NOTE
     If you choose an alphabetic attribute value, you can either clear (ignore) or select (include) the Ignore Case checkbox.
   - **Comparison Type** Specifies the type of comparison to be made against the value that is specified in the Attribute field. Only the comparison types appropriate to the attribute data type are available.
   - **Ignore Case** Determines whether the comparison is case-sensitive. If you do not select this checkbox, the comparison is case-sensitive. This selection is only enabled when it is appropriate for the data type of the attribute you selected.
   - **Attribute Value** Enter the desired attribute value to search.
   - **Devices Only** Specifies that the search results list includes only devices.
4. (Optional) To use a wildcard character or regular expression in the Attribute Value field, select a valid attribute in the Attribute field.
5. Select Matches Pattern in the Comparison Type field. Then, select one of the following options:
   - **Specify Wildcard Now**

Lets you search for a value using a wildcard. The following wildcards are available:

- *****
  Matches *any number* of characters.
  For example, 'switc*' returns 'switch' and 'switch-router.'
- **?**
  Matches any *single* character.
  For example, 'switc?' returns 'switch' but it does not return 'switch-router.'
  Both wildcards can be used anywhere and in any combination for a wildcard match.

  NOTE
  'Matches Pattern' is not a valid comparison type for all attributes.
- **Specify RegExp Now**
  Specifies that you want to create a search using Perl Compatible Regular Expression (PCRE) matching on attributes of the type 'text string'. Text string searches are available only for Matches Pattern comparison types. PCRE matching helps you to find and group models using specific pattern searches that are more advanced than existing searches or wildcard searches can provide.

  NOTE

  By default, all users have the privilege to enter regular expressions. Administrators can disable this privilege on a per-user basis.

6. (Optional) To conduct a compound search clause or a single search clause, do *one* of the following steps:
   - To conduct a search that is based on a single expression, click OK.
   - To conduct a search that is based on a compound clause, complete the following steps to build a compound search clause:

   a. a. Click Show Advanced. The compound expression box and logical operator buttons appear.
      b. Click Add to move the single expression you created in Step 4 to the compound expression box.
      c. Click one of the following logical operator buttons to build a compound expression: New AND; New OR; or AND/OR.

         NOTE
         The compound expression is represented in a tree structure that is grouped by logical operators (AND/OR). Each logical operator in the tree can include any number of attribute criteria nodes and logical operator nodes. For more information, click Hints in the Advanced section.
      d. (Optional) Click the Add Existing button to create a global collection from an existing attributebased search, actionbased search, or relationbased search.
      e. Repeat Steps 4 through 5 for each compound search expression you want to build.
7. Click OK.
   The Advanced Search mechanism locates and places a copy of all matching modeled elements (previously defined in the Universe topology) into the global collection.

   NOTE
   For more information about searches, see the Administration section.
8. (Optional) Select the Real-Time Update checkbox.
   This option disables the update interval.
   Also, it adds or removes models to and from the global collection when the models meet or no longer meet the search criteria.
9. (Optional) Specify a value in the **Run search to update Global Collection membership every <> hours** field.
   This field determines how often you want OneClick to conduct a search to update the dynamic members that are defined in the global collection.

   NOTE

   The option to associate a schedule with a global collection is not available during the initial creation of a collection.

10. Click OK.
    The Search Options dialog closes and the Create Global Collection dialog opens.
11. Click Landscapes to identify which landscapes you want Search to include when searching models to populate the global collection.
12. Click OK.

The global collection of dynamic members is created.

Global Collection Search Recommendations

The following information provides search criteria recommendations when defining global collections with dynamic members. The order of the criteria can affect the search performance.

The order of attribute criteria is based on two categories: *storage of information* and *data type.*

- **Storage of information**
  Order the attributes from least CPU (quickest access) to most CPU (slowest access), as follows:
  - **Memory flag** (least CPU/quickest access)
  - **Database flag**
  - **Calculated**
  - **External flag** (most CPU/slowest access)
- **Data type**
  Order the attributes from quickest comparison to slowest comparison, as follows:
  - Integer, counter, enumeration, model type handle (quickest comparison)
  - IP address, octet string
  - Text string (slowest comparison)

Combining the two categories of criteria, the overall attribute placement for complex searches of AND/OR order from top to bottom is as follows:

1. **Memory flag**
   a. Integer, counter, enumeration, model type handle
   b. IP address, octet string
   c. Text string
2. **Database flag**
   a. Integer, counter, enumeration, model type handle
   b. IP address, octet string
   c. Text string
3. **Calculated**
   a. Integer, counter, enumeration, model type handle
   b. IP address, octet string
   c. Text string
4. **External Flag**
   a. Integer, counter, enumeration, model type handle
   b. IP address, octet string
   c. Text string

Example

You would like to define a global collection containing dynamic members that are based on the following search criteria (in no particular order):

- **ifDesc**
- Topology model name string
- Network address
- Model type handle

Using the recommended ordering logic, we recommended the following order:

1. Model type handle (memory flag : model type handle)
2. Network address (memory flag/database flag : IP address)
3. Topology model name string (calculated flag : text string)
4. **ifDesc** (external flag : text string)

Create a Global Collection of Dynamic Members from an Existing Search

You can create a global collection of dynamic members from an existing attribute-based search, action-based search, or relation-based search. The existing searches can be found in the **Locater** tab.

> **WARNING**
> You can create a global collection that is based on an existing search. However, the association between the global collection and the existing search is not maintained. After the global collection has been created, any modifications to the search criteria for the global collection must be made in the global collection itself. Any changes to the existing search (in the **Locater** tab) on which the global collection was originally based are not propagated to the global collection.

Follow these steps:

1. In the **Locater** tab, locate the existing search that you want to base your global collection on.
2. Right-click the existing search and select Create Global Collection From.
   The Create Global Collection dialog opens.
3. Do the following, as necessary. The type of search you choose determines which options appear:
   - Enter a value for the search criteria. Options can include Matches Pattern, Equal To, Contains, and Starts With.
   - Select the Ignore Case check box to make the search case-insensitive.
   - Click Landscapes to identify which landscapes you want the Search to include when searching models to populate the global collection.
   - Click the List button and either enter a list of values to include in the search, or click Import to import a list of values. Click OK, and then click OK again.

   The Create Global Collection dialog reopens.
4. Complete the following fields:
   - **Name**
     Specifies the name for the global collection.

     > **NOTE**
     > If your global collection name matches an existing global collection name, a warning appears. Click Yes to continue naming the global collection, or click No to enter another name for the global collection.

Owner

Specifies the user who is responsible for the global collection.

- **Description**
  (Optional) Specifies a description for the global collection.
- **Security String**
  (Optional) Specifies a security string expression to prevent certain users from viewing the contents of this global collection.

NOTE
For information about security string expressions, see the Administrating *section.*

(Optional) Do the following steps to add another existing attribute-based, action-based, or relation-based search:

1. Click Search Options, click Show Advanced, and then click the Add Existing button.
   The Add Existing Search dialog opens.
   Note: Adding an existing search to your custom search copies the existing search and embeds it, as it is now, into your custom search. If you modify the existing search later, your custom search does not change. Your custom search contains only a *copy* of that existing search, as it was when you first copied it.
2. Select the existing search that contains the criteria you want to add to the current search and click OK.
   The Search dialog opens.
3. Do the following steps:
   – Enter a value in the provided field or select a value from the drop-down menu, if the drop-down menu is available.
   – Click Landscapes to identify which landscapes you want to include when searching models to populate the global collection.

   Click OK.
   The criteria that you selected is added to the compound expression.

   NOTE
   For more information about searches, see the Administrating *section.*

Click OK.

The Create Global Collection dialog closes and the global collection is created. The global collection appears in the Navigation panel under the Global Collections folder.

Create a Global Collection of Static Members

You can create a global collection of static members on the fly from a topology view.

Follow these steps:

1. In a topology view, do one of the following steps to designate the modeled devices that you want to add to a global collection:
   – To select a single modeled device, right-click a modeled device in the Navigation panel and select **Add To**, **Global Collection(s)**.

     NOTE
     You can right-click a single modeled device in *any* topology view to add the modeled device to a static global collection.
   – To select multiple modeled devices in any topology view, do the following steps:
     a. Press and hold the SHIFT key and individually select the modeled devices.
     b. While the SHIFT key is pressed, right-click the last selected modeled element and select **Add To**, **Global Collection(s)**.

     NOTE
     You can also select one or more modeled elements in the List tab and add them to a global collection.

   The Select Global Collections dialog opens.
2. Click Create.
   The Create Global Collection dialog opens.
3. Complete the following fields as needed:
   – **Name**
   Specifies the name for the global collection.

NOTE
If your global collection name matches an existing global collection name, a warning appears. Click Yes to continue naming the global collection, or click No to enter another name for the global collection.

Owner

Specifies the user who is responsible for the global collection.

- **Description**
  (Optional) Specifies a description for the global collection.
- **Security String**
  (Optional) Specifies a security string expression to prevent certain users from viewing the contents of this global collection.

  NOTE
  For information about security string expressions, see the Administration section.

Click OK.

The global collection of static members is created and appears in the Navigation panel under Global Collections.

Add Static Members to a Global Collection

You can add static members to an existing global collection.

Follow these steps:

1. In any topology, do *one* of the following steps to designate the modeled elements that you want to add to a global collection:
   – **Single modeled element selection:** In the Navigation panel, right-click a modeled element and select **Add To**, **Global Collection(s)**.
   The Select Global Collections dialog opens.

   NOTE
   Alternatively, you can right-click a single modeled element in a topology view and can select **Add To**, **Global Collection(s)**.
   – **Multiple modeled element selection:** To multiselect modeled elements in a topology view, take the following steps:
   a. Press and hold the SHIFT key and individually select the modeled elements.
   b. While the SHIFT key is pressed, right-click the last selected modeled element and select **Add To**, **Global Collection(s)**.
   The Select Global Collections dialog opens.

   NOTE
   Or you can multiselect one or more modeled elements in the List tab and can add them to an existing global collection.
2. Select the name of the global collection where you want to add the modeled elements, and click OK.
   The static members are added to the global collection.

Remove Static Members from a Global Collection

You can remove static members from an existing collection.

Follow these steps:

1. In the Global Collections navigation tree, right-click the static member that you want to remove from the collection and click Remove.
   A dialog prompts you to confirm the deletion.

2. Click Yes.
   The static member is removed from the collection.
   The Remove operation removes the element from the collection, but it does not destroy the modeled element. If the modeled element exists in other topologies, it continues to exist in those topologies. If the modeled element does not exist in any other topology, it is placed in the Lost and Found and later destroyed.

   NOTE
   If you attempt to remove a *dynamic* member from a collection, an error message appears. The error message informs you that the selected member was added through a search criterion. In this case, redefine the search criteria to remove the dynamic member.

Edit Dynamic Members in Existing Global Collections

You can edit dynamic members in existing global collections.

Follow these steps:

1. In the Explorer tab of the OneClick Navigation panel, navigate to the Global Collections node.
2. Right-click the collection and select Edit Global Collection.
   The Edit Global Collection dialog opens.
3. Edit the following fields as needed:
   - **Name**
     Specifies the name for the global collection.

     NOTE
     If your global collection name matches an existing global collection name, a warning appears. Click Yes to continue naming the global collection, or click No to enter another name for the global collection.
   - **Owner**Specifies the user who is responsible for the global collection.
   - **Description**
     (Optional) Specifies a description for the global collection.
   - **Security String**
     (Optional) Specifies a security string expression to prevent certain users from viewing the contents of this global collection.

     NOTE
     For information about security string expressions, see the Administration section.
4. Click Search Options to modify the search settings for this global collection.

   NOTE
   If the search criteria of a global collection has been identified as having the capacity to result in degraded performance, make an audit.

   Sometimes you cannot mitigate the potential performance impact by changing the search criteria. In this case, change the global collection Update Options to only update the collection members at a scheduled time.
   To schedule the membership update:
   Select the Schedule button.

   NOTE

   - Avoid scheduling multiple updates at the same time, because it increases the potential to affect DX NetOps Spectrum performance.
   - In a DSS environment that encompasses multiple time zones, the scheduled update times are local to each of the SpectroSERVERs. Consider this behavior when scheduling updates for any global collection that spans multiple landscapes.

WARNING
If a schedule is applied at the same time that the search criteria are changed, some landscapes could be updated when the changes are committed. This behavior is a known anomaly. To avoid this behavior, apply a schedule and commit the change before you change the search criteria.

5. Click Landscapes to modify the landscapes for this global collection.
6. Click OK

Your changes are saved and the dialog closes.

Auditing Global Collections

Several reasons could prompt you to audit a global collection, for example:

- Performing general housekeeping (identifying which collections are no longer needed).
- Determining who changed attributes of a collection, such as the update interval, search criteria, or name.
  The following events can be used to gain information about changes that were made to global collections:
  - **Event 0x1a100**
    Generated when the name of a global collection is changed.
  - **Event 0x1a110**
    Generated when the owner of a global collection is modified and indicates the OneClick user that has modified the attribute.
  - **Event 0x1a101**
    Generated when the search criteria is modified. This event indicates when the search criteria was changed and what it was changed to.
  - **Event 0x1a111**
    Generated when the update method of a global collection is changed. You can use this event to determine when the method was changed and what it was changed to.

    NOTE
    If the changes are made to the update method for the global collection using CLI, the 0x1a111 and 0x1a110 events are not generated.
- Determining the mitigation of performance impacting collections.
  When the search criteria of a global collection has the capacity to result in degraded DX NetOps Spectrum performance, make an audit. Using the audit findings, determine how to mitigate best the impact on the performance. Some indications that search criteria for a dynamic global collection could be resulting in degraded DX NetOps Spectrum performance include:
  - A SpectroSERVER performance event of type 0x10f20 or 0x10f21 has been generated on the global collection model.
  - OneClick becomes unresponsive or disconnects from the Tomcat server during the dynamic update for the global collection.

  For either symptom, we recommend that you examine the global collection to determine whether it is necessary. If the collection is still needed, the next step is to look at the search criteria to determine if it can be made more efficient.

Copy Annotations from One Global Collection to Another

You can copy annotations (text) from one global collection and paste them into another global collection.

Note: You must have administrative privileges to copy annotations.

Follow these steps:

1. Expand Global Collections in the Explorer tab and select the global collection from which you want to copy annotations.
2. Click the Topology tab in the Contents panel.

Topology information for the global collection appears.

3. Click  (Edit) in the Topology tab toolbar.

   NOTE
   You must have administrative privileges to put a topology in Edit mode.
4. Select all of the annotations you want to copy and click Copy in the Topology tab toolbar. To select several annotations at once, press the CTRL key and click each annotation.

   NOTE
   If you select all of the annotations, the relative placement of the annotations is preserved.
5. Select the global collection that you want to copy the annotations to from Global Collections, in the Explorer tab.
6. Click the Topology tab in the Contents panel.
7. Click  (Edit) in the Topology tab toolbar and then click Paste.
   The annotations are copied to the global collection you selected.

Copy Models from One Global Collection to Another

You can copy models from one global collection and paste them into another global collection.

NOTE
You must have administrative privileges to copy models.

Follow these steps:

1. Expand Global Collections in the Explorer tab and select the global collection from which you want to copy models.
2. Click the List tab in the Contents panel.
   A list of all of the models in the global collection appears.
3. Select all of the models you want to copy and click Copy in the List tab toolbar. To select several models at once, press the CTRL key and select each model. To select a group of models at once, press the Shift key and select one model in the list. Then, select another model in the list. All of the models between the two selected models are also selected.
4. Select the global collection that you want to copy the models to from Global Collections in the Explorer tab.
5. Click the List tab in the Contents panel and then click Paste in the List tab toolbar.
   The models are copied to the global collection you selected.

Find the Global Collections for a Model

You can determine whether a model belongs to a global collection of dynamic members, static members, or both.

Follow these steps:

1. Select the model for which you want to view global collection information.
2. Click the Information tab in the Component Details panel and scroll down to the Global Collections Memberships subview.
3. Expand the Global Collections Memberships subview.
   The Static Global Collection Memberships subview and the Dynamic Global Collection Memberships subview appear.
4. Expand either subview to review the global collections that the model belongs to. If the model does not belong to any global collections, the tables within these subviews are empty.

Create a Global Collection Hierarchy

If you intend to organize your global collections using folders, set up the OneClick Navigation panel with a Global Collection Hierarchy. In this Global Collection Hierarchy, you can create multiple levels of folders to represent previously defined global collections.

Follow these steps:

1. In the Explorer tab of the Navigation panel, right-click the Global Collections Hierarchy node and select New Folder.
2. Type a descriptive folder name in the New Folder dialog and click OK.
   The folder appears in the Global Collection Hierarchy tree.
3. In the Global Collection Hierarchy tree, perform any of the following tasks:
   - **Build more top-level folders:** Repeat Steps 1 and 2.
   - **Build one or more subfolders:** To create a subfolder, right-click a top-level folder and select New Folder.
   - **Populate a folder at any level:** To populate a folder with one or more collections, follow these steps:
     a. Right-click the folder and select Add Global Collections.
     b. In the Select Global Collections dialog, select the name of the collection you want to add and click OK.

NOTE
The Select Global Collections dialog represents the list of collections that are previously created in the Global Collection topology.

Modeling Manually in the World Topology View

You can represent your network geographically by creating a World topology view. In a World topology view, you can model several layers of container views to depict your network locations. For example, you can create container views of network infrastructure from a national or regional level all the way down to an individual room that contains network equipment.

How to Model Locations

When modeling multiple containers representing locations within your network infrastructure, it is recommended that you use this process:

Step 1: Create top-level location views: From the World topology node, you can begin depicting the top-level view of any network location by modeling one of the following top-level containers:

- Building
- Site
- Region
- Country

Step 2: Create one or more sublevel location views: Depending on the top-level container that is modeled, you can then depict one or more sublevel containers.

For this top-level container:	You can model any of these sub-containers:
Building	Floor, Room, or Section
Site	Building
Region	Building or Site
Country	Region, Building, or Site

Step 3: Populate a room container view with modeled devices: After you have created a room container view, you can populate that view with modeled devices.

NOTE
The best practice is to model devices in the Universe topology view and then copy and paste these devices to the World topology view. You can use the **Create Model by IP** or the **Create Model by Type** option to manually model devices in a World topology view. However, this alternative approach is not recommended, because the Universe topology views represent the connectivity views of your network.

Define a Top-Level or Sub-Level Location View

You can create a top-level and sublevel location view, as well as populate a room container view with modeled devices.

Follow these steps:

1. In the Explorer tab of the OneClick Navigation panel, do one of the following steps:
   - **To define a top-level location view:** Click the World topology node to display the World topology view in the Topology tab of the Contents panel.
   - **To define a sublevel location view:** Click one of the top-level view folders appearing under the World topology node. The World topology view for that folder appears in the Topology tab of the Contents panel.
2. Click
   
   (**Creates a new model by type**) in the Topology tab toolbar.
   The Select Model Type dialog opens.
3. In the Container tab of the Select Model Type dialog, select a container type that best describes the network location you are depicting. Click OK.
   The Create Model of Type dialog opens.
4. Specify a name that best describes the network location in the Name field.
5. Specify a security string in the Security String field if you want to secure this view from certain users.

   NOTE
   For more information about securing views, see Provision Access to Modeled Elements.
6. Click OK.
   The named icon container appears in the top-level (or sublevel) view of the World topology. The named folder representing the container appears in the Navigation panel under the World topology node.
7. (Optional) Click the Edit mode button in the World topology view to move the container icon or annotate this view further.

To populate a Room container view with modeled devices:

1. Go to the Universe topology view and copy the modeled devices that you want to display in a World topology view.
2. In the World topology view, navigate to a room container-type view.
3. In the room container-type view, paste the modeled devices.
4. Move the modeled devices to the desired location within this view.

   NOTE
   You cannot paste the same model into two different room containers. For example, you try to paste these same models into a different room container. You are asked whether you want to move them to the new room container or to keep them in the original room container.

If you determine you want to model new devices directly in the World topology, you can use one of these options:

- The Create Model by IP option
- The Create Model by Type option

Modeling Manually in the TopOrg Topology View

You can manually model your network in the TopOrg topology when you want to group infrastructure models by organizational units or by services. For example, you can create a TopOrg topology view that depicts devices that are essential for supporting a network service, such as email. You can also depict services by department or by individual responsibility.

> **NOTE**
> When you populate a TopOrg topology view, copy the modeled elements from a Universe topology view and paste them into a TopOrg topology view. Or you can use the Create Model by IP or the Create Model by Type option to manually model devices in a TopOrg topology view. However, we do not recommend this alternative because the Universe topology views represent the connectivity views of your network.

How to Model Services in the TopOrg Topology

Within the **TopOrg** topology, you can create multiple levels of containers. These multiple container levels represent organizations or individuals responsible for tracking the performance of mission critical services in your IT infrastructure.

When modeling multiple organizational containers in the **TopOrg** topology, consider following this process:

1. **Create layers of ownership or responsibility:** Using the Model by Type dialog, depict one or more containers that represent a department, individual, customer, or enterprise that is:
   - Supported by a network service, or
   - Responsible for tracking the performance of a network service.
2. **Populate Service_Owns container with supporting devices:** Populate the Service_Owns container with the modeled devices supporting the network service. You can populate these containers by copying and pasting modeled devices from the Universe topology to the **TopOrg** topology. You can also populate these containers by defining new devices using the Create Model by IP Address dialog.

Define Service-Related Organizational Views

You can create organizational and **Service_Owns** containers as well as populate a service_owns type container with modeled devices.

Follow these steps:

1. In the Explorer tab of the OneClick Navigation panel, click the TopOrg topology node.
   The TopOrg topology view displays in the Topology tab of the Contents panel.
2. Click
   
   (Creates a new model by type) in the Topology tab toolbar.
   The Select Model Type dialog opens.
3. Click the Containers tab, select a container type that best describes the organization you are depicting, and click OK.
   The Create Model of Type dialog opens.
4. Specify a name that best describes the organization that is responsible for tracking the performance of the network services or that these services supports.
5. (Optional) Specify a security string in the Security String field when you want to secure this view from certain users.
6. Click OK.
   The named container icon appears in the top-level view of the TopOrg topology. The named folder representing the container appears in the Navigation panel under the TopOrg topology node.
7. Repeat Steps 2 through 6 for each organizational or Service_Owns container you want to depict in the TopOrg topology.

Populate Service_Owns or Org_Owns Containers

Populate the Service_Owns container with the modeled devices supporting the network service. You can populate these containers by copying and pasting modeled devices from the Universe topology to the **TopOrg** topology. You can also define new devices using the Create Model by IP Address dialog.

Follow these steps:

1. Go to the Universe topology.
2. Copy the modeled devices that you want to display in a **TopOrg** topology view.
3. Go to the **TopOrg** topology view.
4. Navigate to a Service_Owns container view.
5. Paste the modeled devices.

> **NOTE**
> To model new devices in the **TopOrg** topology, you can use either the Create Model by IP option or Create Model by Type option.

Using and Deleting Favorites

Using Favorites

The Favorites folder contains modeled elements that the user has tagged for easy reference.

In the Navigation panel Explorer tab, you can add any OneClick element below the landscape level to your Favorites folder by right-clicking the element and choosing Add To, Favorites. You can also add Global Collections to your favorites by right-clicking your Favorites folder and choosing Add Collections.

To remove an element from the Favorites folder, right-click the element within the Favorites folder and choose Remove.

> **WARNING**
> If you right-click the element within the Favorites folder and you choose Delete, the element is removed from the Favorites folder. Plus, some models could also be deleted permanently from the system.

You can create subfolders by right-clicking Favorites (or a subfolder within Favorites) and choosing New Folder. Use the right-click menu to cut, copy, paste, rename, and delete subfolders.

Deleting from Favorites

Consider the following behaviors when deleting elements from the Favorites folder:

- When deleting containers from the Favorites folder, container models are permanently deleted from the system. Any models within the container are sent to the Lost and Found.
- When deleting a global collection from the Favorites folder, the global collection is permanently deleted from the system. Any models within the global collection are removed from favorites but are not deleted from the system.
- When deleting a model from the Favorites folder, the model is permanently deleted from the system. The model is *not* sent to the Lost and Found.

Lost and Found Model Information Subview

The Lost and Found Model Information subview lets you clear unattached models that are stored in the Lost and Found repository. The unattached models are models that have been cut but not pasted, models that Discovery could not resolve, and so on.

To access the Lost and Found Model Information view, select **LostFound** in the Navigation panel and select the Information tab in the Component Detail panel.

The Lost and Found model information view includes the following options:

- **Automatic Model Destruction**
  Specifies whether models in the Lost and Found are destroyed at specified times.
- **Next Model Destruction Date and Time**
  Specifies the next scheduled date and time at which models in Lost and Found are destroyed when Automatic Model Destruction has been enabled. The value of the Model Destruction Interval determines this value.
- **Model Destruction Interval**
  The interval (in seconds) at which models in Lost and Found are destroyed when Automatic Model Destruction has been enabled.
  Default: 24 hours

Device and Interface Threshold Settings

DX NetOps Spectrum includes several device and interface alarms configured with three variables to define the alarm and reset conditions:

- **Threshold Setting**
  Specifies the threshold setting above which an alarm condition can exist.
- **Reset Level**
  Specifies the reset level below which an existing threshold alarm condition is automatically cleared. DX NetOps Spectrum does not generate subsequent alarms for the parameter until the value falls below the reset level.
- **Allowed Threshold Violation Duration**
  Specifies how long the parameter can be greater than the threshold setting, in seconds, before DX NetOps Spectrum generates an alarm.

You can configure device thresholds so that an alarm is generated if a given threshold is exceeded for a certain duration.

> **NOTE**
> To configure device and interface threshold alarm settings, the Device Threshold attribute on the SpectroSERVER must be enabled. You can turn off individual device or interface thresholds by setting the threshold and reset level variables equal to zero.

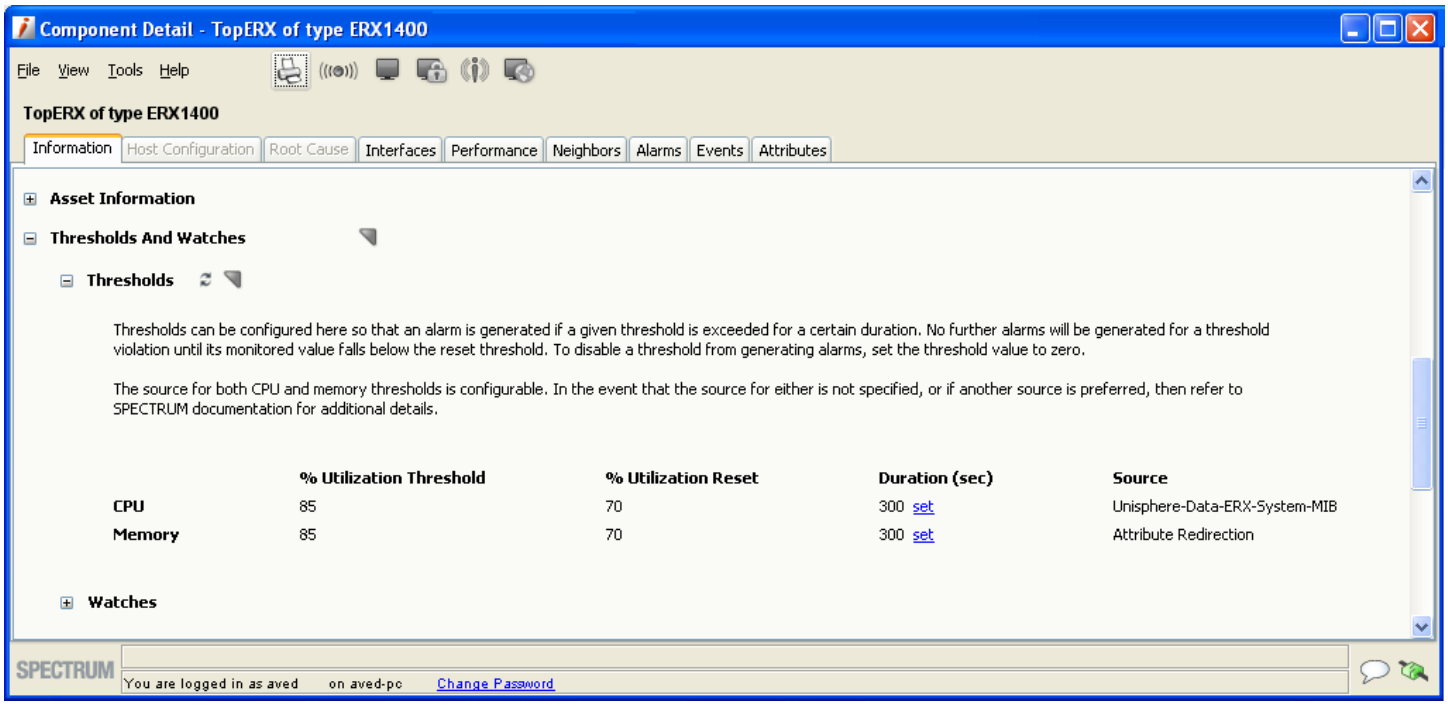
Device Threshold Settings

You can configure device thresholds so that an alarm is generated if a value is exceeded for a certain duration. No further alarms are generated for a threshold violation until its monitored value falls below the reset threshold. To disable a threshold from generating alarms, set the threshold value to zero.

The device threshold settings available are ‘% CPU Utilization’ and ‘% Memory Utilization’. You can access these OneClick settings in the following ways:

- The Thresholds and Watches subview in the Information tab for the selected device.
- The Attribute Editor, Thresholds, Device Thresholds grouping.
  When using the Attribute Editor to set the Device Threshold attributes, the Allowed Threshold Violation Duration attribute is not available, and the default value of 300 seconds (five minutes) is used.

You can view the source that is used to calculate the CPU and memory utilization for a device in the Thresholds and Watches, Thresholds subview in the Information tab of the Component Detail panel.



Example % CPU Utilization Default Settings

This example illustrates how the default % CPU Utilization Threshold settings work together to trigger an alarm. The example is illustrated in the following graphic. The default settings for this alarm setting are as follows:

- % CPU Utilization Threshold = 85%
- % CPU Utilization Reset = 70%
- Allow Threshold Violation Duration = 300 seconds

Using the default settings, when the device's % CPU Utilization parameter exceeds the threshold setting of 85% at time Y, DX NetOps Spectrum begins the 300 second Allowed Threshold Duration timer.

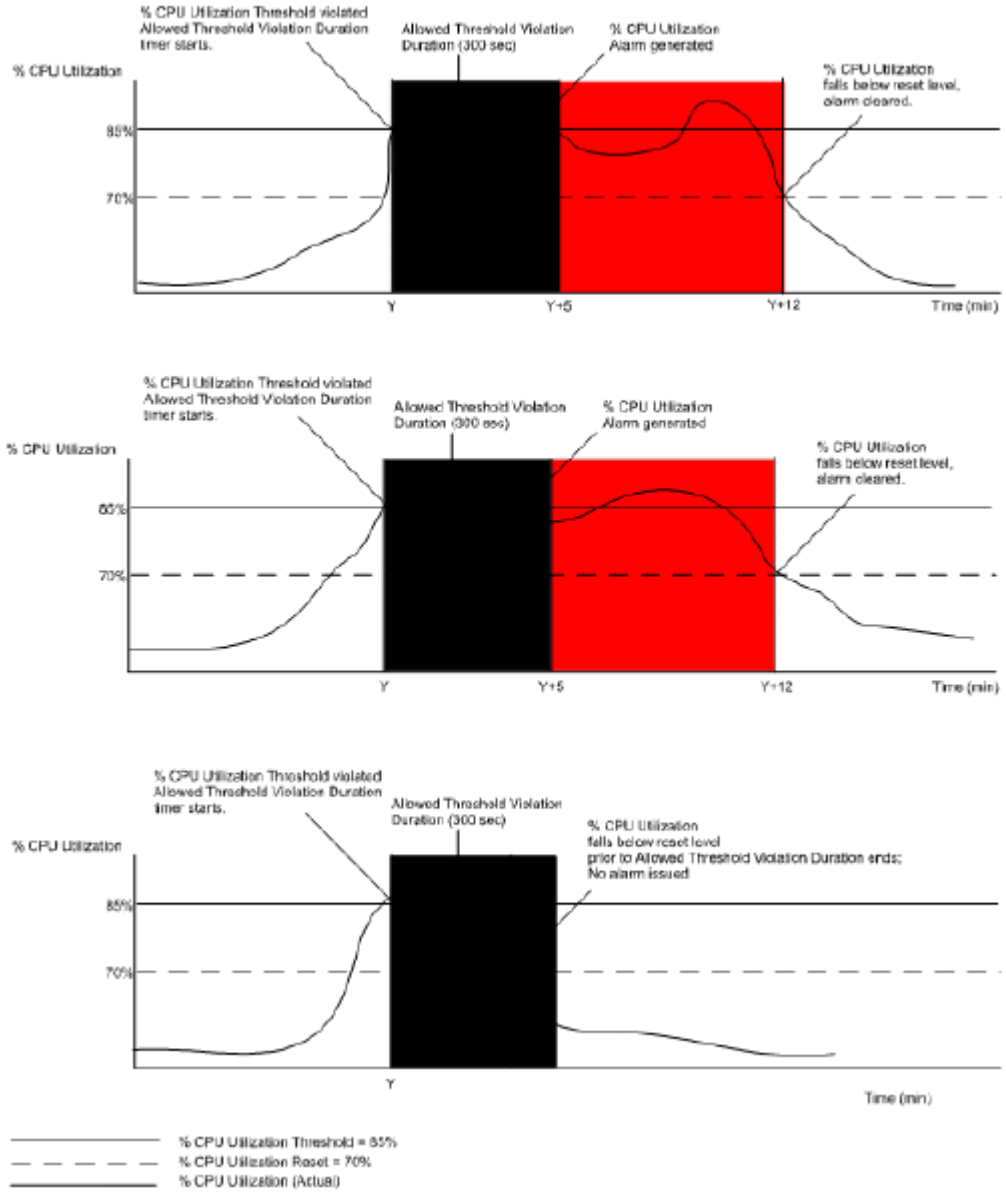
The % CPU Utilization does not fall below the reset value of 70% for the duration of the timer.

At time = Y+5 minutes, DX NetOps Spectrum triggers a % CPU Utilization alarm for the device.

DX NetOps Spectrum will not generate another % CPU Utilization alarm until this alarm is cleared manually or automatically.

At time = Y+12 minutes, the device's % CPU Utilization falls below the reset value of 70%. DX NetOps Spectrum clears the % CPU Utilization alarm for the device.

Illustration: % CPU Utilization Default Settings



Interface Threshold Settings

The following interface thresholds parameters are available:

- **% Utilization Threshold**
  Defines the level of port capacity used that triggers an alarm condition for a port.
- **Packet Rate Threshold (packets/sec)**
  Defines the number of packets per second that triggers an alarm condition for a port.
- **% Error Rate Threshold**
  Defines the error rate on a port that triggers an alarm condition.
- **% Discarded Threshold**
  Defines the percentage of discarded packets on a port that triggers an alarm condition.

Each of these attributes has a reset value and Allowed Threshold Violation Duration timer attribute setting. You can access these OneClick settings in the following ways:

The Thresholds and Watches subview in the Information tab for the selected device interface.

- The Attribute Editor, Thresholds, Interface Thresholds grouping.

NOTE
See Threshold Attributes for information about accessing Device Threshold settings using the Attribute Editor.

Update Device Interface and Connection Information

DX NetOps Spectrum can perform automatic discovery and mapping of a device's interfaces and connections based on the following events and conditions:

- A change in the number of configured interfaces on a device.
- When a device sends a LINK up trap.
- When DX NetOps Spectrum reconfigures a modeled device.

OneClick administrators can also manually update this information about a modeled device. See Using OneClick for information about viewing a device's interface, sub-interface, and connection information.

Automatic Updates of Device Interface and Connection Information

You can use the following attributes to configure DX NetOps Spectrum to automatically update interface and connection information about a device.

- Automatically Reconfigure Interfaces
- Discover Connections After Link-Up Events
- Create Subinterfaces
- Discovery After Reconfigure
- Topologically Locate Model

Automatically Reconfigure Interfaces

When this attribute is set to Yes, DX NetOps Spectrum monitors the device for a change in the number of configured interfaces. If it detects a change, DX NetOps Spectrum automatically updates the device model to reflect the interface changes. The updated interface information appears in the Interfaces view for the device.

Discover Connections After Link-Up Events

DX NetOps Spectrum automatically discovers and maps a model's connections one poll interval after it receives a Link Up trap from a device when the Discover Connections After Link-Up Events attribute is set to Yes. This delay lets the device fully reconfigure its related SNMP tables before DX NetOps Spectrum reads them. The Poll Interval setting for the device appears in the Information view, SPECTRUM Modeling Information subview.

Special Considerations for Flapping Interfaces

A "flapping" interface is one that is constantly coming up and going back down, usually because of a problem on the device. When Device Discovery After Link-Up Events is set to 'Yes,' DX NetOps Spectrum excludes LinkUp traps from flapping interfaces. As a result, the stream of LinkUp traps from a flapping interface does not interfere with a LinkUp trap for another interface on the same device. The connection discovery action can run as expected.

When DX NetOps Spectrum detects a flapping interface, a minor alarm is generated on the related device. After a default interval of 10 minutes without receiving a trap from that interface, the alarm clears. The default settings that are used to

identify and track flapping interfaces are configured using Event Rules associated with Events 0x220002, 0x220006. The default settings are as follows:

- An Event Sequence Rule on Event 0x220002, which generates Event 0x220006 if a LinkUp trap is received and is followed by a LinkDown trap from the same interface within 60 seconds.
- An Event Rate Window Rule on Event 0x220006, which generates Event 0x220007 if 15 0x220006 events are generated within 5 minutes (300 seconds). Event 0x220007 generates a Minor alarm on the device.
- An Event Pair Rule on Event 0x220006 which generates Event 0x220008 if Event 0x220006 is not followed by Event 0x220007 within 10 minutes (600 seconds). Event 0x220008 clears the minor alarm generated by Event 0x220007.

The default values generate an alarm after 15 LinkUp/Down trap pairs are received. Also by default, the alarm is cleared 10 minutes after the last LinkUp/Down Trap pair is received. You can modify these settings by defining flapping interface event thresholds. The applicable rules are specified in the *<$SPECROOT>*/SS/CsVendor/IETF/EventDisp file.

> **NOTE**
> For information about manually editing the files, for information about DX NetOps Spectrum events and event rules, and for information about changing the event rules associated with events 0x220002 and 0x220006, see the Event Configuration .

Create Sub-Interfaces Attribute

When this attribute is set to Yes, and the modeled device supports RFC 1573, DX NetOps Spectrum models the sub-interfaces for the device. DX NetOps Spectrum differentiates between physical and logical interfaces. It creates sub-interfaces using the logical interface information that it gathered from the device. A sub-interface appears in the Interfaces tab for a device, nested beneath the logical interface where it is configured.

Discovery After Reconfigure

When this attribute is set to Yes, DX NetOps Spectrum rediscovers device connections each time that the device model is reconfigured. Both an interface reconfiguration and a manual device reconfiguration trigger a rediscovery when this attribute is enabled.

Topologically Relocate Model

When this attribute is set to Yes, DX NetOps Spectrum determines whether a device model must be moved to a different topology during a Discovery. DX NetOps Spectrum moves the device if necessary, based on updated connection mapping.

Manually Updating Interface and Connection Information

You can manually initiate an interface reconfiguration and connection discovery using the following options in the Tools, Reconfiguration menu and the Reconfiguration subview on the Information tab:

- Reconfigure Model
- Discover Connections
- Rediscover SNMP MIBs
- Rename Interface Models
- Reevaluate Model Name
- Reevaluate NCM Device Family

About Reconfiguring Models

When you activate a Reconfigure Model action, DX NetOps Spectrum finds the interfaces on the device and updates the device interface modeling.

NOTE
Reconfigure Model actions do *not* change the model type. To change the model type, either run the NewMM.pl script or delete and remodel the device by IP address. For more information about running NewMM.pl, see Install DX NetOps Spectrum .

The following parameters are reevaluated during a Reconfigure Model action:

- **Device Type**
  Verifies the current Device Type attribute value.
- **Model Name**
  Checks the Model Naming Order setting on the VNM, and determines whether the device model requires a change.
- **Application Discovery**
  Performs a Reconfigure SNMP MIBs action.
- **Interface Discovery**
  Determines which interfaces exist on a device, and updates the device modeling as needed.
- **Normalized Source**
  Verifies the attribute to use for gathering device CPU and memory usage information.
- **Serial Number**
  Verifies the device serial number, if available, and updates the device model if necessary.
- **802.3ad Trunk Memberships**
  Checks to see if the device interfaces are members of a 802.3ad trunk.
- **NCM Device Family**
  Checks the Device Family value that Network Configuration Manager uses to groups devices by vendor.

Reconfigure a Model

You can reconfigure a model to update device interface information. When you activate a Reconfigure Model action, DX NetOps Spectrum finds the interfaces on the device and updates the device interface modeling.

Follow these steps:

1. Locate the model.
2. Right-click the model and select Reconfiguration, Reconfigure Model.
   The Reconfigure Model dialog opens, showing the progress of the requested action.
3. Click OK.
   The Reconfigure Model dialog closes. The model is reconfigured.

Discover Connections

Starting from this release, you can run Discover Connections on multiple devices. When you run the Discover Connections command, DX NetOps Spectrum performs a Discovery on the selected devices. Discovery data lets DX NetOps Spectrum update and remap the device model connection information. You can also use this functionality to discover connections in a network container.

NOTE

If the discovery is already running on a device, the remaining devices are added to the Discovery queue. A pop-up message appears with the following message:"Another discovery process is currently in progress. Only one discovery process at a time is allowed per <Landscape Name> landscape, queuing request for background processing."

To know how to view the queue, refer View Queue.

Follow these steps:

1. Locate the container for which you want to discover connections in the Explorer tab.
2. Right-click the container, and select Reconfiguration, Discover Connections.

The Discover Connections dialog shows the progress of the requested action. If connections are successfully discovered, this dialog indicates success.

3. Click OK.
   The Discover Connections dialog closes. The connections among the devices in the selected LAN container now appear as pipes in the Topology view.

Discover Connections Queue

The Discover Connections Queue feature allows you to initiate multiple discover connectivity tasks (for example, discover connections after reconfiguration) by adding them to a queue. You do not have to wait for a discovery task to complete to initiate a new one. When the previous task completes, the discovery tasks that are added to the Discover Connections Queue start automatically. You can review the status of a discovery task in the Events tab.

If a mapping process is in progress and you initiate a new Discover Connection action, the following message appears:

“Another discovery process is in progress. Only one discovery process at a time is allowed per landscape, so it is queued.”

NOTE

If you trigger the Discover Connections action through REST API, you get the message as 'SUCCESS' when the action is queued.

Also, if the “Discovery After Reconfigure” or the “Discover Connection After Link-up Events” option is set to 'Yes' then the Discover Connectivity action triggers are queued to Discover Connections Queue.

Once the task that is already running completes, a new task in the queue starts. When you select a model, the discovery status messages for that model appear in the Events tab.

View Discover Connections Queue

To view the queue, run the following CLI command on the TopologyWrkSpc model. The queue entries are written to the $SPECROOT/SS/ADiscDebug file.

NOTE

To enalbe the ADiscDebug file, go to VNM -> AutoDiscovery Control -> Debug Options -> Debug AutoDiscovery then set the value to 'ON'.

./update action=0x10601 mh=themodelhandleofthe"TopologyWrkSpc"model

Example: > ./update action=0x10601 mh=0x10003aupdate action: successfulResponse has 0 attributes:

In the ADiscDebug file, the following message appears: 'Dumping the pending rediscovery models'

Clear Discover Connections Queue

To clear the discover connection queue, run the following CLI command on the TopologyWrkSpc model:

./update action=0x10602 mh=themodelhandleofthe"TopologyWrkSpc"model

Example: > ./update action=0x10602 mh=0x10003aupdate action: successfulResponse has 0 attributes:

In the ADiscDebug file, the following message appears:
'Clearing the pending rediscovery queue'

Rediscover SNMP MIBs

When a device model is created, DX NetOps Spectrum automatically creates models for each of the major and minor applications the device supports. Click Reconfigure SNMP MIBs to retrieve application support information from the device. The application models for the device are updated with any changes.

Rename Interface Models

Use this function to update a device's interface model names after changing the device's Interface Name Primary Suffix attribute or the Interface Name Secondary Suffix attribute. Using this command forces DX NetOps Spectrum to rename the interface models using the current values of both the primary and secondary suffixes for the interface model. Some of the suffix options include ifName, ifAlias, ifDescr, and ifIndex.

Entity Table Interface Stacking

When modeling interfaces, DX NetOps Spectrum uses the information contained within the MIB II ifStackTable to determine their logical stacking. For example, in the case of a frame relay interface with DLCI sub-interfaces, DX NetOps Spectrum attempts to stack the interfaces using information in the ifStackTable.

If you set the use_if_entity_stacking (0x12a83) attribute to TRUE on a device model in the Attributes tab, DX NetOps Spectrum attempts to use information from RFC2737 (Entity MIB) to determine interface stacking if the ifStackTable method fails. If an interface does not support ifStackTable, but the interface *does* support the Entity MIB, DX NetOps Spectrum will attempt to stack the interface model using information in the entPhysicalTable.

Note: This is done on a case-by-case basis as some vendors do not implement the RFC2737 indexing scheme correctly, which can cause interfaces to be incorrectly stacked.

Reevaluate Model Name

Determines whether to change the device's model name based on the VNM Model Naming Order setting for the VNM managing the device. See SpectroSERVER Control Subview for information about the VNM Model Naming Order setting.

Reevaluate NCM Device Family

Automatically places a device in the proper device family after a firmware upgrade. For example, if you have a Cisco device that appears in the CatOS family and you then upgrade this device with new firmware and it changes to CiscoIOS, the device does not switch its family automatically. Instead, you can update it using the Reconfigure menu.

Access Interface and Connection Update Controls

You can access to the interface and connection update controls described in this section as shown in the following table:

Attribute	Tools, Reconfiguration Menu	Reconfiguration	Attribute Editor
Automatically Reconfigure Interfaces		X	X
Discovery Connections After Link Up Events		X	X
Discovery After Reconfigure		X	X
Create Sub-Interfaces		X	
Topologically Relocate Model		X	X
Reconfigure Model	X	X	
Discover Connections	X	X	
Rediscover SNMP MIBs	X	X	
Rename Interface Models	X	X	
Reevaluate Model Name	X	X	
Reevaluate NCM Device Family	X		

Tools, Reconfiguration Menu

The Tools, Reconfiguration menu provides quick access to reconfiguration actions you can perform on a selected device model. You can also access this menu by right-clicking the device you want to reconfigure.

Reconfiguration Subview and Advanced Subview

The Reconfiguration subview provides access to attributes that control when DX NetOps Spectrum updates a device's interface, connection, and topology information. You can also manually reconfigure a device and discover a device's connections from this subview.

The Advanced section of the Reconfiguration subview provides access to individual model reconfiguration functions that occur as part of the Reconfigure Model function. In some cases, you may want to perform these actions separately instead of performing an overall Reconfigure Model action.

Attribute Editor

Use the Attribute Editor to access some of the interface and connection update parameters for many models or modeled devices.

The Discover Connections Queue feature in the latest release of DX NetOps Spectrum allows you to initiate multiple discovery tasks by adding them to a queue. You do not to wait till a discovery task to complete to initiate a new one. The discovery tasks added to the Discover Connections Queue are started automatically if the previous task is complete. You can review the status of a discovery task in the Events tab.

Redundant Connections

If a modeled device is configured with a pool of IP addresses available to it for use in communicating on the network, DX NetOps Spectrum can use these IP addresses to create redundant connectivity with that device. If the redundancy feature is enabled on a device and DX NetOps Spectrum cannot reach the device using the designated primary address, DX NetOps Spectrum attempts to re-establish contact using the list of available IP addresses.

Redundancy Preferred Addresses List

A device that supports the DX NetOps Spectrum redundancy feature has a Redundancy Preferred Addresses list containing the device's interface IP addresses that is created when the router is originally modeled. DX NetOps Spectrum uses this list in determining redundant connectivity to the device. Devices have a primary address that is determined when the device is modeled. The DX NetOps Spectrum modeling process includes loopback functionality. If the VNM is configured to use the loopback feature, the first valid loopback address detected for the device is used as device model's primary address.

Removal of Shared IP Addresses from Preferred List

DX NetOps Spectrum automatically removes the IP addresses that it detects as "shared" from a Redundancy Preferred Addresses list for a device model. DX NetOps Spectrum places these shared addresses in the Redundancy Excluded Addresses list. Such shared IP addresses are not used when DX NetOps Spectrum attempts to restore communication with a lost device that was using redundant IP addresses.

> **NOTE**
> If you manually add a shared IP address to the Redundancy Preferred Addresses list, DX NetOps Spectrum automatically moves it back to the Redundancy Excluded Addresses list. You cannot see this list until you close and reopen the current view.

Device Primary Address

By default, the primary address for a device is the IP address assigned to the device for network communications. You can change the primary address in OneClick if the network address for the device changes. If DX NetOps Spectrum cannot contact the device using its primary address, it attempts to contact the device using the first IP address in the Redundancy Preferred Addresses List.

You can change a device's primary address and the IP addresses listed in the Redundancy Preferred Addresses list using the IP Redundancy subview.

IP Redundancy Subview

The IP Redundancy subview displays the attributes and settings DX NetOps Spectrum uses to create and monitor redundant communication paths to the device. Access the IP Redundancy subview by selecting a device in either the Navigation panel, the List tab, or the Topology tab, and then selecting the Information tab in the Component Detail panel. The IP Redundancy subview appears in the Information tab.

Enable Redundancy

When this attribute is set to Yes, DX NetOps Spectrum uses the addresses in a modeled device's Redundancy Preferred Addresses list, if it exists, to contact a device when the primary address is not available.

Generate Redundancy Alarms

When this attribute is set to Yes, DX NetOps Spectrum generates an alarm when a device cannot be contacted using its primary address.

Select Preferred Redundant Addresses

The Redundancy Preferred Addresses list displays IP addresses that a device can use for communicating on the network. You can manually add or remove IP addresses to or from the Redundancy Preferred Addresses list for a device.

Follow these steps:

1. Select the device with a redundant IP address.
2. Click the Information tab and expand the IP Redundancy subview.
3. Click Configure next to Primary Address.
   The Preferred Addresses dialog opens.
4. Click Add below the Redundancy Preferred Addresses list.
5. Enter the IP address that you want to add to the list in the Add IP Address dialog, and click OK.
   The IP address now appears in the Redundancy Preferred Addresses list.
6. Click OK.
   The changes to the device's Redundancy Preferred Addresses list are applied.

Exclude Redundant Addresses

The Redundancy Excluded Addresses list displays IP addresses that a device cannot use for communicating on the network. You can manually add or remove IP addresses to or from a device's Redundancy Excluded Addresses list.

Follow these steps:

1. Select the device for which you want to add an excluded IP address.
2. Click the Information tab and expand the IP Redundancy subview.
3. Click Configure next to Primary Address.
   The Preferred Addresses dialog opens.
4. Click Add below the Redundancy Excluded Addresses list.

5. Enter the IP address that you want to add to the excluded list Add dialog and click OK.
   The IP address now appears in the Redundancy Excluded Addresses list.

   NOTE
   You can also select an address in the Preferred list and move it to the Excluded list by clicking the single right arrow located between the two lists. Similarly, you can move an address from the Excluded list to the Preferred list by selecting the address in the Excluded list and clicking the single left arrow located between the two lists.

About Shared IPs in Device Communication

Since any of your shared IP addresses will already have been added to the Redundancy Excluded Addresses list, the DX NetOps Spectrum redundancy intelligence will never attempt to communicate with a device using a shared IP. DX NetOps Spectrum will also never assign a shared IP address to the PrimaryAddress attribute.

Also, if you try to write a new value to a device model's NetworkAddress or PrimaryAddress attribute, if it is a shared IP address, the new value is not written and a warning dialog opens.

Interface Reconfigurations

Device models can potentially reconfigure interfaces at every poll cycle. This reconfiguration increases CPU usage and generates SNMP traffic due to interface table changes or interface stack table changes.

To disable interface reconfiguring, do the following:

- Set the Use_If_Table_Last_Change (0x11f7f) attribute to FALSE to disable interface reconfiguring for an interface table.
- Set the Use_If_Stack_Last_Change (0x130bc) attribute to FALSE to disable interface reconfiguring for an interface stack table.

You can configure a device model to trigger an alarm if it is continually reconfiguring interfaces. By default, a device model triggers a minor alarm if any one of the following sequences occurs in a 31-minute timeframe:

- Six interface reconfigurations for an interface table
- Six interface reconfigurations for an interface stack table
- Both sequences.

Primary IP Address Modification

The primary IP address is the address DX NetOps Spectrum uses to communicate with a modeled device. You can change a modeled device's primary IP address. There are three ways to change a device's primary IP address:

- Change a device's primary IP address in the device's preferred address list.
- Change a device's primary IP address to an interface's primary IP address.
- Change a device's primary IP address to an interface's secondary IP address.

Change the Primary IP Address for a Device in the Preferred Address List

You can change the primary IP address for a modeled device. If you know the IP address that you want to use to contact the device, you can change the address in the preferred address list for that device.

Follow these steps:

1. In the Explorer tab, select the device whose primary IP address you want to change.
2. Click the Information tab and expand the IP Redundancy subview.
3. Click Configure next to Primary Address.

The Preferred Addresses dialog opens. The current primary address for the device appears in the Primary Address field.

4. Select an IP address from the Redundancy Preferred Addresses list on the left side of the dialog, and click Primary.
   The IP address you selected appears in the Primary Address field. The IP address that had originally been the primary address now appears in the Redundancy Preferred Addresses list.
5. Click OK.
   The change to the device's primary address is applied and the Preferred Addresses dialog is closed.

Change the Device IP Address to an Interface Address

You can change the primary IP address of a modeled device to an interface address. In some situations, you do not know the IP address to contact a device, but you do know the device interface that you want to use. You can change the primary IP address for that device by selecting an interface primary IP address.

> **NOTE**
> For more information about interfaces, see the *Operator section*.

Follow these steps:

1. In the Explorer tab, select the device whose primary IP address you want to change.
2. Click the Interfaces tab in the Component Detail panel.
   Interface information for the selected device opens in table format.
3. Right-click the interface you want to use to contact the device, click Configure Primary Address for Device, and then click Use Interface IP As Primary Address to set the interface's primary IP address as the device's primary IP address.
   A confirmation dialog opens.
4. Click Yes.
   The change to the device's primary address is applied.

> **NOTE**
> You cannot select an interface that does not have a primary IP address configured for it (for example, if the IP Address column in the interfaces table is blank). If you try to select the interface, an error message appears.

Change the Primary IP Address for a Device to use an Interface Secondary IP Address

You can change the primary IP address of a modeled device. In some cases, you do not know the IP address that you want to use to contact a device, but you do know the particular interface. You can change the primary IP address for a device by selecting an interface and then selecting one of the interface secondary IP addresses.

> **NOTE**
> For more information about interfaces, see the *Operator section*.

Follow these steps:

1. In the Explorer tab, select the device whose primary IP address you want to change.
2. Click the Interfaces tab in the Component Detail panel.
   Interface information for the selected device opens in table format.
3. Right-click the interface that you want to use to contact the device.
4. Click Configure Primary Address for Device.
5. Click Use Secondary IP to set one of the interface secondary IP addresses as the primary IP address for the device.
   The Interface IP Mask Table opens.
6. Select an IP address to use from the list of secondary IP addresses and click Use as Primary Address for Device.
   A confirmation dialog opens.
7. Click Yes.
   The change to the device's primary address is applied.

NOTE
You cannot select an interface that does not have any IP addresses configured for it (for example, if the Secondary IPs and IP Address columns in the interfaces table are blank). If you try to select the interface, an error message appears.

IPv6 Information

You can view IPv6 information for devices that support RFC2465 and RFC2452 MIBs in the Information tab of the Component Detail panel. The following tables provide specific information:

- IPv6 Interface Configuration Table
- IPv6 Routing Table
- IPv6 Address Table

Editing and Enhancing Topology Views

This section contains the following topics:

- Topology Edit Mode
- Set Topology View Edit Mode Preferences
- Modifying Topology Views

Topology Edit Mode

Topology Edit mode refers to the condition you place a topology view in when you want to edit its appearance. When you place a view into Edit mode, you automatically prevent other users from editing that view. Once in Edit mode you can use its drawing tools to draw rectangles, ellipses, lines, or text boxes. After you create these items in a view, you can later apply styles or colors to them.

The following image shows an example of the Edit mode toolbar that appears when you enter Edit mode:

Access Edit Mode

If your user account has the required privileges, you can access Edit mode to modify the current topology view.

Follow these steps:

1. Click the Topology tab in the Contents panel.
   The Topology view and the Topology toolbar open.
2. Click
   
   (Edit) in the Topology tab toolbar.

The Edit mode toolbar opens. The topology view is locked to prevent other users from editing this view.

3. Modify the topology view.
4. Click Save.
   Your changes are saved, and you exit Edit mode.

Edit Mode Toolbar

The following table describes the editing tools that you can access and use from the Edit mode toolbar.

Tool	Description
	**Move Tool:** Moves modeled elements in a view.
	**Rectangle Tool:** Draws rectangles. Click and hold the rectangle button to access additional tools.
	**Ellipse Tool:** Draws ellipses. Click and hold the Ellipse Tool button to access additional tools.
	**Line Tool:** Draws lines.
	**Text Box:** Creates a text box used to enter text.
	**Font Properties:** Opens the Select Font dialog for the selected text annotation. Choose a font family, style, and size from the respective columns in the Select Font dialog. You can also choose the text foreground and background color and whether to show the text background.
	**Shape Color:** Opens the Select Shape Color dialog for the selected annotation. Select a shape color in the Select Color dialog.
	**Line Weight:** Sets the line weight for lines, ellipses, and rectangles.
	**Line Pattern:** Sets the style for lines, ellipses, and rectangles.
	**Group:** Group selected modeled elements in a view.
	**Ungroup:** Ungroup selected modeled elements in a view.
	**Background Editor:** Changes topology background characteristics (edit mode grid, grid spacing and color, background color, image, and size).
Arrangement	**Arrangement drop-down list:** Contains the following options for arranging the elements in the topology: Radial, Tree, or Manual.

Set Topology View Edit Mode Preferences

You can set preferences to specify how you want Edit mode to behave.

To set topology view Edit mode preferences

1. Click View, Preferences.
   The Set Preferences dialog opens.
2. Expand the Topology Tab folder in the Name column.
3. Click any of the following options to make changes:
   – **Annotation Font**
   Specifies the default font settings for topology annotation text. You can modify font, style, size, and background and foreground colors.
   – **Grid Properties**
   Specifies the following settings for the grid that can appear in the Topology tab in Edit mode:
   • **Show grid:** Set the size of the grid squares using the value displayed using the Show grid option. Decreasing the value decreases the size of the grid squares, while increasing the value increases the size of the grid squares.
   • **Snap to grid:** Enables snap-to-grid while the topology view is in Edit mode, making it easier to align modeled device icons in the topology view.
   – **Initial Zoom**
   Specifies the zoom behavior for topology views when they are first shown.
   – **Show Pipe Label**
   Specifies whether you want to show pipe labels in the topology view.
4. Click OK.
   Your changes are saved and the Set Preferences dialog closes.

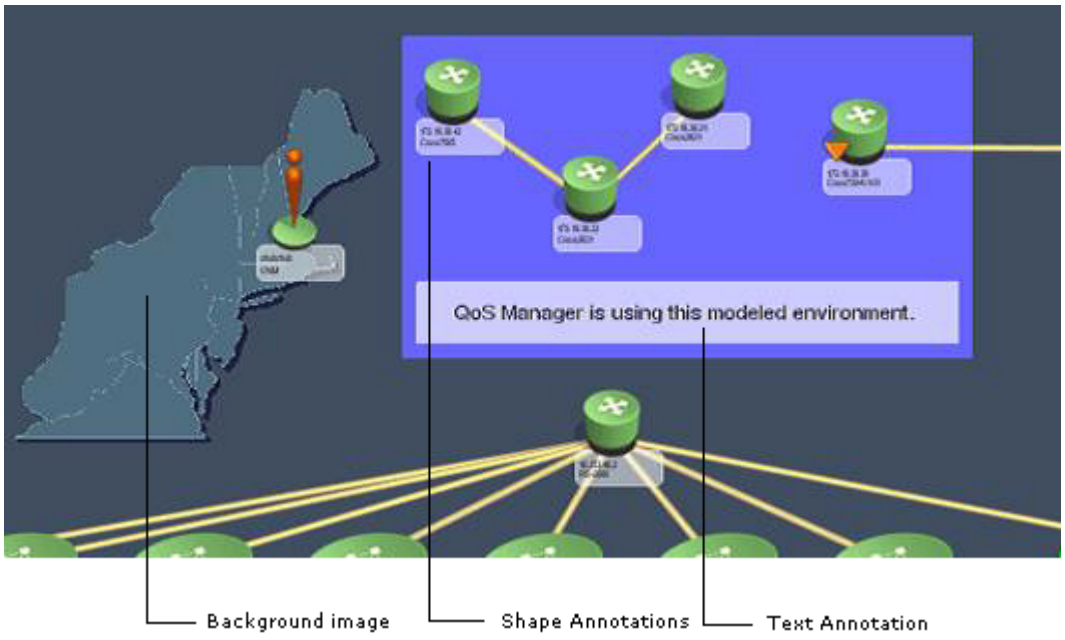
   NOTE
   You can also set these preferences in the Background Editor dialog.

Modifying Topology Views

You can modify the appearance of any topology view by using the Edit feature in the view. Some of the enhancements you can make include the following:

- Change the background characteristics of a view.
- Add lines, rectangles, or ellipses to a view.
- Change the placement of modeled elements in a view.
- Change font characteristics in a view.

The following image shows an example of an enhanced topology view:



Multi-User Considerations

Be aware that OneClick topology view enhancements are shared across all users. Also, when you edit a view using the Edit mode button in a Topology tab toolbar, DX NetOps Spectrum automatically prevents all other users from editing that view until you have finished.

Resize Model Icons

You can resize model icons displayed in a topology view.

Follow these steps:

1. Switch to Edit mode, as described in Access Edit Mode.
2. Select the model icon that you want to resize in a topology view.
   A green box appears around the icon.
3. Click one of the corners of the green box, and drag the icon to resize it.
   The icon size changes proportionally.
4. Deselect the icon when it reaches the desired size.
5. Click Save.
   The model icon is resized, and you exit Edit mode.

Add Shapes, Lines, or Text to a View

You can add rectangles, ellipses, lines, or text in a topology view.

Add a rectangle or ellipse to a topology view in Edit mode.

Follow these steps:

1. Switch to Edit mode as described in Access Edit Mode.
2. In the Edit mode toolbar, select the desired style for the shape from the menu by clicking the Rectangle Tool or the Ellipse

   Tool 

The pointer changes from an arrow to a crosshair when hovering over the background area of the topology view.
3. In the desired location, click and drag the pointer starting at the upper left corner of the shape and ending at the lower right-hand corner of the shape.
4. Release the mouse button.
The shape you created now appears in the view, behind any existing models or pipes.

Add a line to a view in Edit mode.

Follow these steps:

1. Switch to Edit mode as described in Access Edit Mode.
2. In the Edit mode toolbar, click the line tool  .
The pointer changes from an arrow to a crosshair.
3. In the topology view, click and drag the pointer to draw the line.
4. Release the mouse button.
The line appears in the view, behind any existing models or pipes.

Add text to a view.

Follow these steps:

1. Switch to Edit mode as described in Access Edit Mode.
2. In the Edit mode toolbar, click the text box  .
The pointer changes from an arrow to a crosshair pointer.
3. Click in the location where you want the text box to begin.
A text box appears.
4. Type the text in the text box.
5. Click outside the text box to exit.
The text boundaries of the text box disappear. The text is placed in the background of the topology view.

Change Shapes, Lines, and Text Characteristics

You can apply different font properties to text, colors to shapes, or line weights to lines.

> **NOTE**
> We recommend setting properties for shapes, lines, or text before adding (or drawing) these elements in a view as a best practice.

You can change text font properties in a topology view.

Follow these steps:

1. Switch to Edit mode as described in Access Edit Mode.
2. In the Edit mode toolbar, click the text you want to change and click Font Properties  .
The Select Font dialog opens.
3. Select the desired font family, style, size, foreground color, and background color.
The preview pane shows the font properties that you selected.
4. Click OK.
The font properties are applied to the selected text.

You can apply color to shapes or lines in a topology view.

Follow these steps:

1. Switch to Edit mode as described in Access Edit Mode.
2. Click the shape or line that you want to edit.
3. in the Edit mode toolbar, click Shape Color  .

   The Select Color dialog opens.
4. Modify the color settings by clicking each of the tabs in the Select Color dialog:
   - **Swatches**
     Specifies the color of the shape or line. Select a color from the palette. A preview of the selected color appears at the bottom of the dialog. If you selected and previewed multiple colors, the colors you have chosen appear in the Recent color grid for re-selection.
   - **HSB**
     Specifies the Hue, Saturation, and Brightness settings associated with standard color selected from the Swatches tab and shown in the preview at the bottom of the dialog. Use the slider to increase or decrease the settings associated with Red, Green, and Blue colors. Or, you can individually change the color settings associated with the Hue (H), Saturation (S), and Brightness (B).

     NOTE
     When you change the color settings in the HSB tab, the color settings in the RGB tab are updated respectively.
   - **RGB**
     Specifies customization settings for a standard color chosen on the Swatches tab. Use the sliders to customize the standard color by adding more or less red, green, or blue.
5. Click OK.
   The color settings are applied to the shape or line you selected.

You can also apply line weight and patterns to components of a topology view.

Follow these steps:

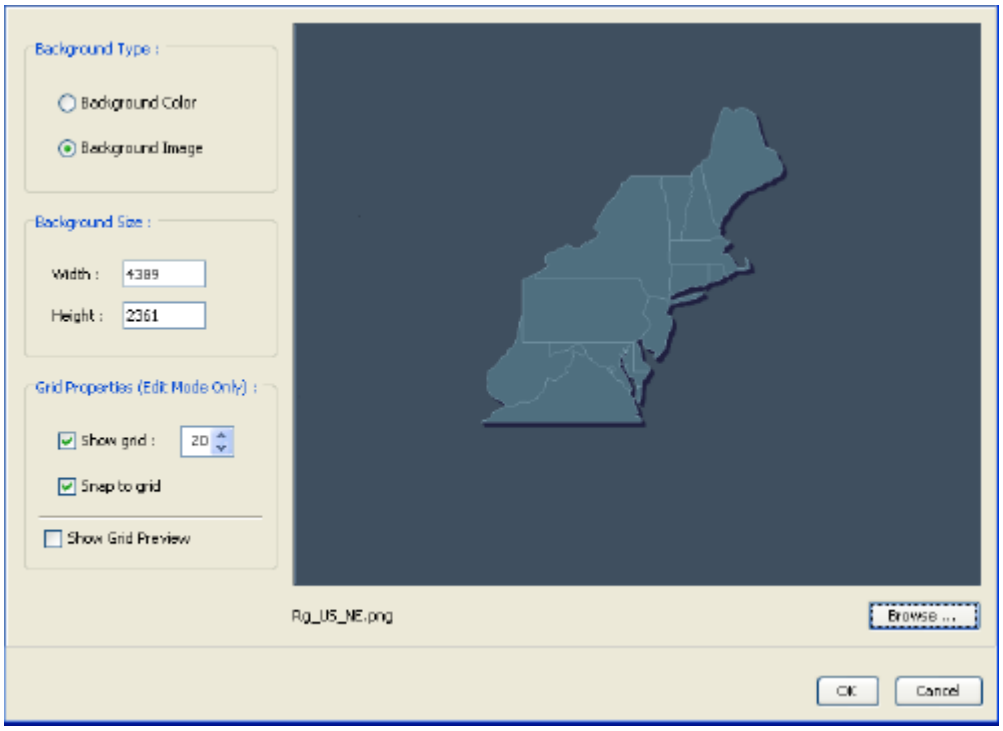
1. Switch to Edit mode as described in Access Edit Mode.
2. Select the line that you want to edit.
3. In the Edit mode toolbar, click Line Weight or Line Pattern  .

   A menu appears.
4. Click the desired line weight or line pattern.
   The selected line weight or pattern is applied to the line.

Background Editor

Use the Background Editor dialog to modify the appearance of a topology view's background. You can modify a topology view's background color, add a background image, or change the size of the background. For example, you might want to change the background size of the topology view to create additional room for modeling network entities.

The following image shows the Background Editor dialog:



Modify the Topology Background

You can use the Background Editor to modify a topology background. You can change the background color, add a background image, or change the size of the background. For example, change the background size of the topology view to create additional room for modeling network entities.

Follow these steps:

1. Switch to Edit mode as described in Access Edit Mode.
2. In the Edit mode toolbar, click the Background icon 
   The Background Editor dialog opens.
3. Specify whether to change the color or image in the Background Type section.
4. Click Browse to preview colors or images.
   The 'Select Topology Background Image' or 'Select Topology Background Color' dialog opens.
5. Select the desired image or color and click OK.
   A preview of the selected item appears in the Background Editor dialog.
6. Click OK to apply the changes to the background.
   The background view refreshes to reflect the changes.

You can use a similar procedure to change the background size.

Follow these steps:

1. Switch to Edit mode as described in Access Edit Mode.
2. In the Edit mode toolbar, click the Background icon 
   The Background Editor dialog opens.

3. Specify pixel values in the Width and Height fields.
4. Click OK.
   The background view refreshes to reflect the changes.

Group Items in a View

You can group items in any OneClick topology view. Grouping items within a view lets you edit, move, copy, paste, or delete items as one group. One of the most common group operations you may perform within a topology view is to group text (annotations) with modeled devices.

To group items in a view

1. Switch to Edit mode as described in Access Edit Mode.
2. Press and hold down the Shift key, and select the items you want to represent as a group.
3. In the Edit mode toolbar, click the Group
   icon 
   The selected items in the view are represented as a single group.

Ungroup Items in a View

Ungroup a set of grouped items within a view when you want to edit the items, or when you want to move individual items.

Follow these steps:

1. Switch to Edit mode as described in Access Edit Mode.
2. Click one of the grouped items to select the entire group.
3. Click the Ungroup
   icon 
   The items are ungrouped. You can now select individual items.

Send Items to the Back

You can send items to the back of a view so that other items within that view appear to be in front.

Follow these steps:

1. Switch to Edit mode, as described in Access Edit Mode.
2. Right-click an item, and select Send to Back.
   The item moves to the back of the view relative to other items in the view.

Bring Items to the Front

You can bring items to the front of a view relative to other items within that view.

Follow these steps:

1. Switch to Edit mode as described in Access Edit Mode.
2. Right-click the item you want to move and select Bring to Front.
   The item moves to the front of the view relative to other items in the view.

Model Attributes

Model attributes can be used to set values on models, set values directly on devices, turn DX NetOps Spectrum features on or off, configure DX NetOps Spectrum features, set default values in the DX NetOps Spectrum modeling catalog, and

so on. You can modify an attribute which is associated with a device's MIB object, thus changing the device's value for that object without having to use the device's local management. For example, you can modify a device's contact details. You can modify the Maintenance or Hibernation mode attribute to control those features.

> **WARNING**
> Use caution when changing default settings for models; this can affect the overall performance of DX NetOps Spectrum. Additionally, attribute value changes you make to the selected models will affect the same values for similar models created in the future, and for any existing model if that model's type is using the default value.

You can review and modify model attributes as follows in OneClick:

- **Information tab:** Use the Information tab in the Component Detail panel to view and modify certain common attributes for a single model. See Attributes in the Information Tab for more information about viewing and setting attribute values for a single model from the Information tab.
- **Attributes tab:** Use the Attributes tab in the Component Detail panel to access every possible attribute for a selected model. You can also create custom views of attributes and modify each one as needed, depending on your access rights. See Attributes Tab for more information about viewing and setting attribute values for a single model using the Attributes tab.
- **Attribute Editor:** Use the Attribute Editor to modify non-list attributes for a model or subset of models as well as to modify default attribute values in the DX NetOps Spectrum modeling catalog. If you change any attribute values and apply them to the DX NetOps Spectrum modeling catalog, each device model that is subsequently created based on that model type will use the new attribute value. See OneClick Attribute Editor for information about using the Attribute Editor. You must have administrative read/write privileges for those models you want to configure with the Attribute Editor.

Attributes in the Information Tab

You can view and set attribute values for individual models using the Information tab. Subviews in the Information tab display grouped categories of information available for the model. The subviews available in the Information tab depend on the selected model.

The attribute values that appear in the Information tab for the selected model are a result of a combination of the following processes:

- Automated discovery and modeling
- Manual modeling
- Using the Attribute Editor
- Direct entry using OneClick
- Default DX NetOps Spectrum values

You can set some attribute values that appear in the subviews of the Information tab. Specific attributes that you can set depend on the model selected, privileges applied to model or model types, and other factors. You can change the values of attributes for which 'set' appears next to the attribute value.

VNM Attributes in the Information Tab

In OneClick, you can view and set various attributes for each Virtual Network Machine (VNM), or SpectroSERVER, in your DX NetOps Spectrum installation. The attributes available in the VNM Information view depend on the add-on applications that are installed as part of your DX NetOps Spectrum environment. The VNM attributes are grouped into subviews for specific applications and functionality. Most of the attributes have descriptive tooltips.

General Information Subview

The General Information subview provides information about the VNM such as its network or IP address, condition, contact status, and when it was last polled successfully. With administrator privileges, you can set the VNM rollup alarm attributes. See Rollup Alarm Settings for more information. It also contains the following option:

- **Percent Models Activated**
  The percentage of models in the SpectroSERVER database that have been activated. The VNM icon will not change from its initial (blue) state until this value reaches 100%. This is useful to determine how close the SpectroSERVER is to becoming fully active after a restart. This value is also displayed in the message area of the DX NetOps Spectrum Control Panel.

DX NetOps Spectrum Modeling Information Subview

The DX NetOps Spectrum Modeling Information subview provides information about attributes such as SNMP community string, landscape, device type, and model type name.

Online Database Backup Subview

Use the settings available in this subview to configure online backups of the DX NetOps Spectrum database.

- **Automatic Backups**
  Specifies whether the DX NetOps Spectrum database is automatically backed up.
  Default: Disabled.
- **Backup Interval**
  Specifies how often, in hours and minutes, the DX NetOps Spectrum database is automatically backed up.
- **New Backup Date & Time**
  Specifies the date and time of the next database backup.
- **Backup Compression**
  Specifies whether to compress the backup file using the default compression mode.
  Default: Enabled.
- **Prefix for Backup File Name**
  Specifies the prefix used in the database backup file name. File names are appended with the date the backup occurred.
- **Backup Directory**
  Specifies the directory on the server where the backup files are written to. You must know the full path to the directory, as this is not a browse function.
- **Minimum Required Disk Space (MB)**
  Specifies the amount of free disk space that must exist on the server for a backup to start.

  NOTE
  You can initiate an online backup immediately by clicking Begin Backup Now.

SpectroSERVER Control Subview

The SpectroSERVER Control subview lets you configure various aspects of each of your local landscapes through various attributes and settings. It also contains the following views:

- Alarm Information Subview
- Event Log Information Subview
- Statistics Log Information Subview
- Thread Log Information Subview

The attributes and settings available in the SpectroSERVER Control subview include the following:

- **Device Thresholds**

Set the Device Thresholds attribute to Enabled to activate the threshold functionality on devices supporting threshold. Each threshold values setting must also be set to a non-zero value for the threshold to be active.
Default: Enabled

- **Auto Connects**
  Specifies whether DX NetOps Spectrum attempts to resolve the port connections when a pipe is created between two device models. This functionality will use the options that are enabled in the AutoDiscovery Control subview to resolve the port connections. Disabling Auto Connects can improve DX NetOps Spectrum performance if your modeled network contains management modules that support non-standard MIBs.
  Default: Enabled
- **Copy Users When Copying Group**
  If the Copy Users when Copying Group attribute is set to Yes, whenever you copy a group or a user in a group from one landscape to another, the group and all users in the group are copied as well.
  Default: Enabled
- **Log When Device Cannot Be Contacted**
  Specifies whether to continue logging attribute values (such as contact status) for models that have lost primary management contact with the devices they represent. In most cases, this is undesirable since it results in extra traffic to a part of the network where there may already be a problem. Hence, this option is disabled by default, and logging is automatically suspended for a device when contact is lost.
- **VLAN Configuration**
  Specifies whether Virtual Local Area Networks (VLANs) are modeled for networks on this VNM.
  Default: Disabled
- **Server Polling**
  Stops SpectroSERVER from polling the devices it is managing on the network. When SpectroSERVER polling is stopped, the VNM icon displays a gray condition status but no alarm will be generated. To restart SpectroSERVER polling of models, click Start.
- **Minimum Disk Space (kBytes)**
  Specifies the minimum amount of free disk space in kilobytes that must exist on the partition that the SpectroSERVER starts from for the SpectroSERVER to start. When the available space is less than this amount, a shutdown message is generated and the SpectroSERVER shuts down.
  Default: 2000
- **Use Fully Qualified Host Name**
  Specifies whether the domain name is included with the host name when the Name Service selection is placed first in the Model Naming Order list. For example, if you select Yes here, the model's icon would be created with a fully qualified name such as myhost.ca.com. If you select No here, the model's icon would be created without a fully qualified name such as myhost. This only applies when you use the device name returned from the operating system.
  Default: Yes
- **Model Naming Order**
  Specifies the order of the list of sources used by DX NetOps Spectrum to create model names for new models. If the first source at the top of the list is not available for a device, DX NetOps Spectrum attempts to use the next source in the list. The default order is as follows, with the top source being the first in priority:
  – SysName
  – IP Address
  – Name Service

  After changing the model naming order, click 'Reevaluate All Model Names' to have DX NetOps Spectrum run through all the models in the database and rename each one using the new model naming order.
  The following additional scenarios will trigger the device model name to be reevaluated using the current model name selection. It will not reevaluate based on a new model name selection:
  – If the IP address of the device changes and the model naming is based on IP Address or Name Service
  – If the Reconfiguration, Reevaluate Model Name(s) action is manually applied
  – If the Reconfiguration, Reconfigure Model action is manually applied

NOTE
If you do not want a specific device model name to be changed, set the value of the model's LOCK_MODEL_NAME (0x12a52) attribute to TRUE. This attribute locks the model name value so that it will not be changed.

- **Use Loopback**
  If Use Loopback is set to Yes, the SpectroSERVER will use the loopback interface as a primary agent address.
  Default: No
- **Loopback if Description**
  Enter a string in this field to identify a preferred loopback interface for DX NetOps Spectrum to use when modeling the device. DX NetOps Spectrum compares the string entered with the if_descr entries in the device IFTABLE for loopback interfaces only. If a match is found, DX NetOps Spectrum uses that loopback interface when it models the device. If there is no match, or no value is specified, DX NetOps Spectrum chooses the loopback interface on the device with the lowest if_index value.
- **Update Event Configuration**
  Updates the SpectroSERVER with current alert and event mappings.

Alarm Information Subview

The Alarm Information subview provides the number of each type of generated alarm.

- **Active Alarms**
  Displays currently outstanding alarms by severity.
- **Total Active Alarms**
  Displays the sum of the outstanding alarms.
- **Total Alarms**
  Displays the break-down of the different types of alarms generated since the last server restart.

  NOTE
  Blue alarms that are caused by the creation of location or organization models are never cleared.
- **Total Alarms Generated**
  Displays the total number of alarms generated since the last server restart.

Event Log Information Subview

The Event Log Information subview provides information related to the event logs. This subview contains the following settings:

- **Events Generated**
  Indicates the total number of events generated since the last server restart.
- **Locally Stored Events**
  Indicates the number of event records currently held in the database. This field will read "0" unless the Archive Manager is shut down. This will serve as backup storage area for database records until the Archive Manager is restarted.
- **Events Purged**
  Indicates the number of event records written to the archive since the last server restart.
- **Max Log Size**
  Indicates the maximum number of event records held in the database. When this number is reached, records will be deleted.

Statistics Log Information Subview

The Statistics Log Information subview provides information related to the statistics logs. This subview contains the following settings:

- **Records Generated**
  Indicates the total number of statistic records generated since the last server restart.
- **Locally Stored Records**
  Indicates the number of statistic records currently held in the database. This field will read "0" unless the Archive Manager is shut down. This will serve as a backup storage area for database records until the Archive Manager is restarted.
- **Records Purged**
  Indicates the number of statistic records written to the archive since the last server restart.
- **Max Log Size**
  Indicates the maximum number of statistic records held in the database. When this number is reached records will be deleted.

Thread Information Subview

The Thread Information subview provides information about the configuration and usage of threads. Comparing the In Use and Available columns for polling, logging, notification, and timer threads can help in determining if SpectroSERVER is running out of thread resources.

Trap Management Subview

- **Unmanaged Trap Handling**
  Specifies whether DX NetOps Spectrum processes “unmanaged” traps. Unmanaged traps are traps that come from devices which were not modeled in DX NetOps Spectrum. By default, the SpectroSERVER creates event records for any “unmanaged” traps it receives As long as this setting is enabled, SpectroSERVER processes these unmanaged traps just as it processes traps from modeled devices; that is, until a trap “storm” occurs (as defined by the Trap Storm Rate and Trap Storm Length attributes).
  The processing of unmanaged traps not only lets the network administrator know about unmodeled devices that may need to be modeled, but also allows monitoring of overall trap traffic. And it provides troubleshooting capabilities when traps are not mapped correctly. However, unmanaged trap handling can place a significant performance burden on the event logging and the Archive Manager. Depending on your priorities, you can use this setting to disable unmanaged trap handling entirely, or you can leave it enabled but limit it through the trap storm rate and length settings. Remember though that these settings also govern trap processing for modeled devices as well.

  > **NOTE**
  > Currently, only VNM models and EventAdmin models (created by users of the Southbound Gateway Toolkit) offer views that let you adjust these settings. For most device models, however, you can use the Attributes tab to create a custom view where you can adjust the default trap storm rate and length settings for that model. For more information about trap storm detection, see How Trap Storm Detection Works.

  Default: Enabled
- **Enable Trap Director**
  Lets you enable Trap Director when you want a given SpectroSERVER to forward incoming traps to models on remote landscapes in a distributed SpectroSERVER environment.
  Default: Disabled
- **Remote Forwarding Cache Age Out (minutes)**
  Lets you configure the Remote Forwarding Cache age out time (in **minutes)**
  Default: 180 minutes

How Trap Storm Detection Works

The SpectroSERVER can block the processing of traps that are coming from managed and unmanaged devices when a threshold is reached. Excessive traps that are coming at a high rate can take down your SpectroSERVER and Archive Manager. You can enable the trap storm detection at your SpectroSERVER or at the level of a modeled

device. When devices that are modeled in DX NetOps Spectrum send more than 20 traps per second, you must adjust traps_per_sec_storm_threshold so that trap storm detection does not limit the ability to receive traps.

You can enable trap storm detection at any level by configuring the following two attributes. These attributes are available under the Attributes in the Component detail pane for the selected VNM model or for a selected device model:

- **traps_per_sec_storm_threshold**
  Defines the rate at which traps are received per second from a managed or unmanaged device. When this rate is sustained for the amount of time that is specified by the TrapStormLength, the SpectroSERVER stops the processing of traps from that unmanaged or managed device.
  Default: 20 traps per second
- **TrapStormLength**
  Defines the time in seconds for which the traps_per_sec_storm_threshold value is sustained. SpectroSERVER considers it a trap storm and disables the processing of traps from that unmanaged or managed device.
  Default: 5 seconds

When traps received from any device reach the configured thresholds, the SpectroSERVER identifies this rate as a trap storm. The SpectroSERVER stops handling traps from that device and traps from other devices are not blocked. SpectroSERVER trap storm detection logic is based on each IP address of an unmanaged or a managed device (trap source) that sends traps to SpectroSERVER. As a result, you can configure each device to send traps to the SpectroSERVER at the appropriate rate.

> **NOTE**
> SpectroSERVER does not stop the processing of unmanaged traps when the overall trap storm rate from all the unmanaged devices exceeds the single trap storm threshold rate of an unmanaged device. As a result, you can configure each unmanaged device to send traps to the SpectroSERVER at the appropriate rate.

AutoDiscovery Control Subview

The attributes available in the AutoDiscovery Control subview affect actions that occur during Discovery and Modeling sessions. If you have a DSS environment, you must make any changes in these settings to all your SpectroSERVERs.

These parameters are applied when you are using the Discover LANs functionality available in a device model's Redundancy and Model Reconfiguration Options view, the Discover Connections functionality available from the right-click menu for a container model (LAN, Network, and so on), or the Auto Connects functionality used to resolve port connections when you manually draw a connection between two models. These parameters are also applied when you use the Discover Connections functionality with the Model by IP or New Model commands.

> **NOTE**
> Each of these parameters is also available when you are selecting modeling options for Discovery. Parameters set in the AutoDiscovery modeling options override the default values for that AutoDiscovery.

Modeling and Protocol Options Subview

The modeling and protocol attributes affect how DX NetOps Spectrum discovers and models elements on a network using the following functionality:

- Discovering and modeling LAN functionality available when reconfiguring a device model.
- Discovering connections functionality for a container model (LAN, Network, and others).
- Auto Connects functionality used to resolve port connections when you manually create a connection between two models.
- Discover Connections functionality when creating a new model.
- **Create WA_Link Models**
  Creates a WA_Link model between the interfaces of two routers linked by a wide area connection. This occurs during layer 3 mapping. If this option is not selected, the two linked interfaces are directly connected without the WA_Link model. See Wide Area Link Monitoring for information about Wide Area Link models and how they are used.

Default: Yes

- **Create LANs (IP subnets)**
  Specifies whether DX NetOps Spectrum uses a LAN container to represent an IP Subnet. Discovery creates the LAN container during the Layer 3 mapping process for any router interface that routes to a local LAN.
- **Create Physical Addresses**
  When this option is enabled, a physical address model is created for any MAC address that is not associated with any modeled device but was heard by a switch. The layer 2 mapper attempts to find a connection for each address found. If a connection is found, a Fanout is created and the physical address is associated to it through Connects_To. If no connection is found the model is placed in Lost and Found. This option is not recommended.
- **Create 802.3 Fanout**
  If this parameter is set to Yes and if DX NetOps Spectrum cannot make an accurate connection among three or more interfaces, a Fanout model named “802.3_Segment” will be created and these interfaces will be connected to the Fanout model. If this parameter is set to No, a Fanout model will not be created for the interfaces that have unclear connection information, and therefore these interfaces will not be mapped. However, if there is a data relay device’s interface among these interfaces, and all other interfaces are for end node devices, a Fanout model with name “Rpt_Segment” will be created.

Note: If you have 50 or more connections to a single Fanout model, consider changing this model to a Shared Media Link. The Shared Media Links must be modeled manually. These models can provide more control over fault management behavior when multiple connections are monitored. Unlike a Fanout model, Shared Media Links provide configurable thresholds for handling downstream connections that report problems. For example, a Fanout model reports a problem only when *all* downstream connections are down. However, a Shared Media Link can report the problem sooner, as when 60 percent of the downstream connections are down.

- **IP Address Tables**
  Discovery disables Layer 3 mapping and maps only the Layer 2 connections, when this option is disabled. In addition, when this option is disabled, Discovery automatically disables the IP Route Tables option, the Create WA_Link Models option, and the Create LANs (IP subnets) option.
  Default: Yes
- **IP Route Tables**
  Specifies whether DX NetOps Spectrum will use the IP Address Table to map routers. This option is set to No by default because these tables can be very large and very time-consuming for DX NetOps Spectrum to read. When this option is enabled, DX NetOps Spectrum will not be able to map unnumbered IP interfaces (0.0.0.0).
- **Source Address Tables**
  If this is set to Yes, DX NetOps Spectrum will use the device’s Source Address table when discovering connectivity information about this device.
- **Spanning Tree Tables**
  If this is set to Yes, DX NetOps Spectrum will use the device’s Spanning Tree table when discovering connectivity information about this device.
- **Discovery Protocol Tables**
  Set the Discovery Protocol Tables attribute to Yes to allow DX NetOps Spectrum to map device connectivity using discovery protocol MIB information. Currently, the following discovery protocols are supported:
  – Nortel Discovery Protocol
  – Cisco Discovery Protocol
  – Extreme Discovery Protocol
  – Cabletron Discovery Protocol
  – Alcatel Discovery Protocol
  – Foundry Discovery Protocol
  – Link Layer Discovery Protocol
- **Traffic Resolution**

If the Traffic Resolution parameter is set to Yes, DX NetOps Spectrum will use network traffic data (ifInOctet and ifOutOctet statistics) to determine connections between interfaces, and in many cases eliminate the need for a Fanout model.

- **ARP Tables**
  When enabled, DX NetOps Spectrum uses the ARP table to determine pingable MAC addresses for the connectivity mapping.
- **ATM Protocols**
  If the ATM Protocols parameter is set to Yes, the ATM Discovery runs against all ATM switches in the SpectroSERVER database.
  Default: No
- **SNMP Community Strings**
  Create, order, and delete community strings and profiles for SNMP v1, v2c, and v3, which are used, in order, when DX NetOps Spectrum attempts to access and model devices that were discovered using SNMP and for which no device community string was provided.
- **SNMP Ports**
  The SNMP Ports section lets you create, order, and delete the list of ports to use when accessing and modeling devices. To add port numbers to this list, click Add under the SNMP Ports field, enter the port number and click OK.
- **IP Exclusion List**
  A list of IP addresses or IP address ranges that will be ignored and which will not be modeled when devices are discovered.

Trap Based Continuous Discovery Subview

Use the Trap Based Continuous Discovery subview to configure DX NetOps Spectrum to automatically create a device model when it receives an SNMP or syslog trap from a device not already modeled. When the SpectroSERVER receives an unmanaged trap, it asserts an event on the VNM model indicating that an unmanaged trap was received.

All models created using Trap Based Continuous Discovery are placed in the New Device Container. DX NetOps Spectrum places new models created by a scheduled continuous discovery or by an unmanaged trap into this container.

- **Unmanaged Trap Discovery**
  Set the Unmanaged Trap Discovery attribute to Yes to discover and model the source of an unmanaged trap using the IP address sent with the trap. This includes both SNMP and syslog traps from devices as well as Agent Log file matching traps. See SpectroSERVER Control Subview for information about how to enable unmanaged trap handling on the VNM.
- **New Devices In Maintenance**
  Set the New Devices In Maintenance attribute to Yes to have new device models created based on an unmanaged trap put into maintenance mode when they are discovered.
- **Create Pingables**
  Set the Create Pingables attribute to Yes to have DX NetOps Spectrum model devices that cannot be modeled using SNMP as type ‘Pingable,’ if the devices respond to a ping (ICMP) echo request.
- **Discover Connections**
  If the Discover Connections attribute is set to Yes, DX NetOps Spectrum attempts to discover and model the connections for devices discovered by Trap-based Continuous Discovery.

Debug Options Subview

The Debug Options subview lets you turn on the AutoDiscovery debugging functionality using the following settings:

- **Debug AutoDiscovery**
  Set the Debug AutoDiscovery attribute to ‘On’ to have DX NetOps Spectrum create a debug output file containing data on the status of the device modeling and mapping process for each Discovery session. These files are available at *<$SPECROOT>/SS/ADiscDebug_<timestamp>*. The Debug AutoDiscovery option is useful when the discovery

modeling or mapping process is hanging or when there are connectivity mapping issues. In these cases, the output file indicates where and on which devices any difficulties were encountered.

NOTE
In addition, you can debug a particular device's connectivity mapping process. To do this, set the Debug AutoDiscovery option to On.

When Discovery's modeling process is running, DX NetOps Spectrum prints out all connection information. This information includes the data collected from bridge tables, Proprietary Discovery Protocols, Spanning Tree tables, potential connections, errors encountered, and any additional, pertinent information related to the mapped devices.

- **Abort Discovery**
  The Abort Discovery button lets you stop and cancel a currently running AutoDiscovery.

Fault Isolation Subview

The Fault Isolation subview lets you configure various aspects of the DX NetOps Spectrum fault isolation functionality. For more information about this view, see Fault Isolation Settings.

Live Pipes Subview

Live Pipes functionality lets you enable port status monitoring for individual links and view link status. In DX NetOps Spectrum, a *link* is a connection between two devices that DX NetOps Spectrum has resolved to the port level. For more information about Live Pipes and network fault management, see Live Pipes and Fault Management.

The Live Pipes attribute must be set to 'Enabled' on the VNM to enable Live Pipes functionality on the VNM.

If you have administrator privileges, you can set other attributes in this view: Alarm Linked Ports, Suppress Linked Port Alarms, and Port Always Down Alarm Suppression.

Live Pipes and Global Collections in DSS Environments

In a DSS environment, the Live Pipes attribute must be set to the same value on all VNMs so that the Live Pipes functionality provides accurate link connection information in Global Collections. If Live Pipes is set to different values on VNMs in a DSS setup, the Live Pipes information in Global Collections will be unpredictable.

Alarm Management Subview

The Alarm Management subview lets you control some aspects of alarm management.

The AlarmMgmt model, which governs the Alarm Management subview, is a SpectroSERVER application. The AlarmMgmt model inherits the security string of the VNM model only if you have not independently changed the security string of the AlarmMgmt model.

For example, the security strings for the VNM and AlarmMgmt models are initially empty. You change the AlarmMgmt model security string to "Jack" and later, you change the security string for the VNM model to "Jill." The AlarmMgmt mode security string is not changed to "Jill."

AlarmMgmt model attributes are not distributed. Bring up the Alarm Management subview for each SpectroSERVER whose alarm management attribute values you want to change. Changing an attribute on one SpectroSERVER does not apply to any other SpectroSERVER.

WARNING
Displaying Initial and Suppressed alarms is not recommended in OneClick. These alarms can generate a significant volume of network traffic. DX NetOps Spectrum generates initial and suppressed alarms if the Disable Initial Alarms and Disable Suppressed Alarms settings for the Virtual Network Machine (VNM) managing your network are set to Yes.

The Alarm Management subview contains the following attribute settings:

- **Generate Alarm Events**
  Enables the generation of alarm change events; DX NetOps Spectrum creates events (viewable in the Events tab) for alarm changes based on alarm creation, updating, and clearing events.

  NOTE
  If the Generate Alarm Events option is disabled, you do not see Alarm History in the Alarms view.

  Default: Yes
- **Add Events to Alarms**
  Controls whether alarm change events are added to each alarm. If disabled, alarm change events are not displayed under the Events tab of the Alarm view.
  Default: No
- **Age Out Residual Alarms Only**
  Specifies whether only residual old alarms are cleared. Residual alarms are alarms that existed before SpectroSERVER restart and have not been reverified. If enabled, DX NetOps Spectrum clears only residual alarms that are based on the Alarm Age Out timer setting.
  Default: Yes
- **Alarm Age-Out Time (hours)**
  Defines how long an alarm can exist in DX NetOps Spectrum. Once an alarm has existed for the number of hours that you specify by this attribute, it is a candidate for automatic removal. To disable this functionality, set this attribute to zero (0).
  Every hour, DX NetOps Spectrum checks the status of all alarms in the landscape and uses this option to determine whether alarms are cleared. Therefore, an alarm is not removed at the precise moment when its existence time has exceeded the time-out. An alarm can be, at most, an hour “overdue.”
  You cannot age out non-user clearable alarms. The system functionality that generates such alarms, clears them. Maintenance Mode, Device Has Stopped Responding, and Management Agent Lost are the non-user clearable alarms.

  NOTE
  An aged out alarm which is cleared displays the "System.Alarm_AgeOut" value in its corresponding "Cleared By" column under "Cleared Alarms History" tab. The corresponding cleared event also displays this value in its "Cleared By" column under Events tab.
- **Disable Initial Alarms**
  Specifies whether to generate an alarm when the condition of a model changes to Initial. Because these changes in the condition of a model can cause a flood of alarms, disabling this option can improve system performance.
  Default: Yes

  NOTE
  If Initial, Suppressed, or the Maintenance alarms are disabled and later enabled, these alarms are not displayed in the Alarm view for existing models. Only the alarms that are generated after this option is enabled appear in the view.
- **Disable Suppressed Alarms**
  Specifies whether to generate an alarm when the condition of a model changes to Suppressed. Because these changes in the condition of a model can cause a flood of alarms, disabling this option can improve system performance.
  Default: Yes
- **Disable Maintenance Alarms**
  Specifies whether to generate an alarm when the condition of a model changes to Maintenance. Because these changes in the condition of a model can cause a flood of alarms, disabling this option can improve system performance.

  NOTE
  For more information about putting devices in Maintenance mode, see the Using OneClick section.

Default: No

- **Auto UnAcknowledge On New Occurrence**
  Specifies whether to unacknowledge the new occurrence of an alarm.
  Default: No

BGP Manager Subview

The BGP Manager subview lets you globally configure BGP peer session monitoring.

The BGP Manager subview contains the following attribute settings:

- **BGP Peer Session Monitoring**
  Monitors the status of the peer session on the BGP port at the polling interval of the port model's Polling_Interval Attribute value if this setting is enabled and the live pipe on the BGP peer session port is turned on. If you disable this option, an event of type 0x220018 is generated on the BGP downed port models to clear the BGP alarm.
  Default: Disabled
- **BGP Peer Session Discovery Interval (minutes)**
  Indicates the interval for BGP peer session Discovery. If BGP Peer Session Monitoring is enabled, BGP Peer Session Discovery initially runs on each BGP device at SpectroSERVER startup and when a new BGP device is modeled. After, BGP Peer Session Discovery runs according to the interval you set.
  Default: 24 hours

Network Configuration Manager Subview

The Network Configuration Manager subview provides information about Network Configuration Manager.

This subview contains the following setting:

- **Export Directory**
  Specifies the local directory to which you want to export configuration text files. If you want to export configuration text files to a network share, you must specify the UNC path to the directory. For example, \\Shared_Server\Export \ExportFiles.

TFTP Configuration Subview

The TFTP Configuration subview provides information about the Trivial FIle Transfer Protocol (TFTP). TFTP transfers configuration files.

This subview contains the following settings:

- **Default TFTP Directory**
  Specifies the TFTP server path.
- **TFTP Transfer Timeout (sec)**
  Specifies the maximum time (in seconds) for a data transfer to complete.
  Default: 50 seconds, which means the data must be completely transferred within 50 seconds.

  NOTE
  For more information about the TFTP server, see the Network Configuration Managersection.

FTP Configuration Subview

The FTP Configuration subview provides information about the File Transfer Protocol (FTP).

This subview contains the following settings:

- **FTP Username**

Specifies the FTP server username.

- **FTP Password**
  Specifies the FTP server password.
- **Default FTP Directory**
  Specifies the FTP server path.

> **NOTE**
> For more information about the FTP server, see the Network Configuration Manager section.

Thresholds And Watches Subview

You can create, configure, and administer watches in OneClick. View and configure watches from a table in the Thresholds And Watches subview.

> **NOTE**
> You can access the Thresholds and Watches subview from the Information tab for a model.

The Watches table displays information for each watch defined on that model. The Watch Status column displays the watch condition with color codes as follows:

- **Gray**
  Indicates that the watch is inactive. The watch is not currently running because it has not been activated.
- **Blue**
  Indicates the initial state of the watch. The watch is activated but has yet to run for the first time.
- **Green**
  Indicates that the watch is active and running without any violation.
- **Yellow**
  Indicates that the watch threshold is violated.
- **Red**
  Indicates that the watch failed to evaluate. The text explains the reason.

The toolbar buttons let you do the following:

- Activate
- Deactivate
- Create
- Edit
- Copy
- Delete
- Display watch information
- Print watch information
- Export the Watches table

Host Security Information Subview

When a client application connects to a SpectroSERVER, DX NetOps Spectrum reads the .hostrc file to obtain a list of valid hosts. If a host name from the .hostrc file does not resolve to a network address, you will receive a “Permission Denied” error message. In addition, an event and an alarm (Event00010e01, Prob00010e01) will be generated on the VNM indicating that there are unresolved host names.

To help you find the cause of this problem, the Host Security Information subview displays a list of resolved and unresolved host names.

Modeling Gateway Subview

You can view information about recent imports in a table in the Modeling Gateway subview.

The Modeling Gateway table displays information about recent imports. The number of import files listed is controlled by the Max Records field. The default value for the Max Records field is 30.

> **NOTE**
> For more information about the Modeling Gateway table, see the Modeling Gateway Toolkitsection.

IP Services Subview

The IP Service subview provides information about VPN Manager and VPLS Manager. Further options are available depending upon the products you have installed.

Logical Connection Import Subview

The Logical Connection Import subview lets you create logical connections between virtual link models by importing a comma-delimited, ASCII file (text file or XML file) that defines the connections. You can define connections that include two ATM models or an ATM model and a Frame Relay model. Click the Import button to import a file.

> **NOTE**
> For more information about logical connections between virtual link models, see the ATM Circuit Manager section.

Shared IP Detection and Alarming

The following settings control when DX NetOps Spectrum generates alarms for shared IP addresses.

- **Shared IP Alarming**
  Specifies whether shared IP alarming is enabled.
  Default: Disabled

  > **NOTE**
  > Shared IP alarms will be cleared when you set the Shared IP Alarming attribute to Disabled.
- **Currently Shared IP Addresses**
  Specifies which IP addresses are currently considered “shared” in DX NetOps Spectrum.

  > **NOTE**
  > The IP addresses in the loopback subnet are displayed as shared addresses in the ‘Currently Shared IP Addresses’ list, however, no alarms will be triggered based on these addresses to help prevent multiple unnecessary alarms for a known and desired configuration.
- **Allowed Shared IP Addresses**
  Specifies which IP addresses can be shared in DX NetOps Spectrum. Click Add or Remove to modify this list as needed.

  > **NOTE**
  > You can modify a device model's NETWORK_ADDRESS (0x12d7f) attribute and PrimaryAddress (0x12d80) attribute if the device model's IP address is included in the Allowed Shared IP Addresses list on the VNM model.

Shared IP Alarms and Events

When DX NetOps Spectrum detects that two or more devices share one or more IP addresses, and you have configured DX NetOps Spectrum to generate alarms in this case, you will see an orange alarm on all device models that share the IP address or addresses. The event generated on each device will contain a list of all device models involved as well as a list of all shared IPs. The event will look similar to the following:

```
Device {X} of type {Y} has the following shared IP addresses:
<list of shared IPs and devices>
```

Since the detection of shared IP addresses is dependent on DX NetOps Spectrum device models, each time a new device model is created or destroyed, a new event containing updated data may need to be generated on the devices that have shared IPs.

No Unique IP Alarms and Events

If a device is found to contain no unique IP addresses, then a red alarm is asserted on it to notify you of this condition because no reliable communication or management may be made with that device. The event will look similar to the following:

```
Device {X} of type {Y} has no globally unique IP addresses.  Each of the following addresses is shared with
 another device:
<list of shared IPs and devices>
```

Network Address Is Shared

If you manually create a device using a shared IP address as the Network_Address you receive an event such as the following:

```
Device {X} of type {Y} has its Network Address set to an IP that is currently shared by multiple devices.  No
 reliable communication or management may be made with this device.  The shared IP {shared IP} is shared by
 the following other devices:
<list of other devices>
```

Troubleshooting Tip:

In case you cannot update network_address on switch model due to "The update to attribute Network_Address was disallowed" error, see the KB article.

Configuring Allowed/Non-Alarming Shared IP Addresses

You can configure DX NetOps Spectrum with a list of IP addresses, IP address ranges, or subnets for which sharing between multiple devices is allowed. Populate the Allowed Shared IP Addresses list with the addresses to share. The IP addresses on this list do not generate alarms.

You can use OneClick to add or remove IP addresses, IP address ranges, or subnets from this list. Shared alarms are cleared when you add an IP address to the list. Adding an IP address causes the associated device to have no more shared IP addresses that generate alarms.

Mtype Based IfType Filtering Subview

From 10.3, you can filter interfaces of a device model which you do not want to model and monitor in DX NetOps Spectrum. In OneClick, The 'Mtype Based IfType Filtering Subview' allows you to filter interfaces of a device model.

You can access the 'Mtype Based IfType Filtering Subview' in the Information tab for a selected VNM model in the OneClick.

Mtype Iftype Map

The Mtype Iftype Map settings allow you to map the device model of the interfaces, which you want to filter and do not want to model.

- **Mtype Value**Specify the model type (MType) of the device model.For example: Rtr_Cisco Mtype value is 0x21000c
- **IfType Value**
  Specify the interfaces type (Iftype) values, which you do not want to model. The values must be comma (,) separated for multiple Iftype values.
  For example: The Iftype values for Ethernet interface (6) and Prop Virtual Interface (53) values are mentioned as 6,53 (refer to the screenshot)

  NOTE
  If these settings are done after the discovery of a device model, you must reconfigure the models to apply these settings.

CreateWALinkForPropVirtualInterface Attribute

The following attribute has been added to the VNM model type:

```
CreateWALinkForPropVirtualInterface
```

Type: Boolean

Default: False

Attribute ID: 0x1321b

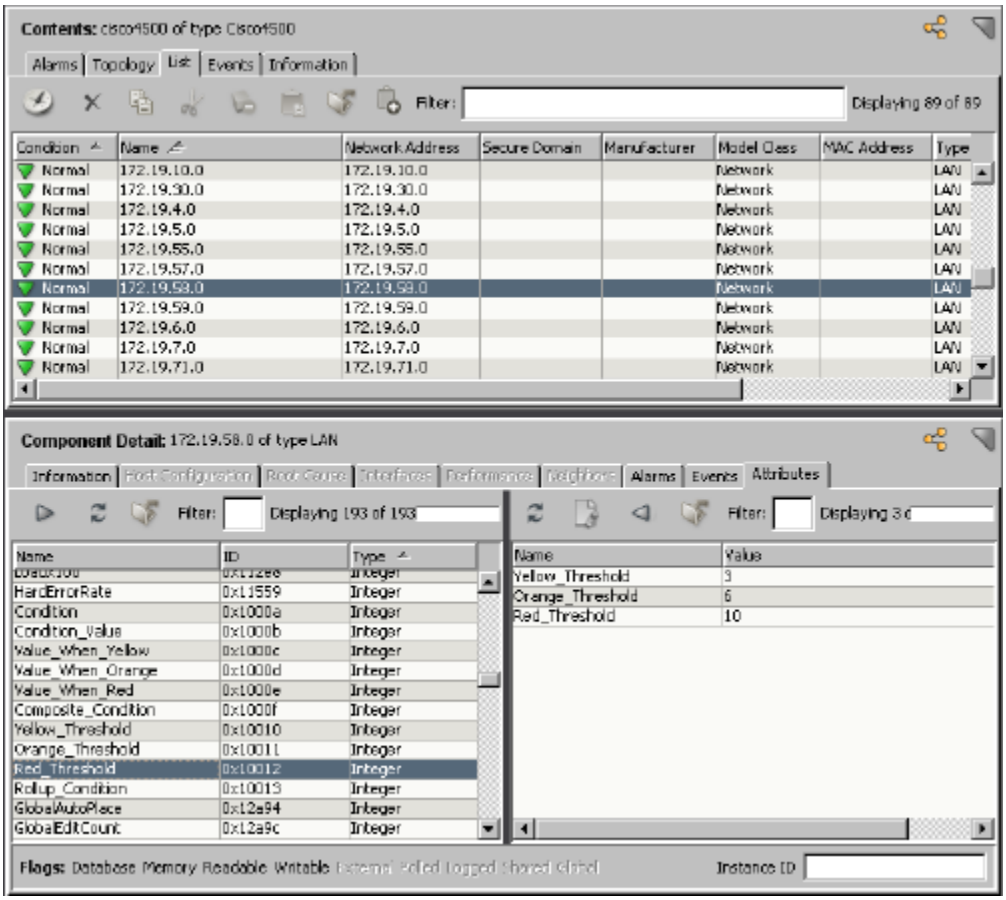
You do not have a separate view for this attribute. Therefore, to view this attribute, navigate to the Component Details window of the VNM model type and click the Attributes tab.

You can set the attribute value to True to create a WA_Link connection between proprietary virtual interfaces. Previously discovered devices and connections are not affected by changing the value of this attribute. To view the changes, run the discovery again.

Attributes Tab

The Attributes tab in the Component Detail panel provides access to all of a selected model's attributes. From the Attributes tab you can select one or more attributes related to the specific model or model type and review details, poll values, export values, and edit each attribute as needed, depending on your access rights. You can also use the Attributes tab to cycle through models, quickly checking the same attribute for each one, to review attribute flags by scanning the Flags information located at the bottom of the Attributes view, or to review values for list attributes individually or all at once, depending on your preference.

The following graphic shows an example of the Attributes tab. The attributes being shown in the Attributes tab belong to the model selected in the List tab:



Access Attributes from the Attributes Tab

You can access attributes from the Attributes tab and personalize your view of them as required.

Follow these steps:

1. Select the model whose attributes you want to view or edit.
2. Click the Attributes tab in the Component Detail panel.
   A list of attributes appears in the left half of a split panel.
3. (Optional) Enter text in the Filter field at the top of the list to filter it.
4. Double-click each attribute that you want to display in a view.
   Each attribute that you double-click appears in the right-side of the panel along with its value.
5. (Optional) Click a column header to sort the attributes as needed.
6. (Optional) Select an attribute in the right-side of the panel and click the left arrow button at the top of the panel to move the selected attribute back to the left-side of the panel when you no longer want to review it.

Edit Attributes in the Attributes Tab

You can edit attributes for a single model from the Attributes tab.

Follow these steps:

1. Select the model whose attributes you want to modify.
2. Click the Attributes tab in the Component Detail panel.
   The model's available attributes appear in the left side of the panel.

3. Double-click each attribute that you want to edit.
   Each selected attribute and its value appear in the right side of the panel.
4. Double-click an attribute in the right side of the panel.
   If the selected attribute can be modified, the Edit dialog opens.
5. Clear the 'No Change' check box in the Edit dialog to enable editing.
6. Modify the attribute as needed, and click OK.
   The Attribute Edit Results dialog opens, indicating whether the attribute edit was successful.

   NOTE
   Click Undo in the Attribute Edit Results dialog to revert to the original attribute settings.
7. Click Close.
   The attributes have been edited and the Attribute Edit Results dialog closes.
8. (Optional) Click Export to send the selected attributes and their values to a CSV file, a text file, or web page.

Edit Multiple Attributes at Once in the Attributes Tab

You can edit multiple attributes simultaneously in the Attributes tab.

Follow these steps:

1. Select the model whose attribute values you want to modify.
2. Click the Attributes tab in the Component Detail panel.
   The available attributes for this model appear in the left side of the panel.
3. Double-click each attribute that you want to edit.
   Each selected attribute and its value appear in the right side of the panel.
4. Click the Edit button in the toolbar of the right panel.
   The Edit dialog lists the attribute values that are available for editing from your selected list.
5. Clear the 'No Change' check box in the Edit dialog for each attribute, modify each attribute as required, and click OK.
   The Attribute Edit Results dialog opens, indicating whether each edit operation was successful.

   NOTE
   Click Undo in the Attribute Edit Results dialog to revert to the original attribute settings.
6. Click Close.
   The attributes have been modified. The Attribute Edit Results dialog closes.
7. (Optional) Click Export to send the selected attributes and their values to a CSV file, a text file, or web page.

Examine the Same Attribute on Multiple Models

From the Attributes tab, while you are in either the List tab or the Topology tab, you can select an attribute or multiple attributes and quickly view the attribute value on a number of models.

Use the List tab to examine the values of the same attributes for multiple models.

Follow these steps:

1. Select the first model for which you want to modify attribute values from the List tab in the Contents panel.
2. Click the Attributes tab in the Component Detail panel.
   The model's available attributes appear in the left side of the panel.
3. Double-click each attribute whose values you want to review.
   Each selected attribute and its value appear in the right-side of the panel.
4. Press the down or up arrow on your keyboard to review the same attributes for a different model.
   The Attributes view refreshes to display the values for the same attributes on the selected model.

   NOTE
   If you move to a different container, these attributes remain selected. The attributes stay in the Attributes tab until you exit OneClick.

View List Attribute Values

You can use the Attributes tab to review the values of a model's list attributes.

To view the value for a particular instance of a list attribute

1. Select the model for which you want to view attribute values.
2. Click the Attributes tab in the Component Detail panel.
   The model's available attributes appear in the left side of the panel.
3. Double-click the list attribute for which you want to review values.
   The list attribute and the value of the first instance in the list appear in the right-side of the panel.
4. Do *one* of the following to review the list attribute's values:
   – In the Instance ID field at the bottom of the Attributes tab, type the OID of the particular value you want to view, and press Enter.

     NOTE
     The Instance ID applies to the list attributes in the right-side panel. If you place a new list attribute in the right-side panel, the value displayed in the Value column corresponds to the OID specified in the Instance ID field.

   – Click the table link in the Value column to open a dialog which displays the instances and values for the list attribute.

     NOTE
     You can perform a number of actions from this table including the following: refresh values, print values, and export values.

Update Attribute Values

The values of selected attributes are not dynamically updated. They reflect the value returned as of the SpectroSERVER poll prior to their selection.

To update attribute values, click Refresh in the toolbar in the right panel of the Attributes tab.

The values of all the attributes you originally selected from the left panel refresh to display any new values.

OneClick Attribute Editor

The Attribute Editor is an advanced DX NetOps Spectrum utility used to configure management 'policies' that govern how DX NetOps Spectrum manages network devices and their components. It is best suited for performing bulk attribute changes on multiple devices models.

You can change attribute values for one or more selected models in a view. The Attribute Editor dialog groups attributes into categories. You can edit the default settings provided within these categories, or you can define additional attributes to edit within the User Defined category.

Open Attribute Editor

You can open the Attribute Editor in OneClick by right-clicking any model and selecting Tools, Utilities, Attribute Editor.

You can also launch the Attribute Editor from anywhere in OneClick where you can select a model, including the List tab, the Explorer tab, Interfaces tab in Component Details panel, the Locater Results tab, or from the Tools menu.

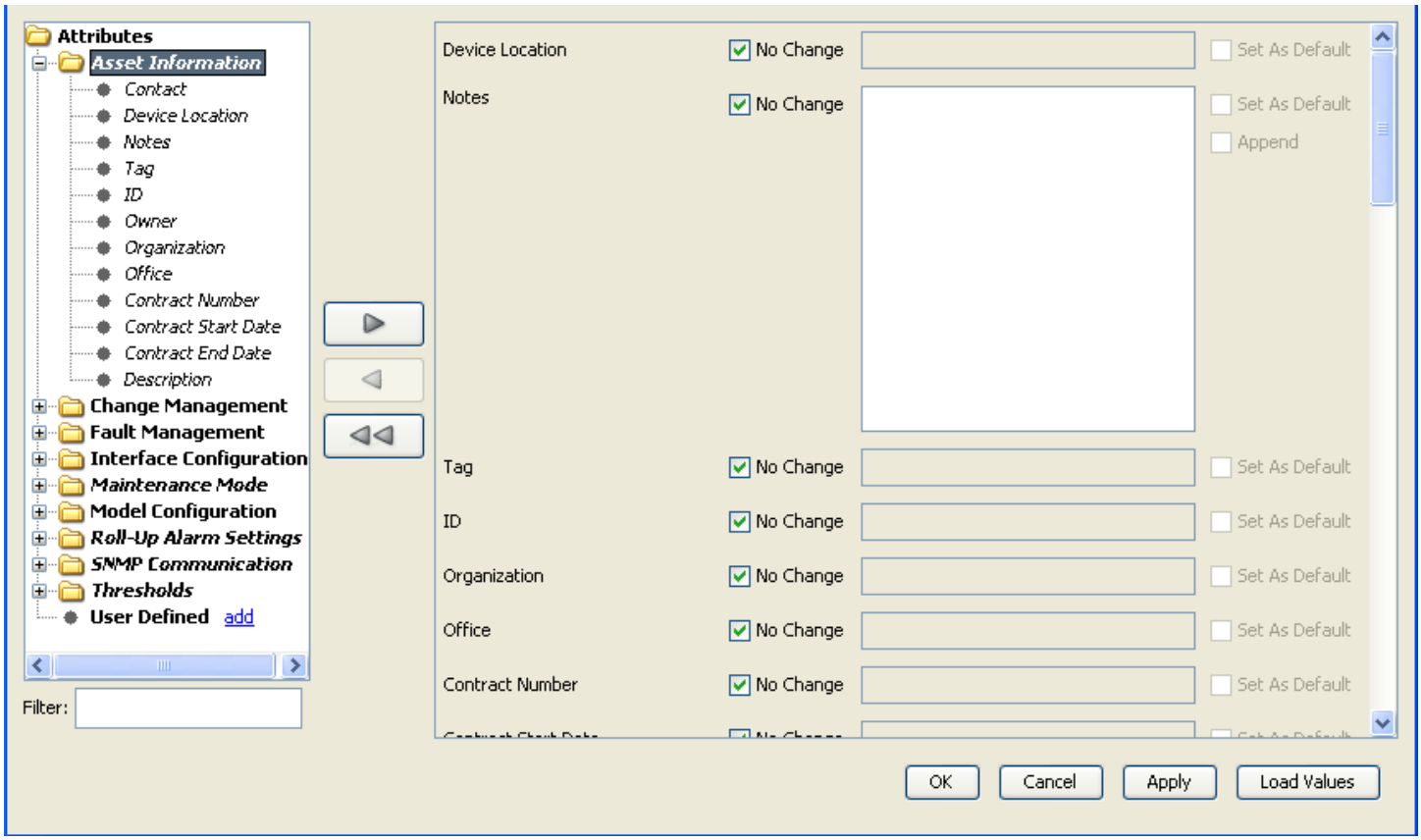
Open Attribute Editor with Device Context

You can open the Attribute Editor within the context of selected models. Select a model or multiple models in the List tab, the Explorer tab, or the Interfaces tab. Then right-click and select Utilities, Attribute Editor. The Attribute Editor opens with the context of the selected models.

Any changes that you make apply to the selected models. If you select Set as Default, the changes are applied to the DX NetOps Spectrum modeling catalog for the selected models.

Attribute Editor Dialog

The Attribute Editor includes a right and left panel. The left panel groups attributes in a tree display. The right panel provides an editing area to view current attribute values and make changes.



Task-Oriented Attribute Groupings

The left panel of the Attribute Editor provides attributes grouped by tasks you perform on devices and by categories of attribute types. The SNMP Communications folder groups attributes related to tuning SNMP communications between the SpectroSERVER and a device.

Filter Attribute Categories

You can type text in the Filter text box to locate attributes in the attribute categories in the Attribute Editor left panel. When you type text in the Filter text box, attribute categories that do not contain an attribute that matches the filter appear as bullets, becoming inaccessible. All attribute categories that contain an attribute that matches the filter appear as folders that you can expand and you can select the subcategories within to move into the right panel for editing.

For example, if you want to find attributes related to alarms, type **alarm** in the Filter text box.

Attribute Edit Panel

To edit attribute values, select the attribute category from the left panel, then click the right facing arrow to place the attributes in the Editor panel. An attribute that has been placed in the Editor panel appears in *italics* in the left panel.

Note: Tooltips are available for some attributes when they appear in the right panel for editing.

The Attribute Editor provides the following options:

- **No Change:** The No Change setting appears to the left of most attribute input fields. When No Change is selected, the input value, if any, is not written when you click Apply or OK. When you make changes to the attribute value by selecting a value or by clicking in the input field, No Change is automatically cleared. Clear No Change or click in the input field to makes the attribute value editable.
- **Set as Default:** This option appears to the right of most attribute input fields in the Editor panel. If you select Set as Default, the value is written to the model types in the DX NetOps Spectrum modeling catalog when you click Apply or OK. All future models that use these model types inherit the new value.

  WARNING
  The changes are made to the modeling catalog. As a result, existing models that use the current default value(s), and any new models that are created in the future inherit these new values. The type of device that they represent is irrelevant. Changing the default value affects existing models that you did not explicitly select, but these changes might not take effect until after a server restart. Existing models that use different value(s) are not changed.

  A model is an instance of a model type. The model type has default values for every attribute. When the model is created, every attribute that is not explicitly set inherits the default that is set on the model type.
  Once you create an instance of that model type, the new model can have its own values for every attribute. By default, it does not have a value for the ifModelNameOption. When you edit the model to change the ifModelNameOption, the model has its own value for that attribute and no longer uses the default that is set in the model type. From that point on, the new model only uses its own value and does not use the default setting of the model type.
  You can also edit the model and enable the option that sets this new value as the default. The new default then affects all of the following:
  – This new model.
  – The model type (which now has a new default for that attribute).
  – All existing models of that model type that lack their own default value. They now use the new default because they still point to the model type value.
  – All new models of that model type. These models also use that model type default value.

  However, models of that model type that existed before the attribute change and had their own value set for this attribute are not changed. They still use their own custom setting.
  For example, assume that you use the Attribute Editor to change Model A to use ifAlias (11f7e). You then change Model B to use ifAlias and enable the Set As Default option. All models that use that model type will then use ifAlias. If you then change Model B and the default value to use ifDesc (1134b), all models *except for* Model A will use that new value. Model A does not use it because it already has its own value for that attribute, set to ifAlias. Models C, D, and E also had their own values set when you changed Model B and designated ifAlias as the default. Therefore, Models C, D, and E are similarly unchanged.
- **Load Attribute Values:** You view the current value for a set of selected attributes when you have launched the Attribute Editor in the context of a specific model. After populating the edit panel, click Load Attribute Values to view the current values for the attributes. If the selected models do not use or have a value for an attribute, no value displays when you click Load Attribute Values.

If you have launched Attribute Editor in the context of multiple model types, they can have different values for the same attribute. If this occurs, the Select Model dialog opens when you click Load Attribute Values. Select the model for which to load attribute values, and click OK.

When you click Apply or OK in the Attribute Editor, DX NetOps Spectrum attempts to write the new attribute values and displays the Attribute Edit Results dialog.

Attribute Edit Results Dialog

The results of attribute value changes appear in the Attribute Edit Results dialog. Each item in the table represents the result of a single attribute written to a model. The Result column indicates whether the write operation succeeded or failed. The Old Value and New Value columns show the original value and the last written value. If the write operation failed or the previous value could not be obtained, the corresponding field will display N/A. If the write operation failed, for example, if the device did not respond, you can select the item in the table and click Retry.

Click the Undo button to undo the selected successful attribute value change in results list if necessary.

User-Defined Attributes

In the Attribute Editor dialog, you can create a list of attributes that display when you expand the User Defined category. After you create this list, you can access the user-defined attributes. You can at any time remove the user defined attributes.

Create User-Defined Attributes

Each OneClick user can create a unique set of user-defined attributes. You can select user-defined attributes using the Attribute Selector dialog.

Follow these steps:

1. In the left panel of the Attribute Editor dialog, next to the User Defined folder, click Add.
   The Attribute Selector dialog opens.
2. If you have more than one model type selected, select the model type whose attributes you want to edit from the left pane of the Attribute Selector dialog.
   The attributes for the selected model type appear in the right pane of the Attribute Selector dialog.
3. Select the attribute to edit from the list, and click OK.

   NOTE
   Use the Filter text box to quickly locate an attribute or model type in the list.

   The attribute that you selected appears in the User Defined category in the Attribute Editor dialog. You can only add attribute at a time to the User Defined category.
4. Repeat this process to select additional user-defined attributes.

   NOTE
   Remove user-defined attributes by clicking the remove link next to the attribute that you want to remove.

Change Attributes in Conjunction with Search

You can use the Attribute Editor feature in conjunction with the Search feature in the Locater tab (Locater Search). By using the Locater Search feature with the Attribute Editor feature you can locate all models meeting certain criteria and attempt to change the attribute values on those matching models.

The following example combines ‘creating and running a new search’ with ‘changing attributes through the Attribute Editor.’ It walks you through adding user-defined attributes and writing changes to the component (that is, SpectroSERVER, devices, interfaces) meeting the search criteria.

NOTE
For more information about using the Search feature in the Locater tab, see the Using OneClick section.

Example Define a Search to Create an Attribute for Editing

This example demonstrates how to create and run a search to locate the GlobalConfig model on the SpectroSERVER. It then shows how to use the Attribute Editor to add the HibernationCommSuccessTries attribute to the User Defined category so that you can update the value as needed.

NOTE
The value for HibernationCommSuccessTries determines the number of successful attempts the SpectroSERVER must make to devices in hibernation mode before the devices can resume normal management communication. By default, the value of this attribute is 3.

To define a search to create a user-defined attribute for edit

1. In the OneClick Locater tab, to create a new search,
   click 
   The Create Search dialog opens.
2. Select 'Model Type Name (0x10000)' from the Attribute drop-down list.
3. Select 'Equal To' from the Comparison Type drop-down list.
4. Type **GlobalConfig** in the Attribute Value field.
5. Click Save As, type a name for the search (for example, 'Hibernation attempts'), and click OK.
6. Click OK in the Create Search dialog.
7. Select the search you just created ('Hibernation attempts') in the Locater tab.
8. To launch the selected search,
   click 
   The Select Landscape to Search dialog opens.
9. Select the landscapes to search and click OK.
   The search results appear in the Results tab.
10. Right-click the GlobalConfig entry and select Utilities, Attribute Editor.
    The Attribute Editor dialog opens.
11. Click the add link in the left panel of the Attribute Editor dialog, next to the User Defined folder.
    The Attribute Selector dialog opens.
12. Click the folder named Other in the left panel of the Attribute Selector dialog.
13. In the Filter text box (below the left panel) type **GlobalConfig** and select the GlobalConfig entry under the 'Other' folder.
14. In the Filter text box under the right panel of the Attribute Selector dialog, type **HibernationCommSuccessTries**.
    The HibernationCommSuccessTries attribute appears in the Attribute for GlobalConfig list in right panel.
15. Double-click the HibernationCommSuccessTries entry in the list to add it to the User Defined category.
16. In the Attribute Editor dialog, edit the user-defined attribute value by selecting it in the left panel and clicking the right arrow button to move its associated attribute fields to the editing panel.
17. In the right panel, edit the attribute values as desired then click Apply to write the changes to the component.
    The Attribute Edit Results dialog opens listing the results of the changes made.

Edit Attributes for Specific Devices or for Model Types

This section provides examples for changing an attribute value for a specific model type, or for a specific set of devices.

Example 1: Edit Interface_Polling_Interval for Cisco Devices Supporting IPsec

Cisco IPSec tunnel interface management is available in DX NetOps Spectrum for Cisco devices that support the IPSec related MIBs. Once modeled, the tunnel models are updated every hour. If your environment requires less or more frequent updates to the tunnel models, use the Attribute Editor to change the polling interval.

The attribute Interface_Polling_Interval defines how frequently DX NetOps Spectrum monitors the MIB associated with the tunnel interface models so that the modeling is up to date. To disable this monitoring, set the Interface_Polling_Interval attribute to 0. To change the frequency at which DX NetOps Spectrum monitors these MIBs, change the value for the attribute to the desired number of seconds between polling cycles.

Edit the Interface_Polling_Interval attribute for Cisco devices.

Follow these steps:

1. Locate the Cisco routers on your network by creating a new Global Collection of Cisco routers that have the Interface_Polling_Interval set to 3600 seconds.
2. Select the Cisco routers whoseInterface_Polling_Int erval value you want to edit from the Global Collection.
3. Right-click and select Utilities, Attribute Editor.
4. Add the Interface_Polling_Interval to the User Defined attributes list, as described in Create User-Defined Attributes.
5. Move the Interface_Polling_Interval attribute into the right panel for editing.
6. Enter the value in seconds of the polling interval.
7. Click OK.
   The Attribute Edit Results dialog displays the results for each device whose attribute value you attempted to change.
8. If the change failed on any of the selected devices, select them and click Retry.
9. Close the Attribute Edit Results dialog, and click OK to close the Attribute Editor.

Example 2: Edit the DeviceTypeDiscEnable Attribute for Specific Devices

The DeviceTypeDiscEnable attribute is used to allow or prevent changes to the device type name value for device models and model types. You can modify the value for this attribute.

To edit DeviceTypeDiscEnable for specific models, you first must add the DeviceTypeDiscEnable attribute to the User Defined category. Once you have done that you can begin to select the devices on which you want to prevent device type name customizations, as described in the following procedure.

Edit DeviceTypeDiscEnable for specific device models.

Follow these steps:

1. Add the DeviceTypeDiscEnable attribute to the User Defined category using the procedure described in using the procedure described in Create User-Defined Attributes.
2. Select the Locater tab in the Navigation panel.
3. Expand the Devices folder and double-click By Model Name.
   The Search dialog opens.
4. Enter the name of the device type model on which you want to prevent device type name customizations; select all applicable landscapes as necessary and click OK.
   The devices using the device type model specified appear in the Lists tab.
5. Select the specific devices on which you want to prevent customizations.
6. Right-click and select Utilities, Attribute Editor to launch the Attribute Editor.
7. Expand the User Defined folder, select DeviceTypeDiscEnable, and click the right arrow to place the attribute in the right-side editing panel.
8. Set the attribute value to No and click Apply.
9. Verify that ‘Set as Default’ is not selected.
   The Attribute Edit Results dialog opens and displays the results of the edit, either successful or unsuccessful. If successful, the DeviceTypeDiscEnable attribute is now set to false, or no, on the devices you selected in Step 5.

Example 3: Edit DeviceTypeDiscEnable for a Model Type

You can set an attribute value and apply it to the model type in the DX NetOps Spectrum modeling catalog and to all device models. Also, all device models created in the future using the model type will use the attribute value that you set in this manner.

Edit DeviceTypeDiscEnable for a model type.

Follow these steps:

1. Add the DeviceTypeDiscEnable attribute to the User Defined category using the procedure described in Create User-Defined Attributes.
2. Create a search using the Locater tab in the Explorer that finds some or all the device models that use the new custom model type.
3. Select one of the device models using the new custom model type in the search results list, right-click it and select Utilities, Attribute Editor.
   The Attribute Editor opens with the context of the select device.
4. In the Attribute Editor, select the DeviceTypeDiscEnable attribute in the User Defined category and move it to the Attribute Editing panel by clicking the right arrow.
5. Set the attribute value to No.
6. Select Set as Default to apply this change to the DX NetOps Spectrum catalog for the model type used by the device model selected in Step 3.
7. Click Apply.
   A warning message appears.
8. Click Yes.
   This action sets the attribute value for all the device models using the model type, and applies it to the model type in the DX NetOps Spectrum modeling catalog and to all device models. All future device models created using this model type will have the DeviceTypeDiscEnable attribute set to No, and the Device Type Attribute cannot be overwritten.

Model Type Reevaluation

In cases where devices have been replaced on your network, IP addresses may be assigned to new devices without your knowledge. Therefore, device models periodically verify that they are modeled using the correct model type. If the model type no longer matches the device identify, an alarm is generated on the model.

By default, this reevaluation of model types occurs every 24 hours. You can change this setting for all models.

Edit the Model Type Reevaluation Interval

You can modify the interval for model type reevaluation. The MTypeVerifyInterval attribute on the VNM model determines the reevaluation interval, which is every 24 hours by default. Set this value to 0 to disable model type reevaluation.

Follow these steps:

1. Select the VNM model and click the Attributes tab in the Component Detail panel.
   The available attributes for the VNM model appear on the left side of the panel.
2. Locate the MTypeVerifyInterval attribute by typing it in the Filter field.
   The MTypeVerifyInterval attribute appears on the left side of the panel.
3. Double-click the MTypeVerifyInterval attribute.
   The MTypeVerifyInterval attribute and its value appear on the right side of the panel.
4. Double-click the MTypeVerifyInterval attribute on the right side of the panel.
   The Edit dialog opens.
5. Clear the ‘No Change’ check box in the Edit dialog and type **0** in the field.
   The attribute value is changed to 0.
   The Attribute Edit Results dialog indicates whether the modification succeeded.
6. Click Close.
   The Attribute Edit Results dialog closes; model type reevaluation has been disabled for all models.

Change Management Attributes

This group of attributes lets DX NetOps Spectrum maintain up-to-date configuration information about modeled devices. The following attributes in this grouping let DX NetOps Spectrum interrogate a device and gather information about its interfaces and connections after a specific event occurs:

- Automatically Reconfigure Interfaces
- Discovery After Reconfigure
- Discover Connections after Link Up Events
- Topologically Relocate Model

See Update Device Interface and Connection Information for more information about these attributes, and about configuring DX NetOps Spectrum to maintain updated device configurations.

The following attributes in this grouping configure DX NetOps Spectrum to have redundant ways of contacting devices:

- Enable Redundancy
- Generate Redundancy Alarms

DX NetOps Spectrum will use multiple IP addresses in a cascading manner to contact devices if a device fails to respond to queries made on its primary address. The devices must have multiple IP addresses configured in their IP tables.

Interface Configuration Attributes

Interface Configuration Attributes

You set the value for a set of model type interface configuration attributes in the Interface Configuration grouping. Some of the attributes available include the following:

- **Admin Status**
  This attribute sets the administrative status for an interface.
- **Create Sub-Interfaces**
  Set this attribute to Yes, and if a device supports RFC1573, DX NetOps Spectrum will model the device's sub-interfaces.
- **Interface Name Primary Suffix**
  DX NetOps Spectrum uses the value of this required attribute to name interfaces for the model or models for the Rename Interface Models action. Choose from a set of available suffixes in the drop-down list when editing this attribute.
- **Interface Name Secondary Suffix**
  DX NetOps Spectrum uses the value of this optional secondary attribute to name interfaces for the model or models for the Rename Interface Models action. It is prefixed by an underscore (_) and follows the value of the primary suffix. This secondary attribute is optional.

Stale Interfaces

DX NetOps Spectrum handles interface definitions that are temporarily removed from their corresponding MIB tables using its Stale Interface functionality. These types of situations can include:

- A module is temporarily removed from a device
- A configured tunnel temporarily goes down on a device

In these situations, it is advantageous for DX NetOps Spectrum to retain the interface modeling information rather than immediately destroying them. This prevents useful model-specific attributes and resolved connections from being lost.

DX NetOps Spectrum determines that an interface model is stale when no corresponding entry exists in the MIB where the interface is defined. Once the stale interface 'ages out' DX NetOps Spectrum removes it from the model. The age out period for an interface is defined by the Stale Interface Age Out attribute (in minutes) on the device model.

An event is generated when DX NetOps Spectrum determines an interface is stale. If the stale interface has resolved connections, a minor alarm is generated on the interface model. If DX NetOps Spectrum determines that the interface is no longer stale before the age out period expires, and prior to a reconfiguration which causes the interface model to be destroyed, an event is generated and any stale interface alarm is cleared.

- **Enable Stale Interface Alarms**
  Set this attribute to Yes to enable DX NetOps Spectrum to generate a minor alarm on an interface model when the interface becomes stale and it has resolved connections. Set this attribute to No if you do not want this condition to generate an alarm.
- **Stale Interface Age Out (min)**
  This attribute specifies the amount of time in minutes that DX NetOps Spectrum waits for the stale interface to 'age out' before removing the model. To disable the stale interface functionality, set this attribute to 0.
  Default: 120

Maintenance Mode Attributes

You can set values for the Maintenance and Hibernation mode attributes in the Attribute Editor's Maintenance Mode folder. You can also set these attribute values and create and apply maintenance mode schedules in the model's Information tab.

> **NOTE**
> For more information about Maintenance and Hibernation modes and managing maintenance mode schedules, see the Using OneClick section.

Rollup Alarm Attributes

Access the DX NetOps Spectrum attributes used to manage rollup alarm settings (conditions) and threshold levels in the Attribute Editor's Roll-Up Alarm Settings folder. You can also view and set these attributes in the Information tab under the General Information subview. You can adjust these attributes for containers modeled on your network and for the DX NetOps Spectrum container model library. The following section lists the attributes, describes how they are used, and defines their default values.

Note: Change threshold levels carefully; you may see an increase in generated alarms if threshold levels are set lower, or a decrease in generated alarms if levels are set higher.

- **Value When Yellow**
  The point value of a Yellow alarm condition existing in a child towards the roll up alarm threshold value for the parent container.
  Default: 1
- **Value When Orange**
  The point weight of an Orange alarm condition existing in a child towards the roll up alarm threshold value for the parent container.
  Default: 3
- **Value When Red**
  The point weight of a Red alarm condition existing in a child towards the roll up alarm threshold value for the parent container.
  Default: 7
- **Yellow Threshold**
  The minimum points needed to trigger a Yellow roll up alarm for a container.

Default: 3

- **Orange Threshold**
  The minimum points needed to trigger an Orange roll up alarm for a container.
  Default: 6
- **Red Threshold**
  The minimum points needed to trigger a Red roll up alarm for a container.
  Default: 10

Model Status and Alarm Conditions

OneClick uses rollup alarm thresholds and model alarm thresholds to determine the status for modeled entities. OneClick displays two types of status for modeled entities, Condition and Rollup Condition. The following section lists the details about what these conditions describe, and what OneClick applies them to.

- **Condition**
  Applies to all models. Reflects the current contact or alarm status of the model itself.
- **Rollup Condition**
  Applies to container models, such as networks, LANs, and WANs. Reflects the composite status of all the other models in the container, which are sometimes referred to as their children.

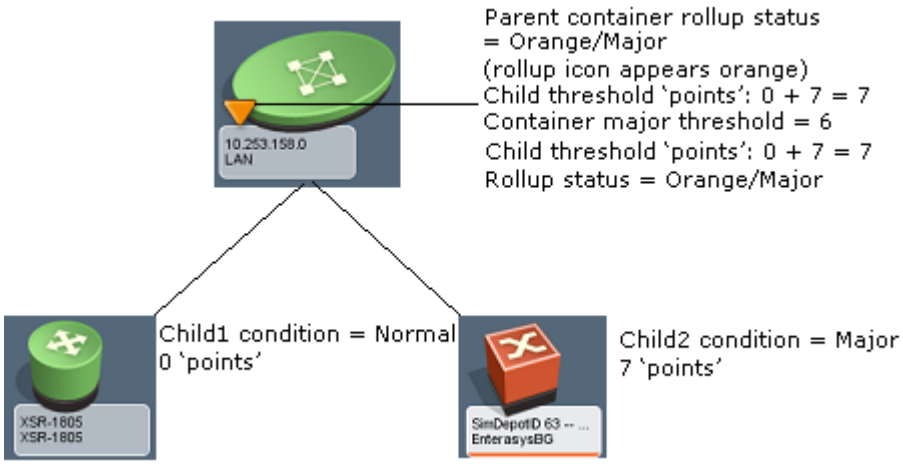
If a modeled device or interface exists in a container, its condition rolls up to the parent container and is reflected in the container's rollup condition. The model status types listed previously rely on threshold values to determine when and how to use the associated color indicators. The rollup condition is displayed using an inverted triangle that appears adjacent to a container icon in the container's Information tab.

Rollup Condition Thresholds

A container model has attributes that define values for alarm conditions that may exist on the children of the container. A container model also has attributes that define when its rollup alarm conditions are triggered. The combined value of all the alarm conditions for a container's children is used to determine the rollup alarm condition for the container.

The following illustration shows a container that has a rollup condition of orange, or major, based on the alarm conditions of its two children. The rollup alarm setting for the container uses the default values listed previously.

- One child has a green or normal condition; this contributes zero points toward the container's rollup condition.
- The other child has a red or critical condition that contributes seven points toward the container's rollup condition (Value When Red = 7).
- The total value of the alarm conditions on the container's children is 7.
- The rollup thresholds for the container use the default values for the Rollup alarm settings listed in this article. The Orange Threshold value = 6, so the container's rollup appears as Orange, indicating a major alarm condition.



NOTE
Configure the Rollup Alarm settings and Fault Management settings of the container model according to your requirement. Otherwise, child alarms do not roll up the container model in the Explorer view.

SNMP Communication Attributes

You can tune overall SNMP communications by changing the values of the attributes in the Attribute Editor's SNMP Communication folder. The following attributes define how DX NetOps Spectrum communicates with a device:

- **SNMP Community String**
  Lets the SpectroSERVER communicate with devices on your network.
- **DCM Timeout (ms)**
  The number of milliseconds the polling agent will wait for a response from the device before timing out.
- **DCM Retry Count**
  Specifies the number of times the SpectroSERVER retries to establish device communication after the DCM timeout value expires.
- **Polling Interval (sec)**
  The number of seconds between polls DX NetOps Spectrum makes to devices.

  NOTE
  Increasing this number results in less SNMP-related traffic on your network and a smaller load on the SpectroSERVER. Decreasing this number for mission critical devices and interfaces lets you see updated information about these devices in OneClick more often. This can improve your ability to see potential issues on the network before they affect network performance. A decreased Polling Interval will result in more SNMP network traffic generated by DX NetOps Spectrum.
- **Poll To Log Ratio**
  The number of polls per log. If it is set to 3, then data is logged every third poll.

Threshold Attributes

The Thresholds grouping contains the DX NetOps Spectrum device and interface threshold settings.

NOTE
For more information about interface threshold parameters, see the Using OneClick section.

Calculating CPU and Memory Utilization

The MIB objects that are read to calculate the source of the CPU and memory utilization can be customized for an individual device model or set as the default for a given model type. This is only necessary if DX NetOps Spectrum is unable to identify a source for CPU and memory utilization, or if you prefer to use a different source.

Do one of the following to customize CPU and memory utilization:

- Modify the order and the types of sources to be tested.
- Modify the source attributes used by the attribute redirection type.

Attribute redirection is the process of using well-known attributes as a pointer, or redirectors, to proprietary attributes. For example, the well-known attribute NRM_CPUUtilAttr (0x12e2d) on the AirespaceSw model type holds the attribute ID of the Airespace proprietary CPU attribute, agentCurrentCPUUtilization (0x4b605ae).

With attribute redirection, it is possible for generic code to reference unique attributes for each device model or model type. Additionally, attribute redirection lets you change the source attributes without having to restart the SpectroSERVER. By default, for most model types, attribute redirection is the first source type that is tested. Therefore, in most circumstances, you only have to modify the attributes that attribute redirection is using.

NOTE
Only certain attribute types can be used for attribute redirection.

Before you make changes to the normalized CPU and memory intelligence, understand how DX NetOps Spectrum calculates CPU and memory utilization.

DX NetOps Spectrum does the following to calculate CPU and memory utilization:

1. DX NetOps Spectrum identifies the source that is used to calculate the CPU and memory utilization. DX NetOps Spectrum identifies the source any time the device is reconfigured.
   A list of the possible sources is provided in a new preference attribute. Each source is tested in order. If a valid source is found, the source intelligence, the attributes that are used, and the attribute's model type are stored for reuse during the utilization calculation. This helps ensure that the source does not change until the source is reconfigured. If the list of sources is empty, or a valid, functioning source is not identified, the source type is set to “None” and no further reads are made to the device.
   In general, the sources are tested in the following order:
   – Attribute redirection
   – CA proprietary intelligence
   – Standard intelligence (RFC 2790 and Net-SNMP)
2. DX NetOps Spectrum performs the actual calculation of the utilization using the correct source, attribute IDs, and model type handles that were identified. The running attribute IDs list and the running model handle list are passed into the calculation method each time, to help ensure that the same attributes are read.

Normalized CPU Utilization Calculation Requirements

You can calculate the normalized CPU utilization for any device using attribute redirection as the source for the utilization, however, the device must meet the following requirements:

- The device must have a single MIB object which has a data type of either integer, 64-bit integer, text string, float, or real. The MIB object must be a scalar object or a list object.
- If the MIB object has a data type of text string, the values of the MIB object are valid if the text string represents a valid number. For example, 9.4, 43, and 1200, are considered valid text strings. A text string that contains numbers with extra text is not valid. For example, 43% is considered invalid.
- If the MIB object is a list object, each instance in the list must report a valid CPU value (no filtering of lists is provided). The reported value must be an instantaneous usage (or an aggregate of a short time period).
- The MIB object must report the utilization of all CPUs in the device in units of 0-100 percent.

  NOTE
  DX NetOps Spectrum does not attempt to adjust invalid results. For example, if an attribute returns 110 percent utilization, DX NetOps Spectrum reports the 110 percent utilization. Verify that the values that are reported in the attribute always yield the correct utilization, once calculated. If negative values are returned, thresholds can indicate a violation, even though a threshold was not exceeded.

Normalized Memory Utilization Calculation Requirements

You can calculate the normalized memory utilization for any device, using attribute redirection as the source for the utilization, however, the device must meet the following requirements.

The device must have one of the following:

- A single MIB object which is of the data type integer, 64-bit integer, text string, float, or real, and is either a list or a scalar. In addition, this object must report the utilization of all memory for the device in units of 0-100 percent.

  NOTE
  DX NetOps Spectrum does not attempt to adjust invalid results. For example, if an attribute returns 110 percent utilization, DX NetOps Spectrum reports the 110 percent utilization. Verify that the values that are reported in the attribute always yield the correct utilization, once calculated. If negative values are returned, thresholds can indicate a violation, even though a threshold was not exceeded.

- If the MIB object has a data type of text string, the values of the MIB object are valid if the text string represents a valid number. For example, 9.4, 43, and 1200, are considered valid text strings. A text string that contains numbers with extra text is not valid. For example, 9.4 MB, 43 MB, and 1,200 are considered invalid.
- Two or more MIB objects which are all either scalars or lists. Additionally, two of the objects must report either the total amount of free memory, used memory, or total memory. The units in which each MIB object reports its respective value *must* be the same.

  NOTE
  DX NetOps Spectrum does not attempt to verify that the units in which each MIB object reports its respective value are the same.

The values reported by the MIB objects must be instantaneous (or an aggregate of a short time period).

Normalized CPU Utilization Attributes

DX NetOps Spectrum uses the following attributes to calculate the normalized CPU utilization for a device:

- **NRM_CPUIntelPref**
  Lists possible sources to test when identifying normalized CPU utilization. These sources are tested in the order in which they appear in this attribute.
- **NRM_DeviceCPUUtilization**
  Reports the device's CPU utilization. The normalized CPU utilization calculation is triggered based on what this attribute reports.
- **NRM_DeviceCPUUtilizationNames**

Contains the names of each instance of the CPU utilization value. By default, the instance is displayed as:

```
CPU: <instance>
```

– ***<instance>***
Is the instance ID of each CPU utilization value.

NOTE
If the NRM_CPUUtilNameAttr attribute is available, the NRM_CPUUtilNameAttr attribute setting is used to populate the NRM_DeviceCPUUtilizationNames attribute. The names of each instance of the CPU utilization value is set once. If the NRM_CPUUtilizationNameAttr attribute setting changes, you must reconfigure the model to pick up the name change.

- **NRM_CPUAttr_Source**
Contains the source that is currently being used to calculate the normalized CPU utilization. If the intelligence ID is attribute redirection, this attribute says 'Attribute Redirection'. If the intelligence ID is CA - Proprietary, this attribute lists the MIB that DX NetOps Spectrum reads the values from.

The following attributes are used to utilize attribute redirection as the source for calculating the normalized memory utilization for a device:

- **NRM_CPUUtilAttr**
(Required) Points to the attribute that reports CPU utilization in percent. You populate this attribute with the attribute ID of the attribute which reports CPU utilization for the device. The attribute can be either a list or a scalar, and must be one of the following data types: counter, gauge, int, real, or 64-bit long.

  NOTE
  DX NetOps Spectrum does not attempt to adjust invalid results. For example, if an attribute returns 110 percent utilization, DX NetOps Spectrum reports the 110 percent utilization. Verify that the values that are reported in the attribute always yield the correct utilization, once calculated. If negative values are returned, thresholds can indicate a violation, even though a threshold was not exceeded.

- **NRM_CPUUtilNameAttr**
(Optional) Points to the attribute that reports the identifying information for the CPUs. This attribute holds the attribute ID of the attribute which reports the names associated with each instance of CPU utilization for this device.
This attribute can be a scalar or a list, but it must be the same data type as the NRM_CPUUtilAttr attribute and the names must be ordered such that the first element in the list matches the first element in the utilization list. If you do not enter an attribute ID, or if the attribute ID you provide is not valid, the instance ID is appended to “CPU:<*name*>” to create the name for that CPU value. Any data type is accepted.

  NOTE
  OctetStrings are treated as printable text strings.

- **NRM_CPUModelTypeToRead**
(Optional) Lists the model type handle of an application model that the NRM_CPUUtilAttr and the NRM_PUUUtilNameAttr attributes are read from.

Normalized Memory Utilization Attributes

DX NetOps Spectrum uses the following attributes to calculate the normalized memory utilization for a device:

- **NRM_MemoryIntelPref**
Lists possible sources to test when identifying normalized memory utilization. These sources are tested in the order in which they appear in this attribute.
- **NRM_DeviceMemoryUtilization**
Reports the device’s memory utilization. The normalized memory utilization calculation is triggered based on what this attribute reports.
- **NRM_DeviceMemoryUtilizationNames**
Contains the names of each instance of the memory utilization value. By default, the default is displayed as:

`Memory: <instance>`

– ***<instance>***
Is the instance ID of each memory utilization value.

NOTE
If the NRM_MemoryUtilNameAttr attribute is available, the NRM_MemoryUtilNameAttr attribute setting is used to populate the NRM_DeviceMemoryUtilizationNames attribute. The names of each instance of the memory utilization value is set once. If the NRM_MemoryUtilizationNameAttr attribute setting changes, you must reconfigure the model to pick up the name change.

- **NRM_MemAttr_Source**
Contains the source that is currently being used to calculate the normalized memory utilization. If the intelligence ID is attribute redirection, this attribute says 'Attribute Redirection'. If the intelligence ID is CA - Proprietary, this attribute lists the MIB that DX NetOps Spectrum reads the values from.

The following attributes are used to utilize attribute redirection as the source for calculating the normalized memory utilization for a device:

- **NRM_MemoryUtilAttr**
Points to the attribute that reports the memory utilization in percent.
- **NRM_MemoryUsedAttr**
Points to the attribute that reports the used memory in units, for example, bytes, kilobytes, megabytes, gigabytes, and so on.
- **NRM_MemoryTotalAttr**
Points to the attribute that reports the total memory in units, for example, bytes, kilobytes, megabytes, gigabytes, and so on.
- **NRM_MemoryFreeAttr**
Points to the attribute that reports the free memory in units, for example, bytes, kilobytes, megabytes, gigabytes, and so on.

NOTE
To calculate the normalized memory utilization for a device using attribute redirection as the source for the utilization, either NRM_MemorUtilAttr must be populated, or two of the following three attributes must be populated: NRM_MemoryUsedAttr, NRM_MemoryTotalAttr, NRM_MemoryFreeAttr.

- **NRM_MemoryUtilNameAttr**
(Optional) Points to the attribute that provides the memory utilization names.
- **NRM_MemoryModelTypeToRead**
(Optional) Lists the model type handle of an application model that the NRM_MemoryUtilAttr, NRM_MemoryUsedAttr, NRM_MemoryTotalAttr, NRM_MemoryFreeAttr, and the NRM_MemoryUtilNameAttr attributes should be read from.

Calculate Normalized CPU Utilization

You can calculate the normalized CPU utilization for a device whose utilization is not calculated out-of-the-box. You can also recalculate the utilization using different attributes than those attributes that are used by default. You can use different attributes on a per model or a per model type basis. To use different attributes on a per model type basis, use the Attribute Editor or the Model Type Editor to change the default attribute values for the model types.

NOTE
For more information about the Model Type Editor, see the Model Type Editor section.

To determine if the normalized CPU utilization is already calculated for a device, view the Source column in the Thresholds and Watches, Thresholds subview in the Information tab of the Component Details panel.

NOTE
DX NetOps Spectrum does not attempt to adjust invalid results. For example, if an attribute returns 110 percent utilization, DX NetOps Spectrum reports the 110 percent utilization. Verify that the values that are reported in the attribute always yield the correct utilization, once calculated. If negative values are returned, thresholds can indicate a violation, even though a threshold was not exceeded.

Follow these steps:

1. Verify that the device meets the specified requirements.
2. Identify the attribute that reports CPU utilization and place this attribute ID into NRM_DeviceCPUUtilAttr.
3. Identify the attribute that reports the identifying information for all the device CPUs, if the attribute exists.
4. Place the attribute ID into NRM_CPUUtilNameAttr.
5. Identify the origin of the attribute that reports CPU utilization and the attribute that reports the identifying information for the CPUs. If these attributes originate on an application model, enter the model type handle of the application model into NRM_DeviceCPUModelTypeToReadAttr. Otherwise, leave this attribute empty.
   If the DeviceCPUModelTypeToReadAttr is empty, DX NetOps Spectrum attempts to read the specified attribute from the device model. If DeviceCPUModelTypeToReadAttr is populated, DX NetOps Spectrum attempts to find an associated application model with that model type handle. If the associated application model, with the specified model type handle, is not found, or the attributes do not exist on that model, attribute redirection is not considered to be a valid source for calculating the utilization.
6. Reconfigure the device model.
   If attribute redirection fails, DX NetOps Spectrum attempts to test other available sources. If a valid, functioning source is not identified, the source column in the Thresholds and Watches, Thresholds subview displays 'None'. The normalized CPU and normalized memory performance graphs report 'Not available', and no further reads are made to the device. If a valid, functioning source is identified, the source column displays the successful source.

Calculate Normalized Memory Utilization

You can calculate the normalized memory utilization for a device whose utilization is not calculated by default, or for a device whose utilization has not been automatically calculated. You can also recalculate memory utilization using attributes that are not used by default. You can use different attributes for individual models or for selected model types. To use different attributes on a per model type basis, use the Attribute Editor or the Model Type Editor to change the default attribute values for the model types.

Note: For more information about the Model Type Editor, see the Model Type Editor section.

To determine whether the normalized memory utilization is already calculated for a device, view the Source column in the Thresholds and Watches, Thresholds subview in the Information tab of the Component Details panel.

NOTE
DX NetOps Spectrum does not attempt to adjust invalid results. For example, if an attribute returns 110 percent utilization, DX NetOps Spectrum reports the 110 percent utilization. Verify that the values that are reported in the attribute always yield the correct utilization, once calculated. If negative values are returned, thresholds can indicate a violation, even though a threshold was not exceeded.

Follow these steps:

1. Verify that the device meets the specified requirements.
2. Identify the attribute that reports memory utilization, if it exists, and place this attribute's handle into NRM_DeviceMemoryUtilizationAttr. If the device does not support an attribute that reports memory utilization, identify two of the following three attributes:
   - Used Memory
   - Free Memory
   - Total Memory

Populate the related attribute, NRM_Memory*XXX*Attr, where *XXX* is Free, Used, or Total. If more than the required attributes are provided, all the attributes must exist on the same model type.

NOTE
These attributes must report memory utilization in the same unit, for example, bytes, kilobytes, megabytes, gigabytes. However, DX NetOps Spectrum does not verify that each attribute reports the same units.

If more than the required number of attributes are provided, DX NetOps Spectrum uses the following order of precedence to determine which attributes to use:

a. NRM_MemoryUtiliAttr
b. NRM_MemoryUsedAttr and NRM_MemoryTotalAttr
c. NRM_MemoryFreeAttr and NRM_MemoryTotalAttr
d. NRM_MemoryFreeAttr and NRM_MemoryUsedAttr

The first instance that returns a valid set of values is used.

3. Identify where these attributes originate. If these attributes originate on an application model, enter the model type handle of the application model into NRM_DeviceMemoryUtilizationNameAttr. Otherwise, leave this attribute empty. If the NRM_DeviceMemoryUtilizationNameAttr is left empty, DX NetOps Spectrum attempts to read the specified attribute from the device model. If NRM_DeviceMemoryUtilizationNameAttr is populated, DX NetOps Spectrum attempts to find an associated application model with that model type handle. If the associated application model, with the specified model type handle, is not found, or the attributes do not exist on that model, attribute redirection is not considered to be a valid source for calculating the utilization.
4. Reconfigure the device model.
   If attribute redirection fails, DX NetOps Spectrum attempts to test other available sources. If a valid, functioning source is not identified, the source column in the Thresholds and Watches, Thresholds subview displays 'None', the performance graphs report 'Not available', and no further reads are made to the device. If a valid, functioning source is identified, the source column displays the successful source.

Troubleshoot CPU and Memory Utilization Calculation

If DX NetOps Spectrum is returning incorrect or invalid CPU and memory utilization calculation values, reconfigure the model.

The attributes that DX NetOps Spectrum reads, and the model handles that DX NetOps Spectrum reads from, are all cached to verify that the same source is used during the utilization calculation. If a source is no longer available, or has been changed, invalid or incorrect values are reported until the device model is reconfigured.

NOTE
Name attributes are only reevaluated when the model is reconfigured and then cached. Therefore, if instances change, or the names associated with a given instance change, reconfigure the model.

If DX NetOps Spectrum is not selecting the source that you want to use, do the following:

1. Verify that the order of possible sources you want to test is set correctly in the CPU intelligence preference attribute, NRM_CPUIntelPref, and memory intelligence preference attribute, NRM_MemoryIntelPref.
2. Verify that the device supports the source you want to use. For example, if you want to use attribute redirection, confirm that the attributes are supported on the device model or specified application model.

Fault Management

This section describes how you can manage the faults in your network.

Configure Cross-Landscape Fault Correlation

In a distributed SpectroSERVER (DSS) environment, a network administrator may need to model a router from a local landscape in a remote landscape for its connections to participate in fault isolation for the remote landscape. This *proxy* model doesn't need to participate in alarm generation in the remote landscape because it already does so in the local landscape where it is modeled "normally." It is in the local landscape that alarms for the router are tracked and trouble tickets are created.

In such a scenario, Cross-Landscape Fault Correlation can prevent multiple red alarms for the same outage. With the Enable Event Creation attribute set to FALSE in the proxy model's Fault Management subview, DX NetOps Spectrum suspends the creation of events for the model (and any component models such as boards or ports). This effectively disables alarms for the model, but unlike maintenance mode, SNMP communication with the proxy model continues, and so too does its participation in fault isolation.

> **NOTE**
> For more information about distributed network management, see the Distributed SpectroSERVER Administration section.

Designate a Model as a Proxy Model

You can designate a device model as a proxy model from the Information tab of the selected device model. Setting a device model up as a proxy model disables event creation for that model.

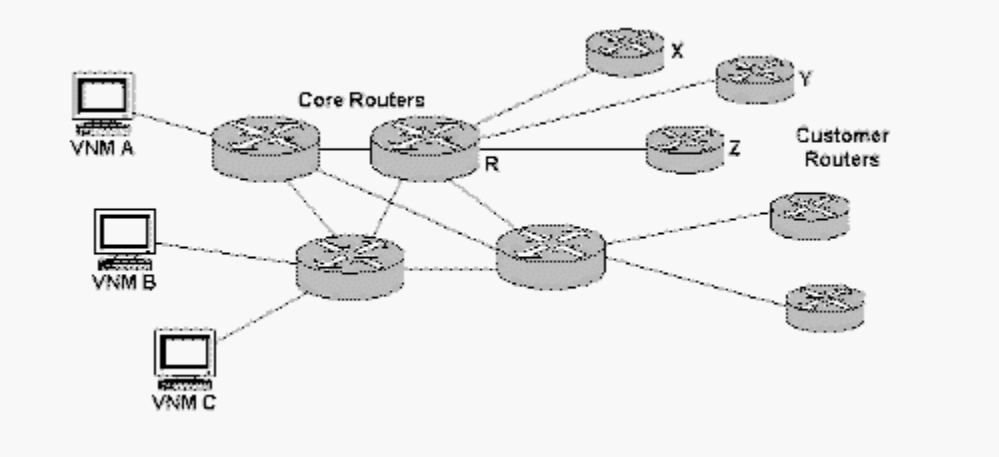
> **NOTE**
> When you have multiple proxy models in a Global Collection topology, you can collapse them to a single icon, merging all connections.

Follow these steps:

1. Select the device model to designate as a proxy model.
2. Click the Information tab, and expand the DX NetOps Spectrum Modeling Information subview.
3. Locate the 'Is a Proxy Model' setting, click 'set,' and select Yes.
   Event creation is disabled for this model. The model now serves as a proxy model.

Cross-Landscape Fault Correlation Example

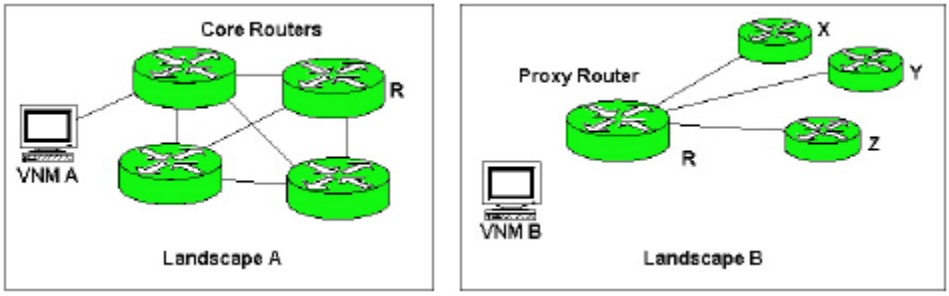
The following diagram provides an example of a network with multiple landscapes:



Landscape A, the local landscape, contains core routers, including Router R. Landscape B, the remote landscape, contains customer routers with the core Router R modeled a second time, this time as a proxy, as shown in the following

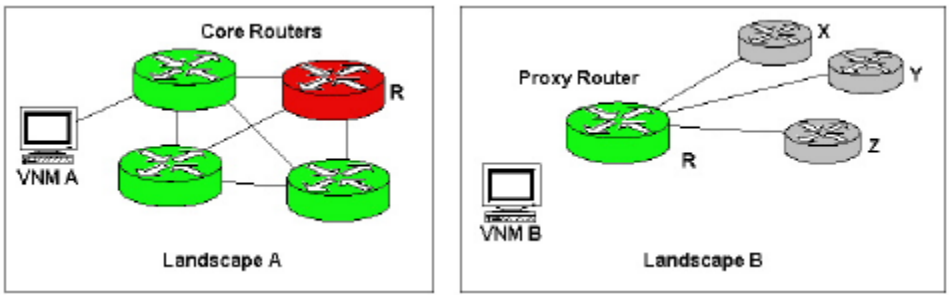
diagram. This proxy model has its IsEventCreationEnabled attribute set to No. The device is being polled, but will not generate events or alarms.

Router R in Landscapes A and B:



If Router R goes down, as shown in the following diagram, Landscape B loses contact with the proxy and customer routers. However, only one red alarm is generated, in Landscape A. The proxy router stays green in Landscape B, where the alarms are suppressed because the proxy model's IsEventCreationEnabled attribute is set to No.

Alarm with Cross-Landscape Fault Correlation:



Configuring Port Status Monitoring

DX NetOps Spectrum provides the following methods of monitoring the status of ports:

- **Link Traps**
  Link traps allow you to monitor the status of ports without the cost of polling. However, traps are not always the most reliable notification mechanism of port status.
- **PollPortStatus**
  The PollPortStatus feature lets you poll the status of a port even if its connectivity is not modeled in DX NetOps Spectrum.
- **Live Pipes**
  Live Pipes let you turn on port status monitoring for individual links. This is a more reliable monitoring method than traps because DX NetOps Spectrum will periodically poll the status of the link (with an increased cost in performance). In addition, Live Pipes let you graphically verify which links are being monitored.
- **WA_Link Port Monitoring**
  WA_Link models automatically enable a live pipe for any connected ports.
- **NetworkLinkType**
  DX NetOps Spectrum automatically maintains the port-level attribute NetworkLinkType based on the model class of the two connected devices. The attribute lets you set up management policies based on the type of link a port is involved in.

  NOTE
  For more information about the NetworkLinkType attribute and management policies in general, see the Policy Manager section.

The possible values for NetworkLinkType include the following:

- 0 = No Link
- 1 = Router Link
- 2 = Switch Link
- 3 = Shared Access Link
- 4 = End Station Link
- 5 = Wide Area Link
- 6 = Internal Link
- 7 = Unknown Link
- 8 = Network Cloud Link

If the connectivity of the port is not modeled, NetworkLinkType will be set to Unknown Link. When the connectivity of the port is modeled, the value of NetworkLinkType is maintained as described in the following table:

Connection	Attribute Value	Link Type
Router > Router	1	Router
Router > Switch Router	1	Router
Router > Switch	1	Router
Router > Hub	1	Router
Router > Workstation Server	1	Router
Switch Router > Switch Router	2	Switch
Switch Router > Switch	2	Switch
Switch Router > Hub	3	Shared Access
Switch Router > Workstation Server	4	End Station
Switch > Switch	2	Switch
Switch > Hub	3	Shared Access
Switch > Workstation Server	4	End Station
Hub > Hub	3	Shared Access
Hub > Workstation Server	4	End Station
Any port connected to a WA_Link model	5	Wide Area
Any backplane connecting inside a hardware device	6	Internal Link
EVPN Discovery is run and Provider_Clouds are created	8	Network Cloud Link

Port Status Polling Criteria

In general, the status of a port is polled when the following criteria are met:

- The PollingStatus (0x1154f) of the port model must be TRUE.
- The Polling_Interval (0x10071) of the port model must be non-zero.
- The PollingStatus of the port's device model must be TRUE.
- Neither the port model nor the port's device model can be in maintenance mode; the isManaged (0x1295d) attribute for both must be set to TRUE.

If this criteria is met, port polling occurs with a frequency equal to the Polling_Interval setting.

However, either of the following two conditions override the default polling frequency:

- The port has been down since it was modeled in DX NetOps Spectrum and the Port Always Down Alarm Suppression attribute is set to Enabled.

  NOTE
  If the Port Always Down Alarm Suppression attribute is set to Disabled, the port will be polled as described above.
- The port is administratively down (that is, ifAdminStatus attribute is set to Down).

If these criteria are met, the polling frequency is reduced to once per hour (every 3600 seconds). Plus, all red alarms on the down ports are suppressed, and a gray condition is asserted. Administratively down ports remain brown.

Port Status Events and Alarms

The port status monitoring engine uses the events and alarms listed in the following table to notify you of a change in status.

Event Description	Event ID	Alarm Description	Alarm ID	Port Condition Color
Port status good	0x10d10	N/A	N/A	Green
Port status bad	0x10d11	BAD LINK	0x1040a	Red
Port status disabled	0x10d12	LINK DISABLED	0x1040b	Brown
Port status unknown	0x10d13	LINK STATUS UNKNOWN	0x1040e	Gray
Port status unreachable	0x10d14	UNREACHABLE LINK	0x1040c	Gray
Port status initial	0x10d15	N/A	N/A	Blue
Port lower layer down	0x10d16	BAD LINK, BUT ALARM WAS SUPPRESSED	0x1040f	Gray
Port up, but linked with down port	0x10d17	LINK MAY NOT BE UP	0x10410	Gray
Port connected to down port or device	0x10d18	PORT ALARM SUPPRESSED	0x10411	Gray
Port status bad, but connected to WA_Link whose LinkFaultDisposition is LinkOnly	0x10d2d	PORT ALARM SUPPRESSED	0x10d2d	Gray

Link Traps

Traps provide a means for network devices to let a management system know that a significant event has occurred on the network. Link Down and Link Up traps are perhaps the most important traps when it comes to port status monitoring. These traps tell the management system that a port has either become inoperable, or has come back up.

When DX NetOps Spectrum receives a Link Down trap, it polls the status of the corresponding port once to verify its status and generates one of the events and alarms listed in Port Status Events and Alarms on the affected port.

NOTE
DX NetOps Spectrum generates a yellow alarm on the device model to allow easy access to vendor-specific trap data, however it no longer generates the trap-specific events and alarms on the affected port model.

Once a Link Down trap is received, DX NetOps Spectrum sets the OutstandingLinkDownTrap attribute on the port to TRUE. This will cause DX NetOps Spectrum to poll the status of the port regardless of the port status polling criteria. When DX NetOps Spectrum receives a Link Up trap for the port, or when the port's status is determined to be up based on a poll, the value of the OutstandingLinkDownTrap attribute is set to FALSE and polling will take place based on the value of the port status polling criteria. For more information about when a port is polled, see Port Status Polling Criteria.

When all of the ports for which DX NetOps Spectrum has received a Link Down trap are back up, the yellow alarm on the device will be cleared.

You can use the following attributes to control how DX NetOps Spectrum handles link traps:

- **AlarmOnLinkDownIfTypes**
  This attribute contains a mapping of ifType values to a value which determines how to handle the trap for that particular ifType and model type (0 for never, 1 for always, and 2 for check admin). This can be customized in the MTE on a per-model type basis. When a port model is created, its AlarmOnLinkDownTrap (0x11fc2) attribute is automatically populated with the value which corresponds to its particular ifType.
  ID: 0x1290f
- **AlarmOnLinkDownTrap**
  This attribute sets the alarm generation behavior for receipt of a LINK DOWN Trap Event. Possible settings are:
  – Never (0) = Do not generate an alarm upon receipt of LINK DOWN trap
  – Check Status (1) = Generate an alarm based on the current Admin Status (UP = Generate a Red alarm, otherwise generate a Brown alarm)
  ID: 0x11fc2
- **AssertLinkDownAlarm**
  This attribute is used to determine if the yellow alarm should be generated on the device model. It is read from the port model for which DX NetOps Spectrum received the trap. This attribute is available from the port model's Attributes tab.
  ID: 0x12957

Interface Trap Configuration

For many device models, you can configure the processing of link down traps received for individual port models through the port model's Interfaces tab, Component Detail view, Attributes tab. Here you can access the attributes that let you suppress link down alarms for the selected port model or its parent device model. Consult the DX NetOps Spectrum management module section for the type of device you are interested in to see if that module supports such trap configuration.

> **NOTE**
> You can also use Locater search to select a set of port models and then the Attribute Editor to update in bulk.

PollPortStatus Feature

The PollPortStatus feature lets you monitor the status of a port even if the port's connectivity is not modeled. The PollPortStatus attribute exists for both Device and Port models with a different attribute ID for each of the two model types. This lets you enable and disable port status polling at the device or port level. By default, PollPortStatus is set to TRUE at the device level and FALSE at the port level.

> **NOTE**
> To reduce network traffic, SNMP reads for polled ports on the same device are grouped together into larger SNMP requests. This provides performance benefits that are most noticeable when many ports are polled by this method in a single SpectroSERVER.

Utilizing PollPortStatus to Watch a Connected Port's Status

The PollPortStatus attribute at the device level (0x12809) controls port status polling on a per-device basis. If TRUE, polling is enabled for that device. If FALSE, no ports will be polled, even if a port model's PollPortStatus attribute is TRUE. When changed to FALSE, alarms will be cleared for any port which is not involved in a live pipe.

The PollPortStatus attribute at the port level (0x1280a) controls polling for each port model. If this attribute is set to TRUE (and PollPortStatus for the device is also set to TRUE), the status of the port will be polled, and alarms will be generated if needed. When changed to FALSE, any alarm on the port will be cleared. The following table shows that a port's status will be polled only if PollPortStatus is TRUE for both a given port model and its device model.

Device Model's PollPortStatus Value	Port Model's PollPortStatus Value	Results
FALSE	FALSE	Port status is not polled for any port on device
TRUE	FALSE	Port status is not polled for this port on device
FALSE	TRUE	Port status is not polled for this port on device
TRUE	TRUE	Port status is polled for this port on device

DX NetOps Spectrum watches the polling interval to determine when to poll port status. When a port is polled, the port's status is determined and an appropriate alarm is generated (RED, BROWN, or GRAY) if necessary.

If a BAD LINK alarm is generated on a port (alarm code 0x1040a), and later polling on that port is disabled by changing the value of PollPortStatus to FALSE and disabling Live Pipes, an event will be generated to automatically clear the BAD LINK alarm.

> **NOTE**
> The PollPortStatus attribute can be set to TRUE while the Live Pipes functionality is enabled. This will not cause redundant network traffic.

Enabling Port Status Polling

You can enable port status polling for all future models of a given type using the Model Type Editor (MTE), or on a current, per-model basis using the Command Line Interface (CLI). For example, use the MTE to set PollPortStatus to TRUE for both device and port model types. Then, when polled, interface models will generate appropriate alarms when needed. PollPortStatus can also be set at both the device and port level using the Global Attribute Editor.

> **NOTE**
> For information about using the CLI to enable or disable PollPortStatus for a single model, see the Command Line Interface *section.*

Fault Isolation Settings

The Fault Isolation subview in the VNM model's Information tab lets you configure various aspects of the DX NetOps Spectrum device fault isolation functionality. It contains the following settings:

- **ICMP Support Enabled**
  Specifies whether an attempt should be made to contact a device using the ICMP protocol when trying to ascertain the fault status of the device. When the ICMP_SUPPORT attribute is enabled (set to TRUE), DX NetOps Spectrum looks at the setting of the ICMP_SUPPORT attribute at the device level. If ICMP support is also enabled on the device, DX NetOps Spectrum attempts to contact the device using the ICMP protocol. However, when ICMP_SUPPORT is disabled (set to FALSE), this setting takes precedence over the setting at the device level and prevents any attempt to contact the device using the ICMP protocol for fault isolation.
  Default: Yes
- **ICMP Timeout (msec)**
  Specifies the amount of time (in milliseconds) DX NetOps Spectrum waits for a response to an ICMP ping. If a response is not received within this period of time, DX NetOps Spectrum assumes the device has timed out.
  Default: 3000 milliseconds (3 seconds)
- **ICMP Try Count**

Specifies the total number of attempts made to contact a device using the ICMP protocol before DX NetOps Spectrum determines the device is down.

- **Lost Device Try Count**
  Specifies the number of retries that DX NetOps Spectrum attempts for each SNMP request sent to a device after contact with the device is lost.
  Default: 1
- **Port Fault Correlation**
  Enables port fault correlation and specifies how it should be configured.
  Default: All Connected Ports
- **Contact Lost Model Destruction**
  Specifies whether a device is automatically destroyed when its CONTACT_STATUS is set to false. When enabled, models that have had their CONTACT_STATUS set to lost for a specified period of time, as determined by the Destruction Delay (sec) setting, are destroyed automatically. When disabled, models will never be automatically destroyed as a result of the value of the CONTACT_STATUS attribute.
  Default: Disabled
- **Destruction Delay (sec)**
  Specifies the length of time (in seconds) that a model must continuously have its CONTACT_STATUS attribute set to lost before it is automatically destroyed.
  Default: 604800 seconds (7 days)
- **Destruction Event Generation**
  This field controls whether an event message is created when a model is automatically destroyed. When this field is set to Enabled, an event will be generated each time a model is destroyed as a result of its CONTACT_STATUS being continuously set to lost for the length of time specified in the Destruction Delay Time field. When this field is set to Disabled, no event message will be generated.
  Default: Enabled
- **Router Redundancy Retry Count**
  Specifies the number of times DX NetOps Spectrum will attempt to contact a router's redundant IP addresses if contact is lost with its primary address. The polling interval setting determines the amount of time between each attempt.
  Default: 2
- **Unresolved Fault Alarm Disposition**
  If information about the connectivity of your network model is incomplete, DX NetOps Spectrum may be unable to find the root cause of a network outage. In this case, the status of all devices affected by the outage is set to gray, and a red unresolved fault alarm is generated. This alarm indicates that DX NetOps Spectrum has lost contact with a group of devices, but was unable to pinpoint the cause.
  All of the devices to which DX NetOps Spectrum has lost contact are listed in the impact scope of the alarm. The model name and other details of the lost devices also appear in the event that generated the alarm.
  The Unresolved Fault Alarm Disposition field lets you control how the unresolved fault alarm is generated. When set to Fault Isolation Model, the alarm will be generated on the Fault Isolation model. When set to Device In Fault Domain, the alarm will be generated on one of the devices with which DX NetOps Spectrum has lost contact. When determining which device to generate the alarm on, DX NetOps Spectrum looks for the device with the highest criticality. If the highest criticality is shared by two or more devices, DX NetOps Spectrum generates the alarm on the first of these devices that it finds. If all devices have the same criticality, then DX NetOps Spectrum chooses the device with the lowest model handle.

  NOTE

  The default behavior is to place Unresolved Fault Alarms on the Fault Isolation model. Most users prefer to have unresolved fault alarms placed on a device in the fault domain, where the problem is more visible. This default behavior can be changed from "Unresolved Fault Alarm Disposition.
- **WA Link Fault Isolation Mode**
  Specifies whether WA_Link models are considered neighbors for fault isolation purposes. Options are Normal and Transparent.
  Default: Normal

Port Fault Correlation

DX NetOps Spectrum lets you customize its fault isolation algorithm to resolve the root cause of a network outage to the port level. This is most desirable in cases where a single physical port, such as a Frame Relay interface, supports multiple logical connections to remote devices. If the physical port goes down, DX NetOps Spectrum can suppress the alarms on all downstream devices in favor of a single red alarm on the physical interface, thus significantly reducing the number of alarms which need attention. The impact severity and scope of the red alarm on the physical interface will contain all downstream devices, as well as the physical interface.

Port Fault Correlation Options

Use the Port Fault Correlation setting in the VNM model's Fault Isolation subview to configure port fault correlation.

- **Disabled**
  Disables port fault correlation. The root cause of a network outage will remain a red alarm on a device model. However, Fault Isolation will still examine all of the device's connected ports to see if they are all in Maintenance Mode. If so, the alarm on the device model will be suppressed.
- **All Connected Ports**
  Port fault correlation will run, examining all ports that exist on "up" neighbors which are connected to the down device as possible root causes of the outage. No additional manual configuration is required.
- **Management Neighbors Only**
  Enables port fault correlation to run but only examine ports which were previously configured manually as management neighbors as possible root causes of the outage.
- **All Connected Ports -- Multiple Devices Only**
  Enables port fault correlation, examining all ports that exist on "up" neighbors which are connected to the down device as possible root causes of the outage. However, DX NetOps Spectrum will only resolve the outage to the port level if there is more than one device model that would have a red alarm which can be correlated to the port alarm. If only one connected device alarm can be correlated to the port alarm, DX NetOps Spectrum will not suppress the device alarm. Instead, both the port and device alarm will be generated.

Port Fault Correlation Criteria

The following criteria must be met for the root cause of an outage to be resolved to the port level:

- The down device must have only one "up" neighbor. If the down device has more than one "up" neighbor, port fault correlation will not be performed. This is done to reduce the number of alarms for a single problem. If multiple up neighbors were a valid criteria, and all connected ports were down, multiple red alarms would exist, all with the same impact severity and scope. If a device has more than one up neighbor, DX NetOps Spectrum assumes the problem lies with the device, not the upstream ports and creates a single red alarm on the device.
- The down device must have at least one connected port (or management neighbor port) on an "up" neighbor that is down.
- If multiple ports on the "up" neighbor connect to the down device (such as link aggregation), all of the ports must be down.
- A port is considered "down" if it is either operationally down, or the port model has been put into Maintenance Mode.
- There must be an alarm on at least one of the down ports. Otherwise, there would be no alarm to which DX NetOps Spectrum could resolve the outage.
- If Port Fault Correlation is set to Management Neighbors Only, management neighbors for the down device must have been configured before the outage occurred.

Port Fault Correlation Caveats

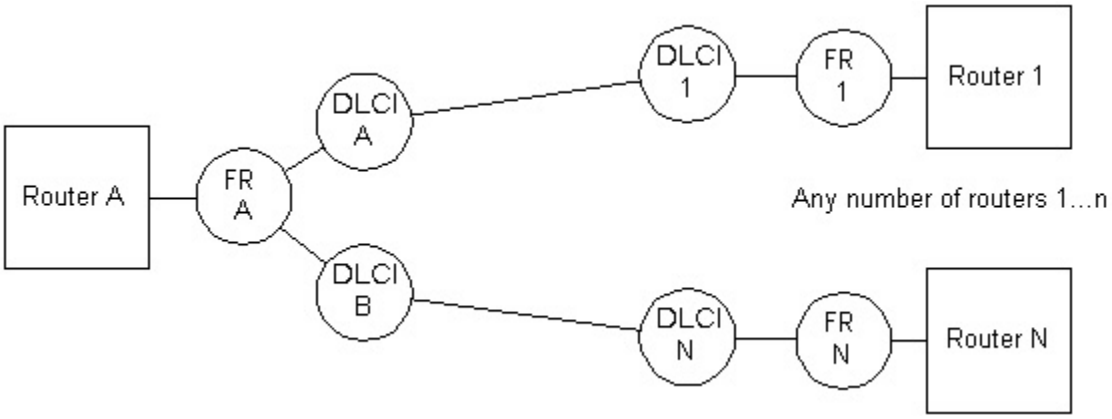
Port Fault Correlation overrides the Suppress Linked Port Alarms setting in the Live Pipes subview. When set to TRUE, this setting suppresses the alarm on an upstream port if it's connected to an unreachable device. If Port Fault Correlation is enabled, and the upstream port is the root cause of an outage, DX NetOps Spectrum forces the upstream port to alarm.

Only the Criticality of the alarmed port will be used in the impact severity and scope calculation of the root cause alarm. The Criticality of any sub-interfaces (such as DLCI ports) will not be included.

Port Fault Correlation is supported by Device models only. Models such as Fanouts and Unplaced do not support this feature. WA_Link models have their own mechanism for supporting port fault correlation, Link Fault Disposition, which is explained in Wide Area Link Monitoring.

If multiple ports on the “up” neighbor connect to the down device (e.g. link aggregation), and all of the ports are down, multiple red alarms will exist as the root causes of the outage. Each red alarm will contain the same impact severity and scope. The root cause of the outage in this case is all of the ports, not just one of them.

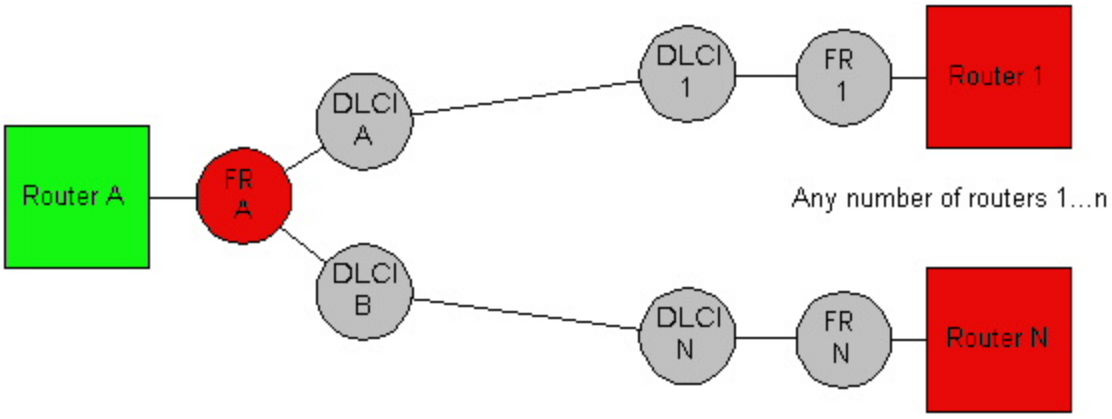
Example Port Fault Correlation Scenario 1



The previous diagram assumes that DX NetOps Spectrum must communicate through Router A to reach Routers 1 through N, and that this is the only means by which DX NetOps Spectrum can reach them. Each remote router is connected to Router 1 using a frame relay link. In DX NetOps Spectrum, this is modeled by connecting each DLCI port model to the other device.

If the physical frame relay interface (FR A) goes down in this scenario, all virtual circuits provisioned on the interface will go down as well. With Port Fault Correlation disabled, the alarms shown in the following diagram will occur.

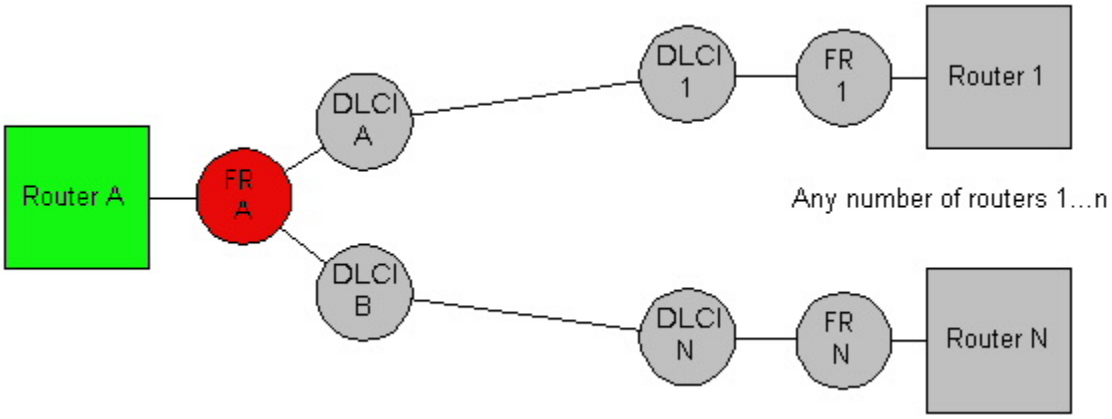
Fault Scenario 1: Alarms without Port Fault Correlation



If a trap is received for FR A going down (or a live pipe is configured to be on), the physical frame relay interface will have a red alarm on it. In addition, all routers connected to the frame relay interface will have a red alarm on them. This could mean multiple red alarms could be generated by DX NetOps Spectrum for a single problem.

Port Fault Correlation reduces the number of alarms generated for this problem to a single alarm without requiring any manual configuration beforehand. The following diagram shows the results with Port Fault Correlation enabled.

Fault Scenario 1: Alarms with Port Fault Correlation



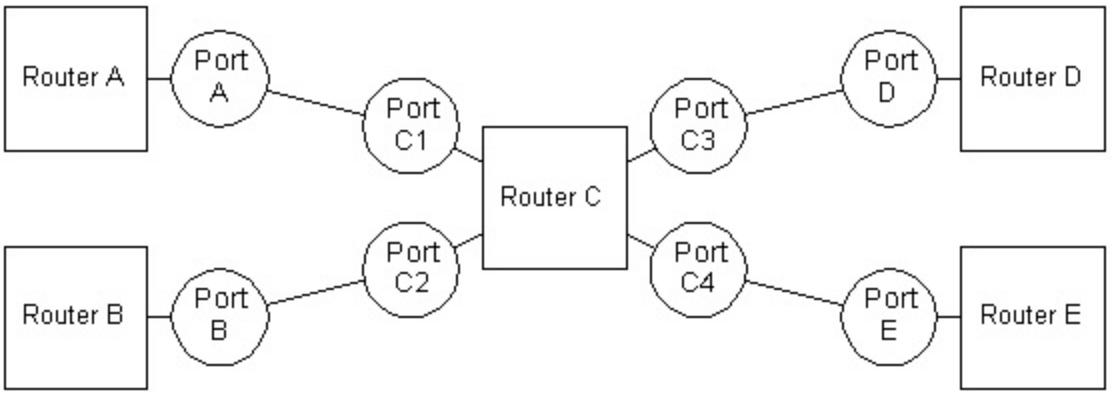
A single red “Bad Link” alarm will be seen in the Alarms tab. That alarm will have an Impact Scope and Severity which contains the following models: FR A, Routers 1 through N, and all unreachable devices that are downstream from Routers 1 through N.

Example Port Fault Correlation Scenario 2

This fault scenario illustrates the benefits of setting the Suppress Linked Port Alarms and Port Fault Correlation attributes as recommended in Suggested Port Fault Settings for Optimal Fault Notification.

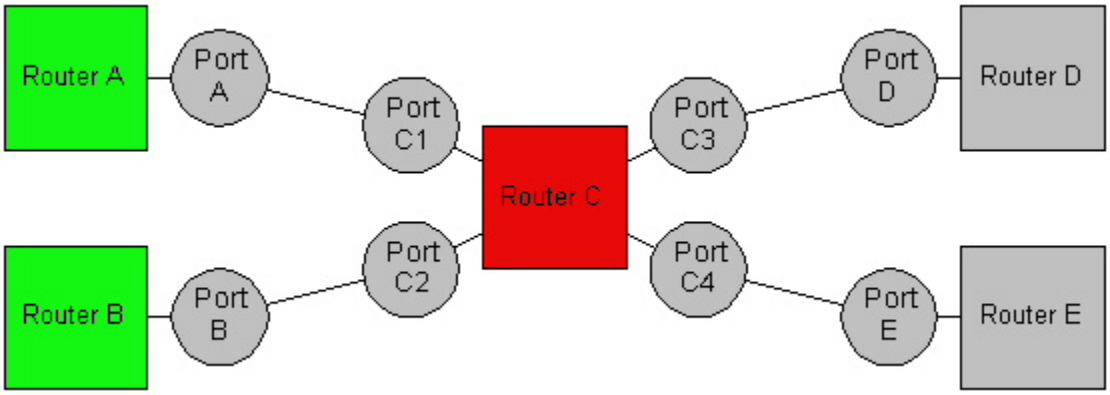
The following diagram assumes that the VNM must communicate through Routers A and B to reach Routers C, D, and E, and that is the only means by which the VNM can reach them. In DX NetOps Spectrum, port-level connectivity is modeled as shown.

Fault Scenario 2: Multiple “Up” Neighbors



In this scenario, Router C goes down, which causes DX NetOps Spectrum to lose contact with Routers C, D, and E, and makes Ports A and B go down as well. If Suppress Linked Port Alarms is set to TRUE, and Port Fault Correlation is set to All Connected Ports, only a single red alarm on Router C will result, as shown in the following diagram:

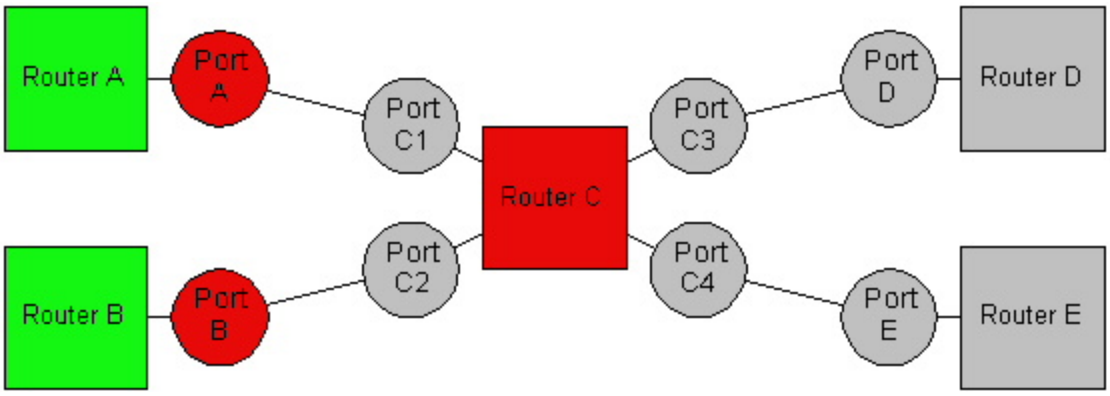
Fault Scenario 2: Multiple “Up” Neighbors



The upstream ports (Ports A and B) have their alarms suppressed because Suppress Linked Port Alarms is set to TRUE. Even though Port Fault Correlation is enabled, Router C has multiple “up” neighbors, so the fault won't be resolved to the port level. When this occurs, DX NetOps Spectrum assumes the fault is with the device itself, not the connected ports.

If you set Suppress Linked Port Alarms to FALSE, and Port Fault Correlation is still set to All Connected Ports, Router C and the upstream ports will be alarmed (if the status of the ports is being polled, or DX NetOps Spectrum receives a LinkDown trap), as shown in the following diagram:

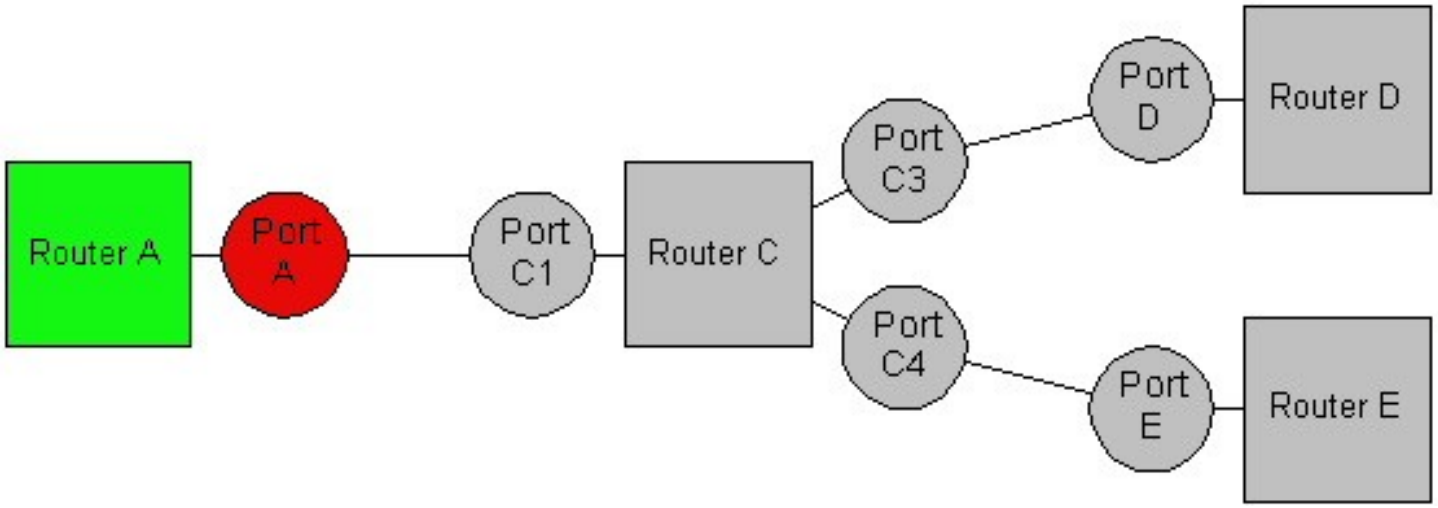
Fault Scenario 2: Multiple “Up” Neighbors



Once again, the fault wasn't resolved to the port level because Router C has multiple “up” neighbors. Since Suppress Linked Port Alarms is disabled, DX NetOps Spectrum will alarm the upstream ports.

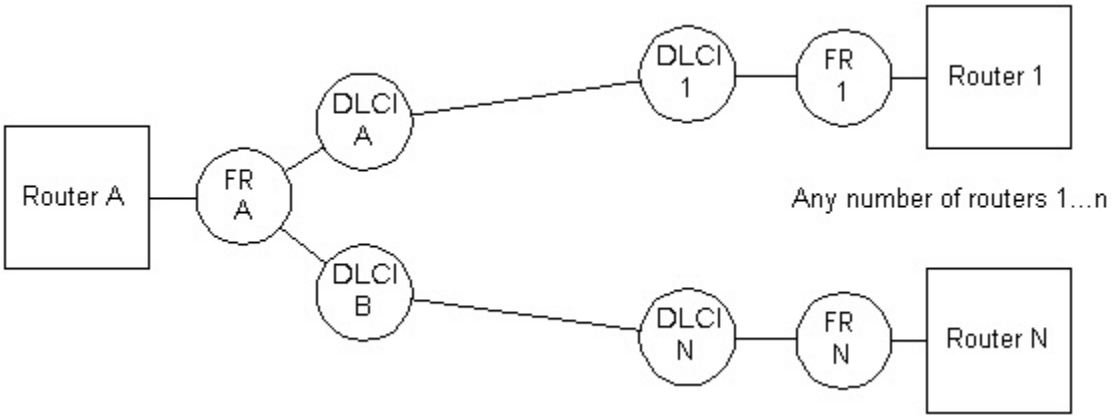
If Router C had only one “up” neighbor, as shown in the following diagram, even if Suppress Linked Port Alarms were set to TRUE (assuming Port Fault Correlation is still set to All Connected Ports), DX NetOps Spectrum will resolve the fault down to the port level. Port Fault Correlation forces the upstream port to be alarmed, and the alarm on Router C is suppressed.

Fault Scenario 2: Single “Up” Neighbor



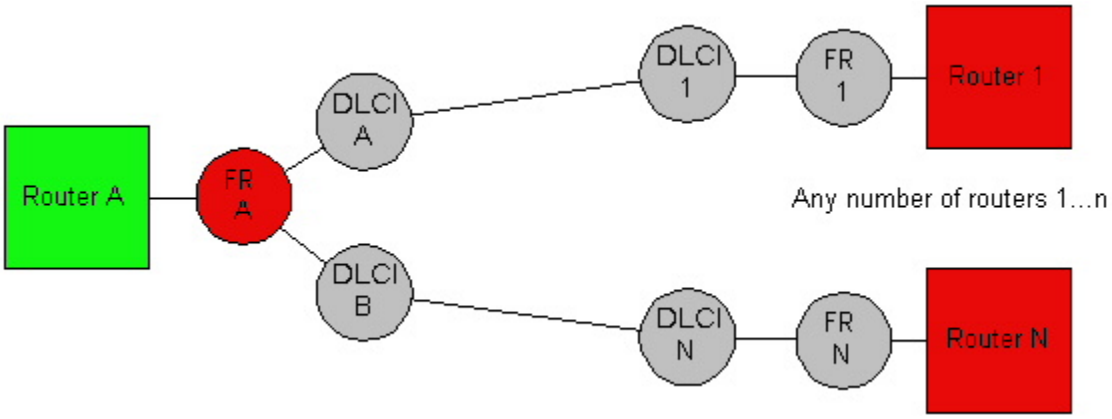
Example Port Fault Correlation Scenario 3

This fault scenario demonstrates what happens when Port Fault Correlation is set to All Connected Ports--Multiple Devices Only. It assumes that DX NetOps Spectrum must communicate through Router A to reach Routers 1 through N, and that this is the only means by which DX NetOps Spectrum can reach them. Each remote router is connected to Router 1 using a frame relay link. This is modeled by connecting each DLCI port model to the other device, as shown in the following diagram:



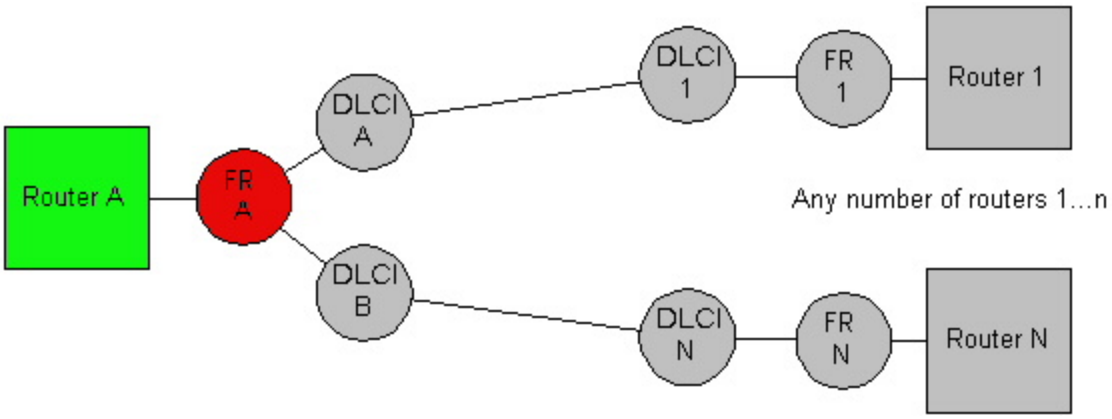
Assume the physical frame relay interface (FR A) goes down. This means that all virtual circuits provisioned on the interface will go down as well. With Port Fault Correlation disabled, the alarms shown in the following diagram will occur:

Fault Scenario 3: Alarms without Port Fault Correlation

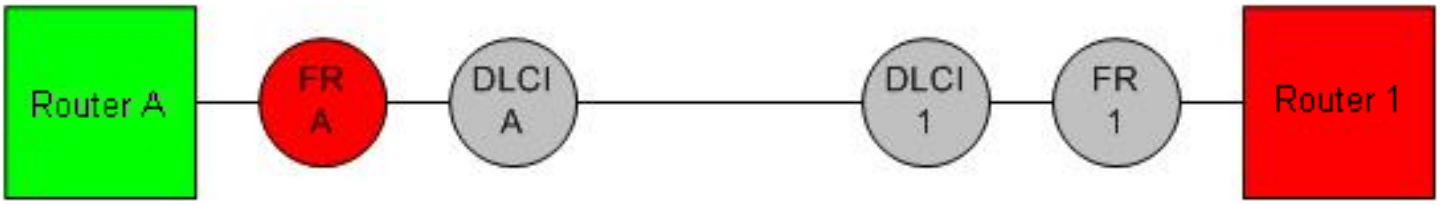


With Port Fault Correlation set to All Connected Ports--Multiple Devices Only, the alarms in the following diagram will occur because multiple devices can be correlated to the frame relay interface.

Fault Scenario 3: Port Fault Correlation Set To “All Connected Ports--Multiple Devices Only”



With Port Fault Correlation set to All Connected Ports--Multiple Devices Only, if there is only one router lost because of a down link, then the alarm on the remote router will not be suppressed.



Port Fault Correlation Anomalies

If a red alarm is generated on a port model as the root cause of an outage, you may then choose to put that port model into Maintenance Mode. If so, the red alarm would be replaced with a brown alarm. The brown alarm will still contain the same impact severity and scope (except the maintenance port will no longer contribute to the impact). If you then decide to take the port out of Maintenance Mode, the red alarm will reappear. It is possible, in this scenario, for the impact scope and severity of the red alarm to be lost.

Wide Area Link Monitoring

DX NetOps Spectrum polls any ports connected to a WA_Link model for status automatically. This polling is controlled by the PollingStatus and Polling_Interval of the WA_Link model.

If the PollingStatus of a WA_Link model is TRUE, and its Polling_Interval is non-zero, DX NetOps Spectrum automatically makes the pipes that are connected to a WA_Link "live" and sets the PollPortStatus for port models to TRUE. The live pipe lets you visually verify that DX NetOps Spectrum is monitoring the status of the connected ports.

If you disable the live pipe but leave the PollingStatus of the WA_Link set to TRUE, the Polling_Interval set to a non-zero number, and the ports' PollPortStatus set to TRUE, DX NetOps Spectrum continues to monitor the status of the ports.

> **NOTE**
> WA_Link models can only represent point-to-point connections, such as T1 and T3 lines. No more than two devices can be connected to it at a time.

LinkFaultDisposition

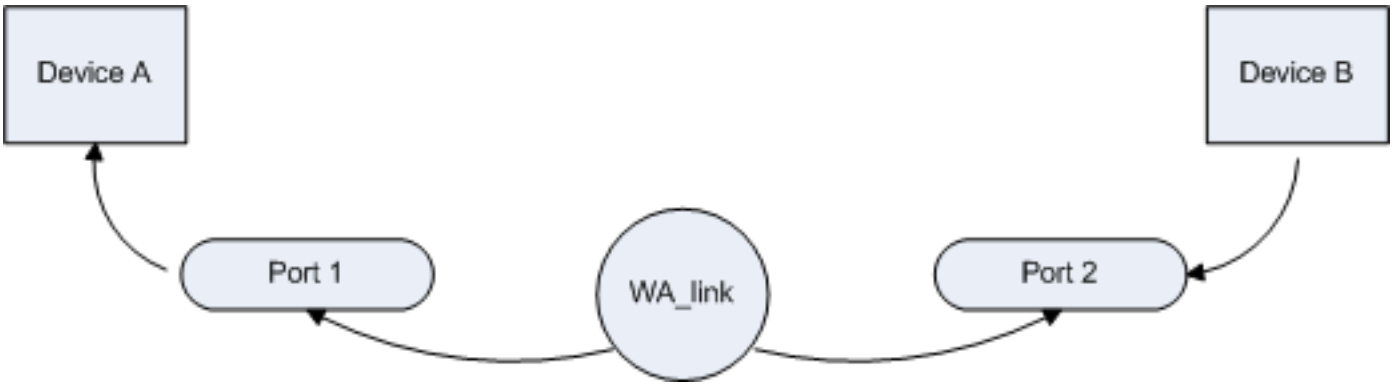
The LinkFaultDisposition setting provides control and flexibility over fault alarming. When a wide area connection goes down, alarms can be generated on ports and on the link model. You can set the LinkFaultDisposition (0x129e2) attribute from the WA_Link model Attributes tab.

LinkFaultDisposition can be set to one of the following three modes:

- **BothPortsAndLink**
  If set to BothPortsAndLink, alarms are generated on both the connected ports and on the link model. This is the default setting.
- **PortsOnly**
  If set to PortsOnly, only the connected ports are alarmed, and the WA_Link is suppressed.
- **LinkOnly**
  If set to LinkOnly, only the WA_Link model will be alarmed, and the ports will be suppressed.

Wide Area Link Monitoring Scenarios

Consider the sample WA_Link network topology shown in the following diagram:



The following tables illustrate two possible WA_Link monitoring scenarios for this topology.

Scenario 1: Link Goes Down, Device B Loses Contact

LinkFaultDisposition	Port 1 Condition	WA_Link Condition	Port 2 Condition
BothPortsAndLink	Red	Red	Gray
PortsOnly	Red	Gray	Gray
LinkOnly	Gray	Red	Gray

NOTE
If the Suppress Linked Port Alarms setting is set to TRUE, then the alarm on the upstream port will be suppressed, even if Link Fault Disposition is set to BothPortsAndLink. If Link Fault Disposition is set to Ports Only, and the Suppress Linked Port Alarms setting is set to TRUE, DX NetOps Spectrum generates an alarm on the Port model.

Scenario 2: Link Goes Down, Device B Remains Reachable

LinkFaultDisposition	Port 1 Condition	WA_Link Condition	Port 2 Condition
BothPortsAndLink	Red	Red	Red
PortsOnly	Red	Gray	Red
LinkOnly	Gray	Red	Gray

NOTE
If the Suppress Linked Port Alarms setting is set to TRUE, then only one of the ports is alarmed (red). The other port will be suppressed (gray). If Link Fault Disposition is set to Ports Only, and the Suppress Linked Port Alarms setting is set to TRUE, DX NetOps Spectrum generates an alarm on the Port model.

Wide Area Link Modeling Best Practices

When you model a Wide Area Link, be sure to supply the IP address for the Network_Address parameter. As a best practice, supply values for the WA_Link model Network_Address and Network_Mask attributes based on the subnet of the connected router interfaces. In a proxy environment, the "real" and "proxy" links must have the same value for Network_Address.

DX NetOps Spectrum relies on the Network_Address (0x12d7f) attribute of a WA_Link to find duplicate WA_Link models and to collapse proxies into a single icon.The Network_Address is the only unique attribute that can be used to find duplicate WA_Links. However, this attribute is not automatically populated by DX NetOps Spectrum unless the WA_Link models are created by Discovery. When you manually create WA_Links, the Network_Address and Network_Mask attributes are not automatically populated, even when they are connected to router interface models with valid IP addresses.

In our testing, we have found that the Global Collection topology view does not collapse proxied WA_Link models correctly if the Network_Address parameter is not configured.

Set the Network_Address attribute of the WA Link model to the network ID of the subnet to which the interface belongs. For example, a serial interface has an IP address of 10.253.9.2 and a subnet mask of 255.255.255.252. Set the Network_Address attribute of the WA Link to 10.253.9.0 (10.253.9.2 with subnet mask 255.255.255.252).

In addition, do not simply draw pipes between the router and WA_Link icons. Proper WA_Links require a nested WA_Segment. In addition, an interface model on each router must be connected to the WA_Segment (and not to the WA_Link). This modeling paradigm enables DX NetOps Spectrum to establish fully resolved connections. It also enables the proper display of pipes in the Global Collection topology view.

Port Layer Alarm Suppression

Devices that support advanced network technologies, such as Frame Relay and Link Aggregation, have logical entries in the ifTable representing higher layer interfaces. DX NetOps Spectrum models these logical layers according to the ifStackTable. If you set the use_if_entity_stacking (0x12a83) attribute to TRUE on a device model in the Attributes tab, DX NetOps Spectrum attempts to use information from RFC2737 (Entity MIB) to model these logical layers if the ifStackTable method fails.

When a monitored higher layer port goes down (such as a Frame Relay DLCI, or logical trunk interface), DX NetOps Spectrum will query the statuses of all lower layer interfaces before alarming the port which went down. If all of the lower layer interfaces are down as well, DX NetOps Spectrum will suppress the higher layer interface alarm. A key example is a

physical Frame Relay interface going down which has multiple circuits provisioned on it. All of the higher layer DLCI port models will be suppressed, and the single red alarm will exist on the physical Frame Relay interface.

Port Criticality

You can assign a relative importance value to port models using the port criticality (0x1290c) attribute. The criticality of a port is used in the Impact Severity calculations of an alarm on any port which may cause a network outage. You can also display the criticality of a port in the Alarms tab to allow prioritization of port alarms. The port criticality attribute can be set for an individual port from the port model's Attributes tab.

Live Pipes and Fault Management

Live Pipes functionality lets you turn on port status monitoring for individual links and display the status of a link by using status color indicators. A link is a connection between two devices that DX NetOps Spectrum has resolved to the port level. Live Pipes display a combined condition color from the two resolved ports representing each side of the link.

When a pipe is deleted, DX NetOps Spectrum removes all of its underlying associations (such as links_with and connects_to). If the pipe being deleted represents more than one link, DX NetOps Spectrum asks you to confirm the deletion.

Enable or Disable Live Pipes System-Wide

Live Pipes are enabled system-wide by default. Without Live Pipes enabled on a system-wide basis, enabling Live Pipes for individual links is not available.

To enable or disable Live Pipes system-wide

1. Expand the Live Pipes subview in the VNM model's Information tab.
2. Click 'set' in the Live Pipes field and select Enabled or Disabled from the drop-down list as desired.

Enable or Disable Live Pipes on Individual Links

Live Pipes for individual links are disabled by default. All individual pipes are gold or silver until Live Pipes are enabled. When an individual live pipe is enabled, the ok_to_poll (0x11dd8) attribute is set to TRUE for both ports that are involved in the link. The status of the linked ports is monitored if ok_to_poll is TRUE and the port status polling conditions in the Port Status Polling Criteria section are met.

You can enable a live pipe for an individual link.

Follow these steps:

1. Right-click the link that you want to enable as a live pipe, and select 'Enable/Disable Live Links.'
2. Select the check box for the link to enable and click OK.
   The Enable/Disable Live Links dialog closes. The link that you selected is enabled as a live pipe.

Receiving Port Alarms

To receive alarms on a port model, the following conditions must be met:

- Live Pipes must be enabled system-wide
- Live Pipes for the link of interest must be enabled separately

> **NOTE**
> If there is no model on the other side of the link, an alarm can still be generated when the status of the link changes by setting a watch on the MIB-II ifOperStatus attribute. When the ifOperStatus attribute returns a

value other than 1, an alarm is generated. With this method, an alarm can still be generated even if the port is intentionally down (the ifAdminStatus attribute has been set to OFF).

You can set the default value of ok_to_poll in the MTE for any port model type. When a port connection is made, DX NetOps Spectrum will set both ports' ok_to_poll attributes to their MTE default values so that the pipe will automatically become live if the you want. When the connection is removed, the value for ok_to_poll remains at the MTE default value.

When DX NetOps Spectrum notices a change in the link's status, it will generate one of the events and alarms listed in Port Status Events and Alarms on the two ports involved in the link, and will also change the color of the live pipe to reflect its new status.

Port Status Monitoring Settings

The settings described in the following table let you to control the service of Live Pipes. These settings appear in the VNM model's Information tab, Live Pipes subview.

Note: The settings described in this section apply to port status monitoring throughout DX NetOps Spectrum, not just ports associated with Live Pipes.

- **Live Pipes**
  Setting this (global) option to Disabled turns off all pipes in the SpectroSERVER and no status polling will be performed for any ports associated with a pipe. However, any port which also has the PollPortStatus attribute set to TRUE will still be polled for status changes.
  Attribute ID: 0x11df9
- **Alarm Linked Ports**
  Setting this option to TRUE will cause ports with a good status to have a gray alarm generated on it when linked with a bad or unreachable port.
  Attribute ID: 0x11fbd
- **Suppress Linked Port Alarms**
  Setting this option to TRUE causes ports with a bad status to suppress their red alarms and generate a gray alarm if either the linked port or the connected device is bad or unreachable. Only one port in the link will be alarmed red. The other will be gray.
  Attribute ID: 0x11fbe
- **Port Always Down Alarm Suppression**
  Enabling this option will cause DX NetOps Spectrum to suppress red alarms and assert a gray condition if a port has always been down since first being modeled in DX NetOps Spectrum.
  Attribute ID: 0x12a03

  NOTE
  Any port status alarm will be automatically cleared by DX NetOps Spectrum if the port's ok_to_poll and PollPortStatus attributes are both set to FALSE.

Monitoring Physical and Logical Connections

DX NetOps Spectrum can monitor the related physical connections of multilink bundles that are resolved to the logical connection via a Live Pipe. When Live Pipes are enabled on the logical connection, the physical ports on one side of the multilink bundle are polled. If Live Pipes are disabled, the polling on the associated physical ports are stopped.

NOTE
You may need to configure the MultiLinkVirtualIfTypes (0x12e3d) attribute to contain the list of ifType values for multilink bundles for which DX NetOps Spectrum should also poll the related physical connections. This attribute is pre-configured with the pppMultilinkBundle (108) ifType.

To avoid creating multiple alarms when a single physical connection goes down, DX NetOps Spectrum only polls the physical connections on one side of the multilink bundle. The side that is polled is determined by reading the Criticality (0x1290c) attribute for the logical connections and choosing the side with the higher value. If the values are equivalent,

the connection with the lowest model handle is chosen. Also, the NetworkLinkType (0x12a79) attribute on the physical port models on the side that is polled is set to the same value of the NetworkLinkType attribute of the multilink virtual interface model.

The polling of the non-resolved physical ports on one side of the multilink bundle results in the following alarm behaviors:

- An alarm is generated on a physical port model whenever a physical connection that is part of a multilink bundle goes down.
- If the logical connection is down and at least one of the related physical connections is up, an alarm is generated on a multilink virtual interface model.
- If all of the physical connections are down and the logical connection also goes down, the logical interface is put into the suppressed (grey) state.

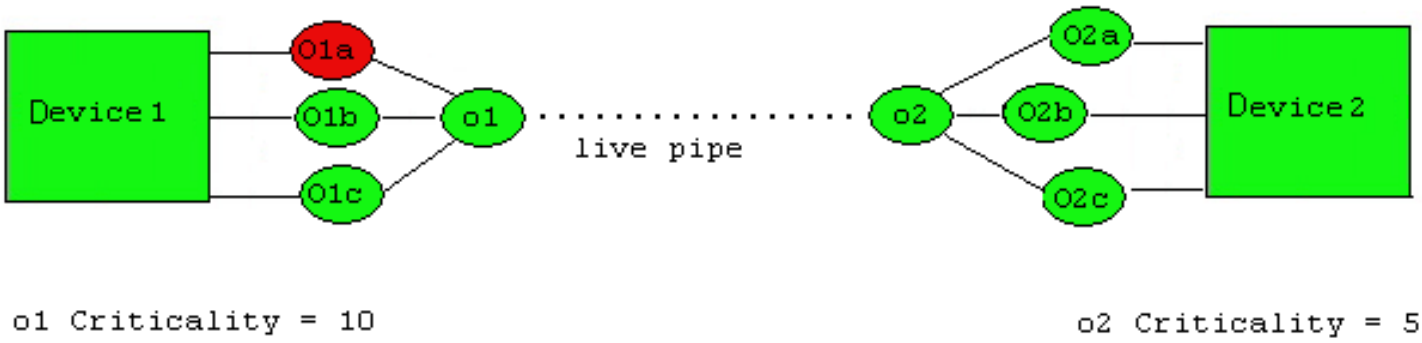
> **NOTE**
> The physical ports on the side that is not being polled remain in the green state at all times, unless some other type of monitoring has been enabled on them.

Examples: Monitor Physical and Logical Connections

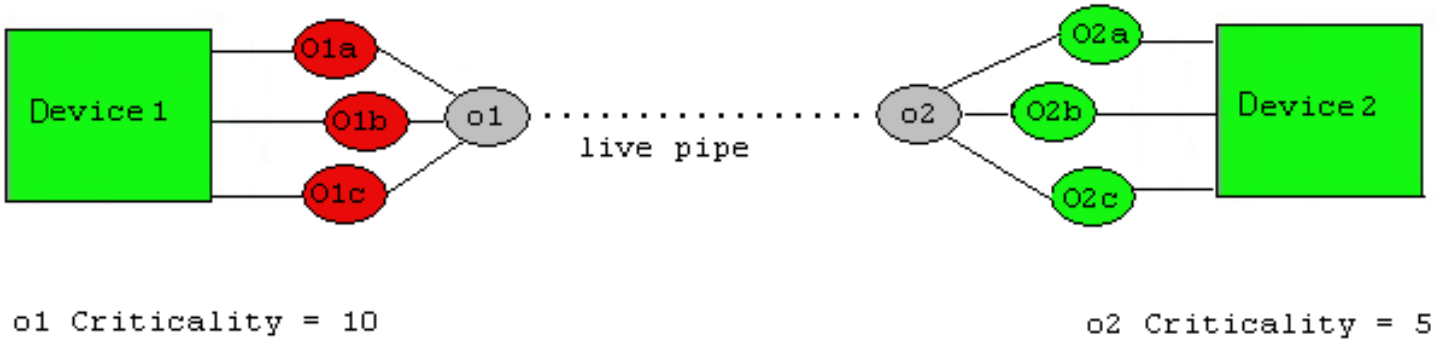
In the following examples, two devices are connected through a multilink bundle of three pairs of physical port links. “O” represents a physical port and “o” represents a logical interface. Live Pipes are enabled on the logical connection between the multilink bundles. There are physical connections between O1a and O2a, O1b and O2b, and O1c and O2c, but these connections are not resolved in DX NetOps Spectrum.

When Live Pipes are enabled on the logical connection between the multilink bundles, the three physical ports on Device 1 (O1a, O1b, O1c) are also polled because o1’s Criticality (10) is greater than o2’s Criticality (5).

If the physical connection between O1a and O2a goes down, an alarm is generated on port O1a and port O2a remains green:



If all three physical connections are down, alarms are generated on ports O1a, O1b, and O1c, ports O2a, O2b, and O2c remain green, and logical connections o1 and o2 are put into a suppressed state:



Suggested Port Fault Settings for Optimal Fault Notification

In SPECTRUM 7.0 and later, the default settings for both Suppress Linked Port Alarms and Port Fault Correlation have changed. The default value of Suppress Linked Port Alarms has changed from FALSE to TRUE. This setting will suppress red alarms on ports that are connected to another down port or unreachable device. The default value of Port Fault Correlation has been changed from Management Neighbors Only to All Connected Ports. This eliminates the need for you to manually configure management neighbors before a fault occurs so that port fault correlation will work properly. If you have not previously used ‘Management Neighbors’ you will not need this setting.

> **NOTE**
> Setting Suppress Linked Port Alarms to FALSE and Port Fault Correlation to Management Neighbors Only will approximate the fault notification behavior of SPECTRUM 6.6 with Service Pack 3.

CA recommends changing the default setting for Link Fault Disposition on WA_Link models. The default setting of BothPortsAndLink will result in multiple alarms if the link fails. Consider changing the setting to Link Only or Ports Only. Link Only is best in environments where the name of the WA_Link models or notes on these models is meaningful. Changing the setting to Ports Only may be better if fault notification consistency is most important. That is, regardless of the topology, you would prefer an alarm on a port model if there is a link failure.

Device Criticality

The Device Criticality setting, accessible from a device model’s Attributes tab, specifies the relative importance of the device within the network being modeled. When DX NetOps Spectrum loses contact with the device, this value is summed for the device itself and all of its downstream neighbors for which a gray condition is now being asserted (because their actual status cannot be determined). The aggregate device criticality value is displayed as the Impact Severity value of the associated alarm in the alarm’s Impact tab. The default value for all devices is “1”; you can increase this value as desired depending on how important you consider the device to be; the higher the number, the more critical the device is to the network.

Configuring Fault Management for Pingables

Device fault isolation is faster and more reliable when device models have knowledge of each other as neighbors. When a fault occurs, each device model that is lost sends an ARE_YOU_DOWN action to all of its neighbor models. Depending on the answers that neighbor device models send, the lost models either turn gray or red.

Establish neighbor relations by creating a Connects_to association between a device model and the port model of another device. The act of pasting a device model on the interface of another device model adds each device model to the other's neighbor list.

Pingable models lack ports. As a result, in older versions of DX NetOps Spectrum, neighbor associations could not be established for Pingable models without the use of an inferred connector, like a Fanout. However, it is a best practice to create a relationship between models that is resolved directly. Placing both neighbor models in a Fanout means that fault resolution occurs indirectly. Fanouts and other inferred connector model types resolve faults differently and less directly during fault isolation.

Connect Pingable Models

You can connect Pingable models to each other in the topology view. Connect models to create a Connects_to association between them and receive more status information. You can connect Pingable models using one of the following methods:

- Establish neighbors by drawing pipes between device models. Drawing a pipe between two Pingable models establishes a Connects_to association between the two models, making them neighbors.
- You can create a Connects_to association between two Pingable models using the DX NetOps Spectrum Command Line Interface. Use the following syntax:

```
./create association rel=Connects_to
lmh=<model handle of Pingable A>
rmh=<model handle of Pingable B>
```

When the Connects_to association is established, you see a gold pipe between the two connected models. When a fault occurs, each Pingable model sends the other an ARE_YOU_DOWN action.

Mapping Traps from Other Models to Pingable Models

You can map traps from multiple IP addresses to a single pingable model using the Command Line Interface (CLI) update command. You create the mapping by adding the IP addresses to the deviceIPAddressList (0x12a53) attribute on the pingable model.

Before you can specify the mapping, you must add the following option to the .vnmrc file if the option is not already included in the file:

```
enable_traps_for_pingables=TRUE
```

You can also remove mappings with CLI, and you can configure OneClick to display IP addresses mapped to Pingable models.

To map traps from other IP addresses to a pingable model

1. Connect to the SpectroSERVER with CLI.

   NOTE
   For more information about using CLI, see the Command Line Interface section.
2. Invoke the update command:

   ```
   ./update
   ```
3. Add additional IP addresses to the deviceIPAddressList attribute (0x12a53) for the Pingable model you want to designate as a trap destination. The following example shows three IP addresses added to the attribute:

   ```
   update mh=<pingable's mh> attr=0x12a53,iid=10.253.8.34,val=0
   update mh=<pingable's mh> attr=0x12a53,iid=10.253.8.65,val=0
   update mh=<pingable's mh> attr=0x12a53,iid=10.253.9.17,val=0
   ```
4. Verify that the IP addresses were added:

   ```
   show attributes attr=0x12a53 mh=<pingable's mh>
   ```

Enable a Device IP Address View for a Pingable in OneClick

If the Device IP Address List category is not included under the Information tab for Pingables, complete the following procedure.

To enable a Device IP Address view for a Pingable in OneClick

1. Open the view-pingabledetails-config.xml file located in the following directory:

   ```
   <$SPECROOT>/tomcat/webapps/spectrum/WEB-INF/topo/config/
   ```
2. Uncomment the following line:

   ```
   <field-subview idref="devipaddrlist-subview-config"/>
   ```
3. Restart OneClick.
   The Device IP Address List category appears under the Information tab for Pingable models. If you map IP addresses to a pingable model, the addresses appear in the list.

Remove an IP Address Mapping from a Pingable Model

When addressing schemes change in your environment, keep model information up to date. Use the DX NetOps Spectrum Command Line Interface (CLI) to remove an IP address mapping from a Pingable model.

Follow these steps:

1. Connect to the SpectroSERVER with CLI.

   NOTE
   See the Command Line Interface section for more information about using CLI.
2. Invoke the update command:
   ```
   ./update
   ```
3. Remove the IP address from the deviceIPAddressList attribute (0x12a53).
   ```
   update mh=pingable mh attr=0x12a53,iid=10.253.8.65,remove
   ```
4. The following example shows one IP address removed from the attribute:
   Verify that the IP address was removed:
   ```
   show attributes attr=0x12a53 mh=pingable mh
   ```

Pingable Model Names

Pingable, or non-SNMP managed, device models are defined in Spectrum as devices capable only of communication through Ping.

The Model Naming configuration that controls how Spectrum determines the names of models it creates is found on the VNM model for each Landscape. It is found in the VNM models Information tab, in the SpectroSERVER Control section.

The primary two configurations that impact model naming are:

- Use Fully Qualified Host Name
- Model Naming Order

User Fully Qualified Host Name has a Yes or No setting. If a devices name is device.company.com, if that is set to Yes, it will be named with, device.company.com

If that is set to No, it will be named with **device**. The default setting is No. Model Naming Order default settings are:

- SysName: SysName is obtained from the MIB Object of the device that lists its SysName value
- IP Address: IP Address is the IP Address of the device
- Name Service: Name Service is the DNS name of the device, also found through the 'nslookup <IP_Address> command.

The default Spectrum Naming configuration for determining Model Names noted above will result in Pingable models being named with IP Addresses. If that order is changed after the Pingable models are created, they will be ignored when the 'Reevaluate All Model Names' function is launched and will not be renamed.

There are two other methods of creating Pingable models. The first, the 'Model by Type' option from the Topology tab will also ignore those naming configurations and will create a model named with IP address if the devices name is not specified when setting up the 'Model by Type' option.

The second, Discovery Configurations launched from the Auto Discovery UI, does honor the Model Naming configuration settings. Some examples of model names when using the Discovery configurations are as follows using the example of the device with Fully Qualified Host Name of 'device.company.com':

Model Naming configuration settings:

Use Fully Qualified Host Name = No
Model Naming Order set to:

- IP Address
- Naming Service
- SysName

Pingable model will be created with IP Address as the models name.

Model Naming configuration settings:
Use Fully Qualified Host Name = No
Model Naming Order set to:

- Naming Service
- SysName
- IP Address

Pingable model will be created with 'device' as the models name.

Model Naming configuration settings:
Use Fully Qualified Host Name = Yes
Model Naming Order set to:

- Naming Service
- IP Address
- SysName

Pingable model will be created with 'device.company.com' as the models name.

If you have already discovered the Pingable models prior to properly configuring the Model Naming configuration options and they need to be changed, we are left with two options:

1. Delete and rediscover, through a Discovery Configuration in the Auto Discovery UI, the Pingable models
2. Manually rename the Pingable models

SNMPv3 Support

> **NOTE**
> SNMPv3 standards require a unique engineID for each SNMP entity (or engine). The SNMP engine/application must have its own unique engineID whether it is a manager or an agent. RFC 3414 and RFC 3418 are the official SNMPv3 standards. See the IETF website (http://www.ietf.org/rfc.html) for more information.

SNMPv3 support includes the following:

- Authentication
- Privacy
- 64-Bit Counters

DX NetOps Spectrummodels and concurrently manages devices that support SNMPv1, SNMPv2c, and SNMPv3.

Support for Diffie-Hellman (DH) Profile on SNMP v3
10.4.1 supports the creation of DH profiles on SNMPv3. This ability provides more robust security mechanism during communication. For more information about how to create a DH profile, see the SNMP v3 Profiles Dialog page.

Secure Domain option in the SNMP v3 profile creation

10.3.1 introduces support for a Secure Domain option in the SNMPv3 profile creation dialog. This feature will ensure privacy and security by restricting v3 profile to the particular SDC specified in a Secure Domain option and preventing users from viewing device profiles belonging to other users. Users have to specify the IP address and configure the secure domain for their devices. For more information about the enhanced support for SNMPV3 Profile, refer to the SNMP v3 Profiles Dialog page.

SNMPv3 Authentication

Starting from 10.3, DX NetOps Spectrum allows '/' and ':' in the snmpv3 username, authentication password, and privacy passwords.

SNMPv3 provides the following levels of security: non-authenticated, authenticated, and authenticated with privacy. Authentication in SNMPv3 uses an algorithm to determine if a message is from a valid source. DX NetOps Spectrum supports the SNMPv3 standard for the authentication of messages. You specify an authentication password for a device model when you create it.

When an SNMP packet is converted to SNMPv3, security parameters are added to the SNMPv3 packet that is sent to the device. The SNMPv3 agent on the device checks the authenticity of the message to verify that the packet came from an authorized source.

SNMPv3 data sent from the device to DX NetOps Spectrum also uses similar security parameters. DX NetOps Spectrum receives the packet and verifies its authenticity.

DX NetOps Spectrum supports the following algorithms for authentication:

- MD5 (Message Digest Algorithm): Produces a 128-bit (16 bytes) message digest. This algorithm is the default. You can model a device configured to use MD5, using ‘Authentication with no Privacy’ or ‘Authentication with Privacy.’
- SHA (Secure Hash Algorithm): Produces a 160-bit (20 bytes) message digest.
- SHA256: Produces a 256-bit (32 bytes) message digest.
- SHA512: Produces a 512-bit (64 bytes) message digest.

Enable SNMPv3 Privacy

Privacy in SNMPv3 uses an encryption algorithm to encode the contents of an SNMPv3 packet to verify that it cannot be viewed by unauthorized entities when routed over the network. DX NetOps Spectrum supports the SNMPv3 standard for the encryption of messages. You specify a privacy password for a device model when you create it.

If configured properly, the SNMPv3 message is sent by DX NetOps Spectrum using the password to encrypt the message before it goes out onto the network. The destination device decrypts the data when it receives it. The return data sent from the device toDX NetOps Spectrumis also encrypted.

DX NetOps Spectrum supports the following encryption algorithms for privacy:

- DES: Data Encryption Standard (DES) is a 64-bit standard that encrypts and decrypts data.
- 3DES: Data Encryption Standard (DES) is a 64-bit standard that encrypts and decrypts data three times.
- AES: Advanced Encryption Standard (AES) is a 128-bit standard, cryptographic algorithm that encrypts and decrypts data.
- AES256: Advanced Encryption Standard (AES 256) is a 256-bit standard, cryptographic algorithm that encrypts and decrypts data.

Follow these steps:

1. In the Topology tab of the Contents panel, click Creates a new model by IP

   The Create Model by IP Address dialog opens.

2. Complete the fields as appropriate.
   - **Network Address**
     Specifies the IPv4 or IPv6 address for the device you want to model.
   - **DCM Timeout (ms)**
     Specifies the timeout between retry attempts (in milliseconds).
     Default: 3000 milliseconds (3 seconds)
   - **DCM Retry Count**
     Type the number of times that the DCM should attempt to send a request to a device that is not responding.
   - **Agent Port**
     Specifies the SNMP agent port.
     Default: 161
3. Select the SNMP v3 option in the SNMP Communications Options section.
   The SNMP Community String field becomes disabled.
4. Click Profiles to create a new SNMPv3 security profile.
   The Edit SNMP v3 Profiles dialog opens.

   NOTE
   The Edit SNMP v3 Profiles dialog is also accessible by clicking Profiles in the Configuration tab in the Discovery Console. For more information, refer to Edit SNMPv3 Profiles dialog.

64-Bit Counters

The SNMPv3 standard provides support for 64-bit counters. DX NetOps Spectrum can access 64-bit counter MIB variables for all SNMPv3 devices that comply with this standard.

SNMPv3 Support Issues

The following are some issues related to SNMPv3 support.

- **get-bulk Command**
  DX NetOps Spectrum support of SNMPv3 does not include the get-bulk command.
- **View Access Control Model (VACM)**
  DX NetOps Spectrum supports the VACM features of SNMPv3, however, VACM is not recommended. DX NetOps Spectrum has features that allow for secure access to devices. If you give DX NetOps Spectrum full view access to all device MIBs, you receive effective monitoring and management performance.
- **Performance and Capacity**
  High processing resources are required for DX NetOps Spectrum to effectively manage SNMPv3 devices. More overhead is consumed using the Authentication and Privacy features due to the time it takes to decrypt and authenticate each message.
  This affects the number of device models that a SpectroSERVER can manage.
- **SNMPv3 Security User Names on SpectroSERVER**
  You cannot use the same user name more than once for the three levels of SNMPv3 (non-authenticated, authenticated, and authenticated with privacy). For example, if you are using the user name “user1” for SNMPv3 level 1 non-authenticated, you cannot use that same user name for SNMPv3 level 2 authenticated or for SNMPv3 level 3 authenticated with privacy.

Edit SNMPv3 Profiles Dialog

The **Edit SNMP v3 Profiles** dialog can be accessed by clicking **Profiles** in the **Configuration** tab in the **Discovery Console** or from the **Create Model** dialogs.

NOTE

For more information on SNMPv3 privacy and authentication options, see SNMPv3 Support.

Configuring the SNMPv3 Profile

1. To add/edit the SNMPv3 profile, do one of the following:
   - From the OneClick Console > Explorer view, right-click **Utilities > Discovery Console** > navigate to the **Configuration** tab > **SNMP Information** section, the select **SNMP v3** option, and click the **Profiles** button.
   - From the OneClick Console > **Contents > Topology** tab, click the create a new model by IP icon  . Select the **SNMPv3** option and click the **Profiles** button.
2. Select the existing profile and click **Modify** to modify a profile, or click **Add** to add a profile.

   NOTE
   This procedure is for non-Diffie-Hellman (DH) profiles creation. For more information about how to create a DH profile on SNMPv3, see the separate section "Support for Diffie-Hellman (DH) Profile on SNMPv3" explained in this article.
3. Enter a name in the **Profile Name** field. This profile name should be unique, for example, for a multitenant configuration, the profile can be <tenantname>_profilename, and in a non-multitenant environment, the profile name can be <SDCIP>_profilename.
4. Enter the same data that has been configured for full MIB access on the device in the **User ID** field.
5. Choose *one* of the following SNMPv3 standard security options from the **Authentication Type** drop-down list:
   - **No Authentication:** Data sent from the DX NetOps Spectrum host system to the SNMPv3 device is not encrypted or authenticated.
   - **Authentication with no Privacy:** Data sent from the DX NetOps Spectrum host system to the SNMPv3 device is authenticated but it is not encrypted.
   - **Authentication with Privacy:** Data sent from the DX NetOps Spectrum host system to the SNMPv3 device is both encrypted and authenticated.

     NOTE
     By default, the MD5 encryption mode option is selected in the **Authentication Protocol** field. If you do not want to use the default option, you can select the SHA, SHA256, or SHA512 encryption option, as appropriate.
6. Select one of the options in the **Authentication Protocol** field:
   - **MD5**
   - **SHA**
   - **SHA256**
   - **SHA512**

     NOTE
     SHA256 and SHA512 are the newly added options in the 10.4.2 release. Ensure that SNMPv3 is enabled on the target device. Also, ensure that the SNMPv3 (SHA256/SHA512) usernames and passwords are created on the target device.
7. Enter a relevant password in the **Authentication Password** field.
8. Re-enter to confirm the password in the **Confirm Authentication Password** field.

   NOTE
   By default, DES authentication option is selected in the Privacy Protocol field. You can select one of the following privacy encryption algorithm options.
9. Select one of the following options in the **Privacy Protocol** field:

- **DES**
- **3DES**
- **AES**
- **AES256**

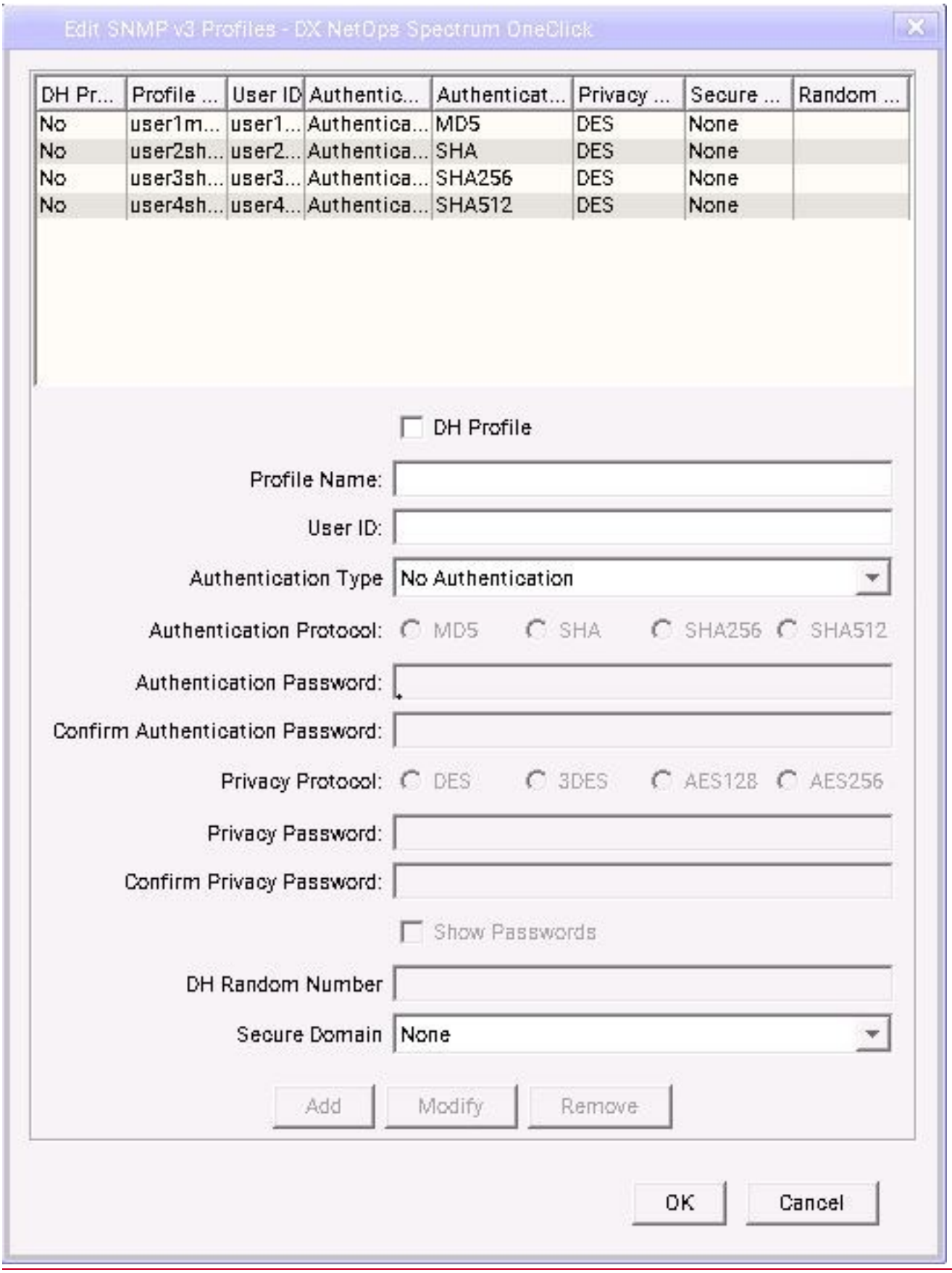
10. In the **Privacy Password** field, enter the same data that has been configured for a full MIB access on the device.
11. Re-enter to confirm the privacy password in the **Confirm Privacy Password** fields.
12. Click **Add/Modify** to update the profiles list with the new/updated profile you have created.

    NOTE
    10.3.1 introduces support for a secure domain option in the SNMP v3 profile creation dialog. This feature will ensure privacy and security by restricting v3 profile to the particular SDC specified in a secure domain option and preventing users from viewing device profiles belonging to other users. Users have to specify the IP address and configure the secure domain for their devices.
13. Click **OK** to save your changes and close the **Edit SNMP v3 Profiles** dialog.

    NOTE
    Select the **Show Passwords** checkbox to view the authentication password and the privacy password entered for the selected profile.

    WARNING
    If you modify the **User ID** field in the **Edit SNMP v3 Profiles** dialog after your model has connected, you will lose contact with the SNMPv3 device. To regain management of the device, right-click the device model in the **Topology** tab of the **Contents** pane, and click **Reconfiguration, Reset SNMPv3 Authentication**.

The following is an example screenshot:


Edit SNMP v3 Profiles - DX NetOps Spectrum OneClick
DH Pr...
Profile ...
User ID
Authentic...
Authenticat...
Privacy ...
Secure ...
Random ...
No user1m... user1... Authentica... MD5 DES None
No user2sh... user2... Authentica... SHA DES None
No user3sh... user3... Authentica... SHA256 DES None
No user4sh... user4... Authentica... SHA512 DES None
DH Profile
Profile Name:
User ID:
Authentication Type
No Authentication
Authentication Protocol:
MD5
SHA
SHA256
SHA512
Authentication Password:
Confirm Authentication Password:
Privacy Protocol:
DES
3DES
AES128
AES256
Privacy Password:
Confirm Privacy Password:
Show Passwords
DH Random Number
Secure Domain
None
Add
Modify
Remove
OK
Cancel

Support for Diffie-Hellman (DH) Profile on SNMP v3

DX NetOps Spectrum now supports the creation of DH profiles on SNMPv3. This ability provides more robust security mechanism during communication.

Create DH Profiles

To create a DH profile, enable the required option and then provide the relevant information. By default, the option to create a DH profile is not selected.

NOTE
Unmanaged traps are not supported on DH profiles.

Follow these steps:

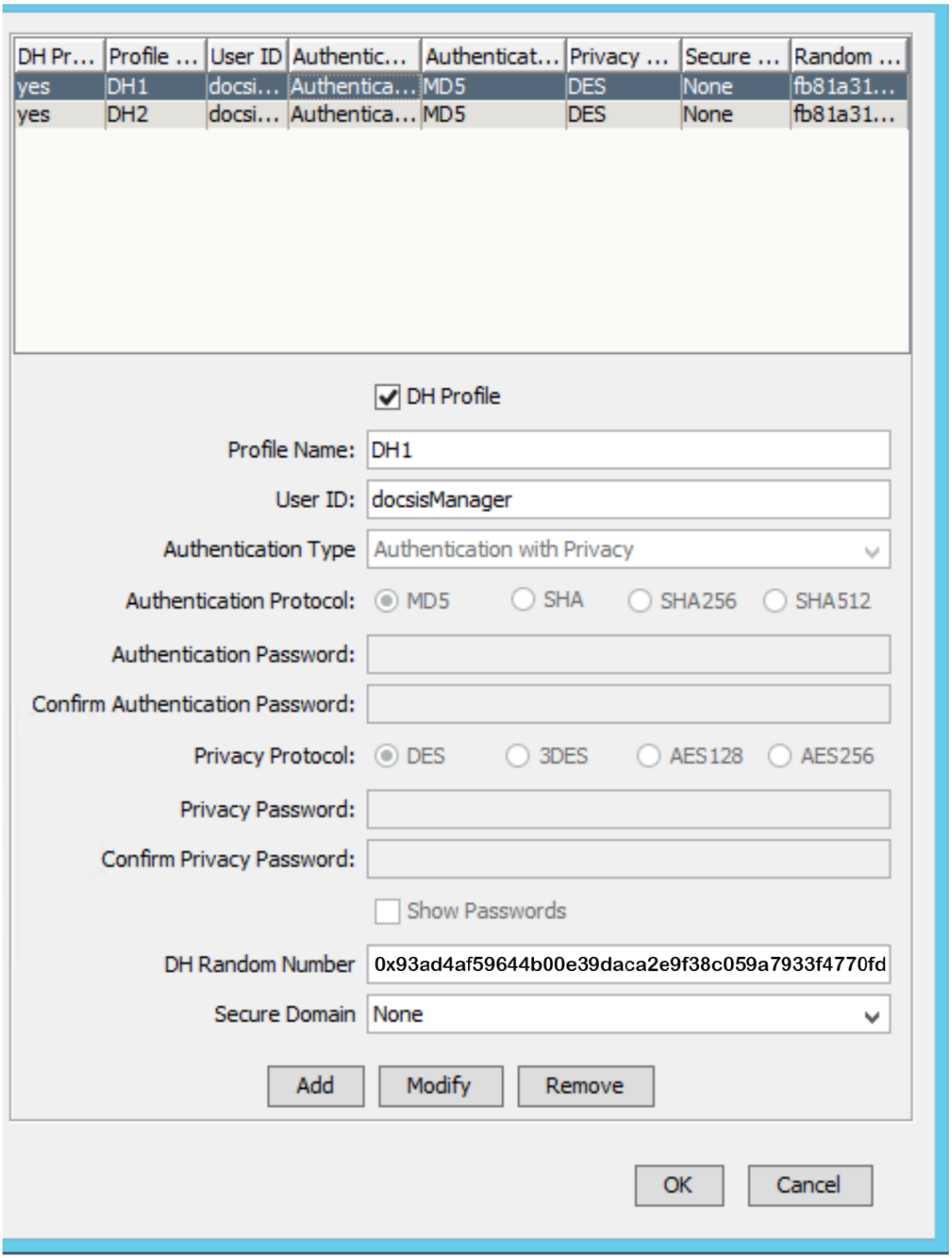
1. In the OneClick Console, click the **Explorer** tab.
2. Right-click in the left pane and select the **Utilities, Discovery Console** option from the context menu.
3. Navigate to the **Configuration** tab, **SNMP Information** section.
4. Select the **SNMP v3** option and click the **Profiles** button.

   NOTE
   When the DH Profile option is enabled, the following fields are disabled and their values are changed automatically:

   - **Authentication Type:** The value is changed to *Authentication with Privacy*.
   - **Authentication Protocol:** The value is changed to *MD5*.
   - **Privacy Protocol:** The value is changed to *DES*.
5. Enable the **DH Profile** option. When this option is enabled, only the following fields are available for entering the information:
   - **Profile Name**
   - **User ID**
   - **DH Random Number**

     NOTE
     The DH random number value must be 256 bits and must start with 0x. An example value is as follows:
     `0x93ad4af59644b00e39daca2e9f38c059a7933f4770fdb648a7e3bcc9c7959c2804cd85f3b4f8a05d70`
   - **Secure Domain**
6. Click **Add** to add the profile
7. Click **OK**.

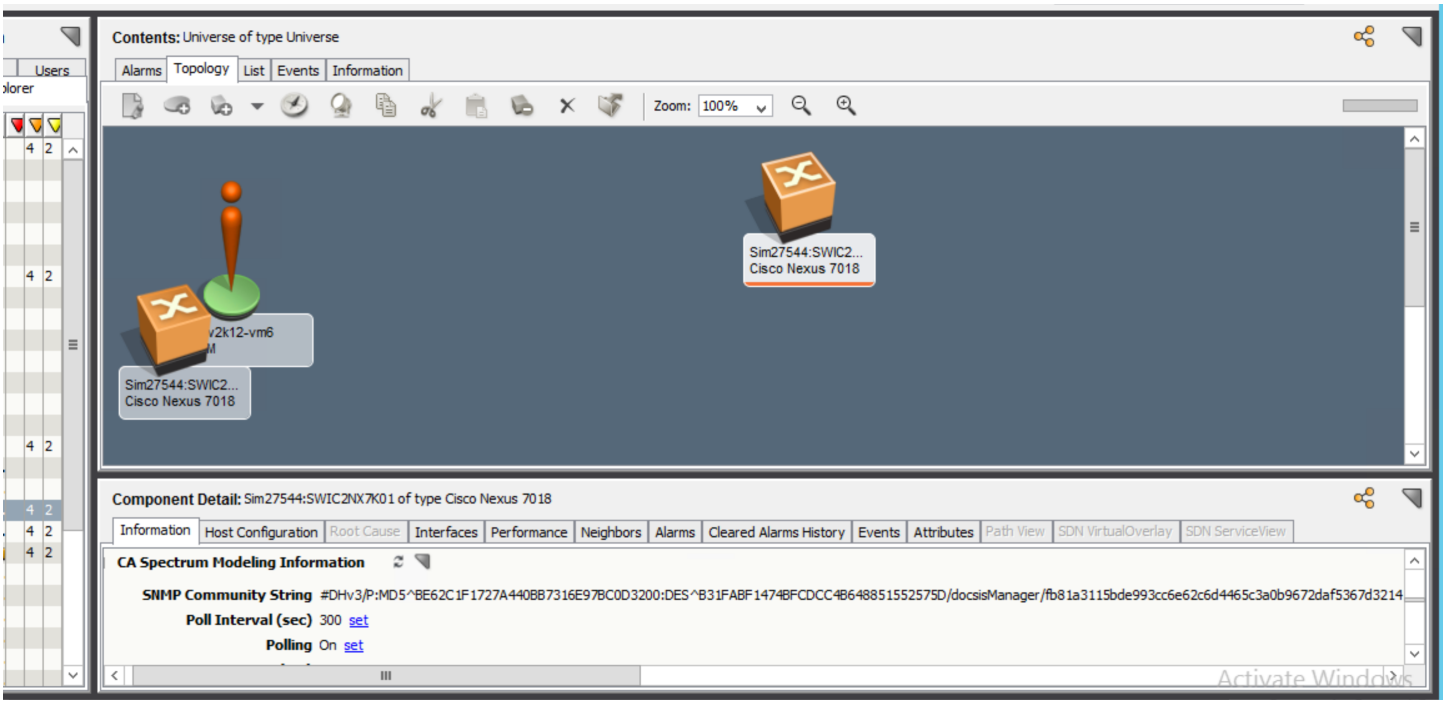
The DH profile is successfully created on SNMP v3. The following 10.4.1 screenshot shows a created DH profile:



After a DH profile device is modeled, the DH SNMPv3 community string will include the following parameters:

- Protocol type (*DH*)
- Authentication type (*MD5*)
- Authentication key
- Privacy protocol (DES)
- Privacy key
- User name
- DH random number

The following screenshot shows the required information:



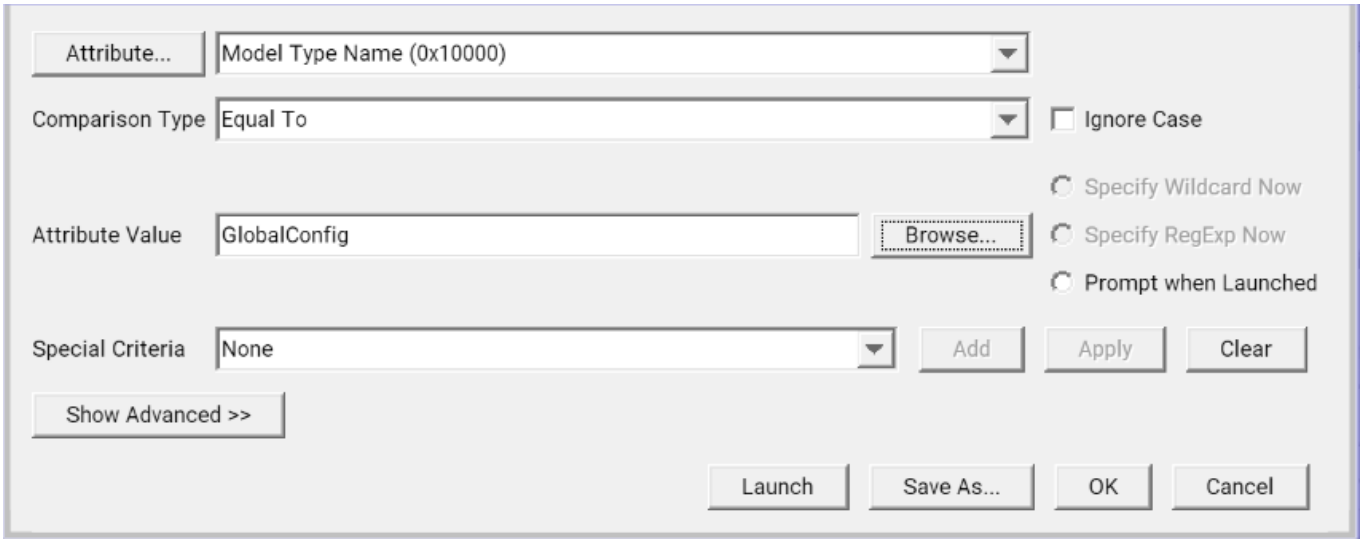
NOTE
If an agent on a device is restarted, then the authentication key and the privacy key will be changed because the public number of the device will get changed. In this case, SpectroSERVER automatically calculates the new authentication key and the privacy key to communicate with the device.

Edit G and P Values on a DH Profile

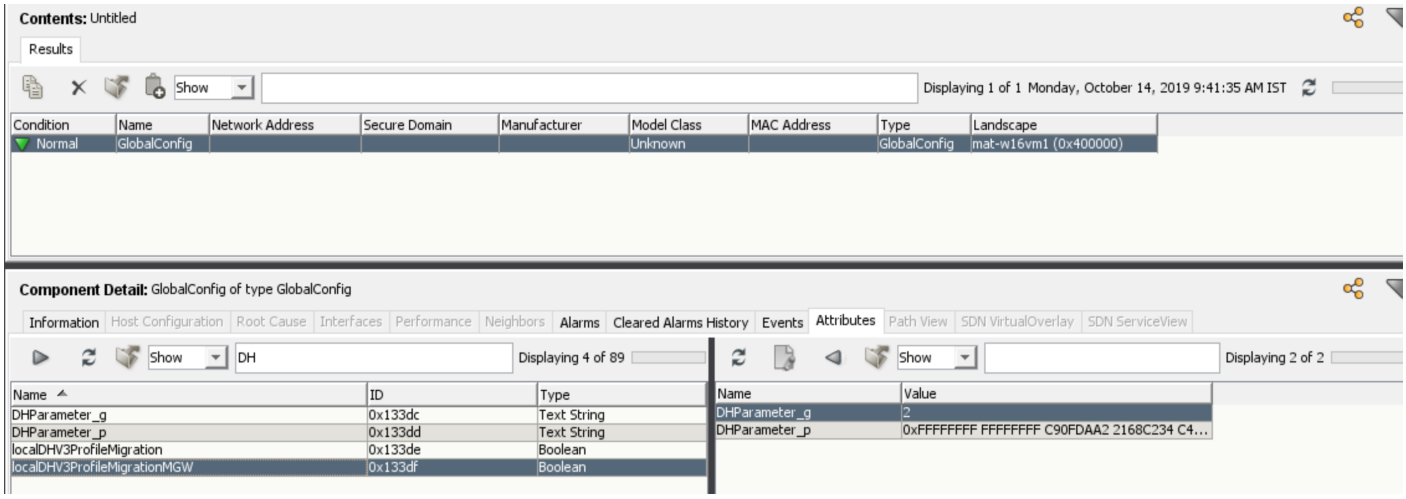
You can edit the G and P values on a DH profile based on your requirements.

Follow these steps:

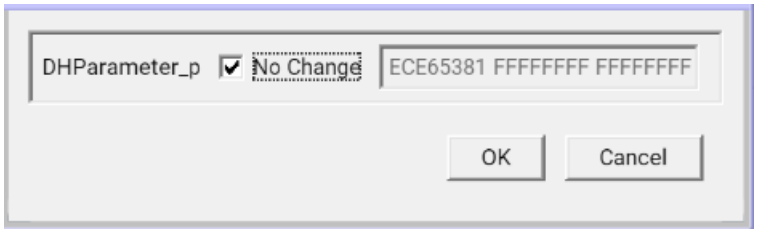
1. Access the OneClick console.
2. Navigate to the **Locater** tab.
3. Click the **Create a new search** icon and enter the information as follows:
   a. Select **Model Type Name (0x10000)** from the **Attribute** drop-down list.
   b. Verify that the value in the **Comparison Type** field is set to **Equal To**.
   c. Select **GlobalConfig** from the **Attribute Value** drop-down list.
   The following screenshot shows the required information:



4. Save the new search.
5. Launch the newly created search.
6. Select the result that is displayed in the **Results** tab in the right pane.
7. Click the **Attributes** tab and search for DHParameter_g and DHParameter_p.
   The parameters are listed in the table. The following screenshot shows the required information:



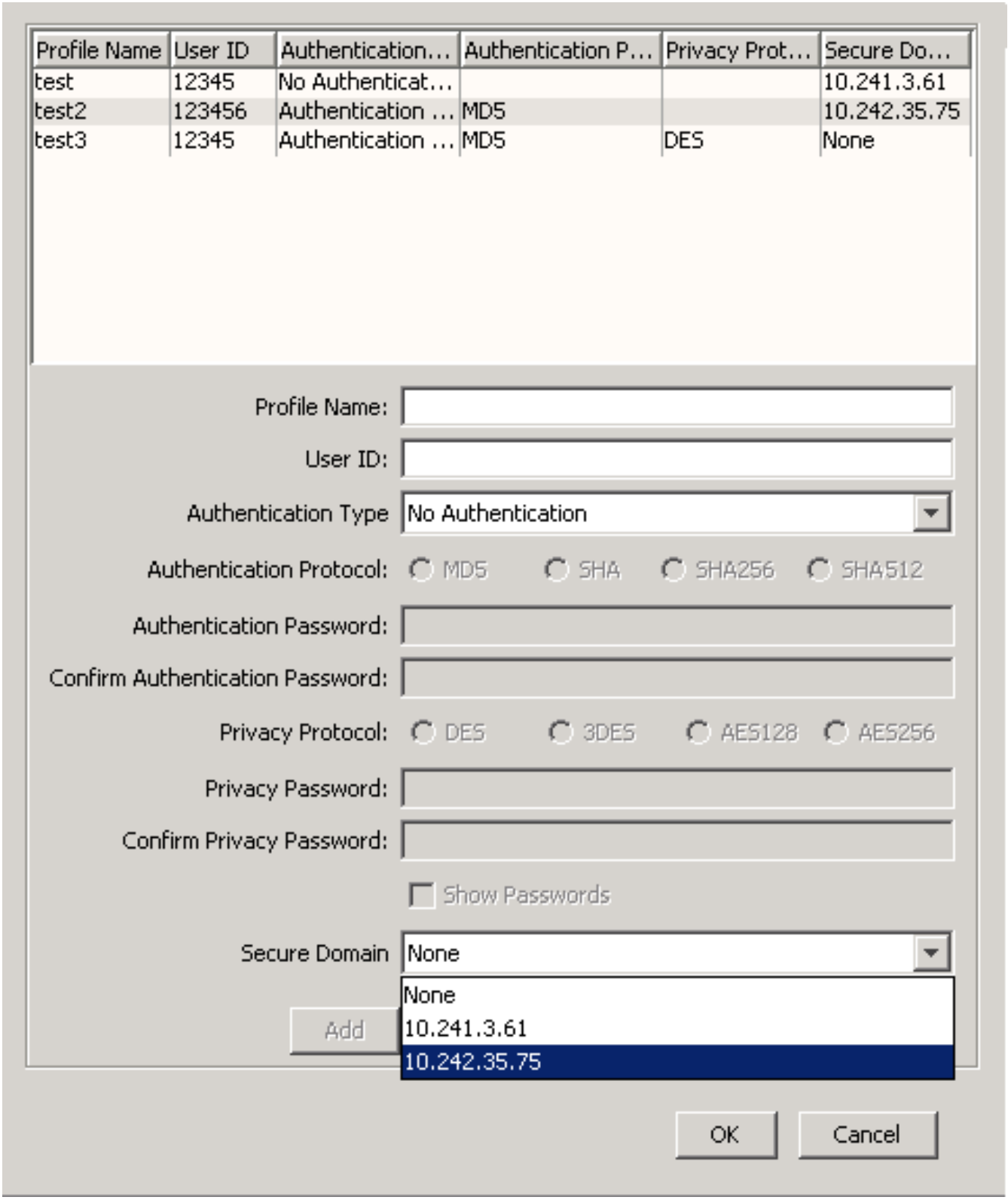
8. Double-click the required parameter (in the right pane) to edit its value and click **OK** to save it. The following screenshot shows the required information:



The values are changed accordingly.

Secure Domain Option

10.3.1 introduces support for a 'Secure Domain' option in the SNMPv3 profile creation dialog. This feature will ensure privacy and security by restricting v3 profile to a particular SDC specified in the **Secure Domain** option and preventing users from viewing device profiles belonging to other users. Users have to specify the IP address and configure the secure domain for their devices. Here is a 10.3.1 screenshot of the SNMPv3 profile creation dialog with the newly added **Secure Domain** option.



To create a v3 profile, specify the IP address of a secure domain along with the other v3 information. If you select **None**, then the corresponding profile pushes all SDCs connected to the landscape. Otherwise, the profile will be unicasted to

a secure domain mentioned in the v3 profile. To model a device with SNMPv3, the selected secure domain option in the CreateModelByIP/ Discovery Console panel should be the same as the secure domain specified in v3 profile.

NOTE

If the selected v3 profile has a secure domain option **None**, then the device can be modeled through any of the selected SDC.

If the SNMPV3 profiles are present in a prior version of the product, then after upgrading to 10.3.1, all these profiles are updated with the **None** option in the **Secure Domain** field. These profiles are broadcasted to all the connected SDCs.

Dump and Reset v3 Profiles at SDC

To dump local and remote profiles at SDC, 0x10337 action can be used on the SDC model handle. Profile details are dumped at “snmpv3profiledump.txt” in the SDConnector/bin folder under SDC. Similarly, to reset all remote profiles at SDC, 0x10336 action can be used.

Troubleshooting

- For unmanaged v3 trap processing, if the trap destination is SDC, then there must be a local profile created at SDC.

Manually Model an SNMPv3 Device

You can manually model SNMPv3 devices and set up new profiles using the Create Model by IP functionality in DX NetOps Spectrum. You cannot model SNMPv3 devices using the Model by Model Type feature in DX NetOps Spectrum.

NOTE
When you discover SNMPv3 devices on Cisco switches with VLANs, you cannot use the community_string@VLAN_ID format to index bridging information for each VLAN. You must create contexts instead. For more information, see the Cisco Device Management section.

Follow these steps:

1. In the Topology tab of the Contents panel, click the Creates a new model by IP icon 
   The Create Model by IP Address dialog opens.
2. Complete the fields as appropriate.
   - **Network Address**
     Specifies the IPv4 or IPv6 address for the device you want to model.
   - **DCM Timeout (ms)**
     Specifies the timeout between retry attempts (in milliseconds).
     Default: 3000 milliseconds (3 seconds)
   - **DCM Retry Count**
     Specifies the number of times that the DCM should attempt to send a request to a device that is not responding.
   - **Agent Port**
     Specifies the SNMP agent port.
     Default: 161
3. Select the SNMP v3 option in the SNMP Communications Options section.
   The SNMP Community String field becomes disabled.
4. Take *one* of the following steps:
   - Select an existing profile from the V3 Profile drop-down list and go to Step 6.
   - Click Profiles to create a new SNMPv3 security profile.

The Edit SNMP v3 Profiles dialog opens.

a. Enter a name in the Profile Name field.
b. Enter the same data that has been configured for full MIB access on the device in the User ID field.
c. Choose *one* of the following SNMPv3 standard security options from the Authentication Type drop-down list:
   No Authentication: Data sent from the DX NetOps Spectrum host system to the SNMPv3 device is not encrypted or authenticated (go to Step 4e).
   Authentication with no Privacy: Data sent from the DX NetOps Spectrum host system to the SNMPv3 device is authenticated but it is not encrypted. Enter the same data that has been configured for full MIB access on the device in the Authentication Password field. Confirm the password (go to Step 4e).
   Authentication with Privacy: Data sent from the DX NetOps Spectrum host system to the SNMPv3 device is both encrypted and authenticated. In the Authentication and Password fields, enter the same data that has been configured for full MIB access on the device (go to Step 4e).
d. Select one of the following options in the **Authentication Protocol** field:
   MD5 (Message Digest Algorithm): Produces a 128-bit (16 byte) message digest.
   SHA (Secure Hash Algorithm): Produces a 160-bit (20 byte) message digest.
e. Enter a relevant password in the **Authentication Password** field.
f. Re-enter to confirm the password in the **Confirm Authentication Password** field.
g. Select one of the following options in the **Privacy Protocol** field:
   - **DES**
   - **3DES**
   - **AES**
   - **AES256**
h. In the **Privacy Password** field, enter the same data that has been configured for full MIB access on the device.
i. Re-enter to confirm the Privacy password in the **Confirm Privacy Password** fields.
j. Click Add to update the Profiles list with the new profile you have created.
k. Click OK to save your changes and close the Edit SNMP v3 Profiles dialog.

5. Select the Discover Connections check box, if appropriate.
6. Click OK in the Create Model By IP Address dialog to accept your selections and close the dialog.
   The model of the SNMPv3 device appears in the Topology tab.

WARNING

If you modify the User ID field in the Edit SNMP v3 Profiles dialog after your model has connected, you will lose contact with the SNMPv3 device. To regain management of the device, right-click the device model in the Topology tab of the Contents panel, and click Reconfiguration, Reset SNMPv3 Authentication.

After you model SNMPv3 devices, you can model the rest of the network using Discovery. Discovery does not overwrite any of the devices that you have already modeled.

Modeling an SNMPv3 Device Using a DX NetOps Spectrum Toolkit

You can use one of the DX NetOps Spectrum toolkits, for example, Modeling Gateway, to create a device model that supports SNMPv3. By default, MD5 is the authentication algorithm that is used and DES is the privacy algorithm that is used. The algorithms can be overridden by the ‘^’ character. Use the following syntax when specifying the SNMP community string for the model.

NOTE

When you want to set the community string with username, authentication and privacy passwords having ':' and '/' following should be used.

Set it using escape(\) for '/' and ':'.
For example if username is Test/user:1, it should be provided as Test\/user\:1 in the community string. Ensure that the profile for Test/user:1 should be already created and existing.

For an SNMP community string that uses both privacy and authentication, use the following syntax:

```
#v3/P:authpassword:privpassword/userid
```

- **authpassword**
  Specifies the authentication password for the device.
- **privpassword**
  Specifies the privacy password for the device.
- **userid**
  Specifies the user ID for the device.

Example 1

```
#v3/P:myAuthPW:myPrivPW/myUserID
```

For an SNMP community string that uses a non-default privacy algorithm (3DES) and a default authentication algorithm, use the following syntax:

```
#v3/P:authpassword:3DES^privpassword/userid
```

- **authpassword**
  Specifies the authentication password for the device.
- **privpassword**
  Specifies the privacy password for the device.
- **userid**
  Specifies the user ID for the device.

Example 2

```
#v3/P:myAuthPW:3DES^myPrivPW/myUserID
```

For an SNMP community string that uses a non-default privacy algorithm (3DES) and a non-default authentication algorithm (SHA), use the following syntax:

```
#v3/P:SHA^authpassword:3DES^privpassword/userid
```

- **authpassword**
  Specifies the authentication password for the device.
- **privpassword**
  Specifies the privacy password for the device.
- **userid**
  Specifies the user ID for the device.

Example 3

```
#v3/P:SHA^myAuthPW:3DES^myPrivPW/myUserID
```

For an SNMP community string that uses authentication only, use the following syntax:

```
#v3/A:authpassword/userid
```

- **authpassword**
  Specifies the authentication password for the device.
- **userid**
  Specifies the user ID for the device.

Example 4

```
#v3/A:myAuthPW/myUserID
```

For an SNMP community string that uses a non-default authentication algorithm (SHA) and no privacy, use the following syntax:

```
#v3/A:SHA^authpassword/userid
```

- **authpassword**
  Specifies the authentication password for the device.
- **userid**
  Specifies the user ID for the device.

Example 5

```
#v3/A:SHA^myAuthPW/myUserID
```

For an SNMP community string that does not use authentication or privacy, use the following syntax:

```
#v3/N/userid
```

- **userid**
  Specifies the user ID for the device.

Example 6

```
#v3/N/myUserID
```

Model an SNMP v2c Device Using a DX NetOps Spectrum Toolkit

To use one of the DX NetOps Spectrum toolkits to create a device model that supports SNMP v2c, use the following syntax when specifying the SNMP community string for the model:

```
#v2/<SNMP community string>
```

- ***<SNMP community string>***
  Specifies the SNMP community string of the device.

Example:

```
#v2/mySNMPcommunitystring
```

Change Security Information for a Device Model

You can change security information for an existing SNMPv3 device model. You can also convert an SNMPv1 device model to an SNMPv3 device model. You must add the appropriate security information to the device model.

Follow these steps:

1. Select the model that you want to modify, and click the Information tab in the Component Detail panel.
2. Expand the DX NetOps Spectrum Modeling Information subview, and click set in the SNMP Community String field.
3. Modify the SNMP Community String using a syntax listed in Modeling an SNMPv3 Device Using a DX NetOps Spectrum Toolkit to create the appropriate string.

   NOTE
   For more information about using CLI commands, see the Command Line Interface section.

Add Context Name Information

You can add the SNMPv3 context name value to be sent with SNMPv3 messages for a particular device.

Follow these steps:

1. Select the model that you want to modify, and click the Information tab in the Component Detail panel.

2. Expand the DX NetOps Spectrum Modeling Information subview, and click set in the SNMP Community String field.
3. Add the context name value to the SNMP Community String field. For example, if the current SNMP community string is:
   ```
   #v3/P:authPass:privPass/myuserid
   ```
4. To insert a context name value of ‘quark,’ add ‘quark’ to the SNMP community string as follows:
   ```
   #v3/P:authPass:privPass/quark/myuserid
   ```

Specify an Authentication Encryption Algorithm on a Per-Model Basis

DX NetOps Spectrum supports both MD5 and SHA authentication encryption, although MD5 is the default. You can specify the alternate encryption algorithm (SHA) by prepending it to the password in the SNMP community string. Prefixing the encryption algorithm on the SNMP community string for a particular device model overrides the default algorithm for that device model only.

To specify a privacy encryption algorithm on a per-model basis

1. In the Topology tab of the Contents panel, click the Creates a new model by IP icon 
   The Create Model by IP Address dialog opens.
2. Complete the fields as appropriate.
   - **Network Address**
     Specifies the IPv4 or IPv6 address for the device you want to model.
   - **DCM Timeout (ms)**
     Specifies the timeout between retry attempts (in milliseconds).
     Default: 3000 milliseconds (3 seconds)
   - **DCM Retry Count**
     Type the number of times that the DCM should attempt to send a request to a device that is not responding.
   - **Agent Port**
     Specifies the SNMP agent port.
     Default: 161
3. Select the SNMP v3 option in the SNMP Communications Options section.
   The SNMP Community String field becomes disabled.
4. Click Profiles to create a new SNMPv3 security profile.
   The Edit SNMP v3 Profiles dialog opens.
5. (Optional) To specify the SHA authentication encryption algorithm, do the following:
   a. Enter a name in the Profile Name field.
   b. Enter the same data that has been configured for full MIB access on the device in the User ID field.
   c. Select Authentication with Privacy from the Authentication Type drop-down list.
   d. Enter the following in the Authentication Password and Confirm Authentication Password field:
      ```
      SHA^<authpassword>
      ```
   e. Enter the privacy password in the Privacy Password and Confirm Privacy Password field.
   f. Click Add to update the Profiles list with the new profile you have created.
   g. Click OK to save your changes and close the Edit SNMP v3 Profiles dialog.
6. (Optional) To specify the SHA authentication encryption algorithm and the 3DES, AES-128, or AES-256 privacy encryption algorithm, do the following:
   a. Enter a name in the Profile Name field.
   b. Enter the same data that has been configured for full MIB access on the device in the User ID field.
   c. Select Authentication with Privacy from the Authentication Type drop-down list.
   d. Enter the following in the Authentication Password and Confirm Authentication Password field:

```
SHA^<authpassword>
```

e. Enter the following in the Privacy Password and Confirm Privacy Password fields:

```
[3DES|AES|AES256]^<privpassword>
```

f. Click Add to update the Profiles list with the new profile you have created.

g. Click OK to save your changes and close the Edit SNMP v3 Profiles dialog.

7. Select the Discover Connections check box, if appropriate.
8. Click OK in the Create Model By IP Address dialog to accept your selections and close the dialog.
   The model of the SNMPv3 device appears in the Topology tab. The authentication and privacy encryption algorithms you specified appear in the SNMP Community String field of the Modeling Information subview for the model. You can also specify a privacy encryption algorithm or an authentication encryption algorithm by clicking set in the SNMP Community field of the Modeling Information subview.

NOTE
The Edit SNMP v3 Profiles dialog is also accessible by clicking Profiles in the Configuration tab in the Discovery Console.

Change the Default Authentication Encryption Algorithm For All Device Models

To change the default authentication encryption algorithm for all device models, modify the ".vnmrc" file.

Follow these steps:

1. Navigate to the following directory:
   <$SPECROOT>/SS/
2. Open the ".vnmrc" file with a text editor.
3. Locate the following line:

```
snmpv3_default_auth_protocol=md5
```

4. To modify the algorithm to use SHA as the default, change the parameter as follows:

```
snmpv3_default_auth_protocol=sha
```

   You have successfully changed the default authentication encryption algorithm.

Specify a Privacy Encryption Algorithm on a Per-Model Basis

DX NetOps Spectrum supports DES, 3DES, AES-128, and AES-256 privacy encryption and uses DES by default. You can specify an alternate encryption algorithm by prefixing it to the password in the SNMP community string. Appending the encryption algorithm on the SNMP community string for a particular device model overrides the default algorithm for that device model only.

To specify a privacy encryption algorithm on a per-model basis

1. In the Topology tab of the Contents panel, click the Creates a new model by IP icon 
   The Create Model by IP Address dialog opens.
2. Complete the fields as appropriate.
   - **Network Address**
     Specifies the IPv4 or IPv6 address for the device you want to model.
   - **DCM Timeout (ms)**
     Specifies the timeout between retry attempts (in milliseconds).
     Default: 3000 milliseconds (3 seconds)
   - **DCM Retry Count**
     Type the number of times that the DCM should attempt to send a request to a device that is not responding.
   - **Agent Port**

Specifies the SNMP agent port.
Default: 161

3. Select the SNMP v3 option in the SNMP Communications Options section.
   The SNMP Community String field becomes disabled.
4. Click Profiles to create a new SNMPv3 security profile.
   The Edit SNMP v3 Profiles dialog opens.
5. To specify the 3DES, AES-128, or AES-256 privacy encryption algorithm, do the following:
   a. Enter a name in the Profile Name field.
   b. Enter the same data that has been configured for full MIB access on the device in the User ID field.
   c. Select Authentication with Privacy from the Authentication Type drop-down list.
   d. Enter the authentication password in the Authentication Password and Confirm Authentication Password field.
   e. Enter the following in the Privacy Password and Confirm Privacy Password fields:
      `[3DES|AES|AES256]^<privpassword>`
   f. Click Add to update the Profiles list with the new profile you have created.
   g. Click OK to save your changes and close the Edit SNMP v3 Profiles dialog.
6. Select the Discover Connections check box, if appropriate.
7. Click OK in the Create Model By IP Address dialog to accept your selections and close the dialog.
   The model of the SNMPv3 device appears in the Topology tab. The privacy encryption algorithm you specified appears in the SNMP Community String field of the Modeling Information subview for the model. You can also specify a privacy encryption algorithm by clicking set in the SNMP Community field of the Modeling Information subview.

NOTE
The Edit SNMP v3 Profiles dialog is also accessible by clicking Profiles in the Configuration tab in the Discovery Console.

Change the Default Privacy Encryption Algorithm For All Device Models

To change the default privacy encryption algorithm for all device models, you must modify the ".vnmrc" file.

To change the default privacy encryption algorithm for all device models

1. Go to the following directory:
   `<$SPECROOT>/SS/`
2. Open the ".vnmrc" file with a text editor and locate the following line:
   `snmpv3_default_priv_protocol=des`
3. Depending on the privacy encryption algorithm you want to set as the default, modify the parameter as follows:
   ```
   snmpv3_default_priv_protocol=3des
   snmpv3_default_priv_protocol=aes (uses AES 128 encryption)
   snmpv3_default_priv_protocol=aes256 (uses AES 256 encryption)
   ```

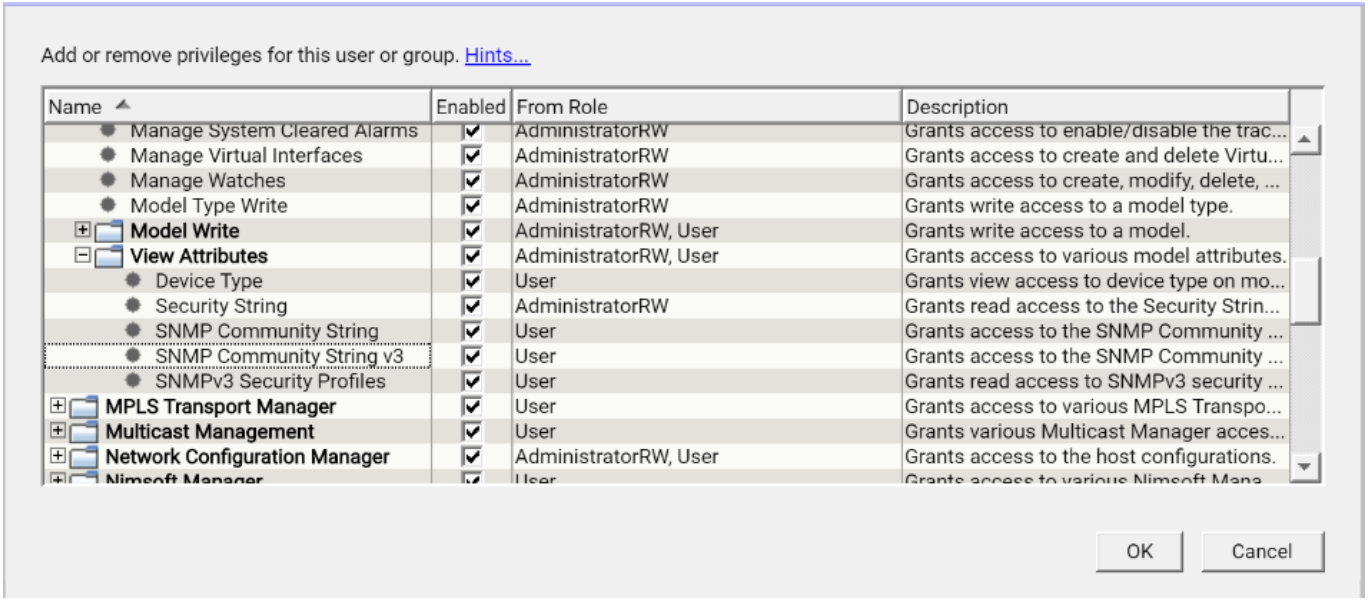
SNMPv3 Community String Access Control

You can control the access to the SNMPv3 community string information on models. You can decide whether you want to display or hide the SNMP v3 community string to specific users. This functionality, therefore, gives you the ability to restrict the display of this information, making it available only to those users who are supposed to view the information.

NOTE
By default, the **SNMPv3 Community String v3** option is enabled. However, if any customization has been done on the options under the **Privileges** section on the lower version of DX NetOps Spectrum for the administrator and operator roles, then after upgrade to 10.4.1, the **SNMPv3 Community String v3** option is displayed as not selected. Furthermore, if no customization has been done, the **SNMPv3 Community String v3** option is displayed as enabled.

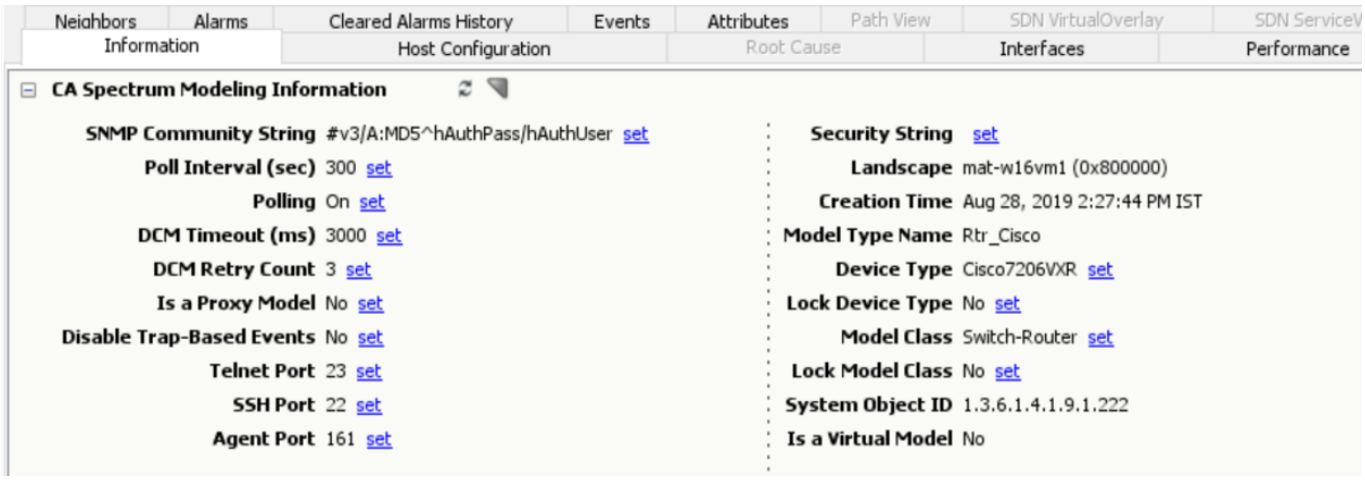
Follow these steps:

1. Access the OneClick interface.
2. Click the **Users** tab in the **Navigation** pane.
3. Locate and select the required user.
4. Click the **Access** tab in the **Contents** pane.
5. Click **Add/Remove** on the **Privileges** tab.
6. Expand **Model Management, View Attributes** to view the options.
   The following screenshot shows the required information:



7. Locate the **SNMPv3 Community String v3** option and perform one of the following actions depending on your requirement:
   – Clear the checkbox.
   This action disables the display of the SNMPv3 community string for the selected user. In this case, the related user cannot see the SNMPv3 community string information; only asterisks (****) are displayed. The following screenshot does not display the community string information:

– Select the checkbox.
This action enables the display of the SNMPv3 community string for the selected user. In this case, the related user can see the information. The following screenshot displays the SNMPv3 Community string information:



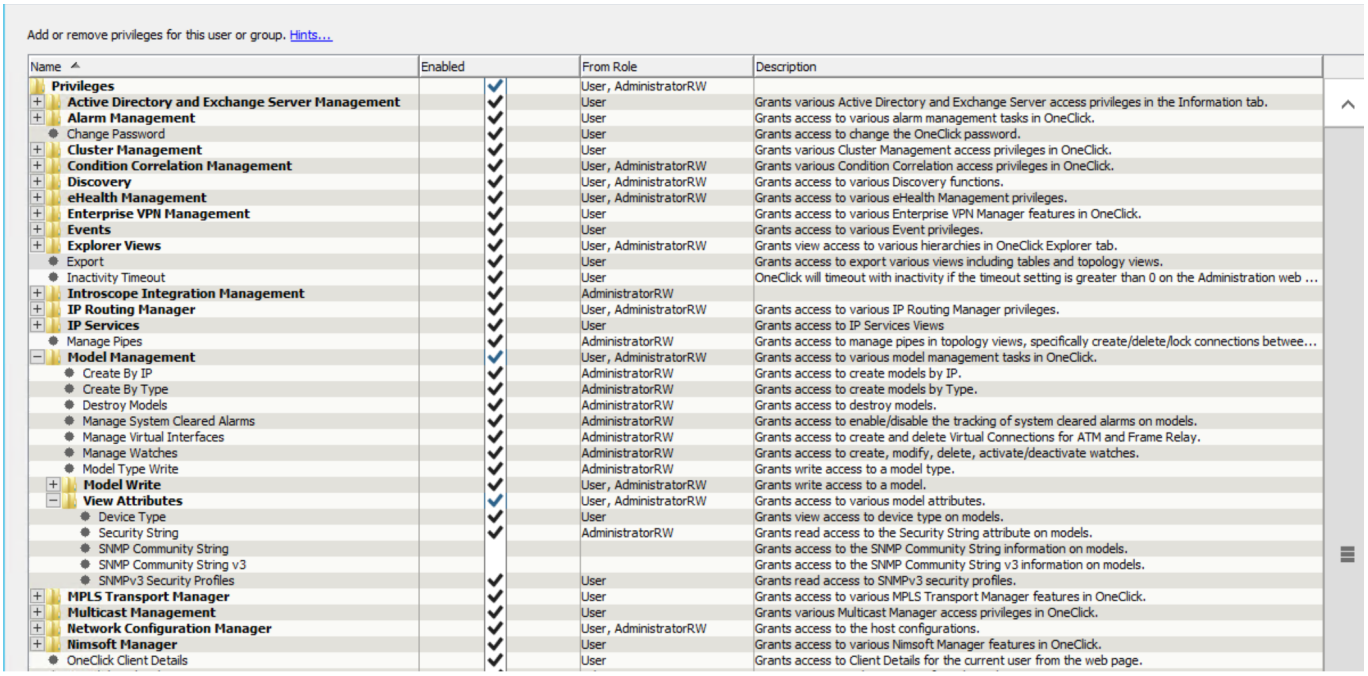
8. Click **OK**.

You have successfully changed the access to the SNMPv3 community string for the required user. Therefore, depending on whether you have enabled or disabled the option, the SNMPv3 community string information is displayed or hidden, as appropriate.

Consideration

If the **SNMP Community String v3** option is disabled (not selected) at the time of upgrading to 10.4.1, then after the upgrade, both **SNMP Community String v3** and **SNMP Community String** will be disabled.

The following screenshot shows the required information. Note that both **SNMP Community String v3** and **SNMP Community String** are not selected:



Troubleshoot SNMPv3 Communication Issues

An error message or alarm is displayed if DX NetOps Spectrum cannot communicate with an SNMPv3 device.

Consider the following:

Is the Device Model's Security Information Correct?

If you changed security information for a particular device model (see), and the new information provided does not match the security information on the device, DX NetOps Spectrum generates an alarm indicating that it cannot contact the device using SNMP.

To troubleshoot this problem, update the security information for the device model to match the information on the device.

What Should I Do When DX NetOps Spectrum Loses SNMPv3 Contact with Cisco Routers After They Have Been Rebooted?

DX NetOps Spectrum can lose communication with Cisco devices, such as Cisco router models like 2621 v12.2 (IOS), 2517 v12.0 (IOS), or 2514 v12.2 (IOS).

SNMPv3 support includes a security feature named replay protection, which guards against SNMPv3 packet deciphering activities over the network. Replay protection checks the following two values on a device whenever a SNMP query is initiated:

- snmpEngineBoots: The number of times the device has rebooted.
- snmpEngineTime: The number of seconds since the snmpEngineBoots counter was last incremented.

DX NetOps Spectrum monitors these values for every device. When SNMP communication is properly occurring, DX NetOps Spectrum and a device are in sync with one another. If a device goes down, DX NetOps Spectrum receives the snmpEngineTime with the value of 0. DX NetOps Spectrum compares the snmpEngineBoots value and, if it has incremented, communication resumes. If the snmpEngineBoots value has not incremented, then DX NetOps Spectrum does not resume communication.

This problem is due to a Cisco IOS firmware bug that will not increment the boot count causing SDManager to stop communication.

To avoid this performance problem, upgrade these routers with the latest Cisco IOS firmware. See http://www.cisco.com for details.

> **NOTE**
> For more information about replay protection, see RFC 3414, section 2.2, Replay Protection.

Previously configured snmpv3 profiles are missing in Spectrum 10.2.2, what is the solution?

Previously configured snmpv3 profiles are missing in the 10.2.2 release, during an upgrade. This issue requires a patch on top of the 10.2.2 version, install the patch to fix the issue. The patch details are **10.02.02.PTF_10.2.208**. The salesforce ID for this issue is 00903607.

Product Intelligence Technology and Capabilities

This section provides information about the various DX NetOps Spectrum Intelligence technology and capabilities:

Inductive Modeling Technology

DX NetOps Spectrum comes with Inductive Modeling Technology™ (IMT), a patented technology that consists of a suite of intelligence circuits which work with the VNM to help configure, manage, and monitor your network.

Static Configuration of Device Models

Many network devices are configured during their manufacturing process and are difficult to modify later. For DX NetOps Spectrum modeling purposes, these devices are considered to have a static configuration--once DX NetOps Spectrum intelligence models these devices, they are not reconfigured. DX NetOps Spectrum creates models for the ports, matching the types of port models to the types of ports on the device, such as T1 or Ethernet. For each port model that is created, an association is established between the port and the device using the HASPART relation and may be viewed within the device model's Interfaces tab. When the model is destroyed, all of the port models associated with the device are also destroyed.

> **NOTE**
> For more information about the Interfaces tab, see the Using OneClick section.

Dynamic Configuration of Device Models

Some network devices can be configured dynamically by removing boards and installing new boards without removing the device from the network. DX NetOps Spectrum intelligence provides for automatic modeling of these devices and their connections upon their creation and then performs verification and remodeling if necessary after each VNM polling cycle. Thus, DX NetOps Spectrum continuously monitors and changes these models to match the actual device on the network.

Whenever a model is created, SpectroSERVER polls the device and creates an appropriate configuration, including the number, type, and order of modules and ports. For each device model created, a relation exists between that model and the parent device via the HASPART model type relation rule. This relation also exists between the boards and any ports on the boards. After each polling cycle, SpectroSERVER re-examines and, if necessary, changes the parent model and its related models to match changes to the device configuration.

Whenever you add a new board to a dynamically configured device, DX NetOps Spectrum creates a model to represent that board and each of the ports on the board. If a board model is destroyed, all of the port models that form part of the board are also destroyed.

Pulled Board List

Creating and destroying models is time consuming. When you remove a board from the device, DX NetOps Spectrum does not destroy the board model, but rather keeps a copy of the board model in a "pulled board list." The board model's HASPART relation to the hub model is removed. DX NetOps Spectrum reassociates this model to the device if you reinstall the old board. If you add a new board to replace the old board, DX NetOps Spectrum associates the new board model to the device and the old board model is placed in the Lost and Found view. If you remove a board model from the pulled board list, the board model is removed from the Lost and Found view and no longer exists.

The following are general pulled board list attributes:

- **Max_Pulled_Bd_Cnt**
  Specifies the maximum number of models allowed to exist in the pulled board list. When this value is exceeded, the oldest model is removed from the list.
- **Pulled_Bd_Cnt**
  Specifies the current number of models in the pulled board list.
- **Pulled_Bd_List**
  Contains a list of board models that have been pulled from a dynamically configured device. When a board is reinstalled in the device, DX NetOps Spectrum removes the board model from the Pulled_Bd_List.

  NOTE
  The Pulled_Bd_List is not readable by the user.

Router Reconfiguration Events

Router reconfiguration actually involves two separate processes: interface reconfiguration, which helps ensure the device's interfaces are properly modeled, and device discovery, which helps ensure proper modeling of other devices, LANs, and so on that are connected to those interfaces. Depending on whether both or either one of these processes occurs, DX NetOps Spectrum generates one of the following events to help you keep track of the configuration changes:

- **ROUTER_RECONFIG_EVENT (0x1001c)**
  This event is generated when a device is reconfigured and both interface reconfiguration and device discovery occur.
- **INTERFACE_RECONFIG_EVENT (0x1001d)**
  This event is generated for a device whenever interface reconfiguration occurs.
- **DEVICE_DISCOVERY_EVENT (0x1001e)**
  This event is generated for a device whenever device discovery occurs such as when connections off the device's interfaces are being rediscovered.

Condition Versus Rollup Condition

DX NetOps Spectrum provides intelligence circuits that let you see changes in your network devices and their performance by simply glancing at the icons that represent them. The icons use color to indicate two different types of status: Condition and Rollup Condition. Condition reflects the contact and alarm status of the modeled device represented by the icon. Rollup Condition is the *composite* status of models that are "children" of the model represented by the icon. (Child models are related to parent models through the "collects" relation in the Topology hierarchy and through the "contains" relation in the Location hierarchy.) The Rollup Condition generally changes as you move up in the hierarchy, because at each level it reflects the blending of the Rollup Conditions from a greater number of individual models.

The location of the Condition and Rollup Condition colors varies according to the type of icon. For device and topology (LAN) icons, Condition is displayed in the diagnostic double-click zone and the Rollup Condition is displayed in the down-

arrow double-click zone for the icon. The circle at the base of location model icons displays either the Condition or the Rollup Condition, whichever is more critical.

The Rollup condition gets updated to the parent models in SDN hierarchy based on the composite condition value. The composite condition is calculated based on alarms on the child models and the configured threshold.

Attributes Determining Condition and Rollup Condition

There is a unique set of attributes that are related to Condition and Rollup Condition. Their values are used in determining Condition and Rollup Condition for models:

- **Condition**
  The Condition attribute value reflects the contact status as well as any more specific alarm in effect for a device model. This value determines the Condition color on topology and location icons as explained in the following table:

Contact Status	Condition	Color	Description
Initial	Initial	Blue	Either the model has not yet established contact with the device it represents, or it represents an insignificant device with which contact has been lost.
Established	Normal	Green	The model has successfully established contact with the device it represents, and the device is functioning normally.
Established	Minor	Yellow	This is the first level of marginal operation. Either the model has successfully established contact with the device it represents but there is an abnormal condition that does not affect overall network operation (perhaps a module has been removed from the device), or the IP address assigned to this model was already assigned to another model.
Lost	Major	Orange	This is the second level of marginal operation. The management agent on the device has failed and is not responding to any communication from DX NetOps Spectrum but the device is still relaying data to its downstream neighbors. This condition occurs only on data-relay type devices such as hubs and is typical of a firmware failure.
Lost	Critical	Red	This condition indicates a total failure of the device and requires management's attention to repair or replace it.

Lost	Suppressed (Unknown)	Gray	Contact has been lost with this device *and* with a device that is upstream from this device (for example, between this device and DX NetOps Spectrum), thus the actual condition of this device is unknown and alarms for the model representing it are suppressed. The gray condition color is also displayed for all models that are downstream from this device. All adjacent (directly connected) models, whether upstream or downstream, will have a contact status of “Lost."
Lost	Maintenance	Brown	DX NetOps Spectrum cannot contact the device because the model has been placed into maintenance mode.

- **Condition_Value**
  Specifies a numeric value that represents a model’s overall condition. This value is passed to a parent model and included in the composite condition. The model’s overall condition is either the condition or the rollup condition, whichever is more severe.

  NOTE
  The condition value indirectly receives the value of the administrator-defined significance level corresponding to a model’s overall condition.

- **Composite_Condition**
  The sum of all condition values for models that are contained by a location model or collected by a topology model.

- **Rollup Condition**
  DX NetOps Spectrum computes the rollup condition attribute value using the administrator-defined rollup threshold and composite condition. The resulting attribute value determines the color that is displayed to indicate the overall condition of models that are contained by a location model or collected by a topology model. The possible colors are:
  - **Green**
    The value of the composite condition attribute for this model’s children is less than the yellow (rollup condition) threshold.
  - **Yellow**
    The composite condition value for this model’s children equals or exceeds the yellow threshold but is less than the orange threshold.
  - **Orange**
    The composite condition value for this model’s children equals or exceeds the orange threshold but is less than the red threshold.
  - **Red**
    The composite condition value for this model’s children equals or exceeds the red threshold.

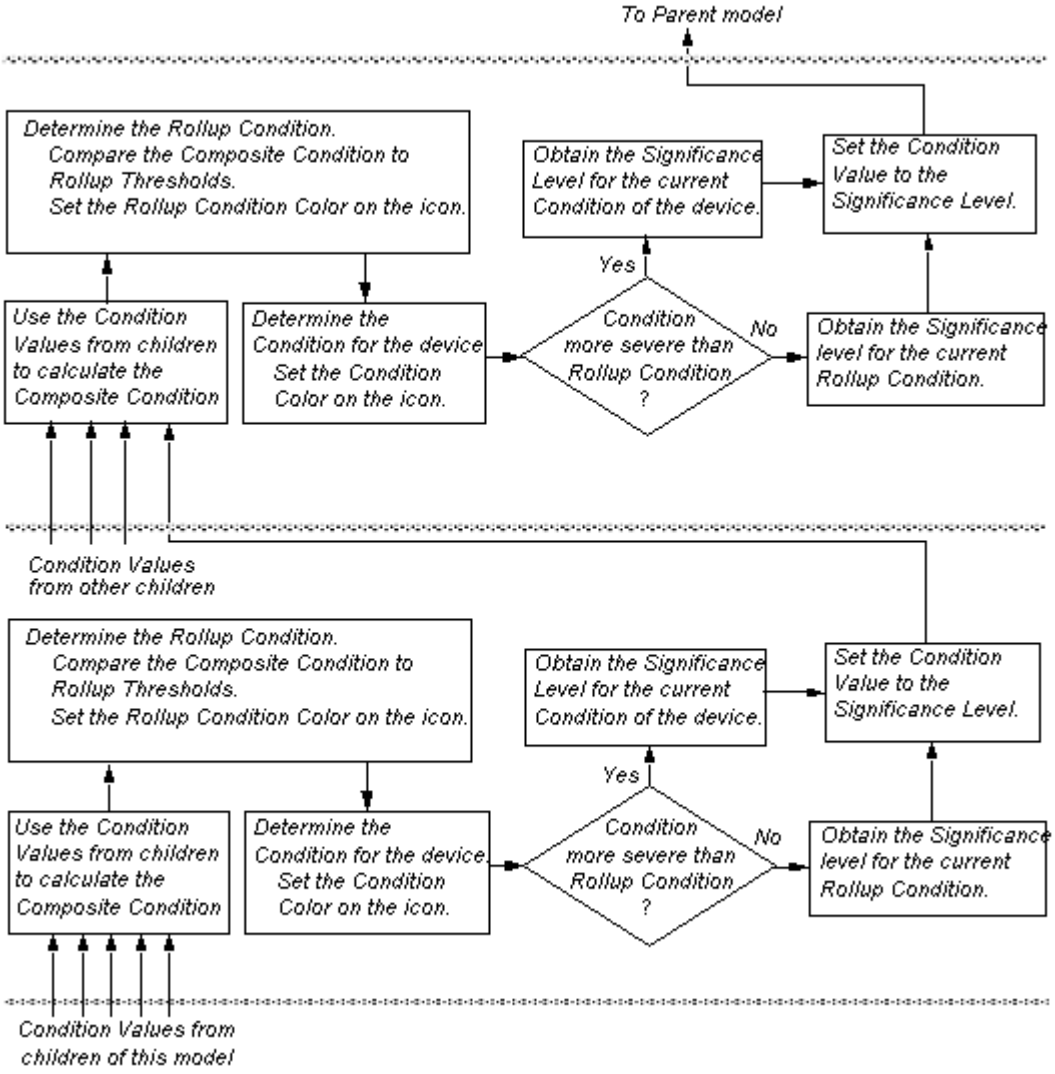
Condition and Rollup Condition Sensitivity

The following two attributes serve as parameters that can be used to emphasize or diminish the impact on rollup condition from the condition values of particular models. By adjusting these attribute values you can control when the rollup condition color changes.

- **Rollup Thresholds**
  The rollup thresholds are the three attributes that control the rollup condition color (yellow, orange, and red) for a model. Rollup thresholds are administrator-defined values that are entered on a model-by-model basis. The composite condition value received from the model's children is compared with these attributes to determine a rollup condition color. For example, if a model's composite condition value is equal to or greater than its orange threshold (but less than its red threshold), the model's rollup condition color is orange.
  The default values for rollup thresholds are:
  – Yellow Threshold = 3
  – Orange Threshold = 6
  – Red Threshold = 10
- **Significance Level**
  The significance level attributes define the numeric value for yellow, orange, and red conditions and rollup conditions. Like rollup thresholds, significance level values are administrator-defined values, and are entered on a model-by-model basis.
  Significance level field labels begin with the words "value when." The default significance level values are:
  – Value_When_Yellow = 1
  – Value_When_Orange = 3
  – Value_When_Red = 7
  Typically, models (devices) are divided into two classes, "significant" or "insignificant." A significant device is any device that requires an administrator's attention for proper network operation. Insignificant devices are typically end-point devices, such as a PC or workstation. Insignificant devices usually toggle between green and blue (Condition Value = 0). Significant models can be made insignificant by changing their Value_When_Red attribute value to 0 (zero).

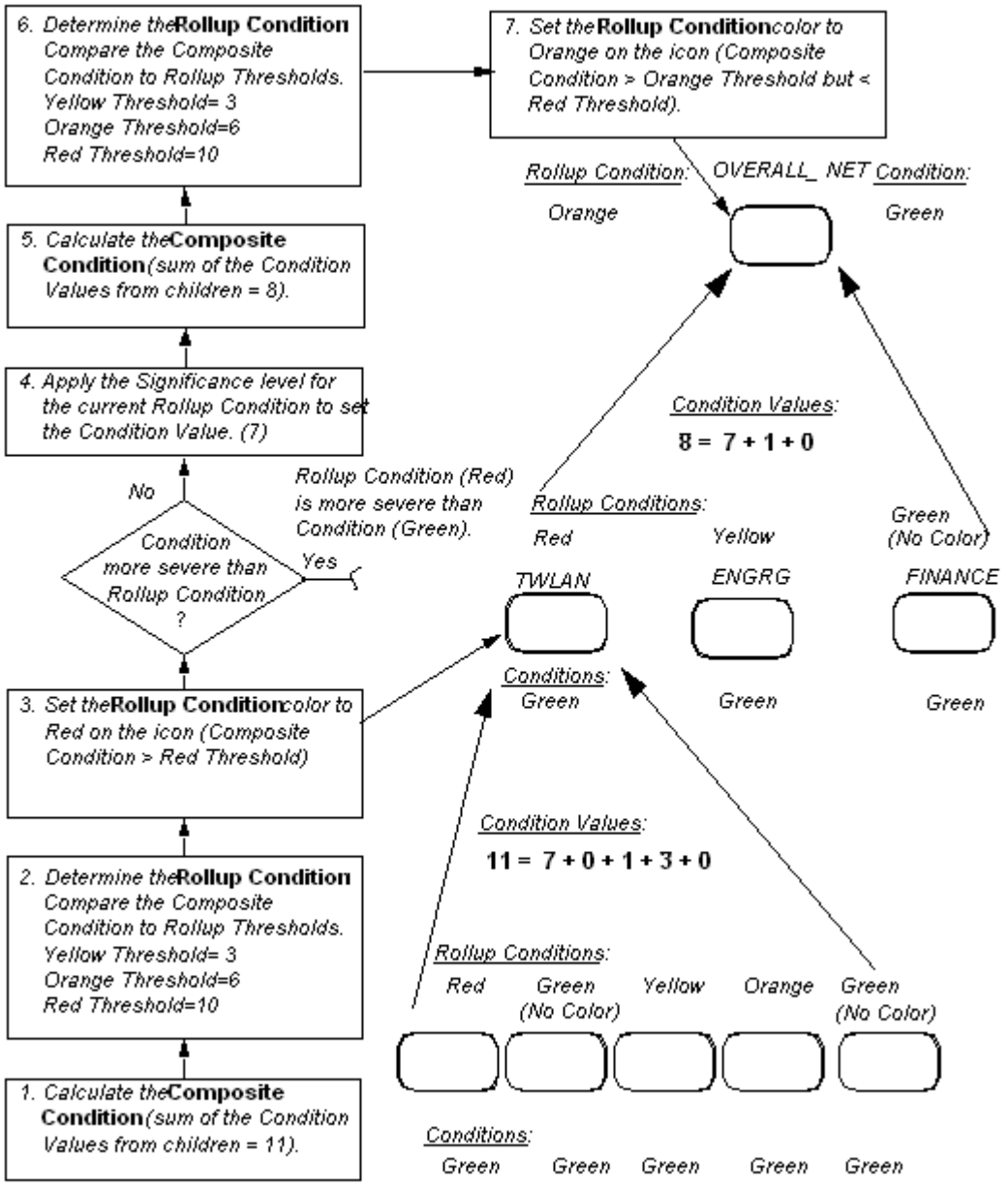
Rollup Condition Flow

An overview of the rollup condition process is illustrated in the following diagram. Read the flow from bottom to top, but keep in mind that it shows a single path in the propagation of rollup condition and that there may be many children passing condition values to a parent model.



Example of Rollup Condition Propagation

The following diagram illustrates the propagation of a rollup condition in the Topology hierarchy.



This example depicts two layers of a Topology hierarchy. The example assumes the use of default Rollup Thresholds and Significance Levels. At the lowest level in the figure there are five devices: two hubs, a router, and two end-point devices. These are contained by TWLAN, a LAN of type 802_3_LAN (as are the two other LANs: FINANCE and ENGRG). The network group model named OVERALL_NET collects these three LAN models.

The following Rollup Conditions and Conditions, at lower levels, determine the top-level Rollup Condition for the network group model OVERALL_NET:

Devices Collected by TWLAN

Hub#1
Condition = Green
Rollup Condition = Red

Condition Value = 7

Hub#2
Condition = Green
Rollup Condition = Orange
Condition Value = 3

Router#1
Condition = Green
Rollup Condition = Yellow
Condition Value = 1

End-Point Devices PC#1 & PC#2
Condition = Green
Rollup Condition = Green
Condition Value = 0

LAN Named FINANCE

Condition = Green
Rollup Condition = Green
Condition Value = 0

LAN Named ENGRG

Condition = Green
Rollup Condition =Yellow
Condition Value = 1

Example Rollup Condition Process

Every model within a Topology view receives its Collects relation from the model that it is collected by. Therefore, all models contribute to the rollup condition of the network group model. The following steps provide a detailed flow of condition values that contribute to the rollup condition for the model named OVERALL _NET shown in the previous example diagram.

1. Determine the Composite Condition for the TWLAN network group model. Composite Condition is the sum of the collected models' Condition Values. In this case, the device models have Condition Values of:
   "Orange" Condition hub model has a Condition Value of 3.
   "Red" Condition hub model has a Condition Value of 7.
   "Green" Condition PC#1 model has a Condition Value of 0.
   "Yellow" Condition router model has a Condition Value of 1.
   "Green" Condition PC#2 model has a Condition Value of 0.
   Therefore, for the TWLAN model:
   Composite Condition = (3 + 7 + 0 + 1 + 0) = 11
2. Determine the Rollup Condition for TWLAN. In this case:
   Composite Value = 11
   TWLAN Yellow Threshold = 3
   TWLAN Orange Threshold = 6
   TWLAN Red Threshold = 10
   Composite Value > Red Threshold
   Therefore:
   Rollup Condition for TWLAN = Red
3. Assign Significance Levels to TWLAN Condition and Rollup Condition. In this case, Significance Levels are:
   Value When Yellow = 1
   Value When Orange = 3
   Value When Red = 7

Therefore:
Rollup Condition = Red Condition = 7

4. Set Condition Value for TWLAN model. In this case:
   Rollup Condition more severe than Condition
   Therefore:
   Condition Value = Rollup Condition Significance Level = 7
   The three network models TWLAN, ENGRG, and FINANCE pass their Condition Values up to the network group model OVERALL_NET. Changes in the Condition or Rollup Condition for the device models at lower levels can produce changes in the topology models further up in the Topology hierarchy. The Rollup Conditions for these three networks produce the following Rollup Condition for OVERALL_NET.
5. Determine the Composite Condition for the OVERALL_NET network group model. Composite Condition is the sum of the collected models' Condition Values. In this case, the network models have Condition Values of:
   Red Condition TWLAN model has a Condition Value of 7.
   Green Condition FINANCE model has a Condition Value of 0.
   Yellow Condition ENGRG model has a Condition Value of 1.
   Therefore, for the TWLAN model:
   Composite Condition = (7 + 0 + 1) = 8
6. Determine the Rollup Condition for OVERALL_NET. In this case:
   Composite Value = 8
   Yellow Threshold = 3
   Orange Threshold = 6
   Red Threshold = 10
   Composite Value > Orange Threshold, but < Red Threshold
   Therefore:
   Rollup Condition for OVERALL_NET = Orange

Fault Isolation

Fault Management is one of the key requirements of network management. A fault is different from an error because it is an abnormal condition that requires management attention and repair. Problems that results in faults could be caused by a bad firmware, a bad hardware, or a bad network. Each of these problems requires a different response from the network manager. Thus the goal is to determine the exact location of the fault and to get the attention of the network administrators as quickly as possible.

DX NetOps Spectrum intelligence is capable of isolating a network problem to the most probable faulty component. To speed up fault isolation and to reduce unnecessary traffic, two actions occur:

- **Are-You-Down Action**
  Upon losing contact with the device it represents, a model sends the Are-You-Down action to all of its neighbors to determine its own condition. If all of the neighbors return a response of TRUE, the condition color of the model turns gray (meaning "my device might be down, but it is impossible to tell because all the neighbors are down"). However, if any of the neighbors return a response of FALSE, the condition color of the model turns red (meaning "my device must be down, because one of the neighbors is up").
- **Are-You-Up Action**
  Upon re-establishing contact with the device it represents, a model sends the Are-You-Up action to its neighbors to speed up the fault isolation. Upon receiving this action, each neighbor returns TRUE if it has an established contact status. If the contact status of the model is lost, and the next-time-to-poll is more than 60 seconds, then the model pings the device for quicker fault isolation.

Every time the status of the model changes, or the information available to DX NetOps Spectrum changes, a new assessment occurs. DX NetOps Spectrum intelligence keeps the topology presentation as current and as accurate as possible, but it depends on correct modeling to accurately assess contact status and determine device failures on the network. Correct modeling includes placing your VNM model in proper relation to the other models that represent your

network; it must have a resolved connection in the Topology view of a model that represents a device to which the VNM host is connected. When the VNM model is properly connected and DX NetOps Spectrum loses contact with a model, the icon representing that model displays a condition color of Gray, Orange, or Red, which helps the network administrators to locate the faults immediately.

Improved Fanout Performance

> **NOTE**
> In 10.4.2, this option is enabled by default; that is, the parameter value is already set to True. In previous releases, the option was not enabled by default.

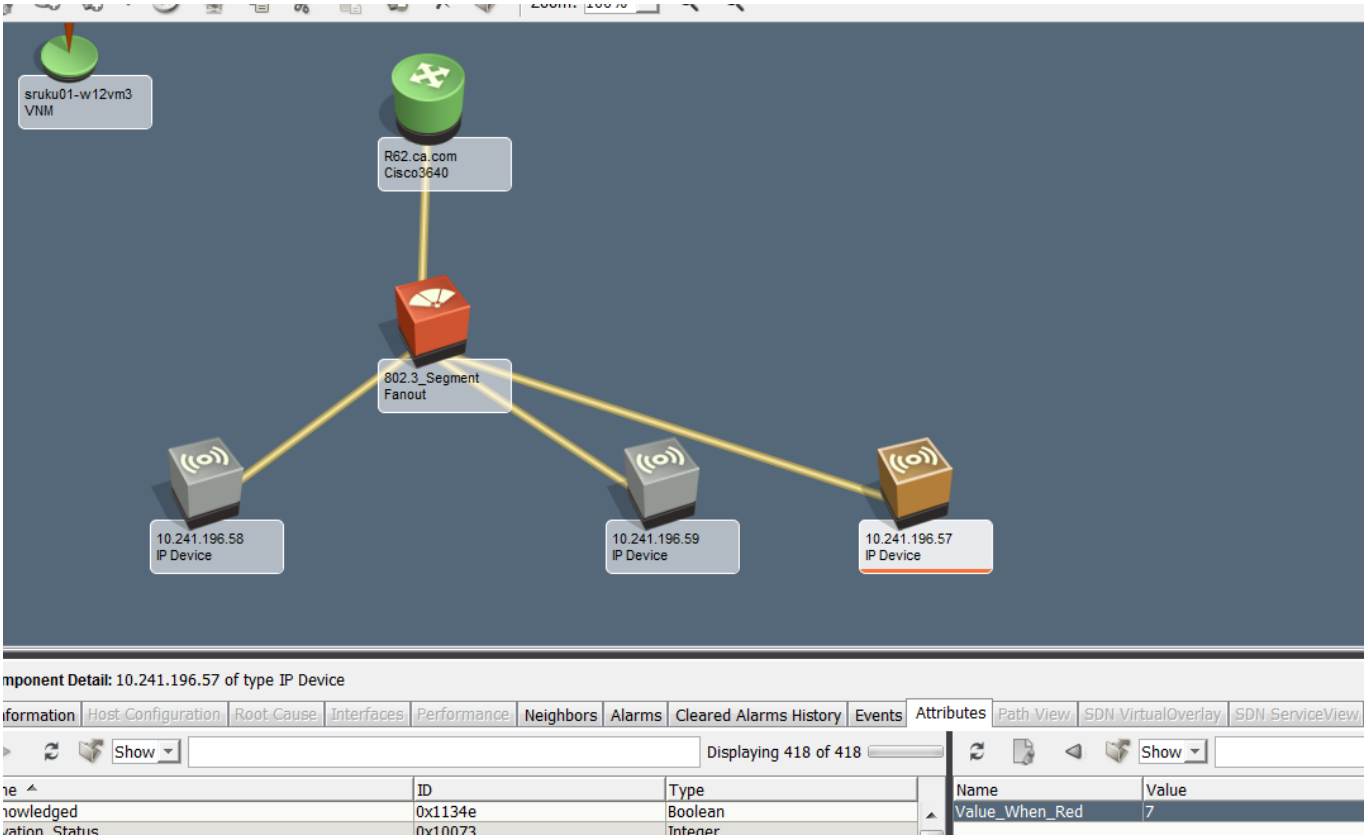
With 10.3.1, the Fanout performance has been improved by propagating the 'Are_You_Down' action to only SNMP-capable neighbors of Fanout. To enable the Fanout performance enhancement, set the parameter attribute 'improve_fanout_performance' to True in the $Specroot\SS\.vnmrc file and restart the SpectroSERVER. By default, this parameter is set to False (in releases prior to 10.4.2). When all the non-SNMP significant device models are down and any SNMP capable device is up, then Fanout condition turns red and the non-SNMP device models are suppressed (shown in Fig.1). To change the non-SNMP insignificant model device to significant model device, refer to the section on How Model Category Affects Contact Status.

Fig.1

NOTE

If a non-SNMP significant device model is in a maintenance mode, then this model is considered as down (shown in Fig.1.1).

Fig.1.1



How Model Category Affects Contact Status

Each fault is associated with a particular condition, which is represented by a particular color that displays on the icon representing the model where the fault occurs. The condition color reflects both the contact status and the alarm status of the model. However, the contact status and condition color asserted for a model also depend upon which of the following categories a model belongs to. The following list summarizes how the categories to which a model and its neighbors belong influences its contact status and condition color.

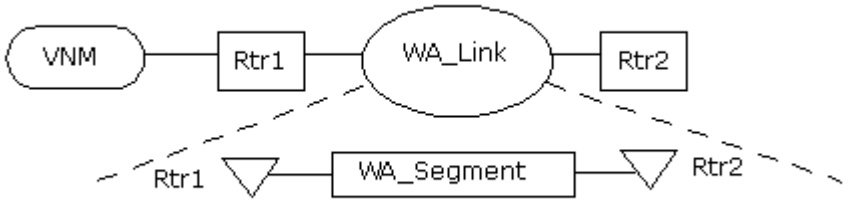
- **Significant Device Models**
  Any device that requires the attention of the administrator for the smooth operation of the network is called a significant device. To change an insignificant model into a significant model change the value of the attribute Value_When_Red (0x1000e) to 7.
- **Insignificant Device Models**
  An insignificant device such as an end-user PC toggles between Blue and Green contact states and does not generate alarms or event messages to get the attention of the administrator. To change a significant model into an insignificant model change the value of the attribute Value_When_Red (0x1000e) to 0.
- **Inferred Connectors**
  These are dumb models that do not poll, but that keep track of a list of their Data Relay neighbors. Possible inferred connectors are: WA_Segment, Fanout, and so on. DX NetOps Spectrum automatically enables Live Pipes for all ports that are connected to a WA_Segment.

  NOTE
  DX NetOps Spectrum intelligence does not expect Fanout models to be connected to each other; thus this configuration results in inaccurate contact status displays. If two Fanouts are connected to each other and each of them is in turn connected to a device with a green contact status, the Fanouts nonetheless turn red.

If two Fanouts are connected to each other with no other devices that are connected to either one, both Fanouts turn gray.

- **Shared Media Link**
  The Shared Media Link is a specialized inferred connector. These models are similar to Fanouts, but the fault management works differently. Unlike a Fanout model, the Shared Media Link model condition is based on configured threshold values.
  Example: If the critical threshold is set to 80, the Shared Media Link turns red when it loses contact with 80 percent of the downstream models.
- **Composite and Discrete Topology Models**
  The contact status of LAN, LAN 802.3, LAN 802.5, and so on, models is determined by the contact status of its collected children. A LAN model with lost contact status turns either red or gray, depending on the condition of its collected models.
- **Wide Area Links**
  Wide Area Links (WA_Links) are modeled with wide area segment (WA_Segment) models. This allows for proper rollup of the Wide Area Link condition. WA_Link models can only represent point-to-point connections, such as T1 and T3 lines, and there can be no more than two devices that are connected to it at a time. Also, you must connect the WA_Segment model to the correct port of the device models.

  NOTE
  WA_Link models can accommodate only one WA_Segment model. If you attempt to paste more than one WA_Segment model into a WA_Link model's Topology view, the second one is destroyed immediately and an alarm is generated.



- **Wide Area Segments**
  WA_Segments poll the InternalPortLinkStatus (IPLS) attribute of each interface model which Connects_To the WA_Segment. This is an active poll, meaning that the IPLS of each connected interface is read at every polling interval rather than simply watched for a change in the attribute. Therefore, DX NetOps Spectrum does not have to lose contact with one of the connected routers for a fault isolation alarm to be generated on a WA_Link.
  The polling of the connected ports' IPLS is regulated by the WA_Link model's Polling_Interval and PollingStatus attributes. When the Polling_Interval changes to zero (0) or PollingStatus goes to FALSE, polling of the connected port's IPLS is stopped.
  If one of the connected interfaces has an IPLS of BAD (for example, Admin Status is ON, but Open Status is OFF), then the WA_Segment's Contact_Status is set to 'lost' and the WA_Segment turns gray. The WA_Link turns red.
  If one of the connected interfaces has an IPLS of 'disabled' (for example, Admin Status is OFF), then the WA_Segment's Contact_Status is set to 'lost' and the WA_Segment turns gray. The WA_Link turns orange. This is because the alarm must be severe enough to be viewed in the Alarms tab, but it is not a "Contact Lost" alarm.
  If the DISABLED interface causes DX NetOps Spectrum to lose contact with the remote router, then the WA_Link turns red. This is the regular InferConnector-type fault isolation working.

Model Category	Connected Models (Neighbors)	Condition Color
Significant Devices (Modeling Hub-types only)	connected to a VNM...	turn Red after losing contact

Significant Devices	with no connections to other models (a zero connector count)...	turn Red after losing contact
Significant Devices	connected to an established Data Relay neighbor...	turn Red after losing contact
Composite and Discrete Topologies	in which all of the collected children have a lost contact status and at least one of those collected children is Red...	turn Red after losing contact
Inferred Connectors	where the Fanout model has lost contact but one of its neighbors is good and the associated port has bad port link status, then it...	turn Red after losing contact
Significant Devices, Inferred Connectors, and WA_Links	where all neighbors have also lost contact status...	turn Gray after losing contact.
Composite and Discrete Topologies	in which all ocs and none of those collected children are Red...	turn Gray after losing contact.
Significant Devices (Modeling Hub-types only)	connected to an end-point neighbor (such as a PC) that has established contact status...	turn Orange after losing contact.
WA_Links	WA_Segment (or Fanout) is good and one of the routers is lost then...	turn Orange after losing contact.
Significant Devices	connected to a model with an Established contact status...	turn Green.
Composite/Discrete Topologies and WA_Links	in which any of the collected children has established contact status, then the LAN will also...	turn Green.
Inferred Connectors	connected to a model with an established contact status where at least one of its neighbors is *Good* and its associated port (port that is connected to the Fanout) status is *Good*...	turn Green.
Significant and Insignificant Devices	not yet connected to other devices...	turn Blue
Composite/Discrete Topologies and WA_Links	when all collected children of a LAN have initial contact status, then the LAN will also have the initial contact status...	turn Blue

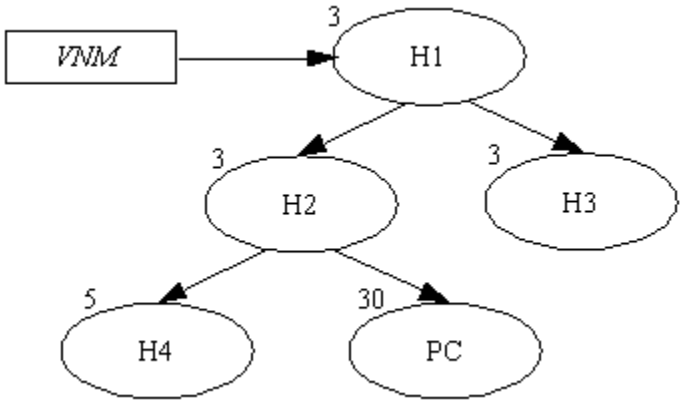
Fault Isolation Examples

The following examples illustrate how DX NetOps Spectrum fault isolation operates with various network configurations and problem scenarios.

Example: Proactive Fault Isolation

This example demonstrates that fault isolation is a proactive mechanism which does not depend upon polling all of the connected models.

Consider a simple network topology as shown in the following diagram. The device H1 is connected to the VNM model. Devices H1, H2, and H3 poll every 3 minutes. H4 polls every 5 minutes. The PC polls every 30 minutes.



Assume that H2 is BAD. As a result H2 turns red, H4 turns gray, PC (insignificant model) turns blue, while H1 and H3 remain green.

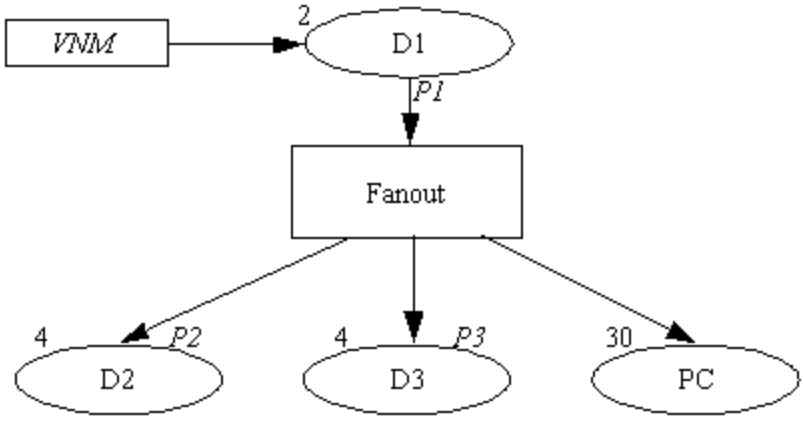
Fault isolation is initiated when H2, H4, or PC polls. If H4 is lost, it sends an Are-You-Down action to H2. If H2 is lost by then, it sends TRUE to H4, otherwise it pings itself and then sends the response to H4. This causes H4 to turn gray.

Now H2 is lost, and it sends Are-You-Down action to H1. Because H1 is established, H2 has to decide between orange and red conditions. H2 pings PC. Since PC cannot respond H2 turns red. The ping from H2 puts PC in a lost state. Since PC is an insignificant device, it turns blue.

Example: Modeling a Fanout.

This example demonstrates fault isolation when modeling a Fanout.

Assume that the Fanout is red and D2, D3, and PC are gray. The following diagram illustrates this scenario.



The Fanout registers a watch on D1's contact status. If D1 goes down, the Fanout turns Gray as a result of the watch trigger.

When D3 eventually polls successfully, D3 has an established contact status and turns Green. D3 then sends an Are-You-Up action to the Fanout. The Fanout reads device P3's (D3's port connection to the Fanout) internal link port status. Assuming the port has a good status, the watch is cleared, and the Fanout turns Green with an established contact status. This means that as long as P1 (D1's port connection to the Fanout) has good internal link port status, the contact status of the inferred connector remains good.

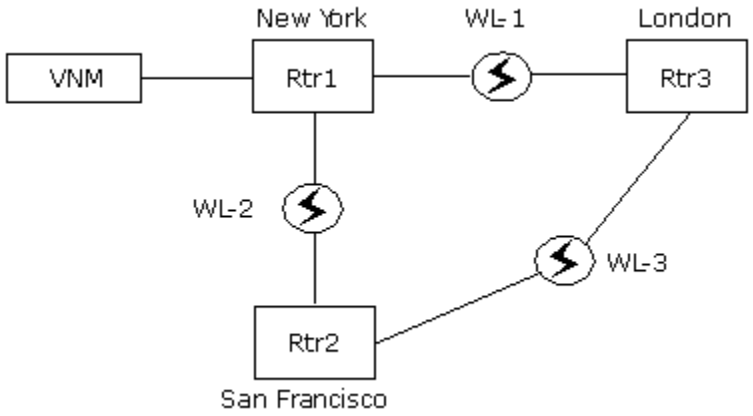
What if D2 goes bad? D2 loses its contact status and sends an Are-You-Down action to the Fanout. The Fanout pings D1, and finds D1 to be good. The intelligence then examines the status of P1. Assuming Link-Status of P1 is good, the Fanout returns FALSE to model D2. This causes D2 to turn Red.

What if P1 is bad? This is the same case as disconnecting the network connection to the Fanout. If D3 polls first, it loses its contact status and sends an Are-You-Down action to the Fanout. The Fanout pings D1 as finds it as a good neighbor. Fanout then reads the internal-port-link-status of the port P1. Because P1 is bad, the Fanout will lose its contact status and turns Red. The Fanout returns TRUE to the model D3. This causes D3 to turn Gray. D2 will also turn Gray in the same way as D3. PC being the insignificant device will turn Blue immediately after losing its contact status.

Example: Redundant Paths Fault Isolation

This example shows how DX NetOps Spectrum manages devices using redundant paths if a link is shut down administratively (i.e., admin-status equals *down*).

The following diagram depicts a network with redundant WA Links. Here VNM manages Rtr3 through link WL-1 and Rtr2 using link WL-2. Assume that the network administrator shuts down the WL-1 link. This causes WL-1 to turn gray. Rtr3 turns red because VNM cannot talk to it through WL-1. The redundancy intelligence of Rtr3 modifies its agent address, so that VNM can talk to it using links WL-2 and WL-3. This causes Rtr3 to turn green again. The link WL-1 still has the gray condition.

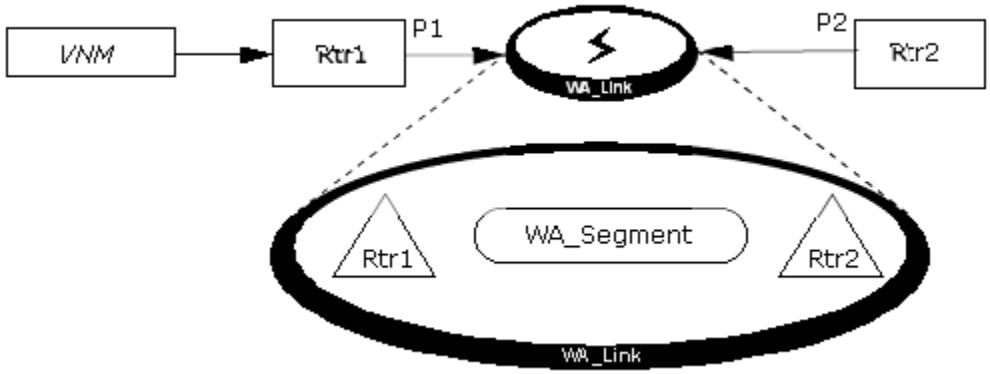


Example: Inferred Connector Fault Isolation

This example demonstrates that fault isolation for an Inferred Connector requires specific modeling. Assume that two routing devices, Rtr1 and Rtr2, are connected at both ends of the WA_Link and that their ports are P1 and P2 respectively.

WA_Link models needs to be associated with a WA_Segment (or Fanout) model through the Collects relation to enable the proper rollup of the WA_Link condition. The devices at either end of the WA_Link needs to be connected to the WA_Segment collected by the WA_Link model. You do this by navigating into the device's Device Topology view and resolving the WA_Segment off-page reference icon to the appropriate port. You can view the connections by navigating into the WA_Segment's view.

This cross-connection is important for fault isolation to work, as shown in the following diagram.



Assume that P1 is the port on Rtr1 and P2 is the port on Rtr2. The routers that are connected to the WA_Segment causes it to behave as described in the following table. Note that the port link status becomes important in determining the status of the WA_Link only when both routers are “contact established.”

Rtr1	Rtr2	WA_Link
Initial	Initial	Blue
Established	Lost	Red
Lost	Lost	Gray
Established	Established	Check Port States*

* If both Rtr1 and Rtr2 have a contact status of *established* then the port status of P1 and P2 determines the condition of the WA_Link. If any port is BAD, the WA_Link turns RED. If any port is DISABLED, the WA_Link turns ORANGE. Otherwise, the WA_Link turns GREEN.

Duplicate Addresses

DX NetOps Spectrum intelligence automatically detects when duplicate IP addresses are entered in the SpectroSERVER database. Although some devices are allowed to have a duplicate IP address, Cabletron hub devices should be configured with only one IP address per device.

DX NetOps Spectrum can model different devices that share IPs, provided that each device has at least one IP that is unique to that device. This modeling policy accommodates certain networking technologies such as load balancing that create identical IP addresses across a range of devices. Devices that share some interface IP addresses can be modeled manually or by using Discovery. However, devices that share all their interface IP addresses in common cannot be modeled manually or by using Discovery.

Alarm condition colors warn you of duplication, as shown here:

- **Same MAC Address & Different IP Address**
  This alarm occurs when there are two or more models with the same MAC address and at least one model with a different IP address.
  Color: Yellow
- **Same IP Address & Different MAC Address**
  This alarm occurs when there are two or more models with the same IP address and at least one model with a different MAC address.
  Color: Orange
- **Same IP Address & Same MAC Address**

This alarm occurs when there are two or more models with the same IP address and the same MAC address (duplicate addresses).
Color: Yellow

- **Duplicate MAC Address**
  The special case alarm for duplicate models where at least one of the models does not have an IP address. Only a Physical_Address model type can have this characteristic.

  NOTE
  Even if the MAC address for two device models is identical, this alarm occurs only when the MAC address of every interface of the two devices is the same.

  Color: Yellow

To get these alarms, a model type needs both the MAC address and the IP address. For example, the model types Pingable and PhysicalAddress do not have both addresses, so you will not see these alarms.

Manually Clear Duplicate Addresses

You can clear duplicate address alarms manually.

Follow these steps:

1. Select the model with the duplicate IP address alarm.
2. Determine whether to allow two devices to have the same IP address. If not, change one of the devices to use a unique IP address, and then use the Update feature to change the IP address within DX NetOps Spectrum.
3. To clear the duplicate, click

   (Clear selected alarms).
   The alarm is cleared. The status color on the model icon returns to a normal green condition unless another alarm is present for the model.

Automatic Naming and Addressing

DX NetOps Spectrum implements an automatic model naming and addressing feature through the AUTO_NAME attribute (attribute ID 0x00011979). The value for this attribute is set to TRUE by default for each model type in your modeling catalog. You can disable automatic naming and addressing on a model type basis using the Model Type Editor (MTE) to set the value to FALSE. Otherwise, the feature functions as described below.

If you create a new model using only the IP address, DX NetOps Spectrum automatically attempts to supply a name for the model in one of three ways:

- Using NIS (Network Information System) or DNS (Domain Naming Service) to get the name from the modeled device
- Checking the local /etc/hosts file for the name associated with the modeled device's IP address
- Using the IP address as the model name

NOTE
The priority order of the source that will be used to supply a name for the model (IP Address, Name Service, or sysName) is dictated by the Model_Naming_Options attribute on the VNM model. This attribute can be modified on the VNM model's control view.

NOTE
For more information about configuring landscapes, see the Distributed SpectroSERVER Administrator section.

Likewise, if you create a new model and supply only a model name, DX NetOps Spectrum will attempt to use NIS, DNS, or the local /etc/hosts file to retrieve the modeled device's IP address.

In either case, as long as the value for AutoName is TRUE for a particular model type, DX NetOps Spectrum will automatically maintain the names for models of that model type as follows: in the event the IP address for a model changes and the original model name was supplied by DX NetOps Spectrum, a new name will be supplied using one of the three methods listed previously. However, if the original name was supplied by the user and differs from the name that would be supplied through automatic naming, then the original name will be preserved.

Board and port models are also automatically named by default, each being assigned the name of the parent device model suffixed with the board/port Instance ID. For example, if a model of model type Hub_CSI_IRM2 is named IRM2_UK, and the modeled device has a port with the Instance ID of 2.5, the name of the port will be IRM2_UK.2.5. If the device name is changed to IRM2_US, then the name of the port becomes IRM2_US.2.5. However, if the device name was user-specified and the user then changes the port name from IRM2_US.2.5 to LAB_PORT, then the automatic naming will not be not used for that port in the event of subsequent IP address changes. Some boards (mainly standalone MIMs) contain their own intelligence. In such cases, setting AUTO_NAME to FALSE as previously described will disable the autonaming intelligence and allow the board's own intelligence to work.

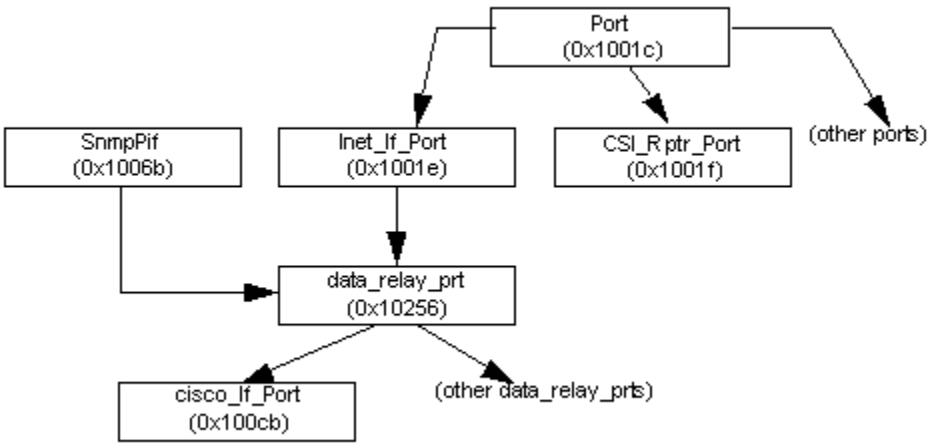
Detection of Firmware Problems

DX NetOps Spectrum allows for automatic detection of problems in some device firmware. Using the connects_to and collects model type relations formed in the Topology views, DX NetOps Spectrum detects if a network management firmware problem exists in the hub device. The connects_to relation denotes that one model is attached or connected to another; for example, a PC model connects to a hub model via a port on the hub. The collects relation denotes that one model collects information from another mode; for example, the Topology view model collects information from the hub devices contained within that view. If DX NetOps Spectrum cannot retrieve management information from a hub device, but can still contact devices connected to this hub, then a hub firmware management problem can be deduced and the hub icon's condition color is set to orange. You should then use the alarm views along with the content in the Impact tab and the Performance tab to help isolate and correct the problem.

Interface Intelligence

Interfaces are ports that have both a physical address and a network address. These types of ports are found on routers and bridges, where network identity is important, unlike a port on a repeater, which might have a physical address, but not a network address.

The following diagram illustrates the derivation of the interface port:



Port is the base model type for all ports. Two basic types of ports are derived from Port: repeater ports and interface ports.

- **Inet_If_Port**
  Derived from Port and Enet Monitor model types. Inet_If_Port is the base class for all interface ports that are potential monitor points for network statistics.
- **Data_relay_prt**
  Derived from Inet_If_Port and SnmpPIf. SnmpPIf is the base class for all model types that communicate with SNMP agents. Data_relay_prt is the base model type for all interface ports that do their own reading and polling. It is an instantiable model type and is used to model a generic interface port. More specific types of interface ports, such as Cisco_If_Port, are derived from data_relay_prt.

The inference handler CsIHInterfaceIntLinkStatus, which computes InternalPortLinkStatus, is attached at Inet_If_Port so that all interfaces inherit the desired functionality.

The intelligence for interfaces that use ifAdminStatus and ifOperStatus are defined in the MIB-II definition of the interface group in RFC 1158. These variables can have the state of ON or OFF, and are defined as follows:

- **ifAdminStatus: desired interface state**
  This is the state that the administrator wants the interface to be in. This attribute shows whether the interface has been shut off. The values of this attribute are ON and OFF.
- **ifOperStatus: current interface state**
  This attribute shows the actual state of the interface. The values of this attribute are UP and DOWN. UP means that the interface is communicating with the network properly. DOWN means the interface has lost connection with the network.

These two variables are used to calculate a DX NetOps Spectrum internal attribute named Internal_Link_Status (IPLS - 0x10f1b). The possible values for this attribute are LINK_STATUS_GOOD (LSG), LINK_STATUS_BAD (LSB), and LINK_STATUS_UNKNOWN (LSU). This attribute is used to create and clear alarms, both on the interface and the device it is part of. It is also used to generate events concerning the attempt to reach of the interface. The following table shows how these variables are calculated.

IfAdminStatus	IfOperStatus	INTERNAL_PORT_LINK_STATUS
ON	ON	LINK_STATUS_GOOD
ON	OFF	LINK_STATUS_BAD
OFF	ON	LINK_STATUS_UNKNOWN
OFF	OFF	LINK_STATUS_UNKNOWN

The INTERNAL_PORT_LINK_STATUS of interface is set to LSU when DX NetOps Spectrum has lost contact with the device.

Interface Alarms

Internal Port Link Status (IPLS) is used to generate alarms for both the device model and the interface model. The only interface alarm is gray. These alarms can only be seen by going into the Alarm Details tab of an interface model. It is interesting to note that the alarm for an interface with an IPLS of LSB have a gray alarm. All models with alarms are displayed in the Alarms tab. This is undesirable for interfaces. Interfaces are considered to be a part of a larger device such as a router. When a router goes down, all its interfaces also go down. A red alarm is generated for the router. It would be confusing to have all of the interfaces also producing red alarms, cluttering the Alarms tab and making it difficult to locate the router.

A device watches the IPLS of each of its interfaces. If any of the interfaces has an IPLS of LSB, then the device generates a yellow alarm with a probable cause of CS_ALARM_CAUSE_PORT_LINK_STATUS_BAD. This alarm, once set, is not be reasserted until it is cleared. Only one alarm is asserted for all ports. The first interface with an IPLS of LSB creates an alarm. The second and successive interfaces with an IPLS of LSB are put into a bad port list. Once the list is clear the yellow alarm is removed.

Each of the interfaces watches its own IPLS. Whenever the interface has an IPLS of LSB or LSU it generates a gray alarm. This alarm only shows up in the interfaces Alarm Details tab.

When the device becomes unreachable, the interface's IPLS is set to LSU. A gray alarm with a probable cause of CS_ALARM_CAUSE_DEV_CONTACT_STATUS_LOST is generated.

When the interface has been administratively shut off the interface's IPLS is set to LSU. A gray alarm with a probable cause of CS_ALARM_CAUSE_ADMIN_SHUT_OFF is generated.

When the interface becomes unreachable, its IPLS is set to LSB. A gray alarm with a probable cause of CS_ALARM_CAUSE_PORT_LINK_STATUS_BAD is generated.

Alarm Configuration

- **Generate Alarm on Port**
  This attribute determines if an event/alarm must be generated on the port for changes to Internal Port Link Status.
- **Generate Alarm on Device** This attribute determines if an alarm should be on device for any port link down traps that are received from the device.

Interface Events

The interface generates two events that deal with its status. The events contain information about the attempts to reach the interface. These events are generated when a device has a yellow alarm due to a bad interface. Each event message contains the interface number and IP address. For example:

```
Tue 20 Jul, 1994 - 13:31:50 Interface 2 (IP   address = 129.128.127.2, type = Gen_IF_Port) on device cisco1 of
  type Rtr_CiscoMIM is unreachable. - (event [00010623])
```

As stated before, IPLS has three possible values. This makes it important to know the last two states of an interface's IPLS to make a proper judgment about its current state. If the interface IPLS is LSB, and it was previously LSU, it is important to know if it was previously LSB or LSG. Due to this, each interface keeps the values of the last two states of IPLS. Events are generated based on these two saved values, and the current value.

Events Generated from IPLS State

The following table shows which states generate events based on the IPLS.

Two Previous	Previous	Current	Event
GOOD	UNKNOWN	GOOD	none
GOOD	UNKNOWN	BAD	UNREACHABLE
GOOD	BAD	GOOD	REACHABLE
GOOD	BAD	UNKNOWN	none
BAD	GOOD	BAD	UNREACHABLE
BAD	GOOD	UNKNOWN	none
BAD	UNKNOWN	GOOD	REACHABLE
BAD	UNKNOWN	BAD	none
UNKNOWN	GOOD	BAD	UNREACHABLE
UNKNOWN	GOOD	UNKNOWN	none
UNKNOWN	BAD	GOOD	REACHABLE
UNKNOWN	BAD	UNKNOWN	none

MPLS Transport Manager

MPLS Transport Manager is a utility that monitors the health of your MPLS core network. Continuously monitoring the MPLS environment can help identify potential performance issues, preventing service interruptions to your customers and helping ensure your customer SLAs are not broken. The MPLS Transport Manager data can also help you pinpoint and effectively troubleshoot problems within your MPLS network by providing the impact of an outage in terms of its affect on the MPLS infrastructure.

A key challenge when monitoring MPLS data is keeping the data accurate. Changes happen dynamically in an MPLS network -- LSPs and their associated Paths can change frequently under certain network conditions. MPLS Transport Manager keeps up with these changes and accurately models the current state of LSPs and Paths. MPLS Transport Manager provides complete visibility into all provisioned LSPs and Paths, and it knows the relationship between an LSP and its primary and secondary Paths.

Access to MPLS Transport Manager is provided in the DX NetOps Spectrum interface. MPLS Transport Manager supports the MPLS implementations of multiple vendors, and both proprietary and standard-based technologies.

Who Should Use MPLS Transport Manager

MPLS is a broad technology defined by hundreds of standards that is continuing to grow, supporting increasingly complex network systems and services. MPLS Transport Manager is intended for TE MPLS environments only. If you have this type of MPLS environment and must adhere to your customer SLAs, you can use MPLS Transport Manager to monitor device outages, their impact on the MPLS network, and effectively troubleshoot these outages.

System Requirements

MPLS Transport Manager is an add-on application that works within DX NetOps Spectrum. In addition to a running SpectroSERVER installation, MPLS Transport Manager requires the following:

- Appropriate MIBs implemented and populated on your MPLS hardware:

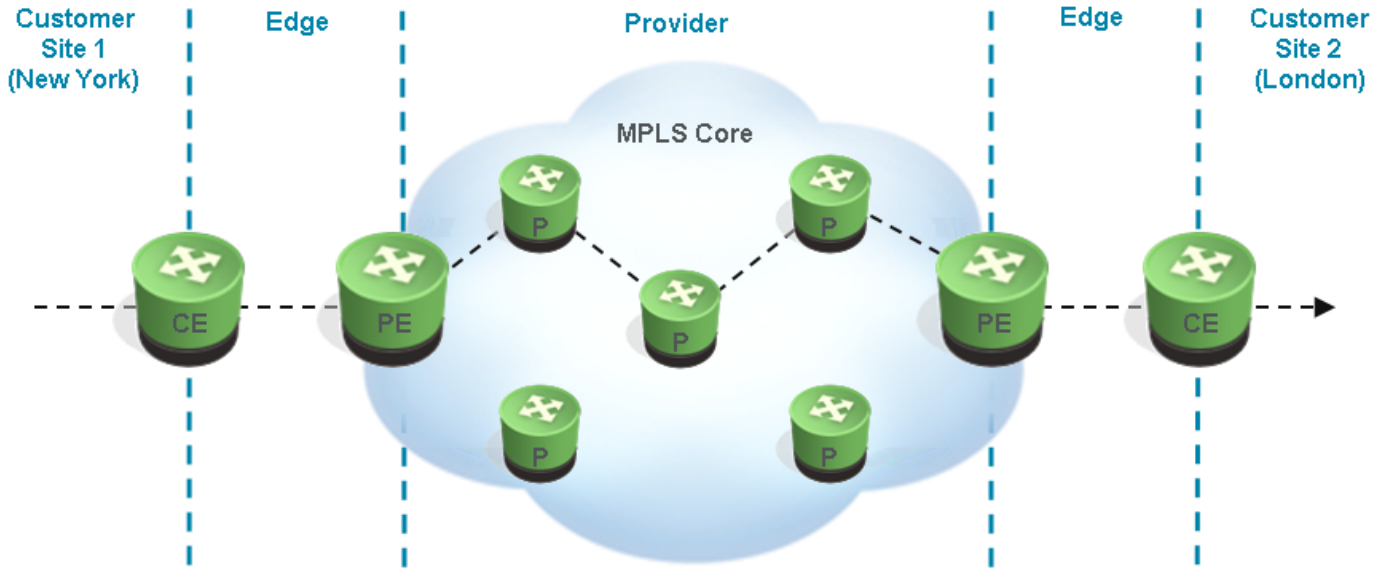
  NOTE
  DX NetOps Spectrum comes with all the required MIBs.

  - MPLS-TE MIB (Cisco devices)
  - Juniper TE MIB (Juniper devices)
  - TIMETRA-MPLS-MIB (Alcatel devices)
- Management module for Cisco, Juniper and Alcatel routers.

  NOTE
  The device models in these modules have the necessary application models needed for discovery of your MPLS environment.

How MPLS Transport Manager Works with MPLS-TE

As a service provider, your goal is to use your MPLS core network to transport data packets from one part of your customer's network to another. For example, your MPLS network may transfer data packets from your customer's New York office to their London office. In this situation, your customer cannot model and monitor their network traffic through your MPLS network, and you, as a service provider, typically cannot model and monitor the client networks in New York or London. The following diagram shows that in this scenario the only interface between the customer network and your MPLS network is the relationship between their Customer Edge (CE) router and your Provider Edge (PE) router:



As shown in the diagram above, a customer communicates from one side of their network to the other using the Provider's MPLS network as follows:

1. A data packet from Customer Site 1 passes through the CE and PE routers on one edge of your MPLS core.
2. Based on its label, the packet passes from router (P) to router through the MPLS core. This route is defined as the primary Path. Each Path is an ordered set of routers through which the packet is passed, moving it from one router (also known as a "hop") to another within the MPLS core. The first device in a Path is the ingress device, and the last one is the egress device.

   NOTE
   Excluding topology changes or load balancing, all subsequent packets with the same source and destination take the same path through the core. If any device along the primary Path becomes inoperable, the packet switches to a secondary Path.
3. A packet passes through each hop in a Path, until it reaches the egress device on the other side of the MPLS core.
4. From the egress device, the packet is passed to its next destination within Customer Site 2.

MPLS Transport Manager models the relationships between the PE routers, Paths, devices, and hops to monitor any changes that can negatively impact the ability to move customer data packets through your MPLS core. For example, if a router goes down, an LSP can change from its primary Path to a secondary Path. Using traps and polling, MPLS Transport Manager is aware of this change -- it logs the event and may trigger an alarm (based on your threshold values) to make sure that you are aware of the change. Knowing that an LSP switched to a secondary Path can help you locate unreachable devices and quickly assess their impact on your clients.

Devices Supported by MPLS Transport Manager

MPLS Transport Manager supports MPLS-enabled Cisco, Juniper, Nokia and Alcatel devices. The following tables describe which features are available and supported by each device.

NOTE
For more information about the Cisco devices, models, and firmware that support MPLS technology, see the MIB Locator on the Cisco website.

MIBs Supported

MIB	Cisco	Juniper
TE-MIB	Yes	No
Juniper MPLS MIB	No	Yes

Functions Supported

Function	Cisco	Juniper
Tunnel/LSP name	Yes	Yes
Tunnel interface	Yes	No
Tunnel primary instance	Yes	No
Total # of paths	Yes	Yes
# of path changes	Yes	Yes
Active path	Yes	Yes
Active path hops	Yes	Yes
Primary path hops	Yes	Yes
Alternate path hops	Yes	Yes*

NOTE
* Some details about Juniper routers may not be immediately available upon discovery. Although the Path model lists all hops for the primary and secondary Paths in the hops table, Juniper routers provide details only when they are included in the active Paths. Therefore, when a secondary Path becomes active, MPLS Transport Manager populates the hops table with details about the Juniper router used in that Path.

Dynamic Discovery Traps Supported

Trap	Cisco	Juniper
Tunnel creation traps	Yes	Yes
Tunnel removal traps	Yes	Yes
Tunnel change traps	Yes	Yes

Configuring MPLS Transport Manager

This section describes how to install and configure MPLS Transport Manager. These are tasks that are typically performed only once per installation by the MPLS Transport Manager administrator.

Install and Configure MPLS Transport Manager

MPLS Transport Manager is included in your DX NetOps Spectrum extraction key. When you install DX NetOps Spectrum, the MPLS Transport Manager components are automatically installed and available for use. However for best results, adjust the configuration settings appropriately.

To install and configure MPLS Transport Manager properly, the administrator must complete these tasks:

1. Install DX NetOps Spectrum.

For existing DX NetOps Spectrum installations, perform an in-place installation to ensure the MPLS Transport Manager components are installed. MPLS Transport Manager supports the distributed SpectroSERVER environment (DSS).

NOTE
For specific installation instructions, see the Installing section.

2. Configure port polling. This option must be enabled for impact to be calculated for ports which are part of an LSP. Although enabled by default, you can disable port polling, if needed.
3. Configure the SpectroSERVER traps. These traps determine how to manage various events that can occur in the MPLS core.
4. Configure impact weight values. These impact weights help to determine which alarms have the greatest impact on your MPLS environment.
5. Configure settings for Path change alarms. These settings help to determine the severity of alarms generated when LSPs switch Paths excessively.

Configure Port Polling on LSPs

"Polling" for LSPs actually refers to polling the devices and interfaces on which the LSP traverses. You can disable polling to limit network traffic, but it comes at the loss of significant functionality in MPLS Transport Manager. Along with traps, the polling mechanism is used to determine the health of the resources that make up the MPLS Paths.

NOTE
Only an administrator performs this task.

To configure port polling on LSPs

1. Open the main MPLS Transport Manager page.
   The main details page opens in the Contents panel for the selected MPLS Transport Manager.
2. Click the Information tab in the Contents panel.
3. Expand the Configuration section.
4. Expand the LSP Discovery subsection.
   LSP discovery options display.
5. Click the 'set' link for the Enable Port Polling option.
   The value for the selected option becomes editable.

   NOTE
   Enabled by default, the polling option must be turned on for impact to be calculated for ports which are part of an LSP.
6. Select the desired value for the field and press Enter.
   MPLS Transport Manager is configured to poll LSPs according to your selection.

Configure SpectroSERVER Processing of MPLS Traps

If your MPLS-enabled devices are properly configured to send traps to the SpectroSERVER host, you can use this trap data to create, delete, or update LSPs. Some environments do not support the use of traps, and you can choose to disable them. However, we recommend that you enable traps when possible, because traps (along with polling) provide the best response to network faults and outages.

NOTE
Only an administrator performs this task.

To configure SpectroSERVER processing of MPLS traps

1. Open the main MPLS Transport Manager page.
   The main details page opens in the Contents panel for the selected MPLS Transport Manager.
2. Click the Information tab in the Contents panel.

3. Expand the Configuration section.
4. Expand the LSP Discovery subsection.
   LSP discovery options display.
5. Click the 'set' link for the following trap that you want to configure:
   - **Create LSP on Trap**
     Creates a new LSP when the TunnelUp trap is received and the device model already exists.
     Default: Yes
   - **Delete LSP on Trap**
     Deletes an existing LSP and its associated Path models when the TunnelDown trap is received and the tunnel no longer exists in the device.
     Default: Yes
   - **Update LSP on Trap**
     Deletes all Paths for an LSP and remodels the LSP when the TunnelRerouted trap is received and an LSP model already exists for the LSP.
     Default: Yes
     Example: When an LSP switches from the primary path to the secondary path, the LSP Path models are deleted and remodeled to display the updated Path information.

   The value for the selected option becomes editable.
6. Select the desired value for the field and press Enter.
   The selected trap is configured for SpectroSERVER processing.

Customize Impact Weights for Alarms

You can assign custom impact weights for three types of LSP problems. These impact weights help determine the alarm level for devices and interfaces used in your LSPs, so you can more quickly identify which devices and interfaces must be resolved first.

NOTE
Only an administrator performs this task.

To customize impact weights for your LSP alarms

1. Open the main MPLS Transport Manager page.
   The main details page opens in the Contents panel for the selected MPLS Transport Manager.
2. Click the Information tab in the Contents panel.
3. Expand the Configuration section.
4. Expand the LSP Impact subsection.
   LSP impact options display.
5. Click the 'set' link for the following impact weight options that you want to configure:
   - **Down LSP Weight**
     Defines the value added to a device or interface alarm's total impact weight when that device or interface is used within an LSP that is reporting all Paths are down.
     Default: 100
     Limits: Integers greater than or equal to 0
   - **Switched LSP Weight**
     Defines the value added to a device or interface alarm's total impact weight when that device or interface is used within an LSP that has switched from its primary Path to another one.
     Default: 10
     Limits: Integers greater than or equal to 0
   - **At Risk LSP Weight**
     Defines the value added to a device or interface alarm's total impact weight when that device or interface is used within an LSP that is reporting at least one of its secondary Paths is down.
     Default: 5

Limits: Integers greater than or equal to 0

The value for the selected option becomes editable.

6. Enter the desired value for the field and press Enter.
   The selected impact weight is customized for LSP alarms.

Managing Your LSP Data

This section describes the basic tasks for discovering, viewing, and managing the models associated with your MPLS infrastructure. Although discovery is an administrator-only task, most of the tasks in this section are for general MPLS Transport Manager operators.

Viewing LSP Details

The primary MPLS Transport Manager view is the OneClick Navigation view. MPLS Transport Manager supports the DSS environment. As a result, MPLS Transport Manager is at the global level, not under the landscape level in the navigation pane. As a result, MPLS Transport Manager can discover LSPs across all the landscapes in your distributed SpectroSERVER (DSS) environment.

The following image shows where MPLS Transport Manager fits into the OneClick Navigation view and shows the hierarchy of MPLS information:

```
[+] Configuration Manager
[+] MPLS Transport Manager
    [-] Head-end device 1
       [-] LSP 1
          [-] Path 1
               Device 1
               Device 2
          [+] Path 2
          [+] Path 3
       [+] LSP 2
    [+] Head-end device 2
    [+] Head-end device 3
[-] SpectroSERVER host
   [+] Universe
```

NOTE
Because MPLS Transport Manager is at the global level, the Landscape information is invalid for MPLS Transport Manager, LspHead, Lsp, and MplsPath models. As a result, the Landscape attribute is removed for these models from the Component Detail, Information, DX NetOps Spectrum Modeling Information. Landscape attribute is displayed only for MPLS devices.

An MPLS environment can have hundreds of LSPs, making it difficult to organize and display them logically. Therefore, MPLS Transport Manager groups LSP details under the LSP head-end devices. The LSP *head-end device* is a Provider Edge router device that is used to create the LspHead model in MPLS Transport Manager. This model groups your LSPs by ingress device in the OneClick Navigation panel, making it easier to view and locate data about your LSPs within the MPLS environment.

From the head-end device level, you can expand the navigation tree to view a list of LSPs that begin from the selected head-end device. Expanding these LSPs in the navigation tree displays a list of Paths within the selected LSP, and expanding a Path displays specific devices used in the Path's hops.

Selecting any entry in the navigation tree displays details about the selection within the Contents panel, and selecting items in the Contents panel displays further details in the Component Detail panel.

To help determine the performance of your MPLS environment and your adherence to customer SLAs, you can use the OneClick views to drill into detailed information for each LSP, such as the following:

- Primary and secondary Paths
- Hops for each Path and their order
- Ingress and egress devices for each LSP
- Alarm state for all models, including the severity and impact

If you cannot find the information that you need through the Navigation panel, you can also search the MPLS Transport Manager data using the Locater tab.

Open MPLS Transport Manager

To monitor your MPLS environment, you must first locate the main MPLS Transport Manager page within OneClick. When you open this page, you can access the features used to monitor the status of your MPLS performance.

To open the main MPLS Transport Manager page

1. Open the OneClick Console.
2. Click the Explorer tab in the Navigation panel.
3. Locate and click MPLS Transport Manager in the Navigation panel.
   The main details page opens in the Contents panel for the selected MPLS Transport Manager.

Discover Your LSPs

To monitor your MPLS environment, you must perform a discovery. LSP discovery creates all device, LSP, and Path models, and these models give you a view into the MPLS core that lets you monitor the health of your MPLS infrastructure.

> **NOTE**
> Only an administrator performs this task.

To discover your LSPs

1. Open the main MPLS Transport Manager page.
   The main details page opens in the Contents panel for the selected MPLS Transport Manager.
2. Click the Information tab in the Contents panel.
3. Expand the Configuration section.
4. Expand the LSP Discovery subsection.
   LSP discovery options display.
5. Click the Run button in the Discovery Status field.

> **NOTE**
> In a distributed SpectroSERVER environment, the "Select Landscapes" dialog opens. The "Select Landscapes" dialog lets you include landscapes for which the LSP discovery is performed. The dialog lets you exclude landscapes so that the LSP discovery is not performed against the excluded landscapes. In a standalone SpectroSERVER environment, this dialog is not displayed as the LSP discovery is performed for only one landscape.

DX NetOps Spectrum discovers your MPLS infrastructure and creates all related models. Your MPLS data is ready to view and monitor.

NOTE
Some details about Juniper routers may not be immediately available upon discovery. Although the Path model lists all hops for the primary and secondary Paths in the hops table, Juniper routers provide details only when they are included in the active Paths. Therefore, when a secondary Path becomes active, MPLS Transport Manager populates the hops table with details about the Juniper router used in that Path.

Filter LSP Discovery Results

If you do not want to monitor all LSPs, you can apply a filter that includes or excludes selected LSPs from discovery and modeling. This feature can help save resources by reducing the number of LSPs DX NetOps Spectrum polls.

To filter LSP Discovery

1. Open the main MPLS Transport Manager page.
   The main details page opens in the Contents panel for the selected MPLS Transport Manager.
2. Click the Information tab in the Contents panel.
3. Expand the Configuration section.
4. Expand the LSP Discovery subsection.
   LSP discovery options display.
5. Click Set in the Path Filter Type field and select one of the following options:
   - Exclusive
   - Inclusive
6. Click Add in the Path Filter field.
   The Add dialog opens, prompting you to enter the Path name.
7. Enter the Path name and click OK.
   The Path name is added to the Path Filter list. Depending on the Path Filter Type you select, LSP Discovery is filtered to include or exclude the listed Paths.

Configuring LSP Discovery During Modeling

DX NetOps Spectrum lets you configure Network Services Discoveries, including LSP Discovery for MPLS Transport Manager, during modeling. As a part of modeling configuration, you can specify which Network Service Discoveries to run with the modeling process.

NOTE
For more information, see the Modeling and Managing Your IT Infrastructure section.

Run Discovery on Selected Models

In any OneClick view that lists models, you can select a set of models and run LSP Discovery for those models only. This ability can help you minimize the DX NetOps Spectrum resources required when troubleshooting or verifying changes to the status of only specific devices.

To run Discovery on selected MPLS Transport Manager models

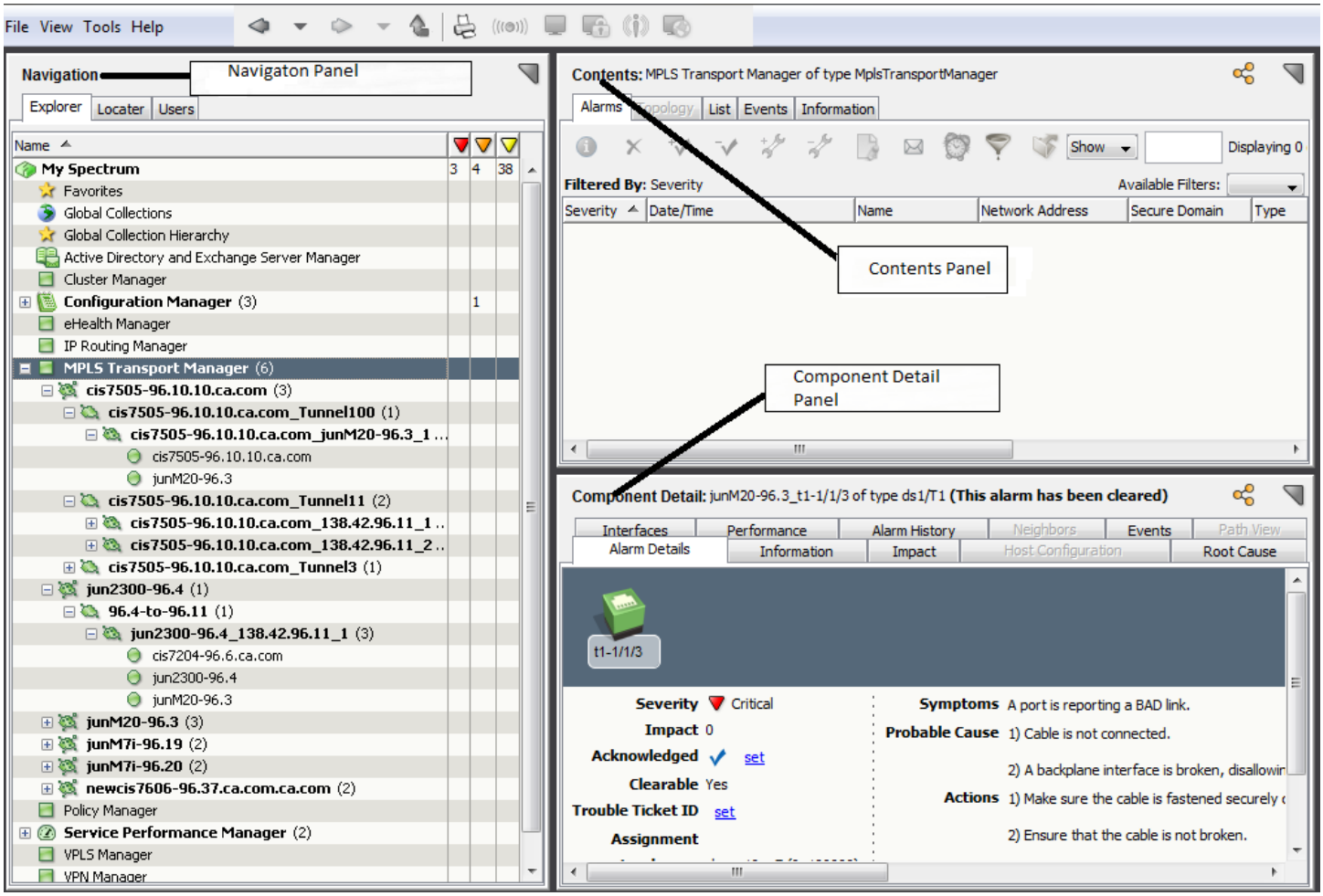
1. Select the models.
2. Click Tools, Utilities, Network Services Discoveries, MPLS Transport Discovery.
   The Discovery process is initiated for the selected models only. You can check the status in the LSP Discovery subview.

Main Window for MPLS

MPLS Transport Manager is integrated into OneClick, and you access the main view through the Explorer tab in the Navigation panel. This tab provides a hierarchical view of the models used to display your MPLS data. This tree structure

helps you manage your customer SLAs and monitor LSP performance by providing a quick way to view details and troubleshoot alarms caused by devices used in your MPLS environment.

When items are selected from the Navigation panel, the item's details display in the Contents panel. When you select items in the Contents panel, additional details display in the Component Detail panel. These three panels are organized as follows:



The Explorer tab in the Navigation panel includes the following items for MPLS Transport Manager information:

- MPLS Transport

Manager

Provides the main access to MPLS models and access to administrator options for configuring the MPLS Transport Manager application. Selecting this item in the navigation tree displays MPLS Transport Manager details in the Content panel on four tabs: Alarms, List, Events, and Information.

Model Class: Application

- *head-end*

device
Groups all LSP models that begin from this device. Selecting a head-end device in the navigation tree displays details in the Content panel on four tabs: Alarms, List, Events, and Information. From these tabs, you can find all LSPs that originate from this device (that is, this device is the LSP's ingress device).
Model Type: LspHead
Example: cisco7505-99.10.xyz.com

- 

LSP
Provides access to LSP details. Selecting an LSP in the navigation tree displays details in the Content panel on four tabs: Alarms, List, Events, and Information. From these tabs, you can view details such as the LSP ingress and egress devices, the LSP's primary and secondary Paths, path change information, and more. You can also select a threshold for path changes.
Model Type: LSP
Example: cisco7505-99.10.xyz.com_Tunnel11

- 

Path
Provides access to Path details. Selecting a Path in the navigation tree displays details in the Content panel on four tabs: Alarms, List, Events, and Information. From these tabs, you can view details such as the Path ingress and egress devices, the Path's hops and their order, the number of unmodeled hops, and more.
Model Type: MplsPath
Example: cisco7505-99.10.xyz.com_cisco7505-99.11.xyz.com_2

View LSP Paths

When analyzing the performance of your MPLS environment, you may need to view the Path details for a specific LSP. You can view details such as how many Paths exist, what is the Path's rank (primary, secondary, and so on), what is the alarm condition for each Path, and more. Viewing this information can help you determine if you are meeting your customer SLAs or can reveal opportunities for improving the performance of your LSP.

To view an LSP Path

1. Open the main MPLS Transport Manager page.
   The main details page opens in the Contents panel for the selected MPLS Transport Manager.
2. Locate and click the LSP on the Explorer tab in the Navigation panel.
   The LSP details display in the Contents panel.
3. Click the List tab.
   All of the LSP's Paths display in the table, showing details for each.

Static vs. Dynamic Paths

Two path types exist in MPLS environments: static (that is, explicit) and dynamic. A static path is an explicitly engineered path from a source device to a destination device, whereas a dynamic path through the MPLS network is computed by the head-end device, based on resource availability and routing protocols.

When analyzing your LSP Paths, knowing the path type can help reveal weaknesses in your MPLS environment and reveal opportunities to optimize the performance of your Paths.

The Path type is available in two locations:

- The List tab of an LSP's Contents pane
- The Information tab of a Path's Component Details pane

Spotlighting LSP Paths

The spotlighting feature in OneClick lets you isolate and visualize model relationships within your network that are not readily visible from the Topology view. For example, the Topology view does not visually distinguish VLANs, VPNs, or LSP Paths, making it more difficult to picture these relationships within the context of your network. With spotlighting, these model relationships are accentuated, showing you where they appear in the network topology.

Using the spotlighting feature, you can select an LSP Path to view in the Topology view. Viewing LSP Path information from this view can help you more easily understand which devices make up the Path. From this view, you can also see if any alarming devices are impacting the Path's performance.

> **NOTE**
> For more information about how to use spotlighting, see the Topology toolbar section.

View Hops in an LSP Path

When analyzing the performance of your MPLS environment, you may need to view the hops in a specific LSP Path. You can view details such as the order of the hops, the devices used for each hop, what is the alarm condition for each modeled device, and more. Viewing this information can help you troubleshoot alarms that occur for an LSP.

> **NOTE**
> Some details about Juniper routers may not be immediately available upon discovery. Although the Path model lists all hops for the primary and secondary Paths in the hops table, Juniper routers provide details only when they are included in the active Paths. Therefore, when a secondary Path becomes active, MPLS Transport Manager populates the hops table with details about the Juniper router used in that Path.

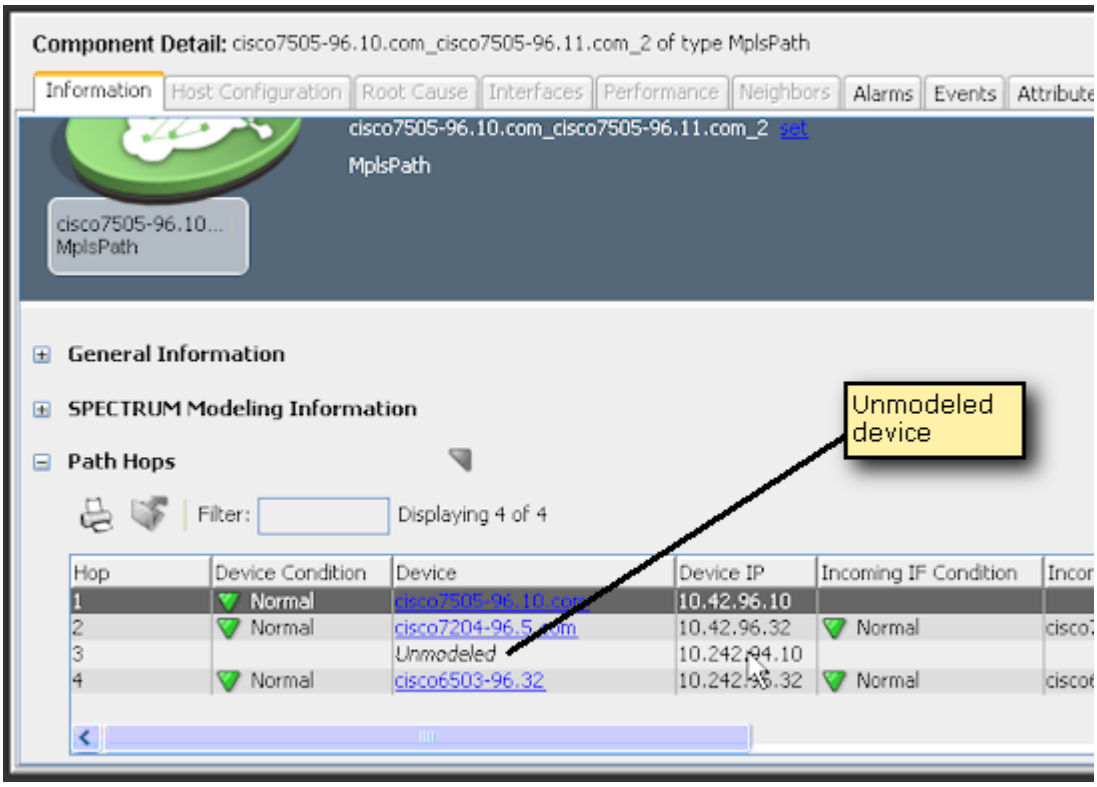
To view hops in a Path

1. Open the main MPLS Transport Manager page.
   The main details page opens in the Contents panel for the selected MPLS Transport Manager.
2. Locate and click the LSP Path on the Explorer tab in the Navigation panel.
   The LSP Path details display in the Contents panel.
3. Click the Information tab.
4. Expand the Path Hops section.
   All hops for the selected LSP Path are displayed in the table.

> **NOTE**
>
> You can also find a Path's hops on the List tab, but we recommend the list on the Information tab. The Information tab displays the order of the hops and displays the ingress/egress interfaces for each hop.

Unmodeled Hops in MPLS Transport Manager

LSP Paths can contain nodes that appear as unmodeled hops. These unmodeled hops are easily recognized in the Path Hops table on the Component Details Information tab. IPs of any unmodeled hops are provided in the table, as shown:



NOTE
You can find paths with unmodeled hops by searching for all Paths and sorting the results by "Unmodeled Hop Count" in descending order.

Unmodeled hops can exist for any of the following reasons:

- The device is not modeled in DX NetOps Spectrum.
- The device is modeled but with an incorrect community name.

Although MPLS Transport Manager can provide Path and hop information without all devices modeled, DX NetOps Spectrum cannot determine the impact to your MPLS environment and customer SLAs. If the device is not modeled, no alarms appear for that device, making it much harder to figure out the cause of a Path going down. If MPLS traps are not sent to the SpectroSERVER or if these traps are disabled in MPLS Transport Manager, DX NetOps Spectrum may not even detect that the Path is down at all. Therefore, it is always best to model all devices used in your LSP Paths.

How to Add Previously Unmodeled Hops

If an unmodeled device appears in a Path's hops, we recommend that you model that device so that the LSP impact can be calculated for alarms generated by the device. Unmodeled devices do not generate alarms and, therefore, cannot provide impact data, making it more difficult to determine accurately the priority of device outages involving your LSPs.

To get MPLS Transport Manager to display the device details in the Path Hops table, follow these steps:

1. Model the device in OneClick using the IP address provided in the Path Hops table.

NOTE
We recommend that you model all MPLS Path nodes on a single server in a standalone SpectroSERVER environment. In a distributed SpectroSERVER environment, verify that the node is modeled in at least one of the landscapes, with the correct community name.

2. Rebuild the MplsPath model. This task requires the following two steps:
   a. Update the MPLS Path data by searching for and deleting all MplsPath models that contain the unmodeled hop. This step is required because rediscovering MPLS devices does not automatically display the newly modeled device in the list of hops. The MplsPath model must be rebuilt to update that device model information.

      NOTE
      You can find paths with unmodeled hops by searching for all Paths and sorting the results by "Unmodeled Hop Count" in descending order.

   b. Run MPLS Transport Manager discovery. The MplsPath models are recreated, and the newly modeled device information appears in the Path Hop table.

Model Names

During discovery, MPLS Transport Manager uses the following unique model types to model your MPLS environment:

- LspHead
- LSP
- MplsPath

These model types help to determine the names of your MPLS models. For LSP models, MIBs determine the names. Therefore, these names are not editable within MPLS Transport Manager. For LspHead models, the model name is initially patterned after the LSP head-end device during discovery, using the device model name as a prefix. Likewise, the MplsPath model name is patterned after the LSP name. This naming scheme helps to identify the relationships between these MPLS models and their related devices.

If a device or LSP model name changes and the corresponding LspHead or MplsPath model name uses its prefix, the LspHead or MplsPath model name automatically changes to match. For example, if you have a head-end device named "juniper2300," the LspHead model name could be "juniper2300.example.com." If the device name changes to "juniper2300_03," the LspHead model name becomes "juniper2300_03.example.com."

However, you may want to apply a unique name, such as a name that corresponds to a primary customer or region for the LSP. You can edit the LspHead and MplsPath model names manually, and these names do not change automatically. For example, you can manually rename the LspHead model name from "juniper2300_03.example.com" to "Router_NY_03." If the name of the device in this example changes, the LspHead model name will not.

Modify an LspHead or MplsPath Model Name

The LspHead model name is initially patterned after the LSP head-end device during discovery, and the MplsPath model name is patterned after the LSP name. However, you can apply a unique name, such as a name that corresponds to a primary customer or region for the LSP. You can edit the LSP Head and Path model names manually, and these names will not change automatically.

NOTE

The LSP model names are not editable, because MIBs determine these names.

To modify an LspHead or MplsPath model name

1. Open the main MPLS Transport Manager page.
   The main MPLS Transport Manager details page opens in the Contents panel.

2. Locate and click the LSP Head-End or Path on the Explorer tab in the Navigation panel.
   The details display in the Contents panel and Component Detail panel.
3. Click the Information tab in the Component Detail panel.
4. Click the 'set' link for the model name that you want to modify.
   The current name becomes editable.
5. Type the desired model name into the field and press Enter.
   The selected model name is modified.

Deleting Models

Models can be deleted from OneClick at any time for several reasons, and deleting these models can have various implications. First, deleting a head-end model also deletes all models underneath in the MPLS Transport Manager model hierarchy. An MPLS rediscovery is required to restore those models, if needed.

Second, an administrator who does not monitor your MPLS environment can decide that a router in his landscape does not need monitoring, so the administrator deletes the device model for that router. If that device model is the ingress router for one or more LSPs, the LspHead model is deleted. In this case, all LSP and Path models below it are also deleted, as described in the first scenario.

Some of these deleted models may have been used in LSPs or Paths, and the MPLS Transport Manager administrator can decide to restore those models. For example, a device model that appears in a Path's hops can be deleted and appear as an unmodeled hop, making it difficult to monitor the performance of the LSP that includes that Path.

Monitoring Performance and SLAs

This section explains the information you need to monitor the performance of your MPLS infrastructure and check your adherence to customer SLAs using MPLS Transport Manager. This section is intended for general MPLS Transport Manager operators.

Monitoring SLAs for MPLS Environments

For an ISP, continuity of service is crucial for each customer, which is why client SLAs are often established. Although all service outages are not avoidable, the customer relies on the ISP to keep the outages to a minimum, adhering to the terms agreed upon in their SLA.

Monitoring an MPLS environment for adherence to customer SLAs can be difficult. MPLS Transport Manager makes this task easier by monitoring the performance of the MPLS devices and providing a way to view the relationship of those devices to your LSPs and Paths. Depending on your needs, you can monitor SLAs in one of the following ways:

- Analyze your LSPs for performance
- Determine if LSP service is interrupted
- Determine which customers are affected by outages or poor performance
- Search for a customer's LSPs to monitor their health

Some of these monitoring tasks are proactive (such as searching for specific LSPs) and some are the result of an alarm triggered by a device in your MPLS environment.

Although SLA information is not maintained in MPLS Transport Manager, you can use the information you have about your SLAs, such as the specific traffic-engineered LSPs used by a client, to determine which customers are impacted by an alarm involving an MPLS device.

Analyzing LSP Performance

To measure the performance of your MPLS environment and the LSPs in the environment, you must analyze the devices and interfaces used by the LSPs. Their status helps to determine the overall health of your MPLS environment and to decide if you must make changes to improve the performance.

The areas that you can monitor for LSP performance include the following:

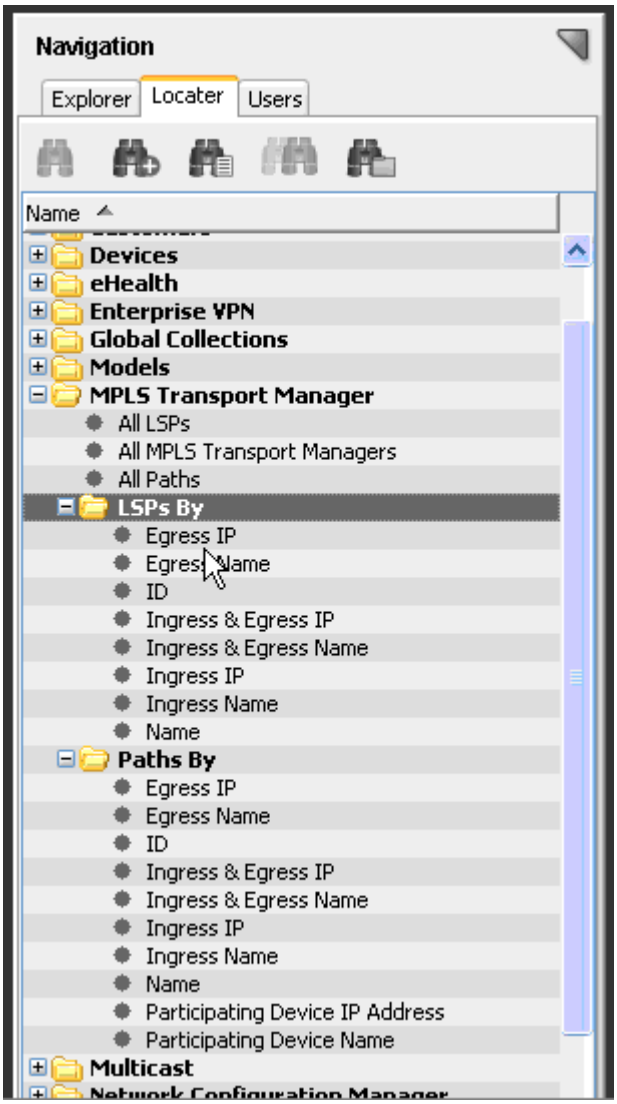
- **Path changes** -- Excessive Path changes can indicate a problem with a device or network connection in your LSP. For example, an LSP may switch to the secondary path 90 percent of the time because a router on the primary path is continually at maximum capacity. In this case, you can analyze the devices and interfaces in the Path to determine the cause and troubleshoot the problem before a service outage occurs on that LSP. MPLS Transport Manager lets you specify a maximum threshold of Path changes for an LSP. If the threshold is breached, an alarm is generated.
- **Device outages** -- If a device outage occurs and causes an alarm, you need to know quickly if the outage has affected any LSPs in your environment. If these devices are modeled in MPLS Transport Manager, the alarm details display information about which LSPs are affected.
- **Alarms** -- All alarms generated from the MPLS environment can provide insight into performance glitches. The impact of alarms generated for device and interface models that affect LSPs is determined by a combination of information, such as which LSP paths are affected (that is, primary, secondary, and so on) and the number of LSPs impacted by the alarm. These alarms can help you drill into the LSP details required to analyze performance issues.

  NOTE
  Only device and interface alarms provide impact weight information. Alarms against the Paths, LSPs, and LspHeads do not have the LSP impact values.

Searching

MPLS Transport Manager does not provide a topology view of your MPLS environment, so using the search feature can help you find details that help you monitor the performance. Searches can locate specific components of your MPLS core, such as locating LSPs by ingress IP or all Paths that use a specific device. These types of searches can help you investigate information related to a specific customer, because you can use details associated with your customer SLAs in your searches, such as specific device or Path names.

The search options are grouped under the MPLS Transport Manager folder, as shown:



For example, if you know the IP address or name of a specific router, you can search for all LSP Paths that use it. Using the list of Paths affected by the router can be useful when performing scheduled maintenance. You can change your Paths to use a different router, or be sure that at least one secondary Path is provided for each LSP before the maintenance. Also, you can give prior notice to any customers affected by those Paths.

Search the MPLS Environment

When analyzing the performance of your MPLS environment, you can quickly locate a specific LSP, Path, or find a group of LSPs, such as all LSPs by ingress IP. These searches help you to view only the details required to effectively monitor the performance of your MPLS network or monitor your customer SLAs.

To search within your MPLS environment

1. Click the Locater tab in the OneClick Navigation panel.
2. Expand the MPLS Transport Manager folder and double-click the type of search you want to perform.
   If no additional search criterion is required, the search results display in the Contents panel. If more criteria is required, a Search dialog opens.
3. Enter your search criteria and click OK.

The search results display in the Contents panel.

Locater Tab for MPLS

In addition to navigating your MPLS environment details on the Explorer tab, you can also perform MPLS Transport Manager searches using the Locater tab in the OneClick Navigation panel. Detailed searches can help you investigate information related to a specific customer, because you can use details associated with your customer SLAs in your searches, such as specific device or Path names.

The Locater tab in the Navigation panel includes the following searches for MPLS Transport Manager information:

- **All LSPs**
  Locates all LSPs that have been modeled in the DX NetOps Spectrum database for the network. Although MPLS Transport Manager does not include the distributed intelligence provided in DX NetOps Spectrum, this option searches multiple landscapes by letting you select multiple SpectroSERVERs in the search parameters.
- **All MPLS Transport Managers**
  Locates all MPLS Transport Manager installations in a list of landscapes you select for the search.

  NOTE
  There can never be more than one MPLS Transport Manager per landscape.
- **All Paths**
  Locates all Paths that have been modeled in the DX NetOps Spectrum database for the network.
- **LSPs By**
  Locates specific LSPs that meet the search criteria for the following search types:
  - Egress IP
  - Egress Name
  - ID
  - Ingress & Egress IP
  - Ingress & Egress Name
  - Ingress IP
  - Ingress Name
  - Name
- **Paths By**
  Locates specific Paths in your LSPs that meet the search criteria for the following search types:
  - Egress IP
  - Egress Name
  - ID
  - Ingress & Egress IP
  - Ingress & Egress Name
  - Ingress IP
  - Ingress Name
  - Name
  - Participating Device IP Address
  - Participating Device Name

Responding to Alarms

This section explains the types of alarms generated for MPLS Transport Manager devices and how to manage them. Although configuring impact weights is an administrator-only task, most of this section is intended for general MPLS Transport Manager operators.

MPLS Transport Manager Alarms

To alert you to problems within your monitored networks, DX NetOps Spectrum generates alarms. When monitoring your MPLS environment, your MPLS Transport Manager models display an alarm state for the following conditions:

- **A device or interface used within an LSP is down**
  Device or interface models generate these alarms when they are reported as down. The impacted LSPs and Paths also display an alarm color and details on the Alarm tab to help you identify the source of the problem. Likewise, the device or interface alarm displays LSP impact details on the Impact tab, such as the following:
  – The number of LSPs affected
  – The total LSP impact of the alarm (that is, the LSP impact value without other types of impact, such as service impact or management lost impact)
  – A few details about the affected LSPs, such as their name, condition, ID, and so on
  In MPLS Transport Manager, the LSP impact weight settings determine how much impact value is added to a device or interface alarm's total impact when an LSP uses that device or interface. Adding these weight values can help determine the priority of your devices or interfaces.

  NOTE
  Only device and interface alarms provide impact weight information. Alarms against the Paths, LSPs, and LspHeads do not have the LSP impact values.

- **Paths are switching too frequently**
  An LSP that is switching Paths too frequently can indicate a network problem for you to address. In MPLS Transport Manager, there are two types of Path change alarms:
  – Aggregate Path change alarms -- The LSP head-end model generates these alarms when the Paths in the LSPs grouped under that head-end model collectively exceed a threshold value for Path changes. The aggregate Path change alarm settings help to determine the threshold value that triggers an alarm for these Path changes.
  – Per LSP (that is, non-aggregate) Path change alarms -- An individual LSP model generates these alarms when Path changes for the LSP exceeds a threshold number within a specified time interval. You can configure the Path change alarm settings for each LSP to determine when the LSP triggers an alarm.

View the LSP Impact of an Alarm

When devices or interfaces used in an LSP trigger an alarm, you can view the impact of the alarm on your LSP. Knowing which LSPs are affected by an alarm can help you determine if a customer SLA is in jeopardy and determine which alarms have the highest priority. To view the impact to your LSPs, you have the following options:

- From an LSP Path model, view the devices or interfaces in an alarm state that are causing it to perform poorly.
- From a device or interface model, view a list of LSPs affected by an alarm.

This procedure describes the first option -- how to view impact details from the LSP Path model.

To use the LSP model to view the impact of an alarm in your MPLS environment

1. Open the main MPLS Transport Manager page.
   The main details page opens in the Contents panel for the selected MPLS Transport Manager.
2. Locate the alarming LSP Path on the Explorer tab in the Navigation panel.
   The LSP Path details display in the Contents panel.
3. Click the Alarms tab.
   Alarms related to the selected LSP Path are displayed.
4. Select an alarm from the table.
   Details about the alarm are displayed in the Component Detail panel.
5. Click the Impact tab.
   The impact details for the selected alarm are displayed.

NOTE
The state of devices on the Impact tab may not reflect the current device state.

How Impact is Determined

Deciding which device and interface alarms cause the greatest impact to your LSPs can be difficult, because the impact depends on how the device or interface is used within your LSP Paths. For example, problematic devices used in an LSP's primary path are much more significant than problematic devices in a secondary path.

The goal of the MPLS Transport Manager impact system is to evaluate and rank device and interface outages in respect to LSPs. For example, a device outage can affect a number of LSPs which would make that device outage more serious than a device outage which affects only one LSP, assuming all LSPs are equal.

NOTE
In the impact calculation, MPLS Transport Manager does not prioritize one LSP as being more important than another. If your SLAs are prioritized, you may need to further calculate the alarms priority by considering your SLA data outside of DX NetOps Spectrum.

To help prioritize the alarms affecting your LSPs, you can assign impact weights for the following situations:

- All LSP Paths are down
- An LSP has switched from using the Primary Path to a secondary Path
- At least one secondary LSP Path is down

When these situations occur, a corresponding impact weight is added to the total impact value for the device or interface alarm. The model with the highest total impact weight shows which alarm has the most impact on the LSPs in your MPLS environment.

NOTE
Only device and interface alarms provide impact weight information. Alarms against the Paths, LSPs, and LspHeads do not have the LSP impact values.

Assuming that you use the default impact weight values, the following example shows how MPLS Transport Manager calculates the LSP impact weights for down devices that are used in your LSPs:

1. Device A is used in the primary Path of only LSP 1. When it goes down, it causes LSP 1 to switch to a secondary Path. The default Switched LSP Weight value (10) is added to the total impact value for the device alarm. The total LSP impact value added for this situation is 10.
2. Device B is used in two secondary Paths for LSP 1. The default At Risk LSP Weight value (5) is added only *once*, even though the device appears in *two* secondary Paths. Only one value is counted per LSP, which is always the most severe impact weight value. So the total LSP impact value added for this situation is 5.
3. Device C is used in a secondary Path for LSP 1 and LSP 2. The default At Risk LSP Weight value (5) is added twice to the total impact value for the device alarm (one for each affected LSP). So, the total LSP impact value added for this situation is 10.
4. Device D is used in the primary Path for LSP 1 and one secondary Path for LSP 2. LSP 1 switches to an unaffected secondary Path, which adds impact weight for the Switched LSP Weight (10). The At Risk LSP Weight value (5) is added for the affected secondary Path for LSP 2. So, the total LSP impact value for this scenario is 15.
5. Device E is used in all Paths for LSP 1, which includes the primary Path and two secondary Paths. Because all Paths are down, the default Down LSP Weight value (100) is added to the total impact value for the device alarm. So, the total LSP impact value added for this situation is 100.

In this scenario, Device E would have the highest priority, because it causes the highest total LSP impact value.

NOTE
LSP impact weights are only one factor contributing to the total impact weight for an alarm. The total impact value for an alarm can include additional impact values caused by other conditions *not* related to your LSP.

Customize Impact Weights for Alarms in LSP

You can assign custom impact weights for three types of LSP problems. These impact weights help determine the alarm level for devices and interfaces used in your LSPs, so you can more quickly identify which devices and interfaces must be resolved first.

> **NOTE**
> Only an administrator performs this task.

To customize impact weights for your LSP alarms

1. Open the main MPLS Transport Manager page.
   The main details page opens in the Contents panel for the selected MPLS Transport Manager.
2. Click the Information tab in the Contents panel.
3. Expand the Configuration section.
4. Expand the LSP Impact subsection.
   LSP impact options display.
5. Click the 'set' link for the following impact weight options that you want to configure:
   - **Down LSP Weight**
     Defines the value added to a device or interface alarm's total impact weight when that device or interface is used within an LSP that is reporting all Paths are down.
     Default: 100
     Limits: Integers greater than or equal to 0
   - **Switched LSP Weight**
     Defines the value added to a device or interface alarm's total impact weight when that device or interface is used within an LSP that has switched from its primary Path to another one.
     Default: 10
     Limits: Integers greater than or equal to 0
   - **At Risk LSP Weight**
     Defines the value added to a device or interface alarm's total impact weight when that device or interface is used within an LSP that is reporting at least one of its secondary Paths is down.
     Default: 5
     Limits: Integers greater than or equal to 0

   The value for the selected option becomes editable.
6. Enter the desired value for the field and press Enter.
   The selected impact weight is customized for LSP alarms.

How Path Change Alarms Work

An LSP that is switching Paths too frequently can indicate a network problem to address. MPLS Transport Manager can monitor the number of Path changes for each LSP, and it can monitor the aggregate Path changes for all LSPs grouped under an LSP head-end device. Understanding the difference between the Per LSP (that is, non-aggregate) and the aggregate method for monitoring Path changes can help you determine which method you prefer for your MPLS environment.

When monitoring **Per LSP Path change alarms**, the individual LSP generates the alarms. This method works as follows:

1. For each LSP you want to monitor, the administrator configures the Path Change Information, which includes specifying the Path Changes Window interval and the Path Changes Threshold value.
2. For each Path Changes Window interval, MPLS Transport Manager performs these steps:
   a. Counts how many times a Path within the LSP switches Paths, either from the primary Path to a secondary Path or vice versa.
   b. Notes the total number of changes at the end of the interval.

NOTE
The number is also added to the # Path Changes field, which represents the number of Path changes since the head-end device was last booted. This value is not used when calculating the Path change alarms.

c. Compares the # Path Changes field with the threshold value selected.
d. Triggers a critical alarm if the number of Path changes is greater than the threshold value.

3. MPLS Transport Manager repeats step 2 for each Path change interval and adjusts the alarm, as needed. Therefore, if subsequent intervals *do not* exceed the threshold value, the alarm is cleared. However, an event is recorded to the event history for each Path change alarm.

When monitoring **aggregate Path change alarms**, the LSP head-end device generates the alarms. This method can generate more than one alarm severity and works as follows:

1. For each LSP head-end device that you want to monitor, the administrator configures the Aggregate Path Change Alarms settings for the MPLS Transport Manager installation. These settings include threshold percentages for three alarm severities and the Path Change Window interval.
2. For each Path Change Window interval, MPLS Transport Manager performs these steps:
   a. Counts the number of Path switches for all LSPs grouped under an LspHead device.
   b. Calculates the percentage of Path changes as follows:
      ```
      (Total Path changes ÷ Total LSPs in the LspHead) x 100 = Percentage of Path changes
      ```
   c. Compares the percentage to the threshold values for each alarm severity.
   d. Triggers an alarm with a severity that corresponds to the highest threshold breached.

      NOTE
      If multiple thresholds are configured with the same value, the highest level of alarm is generated.
3. MPLS Transport Manager repeats step 2 for each Path change interval and adjusts the alarm as needed. Therefore, if subsequent intervals breach a lower alarm threshold, the alarm severity is adjusted accordingly. Also, if Path changes in an interval do not exceed any threshold values, the alarm is cleared. However, events are recorded in the event history for all Path change alarms.

Customize Aggregate Path Change Alarm Settings

LSP Path changes are common, but excessive changes cause an alarm that can indicate a performance problem to address. Customizing the aggregate path change alarm percentages can help you sort these Path change alarms by severity.

To customize aggregate alarm settings for excessive LSP Path changes

1. Open the main MPLS Transport Manager page.
   The main details page opens in the Contents panel for the selected MPLS Transport Manager.
2. Click the Information tab in the Contents panel.
3. Expand the Configuration section.
4. Expand the Aggregate Path Change Alarms subsection.
   Path change alarm options display.
5. Click the 'set' link for the following options that you want to configure:
   – **Minor Threshold %**Defines the threshold percentage for Path changes that triggers a minor alarm. The percentage of Path changes refers to the ratio of Path changes to the total number of Paths within an LspHead model during a single Path Change Window interval.
   Default: 3
   Limits: Integers 0-100

   NOTE
   A value of 0 causes any Path changes to generate an alarm with this severity.

- **Major Threshold %**
  Defines the threshold percentage for Path changes that triggers a major alarm. The percentage of Path changes refers to the ratio of Path changes to the total number of Paths within an LspHead model during a single Path Change Window interval.
  Default: 5
  Limits: Integers 0-100

  NOTE
  A value of 0 causes any Path changes to generate an alarm with this severity.
- **Critical Threshold %**
  Defines the threshold percentage for Path changes that triggers a critical alarm. The percentage of Path changes refers to the ratio of Path changes to the total number of Paths within an LspHead model during a single Path Change Window interval.
  Default: 10
  Limits: Integers 0-100

  NOTE
  A value of 0 causes any Path changes to generate an alarm with this severity.
- **Path Change Window (sec)**
  Defines the time interval in seconds for which the percentage of Path changes is calculated.
  Default: 60
  Limits: Integers greater than or equal to 30

  NOTE
  Only one alarm is generated for a Path change window interval, even if multiple threshold values are set to the same number. For example, if the Minor Threshold % and Major Threshold % values are both set to 5, only a major alarm is generated if the threshold value is breached. The minor alarm is not created.

The value for the selected option becomes editable.

6. Enter the desired value for the field and press Enter.
   The selected Path change alarm setting is customized.

Customize Per LSP Path Change Alarms

LSP Path changes are common, but excessive changes cause an alarm that can indicate a performance problem to address. Customizing the path change alarm settings for each LSP model determines when the LSP triggers an alarm.

To customize Per LSP alarm settings for excessive LSP Path changes

1. Locate and select an LSP in the OneClick Navigation panel.
   Details for the selected LSP appear in the Contents panel.
2. Click the Information tab in the Contents panel.
3. Expand the Path Change Information section.
   Path change alarm options display.
4. Click the 'set' link for the following options that you want to configure:
   - **Path Changes Threshold**

Defines the threshold number of Path changes allowed for an LSP. If the number of Path changes exceeds this value within the path change window interval, the LSP generates a critical alarm.
Default: 5
Limits: Integers greater than or equal to 0

NOTE
A value of 0 causes any Path changes to generate an alarm with this severity.

– **Path Change Window (sec)**
Defines the time interval in seconds for which the percentage of Path changes is calculated.
Default: 60
Limits: Integers greater than or equal to 30
The value for the selected option becomes editable.

NOTE
The # Path Changes field is informational only. It is a non-configurable field that specifies the total number of Path changes since the LSP ingress device was last booted. This number can give you an idea of the frequency of the Path changes.

5. Enter the desired value for the field and press Enter.
The selected Path change alarm setting is customized.

Dynamic MPLS Correlation

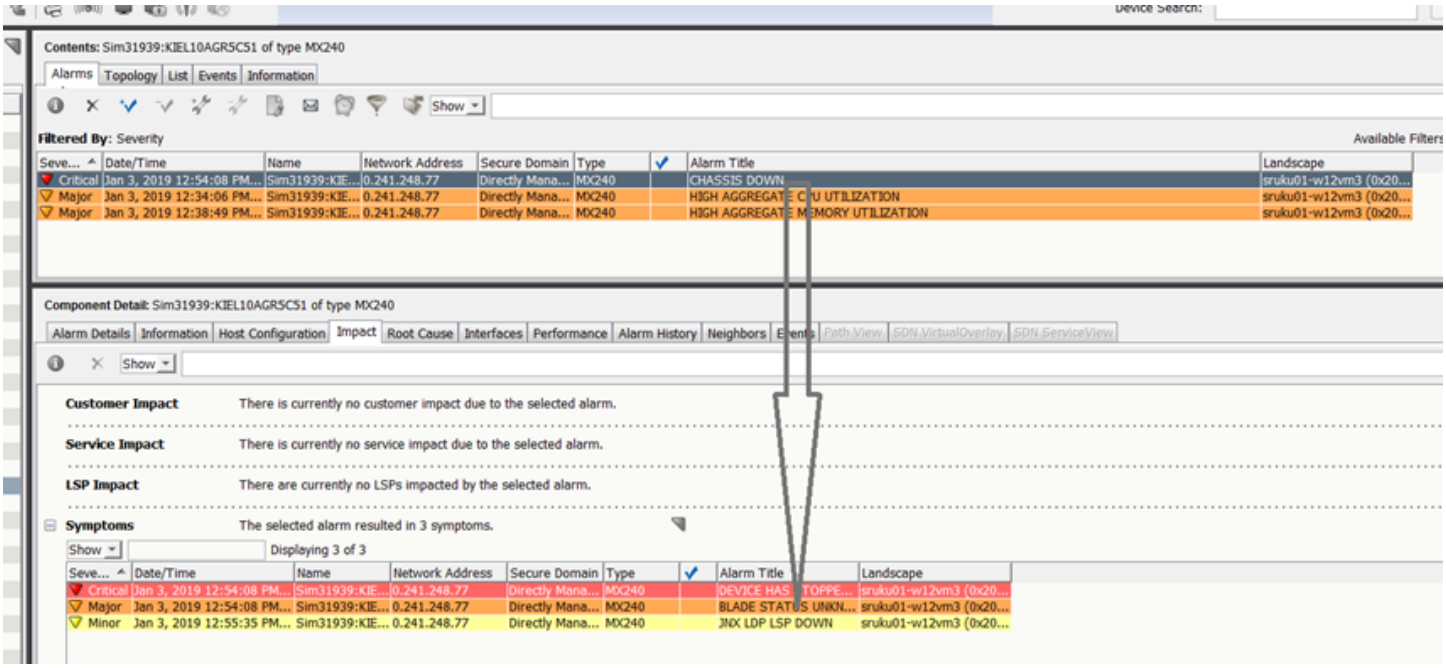
10.3.1 introduces support to address wrongful SNMP trap that is generated when the LSP goes down (as shown in **Fig.1)**. A 'jnxLdpLspDown' trap is received usually when a Juniper device is down, however this trap is generated on the neighboring devices instead of the device it was meant to be generated for. The occurrence has been identified by Spectrum users and now rectified in this new release. This fix ensures that the 'Fec' trap is forwarded from the neighboring device to the rightful device which is down and multiple events concurrent under that trap only, are appropriately displayed.

A 'jnxLdpLspDown’ trap is received in Spectrum when an LSP goes offline. This trap contains the following objects: jnxLdpLspFec jnxLdpRtrid jnxLdpLspDownReason jnxLdpLspFecLen jnxLdpInsta 'jnxLdpLspFec' value contains the device IP which is down and a 'jnxLdpLspDown’ trap is forwarded to the device IP & LSP down alarm is asserted on this device. Device down alarm & LSP down alarm is correlated. There is an LSP down alarm correlation when the chassis goes down.

NOTE

With this change the traps jnxLdpLspUp & jnxLdpLspDown will be processed only if JnprLdpApp is present.

Fig.1.This image describes the prevalent issue which has been fixed in this release:



VPN Manager

VPN Manager lets you discover, model, and monitor Virtual Private Networks (VPNs) within the network environment. VPN Manager provides VPN and Site discovery and modeling, VPN connectivity status, SNMP trap handling, monitoring of Virtual Routing and Forwarding (VRF) conditions, and performs calculation of and alarming on VPN conditions. VPN Manager is a distributed application.

VPN Manager discovers all of the Provider Edge (PE) routers and interfaces that are forwarding traffic for a particular VPN. Based on this discovery, DX NetOps Spectrum creates VPN models representing each unique VPN. These VPN models can then be polled for their current operational status (Up or Down).

In addition, VPN Manager supports VRF Ping and VRF Path Tracing, which enables you to monitor the status of the paths between the sites in each of the modeled VPNs. You can configure threshold alarms to alert you when path changes exceed the configured tolerance.

VPN Manager provides searches that let you quickly find a particular VPN or all VPNs. Search results contain a list of the current VPNs configured within the environment and their current status. You can drill into a single VPN to see the current list of PE interfaces participating within that VPN as well as the current status of the VPN.

VPN Topologies Supported

The common topologies for MPLS Layer 3 VPNs include Full Mesh, Hub and Spoke, and Overlapping VPNs. MPLS VPN Manager supports the following two topologies:

- **Full Mesh** -- The service provider provisions the service so that all customer sites communicate directly with each other. Customer sites are members of the same single VPN. This topology is the most common VPN topologies, and it is generally used by enterprise customers to establish their corporate intranet.
- **Overlapping VPNs** -- Sites can be members of multiple VPNs. Scenarios requiring sites with multiple memberships include the following:

- Establishing a management VPN to customer edge devices
- Establishing a shared service across VPNs
- Implementing an extranet-based service

Overlapping VPN Considerations and Limitations

For overlapping VPN topologies, consider the following notes and limitations when using MPLS VPN Manager:

- Configurations based on route-map statements are not reflected in the mplsVpnVrfRouteTargetTable. DX NetOps Spectrum does not discover or model these configurations.
- The scale of Service Assurance tests in overlapping VPNs becomes more critical because of the greater possibility that connectivity spans multiple VPNs. We recommended enabling Service Assurance tests incrementally to evaluate the performance and resource impacts.
- VPN Sites impact the health of all VPNs in which it has membership. A single VPN Site outage can cause the generation of several alarms (that is, one alarm for each VPN in which the VPN Site has membership). DX NetOps Spectrum does not correlate (that is, suppress) the alarms. Areas affected include:
  - VPN condition calculation
  - VRF test results
  - Aggregate VPN performance (bandwidth usage added to each VPN)
- There is no mechanism to detect when new Route Targets are added, removed, or modified in existing VPNs. There is no background discovery mechanism, and this information is not captured by traps.
- When VPN Sites read some of their configuration from their parent VPN, unexpected behavior can occur when VPN Sites are in multiple VPNs. Configure VPNs similarly for the following options:
  - Enable VRF ping and trace
  - Enable site alarms
  - Model security

MIB Support and Device Compatibility

The MPLS VPN Manager supports:

- Cisco's – MPLS Virtual Private Networks MIB (MPLS-VPN-MIB). Cisco's MPLS-VPN-MIB is based on Draft 3 of the IETF draft MPLS/BGP VPN MIB.
- Juniper’s – Juniper Enterprise VPN MIB; and partially supports Draft 4 of the IETF MPLS/BGP VPN MIB.

From 9.4.1 onwards, – The MPLS VPN Manager supports MPLS-L3VPN-STD-MIB (RFC4382)

NOTE

- – If a CISCO device supports both Draft MIB and MPLS-L3VPN-STD-MIB, MPLS VPN Manager considers the latter for modeling.
  – If a Juniper device supports Draft MIB, as well as the MPLS-L3VPN-STD-MIB. The L3 VPN MIB will take precedence over the MPLS VPN Draft MIB

If models supporting only MPLS-L3VPN-STD-MIB exist in 9.4 or earlier, you need to manually reconfigure those models after upgrading to 9.4.1. This enables the MPLS VPN Manager to discover those VPNs which were not discovered earlier.

For devices which support both Draft MIB and the MPLS-L3VPN-STD-MIB, there is no change in existing functionality. However, we recommend reconfiguring the existing models as it enables the VPN Manager to read the information from the MPLS-L3VPN-STD-MIB.

These MIBs provide access to the following configuration information for VPNs configured on PE router interfaces:

- Virtual Routing/Forwarding (VRF) Instance Table (mplsVpnVrf/mplsL3VpnVrf) - Contains the VPN name and the Route Descriptors (RD) for each VRF.
- VPN Interface Configuration Table (mplsVpnInterfaceConf/mplsL3VpnIfConf) - Associates entries in the ifTable with a VRF.

MPLS VPN Manager uses the following MIBs to support the VRF Ping and the VRF Trace functionality:

- Cisco- RTTMON MIB
- Juniper- RFC2925, Juniper Ping MIB, and Juniper Traceroute MIB

MPLS VPN Manager supports the following tables in the Juniper Enterprise VPN MIB:

- Table of Configured VPNs (jnxVpnTable)
- Table of VPN Interfaces (jnxVpnIfTable)

MPLS VPN Manager functionality is supported by Cisco GSR 12000 and 7500 Series routers running Cisco IOS 12.2(15) T8 or higher in networks with properly configured MPLS VPNs. Juniper support is for JunOS 6.1 or later.

MPLS VPN Manager Interface

Open MPLS VPN Manager

To view the VPNs modeled in your networking environment, open MPLS VPN Manager in OneClick.

To open MPLS VPN Manager in OneClick

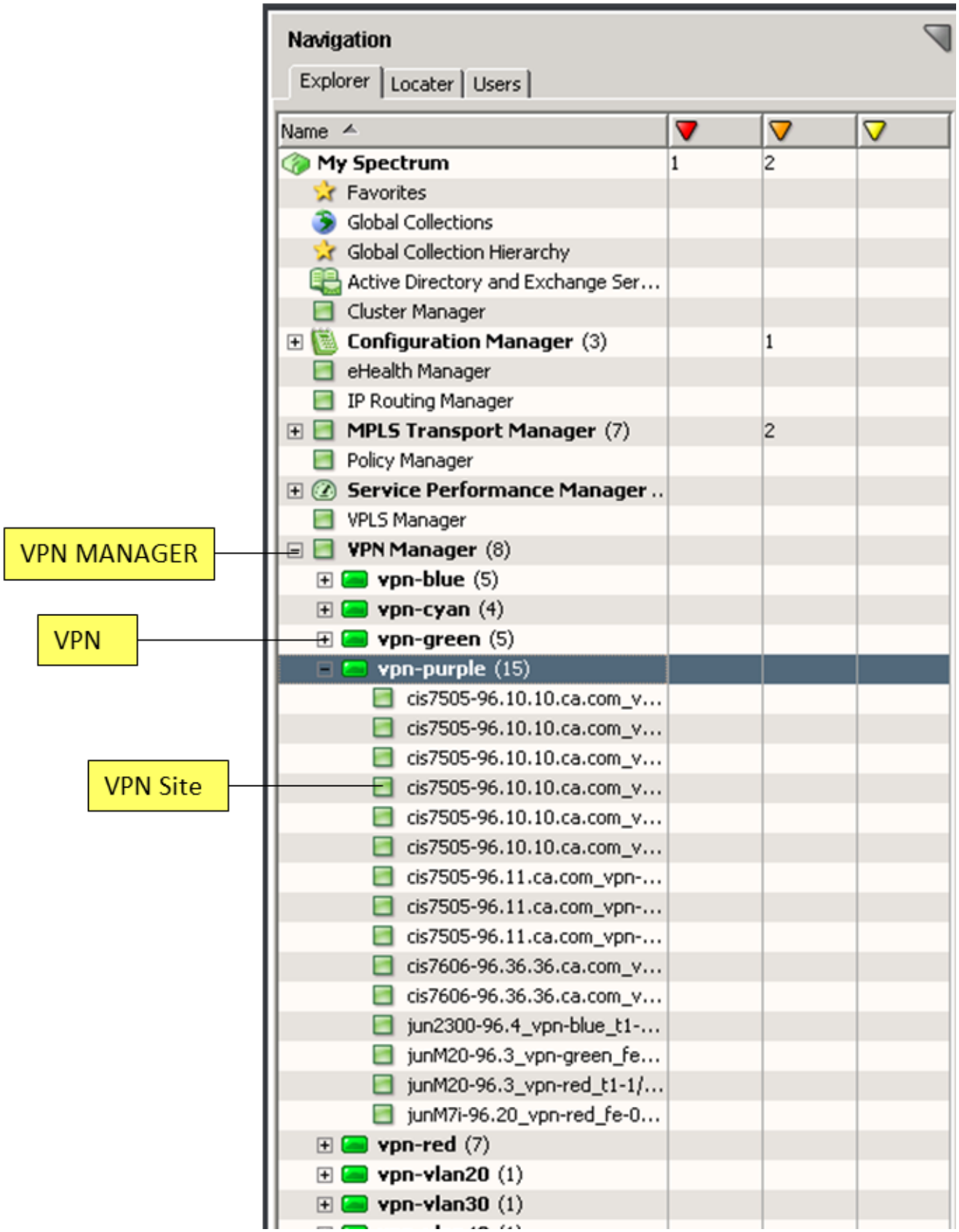
1. Open OneClick.
2. Click the VPN Manager node on the Explorer tab in the Navigation panel.
   MPLS VPN Manager opens, and the Contents panel populates with information about the selected VPN Manager.

MPLS VPN Manager Hierarchy

MPLS VPN Manager can be directly accessed from the Explorer tab of the Navigation panel. Expanding the VPN Manager node displays all of the VPNs managed by the MPLS VPN Manager. Expanding each VPN displays the VPN Sites contained in the VPN.

NOTE
Each VPN Site displays under the VPN in which it has membership. In an overlapping VPN topology, a specific VPN Site may show up under multiple VPNs.

The following graphic is an example of the MPLS VPN Manager hierarchy in the Navigation panel:


VPN MANAGER
VPN
VPN Site
Navigation
Explorer
Locater
Users
Name
My Spectrum
Favorites
Global Collections
Global Collection Hierarchy
Active Directory and Exchange Ser...
Cluster Manager
Configuration Manager (3)
eHealth Manager
IP Routing Manager
MPLS Transport Manager (7)
Policy Manager
Service Performance Manager ..
VPLS Manager
VPN Manager (8)
vpn-blue (5)
vpn-cyan (4)
vpn-green (5)
vpn-purple (15)
cis7505-96.10.10.ca.com_v...
cis7505-96.10.10.ca.com_v...
cis7505-96.10.10.ca.com_v...
cis7505-96.10.10.ca.com_v...
cis7505-96.10.10.ca.com_v...
cis7505-96.10.10.ca.com_v...
cis7505-96.11.ca.com_vpn-...
cis7505-96.11.ca.com_vpn-...
cis7505-96.11.ca.com_vpn-...
cis7606-96.36.36.ca.com_v...
cis7606-96.36.36.ca.com_v...
jun2300-96.4_vpn-blue_t1-...
junM20-96.3_vpn-green_fe...
junM20-96.3_vpn-red_t1-1/...
junM7i-96.20_vpn-red_fe-0...
vpn-red (7)
vpn-vlan20 (1)
vpn-vlan30 (1)

NOTE

You can no longer view Devices listed under VPN Site models in the VPN Manager Explorer view.

The Contents panel displays the Alarm list for the modeled element that you have selected in the Navigation panel. The Component Detail panel displays the Alarm details for the Alarm selected in the Alarm list shown in the Contents panel. If the Alarm list is empty, the Component Detail panel displays the Information view for the modeled element selected in the Navigation panel.

Using VPN Search options in Locater Tab

The Locater tab includes specific predefined searches for VPNs. These searches support cross-server device modeling.

The Locater tab in the Navigation panel includes the following searches:

- **VPN**
  - **All Route Targets**
    Locates all Route Targets that are modeled in the DX NetOps Spectrum database for the selected landscape (Main Location Server, by default) through their Route Target information.
  - **All Sites**
    Locates All Sites that are modeled in the DX NetOps Spectrum database for the selected Landscape.
  - **All VPN Managers**
    Locates all VPN Managers in a list of landscapes that you select for the search in a distributed implementation.

    NOTE
    There can never be more than one VPN manager per landscape.
- – **Site By**
    Locates specific sites that meet the search criteria for the following search types:
  - **Interface**
    Locates and lists all VPN Sites modeled in the DX NetOps Spectrum database based on their interface
  - **Name**
    Locates all VPN Sites modeled in the DX NetOps Spectrum database based on site names
  - **PE Router IP**
    Locates and lists all VPN Sites modeled in the DX NetOps Spectrum database based on the specific IP addresses of the PE Router which you specify in the Search> Site By dialog box.
  - **PE Router Name**
    Locates and lists all VPN Sites modeled in the DX NetOps Spectrum database based on the specific PE Router Name which you specify in the Search> Site By dialog box.
  - **VPN Model Names**
    Locates and lists all VPN Sites modeled in the DX NetOps Spectrum database based on the specific Model Name(s) which you specify in the Search> Site By dialog box.
    If you wish to list the sites present in multiple VPNs, enter the relevant VPN model names (as comma separated values) in the Search > Site By Dialog box.
  - **VPN By**
    Locates specific sites that meet the search criteria for the following search types:
  - **Exported Route Target**
    Locates and lists VPNs based on their Exported Route Target information
  - **Imported Route Target**
    Locates and lists VPNs based on their Imported Route Target information
  - **Interface**

Locates and lists VPNs based on their Interface information

- **Name**
  Locates and lists VPNs based on their VPN Names.
- **PE Router Name**
  Locates and lists VPNs based on their PE Router Name
- **Site Model Names**
  Locates and lists VPNs based on their Site Model Names. A list of common VPNs where there is an intersection of more than one site is displayed.
  If you wish to list the common VPNs with overlapping sites, enter the relevant Site model names (as comma separated values) in the Search > Site By Dialog box.

Follow these steps, to search your MPLS VPN Manager environment:

1. Open OneClick.
2. Click the Locater tab in the Navigation panel.
   The search options are grouped under the VPN folder on the Locater tab, as shown:

3. Expand the VPN node and double-click the type of search you want to conduct.
   A relevant Search dialog opens, based on the parameter selected in a list of landscapes that you select for the search.
4. Follow these steps in the Search dialog box:
   a. For a specific query, enter relevant values in the text box for the search option selected
   b. For multiple searches,
      click 

   c. Follow these steps: in the List of Values dialog:
      - Click Import to import a list
      - Enter a list of values

5. Click OK.
   The search results appear in the Contents panel.

Discovering and Modeling VPNs

To manage the VPNs on your network, you must run VPN Discovery before you can use MPLS VPN Manager. VPN Discovery discovers each VPN and VPN Site currently configured on devices modeled in DX NetOps Spectrum.

Discovery Prerequisites

For VPN Discovery to complete successfully, devices must meet the following prerequisites:

- VPN devices must support the correct MPLS-VPN MIBs.
- You must model the physical components of your network in DX NetOps Spectrum before using the VPN Discovery functionality. You can model the physical components using one of these methods: Discovery, manual modeling, or the Modeling Gateway.

  NOTE
  For instructions about using these mechanisms to model your network, see Modeling and Managing Your IT Infrastructure and Modeling Gateway Toolkit .
- The devices must have MPLS-VPN properly configured.

Configure VPN Discovery Options

Before modeling your MPLS VPN Manager environment, you can select several VPN Discovery options. These options determine how DX NetOps Spectrum finds and models the VPNs in your environment.

To configure the VPN Discovery options

1. Open MPLS VPN Manager.
   MPLS VPN Manager opens, and the Contents panel populates with information about the selected VPN Manager.
2. Click the Information tab.
3. Expand the Configuration, VPN Discovery subview.
4. Click set in the following fields to select your configuration settings:
   - **Model Inactive VPNs**
     Determines whether inactive VPNs in your network environment are discovered and modeled by VPN Discovery.
     Default: No
   - **Enable Dynamic Discovery**
     Determines whether to start the VPN Discovery automatically when a new PE router is modeled. Starting VPN Discovery automatically helps to keep the VPN information current when new devices are added to the network.

     NOTE
     As the MPLS VPN Manager application is running, VPN sites may be created or destroyed when certain traps are received.**Default:** No
   - **Use RD instead of VRF name**
     Determines whether to use the VRF name or Route Distinguisher information from the MIB as the unique identifier when determining VPN membership. This information is used when discovering and naming your VPN site models. By default, DX NetOps Spectrum uses the VRF name. However, if your routers use different VRF names for the same VPN route target, DX NetOps Spectrum creates a separate VPN site model for each new name. In this case, you can avoid multiple models for the same VPN site by configuring DX NetOps Spectrum to use the Route Distinguisher (RD) from the MIB to model your VPN sites.

Default: No

- **VRF/RD Name Filter Type**
  Determines if the VRF/RD names in the 'Global VRF/RD Name Filter' field are included or excluded from modeling. This feature can save unnecessary resources by limiting the number of VPN sites that require monitoring. Options include the following:
  - Exclusive
  - Inclusive
- **Global VRF/RD Name Filter**
  Specifies the VRF/RD names to be included or excluded from modeling. This field is used together with the 'VRF/RD Name Filter Type' field.
- **Default Ping Mode**
  By default, the Ping Mode is set as NoPinging. However, you can select DestinationPinging as the Default Ping Mode for VPN discovery using this feature.

  NOTE
  This option is applicable only to newly discovered sites. This will not update the Ping mode property of existing sites.

You have successfully configured VPN Discovery options.

IfExclusionList

Contents

IfExclusionList is an attribute of the VpnManager (attribute ID: 0x4940185) and is not listed with the VPN Discovery configuration options. This attribute is a text attribute with the default value of 24,131. This value means that by default MPLS VPN Manager does not create VpnSite models for loopback interfaces (ifType=24) and tunnel interfaces (ifType=131).

This attribute can be modified to include additional interface types by using the Attribute Editor in OneClick and adding IfExclusionList as a User Defined attribute.

NOTE
Use the Global_IfExclusionList attribute in Distributed SpectroSERVER environment to apply the changes to all SpectroSERVERs.

Update VPN Models for Overlapping VPN Topology

VPN models created prior to 9.2.1 do not support overlapping VPN topology. DX NetOps Spectrum does not automatically update existing VPN models to support overlapping VPNs. To properly manage your overlapping VPNs in DX NetOps Spectrum, you must manually migrate the existing VPN models.

To update VPN models to use overlapping VPN topology

1. Open MPLS VPN Manager.
   MPLS VPN Manager opens, and the Contents panel populates with information about the selected VPN Manager.
2. Click the List tab.
   The table lists all existing VPN models.
3. Select all VPN models, right-click, and select Delete.
   DX NetOps Spectrum deletes all VPN models and their associated VPN Site models.
4. Run VPN Discovery.
   DX NetOps Spectrum updates your VPN models by recreating the VPN models. These new models support overlapping VPN topologies.

Run VPN Discovery

VPN Discovery is the simplest method of modeling your network. Before you run an on-demand VPN Discovery, ensure that you meet the prerequisites.

WARNING

Before you run VPN Discovery, be sure that all Provider Edge devices are modeled in DX NetOps Spectrum with the read/write community name. MPLS VPN Manager cannot locate your VPNs without these Provider Edge models in the SpectroSERVER.

Follow these steps:

1. Open MPLS VPN Manager.
   MPLS VPN Manager opens, and the Contents panel populates with information about the selected VPN Manager.
2. Click the Information tab.
3. Expand the Configuration, VPN Discovery subview.
   The VPN Discovery options and configurations display.
4. Click Run.
   The Select Landscapes dialog opens, requesting the landscapes on which you want to run VPN Discovery.
5. Select the landscapes and click OK.
   VPN Discovery runs. When complete, the VPN Discovery field status indicator lists the status. Also, a tooltip on this field lists the number of discovered MPLS-VPN devices (single SpectroSERVER) or servers (distributed SpectroSERVER).

Configuring VPN Discovery During Modeling

DX NetOps Spectrum lets you configure Network Services Discoveries including VPN Discovery, during modeling. As a part of modeling configuration, you can specify which Network Service Discoveries to run with the modeling process.

NOTE
For more information, see Modeling and Managing Your IT Infrastructure .

Run VPN Discovery on Selected Models

VPN Discovery is one of the Network Services Discovery options, which are available for models in various OneClick views. Instead of modeling all VPNs in your networking environment, this option lets you quickly model VPNs that are related to selected models in your networking environment.

To run VPN Network Services Discovery on selected models

1. Open OneClick.
2. Select device models related to your VPN environment.
3. Right-click the models and select Tools, Utilities, Network Services Discoveries, VPN Discovery.
   The VPN Discovery process is initiated for the selected models only. You can check the status in the Configuration, VPN Discovery subview for MPLS VPN Manager.

Run VPN Discovery based on Route Target

Overview

You can run the VPN Discovery based on Route Target. Route Target is an identifier that is defined in VRFs for sharing the routes between the VRFs.

Route Target based Discovery benefits:

1. Sites are discovered based on its association with the Route Target. So, VPN Manager recognizes incorrect Route Target usage/ Route Target-Site configuration.
2. Overlapping VPN calculation is done considering both VRF/Site and Route Target/Site associations. This action avoids huge number of sites getting wrongly associated with VRFs.

NOTE

Hub/Spoke configuration functionality is not supported with Route Target based Discovery.

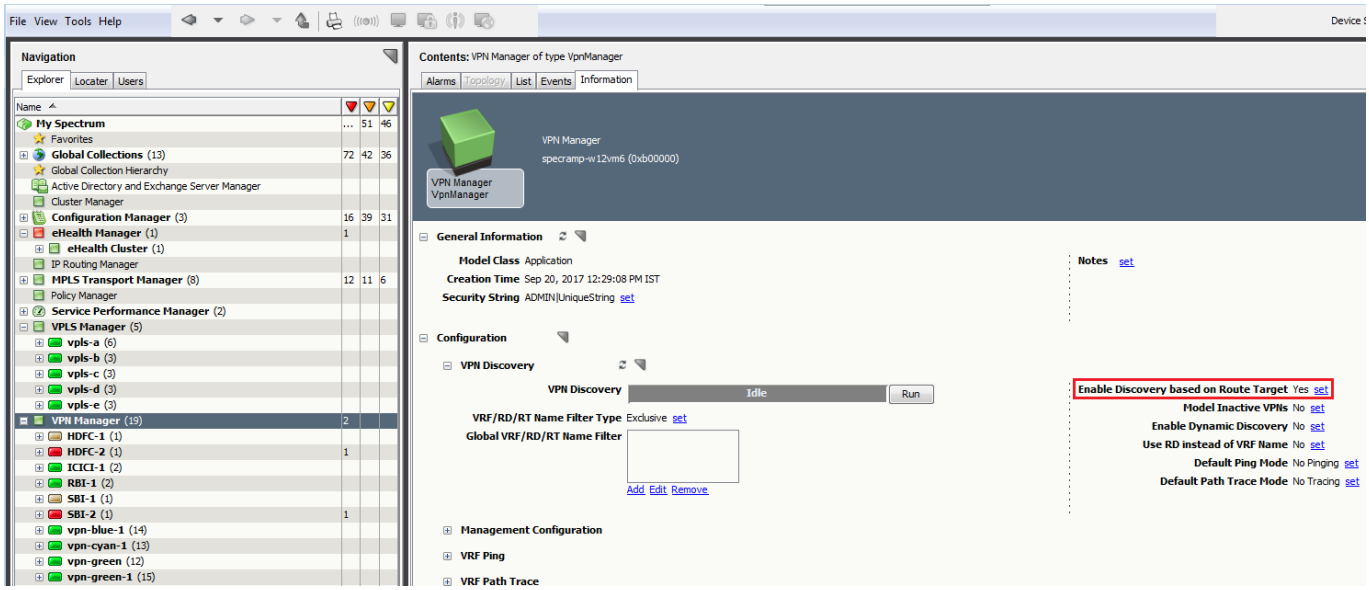
Enable Discovery based on Route Target

You can configure the Route Target based discovery by enabling the 'Enable Discovery Based on Route Target' option in the VPN Discovery sub-view.

When you enable this option, the VPN Manager collects the sites in a Route Target Container based on Route Target/Site association. Associations between VRF and sites are also created based on the Route Target definitions.

Follow these steps:

1. Open OneClick.
2. Click the VPN Manager node on the Explorer tab in the Navigation panel.
3. Click the Information tab in the Contents panel.
   Expand the Configuration, VPN Discovery sub-view.



4. Click 'Set' for the 'Enable Discovery based on Route Target' option.
   Your VPN Discovery option is configured.

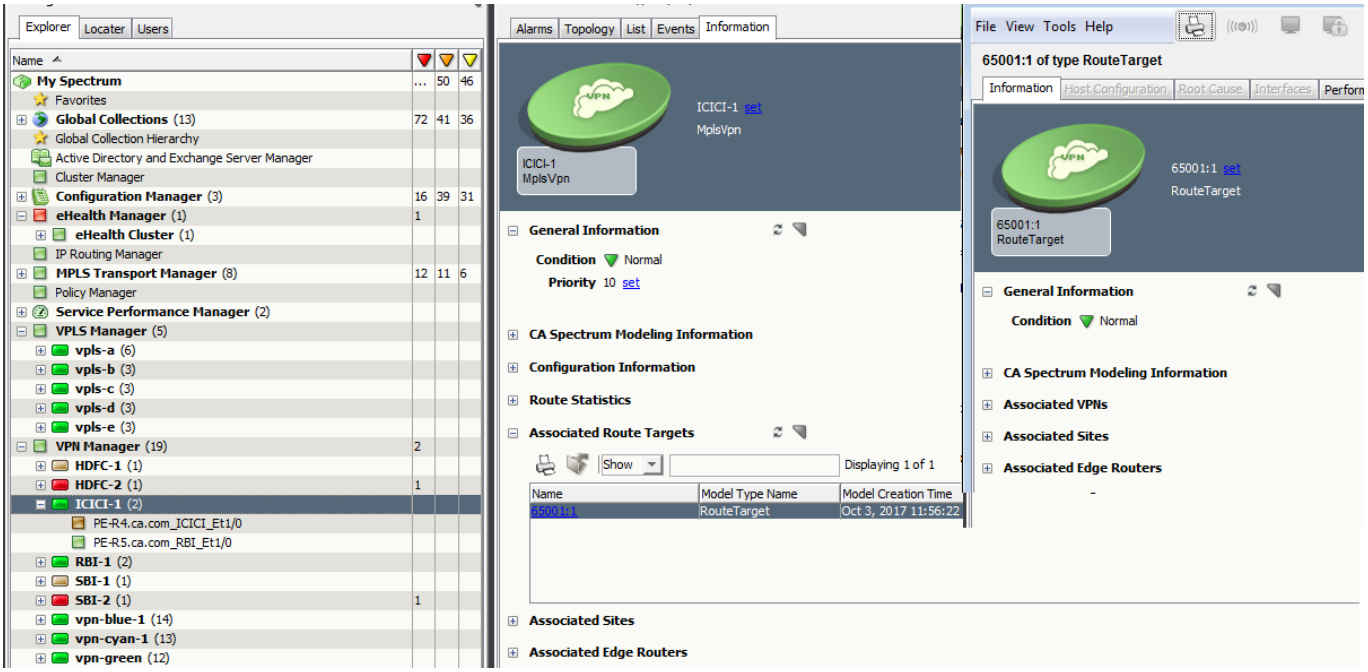
NOTE

You must delete all the VPN Manager models before running a Discovery when the 'Enable Discovery based on Route Target' option is enabled or disabled to switch between the Discoveries.

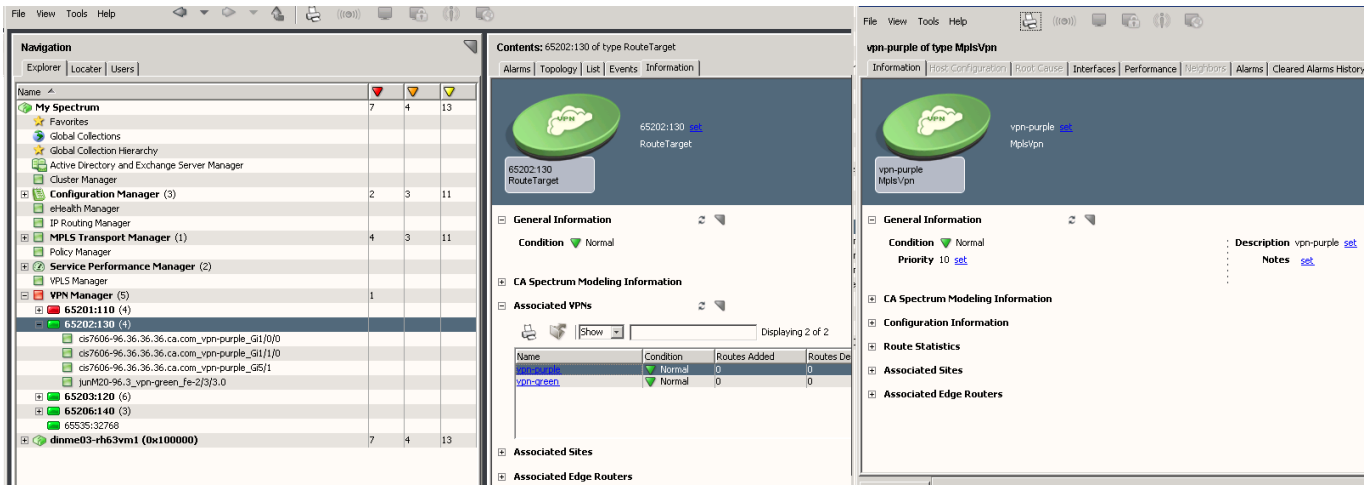
Route Target Model

Starting from the 10.2.2 release, the VPN Manager hierarchy for Route Target based discovery is modified in the Explorer view. The Explorer view now lists the VPN models in the navigation tree instead of Route target models. This

enhancement, the VPN condition alarms that are being asserted on the VPN Manager model are asserted on the VPN model. The VPN Manager hierarchy displays VPN models and associated Route Targets. When a VPN model is selected, the Information tab contains all the information for a selected VPN model. In the Information Tab, 'Associated Route Targets' sub view lists the Route Target type associated with the VPN model as links. When you click the Route Target links, the Component Detail view of the selected Route Target is displayed.



(For 10.2.1 release) The VPN Manager hierarchy displays Route Target Name and associated Sites. When a Route Target (RT) model is selected, the Information tab contains all the information for a selected Route Target model. In the Information Tab, 'Associated VPNs' sub view lists the VPNs associated with the Route Target as links. When you click the VPN links, the Component Detail view of the selected VPN is displayed.



When you select an RT model, the Information tab contains sub-menus which are described in the following sections.

General Information

The General Information section provides you with some basic information about the Route Target model.

Condition

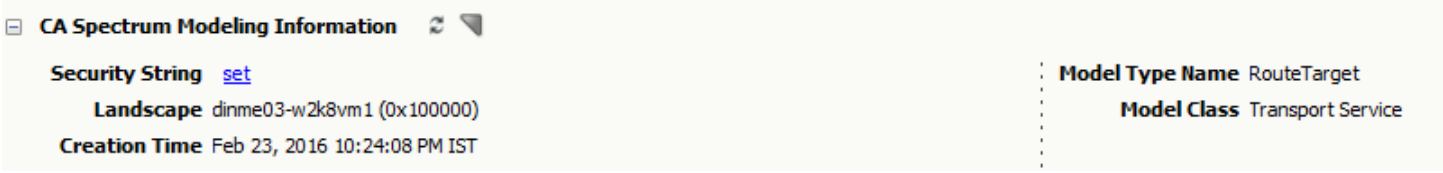
The current condition of the RT.

Notes

You can use this field to save notes about the RT model. Click set, type the notes into the field that is provided, and click Save to save the notes.

Modeling Information

The Modeling Information section provides you with RT modeling information in DX NetOps Spectrum.



Security String

The security string for the RT model.

Landscape

The DX NetOps Spectrum landscape to which the RT model belongs.

Creation Time

The date and time that the RT model was created.

Model Type Name

The RT model name.

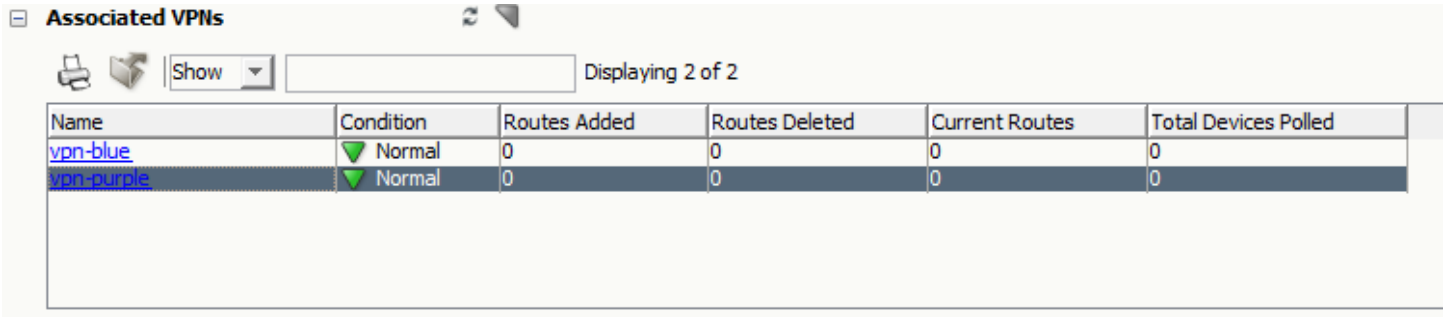
Model Class

The model class of the RT model.

Associated VPNs

The Associated VPNs section provides you with information about the VPNs which are associated with the selected Route Target Model.

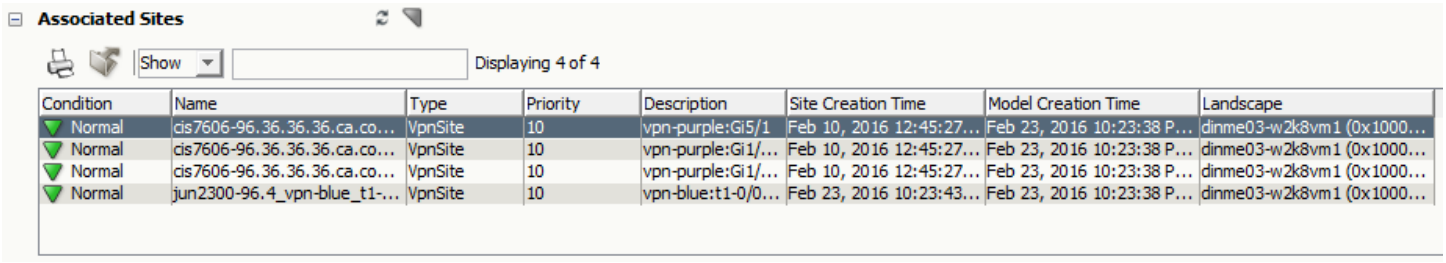
This table lists the VPNs and their information that is associated with the selected RT Model.



Associated Sites

The Associated Sites section provides you with information about the Sites which are associated with the selected RT Model.

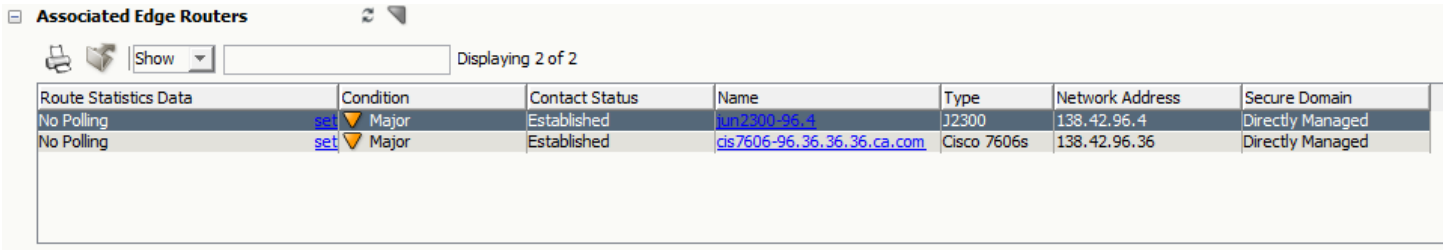
This table lists all the Site models that are associated with the selected RT Model.



Associated Edge Routers

The Associated Edge Routers section provides you with information about the Edge Routers which are associated with the selected RT Model.

This table lists all the Edge Routers that are used by the selected RT Model.



Calculating the Condition of a Route Target

MPLS VPN Manager calculates condition of Route Target based on the following conditions.

1. When condition of site associated with the Route Target model changes, condition of the VPN associated with site changes based on the threshold configuration. VPN condition is then rolled up to the Route Target model and “RT Condition change” (0x4940524) event is generated which has VPN name from which condition is rolled up to the Route Target model
2. VPN condition is rolled up to the Route Target model only during the Site Up/Down condition rollup.
3. VPN condition changes due to VRF Pings are not rolled up to the Route Target model

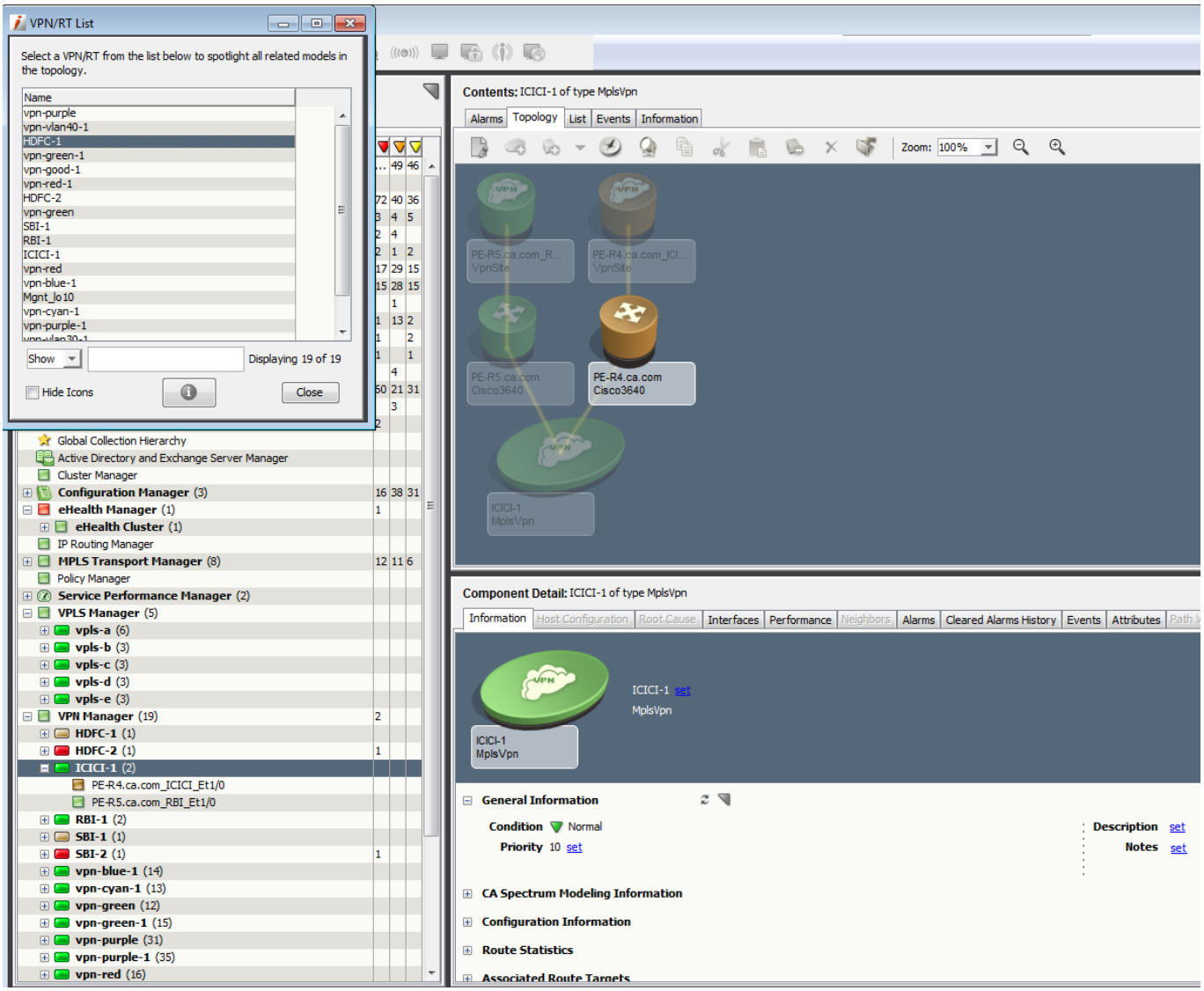
Spotlighting a Route Target

Use the OneClick Spotlight feature to see all models that are related to a Route Target in the Topology view. Spotlighting Route Target in the Topology view helps you more easily determine relationships between Route Target models and the network devices.

Follow these steps:

1. Open OneClick.
2. Expand the desired landscape on the Explorer tab and select Universe.
   Details about the selected Universe appear in the Contents panel.
3. Click the Topology tab.
4. Click (Spotlight View) and select VPN/RT List.
   The VPN List dialog is displayed.
5. Select a VPN or RT from the list.

DX NetOps Spectrum spotlights the selected devices that are configured with selected VPN/RT. Other models are greyed-out in the topology view.



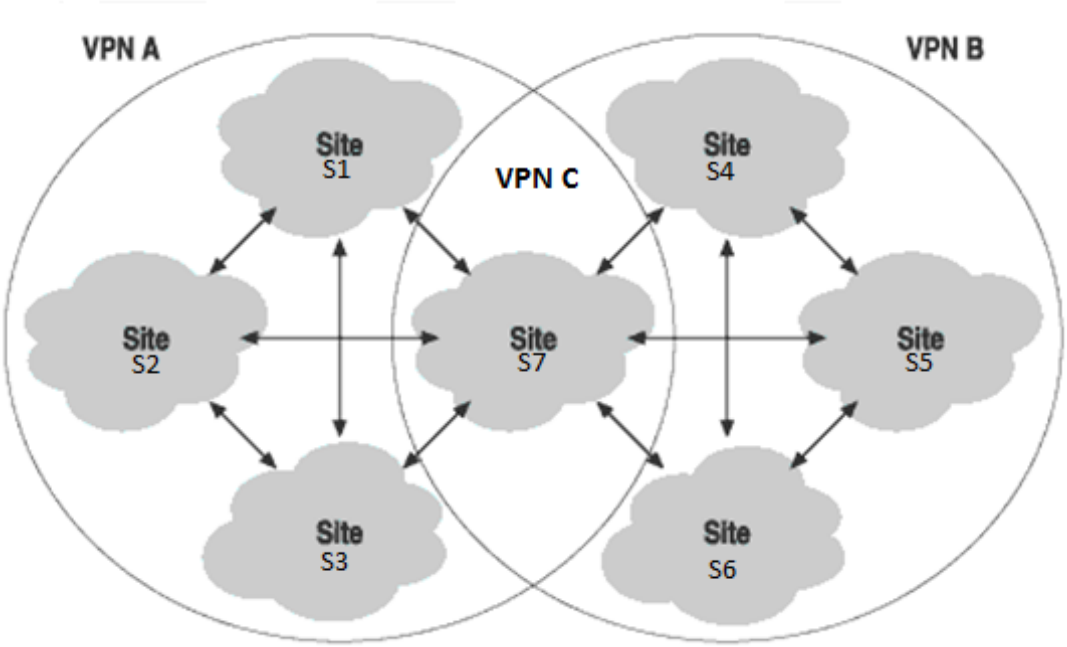
Example Scenario

This section describes a sample configuration and how Spectrum discovers VRF, RT, and Sites when Route Target based discovery is enabled. The following configuration is taken for three devices.

DEVICE 1	DEVICE 2	DEVICE 3

VPN A
{
S1
S2
S3
vrf-import 111:11
vrf-export 111:11
}
	VPN B
	{
	S4
	S5
	S6
	vrf-import 222:22
	vrf-export 222:22
	}
		VPN C
		{
		S7
		vrf-import 111:11
		vrf-import 222:22
		vrf-export 111:11
		vrf-export 222:22
		}

For the preceding configuration, the VPN topology looks as the following image:



Associations between Route Target and Site

Associations are created as shown in the following table are based on the following statements.

- When a Site belongs to VRF that has Route Target defined for importing/exporting the routes, it is considered that the site is using that Route Target.
- When a Site is using more than one Route Target, it is considered as an overlap site for all associated Route Targets.

Route Target	Sites Associated
111:11	S1,S2,S3,S7
222:22	S4,S5,S6,S7

Associations between VRF and Site

Associations are created as shown in the following table are based on the following statements.

- If a VRF (VPN A) has only one Route Target (111:11) defined, all sites in VRF are considered as the own sites of the VRF.
- If a VRF (VPN A) has only one Route Target (111:11) defined, and if that Route Target (111:11) is used by any other VRF (which has more than one Route Target defined i.e.; (VPN C)); then sites from VPN C are added as overlapped sites to VPN A.
- VPN B has only on Route Target (222:22) defined and that is used by VPN C, so sites from VPN C is added as overlapped sites to VPN B.

VRF	Sites Associated
VPN A	S1,S2,S3,S7
VPN B	S4,S5,S6,S7
VPN C	S7

VPN Naming and VPN/Site associations based on Route Target definition

During Route Target based VPN Discovery, VPN sites are associated with VPN models based on the VPN/Route Target association. The following cases are identified to explain how the models and associations are created.

Case 1:

VRF Name	VPN A	VPN A
Site Associated	S1	S2
Route Target Associated	RT-105	RT-110

If VRFs with the same name (VPN A) and different Route Target association (RT-105 and RT-110) are discovered, different model handles for VPN A (RT-105) and VPN A (RT-110) are created. VPN A (RT-105) is named as VPN A and VPN A (RT-110) is named in the format 'VRF name-numerical index' (For example, VPN A-1) as the name VPN A already exists. Respective sites are associated accordingly to VPN models created.

Case 2:

VRF Name	VPN A	VPN Z
Site Associated	S1	S2
Route Target Associated	RT-105	RT-105

If VRFs with different name (VPN A and VPN Z) and same Route Target association (RT-105) are discovered, single model handle is created and named either VPN A or VPN Z based on order of discovery. All sites are associated with the single VPN model handle.

Case 3:

VRF Name	VPN X	VPN Z
Site Associated	S1	S2
Route Target Associated	RT-105 RT-110 RT-220	RT-105 RT-220 RT-110

If VRFs are associated with more than one Route Target, and if the route target definitions are same, single model handle is created and sites from both VRFs are associated with it. For this example, single model handle is created and named either VPN X or VPN Z based on order of discovery. All sites from VPN X and VPN Z are placed under the newly created model handle.

NOTE

When the Route Target definitions are changed on the device, there is no automatic notification like trap to Spectrum about the specific Route Target configuration changes. There is no parameter in MIBS to identify the change in the route target configuration. VPN discovery should be run again so that the VPN models and associations are created/deleted based on the new Route Target configuration read from device MIB.

MPLS-VPN Model Types

DX NetOps Spectrum creates several model types during VPN Discovery to represent different aspects of the MPLS/BGP VPN MIB in MPLS VPN Manager.

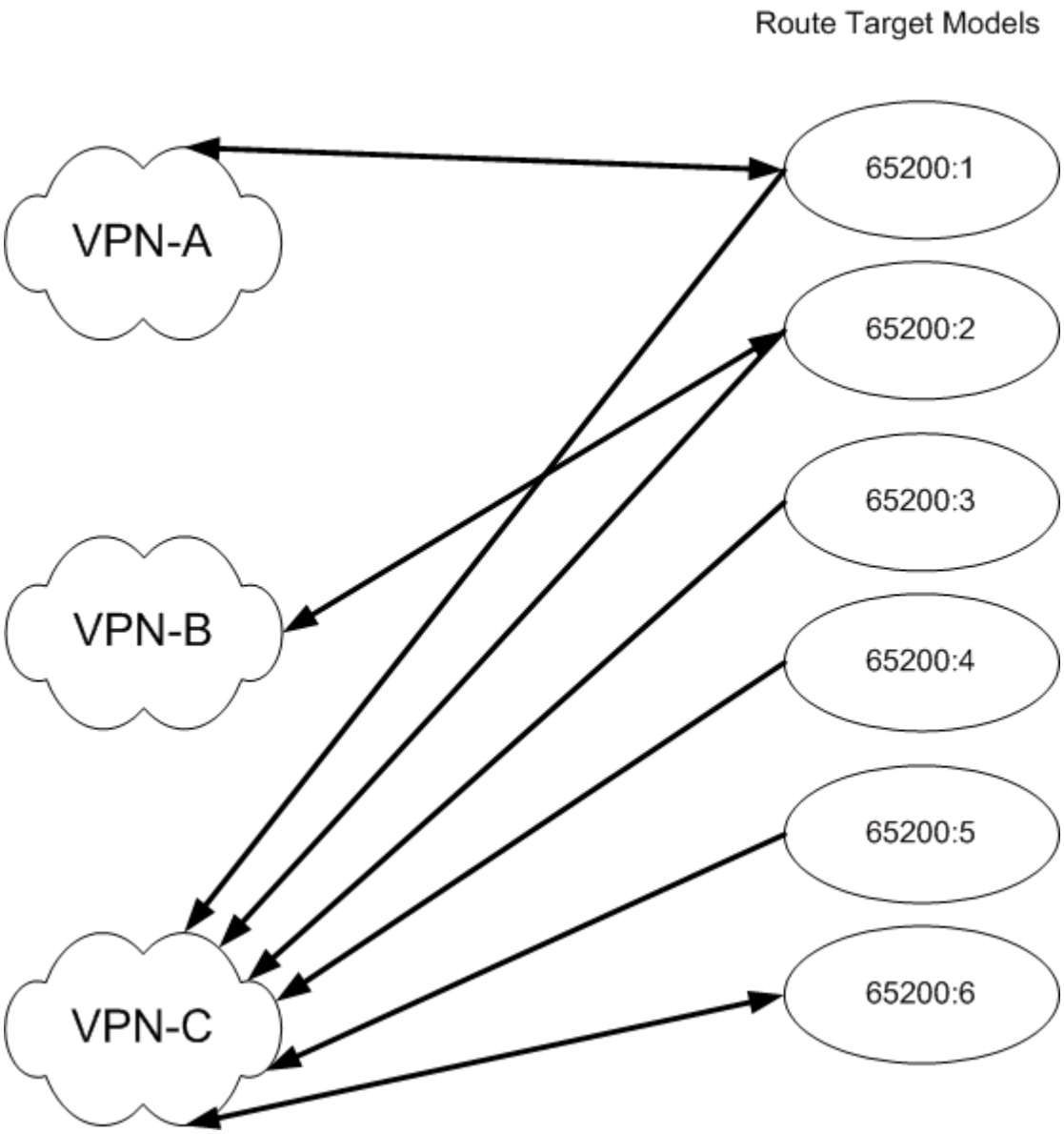
The MPLS VPN Manager model types include the following:

- **VPN Manager**

Represents the MPLS VPN Manager component installed with DX NetOps Spectrum that manages your VPN networking environment. The DX NetOps Spectrum model type for MPLS VPN Manager is VpnManager. This model cannot be destroyed.

- **VPN Site**
  Represents each unique VPN Site that DX NetOps Spectrum discovers during VPN Discovery. DX NetOps Spectrum creates the VPN Site model on the same SpectroSERVER as its associated PE Router. The Model Class attribute for the VPN Site model is set to Transport Service. The DX NetOps Spectrum model type is VpnSite.

  NOTE
  MPLS VPN Manager assumes that each VPN Site is connected to a given PE by a single interface. By default, DX NetOps Spectrum does not create VpnSite models for loopback and tunnel interfaces.
- **VPN Application**
  Represents a modeled device that supports the appropriate MPLS-VPN MIB. This application model must be present for VPN Discovery to successfully discover MPLS-VPN information. The DX NetOps Spectrum model types are MplsVpnApp (for Cisco devices), JNPR_VPN_App (for Juniper devices) and AlcatelMpls_App (for Alcatel). The Model Class attribute is set to Application for these models.
- **VPN**
  Represents each unique VPN that DX NetOps Spectrum models. The Model Class attribute for the VPN model is set to Transport Services. The DX NetOps Spectrum model type is MplsVpn.
- **Route Target**
  Represents the Import and Export of Route Targets. Models of this type do not display in the Navigation view. Instead, they are created on the Main Location Server (MLS). The following diagram shows the Route Target Models created for an overlapping VPN topology scenario:



VPN Site Names

DX NetOps Spectrum generates unique names for VPN Sites during Discovery based on the PE model name, VPN name, and IfName of the interface to which the VPN Site is connected. Using these values helps ensure the VPN Site names in DX NetOps Spectrum are unique.

The VPN Site name follows this structure:

```
ModelName_VPNname_IfName
```

- ***ModelName***
  Represents the name of the PE Router model to which the VPN Site is connected.
  Example: 172.19.38.40
- ***VPNname***
  Represents the name of the VPN to which the VPN Site is connected.

Example: vpn-blue

- ***IfName***
  Represents the IfName of the interface to which the VPN Site is connected.

Example: VPN Site name

DX NetOps Spectrum creates the following VPN Site name when the VPN Site is connected to a VPN named "vpn-blue," a PE Router model named "172.19.38.40," and the interface IfName value is "Fa2/0":

```
172.19.38.40_vpn-blue_Fa2/0
```

VPN Site Model Deletion

A VPN Site model can be deleted in any of the following ways:

- VPN Site models can be manually deleted from the Explorer tab.
- VPN Site models automatically delete when the associated VPN model or Provider Edge (PE) Router model is deleted.

  NOTE
  When a VPN is deleted in an overlapping VPN environment, DX NetOps Spectrum cannot delete the associated VPN Sites with multiple memberships. DX NetOps Spectrum deletes a VPN Site with multiple memberships only when the last VPN referencing the site is deleted.
- VPN models automatically delete when all associated VPN Site models are deleted.
- VPN Site models automatically delete when DX NetOps Spectrum receives an mplsVrfIfDown/ mplsL3VpnVrfDown SNMP trap from the associated PE Router model and the same VRF entry was removed from the PE Router's VRF Table.
- A VPN site model can be associated with multiple device ports.
- A device port model can be associated with multiple VPN site models.
- A VPN Site model will be deleted only if it is not associated with any interface.
- In 10.4.2, stale VPN site models are automatically deleted when no port models are associated with them.

  NOTE
  Previously, such stale VPN site models were not getting deleted. Therefore, if you load a previous database that has stale VPN site entries under VPN Manager, then you need to run the following command from the CLI to delete those entries:

  ./update.exe action=0x4940053 mh= <VPN Manager>

Configuring MPLS VPN Manager

After VPN Discovery, you can configure MPLS VPN Manager to manage the VPN environment effectively. MPLS VPN Manager provides configuration options for VPN Manager models, the individual VPN models, and for VPN Site models, as follows:

- The VPN Manager model configuration lets you specify parameters for all of the VPNs managed by the specified VPN Manager. You can configure VRF Ping and VRF Trace parameters that check on the connectivity of the VPNs in your network and you can configure other parameters relating to how traps and alarms are processed.
- VPN configuration parameters let you set threshold values and other alarm specifics for the VPN Sites within that VPN.
- VPN Site configuration lets you set Trace Mode and Ping Mode, which determine how VRF Path Tracing and VRF Ping should operate within the VPN.

This section describes how to access and set configuration options for MPLS VPN Manager.

Distributed SpectroSERVER Configuration

In a distributed SpectroSERVER (DSS) environment, you can model your PE Routers from any SpectroSERVER. A VPN Site model is created on the SpectroSERVER that its associated PE Router is modeled on. All of the discovered VPNs appear in the VPN Manager area of the OneClick Console. You can enable cross-server Path Tracing and Ping tests to determine the condition of the VPNs that you are managing. The local SpectroSERVERs must have a connection to the Main Location Server (MLS) to support this distributed configuration.

WARNING
An implementation requirement of the distributed VPN Manager is that all MplsVpn models must reside on the SpectroSERVER that is on the MLS machine. When you change the MLS, the MplsVpn and VpnSite models become invalid and are deleted by each SpectroSERVER in the DSS environment.

NOTE
For more information about changing the MLS, see the procedures in Distributed SpectroSERVER Administration .

VPN Manager Configuration

The VPN Manager model manages a set of VPNs on a given DX NetOps Spectrum landscape. The following options are controlled by the VPN Manager model:

- Port polling
- Default model priority
- Traps
- VRF pinging
- VRF path tracing

Configure Port Polling

Port polling must be enabled to update the condition of VPN Site models. But, polling can impact your MPLS VPN Manager performance. To help you optimize your MPLS VPN Manager performance, you can control port polling.

To configure port polling for MPLS VPN Manager

1. Open MPLS VPN Manager.
   MPLS VPN Manager opens, and the Contents panel populates with information about the selected VPN Manager.
2. Click the Information tab.
3. Expand the Configuration, Management Configuration subview.
4. Click set in the following field, select your configuration setting, and click Save:
   – **Enable Port Polling**
   Determines whether MPLS VPN Manager polls ports.
   Default: Yes
   Your port polling option is configured.

Control Default Model Priority

Each model MPLS VPN Manager creates has a priority value. Using priority values, you can quickly sort a list of VPNs or VPN Sites to list those with higher priorities first. During VPN Discovery, DX NetOps Spectrum assigns a default priority value. You can configure the default priority value assigned to new VPN and VPN Site models.

To control the default model priority

1. Open MPLS VPN Manager.

MPLS VPN Manager opens, and the Contents panel populates with information about the selected VPN Manager.
2. Click the Information tab.
3. Expand the Configuration, Management Configuration subview.
4. Click set in the following fields, select your configuration setting, and click Save:
   - **Default VPN Priority**
     Defines the default priority value assigned to newly created VPN models in MPLS VPN Manager. This value lets you sort VPNs in order of their priority to your work.
     Default: 10
     Limits: 0 - 4 billion
   - **Default Site Priority**
     Defines the default priority value assigned to newly created VPN Site models in MPLS VPN Manager. This value lets you sort VPN Sites in order of their priority to your work.
     Default: 10
     Limits: 0 - 4 billion

   Your default model priority values are configured, and VPN Discovery assigns these values to newly created models.

Configure Trap Options

Traps sent from VPN devices can signal MPLS VPN Manager when a VPN device is newly available or no longer available for management in DX NetOps Spectrum. Based on your needs, you can enable or disable these traps from creating or deleting VPN device models.

To configure trap options

1. Open MPLS VPN Manager.
   MPLS VPN Manager opens, and the Contents panel populates with information about the selected VPN Manager.
2. Click the Information tab.
3. Expand the Configuration, Management Configuration subview.
4. Click set in the following fields, select your configuration settings, and click Save:
   - **Create on Trap**
     Determines whether the receipt of a mplsVrfIfUp/ mplsL3VpnVrfUp or jnxVpnIfUp trap results in a new VPN Site or VPN being discovered and modeled by MPLS VPN Manager.
     Default: Yes
   - **Delete on Trap**
     Determines whether the receipt of a mplsVrfIfDown/ mplsL3VpnVrfDown or jnxVpnIfDown trap results in a VPN Site or VPN being deleted from MPLS VPN Manager.
     Default: Yes

   Your trap options are configured.

Configure VRF Ping

MPLS VPN Manager can enable or disable cross-server pinging. Enabling cross-server pinging between domains that are not very related is not recommended. For example, conducting a VRF ping between service customer domains that are highly segmented can reduce performance on the PE Routers. Configuring your VRF ping options lets you optimize your MPLS VPN Manager performance.

To control the default model priority

1. Open MPLS VPN Manager.
   MPLS VPN Manager opens, and the Contents panel populates with information about the selected VPN Manager.
2. Click the Information tab.
3. Expand the Configuration, VRF Ping subview.

Click set in the following fields, select your configuration settings, and click Save:

- **Enable VRF Ping**
  Determines whether MPLS VPN Manager sends VRF pings to monitor VPN connectivity. When set to Yes, MPLS VPN Manager sends VRF pings using RTTMON for Cisco devices and Juniper RFC2925, and using Juniper's Ping extension MIB for Juniper devices and for Alcatel, it is TIMETRA-ICMP-MIB. When set to No, MPLS VPN Manager infers VPN condition from the VPN Site condition.
  Default: Yes
- **VRF Ping Interval**
  Determines the time interval between VRF pings. Setting this value to a higher number reduces network traffic.
  Default: 1200
- **VRF Ping Timeout**
  Determines the time (milliseconds) to wait for a VRF ping to complete.
  Default: 5000
- **Enable Cross-VPN Ping Tests**
  Controls whether VRF pings test connectivity across overlapping VPN boundaries. For example, a VPN Site from vpn-blue normally does not communicate with a VPN Site from vpn-red. If vpn-blue imports the route target from vpn-red, then they *may* communicate. This attribute (EnableTransVpnPing) determines whether MPLS VPN Manager attempts this Service Assurance (SA) test.
  Default: Yes
- **Enable Cross-server VRF Ping Tests**
  Determines whether site-to-site connectivity test occur across multiple SpectroSERVERs. When set to No, site-to-site tests occur on a single SpectroSERVER only.
  Default: No
- **Collapse Bidirectional Ping**
  Determines whether MPLS VPN Manager performs a bidirectional VRF ping. Setting this value to Yes limits the VRF ping to one direction only. Unidirectional VRF pinging works based on the assumption that if one site can successfully receive a response from another, then the network between them is functioning properly. Unidirectional VRF pinging can reduce network traffic by eliminating potentially redundant ping tests.

  WARNING
  Before changing the value of this field, you must stop VRF pings by setting the Enable VRF Ping option to No. After changing the Collapse Bidirectional Ping option, restart VRF pings by setting the Enable VRF Ping option to Yes.

  Default: Yes

Your VRF ping options are configured.

Configure VRF Path Trace

MPLS VPN Manager can enable or disable cross-server path traces. Enabling cross-server path traces between domains that are not very related is not recommended. For example, conducting a VRF path trace between service customer domains that are highly segmented can reduce performance on the PE Routers. Configuring your VRF path tracing options lets you optimize your MPLS VPN Manager performance.

To configure VRF Path Trace options

1. Open MPLS VPN Manager.
   MPLS VPN Manager opens, and the Contents panel populates with information about the selected VPN Manager.
2. Click the Information tab.
3. Expand the Configuration, VRF Path Trace subview.
4. Click set in the following fields, select your configuration settings, and click Save:
   – **Enable Path Trace**

Enables all of the path tracing features for all VPN models managed by the selected VPN Manager model. Setting this option to No disables path tracing.
Default: No

- **Path Trace Interval (seconds)**
  Determines the time interval between invocations of site-to-site path tracing.
  Default: 1200
- **Path Trace Timeout (msec)**
  Determines the time (milliseconds) to wait for a VRF path trace to complete.
  Default: 25000
- **History Limit**
  Defines the maximum number of paths to be remembered between any pair of VPN Sites. To limit memory usage, the maximum History Limit value is 12.
  Limits: 1 - 12
  Default: 5
- **Enable Cross-VPN Path Traces**
  Controls whether VRF path traces cross overlapping VPN boundaries. For example, a VPN Site from vpn-blue normally does not communicate with a VPN Site from vpn-red. If vpn-blue imports the route target from vpn-red, then they *may* communicate. This attribute (EnableTransVpnTrace) determines whether MPLS VPN Manager attempts this Service Assurance (SA) test.
  Default: Yes
- **Enable Cross-server Path Traces**
  Determines whether site-to-site connectivity tests occur across multiple SpectroSERVERs. When set to No, site-to-site tests occur on a single SpectroSERVER only.
  Default: Yes
- **Enable Path Change Alarms**
  Determines whether DX NetOps Spectrum generates alarms when the number of path changes exceeds the thresholds defined by the Critical, Major, and Minor Threshold % values.
  Default: No
- **Critical Threshold %**
  Defines the critical threshold for the percentage of path changes in a given poll cycle. If this value is set to zero, MPLS VPN Manager generates alarms for any path changes.
  Default: 10
- **Major Threshold %**
  Defines the major threshold for the percentage of path changes in a given poll cycle. If this value is set to zero, MPLS VPN Manager generates alarms for any path changes.
  Default: 5
- **Minor Threshold %**
  Defines the minor threshold for the percentage of paths changed in a given poll cycle. If this value is set to zero, MPLS VPN Manager generates alarms for any path changes.
  Default: 3

NOTE
MPLS VPN Manager uses these threshold values for VRF path tracing functionality only.

Your VRF path trace options are configured.

Global Configurations and Local Overrides

In a distributed SpectroSERVER (DSS) environment, the configuration options set on the MLS VPN Manager are applied to the VPN Managers on remote SpectroSERVERs by default. You can override a global configuration on a remote VPN Manager.

NOTE
Use caution when making overrides. Whenever possible, update the model information on the MLS as a best practice and use local overrides for any changes required on a specific SpectroSERVER.

Overriding Global Configurations

You can override a global configuration value by setting a local override on a remote VPN Manager.

To override global configuration values for select VPN Manager

1. Search using the All VPN Managers predefined search.
   The search results appear in the Contents panel.
2. Select the VPN Manager model you want to configure.
   Details about the selected VPN Manager appear in the Component Detail panel.
3. Click the Information tab.
4. Expand the Configuration subview and locate the configuration value to override.
5. Click set for the configuration option.
   The Local Override Panel displays. The Global value is the value set on the MLS VPN Manager.
6. Change the Local value to the desired value, unselect the 'Use global value' check box, and click Save.

   NOTE
   The 'Use global value' check box appears only when the selected value differs from the Global value.

   The global configuration settings are overridden for the selected VPN Manager.

NOTE
When a local override is being used, the attribute value will have an asterisk appended to the end of it in the Configuration subview.

VPN Model Configuration

For each VPN model, you can configure VPN condition alarms. These settings turn alarming on, determine how MPLS VPN Manager generates information about the VPN models, and determine the alarm severity, based on threshold settings. Each threshold is based on the percentage of VPN Sites that are part of the VPN that are “Down.” VPN Sites are determined to be in a “Down” condition either through VRF ping tests or by analyzing the aggregate VPN Site condition for a VPN.

Configuring VPN Condition Alarms

The Component Detail panel's Information tab for a selected VPN model contains the Configuration Information section. Use this section to set the thresholds that determine what, if any, VPN Condition alarms are generated.

NOTE
These threshold values are used for both VRF Ping and the Condition value associated with the VPN Site models. These thresholds are not used by the VRF Trace functionality.

Before accessing the VPN configuration options, you must find the appropriate VPN model. To do this:

1. Perform an All VPNs search or select a VPN model in the Explorer tab and skip step 2.
2. In the Contents panel, select the VPN model you want to configure from the Results list.

Once you have selected the appropriate VPN model, the Component Detail panel's Information tab shows all of the configuration options for the VPN Model. Each of these options are explained in the following section.

To change any configuration value, click the value's set link, change the value, and press Enter.

- **Critical Threshold %**

This value is the percentage of down (unreachable) VPN Sites that will result in the generation of a critical alarm for the model.
default: 15%

- **Major Threshold %**
  This value is the percentage of down (unreachable) VPN Sites that will result in the generation of a major alarm.
  default: 10%
- **Minor Threshold %**
  This value is the percentage of down (unreachable) VPN Sites that will result in the generation of a minor alarm.
  default: 5%
- **Response Time Threshold (msecs)**
  The event threshold for the response time of a successful Ping test.
  default: 250 milliseconds
- **Enable VPN Alarms**
  Enable VPN Alarms is set to Yes by default. Thus when the condition of the VPN model changes, an alarm is generated. When this value is set to Yes and contact is lost to a VPN (with traps properly configured), an event triggers an alarm that is asserted against the VPN Model. When this value is set to No and contact is lost to a VPN, the VPN model turns red, but no alarm is created.
  default: Yes
- **Enable Site Alarms**
  Enable Site Alarms is by default set to No. With this default setting, no alarm is generated when the condition of the VPN Site model changes to critical. When this attribute is set to Yes, an alarm is generated when the condition of the VPN Site changes to Critical.
  default: No

  NOTE
  If a site is participating in multiple VPNs then you need to Enable Site Alarms on the VPN (VRF) which is configured on that site.

- **VPN Manager Path Tracing Active**
  This parameter reflects the value set for the VPN Manager Enable Path Trace parameter.
- **Participate in Path Tracing**
  This parameter enables path tracing for all of the sites in the selected VPN. VPN Manager Path Tracing Active and Participate in Path Tracing must be set to yes in order for path tracing to be active.
- **Enable Path Change Alarms**
  When this parameter is set to Yes, MPLS VPN Manager generates alarms when the number of path changes exceed the thresholds defined in the VPN Manager's VRF Path Trace configuration.

  NOTE
  If you use VRF ping connectivity tests instead of an analysis of the VPN's aggregate VPN Site condition to determine the VPN condition, the number of site-to-site VRF ping test failures is used in place of the percentage of VPN Sites down to define threshold violations. For example, a VPN condition of Good is reported when the number of site-to-site VRF ping test failures is less than the Minor threshold.

VPN Site Model Configuration

VRF Path Tracing and VRF Ping are mechanisms for monitoring your VPNs.

Configure the individual site models within each VPN to be a Source, Destination, Source/Destination, or none of the above. Selecting the most appropriate mode will optimize the communication process to more closely match the role of the sites on your network.

For example, if all sites were placed into SrcDest this would generate a great deal of traffic. For a VPN with 100 sites, there would be tests that are initiated to and from every site resulting in a full mesh of 10,000 tests. A more network-

efficient deployment would be to identify key sites that house critical application servers such as mail, web, or database servers and set only those to Destination.

Configure VRF Path Trace Source and Destination

There are four possible values for Trace Mode:

- **NoTracing**
  Tracing is not enabled for this site.
- **SourceTracing**
  This site can originate a path trace.
- **DestinationTracing**
  This site can receive a path trace from a source.
- **SrcDestTracing**
  This site can either originate or receive a path trace.

The Trace Mode option allows you to select the extent to which a site participates in the automated traceroute functionality. By default all sites are set to NoTracing. Tracing starts only when at least one site is enabled as SourceTracing and another site is enabled as DestinationTracing. These VPN Site models must be members of the same VPN.

You can set the value of the Trace Mode attribute for a particular VPN Site using two methods:

Method 1: Using the Contents Panel

1. Perform an All Sites search.
   When the search is complete, the list of VPN Sites is shown in the Results tab of the Contents panel.
2. Right-click the heading row to display the Table Preferences dialog. Ensure that Trace Mode is checked and click OK.
   The Trace Mode column appears in the list of VPN Sites.
3. Click set to select the value of the Trace Mode.

Method 2: Using the Associated Sites List

1. Select the desired VPN Manager from the Explorer tab.
2. In the Component Details panel, select the Information tab.
3. Select the Associated sites section.
4. Right-click the heading row to display the Table Preferences dialog. Ensure that Trace Mode is checked and click OK.
   The Trace Mode column appears in the list of VPN Sites.
5. Click set to select the Trace Mode.

Configure VRF Ping Source and Destination

There are four possible values for Ping Mode:

- **NoPinging**
  Pinging is not enabled for this site.
- **SourcePinging**
  This site can originate a ping.
- **DestinationPinging**
  This site can receive a ping from a source.
- **SrcDestPinging**
  This site can either originate or receive a ping.

The Ping option allows you to select the extent to which a site participates in the automated ping functionality. By default all sites are set to NoPinging. Pinging starts only when at least one site is enabled as SourcePinging and another site is enabled as DestinationPinging. These VPN Site models must be members of the same VPN.

NOTE
A Default Ping Mode feature has been added to the VPN Discovery Configuration options, you can select the default ping mode from the options in the list.

You can set the value of the Ping Mode attribute for a particular VPN Site using two methods:

Method 1: Using the Contents Panel

1. Perform an All Sites search.
2. When the search is complete, the list of VPN Sites is shown in the Results tab of the Contents panel.
3. Right-click the heading row to display the Table Preferences dialog. Ensure that Ping Mode is checked and click OK. The Ping Mode column appears in the list of VPN Sites.
4. Click set to select the value of the Ping Mode.

Method 2: Using the Associated Sites List

1. Select the desired VPN from the Explorer tab.
2. In the Component Details panel, select the Information tab.
3. Select the Associated sites section.
4. Right-click the heading row to display the Table Preferences dialog. Ensure that Ping Mode is checked and click OK. The Ping Mode column appears in the list of VPN Sites.
5. Click set to select the Ping Mode.

Managing VPNs

This section contains the following topics:

- The VPN Manager Model
- The VPN Site Model
- VPN and VPN Site Performance
- Spotlighting VPNs
- Checking the Status of VPN Paths with VRF Path Tracing
- Calculating the Condition of a VPN
- Trap Support
- Automatically Creating and Deleting VPN Sites
- Threshold Traps

The VPN Manager Model

The VPN Manager model manages a set of VPNs on a given DX NetOps Spectrum landscape. Each of these VPNs contains a series of VPN Sites which are also modeled and managed within the VPN Manager hierarchy. The following steps show you how to view all of the VPNs known to a specific VPN Manager:

1. Perform an All VPNs search.
2. In the Contents panel, VPN Manager displays a list of all VPNs currently known.

Each of these VPNs contain VPN Sites. Both the VPN Models and the VPN Site models display valuable information to assist you in managing the VPNs on your network. The sections in this chapter explain the information available for both VPN and VPN Site models.

The VPN Model

When a VPN model is selected, the Information tab contains all of the configurable options for a selected VPN model. The Information tab contains submenus which are described in the following sections:

General Information

The General Information section provides you with some basic information about the VPN model.

- **Condition**
  The current condition of the VPN.
- **VPN Model Name**
  The VPN model name.
- **Model Class**
  The model class of the VPN model.

  NOTE
  For more information about DX NetOps Spectrum model classes, see SpectroSERVER and DX NetOps Spectrum Databases Overview .
- **Creation Time**
  The date and time that the VPN model was created.
- **Security String**
  The security string for the VPN model.
- **Landscape**
  The DX NetOps Spectrum landscape to which the VPN model belongs.
- **Priority**
  The priority value used for the condition calculation.
- **Description**
  A description of the VPN model that is appropriate to your network. Click set, type the description in the field provided, and press Enter.
- **Notes**
  You can use this field to save notes about the VPN model. Click set, type the notes into the field provided, and click Save to save the notes.

Configuration Information

This section enables you to configure how the VPN monitors certain threshold values that contribute to alarms and the condition of the VPN.

Route Statistics

The route statistics show how many routes have been added to the selected VPN. MPLS VPN Manager calculates these statistics from reading the mplsVpnVrfPerfRoutesAdded, the mplsVpnVrfPerfRoutesDeleted, and the mplsVpnVrfPerfCurrNumRoutes parameters from each device on the VPN. These parameters are added together to produce a VPN-wide total.

For example, when a site is added to a VPN, the mplsVpnVrfPerfCurrNumRoutes and the mplsVpnVrfPerfRoutesAdded parameters are incremented.

These statistics provide a measurement of the current capacity of usage of the VPN.

Associated Sites

This table shows all of the VPN Site models associated with the selected VPN model.

Associated Edge Routers

This table lists all of the edge routers used by the selected VPN model.

The VPN Site Model

When a VPN site model is selected, the Information tab contains all of the configurable options for a selected VPN model. The Information tab contains submenus which are described in the following sections.

General Information

Condition

The current condition of the VPN Site.

Priority

The priority value used for the condition calculation.

Model Class

The model class of the VPN Site model.

> **NOTE**
> For more information about DX NetOps Spectrum model classes, see the Getting Started section.

Model Creation Time

The date and time that MPLS VPN Manager created the VPN Site model.

Security String

The security string for the VPN Site model.

Description

A description of the VPN Site model.

Site Creation Time

The date and time that MPLS VPN Manager created the VPN Site.

Landscape

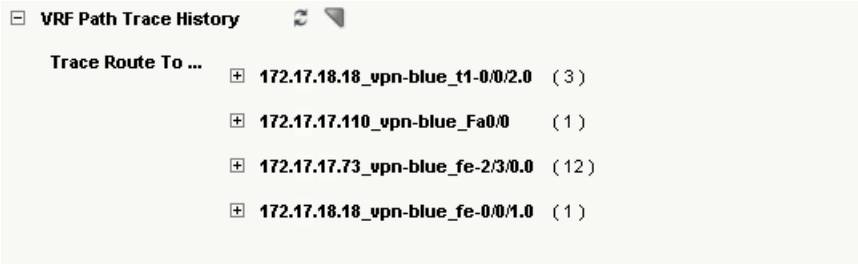
The DX NetOps Spectrum landscape to which the VPN Site model belongs.

Notes

You can use this field to save notes about the VPN model. Click set, type the notes into the field provided, and click Save.

VRF Path Trace History

This section shows the history of all path trace tests from this Site to all peer Sites within the VPN.



Click the + symbol to find specific information about a path trace to the specific site listed as shown in the following graphic.

This menu shows the date and time that the path trace occurred and also shows the IP address of the two VPN sites involved in the path trace. The chart shows specific details about the path between the two VPN sites.

Next Hop Addr

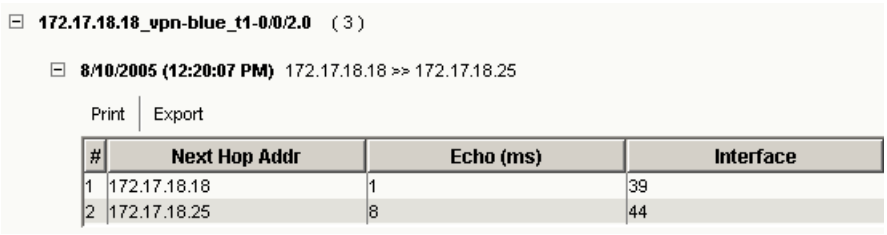
This shows all of the IP addresses of the devices on the path from the originating site to the destination site.

Echo (ms)

The time in milliseconds for the path trace results to be returned.

Interface

The IfIndex value used by the path trace for the device shown in the Next Hop Addr column.



#	Next Hop Addr	Echo (ms)	Interface
1	172.17.18.18	1	39
2	172.17.18.25	8	44

BGP Statistics

The BGP Statistics subview displays details about the BGP peer session involving the selected VPN site device. Details include the following:

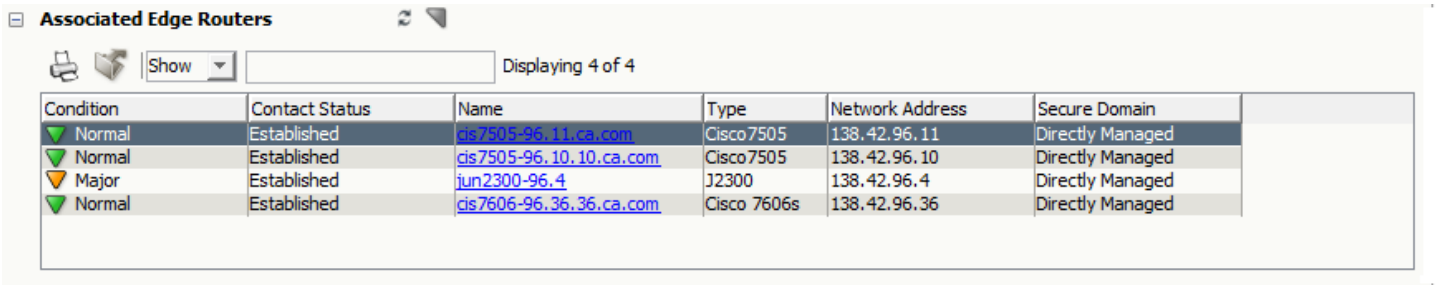
- Local Peer ID
- Remote Peer ID
- Peer State
- Peer Keep Alive
- Peer Hold Time
- Remote AS

NOTE

For more information about BGP peer session monitoring, see Modeling and Managing Your IT Infrastructure.

Associated Edge Routers

The Associated Edge Routers chart shows the edge routers associated with the selected VPN site.

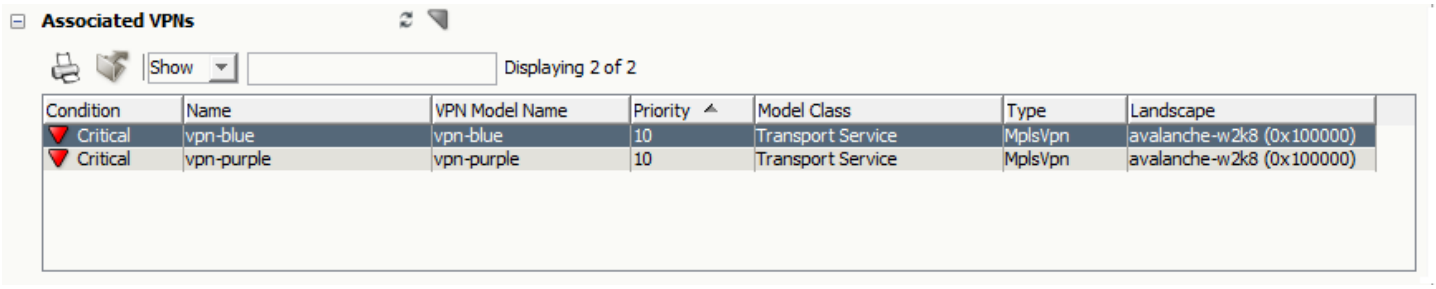


NOTE

If you click the hyperlink in the Name column, you will be redirected to the device location in the respective SpectroSERVER universe.

Associated VPNs

The Associated VPNs chart shows the VPNs associated with the selected VPN site.



VPN and VPN Site Performance

The Performance tab displays performance information for a selected VPN Site or VPN Site Performance. For a selected VPN Site, the performance information is based only on the traffic across the single interface to which the VPN Site is connected. VPN Site performance is a real-time value resulting from the polling of the following interface statistics:

- Bytes In Rate
- Bytes Out Rate
- Out/In Unicast Packet Rate
- Out/In Broadcast Rate
- Out/In Multicast Rate

NOTE

VPN performance data for a selected VPN is an aggregation of the performance data of the component VPN Sites in a VPN.

Spotlighting VPNs

Use the OneClick Spotlight feature to see all models related to a VPN in the Topology view. Spotlighting VPNs in the Topology view helps you more easily determine relationships between VPNs and your network, and between VPNs and other models on your network.

NOTE

When spotlighting in either the VPN or VPN Site Topology views, you can select only a single VPN.

To spotlight a VPN

1. Open OneClick.
2. Expand the desired landscape on the Explorer tab and select Universe.
   Details about the selected Universe appear in the Contents panel.
3. Click the Topology tab.
   The topology of the Universe is displayed.
4. Click the Spotlight View
   icon  .
5. Select VPN List.

   The VPN List dialog opens.
6. Select a VPN.
   DX NetOps Spectrum spotlights the selected VPN by dimming all models in the topology that are not part of the selected VPN.

   NOTE

   To view information about the selected VPN, click the View the Component Detail icon on the VPN List dialog.

Checking the Status of VPN Paths with VRF Path Tracing

MPLS VPN Manager lets you monitor paths within a VPN in two ways. You can enable background path monitoring, which allows paths to be monitored consistently at a preset interval. You can also issue a path trace command on demand to check the path between two VPN sites.

WARNING
You must use a read/write community string when modeling devices in DX NetOps Spectrum for VRF Path Tracing to function properly.

Background Path Monitoring

Background path monitoring traces the paths between VPN Sites based on the configuration options you have chosen. The results of the path traces appear in the VPN Site model's VRF Path Trace History menu.

To use background path monitoring, you must configure the following parameters:

- In the configuration for the VPN Manager model, you must configure the VRF Path Trace parameters. Each of these parameters is explained in the section on VRF Path Trace.
- In the configuration for the VPN model, you must configure the VPN to participate in path tracing, and, if desired, turn on alarms relating to VRF path tracing threshold violations.
- In the VPN Site list, you must configure each VPN Site's trace mode. For instructions see VPN Site Model Configuration.

The VPN Path Trace history results are shown in the VPN Site model's VRF Path Trace History menu in the Information tab.

On-Demand Path Monitoring

An on-demand path trace traces a path between two VPN sites that you have selected. The two VPN sites must belong to the same VPN.

To initiate an on-demand path trace

1. Perform an All Sites search, or locate the desired VPN Site model from the Explorer tab in the Navigation panel and skip step 2.
2. In the Contents panel, locate the VPN Site model you want to configure from the Results list.
3. Right-click the VPN Site model from which you would like to initiate the path trace.
4. Select Select Vrf Trace Start Point.
5. Right-click the VPN Site model that is the destination of the path trace and select Vrf Trace From <VpnSite>. This starts the on-demand path trace.
   The results of the path trace appear in a dialog after the path trace finishes.

Calculating the Condition of a VPN

The overall condition of a VPN is of critical importance to providers of VPN services. VPN Sites impact the health of all VPNs in which it has membership. MPLS VPN Manager provides the following two mechanisms to calculate the VPN condition:

- VRF Ping to determine the state of connectivity
- Condition values determined by alarm threshold values set on each VPN Site model and rolled up to the VPN model

You can choose either mechanism. However, DX NetOps Spectrum cannot use both at the same time. When enabled, VRF Ping takes precedence over the VPN Site condition rollup method. To use the condition rollup method, make sure VRF Ping is disabled.

A single VPN Site outage can cause the generation of several alarms (that is, one alarm for each VPN in which the VPN Site has membership). DX NetOps Spectrum does not correlate (that is, suppress) the alarms. Areas affected include the following:

- VPN condition calculation
- VRF test results
- Aggregate VPN performance (bandwidth usage added to each VPN)

Calculating the VPN Condition using VRF Ping

By default, MPLS VPN Manager uses the state of connectivity between the VPN sites to determine the condition of the VPN. VRF Ping connectivity tests calculate condition for a VPN Site from information obtained from the VRF Table. Using Cisco's VRF Aware Ping (configured through the RTTMON-MIB) and the Juniper Ping MIB extensions to RFC2925, the PE Router can initiate and send pings to any VPN Site on the network. Each entry in the VRF table has information pertaining to one unique Site.

> **NOTE**
> If you use VRF ping connectivity tests instead of an analysis of the VPN's aggregate VPN Site condition to determine the VPN condition, the number of site-to-site VRF ping test failures is used in place of the percentage of VPN Sites down to define threshold violations. For example, a VPN condition of Good is reported when the number of site-to-site VRF ping test failures is less than the Minor threshold. This option is applicable only to newly discovered sites. This will not update the Ping mode property of existing sites.

To ensure that MPLS VPN Manager can use VRF Ping to test for connectivity

1. Verify that you have modeled in DX NetOps Spectrum the devices that are part of the VPN network.
2. Verify that you have used a read/write community string when modeling devices in DX NetOps Spectrum.
3. Verify that Enable VRF Ping is set to Yes on the VPN Manager model.
4. Verify VRF Pings are being run on the device by examining the appropriate MIB on the VPN device (RTTMON for Cisco devices, Juniper Ping MIB extensions to RFC2925 for Juniper device and TIMETRA-ICMP-MIB for Alcatel devices.

5. Verify that the condition of the VPN and VPN Site models is updated correctly and that VRF Pings are successful.
6. Verify the appropriate VPN sites are set as Ping Sources and Destinations.

Calculating the VPN Condition using VRF Ping Response Time Threshold

The result of a VRF Ping test alone informs you of the success or failure of the Ping test. Adding a Response Time Threshold to your VRF Ping tests provides more detailed results than just success or failure.

To ensure that MPLS VPN Manager can use VRF Ping Response Time Threshold

1. Verify you satisfy the requirements listed in Calculating the VPN Condition using VRF Ping.
2. Verify the Response Time Threshold parameter is set to the appropriate value.
3. Verify your Minor, Major and Critical Alarm Threshold % attributes are set to the appropriate values.

Scalability of Ping Tests

The scalability of Ping tests should be considered in large environments where fully meshed testing is performed. Performance testing has shown that full mesh testing beyond 50 sites greatly increases network traffic. The number of Ping tests and the resource requirements can be efficiently managed by organizing your Ping tests.

We recommend that you select a relatively small number of important sites to perform Ping testing. One approach, when the number of sites or remote offices is beyond 50, is to have larger regional offices test back to corporate headquarters or among themselves. For example, in an environment that consists of several regional offices and a corporate headquarters, configuring your corporate headquarters as Ping to Site and your larger regional offices as Ping from Site reduces the network load.

Calculating the VPN Condition using the VPN Site Condition

If you disable VRF Ping by setting the VPN Manager's Enable VRF Ping parameter to No, MPLS VPN Manager can calculate VPN condition based on an analysis of the VPN's aggregate VPN Site conditions. If the Condition value for every Site is “Good,” then the VPN condition is “Good.”

A VPN Site model can have one of four conditions: Initial, Maintenance, Down, or Good. Each of these conditions is explained in the following table. The following values are used to compute the Condition value for a Site:

- Value of ifOperStatus for the physical interface
- Contact status of the PE Router
- Receipt of an mplsVrfIfUp/ mplsL3VpnVrfUp or mplsVrfIfDown/ mplsL3VpnVrfDown trap

VPN Site Condition	Calculation
Initial	No associated IF or IF is initial.
Maintenance Condition	The associated router is in maintenance mode.
Down	IFOperStatus is Down.
Good	All associated IFs are Up.

The condition of the VPN Model is determined by the condition of all of the VPN Sites contained in the VPN. The following list shows how the condition of a VPN is determined.

- **Initial**
  No VPN Sites are modeled or all VPN Site models are “Initial.” The percentage of VPN Sites that are “Initial” is greater than the Minor Threshold.
- **Maintenance Condition**
  All the VPN Site models are in maintenance mode.
- **Minor**

The percentage of VPN Sites down (Site Rollup condition) is greater than the Minor Threshold and less than the Major Threshold.

- **Major**
  The percentage of VPN Sites down is greater than the Major Threshold and less than the Critical threshold.
- **Critical**
  The percentage of VPN Sites down is greater than the Critical Threshold.
- **Good**
  The percentage of VPN Sites down is less than the Minor Threshold.

You can set threshold values for each VPN Model that control whether the VPN Manager generates VPN Condition alarms.

Trap Support- MPLS VPN Manager

The following table lists the MPLS-VPN MIB SNMP traps supported by MPLS VPN Manager. Receipt of either an mplsVrfIfUp/ mplsL3VpnVrfUp trap or an mplsVrfIfDown/ mplsL3VpnVrfDown trap typically results in a change of the VPN Site condition. Events are created based on changes in condition.

NOTE
Each device must be configured to send SNMP traps to the DX NetOps Spectrum VNM machine.

SNMP Trap	Result of Receiving Trap
mplsVrfIfUp/ mplsL3VpnVrfUP	If the VPN Site model already exists, a change in status is reported. Otherwise, a new VPN Site model is created.
mplsVrfIfDown/ mplsL3VpnVrfDown	If both the VPN Site model and the VPN Site on the device sending this trap exist, a change in status is reported. Otherwise, the VPN Site model is deleted.
mplsNumVrfRouteMidThreshExceeded	Event/Alarm
mplsNumVrfRouteMaxThreshExceeded	Event/Alarm
mplsNumVrfSecIllegalLabelThreshExceeded	Event/Alarm
mplsL3vpnVrfRouteMidThreshExceeded	Event/Alarm
mplsL3vpnVrfNumVrfRouteMaxThreshExceeded	Event/Alarm
mplsL3VpnNumVrfSecIllglLblThrshExcd	Event/Alarm
mplsL3vpnNumVrfRouteMaxThreshCleared	Event/Alarm

The following table lists the Juniper traps supported by MPLS VPN Manager.

Trap	Result of Receiving Trap
jnxVpnIfUp	If the VPN Site model already exists, a change in status is reported. Otherwise, a new VPN Site model is created.
jnxVpnIfDown	If both the VPN Site model and the VPN Site on the device sending this trap exist, a change in status is reported. Otherwise, the VPN Site model is deleted.

The following table lists the Alcatel traps supported by MPLS VPN Manager:

Trap	**Result of Receiving Trap**
vRtrMplsLspUp	If the VPN Site model already exists, a change in status is reported. Otherwise, a new VPN Site model is created.
vRtrMplsLspDown	If both the VPN Site model and the VPN Site on the device sending this trap exist, a change in status is reported. Otherwise, the VPN Site model is deleted.
vRtrMplsLspPathUp	If the VPN Site model already exists, a change in status is reported. Otherwise, a new VPN Site model is created.
vRtrMplsLspPathDown	If both the VPN Site model and the VPN Site on the device sending this trap exist, a change in status is reported. Otherwise, the VPN Site model is deleted.

Automatically Creating and Deleting VPN Sites

Currently, MPLS VPN Manager creates a new VPN Site model if a mplsVrfIfUp/ mplsL3VpnVrfUp or jnxVpnIfUp trap is received from a device and a VPN Site model does not already exist for that VPN Site. If a VPN Site model does exist for that VPN Site, MPLS VPN Manager reports a change in its status.

Conversely, if MPLS VPN Manager receives a mplsVrfIfDown/ mplsL3VpnVrfDown or jnxVpnIfDown trap from a device and the VPN Site is already modeled and it exists on the device sending the trap, MPLS VPN Manager reports a change in the status of that VPN Site. If the VPN Site no longer exists on the device that sent the trap, then MPLS VPN Manager deletes the applicable VPN Site model.

NOTE
These actions also depend on the configuration settings for MPLS VPN Manager.

Threshold Traps

MPLS VPN Manager supports threshold traps by creating events and alarms. These traps can be used to monitor the utilization of a PE router:

- mplsNumVrfRouteMidThreshExceeded
- mplsNumVrfRouteMaxThreshExceeded
- mplsNumVrfSecIllegalLabelThreshExceeded
- mplsL3vpnVrfRouteMidThreshExceeded
- mplsL3vpnVrfNumVrfRouteMaxThreshExceeded
- mplsL3VpnNumVrfSecIllglLblThrshExcd
- mplsL3vpnNumVrfRouteMaxThreshCleared
  As the number of VPN Sites increases, these traps indicate that the PE router is approaching its capacity. The mplsNumVrfSecIllegalLabelThreshExceeded threshold trap is used to detect configuration or security violations, such as a PE-CE interface mis-configuration or an attempted VPN break-in using spoofed labels.

The VPN model can be configured to generate alarms based on the condition of the VPN model or VPN Site model.

Model	**Attribute**	**Alarms**
MplsVpn	EnableVpnAlarms	04940405 (Minor) 04940406 (Major) 04940407 (Critical) 04940422 (Maintenance)

MplsSite	EnableSiteAlarms	04940403 (Critical)

Model	Event Condition
MplsVpn	Initial, Maintenance, Minor, Major, Critical, Good
MplsSite	Initial, Down, Good

Troubleshooting MPLS VPN Manager

Common problems and solutions of MPLS VPN Manager are described in the following table:

Description	Solution
MPLS VPN Manager does not currently support the creation of a customer model or the entering of customer VPN information.	You can name a VPN with an associated customer name.
The condition of the VPN Site model may not be updated correctly.	Ensure that Port Polling is enabled in MPLS VPN Manager and that the connections between ports have been resolved by DX NetOps Spectrum. Additionally, traps may be used to determine the condition of a VPN Site. The device must be configured to send traps to DX NetOps Spectrum on behalf of interfaces and VPNs.
A VPN Site alarm is not cleared in cases where the network cable is unplugged and then plugged back in, even though traps are enabled. The mplsVrfIfUp trap is not being sent when the cable is plugged back into the router.	This is a Cisco IOS firmware issue. The mplsVrfIfUp trap is sent in other cases when a VPN Site goes back online (as when the interface is Administratively set to “down” and then back to “up” or when the interface is turned off and then on using Cisco IOS).
A description is added to the VRF configuration on a router after its associated VPN Site has been modeled by MPLS VPN Manager. When you view information for the site in MPLS VPN Manager, the new description is not displayed in either the Description column of a site search Results list or in the Description field of the Component Detail panel's General Information section.	Destroy the VpnSite model, then run a new VPN Discovery that models the site. MPLS VPN Manager then displays the VRF description where appropriate.
A VRF Path Trace will time out with the error message Trace Failed.	Increase the Path Trace Timeout value.
MPLS VPN Manager does not create VpnSite models for each unique VRF entry on Juniper M-Series devices running JUNOS 8.2 R1.7.	Upgrade to JUNOS 8.2 R2.4 or later.
The current implementation creates only a single test model for each source site. This could lead to latency problems (at high site counts) because each test takes a fixed amount of time (typically 5 seconds). With a default poll time of 1200 seconds the system would support approximately 240 sites / vpn. In such cases the number of ping failures might change from one poll cycle to the other.	Increase the VRF Poll interval based on the number of destination sites

Support for Dynamic VPN (DMVPN)

DMVPN (Dynamic Multipoint VPN) is a routing technique that is used to build a VPN network with multiple sites without having to statically configure all devices. It is effective solution for dynamic secure overlay networks. It is a “hub and spoke” network where the spokes communicate with each other directly without having to go through the hub. As a Spectrum user you can now discover, model, visualize the DMVPN topology in Spectrum and Spectrum processes all the DMVPN related traps.

Topology

Spectrum identifies whenever there is a change in the DMVPN topology like a hub or a spoke added to the topology. The DMVPN is a partial dynamic mesh. There are two basic topologies that can be used in the DMVPN networks. The discovered DMVPN networks are modeled under the 'VPN Manager' and on clicking the DMVPN network, the topology displays the hubs and spokes associated with the network.

To discover the DMVPN networks:

1. Navigate to the VPN Manager
2. Run the discovery.

Spectrum will discover the permanent tunnels between the hubs and spokes. The dynamic spoke to spoke tunnels will not be discovered and modeled. The hubs will be positioned on the top and the spokes at the bottom. The label 'Hub' is on the hubs. If there is any hub or spoke added to the DMVPN network, re-run the VPN discovery so that it is visible. Following is the screenshot of the DMVPN topology for a device family.

NOTE

There is no exclusion on the tunnel interface from the VPN Manager.

DMVPN Trap Support

The DMVPN tunnel up and down events will be processed by Spectrum and events will be generated. You will have to configure the devices to send traps to Spectroserver. From the screenshot below, you can see the tunnel up events are received on the Hub device and from the Spoke devices.

Following is a screenshot of the DMVPN Trap Support:

Multicast Manager

Overview

Multicast Manager lets you manage multicast traffic within your network environment. This includes the identification and monitoring of all multicast groups, sources, receivers, and other critical sources like rendezvous point (RP) routers.

Multicast Manager discovers and models Cisco and Juniper devices that support Multicast traffic. Once these devices are discovered, Multicast Manager provides several views which let you monitor the connectivity and performance of these Multicast elements. These views offer details relating to Multicast groups, sources, receivers, routers, and RPs and their relationships to each other.

The information provided in the Multicast Manager views makes troubleshooting Multicast-related outages much more efficient. You can isolate a problem quickly because you can determine the group a receiver belongs to, the RP of the group, the source for the group, the routers and interfaces that a group uses, and the configuration and bandwidth limits of these interfaces. Based on this information, you can prioritize your troubleshooting efforts should multiple problems occur.

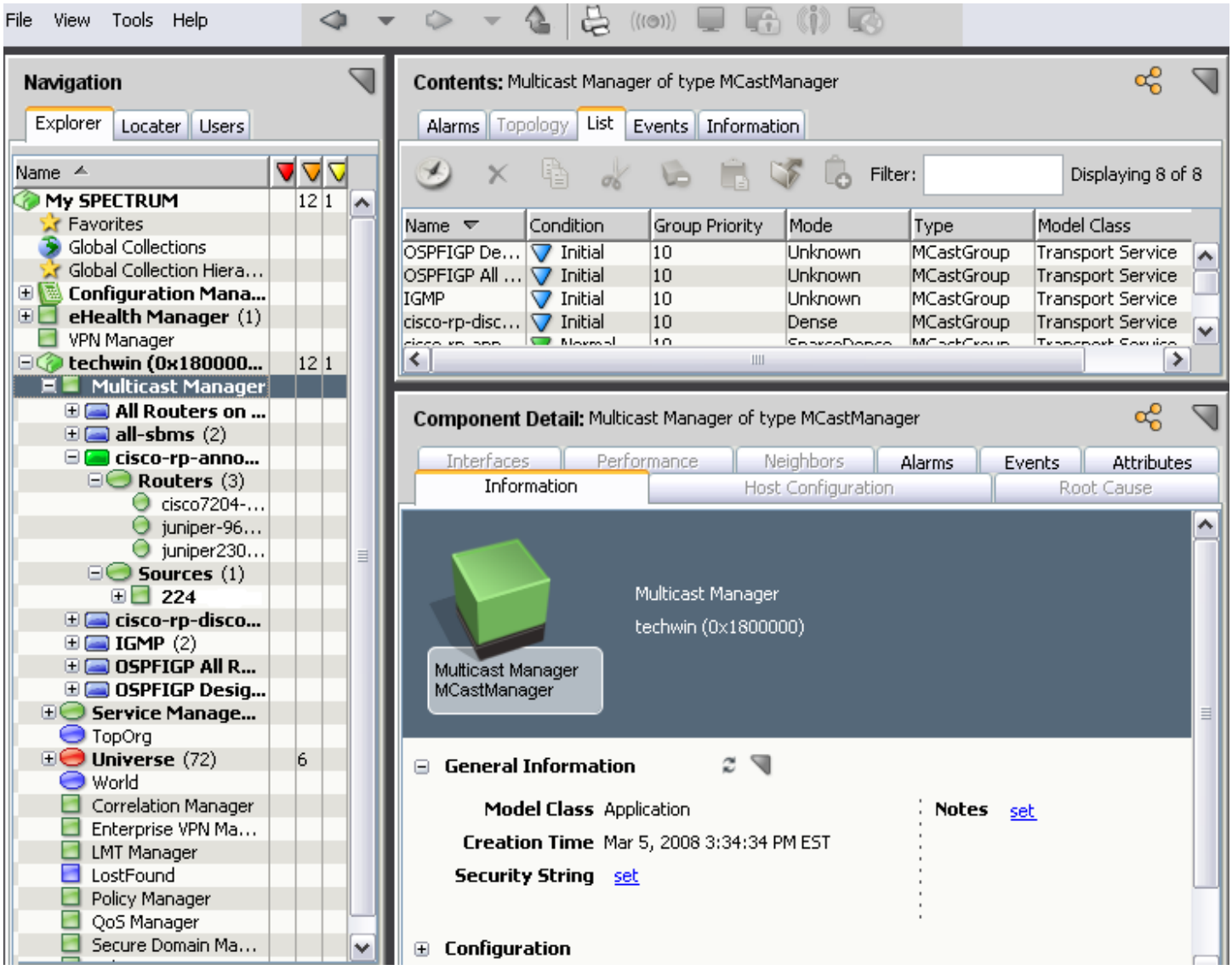
If you purchase a multicast service or a data feed from an external service and you do not have access to the source to obtain its status, Multicast Manager provides you with the ability to create a proxy to the source or you can configure Multicast Manager to exclude the unmanageable source's status in group condition calculation.

Multicast Manager View

You can access Multicast Manager from the Explorer tab of the Navigation panel in the OneClick Console. When you expand the Multicast Manager folder, all of the groups managed by Multicast Manager are listed. When you expand each group, the routers and sources contained in the Multicast Group are listed.

Once the physical components of your multicast network are modeled in DX NetOps Spectrum, you can use Multicast Manager to discover your multicast devices. You can then navigate to or search for these multicast elements in OneClick. The Contents panel and Component Detail panel provide configurations, alarms, events, and other information for a selected Multicast element.

The following shows the Multicast Manager view:



NOTE

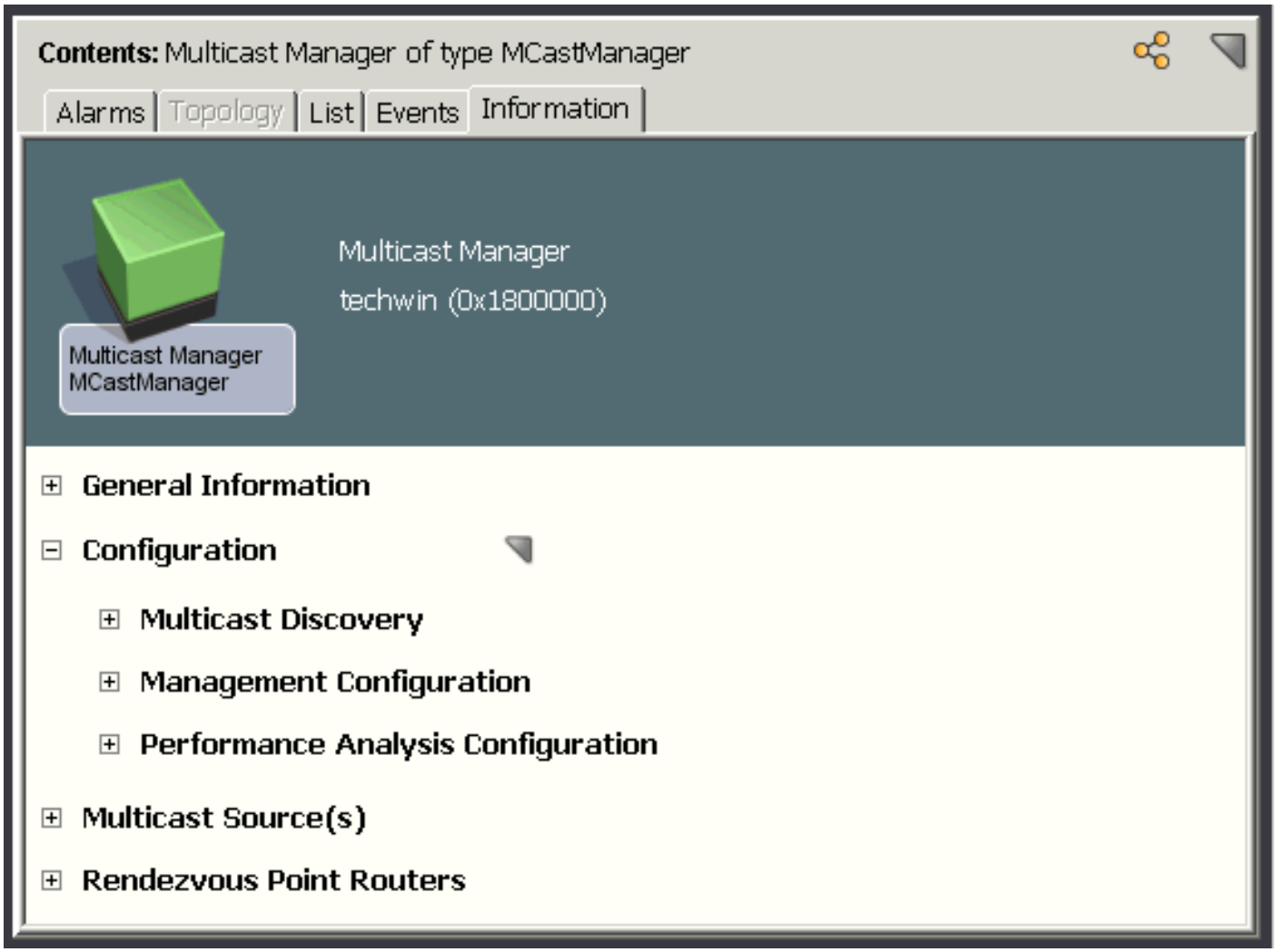
For more information about modeling your network, see the Modeling and Managing Your IT Infrastructure .

Multicast Manager Configuration

The Multicast configuration parameters are available to all users with administrative privileges.

To access Manager configuration parameters

1. Expand the desired landscape from the OneClick Explorer tab and select the Multicast Manager.
2. Click the Information tab in the Contents panel.
3. Expand the Configuration node to display the following options:



Multicast Discovery Subview

You can set the following parameters in the Multicast Discovery subview:

- **Discovery Status**
  Runs a Multicast discovery and displays status.
- **Group Address Filter Type**
  Specifies the addresses that you want to exclude or include. You can select one of the following options to set the filter type:
  - **Inclusive**
    Filters and saves the Multicast addresses that the Group Address Filter specifies.
  - **Exclusive**
    Filters and saves the Multicast addresses except those addresses that the Group Address Filter specifies.
- **Group Address Filter**
  Specifies the Multicast addresses to be saved when the Multicast Discovery is run.

NOTE
The addresses are not filtered and saved if you do not add them in the Group Address Filter. So, even if the Group Address Filter Type is inclusive and the Group Address Filter is empty, all addresses are discovered.

- **Background Create Group Models**
  Creates models for the new Multicast groups, sources, or routers that are discovered from a background discovery.

  NOTE
  The Enable Background Discovery parameter must be set to Yes.
- **Model Sources as Pingables**
  It may not be possible for you to model the device representing the Source of the Multicast traffic because of the IT infrastructure in your particular network environment. You can set one of the following options:
  – Yes (default)
    Creates a model using the pingable model type for each Multicast Source that does not have a corresponding device model. Multicast Manager automatically assigns a model name to this pingable model using the convention <IPAddress>_mcast. The models that are created by running Multicast Discovery are placed in the Multicast Pingables Generic Container in the landscape topology view.
  – No
  – Only if reachable by ICMP ping
    Creates a model only if it gets a response from an ICMP packet.

  NOTE
  By default, Multicast Manager determines the default gateway for the Source model. A correct default gateway is necessary for calculating performance and view in the topology.
- **Enable MSDP Discovery**

  NOTE
  Enables MSDP Discovery when set to yes.
- **Create Groups without Source**
  Creates a group when the source is not manageable. This behavior occurs when the source is initiated outside the network. If there is no manageable source to the group, the group stays in the initial condition and no fault management information is computed. When Create Groups without Source attribute is set to No, Multicast Manager does not create groups for sources that are not manageable.

  NOTE
  Default: Yes
- **Discovery Polling Interval (sec)**
  Determines how often (in seconds) Multicast Manager polls the attribute in the CiscoIPMRoute MIB during a background discovery. This polling determines if new multicast groups, sources, or routers exist.
- **Enable Background Discovery**
  Creates events when new multicast groups, sources, or routers become active on the network. Events for new groups and sources are asserted on the Multicast Manager model. Events for new multicast routes are asserted on the applicable device model.

  NOTE
  Before you enable background discovery, run a manual discovery to find and model the RP routers to be associated with discovered groups.

  To determine if a new entity has become active, Multicast Manager polls an attribute in the Cisco IPMRoute MIB which counts the number of entries in the Multicast routing table.

Management Configuration Subview

You can set the following parameters in the Management Configuration subview:

- **Default Group Priority**
  Sets the priority value given to a group when it is created. This value defines the relative importance of the group and can be used by the network operator to prioritize troubleshooting resources when there are problems with multiple groups.
- **Enable Port Polling**
  If this parameter is set to Yes, when Multicast Discovery is run, it enables port polling on all interfaces associated with group models. This lets port status contribute to the overall group condition.
  Default: Yes
- **Topology Display Units**
  Selects the units displayed in the Topology graph. You can select *one* of the following:
  – % Throughput
  – Pkts/s or Bits/s

  NOTE
  The Topology Display Unit is Pkts/s or Bits/s based on the Performance Collection Type option set under Performance Analysis Configuration.
- **Global Enable Multicast Path Change Detection**
  Enables or disables Multicast Manager's ability to detect topology changes in the Multicast distribution tree. It does this by identifying changes in the device interface receiving Multicast traffic. By default, it is set to No. You can also enable or disable this feature on a group model basis.
- **Total Groups (modeled/known)**
  Shows the number of groups that are modeled and known by Multicast Manager. The number of groups modeled is the number of multicast groups that have been discovered and are presently modeled by Multicast Manager. The number of groups known reflects the number of groups discovered by Multicast Manager via the discovery process, but that are not currently modeled; that is, they have been deleted.
- **Group Count Threshold**
  Displays the group count threshold. This parameter generates a critical alarm on the Multicast Manager model if it is exceeded.
- **Total Sources (modeled/known)**
  Shows the number of sources that are modeled and known by Multicast Manager. The number of sources modeled is the number of multicast sources that have been discovered and are presently modeled by Multicast Manager. The number of sources known reflects the number of sources discovered by Multicast Manager via the discovery process, but that are not currently modeled; that is, they have been deleted.
- **Source Count Threshold**
  Displays the source count threshold. This parameter generates a critical alarm on the Multicast Manager model if it is exceeded.

Performance Analysis Configuration Subview

You can set the following parameters in the Performance Analysis Configuration section:

- **Device Performance Alarms**
  Enables or disables the generation of Multicast performance based alarms on the device; these alarms are asserted in addition to those on the Multicast Group model. If enabled, alarms are produced on the device models in the Group causing the threshold violation.
  Default: Disable
- **Performance Analysis**
  Enables or disables the entire Multicast Performance Monitoring system. You can also enable or disable performance monitoring on a group model basis.
  Default: Enable

NOTE
Performance Analysis must be enabled for performance information to be displayed in the Multicast Topology view.

- **Collection Interval (sec)**
  Determines the interval at which the devices are sampled to obtain performance and IfIndex information. Lowering the interval may have a negative impact on the overall SpectroSERVER performance.
  Default: 300 seconds (5 minutes)
- **Performance Collection Type**
  Sets the performance collection type. You can select Packets/s or Bits/s.
  Default: Bits/s

Multicast Manager Discovery and Modeling

Contents

Multicast Network Modeling

DX NetOps Spectrum discovers and models the physical network infrastructure using Discovery, manual modeling, or the Modeling Gateway. Before using the Multicast Discovery functionality in Multicast Manager, you must first model the physical components of your network in DX NetOps Spectrum using one of these methods.

NOTE
For more information about using Discovery, manual modeling, or the Modeling Gateway to model your network, see the Modeling and Managing Your IT Infrastructure and the Modeling Gateway Toolkit .

Multicast Discovery

Once the physical network components have been discovered, network elements which implement the Multicast MIBs supported by Multicast Manager can be discovered and modeled. This discovery is based on the MIBs specified within Device Support. These MIBs provide the information required to identify all the primary multicast components.

Multicast Manager models multicast groups, sources, and receivers. To do this, Multicast Manager uses group, source, and receiver model types.

Group models represent a domain-wide multicast data stream. The source and receiver models represent the producer and consumer of information in a multicast network. The RP represents the meeting point for sources and receivers in a multicast network.

Run On-Demand Multicast Discovery

You must log in with administrator privileges to run Multicast Discovery.

To run Multicast Discovery

1. Expand the desired landscape from the OneClick Explorer tab and select the Multicast Manager.
   Information and Configurations display in the Contents panel.
2. Select the Information tab in the Contents panel, expand the Configuration folder and then expand the Multicast Discovery node.
   Multicast Discovery controls and configurations display.
3. Set the appropriate Discovery parameters.
4. Click Run next to the Discovery Status field to begin Multicast Discovery.
   The Discovery Status field shows the status of the discovery process.

Run Multicast Discovery on Selected Models

You can configure the Multicast Network Services Discovery from the OneClick views that display models.

To run Multicast Discovery on selected models

1. Select the models.
2. Click Tools, Utilities, Network Services Discoveries, Multicast Discovery.
   The Discovery process is initiated. You can check the status in the Configuration subview.

Configuring Multicast Discovery During Modeling

DX NetOps Spectrum lets you configure Network Services Discoveries, including Multicast Discovery, during modeling. As a part of modeling configuration, you can specify which network service discoveries to run with the modeling process.

> **NOTE**
> For more information, see the Modeling and Managing Your IT Infrastructure .

Multicast Source Discovery Protocol

DX NetOps Spectrum supports the Multicast Source Discovery Protocol (MSDP). If you create a device model using OneClick Discovery, AutoDiscovery, or manual modeling, DX NetOps Spectrum automatically discovers MSDP information on network devices that support the MSDP protocol. This information is used during the Multicast Discovery process to identify Multicast network elements only visible from your network domain through MSDP.

Delete Device Models

You can delete a device model by right-clicking the device and selecting Delete. If a device model is deleted, any associated multicast group, source, or receiver model is also deleted.

Managing the Multicast Network

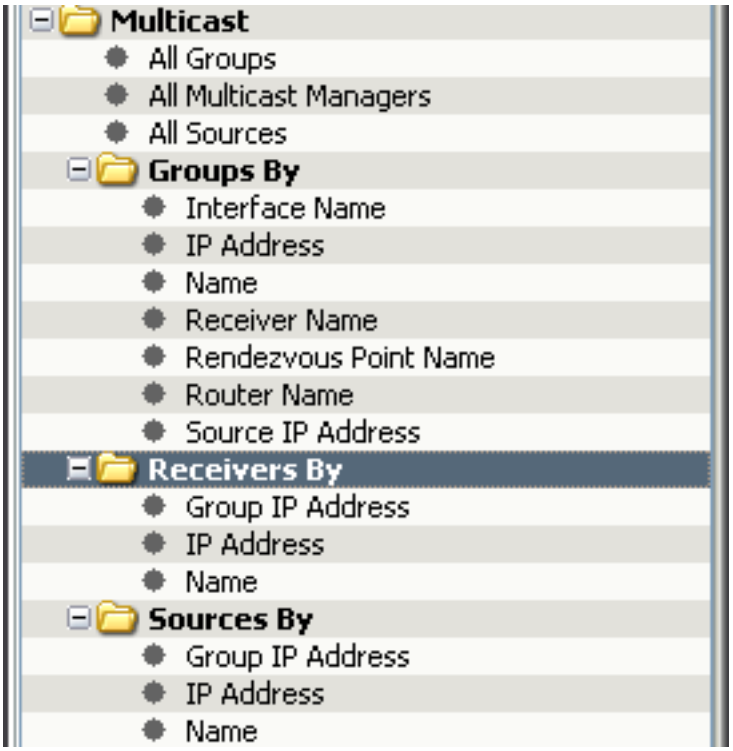
This section about how to manage your multicast network.

Search for Modeled Multicast Elements

Multicast searches, located in the Locater tab on the Navigation panel, lets you search your distributed network for Multicast Managers, multicast groups, multicast receivers, and multicast sources using criteria that you specify. You can use the search results to access a number of views which present management and performance information.

Search results appear in the Contents panel. Detailed information about the modeled device selected in the Contents panel is shown in the Component Detail panel.

The following shows the available multicast searches:



To search for modeled multicast elements

1. Select the Locater tab in the OneClick Navigation panel.
   All available OneClick searches display.
2. Expand the Multicast folder.
   All available Multicast searches display.
3. Double-click a search, enter the appropriate criteria, and click OK.

   NOTE
   Search criteria is case sensitive.

   The search results appear in the Contents panel.
4. (Optional) Use the Filter field in the Contents panel to filter your results.

   NOTE
   For more information about running searches, see the Network Searches .

Group Search Results

The results of group searches are displayed in the Contents panel. Use the Filter field in the Contents panel to filter your results. The following information displays for each group search:

- **Name**
  Indicates the name of the group. This may be the actual name or the IP address of the group.
- **Condition**
  Indicates the overall health of the group. It is calculated based on the condition of the RP routers, source, interconnecting routers, and router interfaces within the group. There are five possible conditions: Initial (blue), Normal (green), Degraded (orange), Maintenance (brown), and Down (red).
  If the condition of the group is either Degraded or Down, an alarm will be generated on the Group model.
  The condition displayed is based on the values in the following table.

NOTE
For Multicast Manager to determine the condition of a group, the group's sources and RP must be modeled in DX NetOps Spectrum and must be contained in the same DX NetOps Spectrum landscape.

- **Group Priority**
  Indicates the relative importance of the group. It can be used to prioritize troubleshooting resources when there are problems with multiple groups. The smaller the number the higher the level of priority.
- **Mode**
  Indicates the multicast mode of the group. The value can be either sparse or dense.
- **Type**
  Indicates the model type of the model that represents the device.
- **Model Class**
  Indicates the model class of the model that represents the device.

NOTE
For more information about model classes, see the SpectroSERVER and DX NetOps Spectrum Databases Overview .

The condition that is displayed for the overall health of a group is based on the values in the following table:

Group Condition	Performance Monitor Alarm Condition	RP Modeled	RP Contact Status isEstablished	Source Modeled	Source Contact Status is Established	Interface Contact Status is Established	Router Contact Status is Established
Good	Good	Y	Y	Y	Y	Y	Y
Minor	Minor	Y	Y	Y	Y	Y	Y
Major	Major	Y	Y	Y	Y	Y	Y
Critical	Critical	Y	Y	Y	Y	Y	Y
Good	Initial	Y	Y	Y	Y	Y	Y
Minor	Good	Y	Y	Y	Y	N	N
Minor	Minor	Y	Y	Y	Y	N	N
Major	Major	Y	Y	Y	Y	N	N
Critical	Critical	Y	Y	Y	Y	N	N
Minor	Initial	Y	Y	Y	Y	N	N
Critical	Good	Y	N	Y	N	Y	Y
Critical	Minor	Y	N	Y	N	Y	Y
Critical	Major	Y	N	Y	N	Y	Y
Critical	Critical	Y	N	Y	N	Y	Y
Critical	Initial	Y	N	Y	N	Y	Y
Good	Good	N	N/A	N	N/A	N/A	N/A
Minor	Minor	N	N/A	N	N/A	N/A	N/A
Major	Major	N	N/A	N	N/A	N/A	N/A
Critical	Critical	N	N/A	N	N/A	N/A	N/A
Initial	Initial	N	N/A	N	N/A	N/A	N/A

NOTE

A Multicast group will be put into maintenance mode if any of the following occur:

If there are any RPs within the Multicast group and they are all in maintenance.

- If there are Multicast sources within the Multicast group and they are all in maintenance.
- If there are any devices within the Multicast group and they are all in maintenance.
- If there are any interfaces within the Multicast group and they are all in maintenance.

Receiver Search Results

The results of receiver searches are displayed in the Contents panel. Use the Filter field in the Contents panel to filter your results. The following information displays for each receiver search:

- **Name**
  Indicates the name of the receiver. This may be the actual name or the IP address of the receiver.
- **Condition**
  Indicates the status of the model that represents the device. Possible conditions are initial (blue), normal (green), minor alarm (yellow), major alarm (orange), critical alarm (red), gray (unknown), and maintenance (brown).
- **Type**
  Indicates the model type of the model that represents the device.
- **IP Address**
  Indicates the IP address of the receiver.
- **Model Class**
  Identifies the model class of the receiver.

  NOTE
  For more information about model classes, see the SpectroSERVER and DX NetOps Spectrum Databases Overview .

Source Search Results

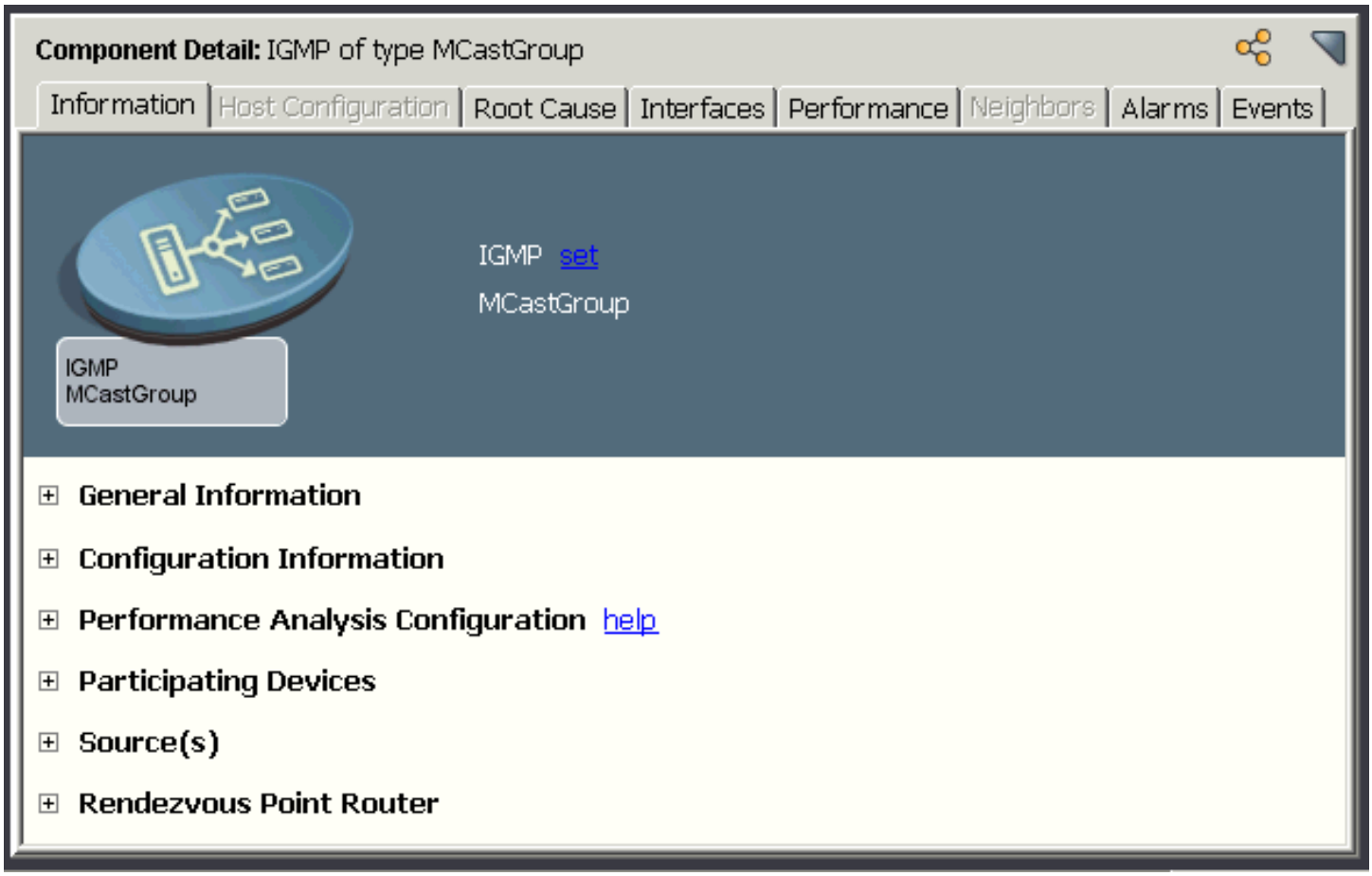
The results of Source searches are displayed in the Contents panel. Use the Filter field in the Contents panel to filter your results. The following information displays for each source search:

- **Name**
  Identifies the name of the source. This may be the actual name or the IP address of the source.
- **Condition**
  Indicates the status of the model that represents the device. Possible conditions are initial (blue), normal (green), minor alarm (yellow), major alarm (orange), critical alarm (red), gray (unknown), and maintenance (brown).
- **Type**
  Indicates the model type of the model that represents the device.
- **Source Address**
  Identifies the IP address of the source.
- **Model Class**
  Identifies the model class of the source.

  NOTE
  For more information about model classes, see the SpectroSERVER and DX NetOps Spectrum Databases Overview .

Multicast Groups

The Component Detail panel contains tabs that let you view multicast information about a selected group. The Information tab provides a number of subviews for the group.



General Information Subview for Multicast Groups

You can set the following parameters for a group model in the General Information subview:

- **Group Address**
  Indicates the IP address of the group.
- **Group Priority**
  Indicates the relative importance of the Group. It can be used by the network operator to prioritize troubleshooting resources when there are problems with multiple Groups. The lower the value of Group Priority for a given Multicast Group, the higher its priority. If you are logged in as an administrator, you can edit this field by clicking set.
- **Model Class**
  Indicates the model class of the model that represents the device.

  NOTE
  For more information about model classes, see the SpectroSERVER and DX NetOps Spectrum Databases Overview .
- **Creation Time**
  Indicates the creation date and time of the Group model.
- **Security String**

Indicates the SNMP community string or password for the device. If you have the appropriate permissions, you can modify the security string. Click set, enter the security string into the available field, and press Enter.

- **Landscape**
  Indicates the DX NetOps Spectrum landscape on which the device is modeled.
- **Notes**
  Indicates notes or comments about the group. To add a note, click Set and then enter the appropriate information. When you have finished entering the note, click Save. If you do not want to save the note that you entered, click Cancel.

Configuration Information Subview

You can set the following attribute in the Configuration Information subview:

- **Ignore initial/unmodeled sources in group condition**
  Lets you choose (on a per-group basis) whether or not to include the status of the source model in the group condition calculation. If you purchase a Multicast service or a data feed from an external service, you may not have access to the source to obtain its status. You can use this option to exclude the source's status in the group condition calculation. If you do own and have access to the source of the multicast stream, you should include the source status in the group status configuration.

Performance Analysis Configuration Subview for Multicast Groups

Performance analysis shows you important information about the flow of multicast traffic on your network. In a properly functioning multicast network, each point of the network (where multicast is enabled for that group) should experience the same traffic flow in packets per second or bytes per second.

Any deviation from the traffic level as measured at the source could indicate one or more of the following scenarios:

- A change in group membership.
- A change in routing where multicast traffic is taking a different path due to load balancing, a redundant failover, or a change from shared to source mode.
- A change in the network's multicast configuration where multicast has been disabled on an interface or device.
- A change in interface status.
- Device or link instabilities.

You can configure how Multicast Manager analyzes the performance of the group model selected. A graphical representation of source traffic is also available. You can set the following parameters in the Performance Analysis Configuration subview:

- **Global Group Performance Analysis**
  Indicates whether the entire Multicast Performance Monitoring system is enabled or disabled. It can be set in the Multicast Manager model's Performance Analysis Configuration. The value of this parameter must be set to enable for the Group Performance Analysis to function.
- **Group Performance Analysis**
  Enables performance monitoring for each Group. This value must be enabled for performance information to be displayed in the Multicast Topology view.
- **Percent Degradation Threshold**
  Defines the percent variation allowed in the monitoring of Group performance. For example, if the source is measured at 100 pps, all other points in the network receiving this Group should be within this percent threshold (+/-). Otherwise, a threshold violation is noted for the Group on that device. The default value is three percent.
- **Enable Multicast Path Change Detection**
  When this parameter is set to Yes, the Multicast Group looks for changes in the interface receiving multicast traffic. If a change is detected, an event is generated on the offending device. This event gives information about the device

that generated the event, the interface receiving the traffic, and the group to which the device belongs. By default, this parameter is set to No.
The following scenarios can cause a change in path to be detected:
  – A router performs a routine load balancing operation and redirects the multicast traffic to be routed around the network by a different path. This does not necessarily indicate a failure.
  – A router along the path of the multicast traffic goes down. In this case a topology change will occur and the device receiving the multicast traffic will detect that the interface sending the information has changed.
- **Enable Path Change Alarms**
  If this parameter is set to Yes, Multicast Manager generates a yellow (minor) alarm when a change in path is detected. For this parameter to operate correctly, Enable Multicast Path Change Detection must be set to Yes.
- **Group Performance Alarms**
  Enables Multicast Manager to create an alarm in response to the violation of performance thresholds defined in the Minor, Major, and Critical Alarm Threshold parameters.
- **Minor Alarm Threshold**
  Sets the Minor Threshold Percent. This value determines the severity of threshold violations on a Group. For example, a Group may have 100 interfaces sampled in the network. If more than the Minor Alarm Threshold percentage and less than the Major Alarm Threshold percentage of those samples are in violation of the Performance Degradation Threshold, a Minor event (and an alarm if Group Performance Alarms is set to enable) is generated.
- **Major Alarm Threshold**
  Sets the Major Threshold Percent. This value determines the severity of threshold violations on a Group. For example, a Group may have 100 interfaces sampled in the network. If more than the Major Alarm Threshold percentage and less than the Critical Alarm Threshold percentage of those samples are in violation of the Performance Degradation Threshold, a Major event (and an alarm if Group Performance Alarms is set to enable) is generated.
- **Critical Alarm Threshold**
  Sets the Critical Threshold Percent. This value determines the severity of threshold violations on a Group. For example, a Group may have 100 interfaces sampled in the network. If more than the Critical Alarm Threshold percentage of those samples are in violation of the Performance Degradation Threshold, a Critical event (and an alarm if Group Performance Alarms is set to Enable) is generated.
  – **Minimum Rate (Bits/sec)**
    Sets the minimum rate for critical alarm threshold.
    Default: 0 Bits/ sec
  – **Maximum Rate (Bits/sec)**
    Sets the maximum rate for critical alarm threshold.
    Default: 4294967295 Bits/ sec

    NOTE
    The Minimum Rate and Maximun Rate for alarm threshold apply only when Performance Collection Type is set to Bit/Sec. Thus these alarms should not occur unless the user inputs something for these values.

Participating Devices Subview

The Participating Devices subview provides information for all of the devices that participate in the multicast group. You can set the following parameters in the Participating Devices subview:

- **Condition**
  Indicates the status of the model that represents the device. Possible conditions are initial (blue), normal (green), minor alarm (yellow), major alarm (orange), critical alarm (red), gray (unknown), and maintenance (brown).
- **Name**
  Indicates the name of the participating device model.
- **Network Address**
  Identifies the network address of the device.
- **Manufacturer**

Indicates the manufacturer of the device that the model represents.

- **Model Class**
  Identifies the model class of the device model.

  NOTE
  For more information about model classes, see the SpectroSERVER and DX NetOps Spectrum Databases Overview .

- **MAC Address**
  Indicates the MAC address of the device.
- **Type**
  Indicates the model type of the model that represents the device.
- **Landscape**
  Indicates the DX NetOps Spectrum landscape on which the device is modeled.

Sources Subview

The Sources subview lists all of the sources for a multicast group. You can set the following parameters in the Sources subview:

- **Condition**
  Identifies the status of the source model. Possible conditions are initial (blue), normal (green), minor alarm (yellow), major alarm (orange), critical alarm (red), unknown (gray), and maintenance (brown).
- **Name**
  Identifies the name of the source model. This may be the actual name or the IP address of the source.
- **Type**
  Identifies the model type of the source model.
- **Model Class**
  Identifies the model class of the source model.

  NOTE
  For more information about model classes, see the SpectroSERVER and DX NetOps Spectrum Databases Overview .

- **Landscape**
  Indicates the DX NetOps Spectrum landscape on which the device is modeled.

Rendezvous Point Router

The Rendezvous Point Router subview lists the RP router for the Multicast Group. You can set the following parameters in the Rendezvous Point Router subview:

- **Condition**
  Indicates the status of the model that represents the device. Possible conditions are initial (blue), normal (green), minor alarm (yellow), major alarm (orange), critical alarm (red), gray (unknown), and maintenance (brown).
- **Name**
  Indicates the model name of the model that represents the device.
- **Network Address**
  Identifies the network address of the device.
- **Manufacturer**
  Indicates the manufacturer of the device that the model represents.
- **Model Class**
  Indicates the model class of the model that represents the device.
- **MAC Address**

Indicates the MAC address of the device.

- **Type**
  Indicates the model type of the model that represents the device.
- **Landscape**
  Indicates the DX NetOps Spectrum landscape on which the device is modeled.

Group Interfaces Information

You can access information about the interfaces associated with a selected multicast group by selecting a group model and then selecting the Interfaces tab on the Component Detail panel. The following information displays for the interface model:

- **Name**
  Indicates the model name of the model that represents the device.
- **Condition**
  Indicates the status of the model that represents the device. Possible conditions are initial (blue), normal (green), minor alarm (yellow), major alarm (orange), critical alarm (red), gray (unknown), and maintenance (brown).
- **Status**
  Indicates the contact status of the link between this interface and the interface to which it is connected.
- **PIM Status**
  Indicates the configured mode of this PIM interface.
- **Mode**
  Indicates the PIM mode of the interface. This value can be sparse, dense or sparseDense.
- **IGMP Version**
  Indicates the version of IGMP that is running on this interface.

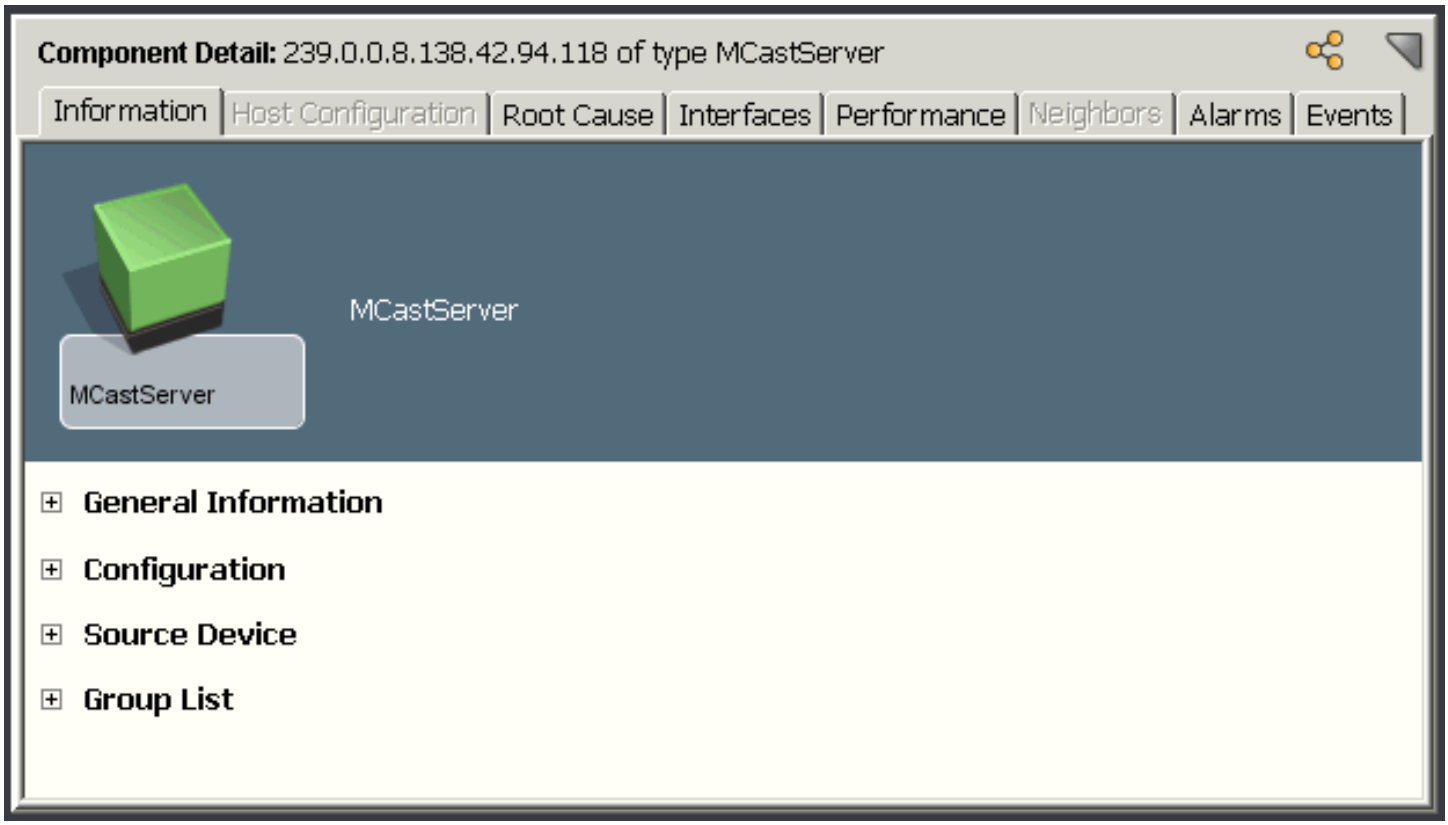
NOTE
If an interface does not participate in a particular protocol, the columns pertaining to that protocol for that interface will not contain any data.

Group Event Information

You can access the Event view for a group by selecting a group model and then selecting the Events tab on the Component Detail panel. The Events tab displays event information for events that occur on the group model.

Multicast Source Management

The Component Detail panel contains tabs that let you view additional information about a source selected from the search results list in the Contents panel. The Information tab provides a number of subviews for the source, as shown in the following example:



NOTE
If a multicast source is unmanageable, you can define a source proxy.

General Information Subview for the Multicast Source Model

You can set the following parameters for the multicast source model in the General Information subview:

- **Condition**
  Indicates the status of the model that represents the device. Possible conditions are initial (blue), normal (green), minor alarm (yellow), major alarm (orange), critical alarm (red), gray (unknown), and maintenance (brown).
- **Source Address**
  Indicates the IP address of the source.
- **Model Class**
  Indicates the model class of the model that represents the device.

  NOTE
  For more information about model classes, see the SpectroSERVER and DX NetOps Spectrum Databases Overview .
- **Creation Time**
  Identifies the creation date and time of the source model.
- **Security String**

Indicates the SNPMP community string or password for the device. If you have the appropriate permissions, you can modify the security string. Click set, enter the security string into the available field, and press Enter.

- **Landscape**
  Indicates the DX NetOps Spectrum landscape on which the device is modeled.

Configuration Subview

The Configuration subview lets you define a proxy to reach the selected source and reconfigure the Default Gateway. You can set the following parameters for a proxy in the Configuration subview:

- **Default Gateway**
  Indicates the IP Address of the device DX NetOps Spectrum should poll to obtain base-line performance statistics for a multicast group. This value will generally be the IP address of the default gateway for the Multicast Source. By default, Multicast Manager determines the default gateway. The default gateway must be correct to utilize the Multicast Performance and Topology views. Click Set to modify the IP address.
  When the default gateway is set to the Virtual IP address of a Hot Standby Routing Protocol (HSRP) group, for Multicast Performance Analysis to work properly, it is necessary to configure the HSRP member routers to send HSRP related SNMP Traps to DX NetOps Spectrum. This lets DX NetOps Spectrum continue to obtain multicast performance statistics when the Active HSRP router changes.
- **Proxy IP**
  Indicates the IP Address of the proxy for an unmanageable multicast source. This lets you compute condition and gather performance data for an unmanageable source using the information from the closest ingress router to the unmanageable source.

Source Device Information

The following information is shown about the source device for a multicast source model.

- **Condition**
  Indicates the status of the model that represents the device. Possible conditions are initial (blue), normal (green), minor alarm (yellow), major alarm (orange), critical alarm (red), gray (unknown), and maintenance (brown).
- **Name**
  Indicates the model name of the model that represents the source.
- **Network Address**
  Identifies the network address of the device.
- **Manufacturer**
  Indicates the manufacturer of the device that the model represents.
- **Model Class**
  Indicates the model class of the model that represents the source.

  NOTE
  For more information about model classes, see the SpectroSERVER and DX NetOps Spectrum Databases Overview .

- **MAC Address**
  Indicates the MAC address of the source device.
- **Type**
  Indicates the model type of the model that represents the source device.
- **Landscape**
  Indicates the DX NetOps Spectrum landscape on which the device is modeled.

Group List Subview

The Group List subview displays information about group models to which the source model sends information. You can set the following parameters for a group model in the Group List subview:

- **Condition**
  Indicates the status of the model that represents the device. Possible conditions are initial (blue), normal (green), minor alarm (yellow), major alarm (orange), critical alarm (red), gray (unknown), and maintenance (brown).
- **Name**
  Indicates the model name of the model that represents the device.
- **Group Priority**
  Indicates the relative importance of the group. It can be used to prioritize troubleshooting resources when there are problems with multiple groups. The smaller the number the higher the level of priority.
- **Mode**
  Indicates the multicast mode of the group. The value can be either sparse or dense.
- **Type**
  Indicates the model type of the model that represents the device.
- **Model Class**
  Indicates the model class of the model that represents the device.

  NOTE
  For more information about model classes, see the SpectroSERVER and DX NetOps Spectrum Databases Overview .

Source Event Information

You can access the Event view for a source by selecting a source model and then selecting the Events tab on the Component Detail view. The Events tab displays event information for events that occur on the source model.

Source Performance Information

Real-time multicast performance statistics can be obtained for a multicast source by selecting the Performance tab in the Contents panel. The Performance view provides a real-time graphical view of source traffic. The information provided in the Performance view is useful for monitoring multicast performance and faults within the multicast environment in a graphical format.

You can select *one* of the following:

- Multicast Source Octet Traffic
- Multicast Source Packet Traffic
- Multicast Source Bit Traffic

For the graph to function properly, the following must be true:

- If you are able to create a physical device model of the source, you should model it as an SNMP host. You can choose one of the following host model types:

  - Host_Compaq
  - Host_Dell
  - Host_Sun
  - Host_systemEdge
  - Host_Device
  - GnSNMPDev
- If you are not able to create a physical device model of the source, you should model the source as a pingable. The Multicast Discovery process creates pingable models automatically for all multicast sources that do not have a corresponding physical device model.
- DX NetOps Spectrum must be able to communicate with the source.
- The default gateway for the source must be modeled in DX NetOps Spectrum using the appropriate router model type.
- If you model the source as a pingable, the default gateway must be correct in the source model's Configuration view.
- DX NetOps Spectrum must be able to communicate with the default gateway for the source.
- The default gateway must support one of the IPM Route MIBs (draft or RFC).
- The routers in a Hot Standby Routing Protocol (HSRP) group must be set to send HSRP traps to the SpectroSERVER for Multicast Manager performance monitoring to work properly, and for performance graphs to show traffic statistics when the active router in the HSRP group has changed because of a failover.

Define a Proxy for an Unmanageable Source

Multicast Manager lets you define a proxy to monitor an unmanageable source. Examples of an unmanageable source (or multicast stream) are groups which originate outside of your network. These could be feeds which traverse the Internet to get to this location or dedicated services like financial data streams which are purchased to feed trading applications.

The proxy should ideally be the ingress router where the multicast stream enters the network. Since you cannot monitor the source, you can configure the ingress router where the multicast stream enters the network to represent the unmanageable source for performance comparisons.

To define a proxy for an unmanageable source

1. Conduct a Multicast Search for the unmanageable source model.
   The Contents panel displays the results of the search.
2. Select the unmanageable source model from the results list.
   Information and configurations display in the Information tab of the Component Detail panel.
3. Expand the Configuration subview.
   The Default Gateway and Proxy IP configuration parameters display.
4. Click set next to Proxy IP and enter the IP Address of the ingress router where the Multicast Stream from the unmanageable source enters your network and press Enter.
   The IP Address of the unmanageable source's proxy displays.

Alarm List

The Contents panel displays the Alarms list for the modeled element that you select in the Navigation panel.

To know more about an alarming element, you can click **Alarm Detail** on the Alarm tab of the Component Detail panel.

If the Alarm list in the Contents panel is empty, the Component Detail panel displays the Information tab for a selected element.

Multicast Receiver Information

Information about multicast receivers modeled as pingables is available in the Multicast Information subview of the pingable model. Expand the Associated Multicast Servers subview to display the receiver's associated multicast servers

and their condition. Expand the Associated Multicast Groups subview to display the receiver’s associated multicast groups and their condition.

View Multicast Topology Information

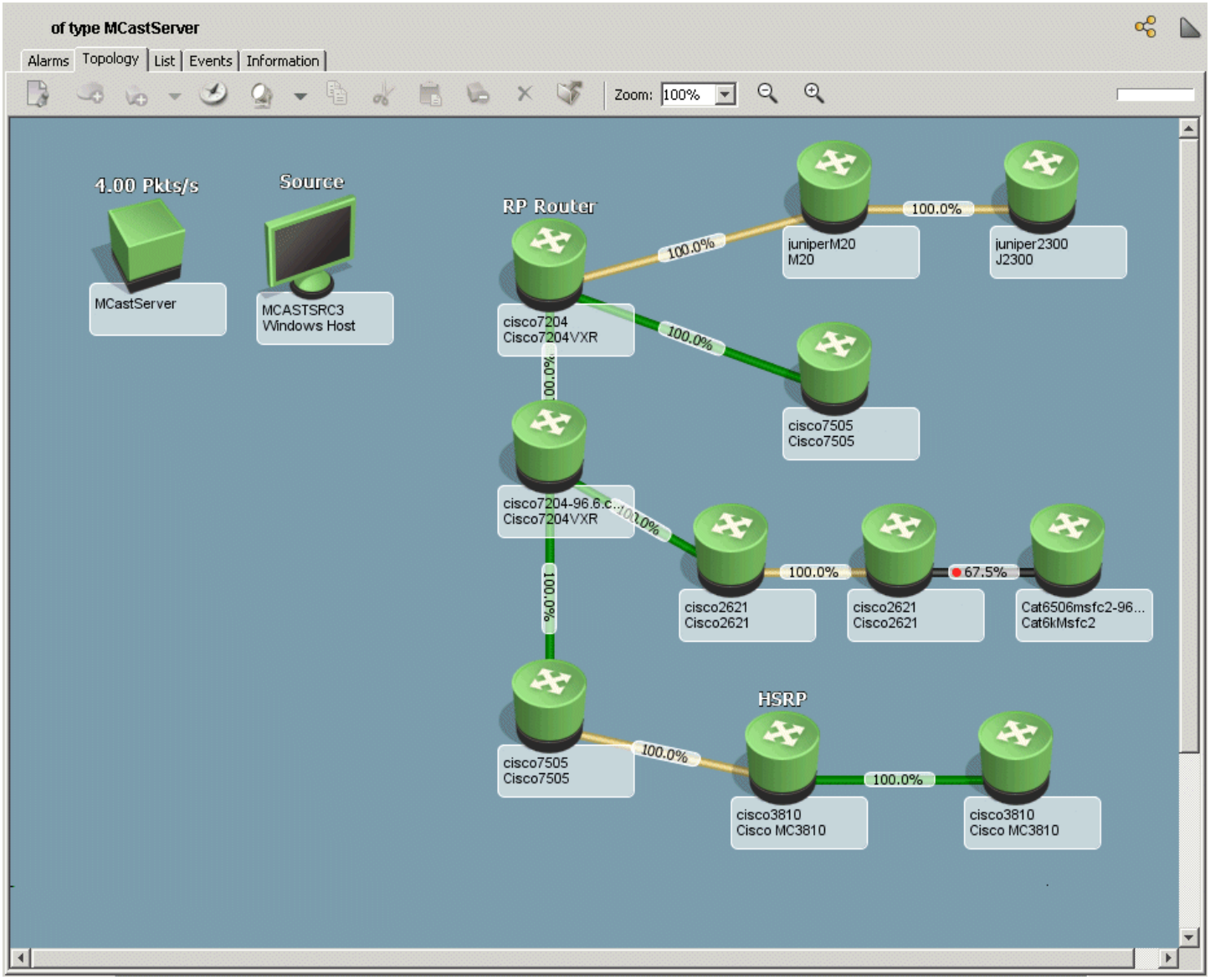
The Multicast Topology view provides a graphical representation of the data flow from a multicast source through the associated devices, and allows you to more easily troubleshoot and gauge the overall impact of traffic flow changes and outages within your environment. You can access a topology view for each source within a multicast group that Multicast Manager has discovered and modeled. The performance information for each segment of the traffic distribution tree displays the percentage of source traffic reaching that segment. The percentage is calculated by comparing the baseline traffic from the source.

Performance Analysis must be enabled on the Multicast Manager model and for any multicast group you want performance information to be displayed for in the Multicast Topology view.

The path structure and performance data displayed is based on the latest poll. It may not necessarily reflect the current state.

To view multicast topology information

1. Expand the appropriate Multicast Manager Group node from the Navigation panel.
2. Expand the Source folder, select the desired source, and select the Topology tab in the Contents panel. Multicast Manager displays a Topology view.



NOTE
Use the Topology Display Units attribute to choose the desired units to display in the topology.

Trap Support by Multicast Manager

The Multicast Manager supports the handling of traps specified by the CISCO-PIM-MIB. The following table describes the supported traps and their information.

Supported Traps	OID	Event Generated	Alarm Generated	Default Alarm Severity
ciscoPIMRPMappingChange	1.3.6.1.4.1.9.9.184.2.0.3	0x2104a5	None	NA
ciscoPIMInvalidRegister	1.3.6.1.4.1.9.9.184.2.0.4	0x2104a6	None	NA
ciscoPIMInvalidJoinPrune	1.3.6.1.4.1.9.9.184.2.0.5	0x2104a7	None	NA

In addition to supporting traps specified by the CISCO-PIM-MIB, Multicast Manager supports the trap specified by the CISCO-HSRP-MIB. Multicast Manager can perform multicast performance monitoring on the active router in a HSRP group when the active router has changed due to a failover.

The following table describes the HSRP trap.

HSRP Traps	OID	Event Generated	Alarm Generated	Default Alarm Severity
cHsrpStateChange	1.3.6.1.4.1.9.9.106.2.6.1	0x0021091e 0x0021091f	0x0021091e 0x0021091e	Minor NA

Devices Supported by Multicast Manager

Cisco and Juniper M Series Devices

Multicast Manager supports the discovery, modeling, and viewing of devices that support multicast traffic using the MIBs specified in the following table:

Supported MIBs	Cisco 12.0	Cisco 12.2 or later
ipMRoute - Cisco	Y	Y
ipMRoute - Experimental	Y	Y
ipMRoute - Standard	N	N
PIM - Cisco	Y	Y
PIM - Experimental	Y	Y
PIM - Standard	Y	Y
IGMP - Experimental	Y	N
IGMP - Standard	N	Y
Functions		
Group Discovery	Y	Y
Source Discovery	Y	Y
Receiver Discovery	Y	Y
RP Discovery	Y	Y
Group Performance	Y	Y
Traps		
Neighbor Loss	N	N
PIM Interface Up	N	N
PIM Interface Down	N	N
RP Mapping Change	Y	Y
Invalid Register	Y	Y
Invalid Join Prune	Y	Y
Dynamic Discovery Traps		
Groups/Sources	N	N

Multicast Manager depends on functionality provided by DX NetOps Spectrum in the Cisco Router management module. Therefore, you must install the Cisco Router management module before or during the Multicast Manager installation.

NOTE

For more information about Cisco management modules, see the Cisco Device Management *section.*

Multicast Manager also supports Juniper M series devices using JunOS releases 6.1 and greater. The Juniper M series management module (SM-JPR1000) must be installed for Multicast Manager to discover and support multicast-enabled, Juniper devices.

NOTE
For more information about Juniper M series devices, see the Device Management Reference *section.*

Troubleshooting Multicast Manager

The Group Condition Is Displayed Incorrectly

Symptom:

The group condition is being displayed incorrectly even though the source is down.

Solution:

If the source has not been modeled by DX NetOps Spectrum, when the physical device goes down there is no way to associate that status with a group on which it depends. Normally, a source down should produce a group down condition but, without the source modeled, the group shows a degraded condition.

Verify that the source is active and modeled by DX NetOps Spectrum, or that you have modeled third-party sources as pingable models.

The Group Condition Is Not Calculated Correctly

Symptom:

The group condition is not being calculated correctly when the model representing the RP is deleted after Multicast Discovery is run.

Solution:

Delete all of the groups and rerun Multicast Discovery.

Symptom:

The group condition is not being calculated correctly when the device representing the RP is not modeled.

Solution:

Model the RP, then rerun Multicast Discovery.

Changes Are Not Reflected in Performance Graphs

Symptom:

I added and removed some multicast sources and so I had to re-run Multicast Discovery. However, my changes are not immediately reflected in the performance graph.

Solution:

Restart the OneClick client.

New Group Models Staying in Initial Condition

Symptom:

New group models stay in Initial condition after a background discovery is run.

Solution:

No RPs will be discovered if a background discovery is run without previously running a manual discovery. Run a manual Multicast Discovery first.

Downward Spikes Display in a Group Performance Graph

Symptom:

When I display a group performance graph during an HSRP switchover, and the HSRP routers are configured to send HSRP traps to the SpectroSERVER, I see a downward spike to 0 for one polling cycle.

Solution:

After the switchover, the graph begins displaying the performance of the new active gateway.

No Data Is Displaying in Performance Graphs

Symptom:

No data is displayed for Group Performance Graph involving routers that are configured for HSRP.

Solution:

The likely cause is that the standby HSRP router has become the active router, and DX NetOps Spectrum was not notified of the change using the HSRP trap. Verify that the HSRP routers are configured to send the HSRP trap to the SpectroSERVER.

Performance Graphs Displaying Last Known Data Points After Contact is Lost

Symptom:

The Performance Graph view for a multicast group model displays the last known data point when contact with the default gateway is lost.

Solution:

This only happens for devices that do not support the IPMRoute MIB. For devices that support the IPMRoute MIB, the Performance Graph view displays a “0” data value until contact with the device is re-established.

Performance Statistics are Displaying as N/A

Symptom:

Performance statistics are displayed as N/A in the Multicast Topology view.

Solution:

The topology cannot gather data from the default gateway. If the performance information remains unavailable for an extended period, verify that the default gateway setting is set to a contactable multicast enabled router within this source’s traffic topology tree.

Use Case Scenarios

This section contains information about common use cases of network management.

Analyze Impact of a Device Alarm on Customer Multicast Traffic

Symptom:

Two unrelated trouble tickets against two core routers are generated at nearly the same time. Consider the following:

- The trouble tickets are routed based on the device class. They are sent to the level 2 operations Group. This Group focuses solely on maintaining the core routing infrastructure.
- The company's network management application has identified the routers in question and their offending components.
- The level 2 support group currently has a backlog of 25 open tickets for a variety of issues.
- The operators are currently using their knowledge of the core network to make decisions regarding issue prioritization.

Assume the following:

1. The customer has previously installed and configured Multicast Manager.
2. The customer has completed a discovery of their physical network.
3. The customer has completed a Multicast Discovery.
4. All multicast services have been modeled with DX NetOps Spectrum.
5. There are a couple of outages and the operator must analyze how these outages affect the multicast topology.
6. There is currently only one network administrator available to handle trouble tickets.

Solution:

1. The first router in the network goes down and is detected by DX NetOps Spectrum. Very shortly after this, the second router goes down.
2. DX NetOps Spectrum generates an alarm for each of the routers that has gone down.
3. The network management system (NMS) operator receives the alarms in the OneClick console and verifies the DX NetOps Spectrum alarms by pinging the routers and not receiving a response.
4. The NMS operator receives an alarm generated on one of the multicast groups modeled by Multicast Manager.
5. The NMS operator selects the multicast group in the OneClick console Alarms tab.
6. The NMS operator opens the Participating Devices subview of the Group’s Information tab and sees that one of its routers has a Down (Red) condition. This router is one of the routers that an alarm was generated on in step 2 above. The network operator now understands that this router is critical to the operation of this multicast group, and of a higher priority than the other router issue.
7. The network operator creates a trouble ticket for the router which is impacting the multicast group. This ticket will show that the problem is a high impact outage and will include information on the impacted groups and applications. The network operator assigns the ticket to the network administrator.
8. The network administrator receives the trouble ticket and begins troubleshooting this high impact problem.
9. The network operator creates a trouble ticket for the second router outage that does not affect the multicast topology and assigns it to the network administrator.
10. The network administrator receives the second trouble ticket and will begin work on it after the multicast problem is resolved. DX NetOps Spectrum has allowed the troubleshooting process to be prioritized to the most critical fault.

An Individual User Is Not Receiving Multicast Traffic

Symptom:

When the stock market opens every day, a multicast feed is established to send real time market data to all the brokers on the company's network. The multicast group uses PIM Sparse mode to communicate to the traders who are widely dispersed throughout the company's operations. Today one broker is not receiving the market data feed but is receiving other multicast traffic.

Assume the following:

1. The customer has previously installed and configured Multicast Manager.
2. The customer has completed a discovery of their physical network.
3. The customer has completed a Multicast Discovery.
4. All multicast services have been modeled with DX NetOps Spectrum.
5. The broker has reported that he is no longer receiving the market data feed.

Solution:

1. The operator sees a DX NetOps Spectrum alarm in the OneClick console indicating that the router port has failed.
2. A trouble ticket is created and the operator begins troubleshooting the router port issue.
3. As part of his troubleshooting procedure the operator uses the Multicast Manager “Groups by Interface” search.
4. This view shows all the affected groups and their priority. Groups 1-3 are affected by this router.
5. In addition, a broker reports that he is no longer receiving the multicast data feed for Group 2.
6. The operator can tell the broker that he is part of the affected group (due to the router failure). A second trouble ticket does not need to be created.
7. The operator fixes the router interface.
8. The router interface is working and market data is now being received by the broker reporting an outage.
9. The operator confirms multicast traffic is being received by the broker and closes the trouble ticket.

Multiple Users Are Not Receiving Multicast Traffic

Symptom:

At the same time each day, a single multicast stream is started. Today the source is started, but none of the users are seeing the multicast traffic, and consequently the business application they use is not functioning.

Assume the following:

1. The customer has previously installed and configured Multicast Manager.
2. The customer has completed a discovery of their physical network.
3. The customer has completed a Multicast Discovery.
4. All multicast services have been modeled with DX NetOps Spectrum.

Solution:

1. Several users contact the network operator to report that they are not receiving their daily service information.
2. The operator brings up Multicast Manager in the OneClick console and does a search on “All Groups”.
3. The results of the search are shown in the Contents panel and all groups have a red condition.
4. The operator examines the source for each group in the Information tab of the Component Detail panel.
5. The operator finds no errors; all sources are operating properly.
6. The operator checks the RP Router for each group.
7. The operator finds that the condition of one of the RP routers is red, indicating that it is down.
8. The operator searches through open trouble tickets and finds a trouble ticket for a router, which is the RP for the group in question.
9. The operator reboots the RP router and the alarm is cleared.
10. Multicast traffic is restored to the network. The operator confirms that the users have service and the trouble ticket is cleared.

A Threshold Violation Has Occurred

Symptom:

At 10:00 A.M., users stop receiving Multicast traffic, consequently, the business application needed by these users is not functioning.

Assume the following:

1. The customer has previously installed and configured Multicast Manager.
2. The customer has completed a discovery of their physical network.
3. The customer has completed a Multicast Discovery.
4. All multicast services have been modeled with DX NetOps Spectrum.
5. The performance monitor configuration has been set up to generate alarms when a performance threshold is violated.
6. The Multicast Manager Device Performance Alarms parameter must be set to Enable.

Solution:

1. Several users contact the network operator to report that their daily service information is broken.
2. The operator brings up Multicast Manager in the OneClick console and does a search on “All Groups”.
3. One of the Groups shows an Orange condition, indicating a major alarm on one of the elements of the Group.
4. The operator checks each of the devices in the Group and finds an alarm on one of the routers. This alarm is also shown in the OneClick alarm console.
5. A trouble ticket is created. The operator begins troubleshooting the router issue and discovers that one of the interfaces on the router is dropping packets. This generated the alarm on the group model, because the traffic tracked by the performance monitor had fallen below the assigned threshold.
6. The operator resolves the port issue and closes the trouble ticket.
7. Multicast traffic is restored to the network.
8. The operator confirms that service has been restored with users.

Network Configuration Manager

Access Network Configuration Manager

To access Network Configuration Manager from the OneClick Console, select Configuration Manager from the Explorer tab. Then, expand the Configuration Manager node; the Device Families, Policies, and Tasks views appear.

From 10.4.2.1, When the privileged role of the user Allow Load Firmware on Multiple Devices is disabled and load firmware task is attempted on more than one device, an error popup should be thrown with the message Restricted Load firmware for a single device, Select the single device to load firmware. when the privileged role of the user **Allow Load Firmware on Multiple Devices** is disabled and load firmware task is attempted on GC with more than one device, an error popup should be thrown with the message Restricted Load firmware for a single device, maintain single device per GC to load firmware.

> **NOTE**
>
> For more information about OneClick, see the Using OneClick section.

Network Configuration Manager Capabilities

Configuration management is the process of identifying and monitoring configurations of single devices and device families that comprise a network. Devices include routers, hubs, and switches.
Using the DX NetOps Spectrum Network Configuration Manager ensures the following benefits:

- Increases the network uptime by reducing the time to resolve network issues.
- Reduces the network support costs by reducing the occurrence of network issues that require reactive troubleshooting and fixes.
- Reduces the network operational costs by reducing the time to administer system-wide changes.

Each device on the network is configured to provide specific services. Details about how a device operates and how it has been customized are contained in its configuration.
Network Configuration Manager lets you perform the following tasks:

- Manage configurations for supported devices that are modeled in DX NetOps Spectrum or OneClick.
- Capture network device configurations and store them in the DX NetOps Spectrum database.
- Compare running and startup configurations.
- Upload Perl configuration scripts.
- Load firmware.
- Export configurations.
- Load and merge the configurations to one or more devices of the same family type.

  NOTE

  Merging content appends information to an existing file (it does not overwrite or restore).
- Verify that the correct configuration is running on a device.
- Set up a schedule of automatic captures and policies to ensure reliable device configurations.
- Detect performance problems by verifying device configurations.
- Maintain a history of network device configurations for comparison and troubleshooting.
- Create policies to monitor content in configurations and verify that device content is compliant.
- Update NCM one device at a time (From 10.4.2.1)

Introduction

Key Terms

The following terms are important for understanding the Network Configuration Manager.

- **Approval Workflow**
  Lets you require configuration changes that are initiated through the Network Configuration Manager to receive approval before being implemented. An approval workflow can be set up to use CA Service Desk tickets or DX NetOps Spectrum authorization privileges for the approval process.
- **Bulk Task**
  Bulk tasks are tasks that you can run on multiple devices. The following bulk tasks are available: Upload, Sync, Save to Startup, Load Firmware task, Reload, and Cancel Reload.
- **Device Family**
  A group of devices that share common methods to access device configurations. Devices that Network Configuration Manager supports out-of-the-box are automatically placed in a device family. You can use the Extension Utility to create more device families.
- **Global Synchronization Task**
  Gathers running configurations for all devices on your network for which Network Configuration Manager is enabled using a schedule. Select a time period and a recurrence frequency to capture configurations from all network-wide supported devices. By capturing the configurations for all devices on your network, you maintain a running configuration history.
- **Load Firmware Task**
  Uploads the firmware to Cisco IOS and the Cisco IOS - SSH Capable devices.
- **Network Configuration Manager Policy**

Monitors content in configurations and verifies that device content is compliant. Policies specify a certain aspect of a device host configuration. A policy is checked and compared every time a host configuration file is captured for a device. Devices that violate the policy can generate an alarm and can be semi-automatically repaired. A policy is checked for the compliance when a configuration change occurs on a device.

- **Reference Configuration**
  A device configuration that serves as a baseline for reference purposes. You can compare other configurations against the reference configuration. You can have an alarm asserted on the device if the current configuration differs from the reference.
- **Reload Task**
  Reloads a device after firmware has been uploaded. This task is available for Cisco IOS and Cisco IOS - SSH Capable devices.
- **Reusable Task**
  A task that persists after it has been executed and can be run again multiple times without being redefined. You can also create a recurring schedule to run a reusable task at predetermined times.
- **Save to Startup Task**
  Writes a current running configuration to the startup configuration of one or more selected devices. A device saves its configuration in the NVRAM (Nonvolatile Random Access Memory). You can run this task on multiple devices.
- **Single Device**
  Representation of a device in your network that DX NetOps Spectrum is monitoring. Configuring a single device overrides all global device family configurations.
- **Sync Task**
  Captures and verifies policy-compliant device configurations for selected devices on your network and shows the results in real time. When a Sync task captures device configuration, it verifies the configuration against all policies pertaining to the device. You can run this task on multiple devices.
- **Upload Task**
  Merges new content into the running configurations of one or more selected devices. You can run this task on multiple devices.

Types of Configurations

The following sections describe the different configurations for a device.

Running Configuration

A running configuration is a version of a configuration that is loaded on a device and defines how the device currently operates. A running configuration only valid for the current run-time session.

Startup Configuration

A startup configuration is the backup version of a configuration that is stored on a device. The startup configuration is used when the device is rebooted. Some devices have primary and secondary startup configurations. A device replaces the previous running configuration with a copy of the startup configuration when it is rebooted.

Configuration File

A configuration file contains a subset of attributes from a running configuration by device manufacturers. Many devices let Network Configuration Manager capture complete configuration files. You can edit captured configuration files.

Supported Devices

Network Configuration Manager supports the device families of the following vendors out-of-the-box:

- Cisco
- Enterasys
- Enterasys
- Riverstone SSR
- Extreme
- Foundry
- Juniper
- Lancom
- Nortel (Baystack and Passport)

Devices that do not fall into one of the out-of-box supported device families can be configured using the Network Configuration Manager Extension Utility. For the full list of NCM supported devices, refer to Devices Supported by Network Configuration Manager.

Device Families

To receive Network Configuration Manager support, a device must be associated with a device family. Devices that are supported out-of-the-box are automatically assigned to the proper device family. A device can only belong to a single device family.

A Network Configuration Manager device family provides a central place to configure access methods. The access methods are used to access device configurations from other family members. You can override Device family settings at the local device. For more information, see Configure a Device Family.

The Network Configuration Manager Extension Utility lets you create more device families on demand, extending Network Configuration Manager to support more devices and vendors. For more information about manually creating more device families and manually moving devices to user-created device families, see Extension Utility.

How Network Configuration Manager Determines Device Families

Network Configuration Manager automatically determines the device family for devices that are supported out-of-the-box. Typically, this determination is made based on the vendor. For more information, see Supported Devices.

Cisco IOS Devices

The following device families exist for the Cisco IOS devices:

- Cisco IOS - SSH Capable (supports SSH/SCP communication mode)
- Cisco IOS (does not support SSH/SCP communication mode)

To place a device into the Cisco IOS - SSH Capable family, the following conditions must be met:

- The device descriptor must indicate a firmware version of 12.2 (18) or greater.
- The feature set must contain letters "K9" indicating the device has the necessary encryption functionality that is needed for SCP.
- SSH access for the device must be unblocked at the time of discovery.

> **NOTE**
> If SSH access to the device is blocked (for example, with a firewall) at the time of discovery, put the device in the Cisco IOS device family.
> For example, a device with the following description is placed in the Cisco IOS - SSH Capable family:
> Cisco IOS Software, 7200 Software (C7200-JK9S-M), Version 12.3(14)T6, RELEASE SOFTWARE (fc2)Technical Support: http://www.cisco.com/techsupportCopyright (c) 1986-2006 by Cisco Systems, Inc.Compiled Thu 05-Jan-06 05:36 by dchih
> A device with the following description is placed in the Cisco IOS family and is not capable of obtaining configurations using SSH/SCP:

Cisco Internetwork Operating System SoftwareIOS (tm) C2600 Software (C2600-J1S3-M), Version 12.3(17a), RELEASE SOFTWARE (fc2)Technical Support: http://www.cisco.com/techsupportCopyright (c) 1986-2005 by cisco Systems, Inc.Compiled Mon 12-Dec-05 1

Cisco NX OS Devices

The Cisco NX OS devices are supported through scripts that use the Net::SSH::Expect modules. The Perl area must be set up with these modules for an out-of box support for Cisco NX OS devices.
For information about setting up your Perl environment, see Perl Modules (see page).

Juniper JUNOS Devices

(For 10.4.1 and earlier) Network Configuration Manager utilizes the JUNOScript API to communicate with the JUNOS devices. Specifically, the JUNOScript API merge command is used to accomplish uploads, as follows:

<load-configuration format="text" action="merge">

JUNOScript support was developed using JUNOScript version 6.3R1. The new releases of JUNOScript API are typically backward compatible.

The JUNOScript API commands differ from the JUNOS CLI commands. As a result, Network Configuration Manager uploads must use the correct format for the upload to succeed.

For more information, see the documentation website of the Juniper on the JUNOScript API.

Example: Using JUNOScript API Format

The following example illustrates how a command entered from the JUNOS CLI command line differs from the JUNOScript API. The command deletes the snmp location field from a device.

A test device has the following block of configuration text, which sets the snmp location field value to 'Boston':

snmp {
name jun2300-96.4;
description "Juniper J2300 w/ JUNOS 9.0R4 built 2008-11-18 18:55:38 UTC";
location Boston;

The following command can be used from the JUNOS CLI command line to delete the snmp location field on this device:

admin@jun2300-96.4# delete snmp location

The following Network Configuration Manager upload deletes the snmp location field from a device:

snmp {
delete: location;
}

Both operations are equivalent; however, the JUNOScript API syntax must be used with Network Configuration Manager uploads.

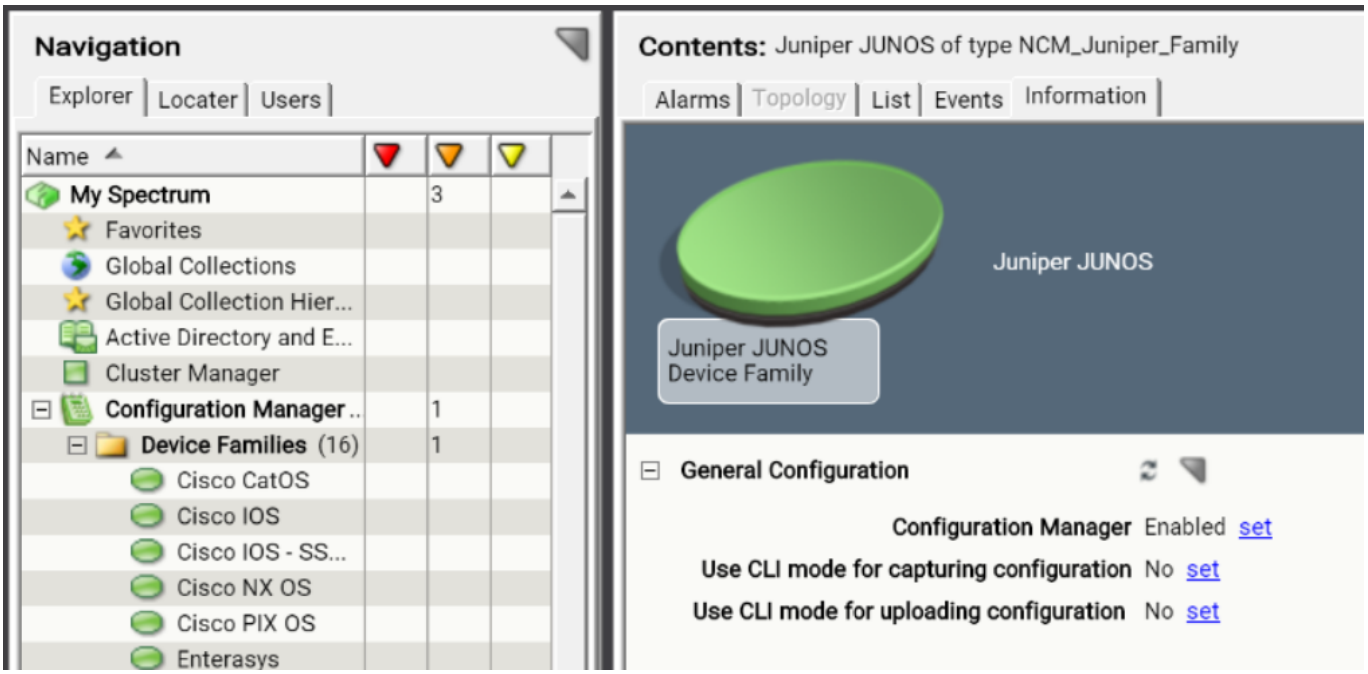
Using CLI Mode for NCM Operations on Juniper JUNOS Device Family

The 10.4.2 release provides the following two new configuration settings in OnceClick for the Juniper JUNOS device family:

- Use CLI mode for capturing configuration
- Use CLI mode for uploading configuration

When you enable these options, NCM starts using the CLI mode to perform the NCM operations (capture and upload) on the Juniper JUNOS devices. Previously, only XML RPC was available for executing the NCM operations on the JUNOS device family. Now, users can decide whether they want to use XMP RPC or CLI mode.

The CLI mode options are disabled by default. Therefore, XML RPC is the default option for the NCM operations on Juniper JUNOS device family. The following screenshot shows that the settings are set to **No** by default:



If you want to use the CLI mode, you can do so by enabling the required configuration settings as follows:

> **NOTE**
> The CLI mode improves the performance as it does not consume high CPU resources while performing the NCM operations.

1. Access the **Explorer** tab in the OneClick UI.
2. Navigate to **Configuration Manager**, **Device Families**, **Juniper JUNOS**.
3. Click the **Information** tab in the **Contents** pane.
4. Expand the **General Configuration** section.
5. Locate the **Use CLI mode for capturing configuration** option for the capture operation and the **Use CLI mode for uploading configuration** option for the upload operation.
6. Click **Set** and change the value to **Yes**.

You have successfully enabled the CLI mode for the NCM operations (capture and upload) on the Juniper JUNOS device family.

Extension Utility

The Network Configuration Manager Extension Utility lets you extend the functionality of Network Configuration Manager beyond its out-of-box support. With the Extension Utility, you can do the following tasks:

- Create more device families on demand. These additional device families can then be configured to extend the Network Configuration Manager functionality on more devices by using Perl scripts. For more information about creating device families, see Create a Custom Device Family. For more information about configuring scripts, see Extension Utility Script Configuration.
- Manage more devices and vendors by using Perl scripts for any of the operations Network Configuration Manager executes on a device. The operations such as capturing or writing a startup configuration. The operations also include capturing or uploading a running configuration; and uploading device firmware, reloading a device, and canceling the reload operation on a device. Scripts can be configured within Network Configuration Manager for each of

these operations. For more information about using customized scripts to perform these operations, see Network Configuration Manager Extension Utility.

NOTE
From 10.4.2, you can use Perl scripts to perform NCM operations on the SDC-modeled devices, too.

- Create customized trap settings for your installation which can be used to correlate the configuration change event information. For more information, see Configure Notification Trap Settings.

Network Configuration Manager Prerequisites

To run Network Configuration Manager and actively maintain a running history of device configurations on the managed network, take the following steps:

- Model devices with read/write community strings if you are using SNMP. For more information, see the Modeling and Managing Your IT Infrastructure section.
- Verify that devices are SCP-enabled if you are using SSH. For more information, see Communication Modes.

Communication Modes

The following table lists communication mode support for devices that are supported in Network Configuration Manager. An 'X' in a column indicates that the communication mode is supported for that device family. When a Perl script is the only way to communicate with the device, you are notified about the method that the script uses.
See Configure a TFTP Server to enable configuration capture and loading for devices that use the SNMP/TFTP communication mode.

Device Family	SNMP/TFTP	Telnet/FTP	SSH/SCP	SSH/TFTP	Perl
Cisco CatOS	X				X
Cisco IOS	X	X			X
Cisco IOS-SSH Capable	X	X	X		X
Cisco NX OS					SSH
Cisco PIX OS					Telnet
Enterasys	X				X
Enterasys/ Riverstone SSR	X				X
Extreme	X				X
Foundry	X				X
Juniper JUNOS			X		X
Lancom LCOS					TFTP/Telnet
Nortel Baystack				X	X
Nortel Passport 8600	X				X
Nortel Passport L3	X				X

SSH v2 Support

Network Configuration Manager supports SSH v2 only. Network Configuration Manager does not support SSH v1. The Cisco devices that support SSH v1 only are not automatically placed in the Cisco IOS-SSH Capable family.
Network Configuration Manager does not support the Juniper devices that support only SSH v1.
To support SSH v2, install or update the firmware on a Cisco or Juniper device. Add the device by following the steps in Place a Device in a Device Family.

Cisco Devices and SCP

The Cisco devices must have Secure Copy (SCP) enabled to use the SSH communication mode.
For more information about SCP, see the documentation for the Cisco IOS Secure Copy feature at http://www.cisco.com.

Unsolicited Notifications of Device Configuration Changes

Network Configuration Manager attempts to capture device configurations immediately after any change occurs. An unsolicited notification of configuration change can be either traps or MIB objects that are sent from the device where the change occurred.

Some devices send SNMP traps when their configuration has changed. The SpectroSERVER then performs a capture and saves the configuration in the database to provide updated configuration data. Network Configuration Manager policies are tested against the most recent configuration captures. For more information, see Device Traps.

Selected information can be parsed from these configuration trap notifications and shown in the Host Configuration table. For more information, see Configure Notification Trap Settings.

Instead of or in addition to sending an SNMP trap, some devices update MIB attributes to signal configuration changes. SpectroSERVER then polls the MIB and captures new configurations when it recognizes changes in the attributes. For more information, see Device MIB Objects (see page).

Network Configuration Manager monitors notifications on a subset of supported devices. You can extend Network Configuration Manager to monitor more traps and MIB objects from other supported devices.

Enabling Unsolicited Notifications of Device Configuration Changes provides the most recent and up-to-date configuration captures for devices in your network. You can disable this feature to avoid unnecessary captures, which involve TFTP transfers that can degrade network performance. For more information, see Configure General Configuration (see page) and Configure Unsolicited Device Configuration Captures on a Single Device.

Device Traps

Network Configuration Manager supports the following two traps:

- Cisco: ciscoConfigManEvent 1.3.6.1.4.1.9.9.43.2
- Juniper: jnxCmCfgChange 1.3.6.1.4.1.2636.4.5

When either of these traps are received, DX NetOps Spectrum generates event 0x00821029. This event then triggers Network Configuration Manager to perform a capture. If you want to trigger a capture for other supported devices, map more configuration change traps to that event.

Device MIB Objects

When any configuration changes occur, the Network Configuration Manager polls MIB objects through the model attributes to determine. This feature is supported on Cisco and Juniper devices that support the following MIB objects:

- CISCO-CONFIG-MAN-MIB: ccmHistoryRunningLastChanged 1.3.6.1.4.1.9.9.43.1.1.1
- JUNIPER-CFGMGMT-MIB: jnxCmCfgChgLatestTime 1.3.6.1.4.1.2636.3.18.1.2

You can extend the attribute polling mechanism to other supported devices. Use the Model Type Editor to create the attribute to poll for configuration change notifications, making it a polled attribute. Then, set the value of the Config_Change_AttrID attribute (0x12bf8) to the attribute ID of the newly created polled attribute. Network Configuration Manager then monitors this attribute for the notification of configuration changes and performs a capture.

Global Collections

A global collection lets you organize views of network devices. A global collection contains devices from multiple vendors. The global collections are useful when executing bulk tasks or creating the Network Configuration Manager policies.

For more information about Global Collections, see the Modeling and Managing Your IT Infrastructure section.

Maintenance Mode

Network Configuration Manager is disabled for any device that is in maintenance mode. To verify whether the device is in maintenance mode, select the device from the Explorer tab and then click the Information tab. Under the General Information view, see the In Maintenance option. If this option is set to "yes", the device is in maintenance mode.

Network Configuration Manager Report Packs

Network Configuration Manager report options are included under the Network Configuration Management report pack in DX NetOps Spectrum Report Manager. Report Manager provides numerous report content, format, and report organization options. You can generate reports with the appropriate type and scope of information for different audiences in your organization who are interested in device configuration changes.
For more information, see Network Configuration Manager Reports from Report Manager and the Report Manager section.

NCM Script Mode Enhancement

From 10.4.2, DX NetOps Spectrum uses the NCM Service (ncmservice) instead of the SRAdmin process to perform the NCM script operations. This implementation helps scale up the NCM script operations. The NCM Service configuration file (config.xml) is available in the <SPECROOT>/NCM folder.

Review the following considerations:

- (For SpectroSERVER) With the PERL script execution, if the upload task fails with a timeout error, try to keep the timeout values low in the script for the `upload_timeout_in_seconds` and `login_timeout_in_seconds` parameters, and increase the thread timeout value in the ../NCM/config.xml file. Change the thread timeout value of the following parameter:
  `<thread-timeout-value type="java.lang.Integer">3600</thread-timeout-value>`
  NOTE
  This parameter specifies the maximum time for which the NCM Service waits for the completion of the script execution. If the script execution takes longer time, increase the value. The default value of this parameter is 300 seconds.
- (For SDC) With the PERL script execution, if the upload task fails with a timeout error, try to keep the timeout value low in the script for the `upload_timeout_in_seconds` and `login_timeout_in_seconds` parameters, and increase the thread timeout value in the ../NCM/config.xml file on SDC. Change the thread timeout value of the following parameter:
  `<thread-timeout-value type="java.lang.Integer">3600</thread-timeout-value>`
  Also, try to increase the Secure Domain timeout value under the Secure Domain Manager configuration field, which should not be more than 10 minutes.

Network Configuration Manager Configurations

Configure Network Configuration Manager

This section describes the fundamental configurations for Network Configuration Manager.

Perform General Configuration

Select some initial settings to determine how Network Configuration Manager performs configuration captures and correlates change events.

Follow these steps:

1. Select Configuration Manager in the Explorer tab.
   Information and settings appear in the Information tab of the Contents panel.
2. Expand the General Configuration subview.
   The General Configuration options appear.
3. Modify the following General Configuration options as needed:
   - **Unsolicited Device Configuration Captures**
     Enables or disables Network Configuration Manager from capturing the configuration of a device when it receives an unsolicited notification from a device. An unsolicited notification of configuration change can be either traps or MIB objects that Network Configuration Manager is monitoring for changes.
   - **Correlation Event Period (seconds)**
     Specifies the amount of time during which configuration change events are correlated. All configuration change events for a particular device that occur during this period is combined into a single event.
     Default: 120
   - **Capture Newly Modeled Device's Configuration**
     Specifies how to handle newly modeled devices on your network at a global level. The available values are:
   - **On Next Global Sync**
     a. Captures newly modeled devices according to the global synchronization schedule.
   - **Do Not Capture**
     a. Disables Network Configuration Manager on the newly modeled device. To enable the Network Configuration Manager functionality, manually enable Network Configuration Manager on the device.
   - **Immediately**
     a. Captures newly modeled devices immediately (once they have been modeled) rather than waiting for the global synchronization to run.
   - **Task Work Queue Size**
     Specifies the maximum number of devices that are processed in parallel on each SpectroSERVER.
     When manually stopping a task that is running, all devices currently in the queue are processed after the stop command is received.
     Default: 10
     Maximum: 40 (in 10.4.2), 100 (in 10.4.2.1), and 200 (10.4.2.2 and above)

Select Configuration History Settings

The following procedure describes how to control the storage of captured configurations. You can maintain captured configurations either by the number of configurations that are kept per device or by a length of time.

> **WARNING**
> When specifying how to store captured configurations, consider the impact on the SpectroSERVER database. If too many configurations are retained, it is possible to fill the SpectroSERVER database with configuration file models.

Follow these steps:

1. Select Configuration Manager in the Explorer tab.
   Information and configurations display in the Information tab of the Contents panel.
2. Expand the Configuration History subview.
   The options that are used to control how captured configurations are stored appear.
3. Select one of the following options:
   - **Specify maximum number of configurations to be stored per device.** This option stores captured configurations per device that is based on a specified number.
     - **Maximum Stored Configurations Per Device**

Specifies the maximum number of stored configurations per device. For example, a number of 25 indicates that the latest 25 configurations for each device reside in the DX NetOps Spectrum database.
Default: 25

- **Specify maximum number of days configurations to be stored.** This option stores captured configurations that are based on a length of time.
  - **Maximum Days Host Configuration Stored**
    Specifies the maximum number of days a host configuration is stored before being destroyed.
    Note: Depending on how frequently configurations are captured, specifying a large time period may cause the SpectroSERVER database to fill with configuration file models.
    Default: 30 (days)
  - **Minimum Stored Configurations Per Device**
    Specifies the minimum number of host configurations that are stored per device. Configurations are maintained even if they have aged out to remain at this minimum value.
    Default: 5

Select Configuration Change Alert Settings

Configuration Change Alert settings control which configuration change events trigger alarms and the types of alarms that are generated. You can select Configuration Change Alert settings to determine the alarms that you see.

Follow these steps:

1. Select Configuration Manager in the Explorer tab.
   Information and configurations display in the Information tab of the Contents panel.
2. Expand the Configuration Change Alert subview.
   The Configuration Change Alert options display.
3. Modify the following Configuration Change Alert options as needed:
   - **Alert Mode**
     Specify the events that trigger an alarm.
     - **Alarm On Any Changes**
       An alarm is generated for configuration changes only.
     - **Alarm On Any Reference Violations**
       An alarm is generated for reference configuration violations only.
     - **Alarm On Any Reference Violations or Changes**
       An alarm is generated for both reference configuration violations and configuration changes.
     - **No Alarm**
       No alarms are generated for any configuration changes.

     Default: No Alarm
   - **Reference Violation Alert Type**
     Specify the type of alarm or event that is asserted when a reference configuration violation occurs. The existing comparison mask is used to determine significant differences between current and reference configurations. Reference violation alarms are automatically cleared when the current configuration matches the reference configuration.
     For information about setting a reference configuration, see Specify a Reference Configuration.
     Valid values are critical, major, and minor alarms and events only.
     Default: Event Only
   - **Configuration Change Alert Type**Specify what type of alarm or event only is asserted when any configuration change occurs.
     Valid values are critical, major, and minor alarms and events only.
     Default: Event Only

Approval Workflows

Approval workflows let you require configuration changes that are initiated through Network Configuration Manager to receive approval before being processed. Approval workflows can be set up to use CA Service Desk tickets or DX NetOps Spectrum authorization privileges for the approval process.

This section describes how to configure approval workflow options. It also describes how to approve tasks if the OneClick approval workflow mode is enabled.

For information about initiating configuration changes using tasks, see Network Configuration Manager Device-Level Tasks, and Network Configuration Manager Bulk Tasks.

> **NOTE**
>
> For more information, see the DX NetOps Spectrum and CA Service Desk Integration.

Configure Workflow Options

Approval workflows let you require configuration changes that are initiated through Network Configuration Manager to receive approval before being processed. Configure approval workflow options to determine how approvals are requested and processed.

Follow these steps:

1. Select Configuration Manager in the Explorer tab. Information and configurations display in the Information tab of the Contents panel.
2. Expand the Workflow subview to open the approval workflow options.
3. Modify the following approval workflow settings as needed:
   Approval Workflow Mode

Specifies whether approval is required for all operations that modify a device. These operations include Upload, Save to Startup, Load Firmware, Reload, and the Cancel Reload tasks.

Disabled Specifies that configuration changes initiated in Network Configuration Manager do not require approval.

ServiceDesk
Specifies that configuration changes initiated in Network Configuration Manager must gain approval through CA Service Desk. When a task is created, a CA Service Desk ticket is generated. If approved, the task is placed into a state in which it can be processed.

If this option is selected, the Configure button is enabled. Click the Configure button to invoke the ServiceDesk Workflow Configuration page, where you can set the initial values for the following fields:
Error Type - Error type values are configured in Service Desk for integrated use with DX NetOps Spectrum. For more information about these values, see the CA Service Desk Implementation section.
Approved Status, Denied Status, Canceled Status, Awaiting Approval Status - Different status values are available depending on the error type. The Status values are configured in CA Service Desk for integration with DX NetOps Spectrum. For information about setting up these values, see the DX NetOps Spectrum and CA Service Desk Integration.

> **NOTE**
>
> If Service Desk approval is enabled and the user who creates the task has Task Approver privileges, CA Service Desk approval is optional. For more information, see Network Configuration Manager Privileges.

OneClick
Specifies that configuration changes that are initiated in Network Configuration Manager can be processed only if initiated or approved by a user with the Task Approver permission. **Default:** Disabled

Include Configuration Changes in Approval Process
Specifies whether configuration content is included in the approval request.
Default: No

NOTE

A user must have the Hide Configuration Changes from Approval Requests permission for this option to take effect. For more information, see Network Configuration Manager Privileges.

Approve a Task in OneClick

If the OneClick approval workflow mode is enabled, a user having the Task Approver permission must approve the tasks from the OneClick console.

NOTE

You can also approve or deny a task from an email notification. When approval is requested for a task, an email is generated and sent to the task approver for approval. Included in the email are links to let you approve or deny the task. Select the appropriate link. The State is updated to reflect whether the task is Approved or Denied.

Follow these steps:

1. Select 'task' in the Tasks folder under the Configuration manager in the Explorer tab.
   The available tasks appear in the List tab of the Contents panel.
2. Right-click the task and select Approve Task, Deny Task, or Cancel Approval Request, as appropriate, from the right-click menu.
   The task State is updated to reflect whether the task is Approved, Denied, or Canceled, respectively.

Configure a TFTP Server

This section describes how to start a Trivial File Transfer Protocol (TFTP) server on a SpectroSERVER system. TFTP transfers configuration files. This process consists of two steps:

- Setting up your system as a TFTP server. This step varies by platform.
- Specifying the TFTP Configuration settings in OneClick.

If you have a distributed SpectroSERVER (DSS) environment, the TFTP servers must be running on every SpectroSERVER to enable Network Configuration Manager functionality.

See Communication Modes for supported device family communication modes.

NOTE
Verify that each device in your network is properly modeled using the appropriate community name (read or write).

Set Up System as TFTP Server

This section describes how to set up your system as a TFTP Server. Instructions vary by platform.

NOTE
Your TFTP server can run on a system other than the SpectroSERVER host system. But the SpectroSERVER computer must be able to access the root directory of the TFTP server, and the root directory on the SpectroSERVER computer must be shared with the TFTP server. For more information, see Considerations When Using Remote TFTP or FTP Servers.

1. Verify that the /etc/services file contains a TFTP entry. To search for the entry, enter the following commands:
   ```
   cd /etc
   grep tftp services
   ```
   You see the following entry in the /etc/services file:
   ```
   tftp          69/udp
   ```
   If this entry does not appear, edit the services file and add it to the “Host specific functions” section.

2. In the /etc/inetd.conf file, find the following line and uncomment it by deleting the pound character (#) from the beginning of the line:
   ```
   #tftp dgram udp6 wait root /usr/sbin/in.tftpd in.tftpd -s /tftpboot
   ```
3. Verify that the entry ends with the -s /tftpboot option. The ending specifies the tftp directory (in this case, /tftpboot).
4. Run the inetconv command.
5. Verify that the tftp service is enabled:
   ```
   svcs | grep tftp
   ```
   The following response is diplayed:
   ```
   online Apr_10 svc:/network/tftp/udp6:default
   ```
6. When your system is set up as a TFTP server, verify that devices are modeled with the read/write community string for TFTP transfer to work.
7. Configure the TFTP settings in OneClick as described in TFTP Configuration Settings.

Configure a Linux System to Support TFTP

The following procedure sets up your Linux system to support TFTP.

Follow these steps:

1. Log in as root.
2. Verify that a TFTP server is installed on your system by running the following command:
   ```
   %rpm -q tftp-server
   ```
   The following message indicates that the TFTP server is installed:
   ```
   tftp-server-<version>.EL3.1
   ```
   If this message does not appear, a TFTP server is not installed. Take the following steps:
   a. Download the TFTP package from the Red Hat website at http://www.redhat.com.
   b. Run the following command:
      ```
      % rpm -i <package.rpm>
      ```
   c. Follow the instructions for installing the rpm package from the Red Hat website.
3. Create the /tftpboot directory and give all users read/write permission to the directory using the following commands:
   ```
   mkdir /tftpboot
   chmod 777 /tftpboot
   ```

   NOTE
   Your TFTP server can run on a system other than the SpectroSERVER host system. But the SpectroSERVER computer must be able to access the root directory of the TFTP server, and the root directory on the SpectroSERVER computer must be shared with the TFTP server. For more information, see Considerations When Using Remote TFTP or FTP Servers.
4. Change to the /etc/xinetd.d directory.
5. Edit the file named tftp as follows:
   ```
   Set disable=no
   ```
6. Save and close the file.
7. Run *one* of the following commands to restart xinetd services:
   – % service xinetd restart
     The following message appears:
     ```
     Stopping xinetd  OK
     Starting xinetd  OK
     ```
   – % killall -HUP xinetd
8. Verify that the TFTP server is running.

NOTE
One method of verification is to run a Network Configuration Manager capture. If you receive a TFTP timeout error /event 0x821001, it indicates that TFTP is not running.

9. Configure the TFTP settings in OneClick as described in TFTP Configuration Settings.

TFTP Servers on Windows

A TFTP server is not typically available on Windows. If you use the TFTP communication mode for any device family, set up a TFTP server. Multiple free or commercial applications are available.

With a TFTP Server on a remote host, see Considerations When Using Remote TFTP or FTP Servers to use that server with Network Configuration Manager.

To configure your Windows system to support TFTP, complete the procedure that is described in TFTP Configuration Settings. The steps in this procedure are applicable to any TFTP server that you install.

Configure a TFTP Server on Windows

This section describes how to configure TFTP settings in OneClick.

NOTE
Before you perform this procedure, make sure that your system has been configured as a TFTP server. For more information, see Set Up a System as a TFTP Server.

Follow these steps:

1. Select Configuration Manager in the Explorer tab.
   Information and configurations display in the Information tab of the Contents panel.
2. Expand the TFTP Configuration subview.
   The TFTP Configuration table appears.
3. Modify the following as needed. Click set to edit a particular field, and press Enter when you have finished.
   - **Default TFTP Host**
     TFTP server IP address for the landscape, by default, the host system running the SpectroSERVER.
     This field lets you change the IP address for the TFTP server globally. For considerations when using a remote host, see Considerations When Using Remote TFTP or FTP Servers.

     NOTE
     The attribute DefaultTftpHost can be configured in the Attribute Editor.
   - **Default TFTP Directory**
     Pathname where TFTP is running. Click set, and enter a valid TFTP server path, such as:
     - For Unix systems, /tftpboot
     - For Windows, C:\win23app\SPECTRUM\NCM\tftp

     NOTE
     Your TFTP server can run on a system other than the SpectroSERVER host system. But the SpectroSERVER computer must be able to access the root directory of the TFTP server, and the root directory on the SpectroSERVER computer must be shared with the TFTP server. For more information, see Considerations When Using Remote TFTP or FTP Servers.

Configure an FTP Server

Configure Network Configuration Manager to use a local FTP server on a SpectroSERVER system (see Communication Modes for supported device family communication modes).

If you are deploying devices that use FTP for file transfers, configure an FTP server. We recommend installing and configuring a native FTP server for your platform. For the Windows platforms, the following links describe how to install and configure the native FTP service.

- Windows Server 2012: http://www.c-sharpcorner.com/UploadFile/cd7c2e/how-to-install-ftp-server-on-windows-server-2012/
- Windows Server 2016: https://www.avoiderrors.com/install-configure-ftp-server-server-2016/

Follow these steps:

1. Select Configuration Manager in the Explorer tab.
   Information and configurations display in the Information tab of the Contents panel.
2. Expand the FTP Configuration subview.
   The FTP Configuration table appears.
3. Modify the following settings as needed. Click set to edit a particular field and then press Enter.
   - **Default FTP Host**
     FTP server IP address for the landscape. By default, the SpectroSERVER runs on this host system.
     This field lets you change the IP address for the FTP server globally. For more information, see Considerations When Using Remote TFTP or FTP Servers.

     NOTE
     The attribute DefaultFtpHost represents this value, which can be configured in the Attribute Editor.
   - **FTP Username**
     FTP username.
   - **FTP Password**
     FTP password.
   - **Default FTP Directory**
     Pathname where FTP is running.
     Note: Your FTP server can run on a system other than the SpectroSERVER host system. But certain requirements apply to the directory. For more information, see Considerations When Using Remote TFTP or FTP Servers.
   - **Landscape**
     DX NetOps Spectrum landscape (for display only).

Considerations When Using Remote TFTP or FTP Servers

By default, the host system running the SpectroSERVER is also the host system for both the TFTP and FTP servers. However, you can set up your TFTP or FTP server to run on a different host system.

To set up your TFTP or FTP server to run on a different host globally, use the Default TFTP/FTP Host and Default TFTP/FTP Directory fields as described in TFTP Configuration Settings and Configure an FTP Server. You can also override the default values for the host by using the Attribute Editor as described in Specify TFTP or FTP Server for a Single Device.

NOTE
Although you can override the TFTP and FTP server host system by device, TFTP and FTP directory settings apply to the entire landscape.

When using a remote host for the TFTP or FTP server instead of the local system where the SpectroSERVER runs, consider the following points:

- Both the specified TFTP and FTP directories must be locally accessible from the DX NetOps Spectrum host system. Share the root directory of the TFTP or FTP server with the computer running the SpectroSERVER.
  - For Unix systems, the remote directory must be mounted using read/write nfs mount.
- When specifying the pathname, use the UNC paths only; local variables or locally mapped directories are not allowed. For example, to access the shared folder 'tftpboot' on the host 'tftpserver', specify the UNC path of \\tftpserver\tftpboot as the default TFTP directory.
- On the Windows systems, the UNC path cannot require a username and password and requires read and write privileges.

NOTE
Because mapped drives are not supported, mapping the network drive and providing a username and password do not circumvent the requirement.

Specify TFTP or FTP Server for a Single Device

The following procedure describes how to specify a separate host system for the TFTP or FTP server at the device level.

NOTE
To set the TFTP or FTP host globally for the landscape, use the Default TFTP Host and Default FTP Host fields, as described in TFTP Configuration Settings and Configure an FTP Server, respectively. Although you can override the TFTP and FTP server host system by device, TFTP and FTP directory settings apply to the entire landscape.

Follow these steps:

1. Select the devices that will use the TFTP or FTP servers on the separate host in the Explorer tab.
2. Click the List tab in the Contents panel and select the devices.
3. Select Utilities from the Tools menu, and select Attribute Editor.
   The Attribute Editor opens.
4. Select the User Defined folder, and click Add.
   The Attribute Selector window appears.
5. Enter “host” in the Filter field of the Attribute Selector window. Select the NCM_FTP_Host and NCM_TFTP_Host attributes and click OK.
   These two attributes now appear under the User Defined folder.
6. Select both the NCM_FTP_Host and NCM_TFTP_Host attributes and click the add arrow.
   Values that you can modify appear in the right pane.
7. Modify the following values for each attribute:
   - **No Change**
     Clear the check box to enable the remaining fields.
   - **IP address**
     Enter the IP address of the host system running the TFTP and FTP protocols.

     NOTE
     If using NAT, use the public IP address.
   - **Set As Default**
     If selected, all newly created devices automatically inherit this value.
8. Click OK. If a confirmation dialog opens, click Yes.
   The Attribute Edit Results page shows the results of the change.
9. Click Close.

Select Settings for Device Configuration Export

You can configure Network Configuration Manager to export device configurations to a text file for historical archiving purposes. You must manually manage this file system outside of DX NetOps Spectrum and OneClick.

Follow these steps:

1. Select Configuration Manager in the Explorer tab.
   Information and configurations appear in the Information tab of the Contents panel.
2. Expand the Export Configuration subview.
3. Click set next to Export Configuration. The default is “Do Not Export.” Select one of the following options:
   - **Export Unique Configurations Only**

Export device configurations only if they differ from previously captured configurations.

- **Export Unique and Global Sync Configurations**
  Export device configurations only if they differ from previously captured configurations or on a global synchronization. For example, one file per device is generated each day if you have configured a global synchronization to run on a daily basis. See About Global Synchronization for more information.

The export configuration displays next to Export Configuration.

4. Click set in the Export Directory column. Then specify a local directory in the SpectroSERVER in which to export configuration text files for Linux and/or Windows. The export files are named with a device name and a timestamp. If you want to export configuration text files to a network share, specify the UNC path to the directory. For example, \ \Shared_Server\Export\ExportFiles.
5. Press Enter.
   The export directory appears.

Configure a Device Family

The device family provides a central location to configure Network Configuration Manager interactions with devices in the device family. Configurations that are made to a device family take effect on all devices that are contained within the family. Device family settings can be overridden at the local device level. For more information, see Configure a Single Device to Override Device Family Settings.

To access the configurations for a device family, select a device family from Device Families in the Explorer tab. Then select the Information tab in the Contents panel. Device family configurations are shown.

The Extension Utility lets you configure a Perl script to handle device interaction for any of the supported Network Configuration Manager operations. For more information, see Network Configuration Manager Extension Utility.

Configure Device Family General Settings

The General Configuration subview contains the Configuration Manager settings. Configuration Manager lets you disable tasks for an entire device family. When Configuration Manager is set to disabled, Network Configuration Manager operations are not performed on any of the devices that are contained by this device family.

> **NOTE**
> Configuration Manager can also be disabled at the local device level if any devices in the Device Family require it. For more information, see Configure a Single Device to Override Device Family Settings.

Follow these steps:

1. Navigate to device family configurations as described in Access Network Configuration Manager Device Family Configurations and expand the General Configuration subview.
   The general configurations for the selected device family appear.
2. Click set next to Configuration Manager to enable or disable Network Configuration Manager tasks and functionality for the device family. Configuration Manager is enabled by default.
   The status of device family communication with Network Configuration Manager displays next to Configuration Manager.

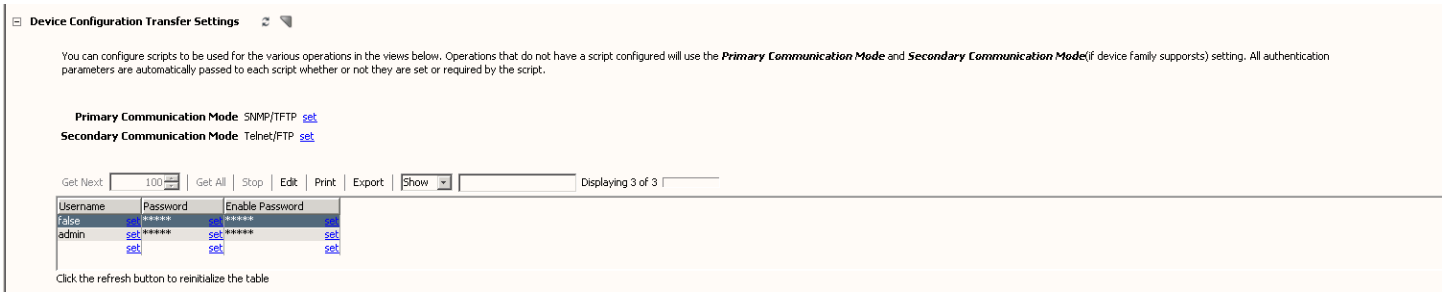
> **NOTE**
> In 10.4.2, two new configuration settings (Use CLI mode for capturing configuration and Use CLI mode for uploading configuration) have been provided in the General Configuration subview for the Juniper JUNOS device family. When you enable these options, Network Configuration Manager (NCM) starts using the CLI mode to perform the NCM operations (capture and upload) on the Juniper JUNOS devices. Previously, only XML RPC was available for executing the NCM operations on the JUNOS device family. Now, users can decide whether they want to use XML RPC or CLI mode. For more information, see the "Using CLI Mode for NCM Operations on Juniper JUNOS Device Family" section in Network Configuration Manager Introduction.

Configure Device Family Communication Mode

All device families that are supported out-of-the-box have a communication mode that determines how Network Configuration Manager (NCM) interacts with the associated devices. Some device families with the default support let you select from multiple communication modes. Depending on the communication mode that is selected, the device username, password, and enable password can be required.

If not all devices in the family can be accessed using the same username, password, and enable password, you can override the usernames and passwords at the local device. For more information, see Configure a Single Device to Override Device Family Settings.

The following image shows the Primary Communication Mode and Secondary Communication Mode setting, which appears in the Device Configuration Transfer Settings subview:



Follow these steps:

1. Select a device family from **Device Families** in the **Explorer** tab, and then select the **Information** tab in the **Contents** panel.
2. Expand the **Device Configuration Transfer Settings** subview to view the communication mode configurations for the device family.
3. Select the primary and secondary communication mode for the selected device family.
   The selected communication mode displays next to Communication Mode.

Modify the following fields as needed:

> **NOTE**
>
> You can set multiple (maximum three) Usernames and Passwords.

- **Username**
  Specifies the user name for accessing the devices.
- **Password**
  Specifies the password for accessing the devices.
- **Enable password**
  Specifies the second password for configuring the devices (supported for Cisco IOS, Cisco IOS-SSH Capable, and Foundry devices only).

The communication mode for the selected device family is configured.
When you perform a capture or upload capture or operations that are supported by NCM, the action tries with Primary Communication Mode and the first Username, Password combination. If the combination fails, it tries with the remaining Username, Password combinations. If all the Username, Password combinations fail with Primary Communication Mode, then it tries with the Secondary Communication Mode with all the Username and Password combinations till the action is success.

NOTE

- If the primary or secondary communication consists of any incorrect username or password, then the device stops responding to the next available credentials it supports. This behavior is due to the connections limit per minute value in the device configuration.
- By default, the communication retries value in DX NetOps Spectrum is set to 5 attempts and the retry attempts in devices also set to 5. To make the communication mode work and try with the next correct credentials, you need to either reduce the retries value in DX NetOps Spectrum or increase the retries value in devices configuration.
- To update the retries value in DX NetOps Spectrum, open $SPECROOT/NCM/config.xml, then add the 'tries type' attribute under the 'root' tag and close the xml file. For example, <tries type="java.lang.String">3</tries>.
- To update the retries value in device configuration, refer to the documentation provided by the device vendor.
- If the script is configured to capture the configuration for the out-of-the-box device family, then primary and secondary communication modes will be ignored.
- If some of the devices are not capturing the configuration with the primary and secondary communication modes, then create a custom family and move the devices from the out-of-the-box family to the custom family. Configure the script and try to capture the configuration.

Configure Device Family Masks

Configure device family masks to exclude device configuration content from configuration comparisons or to hide sensitive information from unauthorized users. Mask configurations are discussed in the following sections.

Device Family Comparison Mask

The Comparison Mask is a list of regular expressions that conceal device configuration content during a comparison with a historical configuration. Any line in the device configuration file that matches a regular expression in the Comparison Mask is ignored during comparisons of configuration files. Network Configuration Manager provides a list of predefined masks that are displayed in the window next to Comparison Mask.

You can add or remove masks.

The mask setting can be overridden at the local device level. For more information, see Configure a Single Device to Override Device Family Settings.

Add a Device Family View Mask

The View Mask is a list of regular expressions that conceal device configuration content from users who lack the OneClick privilege to view the entire device configuration file. The View Unmasked Configurations privilege is required to view the contents of the View Mask field. Use this setting to hide passwords or other content from unauthorized users. You can override a mask setting at the local device level.

Follow these steps:

1. Select Add under Comparison Mask or View Mask.
   The Add dialog opens.
2. Type the mask for the selected device family. For example, for comment lines, enter: [!#]. Supply any regular expression.
3. Click OK.
   The content that you entered for the mask appears. You have set a mask for all devices in the device family.
4. Repeat the previous steps to enter more masks.

For more information about overriding device family settings at the local device, see Configure a Single Device to Override Device Family Settings.

Enter a Mask

You can enter a mask that applies to all devices in a device family.

Follow these steps:

1. Select Add under Comparison Mask or View Mask.
   The Add dialog opens.
2. Enter the mask for the selected device family. For example, for comment lines, enter: [!#] or enter any regular expression.
3. Click OK.
   The content that you entered for the mask displays. You have now set a mask for all devices in the device family.

Configure Notification Trap Settings

You can configure DX NetOps Spectrum to automatically capture device configuration based on trap notifications from a device. You can customize these trap settings for your installation. Specify the information that is parsed from the configuration change trap notifications from the device and displayed in the Host Configuration table. Network Configuration Manager uses these settings to correlate configuration change event information so that events are combined for a particular device.

> **NOTE**
> The Unsolicited Device Configuration Captures setting controls the automatic configuration capture for a device. For more information about this feature, see Unsolicited Notifications of Device Configuration Changes.

Trap format information varies by device family. Out-of-the-box support is provided for Cisco CatOS, Cisco IOS, Cisco IOS - SSH Capable, and Juniper JUNOS device families. The following example shows the Syslog traps that are defaults for the Cisco IOS - SSH Capable device family:

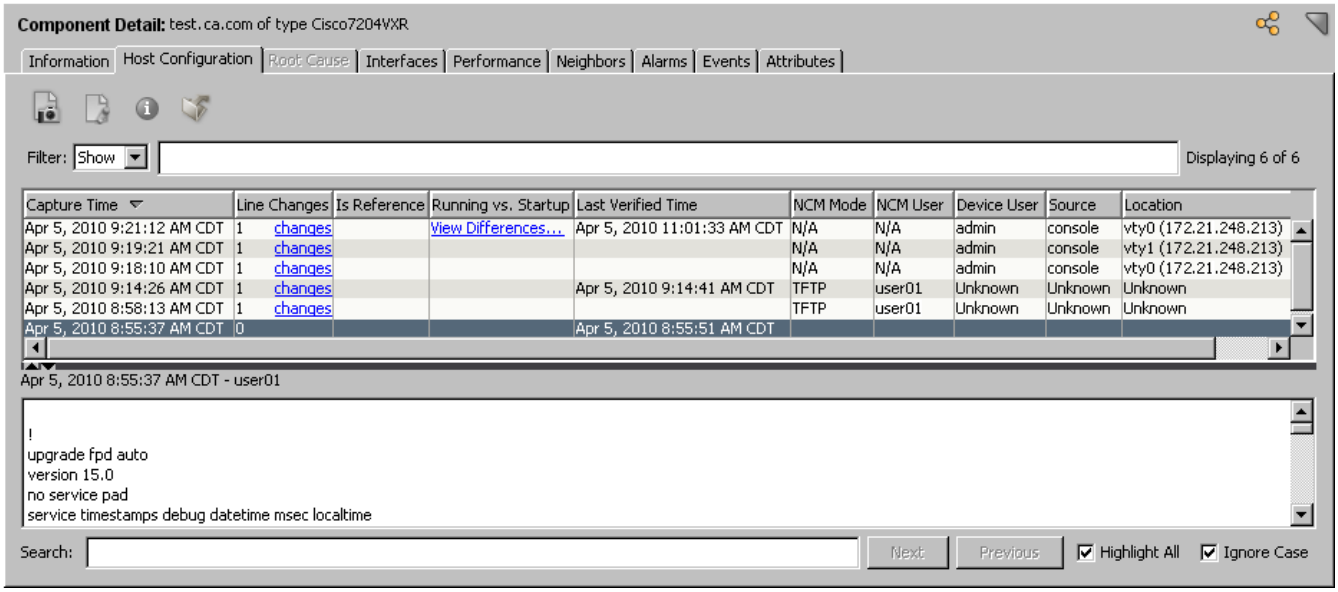
```
Configured from {SOURCE} by {USER} on {LOCATION}
Configured from {LOCATION} by {SOURCE}
```

The following variables represent information that is parsed out from the trap message and shown in the Host Configuration table:

- **SOURCE**
  Corresponds to Source column in the Host Configuration table.
- **USER**
  Specifies the user who was logged in on the device when the changes were made. This value corresponds to the Device User column in the Host Configuration table.
- **LOCATION**
  Corresponds to the Location column in the Host Configuration table.

You can also specify additional message formats if trap messages from the Syslog server are in non-default formats.

The following image shows the Host Configuration table for a Cisco IOS - SSH Capable device, including the table columns:



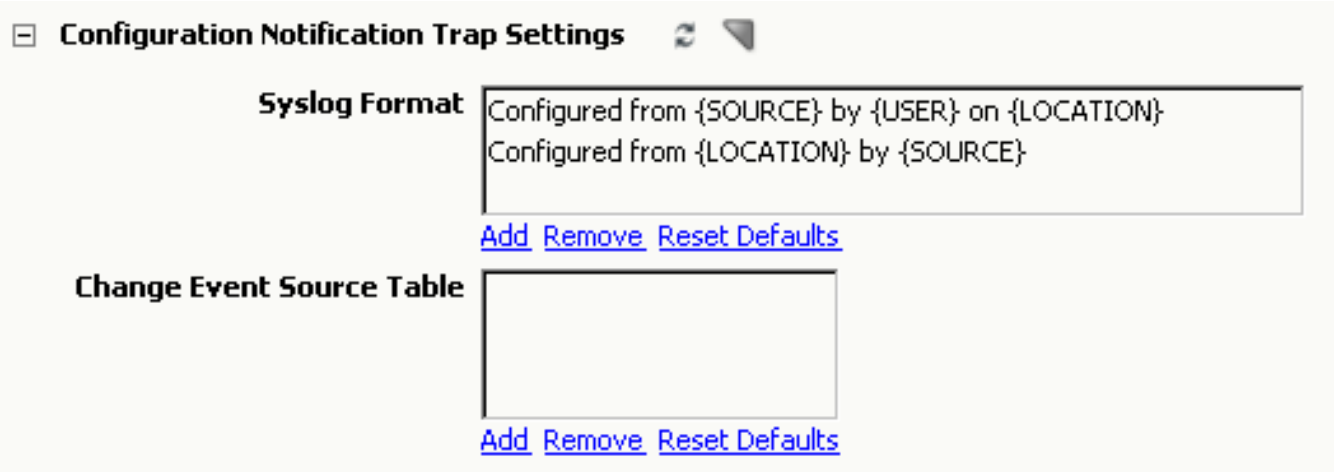
Information is correlated based on device traps, Syslog traps and events, Network Configuration Manager internals, and any other trap that is mapped to the generic change event. Correlation Event Period parameter in Network Configuration Manager General Configuration determines the amount of time during which configuration change events are correlated. For more information, see Configure General Configuration.

For more information about event messages, see Network Configuration Manager Events.

Configure notification trap settings for a device family.

Follow these steps:

1. Navigate to device family configurations as described in Access Network Configuration Manager Device Family Configurations and expand the Configuration Notification Trap Settings subview.
   The configuration notification trap settings for the selected device family appear. This subview is configured with the basic formats of the traps that are received from the Syslog server. The following image shows the default settings for the Cisco IOS device family:



- To add a Syslog format, take the following steps:
  a. Click Add below the Syslog Format box.
     The Add dialog opens.

b. Enter the format of the trap message with any column-specific information that can be parsed out from the message in {}, and click OK.

The new Syslog format is added to the box.

- To add an entry to the Change Event Source Table, take the following steps:
  a. Click Add below the Change Event Source Table box.
  The Add dialog opens.
  b. Enter a Source index entry, and click OK.
  The new entry is added to the table.

Configure a Single Device to Override Device Family Settings

This section describes how to configure a single device to override the configuration of its associated device family. Most device family settings can be overridden at the local device level

Access Network Configuration Manager Settings on a Single Device

Network Configuration Manager settings for a single device are available in the Network Configuration Manager subview.

Follow these steps:

1. Select a device in the Explorer tab.
   Information and configurations appear in the Information tab of the Contents panel.
2. Scroll down the page and expand the Network Configuration Manager subview.
   The Network Configuration Manager device configuration options appear.
   The settings that you select here override the device family settings.

Enable or Disable Network Configuration Manager on a Single Device

All Network Configuration Manager operations can be disabled at the local device. Network Configuration Manager must be enabled on the associated device family for this setting to affect a device. For more information, see Configure Device Family General Configuration.

Follow these steps:

1. Expand the Network Configuration Manager subview as described in Access Network Configuration Manager Settings on a Single Device.
   The Network Configuration Manager device configuration options display.
2. Click set next to Configuration Manager to enable or disable Network Configuration Manager tasks and Network Configuration Manager functionality.

   NOTE
   Configuration Manager is enabled by default.

   The current state of communication with Network Configuration Manager displays next to Configuration Manager.

Configure Unsolicited Device Configuration Captures on a Single Device

Unsolicited Device Configuration Captures can be enabled or disabled at the local device.

NOTE
Unsolicited Device Configuration Captures must be enabled globally for this local setting to have an effect.

For more information, see Unsolicited Notifications of Device Configuration Changes.

Follow these steps:

1. Expand the Network Configuration Manager subview as described in Access Network Configuration Manager Settings on a Single Device.
   The Network Configuration Manager device configuration options appear.
2. Click set next to Unsolicited Device Configuration Captures to enable or disable automatic device captures.
   The value displays next to Unsolicited Device Configuration Captures.

Specify Configuration Change Alert Settings on a Single Device

Configuration Change Alert settings can be enabled or disabled at the local device.

Follow these steps:

1. Expand the Network Configuration Manager subview as described in Access Network Configuration Manager Settings on a Single Device.
   The Network Configuration Manager device configuration options appear.
2. Expand the Local Configuration Change Alert subview.
   The Local Configuration Change Alert options appear.
3. Click set next to 'Use Local Configuration Change Alert Settings' to override the global settings.
   The value appears next to 'Use Local Configuration Change Alert Settings'.
4. Modify the following Configuration Change Alert options as needed:
   - **Alert Mode**
     Lets you specify the events that trigger an alarm.
     - **Alarm On Any Changes**
       Triggers an alarm for configuration changes only.
     - **Alarm On Any Reference Violations**
       Triggers an alarm for reference configuration violations only.
     - **Alarm On Any Reference Violations or Changes**
       Triggers an alarm for both reference configuration violations and configuration changes.
     - **No Alarm**
       Ensures that no alarms are triggered for any configuration changes.

     Default: No Alarm
   - **Reference Violation Alert Type**
     Specifies the type of alarm or event that is asserted when a reference configuration violation occurs. The existing comparison mask is used to determine significant differences between current and reference configurations. Reference violation alarms are automatically cleared when the current configuration matches the reference configuration.
     See Specify a Reference Configuration for information on setting a reference configuration.

Valid values are critical, major, and minor alarms and events only.

Default: Event Only

- **Configuration Change Alert Type**
  Specifies the only type of alarm or event that is asserted when any configuration change occurs.

Valid values are critical, major, and minor alarms and events only.

Default: Event Only

Configure Communication Mode on a Single Device

All out-of-the-box supported devices have a communication mode that determines how Network Configuration Manager interacts with the device. Some out-of-box supported devices let you select from among multiple communication modes.

Depending on the selected communication mode, the device user name, password and enable password may be required.

For more information about configuring the communication mode for a device family, see Configure Device Family Communication Mode.

The following image is an example of the Local Communication Configuration subview:

Configure the communication mode on a single device.

Follow these steps:

1. Expand the Network Configuration Manager subview as described in Access Network Configuration Manager Settings on a Single Device.
   The Network Configuration Manager device configuration options appear.
2. Expand the Local Communication Configuration subview.
   The Local Communication Configuration options appear. The available options depend on the device type.
3. Modify the Communication Configuration options as needed:
   - **Use Local Communication Mode Settings**
     Specifies whether to override the device family communication mode with the Local Default Communication Mode.

     NOTE
     The Local Default Communication Mode is not used for an operation if a script is configured for that operation on the device family.
   - **Local Default Communication Mode**
     Specifies the communication mode if your device lets you choose from multiple communication modes.
   - **Use Local Authentication Settings**
     Specifies whether to override the device family authentication settings. When enabled, the values that are specified in the Local Username and Password fields are used.
   - **Local Username**
     Specifies the user name for accessing the device.
   - **Local Password**
     Specifies the password for accessing the device.
   - **Local Enable Password**
     Specifies the second password for configuring the device (supported for Cisco IOS, Cisco IOS-SSH Capable, and Foundry devices only).

The local communication configuration options for the selected device are set.

Configure a Mask on a Single Device

Configure masks to exclude script content from configuration comparisons or to hide sensitive information from unauthorized users. Masks configured at the local device level override the mask settings of the device's associated device family. Mask configurations on a single device are discussed in the following sections.

Comparison Masks

The Comparison Mask is a list of regular expressions that conceal device configuration content during comparison with a historical configuration. Any line in the device configuration file that matches a regular expression in the Comparison Mask is ignored during configuration file comparisons. Network Configuration Manager provides a list of predefined masks that are displayed in the window next to Comparison Mask. Masks on local devices override the mask settings of the device family.

View Masks

The View Mask is a list of regular expressions that conceal device configuration content from users who lack the OneClick privilege to view the entire device configuration file. Content in the View Mask field is only accessible to operators with the View Unmasked Configurations privilege. Use the mask to hide passwords or other content from unauthorized users. Masks on local devices override the mask settings of the device family.

For more information about the View Unmasked Configuration privilege, see Network Configuration Manager Privileges.

Enter a Mask on a Single Device

You can enter a mask for a single device.

Follow these steps:

1. Expand the Network Configuration Manager subview as described in Access Network Configuration Manager Settings on a Single Device.
   The Network Configuration Manager device configuration options display.
2. Expand the Local Mask Configuration subview.
   The local comparison and view masks options display.
3. Click set next to Use Local Comparison Mask or Use Local View Mask to override the device family settings.
   The value displays next to the option.
4. Select Add under Local Comparison Mask or Local View Mask.
   The Add dialog opens.
5. Enter the mask for the selected device. For example, for comment lines, enter: [!#] or enter any regular expression.
6. Click OK to accept your entries.
   The content that you entered for the mask displays. You have now set a mask for the selected device.
7. Repeat Step 4 to Step 6 to enter more masks.

Network Configuration Manager Extension Utility

The Network Configuration Manager Extension Utility lets you extend the basic functionality of Network Configuration Manager. You can create device families and can manage extra devices and vendors by using Perl scripts for the operations that Network Configuration Manager executes on a device. You can customize trap settings and can use them to correlate configuration change event information.

The following sections describe how to use the Extension Utility to expand Network Configuration Manager support.

Supported Operations

The Network Configuration Manager Extension Utility lets you use Perl scripts to extend Network Configuration Manager to extra devices and vendors. Network Configuration Manager can be extended by providing Perl scripts for any, or all, of the operations Network Configuration Manager performs on a device. The following list summarizes these operations:

- **Capture Startup Configuration**
  Capture device startup configuration.
- **Capture Running Configuration**
  Capture device running configuration.
- **Upload Running Configuration**
  Upload and merge specified content into the device running configuration.
- **Write Startup Configuration**
  Write the device current running configuration to its startup configuration.
- **Reload Device**
  Reboot a device.
- **Cancel Reload**
  Cancel the scheduled reboot of a device.
- **Load Device Firmware**
  Initiate a load of the specified firmware image on the device.

  NOTE
  In 10.4.2, you can use Perl scripts to execute these NCM operations on SDC-modeled devices also.

Scripts can be configured for each of these operations within device families that are created on demand. Any operation that lacks a script is handled as an unsupported operation for the given Device Family and all devices that are contained within it.

The Cisco PIX OS out-of-box device family provides an example of how scripts are used to extend Network Configuration Manager support. (Within these example scripts, the Net::Telnet perl module does *not* support IPv6.)

The utility also lets you use Perl scripts to alter Network Configuration Manager interactions with devices that belong to out-of-the-box device families.

OpenSSH Device Family

The OpenSSH Device Family supports the Network Configuration Management functionality, which you can use to capture and upload device configurations. This OpenSSH functionality includes the Perl scripts to support OpenSSH.

Follow these steps to add any OpenSSH supported devices to this device family:

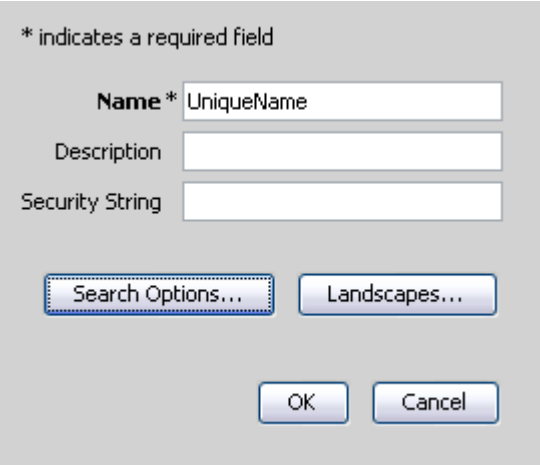
1. Select the device that you want to add to the OpenSSH Device Family
2. Right click the device, then select **Add to**, **Device Family**.
3. Select the NCM OpenSSH Device Family.

Create a Custom Device Family

Network Configuration Manager supports Cisco, Enterasys, Enterasys/Riverstone SSR, Extreme, Foundry, Juniper, Lancom, Nortel Baystack, and Nortel Passport device families out-of-box. The Network Configuration Manager Extension Utility lets you create custom device families.

Follow these steps:

1. Expand Configuration Manager in the Explorer tab of the Navigation panel.
2. Right-click Device Families, and select Create Device Family.
   The Create Device Family dialog opens as shown in the following image:



3. Enter a unique name in the Name field.
4. (Optional) Enter a description and security string.
5. (Optional) Click the Landscapes button to select the Landscapes where you want to place the device family.
6. Select the Search Options button to search for specific devices.
   The Search Options dialog opens. Like a Global Collection, a device family can have both static members that are manually added to the family, and dynamic members that are automatically added using specified search criteria. For more information, see the OneClick Administration section.

   NOTE
   A device can belong to only one device family. If multiple device families contain search criteria that apply to the same device, the first device family to execute the search contains the device.
7. Select OK when you have finished.
   The device family is created and appears under Device Families in the Explorer tab of the Navigation panel. Static members can now be added.

Place a Device in a Device Family

Network Configuration Manager automatically assigns out-of-the-box supported devices to the family. A device that is associated with a device family must be manually moved to a user-created device family. Manually created device families that contain search criteria to define membership do not pull in devices that already belong to a device family. For the search criteria to pull in a new device, the device must not currently be a member of any device family.

You have several options for placing a device in a device family. You can manually make the association.

Follow these steps:

1. Locate the device.
2. Right-click the device and select Add To, Device Family.
   The Select Device Family dialog opens.
3. Select the device family that you want to associate with the selected device.
   If a suitable device family is not displayed, create a custom device family by clicking Create. See Create a Custom Device Family for more information.
   The device is now associated with the selected device family.

   NOTE
   Add `cisco_ssh_check=9K` in the `<SPECROOT>/SS/.vnrmc` file and restart the SpectroSERVER, to place the Cisco 9300 series devices in the correct device family.

You can force a manually created device family to update using its defined search criteria.

Follow these steps:

1. Right-click the device family in the Navigation panel.
2. Select Update Device Family.
   The device family searches for and adds all devices that meet the search criteria if they do not currently belong to a device family.

For more information about device family search criteria, see Create a Custom Device Family.

You can also restore a device to one of the out-of-box supported device families.

Follow these steps:

1. Right-click a device that is not currently associated with a device family.
2. Select Reconfiguration, Reevaluate NCM Device Family.

   WARNING
   Cisco PIX devices do not support the Reevaluate NCM Device Family function.

   Network Configuration Manager reevaluates the device to determine whether it should belong to an out-of-the-box device family.
   If Network Configuration Manager determines that the placement is appropriate, the device is added to the device family.

   NOTE
   The Reevaluate NCM Device Family action on a device that is currently in a manually created device family has no effect.

   NOTE
   For more information about out-of-the-box supported device families, see Supported Devices.

Extension Utility Script Configuration

Perform all interactions with Network Configuration Manager scripts using OneClick. Network Configuration Manager handles all script administration within the DX NetOps Spectrum environment. The available scripting options are discussed in the following sections.

NOTE
The end of line character in each line must be in UNIX format.

Scripting Considerations

When a script is configured for a Network Configuration Manager operation, the script is used for all devices in the family. For example, if scripts are configured for all of the supported operations in the Cisco IOS SSH Capable device family, the Communication Mode setting at the device family and any overridden Communication Mode settings at local devices have no effect. In this example, the scripts for all Network Configuration Manager operations on all devices that are contained in the Cisco IOS SSH Capable Device Family are used.

In the case where only a subset of the Network Configuration Manager operations has scripts that are configured, the Network Configuration Manager uses the Communication Mode that is selected at the device family or overridden at the local device for the operations for which no script is configured.

Username, Password, and Enable Password are always sent to the scripts as command line parameters. The values that are specified in the device family are used unless they are overridden at the local device in which case the locally overridden values are used.

Default Script Command Line Parameters

By default Network Configuration Manager provides the following parameters, in the order shown, to every script. If the script does not make use of these parameters, the script must still be written to accept them.

- Device IP.
- Absolute filename of file containing content to upload. (Upload operation only).
- Device Username.
- Device Password.
- Device Enable Password.

Additional Script Command Line Parameters

Optionally, unlimited additional command line parameters can be configured for each of the supported operations. The parameters are passed on the command line to the script after the default set of parameters. The parameters are passed in the order they are shown in the Additional Script Parameters list.

Upload Running Configuration and Load Device Firmware operations can have additional command line parameters that are configured in such a way that the user is prompted for a value at runtime. A label and default value can also be displayed when prompting at runtime.

Error Code Mappings

Network Configuration Manager provides the ability to map non-zero integer values that are returned by the script to a textual error message which displays in OneClick if the error occurs. This enables script creators to provide detailed information about the failure mode.

Script Error Handling

For Network Configuration Manager to report success of a script-based operation, the script must return a value of zero. Network Configuration Manager assumes that the operation failed if a non-zero value is returned by the script.

Additional Error Detail Returned in STDERR Buffer

If a script returns a non-zero value, in addition to the error mapping above, Network Configuration Manager also looks for any output that is returned by the script in the STDERR buffer. If content is found, it displays in OneClick as additional error information.

Enter a Configuration Script

Network Configuration Manager can use Perl scripts for the following operations:

- **Capture Startup Configuration**
  This script must return the device startup configuration in the STDOUT buffer. All content that is returned in the buffer is considered to be the device startup configuration.
- **Capture Running Configuration**
  This script must return the device running configuration in the STDOUT buffer. All content that is returned in the buffer is considered to be the device running configuration.
- **Upload Running Configuration**
  This script reads the file that is identified by the Absolute Filename parameter (for more information, see Default Script Command Line Parameters). It then uploads and merges the content of the file to the device running configuration.
- **Write Startup Configuration**
  This script causes the device running configuration to be written to its startup configuration.
- **Reload Device**

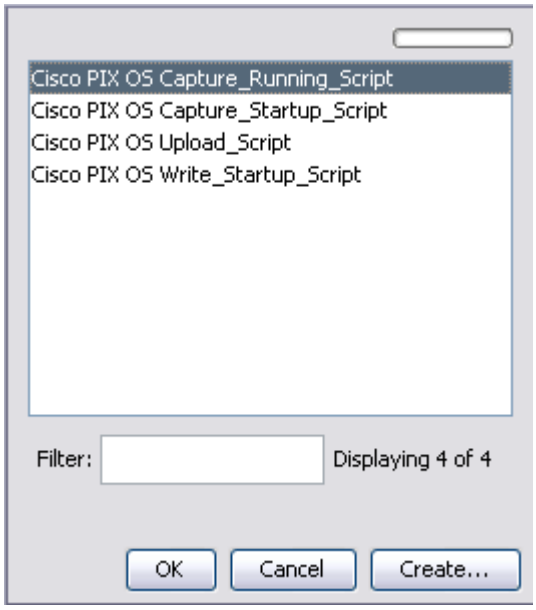
This script reboots a device.

- **Cancel Reload**
  This script cancels a pending or scheduled reboot of a device.
- **Load Device Firmware Configuration**
  This script uploads a new firmware image onto a device and executes all necessary operations to reload the device using this firmware image.

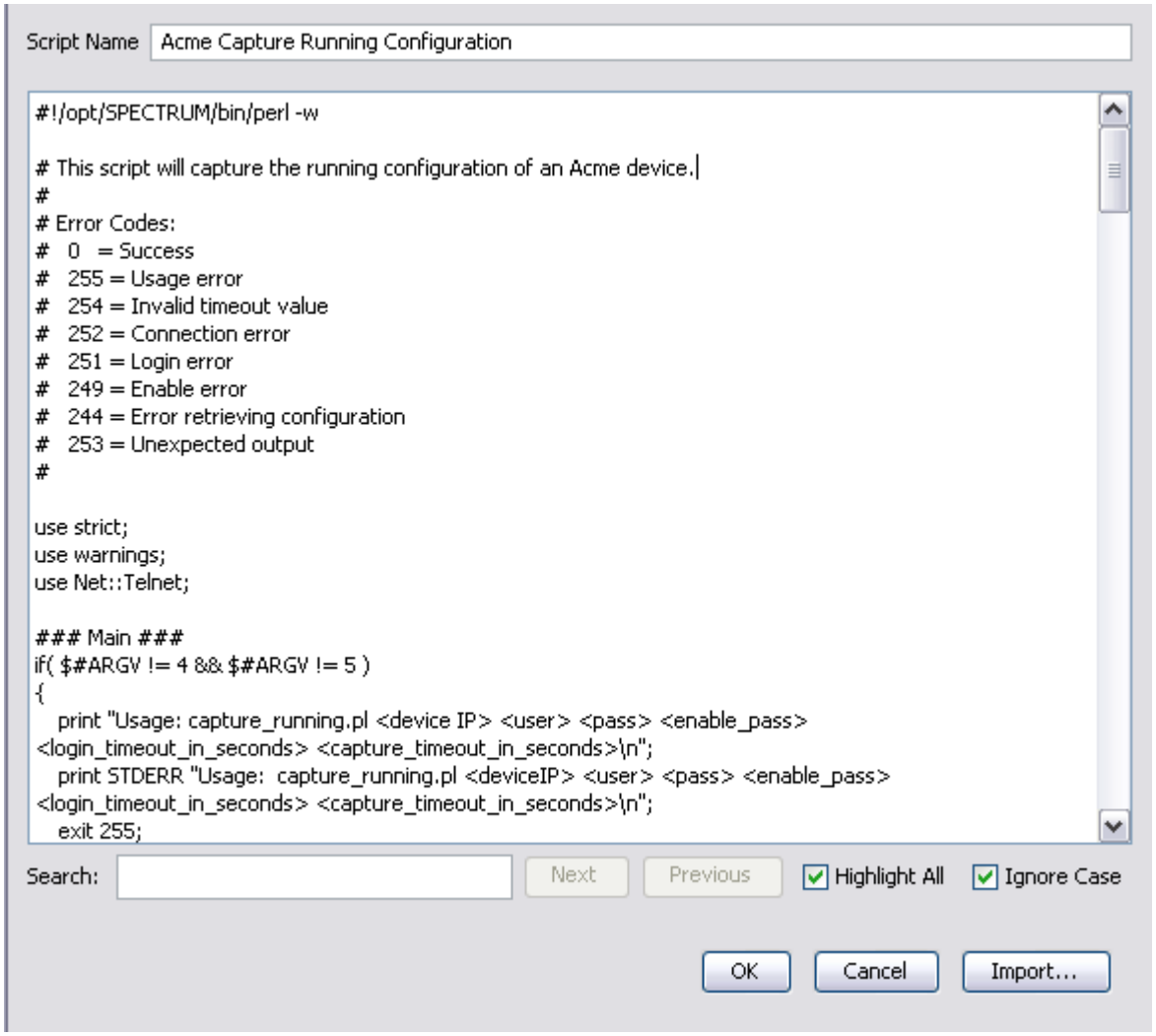
You can select a configuration script for these operations.

Follow these steps:

1. Select a device family from the Explorer tab.
   Information and configurations display in the Information tab of the Contents panel.
2. Expand the Device Configuration Transfer Settings subview.
   The script operation subviews appear.
   Note: For the Cisco IOS and Cisco IOS - SSH Capable device families, the Load Device Firmware Script resides in the Device Firmware Transfer Settings subview.
3. Expand the appropriate script operation subview.
   The available script configuration fields display.
4. Select set next to the script name.
   The Select Script dialog opens as shown in the following example:

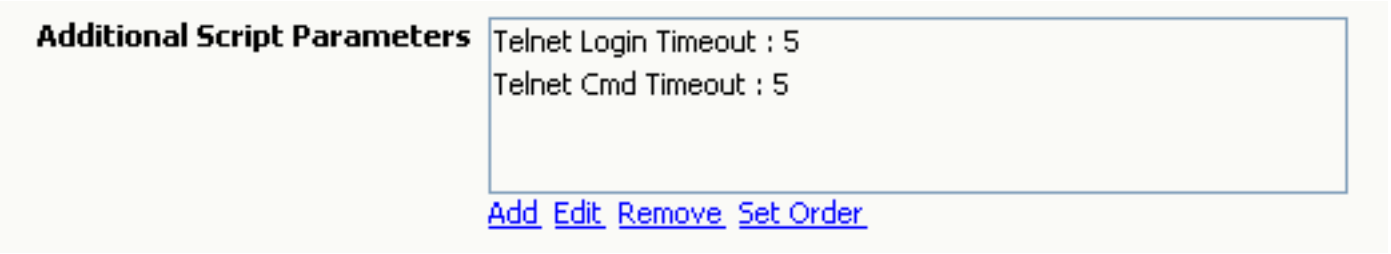


5. Take one of the following steps:
   – If a script that you want to use is available, select the script, select OK, and go to Step 10.
   – If you have not yet created a script for the selected device family, select Create to upload or create one.
6. Supply a unique name for the script in the Script Name field. Paste the script into the field under Script Name, or select Import to import a configuration file that is saved locally on your system.
   The script content appears in the field under the Script Name field, as shown in the following example:

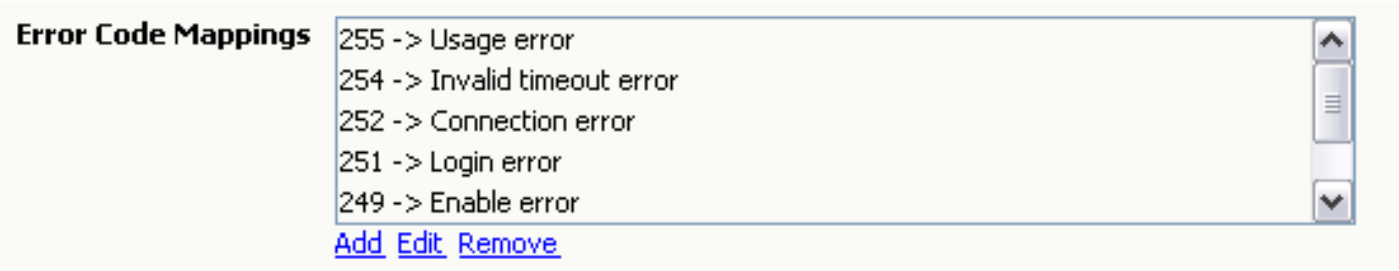


7. (Optional) Edit script content if necessary. Or enter criteria in the Search field to locate specific lines in the script file.
8. Select OK when you have finished importing and configuring the script.
   The script name appears in the Select Script dialog.
9. Select the script, and select OK.
   The script is loaded and is visible in the Script Content field.
10. Add any additional script parameters.

    NOTE
    For more information, see Additional Script Command Line Parameters.
11. Select Add under the Additional Script Parameters field.
    The Add dialog opens.
    a. Enter the parameter name and value. If the operation is Upload, or the task is Load Firmware, you can configure the parameter to prompt you at run time for a value.
    b. Select OK.
       The parameter appears in the Additional Script Parameters field, as shown in the following example:



12. Add any error code mappings. For more information, see Error Code Mappings.
13. Select Add under the Error Code Mappings field.
    The Add dialog opens.
14. Enter the error code in the Error Code field and the corresponding message in the Error Message field, and select OK.
    The error code appears in the Error Code Mappings field, as shown in the following example:



The configuration script is ready to run.

Configuration for Non-Root User on RHEL 8

For a non-root user on RHEL 8, to execute NCM script operations, you must create a configuration with the required entries (as shown in the example) under `/usr/userid/.ssh/` with the appropriate permissions (`chmod 600 ~/.ssh/config`).

Example

```
cat /usr/home/myuser/.ssh/config

     Host *

        StrictHostKeyChecking no

        UserKnownHostsFile=/dev/null

        KexAlgorithms +diffie-hellman-group1-sha1

        Ciphers aes128-ctr,aes192-ctr,aes256-ctr,aes128-cbc,3des-cbc,aes128-cbc,3des-
cbc,aes192-cbc,aes256-cbc
```

NOTE
This behavior is also applicable for non-root SpectroSERVER and non-root SDC.

NCM Support for SSH using Jsch libraries

This release of DX NetOps Spectrum NCM supports SSH using Jsch libraries to perform NCM Captures and uploads for CISCO and Juniper devices only using SSH configuration.

Follow these steps to enable NCM SSH to use JSCH mode:

1. Goto $SPECROOT\NCM\config.xml
2. Set the 'ssh-library type' value to 'jsch' (the default value is 'mindterm')
   <ssh-library type="java.lang.String">mindterm</ssh-library>
   to
   <ssh-library type="java.lang.String">jsch</ssh-library>

For large configuration devices (>8K lines), Jsch sometimes throws the capture failure/capture file does not work error. To fix this, increase the output wait time.

Follow these steps:

1. Goto $SPECROOT\NCM\config.xml
2. Change the jsch-read-datawait type value:
   <jsch-read-datawait type="java.lang.Integer">4</jsch-read-datawait>
   to
   <jsch-read-datawait type="java.lang.Integer">7</jsch-read-datawait>

> **NOTE**
>
> **Information!** DX NetOps Spectrum by default uses MindTerm SSH libraries for NCM activities. In some cases, MindTerm libraries throw premature EOF error, then the user should shift to other libraries such as JSCH or OpenSSH.
> For Windows platform, user can shift to only JSCH libraries. For Linux platforms user can shift to either JSCH libraries or Perl Script using the OpenSSH.
>
> In this release of DX NetOps Spectrum, NCM supports JSCH for CISCO and Juniper device family only. Once the configuration is set to leverage JSCH libraries, all the NCM activities use JSCH SSH API instead of default SSH API (mindterm).

Perl Modules

DX NetOps Spectrum ships all Perl Modules (for the Windows platforms) required to run the Perl scripts that are provided out-of-box. That includes:

- Net::Telnet

In addition, DX NetOps Spectrum also comes with certain perl modules that may be useful in developing scripts for the Extension Utility. These include:

- Net::OpenSSH
- Net::SSH
- Net::SSH::Expect
- Expect
- Net::TFTP
- Net::SCP
- Net::FTP

Perl modules that ship with DX NetOps Spectrum can be viewed at:

```
/opt/SPECTRUM/lib/perl5
```

WARNING
Perl modules that are not compiled and installed correctly may result in failure or other undesirable behavior.

RESTRICTION
NCM OpenSSH is not supported in the Windows platform.

Using SSH-based Perl Scripts for Network Configuration Manager Operations

DX NetOps Spectrum's out-of-the-box scripts-based support for Network Configuration Manager operations is based on the Net::Telnet module. If you want to use SSH-based scripts for Network Configuration Manager operations:

- **Windows**-- DX NetOps Spectrum includes a complete perl install and the Net::SSH::Expect module.
- **Linux** -- You must install perl to a separate location on your system and must configure DX NetOps Spectrum to use that perl.

This Perl installation and configuration on DX NetOps Spectrum is to be done on a per landscape basis in a DSS. You will have to set up perl for each landscape on which you have devices that are modeled to use SSH-based scripts.

NOTE
If you want to continue to use DX NetOps Spectrum's out-of-box scripts once you have configured DX NetOps Spectrum to use your custom Perl install, then your custom Perl area must have the Net::Telnet perl module installed. You can download and install this module from www.cpan.org. Otherwise, DX NetOps Spectrum's out-of-box scripts fail.

To set up SSH-based scripts, follow the instructions specific to your platform.

On Windows

1. Install Perl.
   DX NetOps Spectrum ships with Cygwin's complete version of Perl, so you are not required to install anything more if you want to use scripts based on the Net::SSH::Expect module.
   If you want to use scripts based on some other module, complete one of the following depending on the module you are using:
   - If the perl module is compatible with a version of Perl other than Cygwin, then we recommend that you install that specific Perl onto your SpectroSERVER machine, then install your specific perl module, and then configure DX NetOps Spectrum to use your particular Perl install. (See Configuring DX NetOps Spectrum to Use a Custom Perl Install.)
   - If the perl module that you want to install is only compatible with Cygwin's perl, and is a change module (i.e., it does not require compilation of C libraries), then you can add it to the DX NetOps Spectrum Perl install. Simply place the <Module_Name>.pm file in $SPECROOT/NT-Tools/SRE/lib/perl5/site_perl/5.8
   - If the perl module that you want to install is only compatible with Cygwin's perl and also requires C libraries to be compiled, then this module has to be compiled and shipped with DX NetOps Spectrum. Contact DX NetOps Spectrum support for this enhancement request.
2. Install SSH-based perl modules and SSH program.
   DX NetOps Spectrum ships with the Net::SSH::Expect (and its required) modules and the ssh program (that is required by Net::SSH::Expect). For instructions on how to develop scripts using this module, select the documentation for Net::SSH::Expect on www.cpan.org.
3. Configure DX NetOps Spectrum to use the custom Perl install.
   Because DX NetOps Spectrum's perl is set up for this purpose, you do not have to configure DX NetOps Spectrum to use a custom perl installation.

On Linux

1. Install Perl.
   The OS already has Perl that is installed (check /usr/bin/). This pre-installed Perl can be used for Network Configuration Manager scripts.
2. Install SSH-based perl modules and the SSH program.

You need to download and install the Net::SSH::Expect module, its dependent modules, and the ssh utility.
The dependency tree for Net::SSH::Expect looks like:
Net::SSH::Expect -> Expect -> IO::Pty
where the '->' represents a "requires" relationship.
You can download all of these modules from www.cpan.org
Be sure to install these modules to the custom Perl area you installed above.
This can be done by specifying the full-path of the perl binary when you are installing the modules such as:

```
<PERL_FULL_PATH>/perl Makefile.pl
```

NOTE
Some Perl modules are dependent on C/C++ code libraries. To install such modules, you have to install the gcc compiler so that you can link against the libraries. You can install the gcc compiler by using RPM to add the latest gcc package.

3. Install SSH-based perl modules and the SSH program.
   You need to download and install the Net::OpenSSH module, its dependent modules, and the ssh utility.
   The dependency tree for Net::OpenSSH looks like:
   Net::OpenSSH -> IO::Pty -> File::Path
   where the '->' represents a "requires" relationship.
   You can download all of these modules from www.cpan.org
   Be sure to install these modules to the custom Perl area you installed above.
   This can be done by specifying the full-path of the perl binary when you are installing the modules such as

   ```
   <PERL_FULL_PATH>/perl Makefile.pl
   ```

   NOTE
   Some Perl modules are dependent on C/C++ code libraries. To install such modules, you have to install the gcc compiler so that you can link against the libraries. You can install the gcc compiler by using RPM to add the latest gcc package.
4. Configure DX NetOps Spectrum to use the custom Perl install.
   See Configuring DX NetOps Spectrum to Use a Custom Perl Install and point DX NetOps Spectrum to the perl install area from Step 1 above.

Configuring DX NetOps Spectrum to Use a Custom Perl Install

DX NetOps Spectrum is configured by default to use the Perl that is shipped with it. If you want to use additional perl modules that are not shipped with DX NetOps Spectrum, and have installed them to a perl area, you can configure DX NetOps Spectrum to use your custom perl installation. To set this up, select the Configuration Manager model in the Explorer tab in OneClick. In its Information view, expand the Perl Configuration subview. You find a table that contains the Perl directory configuration per landscape.

Note that the Use Custom Perl option has to be set to Enabled to be able to specify a custom perl directory. Otherwise, the default Perl that comes with DX NetOps Spectrum is used. You can point DX NetOps Spectrum to a custom Perl location that you have installed on a particular SpectroSERVER system.

Follow these steps:

1. On a given landscape, set the Use Custom Perl to Enabled.
2. Once you have enabled the use of a custom Perl area, specify the Custom Perl Directory.

   NOTE
   The Custom Perl directory must contain the full pathname of the directory that contains the perl.exe (Windows) or the perl program (Linux).

For example, if the perl program is located in /usr/local/bin/, specify Custom Perl Directory as /usr/local/bin.

NOTE
You can continue to use the DX NetOps Spectrum default scripts once you have configured DX NetOps Spectrum to use your custom Perl install. But your custom Perl area must have the Net::Telnet perl module installed. You can download and install this module from www.cpan.org. Otherwise, the DX NetOps Spectrum default scripts fail.

You can also disable the use of your custom Perl area and use the default DX NetOps Spectrum Perl.

NOTE
Set the Use Custom Perl to disabled (use CA Spectrum default). Note that when you disable the use of a custom Perl area, the Custom Perl Directory cannot be seen or edited. But when you enable Use Custom Perl again, your previously specified Custom Perl Directory is restored.

Using Additional Perl Modules

If you want to use scripts based on your preferred perl module, you must install the perl module to the area that will be used.

On Windows

Depending on the module that you want to use you have three options:

- If the perl module is compatible with a version of Perl other than Cygwin, then we recommend that you install that specific Perl onto your SpectroSERVER machine, then install your specific perl module and then configure DX NetOps Spectrum to use your particular Perl install. (See Configuring DX NetOps Spectrum to Use a Custom Perl Install).
- If the perl module that you want to install is only compatible with Cygwin's perl, and is a text-based module (i.e., it does not require compilation of C libraries), then you can add it to the DX NetOps Spectrum Perl install. Simply place the <*Module_Name*>.pm file in $SPECROOT/NT-Tools/SRE/lib/perl5/site_perl/5.8
- If the perl module that you want to install is only compatible with Cygwin's perl and also requires C libraries to be compiled, then this module has to be compiled and shipped with DX NetOps Spectrum. Contact DX NetOps Spectrum support for this enhancement request.

On Linux

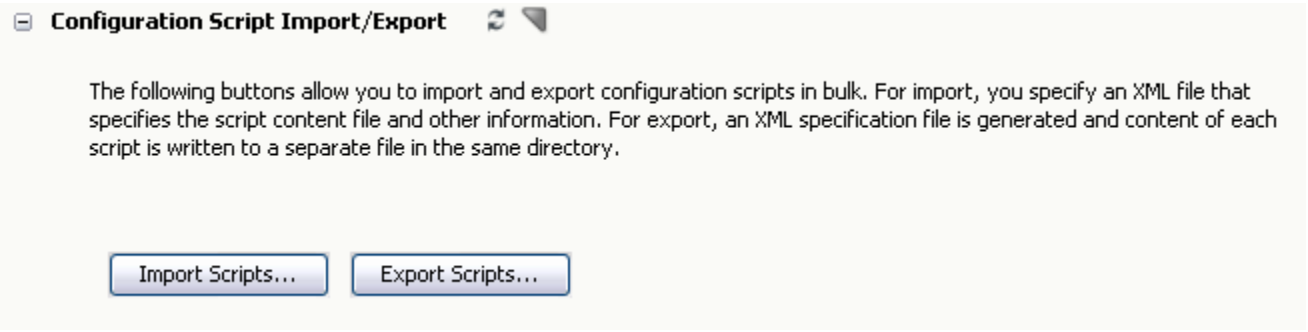
It is required that you install Perl to a separate area on your SpectroSERVER, then install the required perl modules using that Perl and configure DX NetOps Spectrum to use the Perl install area. Once you have installed Perl, refer to the installation instructions of the specific perl module that you want to install. Then refer to Configuring DX NetOps Spectrum to Use a Custom Perl Install. You may refer to the Using SSH-based Perl Scripts for Network Configuration Manager Operations for details about how to use scripts based on the Net::SSH::Expect module but you may use the procedure as a guideline for integrating any perl module.

Import and Export Scripts

Network Configuration Manager lets you import and export scripts in bulk. The scripts are exported to or imported from the file system of the host server that is running the OneClick client.

Follow these steps to export scripts:

1. Select Configuration Manager in the Navigation panel.
2. Select the Information tab in the Contents panel.
   Information and configurations display.
3. Expand the Configuration Script Import/Export subview.
   The Import Scripts and Export Scripts buttons display.



1. Select Export Scripts.
   The Select Scripts To Export dialog opens.
2. Select the script to be exported. Or select multiple scripts.
   The Save as dialog opens.
3. Select the location to save, and supply a name for the XML specification file that is automatically generated during export. The export process generates a file for each selected Perl script using its designated name and the .pl extension. The export process also generates the XML specification file that contains the list of scripts that were exported and the error-mapping information for each. The XML specification file can then be used to import the scripts on the same or different DX NetOps Spectrum environment.
   The selected Perl scripts are exported to the location that you specified.

Follow these steps to import scripts:

1. Select Configuration Manager in the Navigation panel.
2. Select the Information tab in the Contents panel.
   Information and configurations display.
3. Expand the Configuration Script Import/Export subview.
   The Import Scripts and Export Scripts buttons display.
4. Select Import Scripts.
   The open dialog opens.
5. Select the XML specification file that describes the Perl scripts to be imported into Network Configuration Manager. If you are importing scripts that were previously exported from DX NetOps Spectrum, you can use the XML specification file that was generated during that export.
   The XML specification file may also be generated manually by following the format that is shown in the following example.

```
<scripts>
  <script>
    <file-name>ABC_Vendor_Capture_Running_Configuration.pl</file-name>
    <display-name>ABC Vendor Capture Running Configuration</display-name>
    <error-message errorCode="255">Usage</error-message>
    <error-message errorCode="99">Invalid Enable Password</error-message>
    <error-message errorCode="98">Unexpected Response</error-message>
    <error-message errorCode="97">Illegal Telnet Timeout Value</error-message>
  </script>
  <script>
    <file-name>XYZ_Vendor_Capture_Running_Configuration.pl</file-name>
    <display-name>XYZ Vendor Capture Running Configuration</display-name>
    <error-message errorCode="255">Usage</error-message>
    <error-message errorCode="99">Response Timed out</error-message>
    <error-message errorCode="50">Connection Error</error-message>
  </script>
  <script>
```

```
    <file-name>XYZ_Vendor_Capture_Startup_Configuration.pl</file-name>
    <display-name>XYZ Vendor Capture Startup Configuration</display-name>
  </script>
</scripts>
```

- **file-name**
  The name of the Perl file to be imported. This file must exist in the same directory as the XML specification file at the time of import.
- **display-name**
  The name that is used in OneClick to identify this script.
- **error-message**
  (Optional) Describes a mapping of an error code that is returned by the script to a textual error message that is displayed in OneClick if the error occurs. Multiple error-message elements can be specified for each script.

The Perl script XML is imported. The scripts are available when configuring Network Configuration Manager operations on device families.

> **NOTE**
> The import process does not associate the scripts with a device family.

Maintaining a Script Backup and History

Scripts are stored as models in the DX NetOps Spectrum database and therefore are backed up each time DX NetOps Spectrum performs a backup. The script export feature provides an additional backup and means of tracking the history of Network Configuration Manager scripts that is easily accessed and imported back into DX NetOps Spectrum if desired.

Customized Traps

You can extend the functionality of Network Configuration Manager by configuring customized trap settings for your installation. These settings are used to correlate configuration change event information so that events are combined for a particular device. These settings are configured at the device family level. For more information, see Configure Notification Trap Settings.

Capture Huge Configurations

10.3.1 allows you to capture huge configuration files using the 'Capture' function of the Network Configuration Manager by configuring the 'thread-timeout-value' settings. To capture large configuration files, increase the thread-timeout-value as described here:

Follow these steps:

1. Navigate to the **$SPECROOT>NCM>config.xml** file and find the 'thread-timeout-value type'.
2. By default this value is set to 300 as shown here:
   ```
   <thread-timeout-value type="java.lang.Integer" ><300>< / thread-timeout-value>
   ```
3. Change the setting and increase the value as shown here:
   ```
   <thread-timeout-value type="java.lang.Integer" ><increased value in seconds>< /
    thread-timeout-value>
   ```

NCM Enhancement - Configuration Search

10.3.2 introduces 'Configuration Search' a new search filter for NCM devices, which is based on parsing a sub-string value to span all device configurations from a device family and/or from a global search. This enhancement address challenges for regular search operations for configurations, route, interface description, or other global parameters such as Log settings, AAA, SNMP, ACL, and others.

The **Configuration Search** feature includes two major search categories and subcategories, as described:

1. **Search Device Models by Sub-string Search In**: This search criteria contacts SpectroSERVER to fetch device model information. This search criteria fetches configurations including input string in the running configuration, startup configuration, running configuration difference result and the startup configuration difference result.
   a. Startup Configuration: Devices that are configured and captured using the startup command can be identified using a sub-string under this category.
   b. Running Configuration: Devices that are configured and captured using the running command can be identified using a sub-string under this category.
   c. Startup Configuration Difference Result: Users can utilize this search option to find the difference in the configuration of the same device that is configured and captured under the startup configuration.
   d. Running Configuration Difference Result: Users can utilize this search option to find the difference in the configuration of the same device that is configured and captured under the startup configuration.
2. **Search Host Configuration Models by Sub-string Search In**: This search criteria contacts SpectroSERVER to fetch the host configuration models containing input string in its configuration. This search criteria fetches configurations including input string in the running configuration, startup configuration, running configuration difference result and the startup configuration difference result.
   a. Startup Configuration: Host configuration models configured and captured using the startup command can be identified using a sub-string under this category.
   b. Running Configuration: Host configuration models configured and captured using the running command can be identified using a sub-string under this category.
   c. Startup Configuration Difference Result: Users can utilize this search option to find the difference in the host configuration of the same devices that are configured and captured under the startup configuration.
   d. Running Configuration Difference Result: Users can utilize this search option to find the difference in the host configuration of the same devices that are configured and captured under the startup configuration.

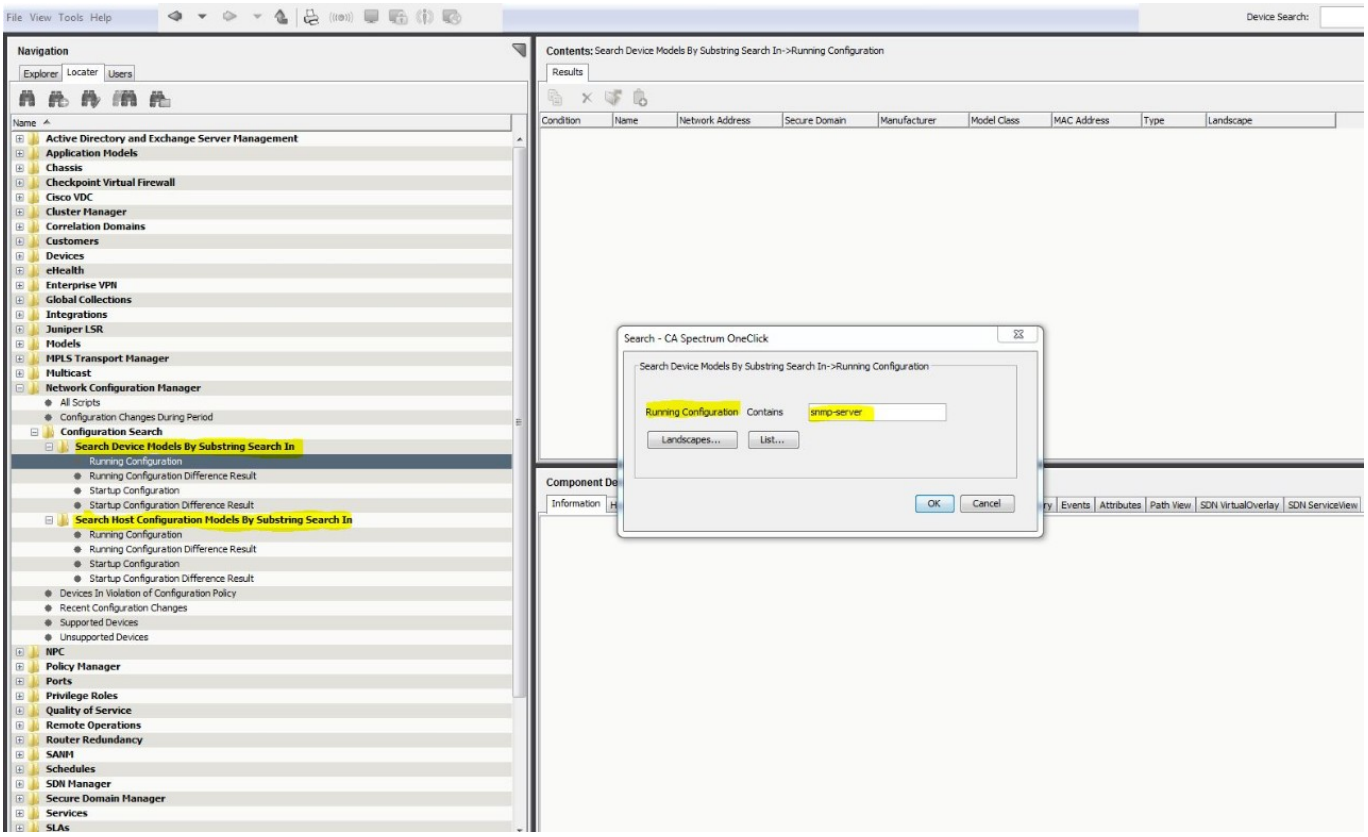
      NOTE

      Right-click on the 'Host Configuration' menu and select the 'Location' option to redirect to the corresponding device model.

      NOTE

      Until the Host Configuration model is added to any of one or more GCs or CorrelationDomains etc., it remains disabled.

This is a screenshot displaying the new Configuration Search feature with the two categories and other subcategories. For example the screenshot displays the Running Configuration window contains the snmp-server sub-string value in the device family.



Global Synchronization Task

This section describes how to set up the Global Synchronization task on your network with Network Configuration Manager. When you run the Global Synchronization task, Network Configuration Manager captures and saves all device configurations.

> **NOTE**
> We recommend that you capture device configurations prior to configuring Network Configuration Manager policies.

About Global Synchronization

A Global Synchronization task gathers running configurations for the devices on your network that have Network Configuration Manager enabled. You can schedule this task to run regularly. You select a time period and recurrence frequency to capture configurations from all network-wide supported devices. For example, capture device configurations after 9 PM and no later than 5 AM on a daily basis. By capturing the configurations for all devices on your network, you maintain a running configuration history.

You can set Global Synchronization to verify whether the startup configurations are the same as the running configurations. If they differ, you can configure Network Configuration Manager to generate an alarm. A Global Synchronization captures the startup configuration and compares it to the running configuration to detect changes. See Types of Configurations for descriptions of startup and running configurations.

NOTE
You can also gather running configurations for selected devices on your network and view the results in real-time by creating an automatic Sync task. See Create Sync Task for details.

About Enterasys/Riverstone SSR Devices

Configuration captures performed on Enterasys/Riverstone SSR devices provide the startup configuration and not the running configuration. Therefore, the device configuration history maintained by Network Configuration Manager is a history of the startup configurations on SSR devices. See Determine How an Enterasys/Riverstone SSR Device Responds to an Upload Task for details about how SSR devices handle Network Configuration Manager configuration uploads.

Configure Global Synchronization

Configure settings for Global Synchronization, such as a schedule. Global Synchronization is performed by the Global Sync Task, which appears under Tasks in the Explorer tab.

Follow these steps:

1. Select Configuration Manager in the Explorer tab.
   Information and configurations display in the Information tab of the Contents panel.
2. Expand the Global Synchronization subview.
   Global Synchronization options appear.
3. Modify the following options as needed:
   - **Global Synchronization Schedule**
     Specify a schedule for the Global Sync Task. Click the Schedule button to access the Select Schedule dialog from which you can select a default schedule or create a custom one. More information on scheduling the Global Sync Task is available in Schedule Global Synchronization.
   - **If Sync Task Not Completed in Allotted Time, Assert Task Alarm**
     Specify a minor, major, or critical alarm if you want to be notified by alarm if the global synchronization task did not complete properly. If you manage many devices and run a scheduled global sync task periodically, a Sync task stops at the end of the scheduled period because capturing configurations can consume bandwidth on slow links.
   - **Include Devices and Device Families on which NCM has been Disabled**
     Specify whether to include in the Failed Device List those devices for which Network Configuration Manager has been disabled.
     Default: Yes
   - **Verify Startup Equals Running Configuration**
     Enable this option if you want to compare a startup configuration against a currently running configuration for devices in your network.
   - **When Startup Differs, Assert Device Alarm**
     Specify a minor, major, or critical alarm if you want to generate an alarm for devices with startup configurations that differ from running configurations.

Schedule Global Synchronization

You can schedule a global synchronization to gather running configurations for all devices on your network. Devices are processed in random order.

If a scheduled global synchronization does not complete in the allotted time, the next execution first randomly processes the unprocessed devices from the previous execution. All remaining devices are then processed in random order.

WARNING
Do not schedule a “one-shot” global synchronization if DX NetOps Spectrum landscapes exist in multiple time zones. Performing this type of task causes the global synchronization to run only in the earliest time zone.

Follow these steps:

1. In Explorer, select Global Sync Task in the Tasks folder.
2. Select the List tab in the Contents panel.
3. Click the Schedule button in the toolbar.
   The Select Schedule dialog opens.
4. Take one of the following steps:
   – Select a default schedule and click OK.
   – Create a custom schedule. Click Create, specify schedule options, and click OK.
   The custom schedule is added to the list of available schedules. Select the new schedule and click OK.

   The Global Sync Task is now scheduled. The schedule appears in the Schedule column of the List table, and the schedule icon appears next to the task in the Tasks folder.

Run an On-Demand Global Sync Task

Run a Global Sync task to gather running configurations for all devices on your network. Devices are processed in random order.

Follow these steps:

1. In Explorer, select Global Sync Task in the Tasks folder.
2. Select the List tab in the Contents panel.
3. Click the Start Selected Task icon.
   The Sync Task Results dialog shows a list of processed devices and the results of the Global Sync.

View Configuration History for a Single Device

You can view the configuration history for a single device in OneClick. For details about uploading configurations to network devices, see Manually Upload Configurations to a Single Device.

Follow these steps:

1. Select a device in the Explorer tab.
2. Verify that the List tab is selected in the Contents panel, and select the Host Configuration tab in the Component Detail panel.
   The following details appear in the Host Configuration table:
   – **Capture Time**
   Lists the time (M-DD-YYY HH:MM:SS) that configuration in this row was first captured on the device.
   – **Captured By**
   Identifies the DX NetOps Spectrum OneClick user who configured the task.
   – **Line Changes**
   Lists the number of relevant changed lines when compared with the previous configuration on the device. Relevant changes include added lines, removed lines, and changed lines. Irrelevant changes are any lines that match the comparison mask. The comparison mask is managed in Mask Configuration settings for the device family. For more information, see Configure Device Family Masks.
   – **Total Line Changes**
   Lists the total number of changed lines (relevant and irrelevant) when compared with the previous configuration on the device. Displays the changes hyperlink if any changes are detected.
   – **Is Reference**
   Indicates the reference configuration for this device.
   – **Running vs. Startup**
   Shows configuration differences in the startup and running configuration files (if applicable). Displays the View Differences hyperlink if differences exist.
   – **Last Verified Time**

Lists the last time (M-DD-YYY HH:MM:SS) Network Configuration Manager verified that the configuration still existed on the device.

- **Last Verified User**
  Identifies the last user to have accessed the device.
- **NCM Mode**
  Identifies the method that was used to transfer new configuration content to the device when a change was initiated with Network Configuration Manager.
- **NCM User**
  Identifies the DX NetOps Spectrum user who initiated the configuration change on the device using Network Configuration Manager.
- **Device User**
  Identifies the user who accessed the device and made the configuration change.
- **Source**
  Identifies the source of the configuration change.
- **Location**
  Identifies the location of the configuration change.
- **Violated Policies**
  Identifies policies that were in violation after this configuration change.
- **Compliant Policies**
  Indicates policies that were compliant after this configuration change.

A new row is created when one or more changes are detected by Network Configuration Manager.

3. Select a row in the Host Configuration table.
   The captured host configuration content appears in the box below the table.
4. (optional) Click the changes hyperlink (if applicable) in the Line Changes column in the Host Configuration table to view added, removed, changed, and irrelevant lines in the configuration of the selected device.
   The Configuration Differences dialog opens. The highlighted text uses the following colors to indicate status:
   - **Green --** These lines were added.
   - **Red --** These lines have been removed.
   - **Blue --** These lines have changed.
   - **Grey --** These lines are irrelevant. Irrelevant changes are lines that match the comparison mask.

     NOTE
     Click Next or Previous to navigate through the differences in the file.
5. (optional) Click the View Differences hyperlink (if applicable) in the Running vs. Startup column in the Host Configuration table to view added, removed, changed, and irrelevant lines.
   The Running vs. Startup dialog displays differences in running and startup configuration files for the captured device. The startup configuration appears in the right column. The highlighted text uses the colors listed in the previous step to indicate status.

Compare Any Two Configurations

You can compare any two host configurations, even if they belong to different devices.

Follow these steps:

1. Select a device in the Explorer tab.
2. Verify the List tab is selected in the Contents panel and select the Host Configuration tab in the Component Detail panel.
3. Right-click a configuration in the Host Configuration table that you want to compare and select Start Compare.
4. Select the second configuration to include in the comparison. Select a configuration for the same device, or select a different device in the Explorer tab. Verify that its configuration information is displayed in the Host Configuration table.

5. Right-click the second configuration to include in the comparison in the Host Configuration table, and select Compare With <*name_of_first_configuration*>.
   The Configuration Differences dialog opens. The highlighted text uses the following colors to indicate status:
   - **Green --** These lines were added.
   - **Red --** These lines have been removed.
   - **Blue --** These lines have changed.
   - **Grey --** These lines are irrelevant. Irrelevant changes are lines that match the comparison mask.

     NOTE
     Click Next or Previous to navigate through the differences in the file.

Specify a Reference Configuration

You can specify a reference configuration for a device with an associated alarm. An alarm can be generated on the device whenever Network Configuration Manager determines that the current configuration differs significantly from the reference.

NOTE
For more information about alarm settings, see Configure Configuration Change Alert.

Follow these steps:

1. Select a device or device family in the Explorer tab.
   Verify the List tab is selected in the Contents panel.
2. Select one or more devices in the List tab whose most recent configuration you want to set as a reference.
3. Right-click the selection and select Set NCM Reference Configuration.
   A confirmation dialog opens.
4. Select Yes.
   The most recent configuration is set as reference for each device selected. The 'Is Reference' field in the Host Configuration table displays a check and the user who set the reference.

You can also manually specify a reference configuration.

Follow these steps:

1. Select a device in the Explorer tab.
   Verify the List tab is selected in the Contents panel and select the Host Configuration tab in the Component Detail panel.
2. Right-click the configuration to use as reference and select Set Reference.
   The configuration is designated as the reference configuration for this device. A check and the user who set the reference appear in the Is Reference field in the Host Configuration table.

You can change or remove a reference configuration specification after you set it. Only one configuration can be set as the reference configuration for a device. If you use the Set Reference or the Set NCM Reference Configuration option on another configuration when one is already set, the original one is automatically cleared, and the new one becomes the designated reference configuration. To clear a reference configuration, use the Unset Reference command from the right-click menu for the configuration in the Host Configuration table.

Configuration Alarms

You can specify alarms to be triggered when certain configuration changes occur. This section describes how to view the differences between configurations that triggered an alarm.

For more information about determining when configuration change alarms are triggered, see Configure Configuration Change Alert and Specify Configuration Change Alert Settings on a Single Device.

View Reference and Running Configuration Differences

You can view and compare reference and current running differences for a single device from the Alarm Details tab.

Follow these steps:

1. Select a device, a device family, or a global collection from the Explorer tab.
2. Select the Alarms tab in the Contents panel.
   Alarms for the selected item are displayed.
3. Select an alarm that displays “Reference and Current Running Configurations are Different” in the Alarm Title column.
4. Select the Alarm Details tab in the Component Detail panel.
   Alarm details are displayed.
5. Click the View Differences hyperlink.
   The Configuration Differences screen displays differences in the reference and current running files for the captured device. The reference configuration appears in the right column.

View Startup and Running Configuration

You can view and compare startup and running configuration for a single device from the Alarm Details tab.

Follow these steps:

1. Select a device, a device family, or a global collection from the Explorer tab.
2. Select the Alarms tab in the Contents panel.
   Alarms for the device, device family, or global collection are displayed.
3. Select an alarm that displays “Startup and Running Configurations are Different” in the Alarm Title column.
4. Select the Alarm Details tab in the Component Detail panel.
   Alarm details are displayed.
5. Click the View Differences hyperlink.
   The Running vs. Startup screen displays differences in running and startup configuration files for the captured device. The startup configuration appears in the left column.

View Global Sync Task Results

You can view lists of devices for which global synchronizations failed and succeeded.

> **NOTE**
> You can control whether to include devices with Network Configuration Manager disabled in the Failed Device List. For more information, see Configure Global Synchronization.

Follow these steps:

1. Expand Configuration Manager, and select Tasks in the Explorer tab.
2. Select the Global Sync Task.
   Information and results appear in the Information tab of the Contents panel.
3. Enter a name, type, condition, or device family in the Filter field to filter the results lists.

Network Configuration Manager Reports from Report Manager

Network Configuration Manager report options are included under the Network Configuration Management report pack in Spectrum Report Manager. Report Manager provides numerous report content, format, and report organization options. As a result, you can generate reports with the appropriate type and scope of information for different audiences in your organization who are interested in device configuration changes.

Report Manager Options

Report Manager provides you with multiple options for generating and managing your Network Configuration Manager reports:

- Generate reports on demand to view the most recent test results.
- Schedule test report generation on a one-time or periodic basis.
- Specify how long you want Report Manager to retain scheduled test reports or how many reports to retain.
- Specify email recipients for scheduled test reports.
- Schedule test reports for other Report Manager users.
- Publish reports in PDF, text, and spreadsheet formats.

NOTE
For detailed information about Report Manager features, see the Report Manager section.

The following reports are available:

- **Configuration Changes: All**
  Displays a summary of changes for all devices that have configuration changes. Each row represents a device to be associated with data that describes its configuration changes.
- **Configuration Changes: Group**
  Displays a summary of changes for devices in a given global collection. Each row represents a device to be associated with rolled up statistics describing its configuration changes.
- **Configuration Changes: Individual Device**
  Displays a list of configuration changes on a given device. Each row displays the time of the change, who made the change, and how many lines were changed. In addition, each row contains a web link to a Java applet that displays the difference between the current configuration and the previous configuration.
- **Detailed Configuration Event Log: All**
  Displays a reverse chronological list of events for all devices and models within DX NetOps Spectrum with Network Configuration Manager activity. Each entry in the list includes the IP address (if applicable), event text, event code, and event creator.
- **Detailed Configuration Event Log: Group**
  Displays a reverse chronological list of events for all devices and models with Network Configuration Manager activity for a specified global collection. Each entry in the list includes the IP address (if applicable), event text, event code, and event creator.
- **Detailed Configuration Event Log: Selected Device or Model**
  Displays a reverse chronological list of events for a specified device or model with Network Configuration Manager activity. Each entry in the list includes the IP address (if applicable), event text, event code, and event creator.
- **Top-N Configuration Changes: All**
  Displays a summary of changes for the "Top-N" devices that have configuration changes, where "Top-N" is defined as the maximum number of records based on the current sorting criteria. Each record represents a device to be associated with rolled-up statistics that describe its configuration changes.
- **Top-N Configuration Changes: Group**
  Displays a summary of changes for the "Top-N" devices that have configuration changes in a global collection. The "Top-N" is defined as the maximum number of records based on the current sorting criteria. Each record represents a device to be associated with rolled-up statistics that describe its configuration changes.

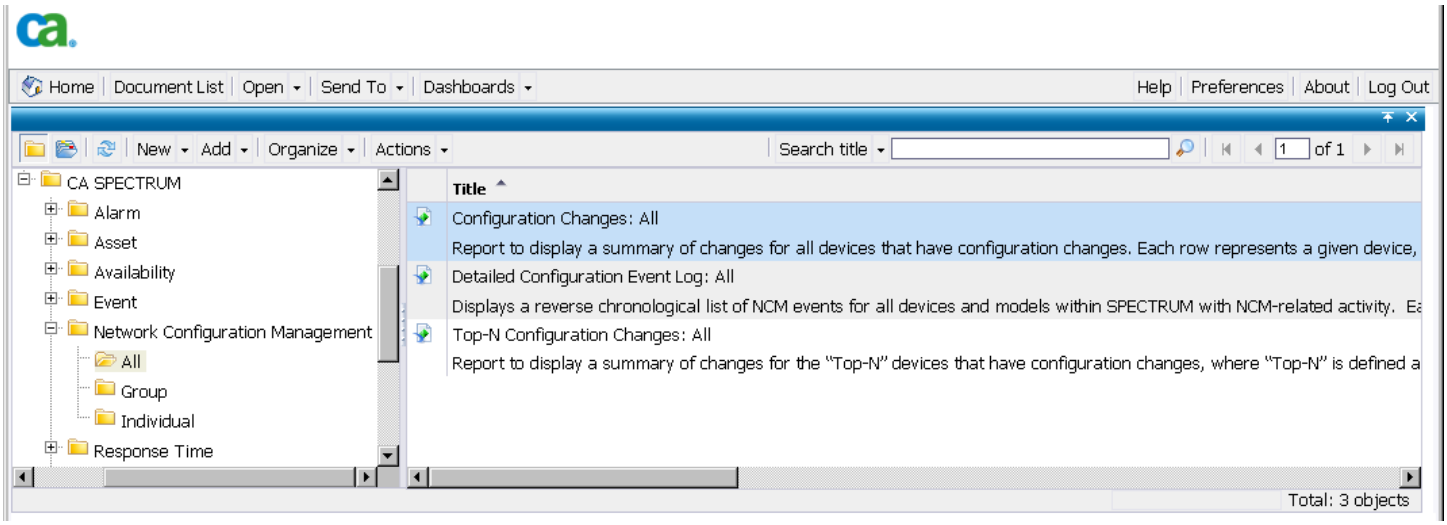
Generate Network Configuration Management Reports with Report Manager

You can generate Network Configuration Management reports using the Spectrum Report Manager.

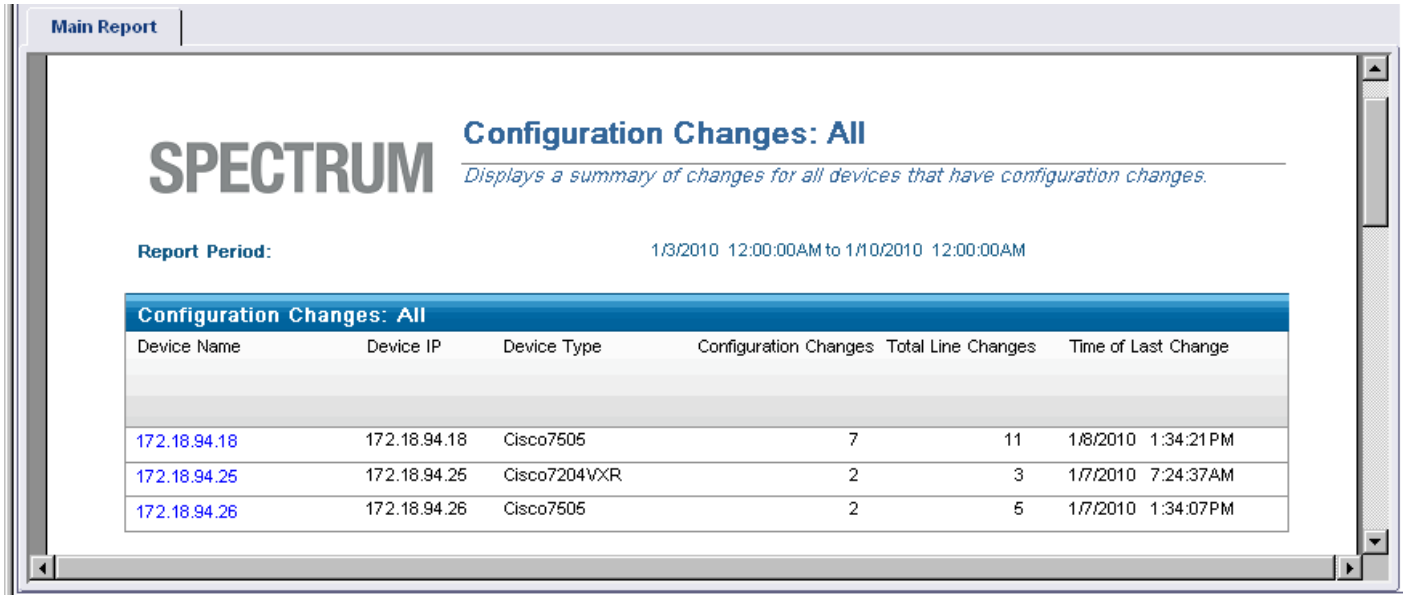
NOTE
The following example only provides an overview of the Network Configuration Management reports and features available in Report Manager. For more information, see the Report Manager section.

Follow these steps:

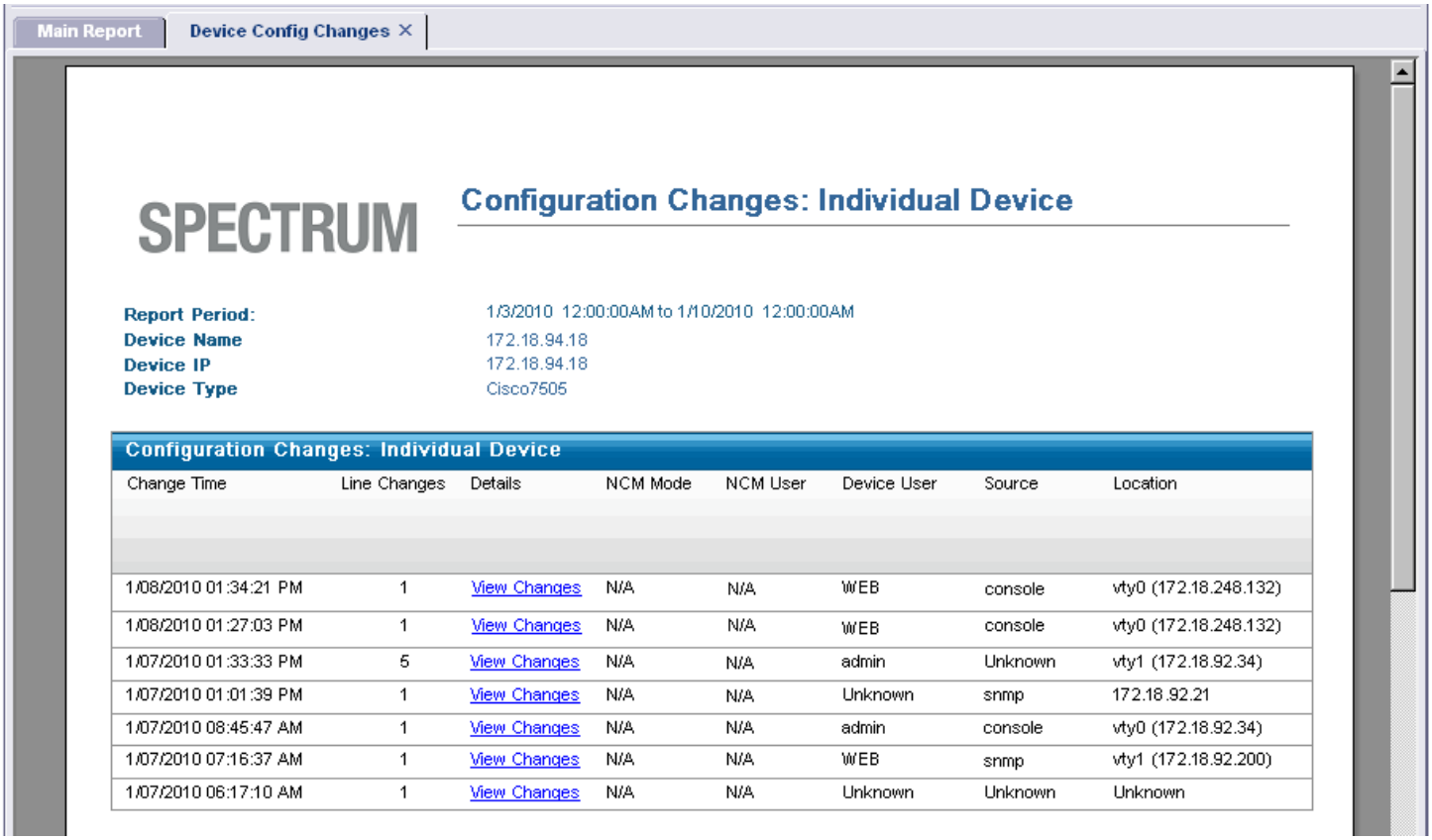
1. Select the type of test to generate.
   The following image shows the Network Configuration Management report options:



1. Configure the report. Select options for date and time range, supply a report title and subtitle, and select the landscape.
2. Click View Report to generate the report.
   The report displays. The following image shows an example of report results:



3. Click a Device Name hyperlink to examine results at the device level.
   The following image shows an example. From this view, you can click the View Changes hyperlink to view added, removed, changed, and irrelevant lines in the configuration of the selected device.



Network Configuration Manager Device-Level Tasks

This chapter describes how to manually capture, export, and upload configurations for devices in your network using Network Configuration Manager.

Manually Capture Configurations

Network Configuration Manager attempts to capture device configurations immediately after any change occurs. An unsolicited notification of configuration change can be either traps or MIB objects that are sent from the device where the change occurred. When it receives an unsolicited notification, the SpectroSERVER performs a capture and saves the configuration in the database to provide updated configuration data. You can also manually capture device configurations in OneClick.

Follow these steps:

1. Select a single device in the Explorer tab.
   The device appears in the List tab of the Contents panel.
2. Select the Host Configuration tab in the Component Detail panel.
   The results of any previous captures display.
3. Click the Capture Configuration icon.
   The results of the capture appear. Either a new configuration appears in the list or the last verified time is updated for the current configuration.

Manually Upload Configurations to a Single Device

You can manually upload a configuration file to a single device on your network. When you upload a configuration file, you merge it into the existing configuration file. You can use this feature to bring a newly installed device or a replacement/ standby remote device quickly online.

When uploading to a device in the Juniper JUNOS device family, use JUNOScript API format. For more information, see Juniper JUNOS Devices.

NOTE
You can upload configurations and view the results in real time by creating a bulk Upload task. For more information, see Create Upload Task.

Approval Not Required

The process to upload a device configuration to a single device varies depending on whether approval workflow is enabled. The following procedure describes the process when approval is not required.

NOTE

For information on workflow approval options, see Configure Workflow.

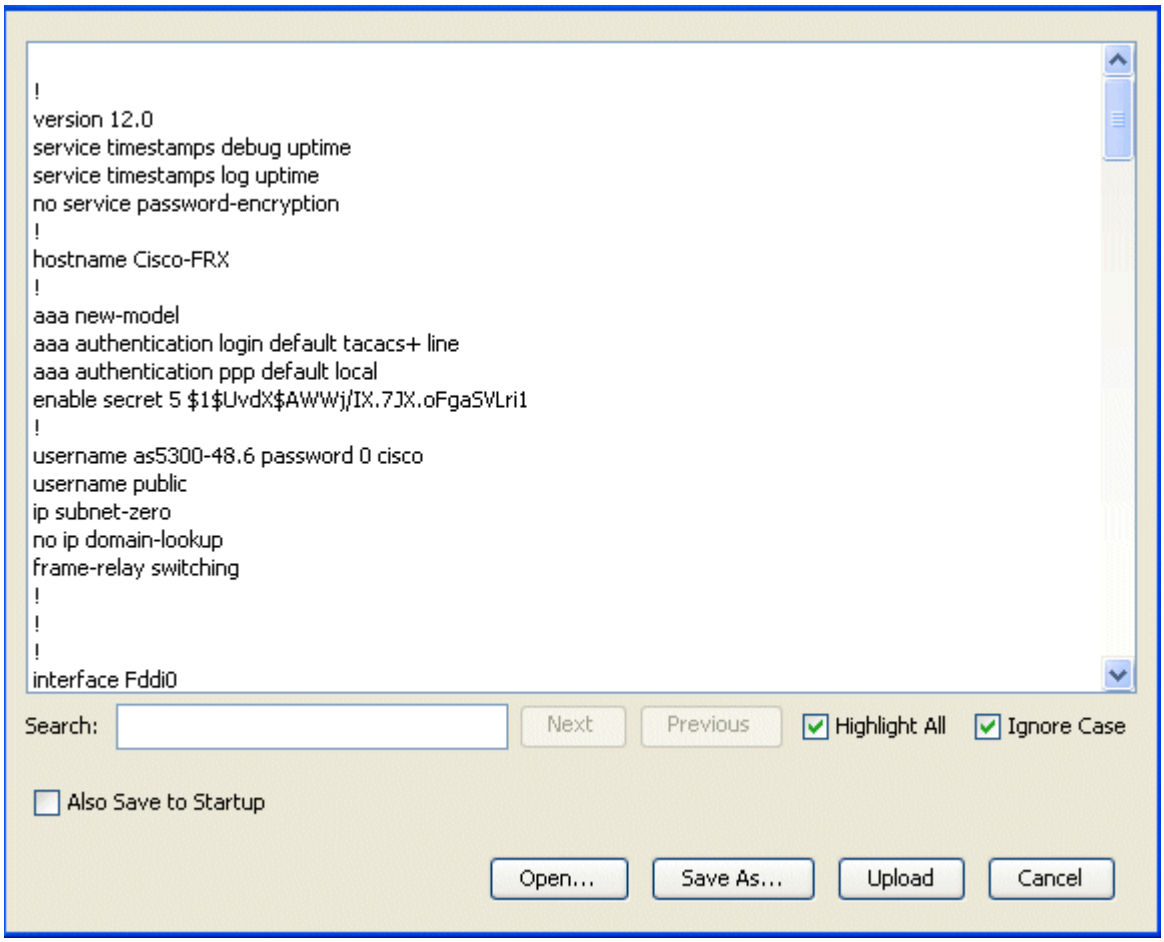
Follow these steps:

1. Select a single device or device family in the Explorer tab.
   The device or devices associated with the selected device family appear in the List tab of the Contents panel.
2. Select the Host Configuration tab in the Component Detail panel.
   The results of any previous captures display.
3. Click the Upload icon.

   NOTE

   If approval is required, the Approval Required dialog appears. Go to Approval Required to create a request for approval.

   The Upload Configuration screen appears with the last known configuration information for the selected device as shown in the following example:



4. Perform any of the following optional steps:
   - Edit configuration content as desired.
   - Enter criteria in the Search field to locate specific lines in the configuration file to change content or to verify content prior to an upload.
   - Select Also Save to Startup to write this configuration to the startup configuration. This will cause the configuration file to be loaded into the device when rebooted.

     NOTE
     This feature is only supported for Cisco, Foundry, and Nortel Passport L3 devices.
   - Click Open to import a previously exported configuration file that is saved locally on your system.
   - Click Save As if you want to save and export this configuration file in txt or html format.
5. Click Upload to upload the configuration file to the selected device.
   The message “The configuration upload succeeded” appears when the procedure is complete.

   NOTE
   Scheduling tasks is an available feature of bulk tasks. If you want to schedule your Upload task, see Network Configuration Manager Bulk Tasks.

Upload Configurations to a Single Device (Approval Required)

If approval workflow is enabled, your configuration changes must be approved before they can be processed. This is accomplished by creating a task for the upload request which can then be run after approval.

NOTE

For information on approval workflow options, see Configure Workflow.

Follow these steps:

1. Select a single device or device family in the Explorer tab.
   The device or devices associated with the selected device family appear in the List tab of the Contents panel.
2. Select the Host Configuration tab in the Component Detail panel.
   The results of any previous captures display.
3. Click the Upload icon.
   The Approval Required dialog appears.
4. Click Yes to continue.
   The Create NCM Task appears.
5. Create the task as follows:
   a. Enter a unique name in the Name field.

      NOTE
      Network Configuration Manager provides a default name (<*task type*>.YY-MM-DD_HH:MM.<*user name*>). For example, Upload.2006-10-17_15:48:04.Administrator.

   b. Enter a description for the task in the Description field.
   c. Select Reusable Task if you want the task to be available after it has run.
   d. Click Edit to specify content for uploading and merging into the device configuration in the Upload Content box. You can also click Open to import content from a text file. After you have made changes, you can click Save As to save and export this configuration file in txt or html.
   e. (optional) Enter criteria in the Search field to locate specific lines in the configuration file.
   f. Select Commit to Startup (if applicable) to copy the entire running configuration to the startup configuration after new content is merged.
   g. Select Alarm device on failure to generate an alarm on each device on which the task fails.
   h. Click Request Approval.

   The Approval Required dialog appears.
6. Select a user and enter an email address for a Task Approver, enter a task description (optional), and click OK to generate the request.
   A confirmation dialog appears indicating the request creation was successful. An email is sent to the Task Approver and the generated task appears in the Tasks folder in the Explorer tab.

   NOTE
   For email configuration information, see the OneClick Administration section.
7. Check approval status and run the task as described in Start a Task.

Network Configuration Manager Bulk Tasks

This chapter describes how to create an on-demand bulk Upload task, Sync task, and Save to Startup task with Network Configuration Manager. These tasks interact with devices by capturing and uploading host configurations. You can create a task to run anytime on a single device or on a list of devices. These on-demand tasks are useful if you want to roll out the same configuration (with minor differences such as IP addresses) to multiple devices on your network.

A task can be defined as reusable. A reusable tasks persists after execution and can be run again. If a task is not reusable, once it is run on a device or global collection, it is sent to the Lost and Found view and purged within 24 hours.

Create Upload Task

Create an automatic Upload task to perform bulk configuration uploads. A bulk Upload task merges new content into the running configurations of one or more selected devices. Devices are processed in random order.

WARNING
If you are uploading to Enterasys/Riverstone SSR devices, see Determine How an Enterasys/Riverstone SSR Device Responds to an Upload Task before continuing with this task. If you are uploading to a device in the Juniper JUNOS device family, you must use JUNOScript API format; for more information, see Juniper JUNOS Devices.

Follow these steps:

1. Select a single device, device family, global collection, a search result entry, or a container (such as Universe) in the Explorer tab.
2. Click the List tab and select the devices for the Upload task.
3. Select Upload Task from the Create NCM Task icon on the toolbar.
   The Upload Task dialog opens.

   NOTE
   If the selected devices do not appear in the Allow tab, click Disallow to display devices that are either disabled from Network Configuration Manager tasks or lack the necessary privileges.
4. Click Continue.
   The Create Task dialog appears.
5. Enter task information as follows:
   a. (Optional) Enter a unique name in the Name field.

      NOTE
      Network Configuration Manager provides a default name (<*task type*>.YY-MM-DD_HH:MM.<*user name*>). For example, Upload.2006-10-17_15:48:04.Administrator.
   b. (Optional) Enter a description for the task in the Description field.
   c. Select Reusable Task to make the task reusable.
   d. Click Edit to specify content for uploading and merging into the device configuration in the Upload Content box. You can also click Open to import content from a text file. After you have made changes, you can click Save As to save and export this configuration file in txt or html.
   e. Enter criteria in the Search field to locate specific lines in the configuration file.
   f. Select Commit to Startup (if applicable) to copy the entire running configuration to the startup configuration after new content is merged.
   g. Select Alarm device on Failure to generate an alarm on each device on which the task fails.
6. If approval is required (as indicated by the Request Approval button), take the following steps:
   a. Click Request Approval.
      The Approval Required dialog appears.
   b. Select a user and enter an email address for a Task Approver, enter a task description (optional), and click OK to generate the request.
      A confirmation dialog indicates that the request was created successfully. An email message is sent to the Task Approver ,and the generated task appears in the Tasks folder in the Explorer tab.

      NOTE

      For email configuration information, see the OneClick Administration section.
   c. Check approval status and run the task as described in Start a Task.

      NOTE
      For information on approval workflow options, see Configure Workflow.

7. If approval is not required (as indicated by the Save button):
   a. Click Save.
      The Task Saved dialog appears.
   b. Take one of the following steps:
      - Click Upload to upload the task to the selected device.
        The Upload Task Results dialog appears, and the generated task appears in the Tasks folder in the Explorer tab. For more on the results dialog, see View Task Results in Real Time.
      - Click Schedule to schedule the task for future execution. Scheduling is described in Schedule a Task.
        The task is saved and runs according to its schedule.
      - Click Close to save the task; it is available to run at a future time. You can edit and run the task by selecting it from Tasks in the Explorer tab under Configuration Manager.

Determine How an Enterasys/Riverstone SSR Device Responds to an Upload Task

Enterasys/Riverstone SSR devices do not respond consistently to configuration uploads. Some of these devices replace both the running and startup configurations with uploaded content. Others merge the uploaded content into both the running and startup configuration. Configuration captures performed on Enterasys/Riverstone SSR devices provide the startup configuration and not the running configuration. Therefore, we recommend testing devices that are running Enterasys firmware to verify the running and startup configurations.

Follow these steps:

1. Select a single SSR device from a search result or container (such as Universe) in the Explorer tab.
2. Click the List tab in the Contents panel.
3. Click the Host Configuration tab in the Component Detail panel.
   Previously captured configurations display.
4. Select the Capture Configuration icon to capture the current configuration of the selected device.
5. Select the Upload icon.
   The Upload Configuration screen appears. The content of the current startup configuration displays in the bottom pane.
6. Edit the existing configuration in the Upload Configuration screen. For example, remove the location line value (or remove a line):
   ```
   system set location "value"
   ```
7. Select the Upload icon again to upload the modified device configuration. Select the Capture Configuration icon again to capture the new configuration from the device.
   If the location is not present in the newly captured configuration, it indicates that the device is replacing (not merging) both the running and the startup configuration with the uploaded content.

Create Sync Task

Create an automatic Sync task to capture and verify policy-compliant device configurations for selected devices on your network and view the results in real time. When a Sync task captures device configuration, it checks the configuration against all policies pertaining to the device and, if specified, against the device startup configuration. Devices are processed in random order.

See Network Configuration Manager Policies for details about Network Configuration Manager policies.

See About Global Synchronization for details about running global sync tasks in background mode.

Follow these steps:

1. Select a single device, device family, global collection, a search result entry, or a container (such as Universe) in the Explorer tab.
2. Click the List tab in the Contents panel and select the devices that you want to include in the Sync task.

3. Click Sync Task from the Create NCM Task icon on the toolbar.
   The 'Select device(s) for Sync Task' dialog opens.

   NOTE
   If the selected devices do not appear in the Allow tab, click Disallow to display devices that are either disabled from Network Configuration Manager tasks or do not have necessary privileges.
4. Enter task information as follows:
   a. Enter a unique name in the Name field.

      NOTE
      Network Configuration Manager provides a default name (<*task type*>.YY-MM-DD_HH:MM.<*user name*>). For example, Sync.2010-09-09_15:48:04.Administrator.
   b. Enter a description for the task in the Description field.
   c. Select 'Alarm on device if startup differs' and appropriate severity to generate an alarm on each device where the captured configuration differs from its startup configuration.
   d. Click Edit Schedule to schedule the task for future execution. Scheduling is described in Schedule a Task.
   e. Select Reusable Task to make the task reusable.
5. Take one of the following steps:
   – Click Save to save the task; it is available to run at a future time. You can run the task by selecting it from Tasks in the Explorer tab under Configuration Manager.
   – Click Run Sync Task Now.
   The Sync Task Results dialog opens, and the generated task appears in the Tasks folder in the Explorer tab. For more information, see View Task Results in Real Time.

Create a Save to Startup Task

Create an automatic Save to Startup task to write a current running configuration to the startup configuration of one or more selected devices. A device saves its configuration in NVRAM (Nonvolatile Random Access Memory).

Devices are processed in random order.

Follow these steps:

1. Select a single device, device family, global collection, a search result entry, or a container (such as Universe) in the Explorer tab.
2. Click the List tab in the Contents panel and select the devices to upload.
3. Click Save to Startup Task from the Create NCM Task icon on the toolbar.
   The Save to Startup Task dialog opens.

   NOTE
   If the selected devices do not appear in the Allow tab, click Disallow to display devices that are either disabled from Network Configuration Manager tasks or lack the necessary privileges.
4. Enter task information as follows:
   a. (Optional) Enter a unique name in the Name field.

      NOTE
      Network Configuration Manager provides a default name (<task type>.YY-MM-DD_HH:MM.<user name>). For example, WriteStartup.2006-10-17_15:48:04.Administrator.
   b. (Optional) Enter a description for the task in the Description field.
   c. Select Reusable Task to make the task reusable.
5. If approval is required (as indicated by the Request Approval button), take the following steps:
   a. Click Request Approval.
      The Approval Required dialog appears.

b. Select a user and enter an email address for a Task Approver, enter a task description (optional), and click OK to generate the request.
A confirmation dialog indicates that the request was created successfully. An email is sent to the Task Approver and the generated task appears in the Tasks folder in the Explorer tab.

NOTE

For email configuration information, see the OneClick Administration section.

c. Check approval status and run the task as described in Start a Task.

NOTE
For information about approval workflow options, see Configure Workflow.

6. If approval is not required, take any of the following steps:
   – Click Schedule to schedule the task for future execution. Scheduling is described in Schedule a Task.
   – Click Save.
   The task is saved for future execution.
   – Click Run Save to Startup Task Now.
   The Save to Startup Task Results dialog opens, and the generated task appears in the Tasks folder in the Explorer tab. For more information, see View Task Results in Real Time.

NOTE
Any “Startup Versus Running Configurations are Different” alarms on task devices are automatically cleared by this task. For more information, see View Startup and Running Configuration Differences.

Firmware Upload

This section describes how to upload firmware for Cisco IOS and Cisco IOS - SSH Capable devices. Uploading firmware can be accomplished by either of two methods:

- Using the Load Firmware task
- Using Extension Utility scripting

WARNING
Uploading firmware is an advanced user feature and requires an expert level of knowledge. Modifying device firmware incorrectly may leave the device in an inoperative state.

About Firmware Upload

Firmware Upload is supported in that if a script is present it will use the script. If no script is present and the device supports the CISCO-FLASH-MIB, the MIB will be used.

Firmware Upload must accomplish certain tasks successfully and in a specific order for the transfer to be successful. This section describes these tasks and the firmware upload process.

NOTE
These tasks are handled by the Load Firmware Task, which is described in Create Load Firmware Task, or by a custom script.

These tasks are:

1. **Upload firmware image from the server to the device.** The device must be instructed to load the firmware image from a well-known server (the image server) to a specified flash or file system name. This upload can take minutes to hours to complete depending on the size of the image file and the network bandwidth.
2. **Upload boot command configuration to the device.** This occurs in three steps:
   a. Capture configuration. The configuration must be captured so that the new command can be inserted into the current configuration.

b. Upload change.
c. Write to NVRAM. The modified configuration must be written to startup so that the device will reload the specified image at boot time.

3. **Run reload script.** The reload command is written directly to enable mode and is not part of the configuration.

The system will use the default protocols or an override script for each phase as appropriate.

> **WARNING**
> If an error occurs in any of these steps, there is no rollback. The device is left in the last successful state.

Privileges

When uploading firmware as described in these sections, the following Network Configuration Manager privileges may be required:

- Load the Device Firmware
- Reload Device
- Schedule a Reload

For more information, see Network Configuration Manager Privileges.

Configure Device Firmware Transfer Settings

The section describes how to configure the protocol and server settings used to transfer the firmware image to the device. These settings are made at the device family level and reside in the Device Firmware Transfer Settings subview, which is available for the Cisco IOS and Cisco IOS - SSH Capable device families only. The devices in these device families support the CISCO-FLASH-MIB.

> **NOTE**
> Firmware Upload is supported out-of-box for Cisco IOS and Cisco IOS - SSH Capable device families only. For all other devices, the Extension Utility may be used to specify a Load Device Firmware script. See Network Configuration Manager Extension Utility for more information.

Follow these steps:

1. Select the Cisco IOS or Cisco IOS - SSH Capable device family from Device Families in the Explorer tab. Information and configurations appearin the Information tab of the Contents panel.
2. Expand the Device Firmware Transfer Settings subview.
   Firmware transfer options let you configure a firmware image transfer from a server or provide a custom script.
3. Take one of the following steps:
   – Modify the Firmware Image Transfer Protocol as needed.
   – Enter a Load Device Firmware script. For details on entering a script, see Enter a Configuration Script.

> **WARNING**
> If a script is present, the script is used, regardless of what is specified for the Firmware Image Transfer Protocol.

Display Cisco Flash Partition Information

In order to successfully upload a new firmware image to a device, you must have enough disk space available to support the image. This section describes a convenient way to review the available resources on a device before attempting a firmware upload.

> **NOTE**
> You can also display partition information for the device when creating the Load Firmware Task using the View Partitions button on the Create NCM Task dialog. For more information, see Create Load Firmware Task.

To display Cisco flash partition information

1. Select a device from either the Cisco IOS or Cisco IOS - SSH Capable device families in Device Families in the Explorer tab.
   Information and configuration settings for the device display in the Information tab of the Contents panel.
2. Expand the Cisco Flash Partitions subview.
   The following information appears:
   - **Name**
     Partition name.
   - **Number of Files**
     Number of files within the partition.
   - **Free Space**
     Amount of space available in the partition. There must be enough free disk space to support the new firmware image to be uploaded.
   - **Total Space**
     Total amount of space allocated to the partition.

Create Load Firmware Task

This section describes how to create a Load Firmware task, which is used to upload firmware to Cisco IOS and Cisco IOS - SSH Capable devices.

> **NOTE**
> To complete this task, you need to specify where on the device to upload the new firmware image. The target location must have enough space available to support the new image. To review the available resources on the device before creating the task, see Display Cisco Flash Partition Information.

Follow these steps:

1. Select a device from either the Cisco IOS or Cisco IOS - SSH Capable device families in Device Families in the Explorer tab.
2. Click the List tab in the Contents panel and select the devices for the Load Firmware task.
3. Select Load Firmware Task from the Create NCM Task icon on the toolbar.
   The Select Device(s) for Load Firmware Task dialog opens.

   > **NOTE**
   > If the selected devices do not appear in the Allow tab, click Disallow to display devices that are either disabled from Network Configuration Manager tasks or do not have necessary privileges.
4. Click Continue.
   The Create NCM Task dialog appears.
5. Create the task as follows:
   a. Enter a unique name in the Name field.

      > **NOTE**
      > Network Configuration Manager provides a default name (<*task type*>.YY-MM-DD_HH:MM.<*user name*>). For example, LoadFirmware.2006-10-17_15:48:04.Administrator.

   b. Enter a description for the task in the Description field.
   c. Select Reusable Task to make task reusable.
   d. Enter Image Information:
      - **Firmware Image Name**
        The file name of the firmware image on the image server.
      - **Destination**

The file name of the image as it will be on the device. Often this is the same name as that on the server and the value will auto-fill.

- **Boot Command**
  The name of the image to boot from. This will auto-fill with the destination name.
  Default: boot system flash
- **Backup Boot Command**
  The name of the image to boot from if an error occurs. This should be set to the current bootable image on the device. This will auto-fill with the destination name.
  Default: boot system flash
- **Reload the device after firmware upload**
  If selected, the Reload Information fields are enabled and the device will be reloaded after the firmware upload is successful.
- **View Partitions**
  Click View Partitions to display the Device Partitions dialog, which displays the available resources on the device. The target location must have enough space available to support the new image.

e. Enter Reload Information (if applicable):

- **Reload Immediately**
  Select Reload Immediately to reload the device immediately after firmware upload is successful. If this option is not selected, use the Timing fields to schedule the reload.
- **Save to Startup (if modified)**
  If the running configuration has been modified but not saved, indicate whether to copy it to startup before the reload begins.
- **Telnet Login Timeout**
  The timeout value (in seconds) to be used for the telnet connection while attempting to log in to the device.
- **Telnet Command Timeout**
  The timeout value (in seconds) to be used while attempting to execute commands over the telnet connection.

f. Click Server Settings and enter the following on the Edit Server Settings dialog to override transfer settings set at the device family level:

- **Protocol**
  The protocol to be used.
- **Server Address**
  The image transfer server address from which the device will copy the firmware image.
- **Time out (seconds)**
  The time out period before the device will fail the copy from the firmware image server.
- **Image Dir**
  The subdirectory on the image transfer server from which the file will be served.
  Note: This may be required if the images are not served from the root directory of the image server.
- **User Name**
  The user name required by the image transfer server.

  NOTE

  This may not be required by the specified protocol.
- **Password**
  The password required by the image transfer server.

  NOTE

  This may not be required by the specified protocol.

6. If approval is required (as indicated by the Request Approval button), take the following steps:
   a. Click Request Approval.
   The Approval Required dialog opens.

b. Select a user, and enter an email address for a Task Approver.
c. (Optional) Enter a task description, and click OK to generate the request.
A confirmation dialog indicates that the request was created successfully. An email message is sent to the Task Approver and the generated task appears in the Tasks folder in the Explorer tab.

NOTE

For information about email configuration, see the OneClick Administration section.

d. Check approval status and run the task as described in Start a Task.

NOTE
For information on approval workflow options, see Configure Workflow.

7. If approval is not required (as indicated by the Save button):
a. Click Save.
The Task Saved dialog opens.
b. Do one of the following:
- Click Upload Firmware to process the task.
The Load Firmware Task Results dialog opens, and the generated task appears in the Tasks folder in the Explorer tab. For more information, see View Task Results in Real Time.
- Click Schedule to schedule the task for future execution. Scheduling is described in Schedule a Task.
The task is saved and runs according to schedule.
- Click Close to save the task; it is available to run at a future time. You can edit and run the task by selecting it from Tasks in the Explorer tab under Configuration Manager.

Create Reload Task

Create a Reload task to reload a device after firmware has been uploaded. This task is available for Cisco IOS and Cisco IOS - SSH Capable devices.

NOTE
The functionality provided by the Reload task is also optionally available in the Load Firmware task.

Follow these steps:

1. Select a device from either the Cisco IOS or Cisco IOS - SSH Capable device families in Device Families in the Explorer tab.
2. Click the List tab in the Contents panel and select the devices for the Reload task.
3. Select Reload Task, Reload Task from the Create NCM Task icon on the toolbar.
The Select Device(s) for Reload Task dialog opens.

NOTE
If the selected devices do not appear in the Allow tab, click Disallow to display devices that are either disabled from Network Configuration Manager tasks or lack the necessary privileges.

4. If approval is required (as indicated by the Request Approval button), take the following steps:
a. Click Request Approval.
The Approval Required dialog opens.
b. Select a user, and enter an email address for a Task Approver.
c. (Optional) Enter a task description, and click OK to generate the request.
A confirmation dialog indicates that the request was successfully created. An email message is sent to the Task Approver, and the generated task appears in the Tasks folder in the Explorer tab.
Note: For information about email configuration, see the OneClick Administration section.
d. Check approval status and run the task as described in Start a Task.
When you start the task after it has been approved, the Reload Task dialog appears

NOTE
For information on approval workflow options, see Configure Workflow.

5. If approval is not required, click Run Reload Task Now.
   The Reload Task dialog appears.
6. Create the task as follows:
   a. Enter Reload Information:
      - **Reload Immediately**
        Select Reload Immediately to reload the device immediately. If this option is not selected, use the Timing fields to schedule the reload.
      - **Warm**
        Reload Warm (skip copying the image to NVRAM and uncompressing it).
      - **Save to Startup (if modified)**
        If the running configuration has been modified, indicate whether to copy it to startup before the reload begins.
      - **Telnet Login Timeout**
        The timeout value (in seconds) to be used for the telnet connection while attempting to log in to the device.
      - **Telnet Command Timeout**
        The timeout value (in seconds) to be used while attempting to execute commands over the telnet connection.
   b. Click OK.
      The Reload Device Task Results dialog opens, and the generated task appears in the Tasks folder in the Explorer tab. For more information, see View Task Results in Real Time.

Create Cancel Reload Task

A Cancel Reload task is used to cancel a pending reboot that has been scheduled on a device. This task is available for Cisco IOS and Cisco IOS - SSH Capable devices.

Follow these steps:

1. Select a device from the Cisco IOS or Cisco IOS - SSH Capable device family in Device Families in the Explorer tab.
2. Click the List tab in the Contents panel and select the devices for the Reload task.
3. Select Reload Task, Cancel Reload Task from the Create NCM Task icon on the toolbar.
   The Select Device(s) for Cancel Reload Task dialog appears.

   NOTE
   If the selected devices do not appear in the Allow tab, click Disallow to display devices that are either disabled from Network Configuration Manager tasks or do not have necessary privileges.
4. If approval is required (as indicated by the Request Approval button), take the following steps:
   a. Click Request Approval.
      The Approval Required dialog opens.
   b. Select a user and enter an email address for a Task Approver.
   c. (Optional) Supply a task description.
   d. Click OK to generate the request.
      A confirmation dialog indicates that the request was successfully created. An email message is sent to the Task Approver, and the generated task appears in the Tasks folder in the Explorer tab.

      NOTE

      For information about email configuration, see the OneClick Administration section.
   e. Check approval status and run the task as described in Start a Task.

      NOTE
      For information about approval workflow options, see Configure Workflow.
5. If approval is not required, perform any of the following tasks:

- Click Schedule to schedule the task for future execution. Scheduling is described in Schedule a Task.
- Click Save.
  The task is saved for future execution. Exit this procedure.
- Click Run Cancel Reload Task Now.
  The Cancel Reload Device Task Results dialog opens, and the generated task appears in the Tasks folder in the Explorer tab. For more information, see View Task Results in Real Time.

Load Device Firmware Script

The Load Device Firmware Script can be used to initiate a load of the specified firmware image on a device as an alternative to the internal MIB-based support for Load Firmware Task (only for the Cisco devices that support CISCO-FLASH-MIB). For more information on using scripts, see Network Configuration Manager Extension Utility.

Managing Tasks

Associating Tasks with Global Collections

Tasks can be associated with global collections. By associating a task with a global collection, the task runs on all members of the collection that support the task type at execution time. Associating a task with a global collection can occur during initial task creation or after the task already exists.

> **NOTE**
> The 'Include Global Collection in NCM Task' privilege is required for a user to associate a task with a global collection. For more information on privileges, see Network Configuration Manager Privileges.

Associate a New Task

You can associate a task with a global collection when you create the task.

Follow these steps:

1. Select the Global Collections node in the Explorer tab.
   A list of defined global collections appears in the List tab of the Contents panel.

   > **NOTE**
   > If no global collections exist, you must create one before you can continue. For more information, see the Modeling and Managing Your IT Infrastructure section.

   On the List tab, select the global collection with which to associate a task.
2. The global collection is highlighted and the Create NCM task icon is enabled.
3. Click the Create NCM task icon and select the task that you want to create and associate with this global collection.
   The 'Select device(s)' dialog for the task appears.
4. Continue with the creation of the task as described in Network Configuration Manager Bulk Tasks or Firmware Upload, depending on the task.
   When you have finished:
   - The new task appears in the NCM Tasks subview on the Information tab for the global collection.
   - The new task appears in the Tasks folder in the Explorer tab. When viewing the Information tab for the task, the global collection with which the task is associated appears in the Global Collections subview.

   When the task is executed, it runs on all members of the global collection that support the task type at the time of execution.

Associate an Existing Task

You can associate an existing task with a global collection, and you can also perform the association while creating the task. The following procedure can be used to add another global collection to an existing task or to remove a collection.

Follow these steps:

1. Select the task from the Tasks folder under Configuration Manager in the Explorer tab.
   Information for the task appears in the Information tab of the Contents panel.
2. Expand the Global Collections subview.
   Any global collections that are associated with the selected task appear in the table.
3. Click the Add or Remove Global Collections icon above the table.
   The Task Members Editor dialog opens.
4. Select global collections from the Available Global Collections pane (on the right). Use the arrows to move them into the Associated Global Collections pane (on the left) to be associated with this task.
   Global collections that are in the Associated Global Collections pane will be associated with this task.
5. Click Save, and then Yes in the subsequent confirmation dialog.
   The associated global collections appear in the table. When the task is executed, it will run on all members of the associated global collections that support the task type at the time of execution.

Scheduling Bulk Tasks

Bulk tasks can be scheduled. Scheduling can be accomplished either at the time of creation of a task or after the task has run (in the case of reusable tasks).

The following tasks can be scheduled: Upload Task, Sync Task, Save to Startup Task, Load Firmware Task, and Cancel Reload Task.

> **NOTE**
> Reload tasks cannot be scheduled through this mechanism; instead, the internal scheduling mechanism of a device is used. If you define a script to accomplish the reload operation, the script must leverage the scheduling mechanism of the device to schedule reload tasks. For more information, see Enter a Configuration Script.

A task can be associated with only one schedule. If a task already has an existing schedule when a new schedule is specified, the previous schedule is removed. You must manually delete recurring tasks; no automatic cleanup is performed.

Tasks are essentially distributed. Each "local" task runs at the scheduled time, based on the local time zone of the local landscape. The recommended best practice is to have all SpectroSERVERs working with the same time zone setting. The Time Completed column in the Succeeded Device List and Failed Device List tables show at what time the task operation was attempted on a particular device. This capability can help determine the time when a task is run in a DSS with multiple landscapes in different time zones.

> **NOTE**
> The Network Configuration Manager Schedule NCM Tasks privilege is required to schedule bulk tasks.

Reusable Tasks

Defining a task as reusable allows you the ability to save and run a task multiple times without having to redefine it. You can also create a recurring schedule to automatically run the task at predetermined times.

For information on scheduling a task, see Schedule a Task.

> **NOTE**
> A task with a recurring schedule will automatically be created as a Reusable task.

A task is designated as reusable when the Reusable Task option is specified during task creation.

The task is identifiable as reusable in following areas:

- In the Reusable field in the List table in the Contents panel
- Within General Task Information in the Information tab in the Component Detail panel

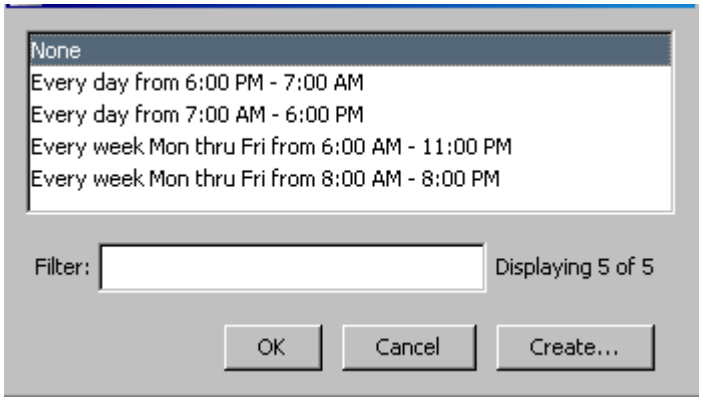
Schedule a Task

Scheduling a task lets you define a task and then specify a future date and time when it runs. Schedule a task to run one time only or on a recurring basis. You can set up the schedule at the time of task creation. Use the Schedule or Edit Schedule button. Or, for reusable tasks, you can set up a schedule at any time.

Note: Tasks run from the Host Configuration tab cannot be scheduled.

Follow these steps:

1. Follow the procedures outlined in Network Configuration Manager Bulk Tasks and Firmware Upload to create any of the following tasks: Upload Task, Sync Task, Save to Startup Task, Load Firmware Task, and Cancel Reload Task. The Schedule or Edit Schedule button appears on the creation dialog where it is available.
   Note: If Approval Workflow is enabled, the Schedule button is not available during task creation; only approved tasks can be scheduled. You must set up a schedule after the task has been created. See below for the procedure.
2. Select the Schedule or Edit Schedule button.
   The Select Schedule dialog appears as shown in the following image:



3. Take one of the following steps:
   – Select a default schedule, and click OK.
   – Create a custom schedule: click the Create button, specify schedule options, and click OK.
   The custom schedule is added to the list. Select the new schedule and click OK.

   The task is now scheduled. The schedule appears next to the Schedule button.

   NOTE
   To remove the schedule for a task, select the default schedule of None.
4. If you want to run this task multiple times, select Reusable Task. If you specified a recurring schedule in the previous step, the task should be created as a reusable task.

   NOTE
   Reusable tasks will not be cleaned up automatically.
5. Take one of the following steps:
   – Save task. If you have specified a schedule and want to run the task from the List tab in the Tasks folder at a later time, click the Save button. The task is created with the associated schedule (if any).
   – Run task. If you want to run the task right away, click the Run button.

   The task is saved and appears in the Tasks folder in Explorer with the Scheduled Task icon
   Schedule information is available in the Schedule field in the List table in the Contents panel and within General Task Information in the Information tab in the Component Detail panel.

You can also create or modify a schedule after task creation for any of the following reasons:

- Before setting up the task to run on a recurring basis, you want to test the task thoroughly.
- Because a task that requires approval cannot be scheduled until after it has been approved, you must first create the task and then wait for the approval.
- Site situations have changed, requiring schedule modification.

Follow these steps:

1. Select Tasks in Explorer and the List tab in the Contents panel.
   All defined tasks appear.
2. Select a task whose schedule you want to create or modify. The following conditions must be met for a task to be eligible for scheduling:
   – Task must be eligible to be run. Either it is a reusable task (Reusable = Yes) or if it is not reusable, then it has not been run yet (Inactive state).
   – If Approval Workflow is enabled, task must be in Approved state.
   The Schedule button in the toolbar is enabled for the task if it can be scheduled.

   NOTE
   If the Schedule button is not enabled, verify that the eligibility conditions are met.
3. Click the Schedule button.
   The Select Schedule dialog opens.
4. Select or create a schedule.

Starting and Stopping Tasks

This section describes how to start, stop, resume, and delete tasks.

Start a Task

This procedure describes how to start a task that has already been created. Tasks can be started if they are not already running and they are reusable tasks or, if they are not reusable tasks, have not been run at all. If approval is enabled, then the tasks have to be in the Approved state.

NOTE
Any configuration change task that requires approval must be approved before it can be run.

Follow these steps:

1. Select the task from the Tasks view under Configuration Manager in the Explorer tab.
2. Select the List tab in the Contents panel.
   Information about the task displays in the List table. A value of Awaiting Approval in the State column indicates the request has been generated but not yet approved; a value of Approved indicates the request has been approved and can be run.
3. Select a task to start.
   The Start button is enabled if the task can be started.
4. Click the Start Selected Task icon to run the task.
   If information is required for a task, refer to the section for that task. Otherwise, the task starts and the Task State value is updated.
   Depending on the task, a results dialog can appear. For more information, see View Task Results in Real Time.

Stop a Task

You can stop a task while it is running.

Follow these steps:

1. Select Tasks under Configuration Manager in the Explorer tab.
2. Click the List tab in the Contents panel, and select a task.

   NOTE
   Only tasks with a State of Running can be stopped.
3. Click the Stop Selected Task icon in the toolbar.
   The task stops, and the Task State value is updated.

Resume a Task

A task can be resumed if it has been stopped and has devices left in the Remaining Device List. When you resume a task, Network Configuration Manager only attempts to run the operation on those devices in the Remaining list. The operation is not reattempted on those devices that previously succeeded or failed and were removed from the Remaining list.

Follow these steps:

1. Select Tasks under Configuration Manager in the Explorer tab.
2. Click the List tab in the Contents panel and select a task to be resumed.

   NOTE
   Only tasks that have devices remaining, as represented by a positive value in the Remaining column, can be resumed.
3. Click the Resume Selected Task icon in the toolbar.
   The task starts, and the State value is updated.

Delete a Task

You can delete tasks in OneClick.

NOTE
Tasks that are running or locked for edit cannot be deleted.

Follow these steps:

1. Select Tasks under Configuration Manager in the Explorer tab.
2. Click the List tab in the Contents panel and select a task for deletion.
3. Click the Delete selected tasks icon in the toolbar.
   The Confirm Delete dialog appears.
4. Click Yes to delete.
   The selected task is deleted.

Viewing Task Information

This section describes how to view information for tasks that have been created and that have been run.

View Task Results in Real Time

After you have started an Upload, Sync, Save to Startup, Load Firmware task, Reload, or Cancel Reload task, a results dialog opens. The name, condition, type, and status (Pending, Failed, or Succeeded) of the task are shown in the Results tab. If Failed, the results appear in the Cause of Failure field.

Note: The task statistics are updated on a 10-second poll cycle.

While the task is running, you can do the following on the results dialog:

- Click the Content tab to view the content that you are uploading.

NOTE
The Content tab is available on the Upload Task Results dialog and Load Firmware Task Results dialog only.

- Click Stop to cancel the task. The task finishes processing any devices that are in progress. It then stops processing any remaining devices.
- Click Close to run the task in the background.

View Critical Statistics on All Bulk Tasks

You can view critical statistics for all bulk tasks simultaneously.

Follow these steps:

1. Select Tasks under Configuration Manager in the Explorer tab.
2. Select the List tab in the Contents panel.
   Statistics for all bulk tasks display.

View Detailed Statistics for a Bulk Task

Take different steps to view detailed statistics for a single bulk task.

Follow these steps:

1. Select a task from Tasks folder in the Explorer tab.
2. Click the Information tab in the Contents panel.
   Information on the task displays.

Task State and Status Values

The Task State (State) and Task Status values identify the current stage of execution for the task. The Task State (State) and Task Status are available when viewing task results or statistics. To access these views, see Viewing Task Information.

Task State

The following are possible Task State (State) values:

Approved
Approval workflow mode has been enabled for this task. The task has been approved by the appropriate Task Approver and can be run.

Awaiting Approval
Approval workflow mode has been enabled for this task. A request for this task has been generated but not yet approved.

Completed
The task has run successfully and is reusable. A task that is on a recurring schedule and has run at least once will have this state.

Completed awaiting Destroy
The task has run and is not reusable. The task will be purged within 24 hours.

Denied
Approval workflow mode has been enabled for this task. A request for this task was generated and has been denied by the appropriate Task Approver.

Inactive
The task has been scheduled but not yet run.

Initializing

Task preparation within DX NetOps Spectrum has started.

Running
The task is currently running. Tasks in this state can be stopped.

Stopping
The task started and then was stopped by the user.

Task Status

The following are possible Task Status values:

Failed
The task did not complete successfully. The results are displayed in the Cause of Failure field.

Pending
The task is currently running. Tasks in this state can be stopped.

Succeeded
The task has completed successfully.

Network Configuration Manager Policies

This chapter describes how to create and configure Network Configuration Manager policies. Network Configuration Manager policies monitor content in configurations and verify that device content is compliant.

> **NOTE**
> We recommend that you have configurations captured prior to setting up Network Configuration Manager policies. See Global Synchronization Task to set up a global synchronization task on your network.

About Network Configuration Manager Policies

A Network Configuration Manager *policy* defines criteria that are used to monitor content for a device host configuration. A policy is checked and compared every time a device host configuration file is captured. Devices that violate the policy can generate an alarm and can receive remediation.

Policies can be created and applied to devices, device families, and to global collections. When it is applied to a global collection, the policy is enforced on all global collection members of the selected device families in that global collection. See Network Configuration Manager and Global Collections for details about setting up global collections.

You can create two types of policies: single-line policies and multi-line block policies. They are described in the following sections.

> **NOTE**
> Configuration captures that are performed on Enterasys/Riverstone SSR devices provide the startup configuration and not the running configuration. Therefore, the startup configuration is used when determining whether a device is compliant with Network Configuration Manager policies. See Determine How an Enterasys/Riverstone SSR Device Responds to an Upload Task for information about how SSR devices handle Network Configurations Manager configuration uploads.

Single Line Policies

A single line policy compares the currently defined host configuration to the policy definition one line at a time. Each line of data in the host configuration is analyzed against the policy. This type of policy is useful when checking for the existence of a single command throughout the entire configuration.

Example

Suppose that your site has a regulation that all switches must have http enabled. A switch is brought online with the following in its configuration:

```
#http configuration
set ip http server disable
set ip http port 80
```

For this device to comply with site regulations, "set IP HTTP server disable" should be "set IP HTTP server enable". To identify and correct this situation, you can create a single line policy to check if the configuration has the line "set IP HTTP server enable". If this line is missing from the configuration, you can specify that an alarm is generated so that the condition can be repaired. From the alarm, the policy violation can be viewed and you then have the option to repair the device by scheduling a task to upload the corrected content.

Multi-line Block Policies

A multi-line block policy compares the currently defined host configuration to the policy a block at a time. The policy attempts to match corresponding blocks between the policy and the current host configuration. A block is designated by start and end tags; only data within qualifying blocks is analyzed by the policy. This type of policy is useful when monitoring settings for a block of configuration text such as an interface configuration. Most devices have multiple interfaces where unique settings for individual interfaces appear in the same configuration file.

There are two options available when enforcing a block policy: the configuration content can be compared to a pre-defined set of policy criteria, or it can be compared with the previous configuration or reference configuration in the configuration history.

When comparing with a previous or reference configuration, lines that have been changed, added, or removed are identified. Comparing with a previous or reference configuration is useful for highlighting changes that occur in the context of the block; changes that occur outside of a designated block will be shown as masked or irrelevant changes.

Example

Suppose that you want to shut down certain interfaces that have been identified by the word "shutdown" appearing in their descriptions. You can identify such devices by defining multi-line block policies in the following ways:

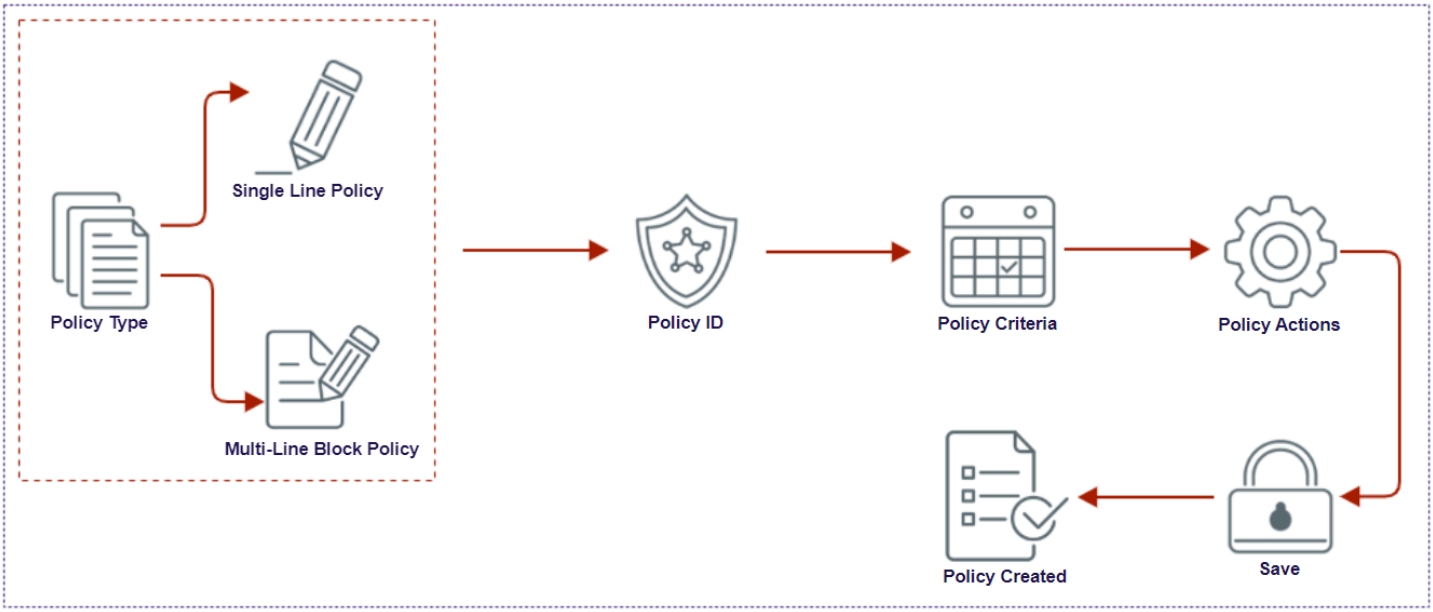
- By comparing to specified contents
- By comparing content with a script
- By comparing to another configuration

After the devices are identified, the shutdown command can then be issued easily for those interfaces that are marked for shutdown as part of the recommended upload for corrective action.

For the detailed implementation of this example of multi-line block policy, see Multi-line Block Policy Example.

Policy Creation Process

The following illustration outlines the high-level policy creation process:



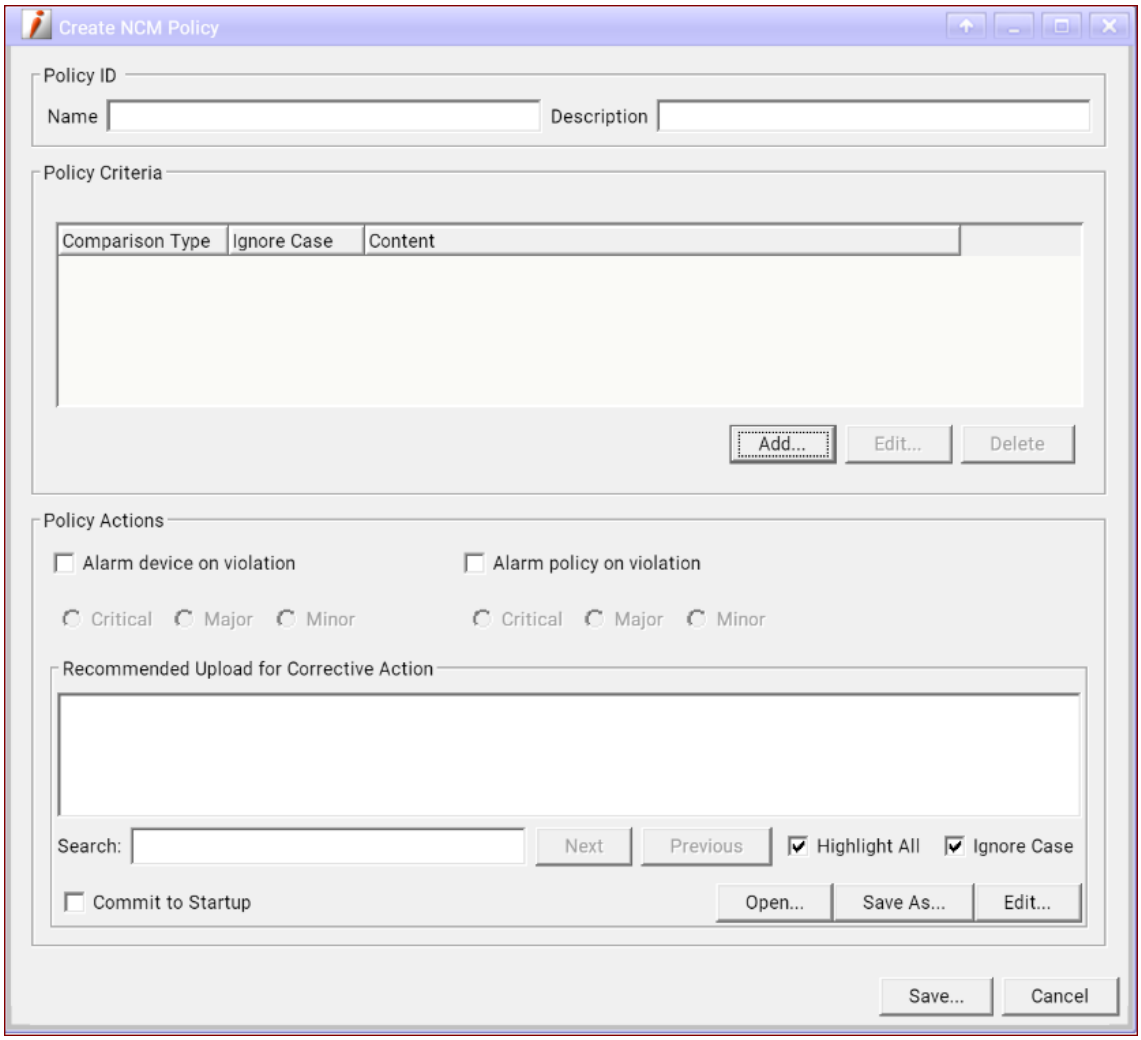
Create a Policy

A policy defines criteria that are used to monitor content for a device host configuration. You can create two types of policies: single-line policies and multi-line block policies. The following procedure describes how to create a Network Configuration Manager policy.

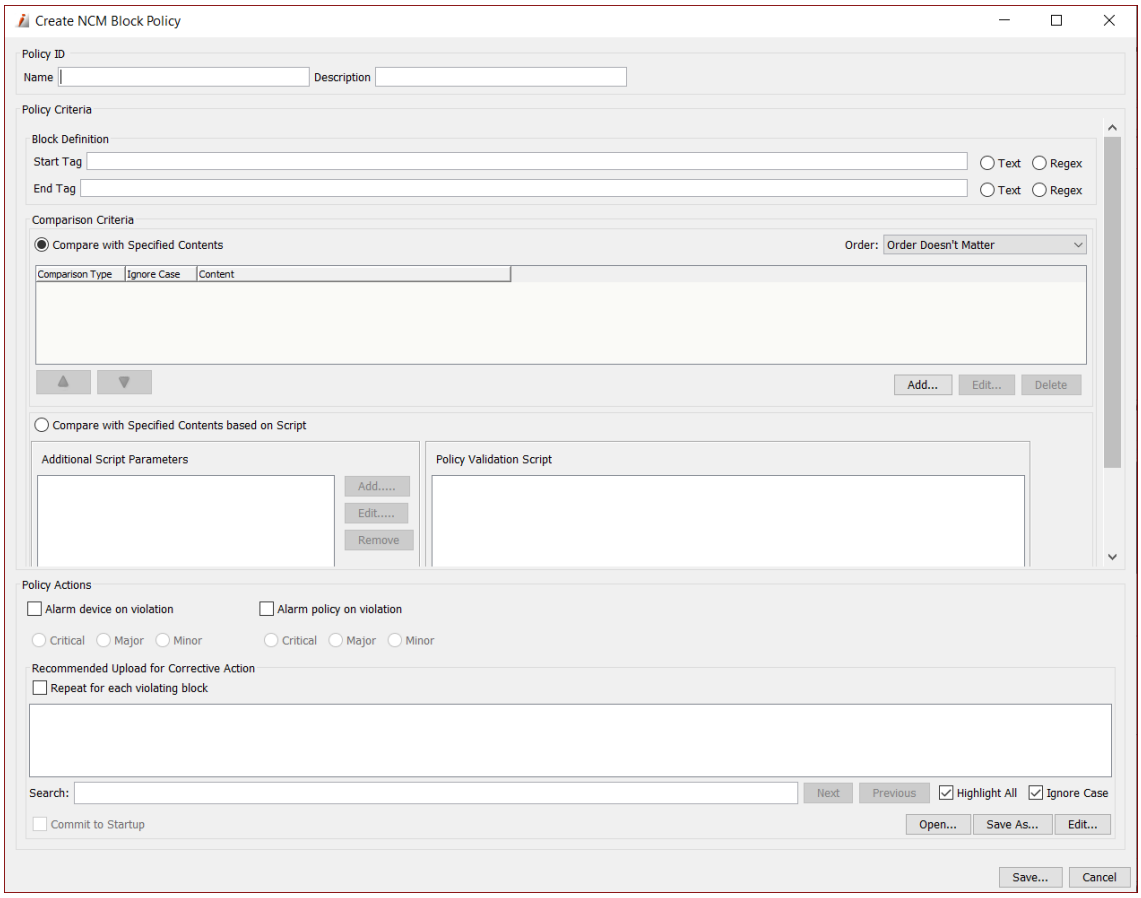
> **NOTE**
> An example is provided in the Multi-line Block Policy Example.

Follow these steps:

1. Access the **Explorer** tab in the OneClick UI.
2. Navigate to **Configuration Manager, Policies**.
3. Click the **List** tab in the **Contents** pane.
4. Select the "Create policy" icon.
   The **Select Policy Type** dialog appears.
5. Select the type of policy you want to create:
   – **Single Line Policy.** Creates a policy where only a single line of configuration is compared at a time. The **Create NCM Policy** dialog opens. The following screenshot shows the single-line policy options:



– **Multi-line Block Policy.** Creates a policy where a host's configuration is compared by qualifying blocks. The **Create NCM Block Policy** dialog opens. The following screenshot shows the multi-line block policy options:



6. In the **Policy ID** section, enter a name and description for the policy.

   NOTE
   In 10.4.2, there is no device family association section in the UI. In previous releases, this section was available.

7. In the **Policy Criteria** section of the dialog, configure policy criteria as follows:
   - For single-line policies, see Single Line Policy Criteria.
   - For multi-line block policies, see Multi-line Block Policy Criteria.
8. In the **Policy Actions** section of the dialog, do the following:
   a. Enter the alarm criteria as follows:
      - **Alarm device on violation**
        Indicates whether to alarm a device when a device is non-compliant with this policy. This is a single alarm on each non-compliant device, viewable in the Alarms tab. You can also select the severity of the alarm (Critical, Major, or Minor). You must enable the policy for this option to take effect.
      - **Alarm policy on violation**
        Indicates whether to alarm the policy when at least one device is non-compliant. This is a single alarm on a single policy, viewable in the Alarms tab. You can also select the severity of the alarm (Critical, Major, or Minor). You must enable the policy for this option to take effect.

   b. Enter a recommended upload for corrective action.
      - For single-line policies, see Single Line Policy Corrective Action.
      - For multi-line block policies, see Multi-line Block Policy Corrective Action.

   c. Select the **Commit to Startup** option to indicate whether to copy the entire running configuration to the startup configuration after the new content is merged.
9. Click **Save**.

The **Save NCM Policy** or **Save NCM Block Policy** dialog appears.

10. Click **Back** if you want to make any edits to the policy or click **Exit** to save the policy.

You have successfully created and saved a policy.

Policy Criteria

The types of policy criteria that you can specify differs between single line and multi-line block policies. This section describes how to specify policy criteria according to the type of policy you have defined. This section contains the following topics:

- Single Line Policy Criteria
- Multi-line Block Policy Criteria
- Policy Criteria Dialog

Single Line Policy Criteria

Use the following procedure to specify the comparison criteria for a single line policy.

To specify criteria for a single line policy

1. On the **Create NCM Policy** dialog in the **Policy Criteria** section, select **Add** to create criteria for comparison.
   The **Policy Criteria** dialog opens.
2. Configure policy criteria as described in Policy Criteria Dialog.
   After the **Policy Criteria** dialog is completed, the new criteria for comparison appears in the table.
3. To add more criteria or to modify existing criteria, use the **Add**, **Edit**, and **Delete** buttons.

The remainder of this dialog, including saving of the policy, is described in Create a Policy.

Multi-line Block Policy Criteria

When defining a multi-line block policy, you must specify two types of criteria: block definition criteria and comparison criteria. Block definition criteria define what constitutes the start and end of a block; comparison criteria define content that is used to compare against the current host configuration.

This section describes how to define this criteria and contains the following topics:

- About Blocks
- Specify Multi-line Block Policy Criteria
- Compare with Specified Contents
- Compare with Specified Contents based on Script
- Compare with Matching Block from Reference or Previous Configuration

About Blocks

When using multi-line block policies, you need to know what constitutes a block in the host configuration file for your device. In the following example for a Cisco IOS - SSH Capable device, a block similar to the following exists for each interface. This block would be delimited by the line "interface *name*" and the comment character "!":

```
interface Loopback0
 description "test 123"
 ip address 138.42.96.6 255.255.255.255
 ip pim sparse-dense-mode
 no ip route-cache cef
 no ip route-cache
 ipv6 address 2002:8A2A:5E12:8A2A:6006::1/128
 ipv6 enable
```

```
  ipv6 rip IPv6-1 enable
 !
```

This information is used when defining the policy. In block policy terminology, this block would be defined by:

1. **Start Tag:** interface *name*
2. **End Tag:** !

You can use either text or regular expressions to define what constitutes the start and end of a block. The following describes how the two options differ when determining what qualifies as a start or end tag.

> **NOTE**
> The values that are defined as Start Tag and End Tag is included as part of the block.

Using Text

When using text, the entire line that contains the matching text is matched to that field. For example, if you use "interface" of Text type as the start tag, this matches every line that contains the word "interface" and regards it as the starting line for a block.

Using Regular Expressions (Regex)

When using regular expressions, only an exact match of the regular expression pattern (and not the entire line) is matched to that field. For example, if you specify "interface abc" as the end tag, then only content up to "interface abc" will be considered as the end of the block. If, instead, you specified "interface abc.*" (where ".*" is a wildcard pattern in regular expressions that matches any characters in a line), then the entire line that matched "interface abc" would be considered as the end of the block.

Specify Multi-line Block Policy Criteria

The following procedure describes how to specify criteria for a multi-line block policy.

Follow these steps:

1. On the **Create NCM Block Policy** dialog in the **Policy Criteria** section, specify the following **Block Definition** criteria. You can use either text or regular expressions to define what constitutes the start and end of a block. For additional explanation on how the two options differ, see About Blocks.

   > **NOTE**
   > The values that are defined as Start Tag and End Tag are included as part of the block.
2. In the Comparison Criteria section, select one of the following options:
   - **Compare with Specified Contents**
     Specifies that the policy compares the current host configuration against user-defined content that is specified in this policy. For more information, see Compare with Specified Contents
   - **Compare with Specified Contents based on Script**
     Specifies that the policy compares the current host configuration against the provided script. For more information, see Compare with Specified Contents based on Script.
   - **Compare with Matching Block from Previous Configuration**
     Specifies that the policy compares the current host configuration to content from a previous or reference configuration. For more information, see Compare with Matching Block from Reference or Previous Configuration.

The remainder of this dialog, including saving of the policy, is described in Create a Policy.

Compare with Specified Contents

When defining a multi-line block policy, you can specify explicitly what content to check for in each block of the current configuration. This procedure describes how to set up user-defined criteria in a multi-line block policy.

To set up user-defined comparison criteria

1. On the Create NCM Block Policy dialog with the 'Compare with Specified Contents' option that is selected, specify the Order. The following are available options:
   - **Order Doesn't Matter**
     Indicates that the order of the criteria is not considered when comparing with the current host configuration. The policy is violated based on content only.
   - **Preserve Order (Allow Extra Lines)**
     Indicates that the specified content must appear in the order that is specified to comply with the policy; however, additional content that is interspersed between what is specified is allowed. The policy will be violated if some of the specified content does not exist in the configuration or if it exists in a different order. The policy ignores unmatched lines.
   - **Preserve Order with No Extra Lines**
     Indicates that the specified content must appear both in the order that is specified and contiguously to comply with the policy. The policy is violated if the configuration block does not match exactly with the specified content; any extra lines in the block content that were not explicitly defined by the specified content violates the policy.
2. Select Add to create criteria for comparison.
   The Policy Criteria dialog appears.
3. Configure policy criteria as described in Policy Criteria Dialog.
   After the Policy Criteria dialog is closed, the new criteria for comparison appears in the table. The order of the criteria in the table will be used if you have specified that order is preserved when comparing to the host configuration.
4. To add more criteria or to modify existing criteria, use the Add, Edit, and Delete buttons.

Compare with Specified Content Based on Script

When defining a multi-line block policy, you can explicitly set a script to validate content for each block of the current configuration. You define custom scripts to validate the content. DX NetOps Spectrum provides a default script for reference. By default, the bash prompt is used to execute the script. You can change the default program using the `verify-block-script-program-type` tag in the `$SPECROOT\NCM\config.xml` file.

Follow these steps:

1. Select **Compare with Specified Content based on Script** in the **Create NCM Block Policy** dialog.
2. Specify the script content and additional parameters to execute the script.
   A temporary file with validating block content is created in the `$SPECROOT\NCM\cache` folder and passed as the first argument to the script.

   NOTE
   The script rewrites TRUE or FALSE (to STDOUT)
   TRUE means Violated
   FALSE means NOT Violated
3. Select **Set** to select the predefined block policy scripts.
4. (Optional) Select **Edit** to modify the script.
   You can change the policy level content, but the script template cannot be changed once defined.
5. (Optional) Use **Clear** button to clear the content of the script.
6. Use the **Additional Script Parameters** field to capture additional parameters for the execution of the script.

For the detailed implementation, see Multi-line Block Policy Example.

Compare with Matching Block from Reference or Previous Configuration

When defining a multi-line block policy, you can specify that the policy compares the current host configuration to the previously captured configuration or to content saved as a reference configuration. Content is compared block by block. This procedure describes how to set up the policy to compare content to either a reference or the previously captured configuration.

NOTE
A reference or previous configuration must exist for the device when testing the policy; otherwise, a Policy Status of Untestable will result.

To compare content with reference or previous configuration

1. On the Create NCM Block Policy dialog with the 'Compare with Matching Block from' option that is selected, specify the type of configuration with which to compare content. The following are available options:
   - **Previous Configuration**
     Indicates that the current host configuration will be compared, block by block, to the most recent captured configuration.
   - **Reference Configuration**
     Indicates that the current host configuration will be compared, block by block, to the configuration designated as reference. For information about setting a reference configuration, see Specify a Reference Configuration.
   - **Reference or Previous Configuration**

     NOTE
     Indicates that the current host configuration is compared, block by block, to a saved configuration. First, the policy looks for a reference configuration. If a reference configuration has not been set for a particular device, then block content is compared against the previous known configuration.

     NOTE
     If a reference or previous configuration does not exist for the device when testing the policy, a Policy Status of Untestable will result.
2. (Optional) Do the following steps to specify a block identifier.
   The block identifier is used to match corresponding blocks between two configurations. You can pick out specific text from within a block and can use it as a block identifier. For example, to compare interfaces that are labeled "interface Loopback*n*" between two configurations, you must identify 'interface Loopback.*' as the block identifier.
   If a block identifier is not specified, the first line of the block is used as the block identifier. This default is sufficient usually to identify the matching block between two configurations.
   a. Select **Advanced**.
      The Specify Block Identifier dialog opens.
   b. Specify the **Block** Identifier and whether the value is Text or RegEx.
      The following is an example of a regular expression that will match corresponding lines that begin with "interface":
      ```
      (?m)^interface .*
      ```
      The following is an example of a regular expression that represents "interface *name*", where only the name of the interface (as opposed to the entire line) will be used to match corresponding blocks:
      ```
      (?m)^interface ([a-z||A-Z||0-9||/]*)
      ```
      NOTE
      When using regular expressions, regular expression capturing groups are leveraged to pick out the block identifier. This is an advanced regular expression concept. Capturing Group 1 is used as the block identifier when using regular expressions. In this example, Group 1 is ([a-z||A-Z||0-9||/]*), which identifies the name of the interface.

      For more information about using text and regular expressions in multi-block policies, see About Blocks.
   c. Select OK.
      The Block Identifier dialog closes.

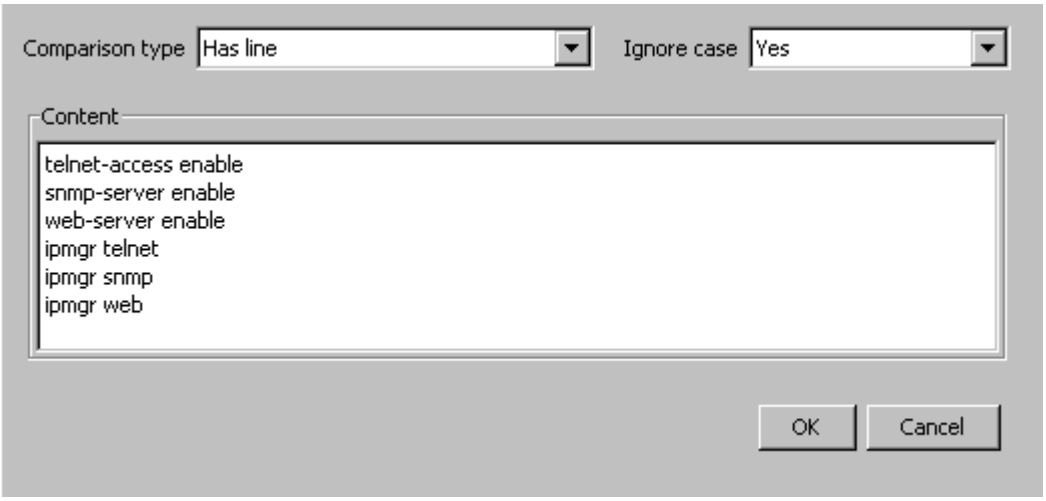
Policy Criteria Dialog

This procedure describes how to complete the Policy Criteria dialog, which is used for defining comparison criteria for single line and multi-line block policies. The content that you specify will be checked and compared every time a device's host configuration file is captured. The Policy Criteria dialog is invoked from the Create NCM Policy dialog.

To define criteria using the Policy Criteria dialog

1. Select a comparison type for the policy. Available comparison types are:
   - **Has line**
     Indicates that the host configuration file contains all lines that are specified. If met, the policy is compliant and passes.
   - **Does not have line**
     Indicates that the host configuration file does not contain the lines that are specified. If met, the policy is compliant and passes.
   - **Contains**
     Indicates that the host configuration file contains these words or symbols. If met, the policy is compliant and passes.
   - **Does not contain**
     Indicates that the host configuration file does not contain these words or symbols. If met, the policy is compliant and passes.
   - **Contains regular expression**
     Indicates that the host configuration file matches these regular expressions. If matched, the policy is compliant and passes.
   - **Does not contain regular expression**
     Indicates that the host configuration file does not match these regular expressions. If not matched, the policy is compliant and passes.
2. Specify whether to ignore upper or lowercase for the content that you enter.

   NOTE
   This setting is not available when using regular expressions.
3. Select in the **Content** box and enter content (full line, sub-string, or regular expression). The following is an example:



4. Select **OK**.
   The **Policy Criteria** dialog closes and you return to the **Create NCM Policy** or**Create NCM Block Policy** dialog, where the new criteria appears in the table.

Recommended Upload for Corrective Action

The setup for recommended upload for corrective action differs slightly between single line and multi-line block policies. This section describes how to configure corrective action according to the type of policy you have defined. This section contains the following topics:

Single Line Policy Corrective Action

The recommended upload for corrective action for a single line policy involves specifying content that once merged into the running configuration makes the device compliant with the policy. This procedure describes how to set up this content.

To enter corrective action for a single line policy

1. Select Edit under the Recommended Upload for Correction Action group.

   NOTE
   You can also select Open to import content from a text file.

   The Edit Corrective Action dialog opens.
2. Enter one or more lines that repair a non-compliant device. This is the content that once merged into the running configuration makes the device compliant with this policy.
3. Select OK.
   The Edit Corrective Action dialog closes and the corrective lines are displayed.

Multi-line Block Policy Corrective Action

The recommended upload for corrective action for a multi-line block policy involves specifying content that once merged into the running configuration makes the device compliant with the policy. Because block policies by nature deal with multiple blocks or occurrences of non-compliant data, you must set up the corrective action to handle this accordingly. This procedure describes how to set up this content.

To enter corrective action for a multi-line block policy

1. Select 'Repeat for each violating block' if you want the corrective action to be effected for each block where a violation occurs. If unchecked, the corrective action is uploaded as-is for the first violating block only.
2. Select Edit under the Recommended Upload for Correction Action group.

   NOTE
   You can also select Open to import content from a text file.

   The Edit Corrective Action dialog opens.
3. Enter one or more lines that repair a non-compliant device. This is the content that once merged into the running configuration makes the device compliant with this policy. Use the Insert Extracted Content button to insert the <extracted_text> tag into your corrective action, which will be replaced by block-specific content when the policy runs. The following shows an example corrective action:

   ```
   interface <extracted_text>
   description "policy violation detected on <extracted_text> by Spectrum"
   !
   ```

   WARNING
   The repair text must be a valid and complete device configuration statement, especially when the repair action is repeated. For example, if the "!" is omitted from the end of the previous example, the corrective action may not be implemented properly, and unexpected results may occur. This is because the statement is not ended correctly: a description needs to end with a new line character or a new line with a "!" character.
4. Select Configure Extracted Content.
   The Edit Extracted Content dialog opens.
5. Enter content to be extracted from each block, and select whether it is Text or a regular expression (RegEx).

- If text, this value is inserted wherever the <extracted_text> tag is found in the corrective action.
- If regular expression, the value that is returned from evaluating the regular expression will be inserted wherever the <extracted_text> tag is found in the corrective action.

The following is an example of a regular expression that represents "interface *name*":

```
(?m)^interface ([a-z||A-Z||0-9||/]*)
```

Using this example, the policy extracts the name of the interface from each block and will insert it into the corrective action.

For more information about using text and regular expressions in multi-block policies, see Multi-line Block Policies.

6. Select OK.
   The Edit Corrective Action dialog closes and the corrective lines are displayed.

View Violations

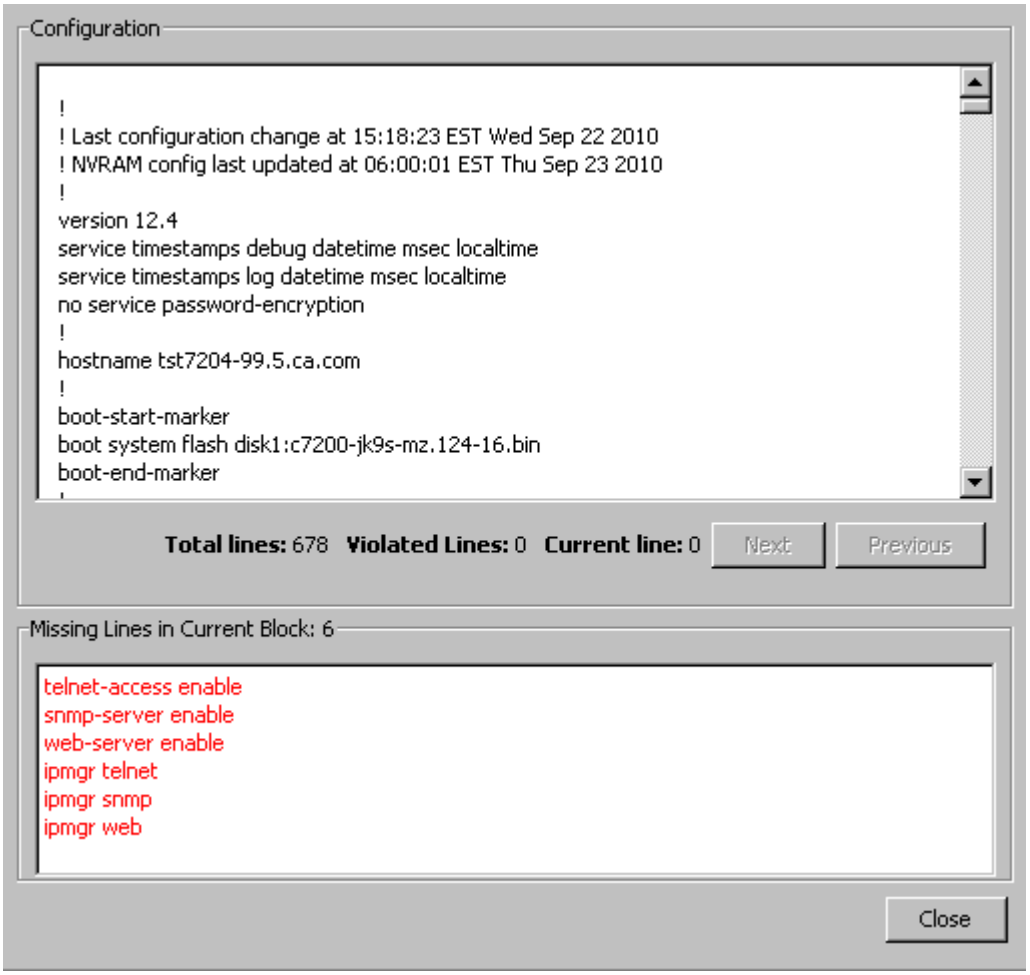
When a device is non-compliant, a View Violations dialog provides information as to the reason why. The dialog that is invoked and the information presented varies based on the policy definition. This section contains the following topics:

- Single Line Policy Violations
- Multi-line Block Policy Violations

Single Line Policy Violations

Violations for all single-line policies are displayed in the View Violation dialog.

The following example shows certain required commands that are missing in the configuration for the device and thus the policy is violated.



The View Violation dialog for all single-line policies contains the following information:

- **Configuration**
  Displays the captured host configuration in its entirety with any violated lines highlighted.
- **Total lines**
  Provides the total number of lines in the configuration file.
- **Violated Lines**
  Provides the total number of lines that violate the policy.
- **Current line**
  Provides the current location within the configuration file.
- **Next**
  Allows you to quickly advance to the next violation.
- **Previous**
  Allows you to move back to the previous violation.
- **Missing Lines in Current Block: *total_number_of_lines***
  Displays those lines that are defined in the policy that were not found in the configuration file.

  NOTE
  For single line policies, there is only one block.

Multi-line Block Policy Violations

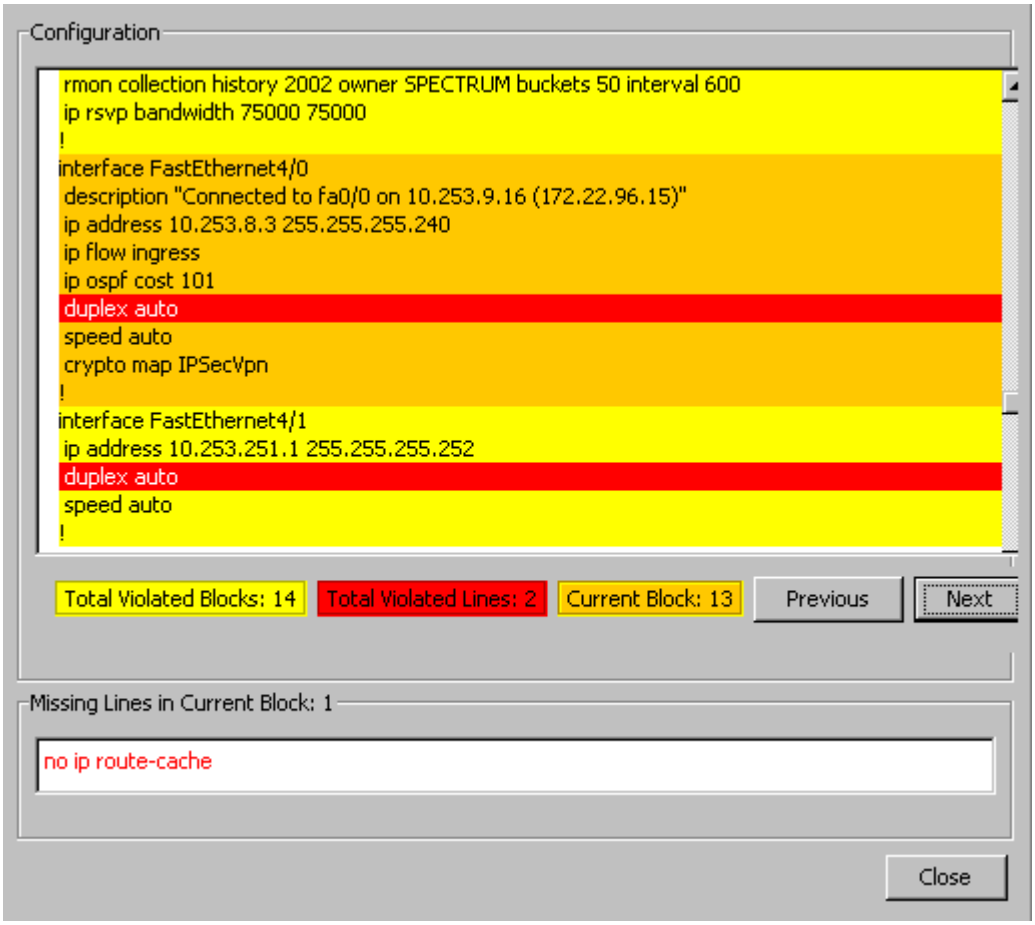
There are two types of criteria that can be used for comparison in a multi-line block policy: user-defined criteria and content from a saved configuration. Because of this, the view violation dialog that appears will be different depending on the violation content to be displayed.

Violations when Compared with Specific Contents

When user-defined criteria is used for comparison in a multi-line block policy, violations are shown on the View Violation Blocks dialog.

The following are examples of View Violation Blocks dialogs for multi-line block policies where the current host configuration is compared to user-defined criteria.

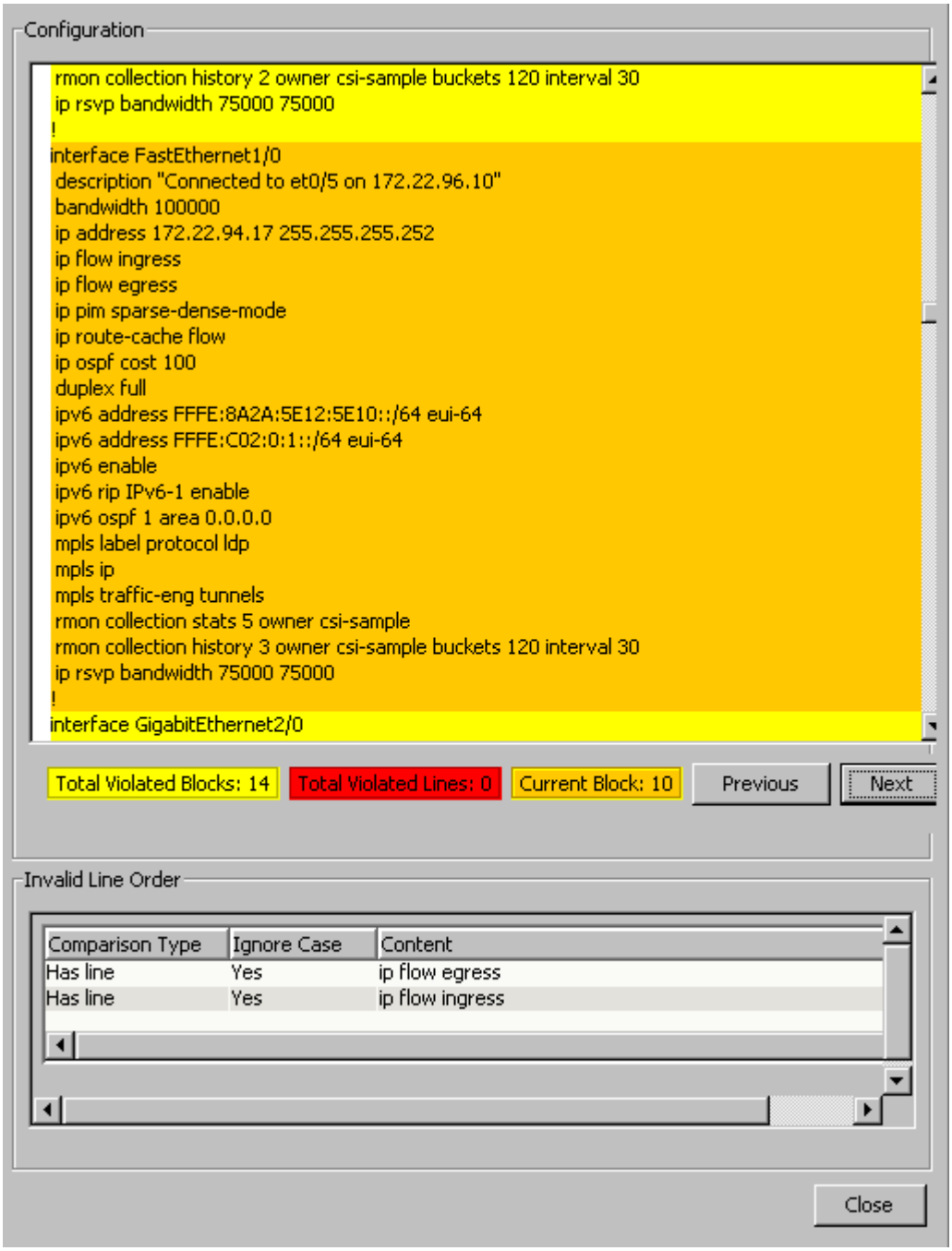
In this example, a policy has been set up to check that 'duplex auto' is not present and 'no ip route-cache' is present for each interface configuration. The violations are identified as follows:



In the next example, a policy has been set up to so that a configuration is compliant if the following commands appear and that they appear in the following order:

```
ip flow egress
ip flow ingress
```

The current configuration violates this policy because although the commands appear, they are not in the correct order, as shown in the following image:



The View Violation Blocks dialog may contain the following information, depending on the violation:

- **Configuration**
  Displays the captured host configuration in its entirety with any violated lines highlighted:
  - **Red --** These lines contain violations.

  Blocks are distinguishable by color:
  - **Orange --** These lines constitute the current block.
  - **Yellow --** These lines are included in a block other than the current block.
- **Total Violated Blocks**

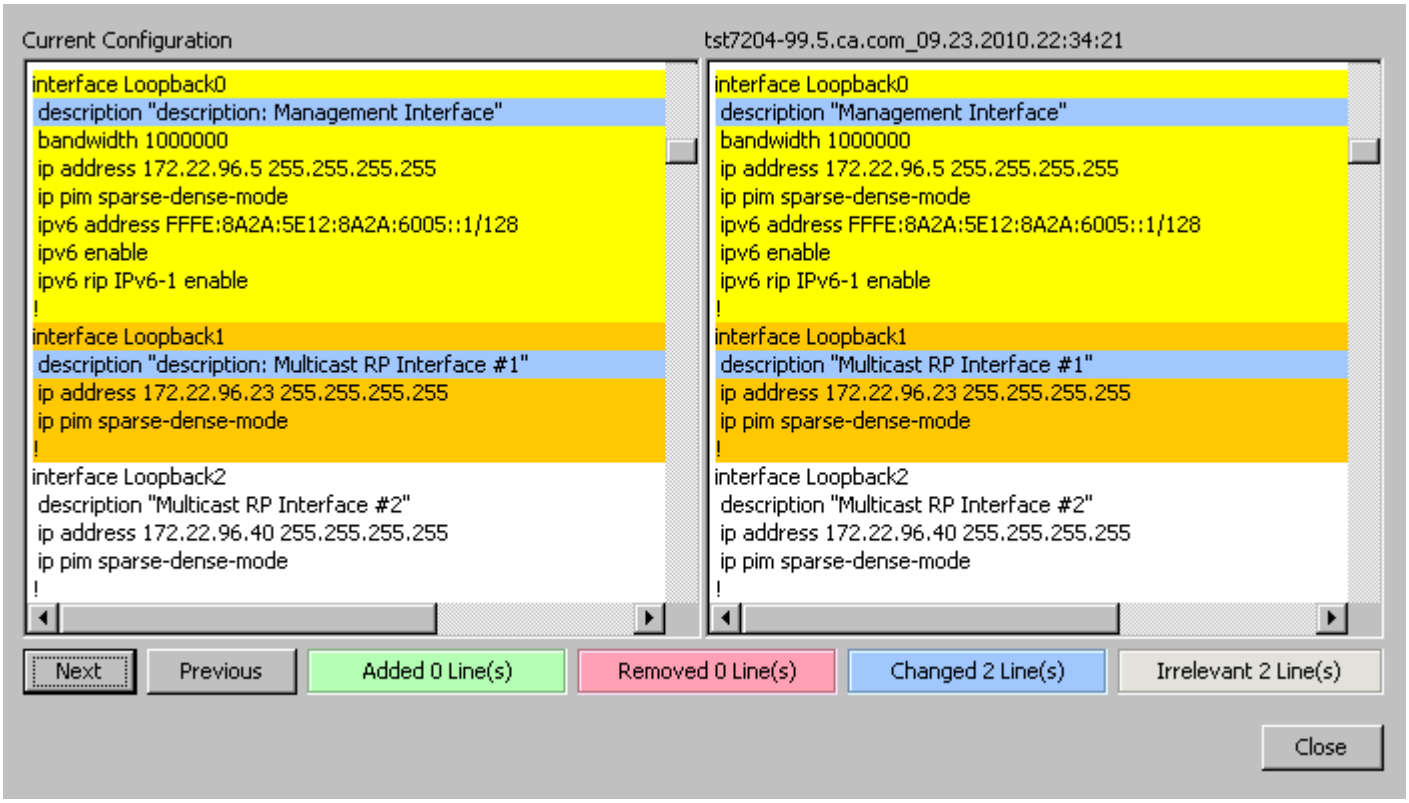
Provides the total number of blocks that contain violations.

- **Total Violated Lines**
  Provides the total number of lines that violate the policy.
- **Current Block**
  Provides the current location within the configuration file. Distinguishable blocks are numbered for identification.
- **Previous**
  Allows you to move back to the previous block containing a violation.
- **Next**
  Allows you to quickly advance to the next block containing a violation.
- **Missing Lines in Current Block: *total_number_of_lines***
  Displays those lines that are defined in the policy that were not found in the configuration file.
- **Invalid Line Order**
  Displays content criteria that has been violated due to its order of appearance in the configuration file.

Violations when Compared with Another Configuration

When a saved configuration is used for comparison in a multi-line block policy, violations are shown on the View Policy Violation dialog.

The following is an example of a View Policy Violation dialog for a multi-line block policy where the current host configuration is compared to a reference configuration and lines have changed; thus, the policy is violated.



The current host configuration is in the left pane and the reference configuration is displayed in the right. Differences between the two configurations are highlighted according to the following key.

Blocks containing differences are highlighted in their entirety and distinguishable by color:

- **Yellow --** These lines constitute a block where a violation occurs.
- **Orange --** These lines constitute a block where a violation occurs.

Individual lines denoting differences are identified as follows:

- **Green --** These lines were added.
- **Red --** These lines have been removed.
- **Blue --** These lines have changed.
- **Grey --** These lines differ but are outside of a qualifying block.

Select Next or Previous to navigate through the differences in the file.

Repair Non-Compliant Devices

In addition to repairing non-compliant devices when setting up policies, you can also initiate repair of non-compliant devices after violations occur. This section contains the following topics:

- Repair Non-Compliant Devices from the Policy Table
- Repair Non-Compliant Devices from a Policy Violation Alarm

Repair Non-Compliant Devices from the Policy Table

You can check and repair policies from the policy table by selecting a device, device family, or global collection. For example, you can repair a non-compliant device.

Follow these steps:

1. Select a device, device family, or global collection that has a configured policy in the **Explorer** tab.
2. Select the **Information** tab in the **Contents** panel.
   Information about the device, device family, or global collection appears.
3. Expand **Network Configuration Policies**.
   The **Network Configuration Policies** table appears. Policies with non-compliant devices have a non-zero value in the **Violators** column.
4. Select a policy that has a non-compliant device, and select the 'Launch repair dialog' icon.
   The **Repair Devices in Violation** dialog appears.
5. Select the **Content** tab to view the content to be uploaded to perform the repair.
6. Select**View Violation** to view the violation of each device.
7. Select **Repair**.
   The **Creating Task** status box appears. The **Upload Task Results** dialog shows the results of the operation.

   NOTE

   You can automate and minimize the above process by running a Jar executable through AlarmNotifier. Contact DX NetOps Spectrum support team to get the executable file and the steps to use it.

Repair Non-Compliant Devices from a Policy Violation Alarm

You can view a violation and can upload or merge the correct content to the device to make it compliant with the policy from the **Alarm Details** tab. Repair a non-compliant device directly from a policy violation alarm.

Follow these steps:

1. Select a device, device family, or global collection that has a configured policy, or a policy (from the Policy node) in the **Explorer** tab.
2. In the **Alarms** tab in the **Contents** panel, select an alarm with “NCM Policy Violated” in the **Alarm Title** column.
3. Select **View Violation Details** in the **Alarm Details** tab in the **Component Detail** panel.
   The**Repair Devices in Violation** page opens.
4. Select **Content** to view the content to be uploaded to perform the repair. Select **View Violation** to view the violation of each device.

The **View Violation** page appears.

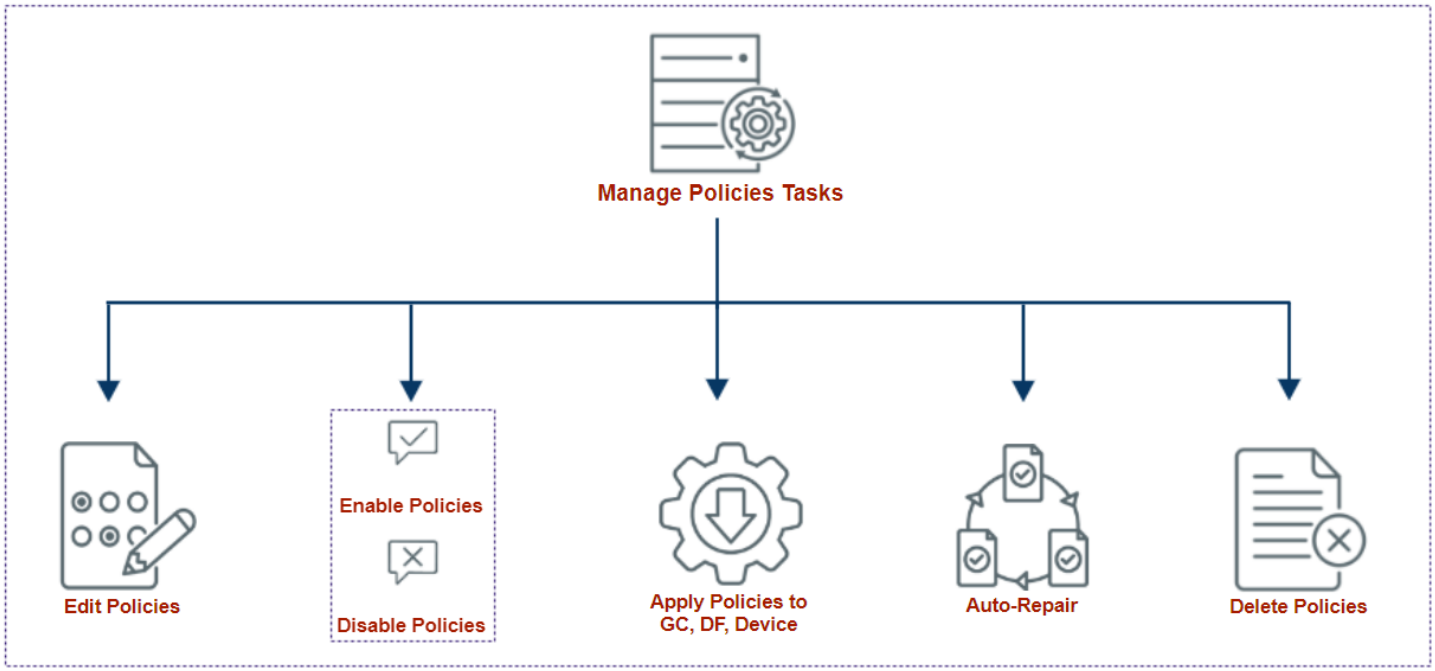
5. Select **Repair**.
   The **Creating Task** status box appears, followed by the **Upload Task Results** page.

Manage Policies

After a policy has been created, you can perform various operations on it. For example, you can edit, enable, disable, or delete it. You can also apply it to devices, device families, or global collections.

Manage Policies Tasks

The following diagram provides an overview of the tasks that help you manage your policies:



Edit Policies

You can edit an existing Network Configuration Manager policy. After you edit a policy, you must save and enable it.

To edit a policy

1. Select a policy under the **Policies** node in the **Explorer** tab.
2. Select the **List** tab in the **Contents** panel.
   A list of policies appears.
3. Select the policy and click the Edit selected policy icon in the toolbar.
4. Select **Yes** on the message confirmation dialog.
   The **Edit NCM Policy** dialog appears.
5. Make changes as necessary and click **Save**.
   The policy is disabled. Enable the policy as explained in Enable and Disable Policies from the Policy Table.

> **NOTE**
> Optionally, for a global collection, you can edit a policy by selecting the global collection, clicking the **Information** tab, and editing the associated policy. For a device, you can select the individual device, click

the **Information** tab, click **Network Configuration Policies**, and edit the associated policy. For a device family, select a device family, click the **Information** tab, click **Network Configuration Policies**, and edit the associated policy.

Enable and Disable Policies

You can enable and disable Network Configuration Manager policies from the policy table.

Follow these steps:

1. Select a policy under the **Policies** node in the **Explorer** tab.
2. Select the **List** tab of the **Contents** panel.
   A list of policies appears.
3. Select a policy and click the 'Enable selected policies' icon to enable the policy.
   Enabling a policy causes any specified alarms to immediately appear for all non-compliant devices and violated policies.
4. (Optional) Select a policy and click the 'Disable selected policies' icon to disable that policy.
   Disabling a policy immediately clears any existing alarms on non-compliant devices and violated policies.

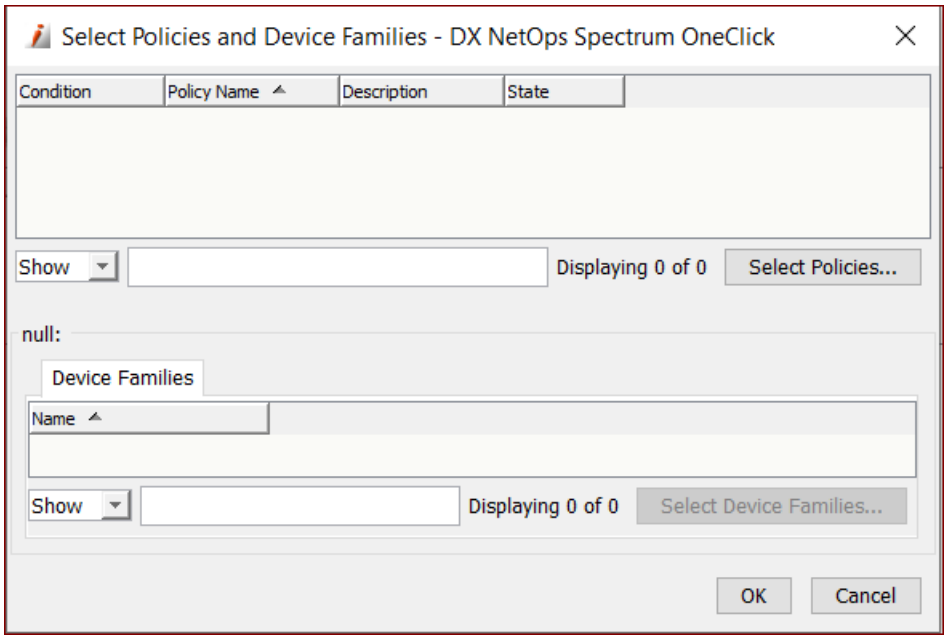
   NOTE
   You can also enable and disable a policy by selecting a global collection, device family, or device. Use the **Information** tab to manage the associated policy.

Apply Multiple Policies to a Global Collection

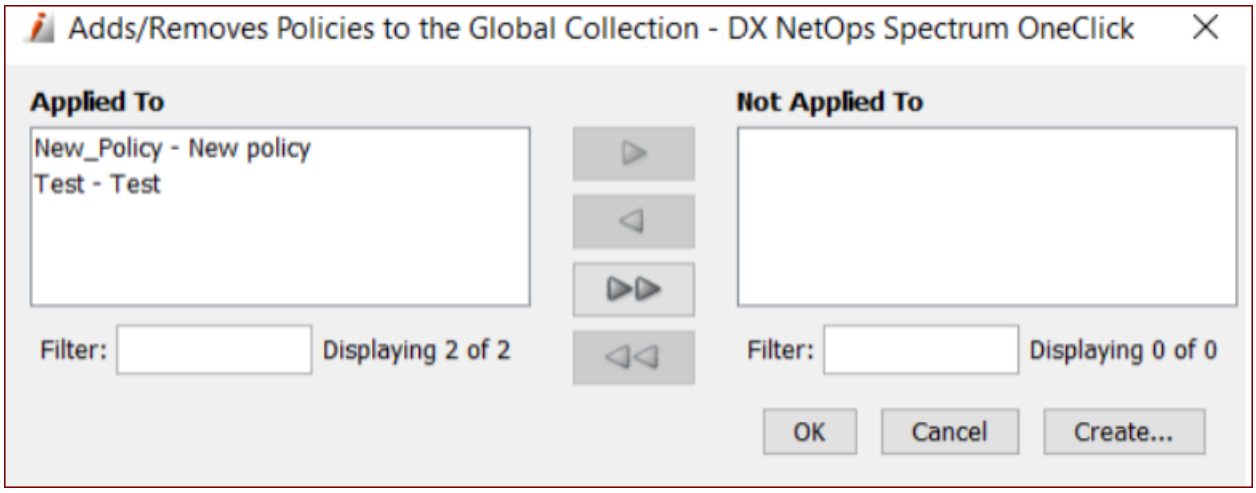
After policies are created, they can be applied to a global collection. When the policies are applied to a global collection, the policies are then enforced to all the global collection members of the selected device families (in that global collection).

Follow these steps:

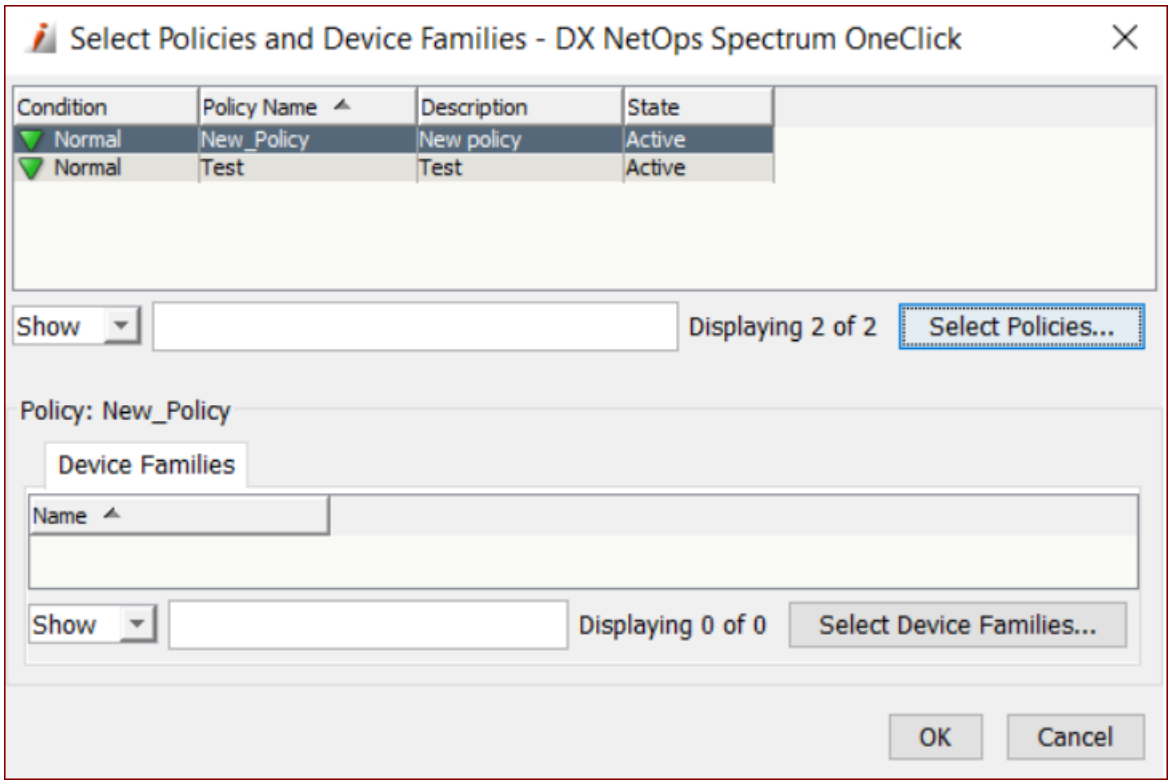
1. Select a global collection in the **Explorer** tab.
   Information for the global collection appears in the **Information** tab of the **Contents** panel.
2. Expand the **Network Configuration Policies** subview.
   The Network Configuration Policies table appears.
3. Click the 'Add/Remove policies to/from the global collection' icon.
   The **Select Policies and Device Families** dialog appears.



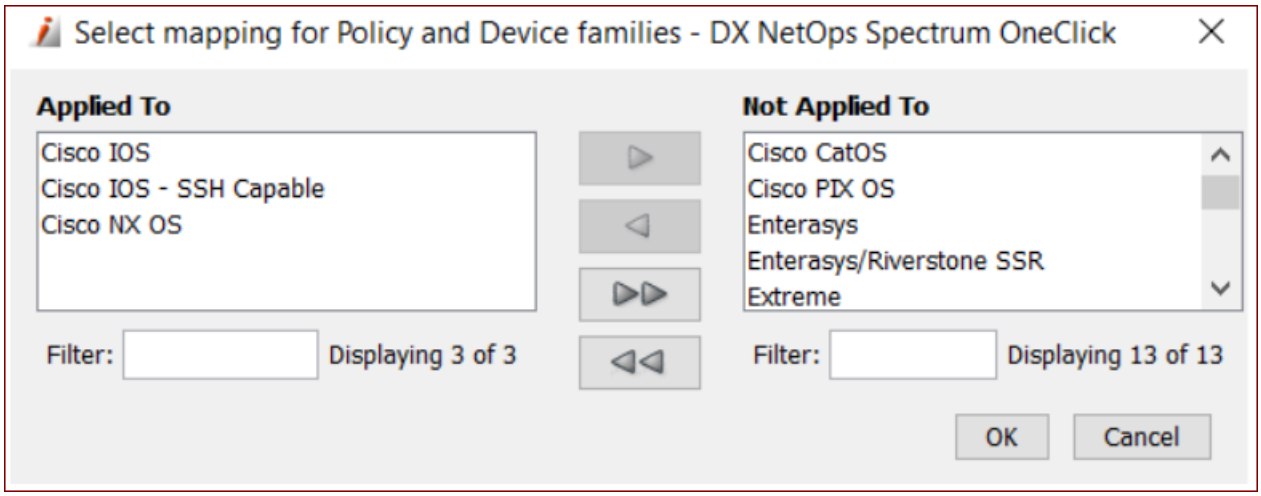
4. Click the **Select Policies** button to select the policies that you want to apply on the global collection. The **Adds/ Removes Policies to the Global Collection** dialog appears.
5. Select the required policies from the **Not Applied To** area and move them to the **Applied To** area. The following screenshot shows the required information that the two policies have been applied to this global collection:



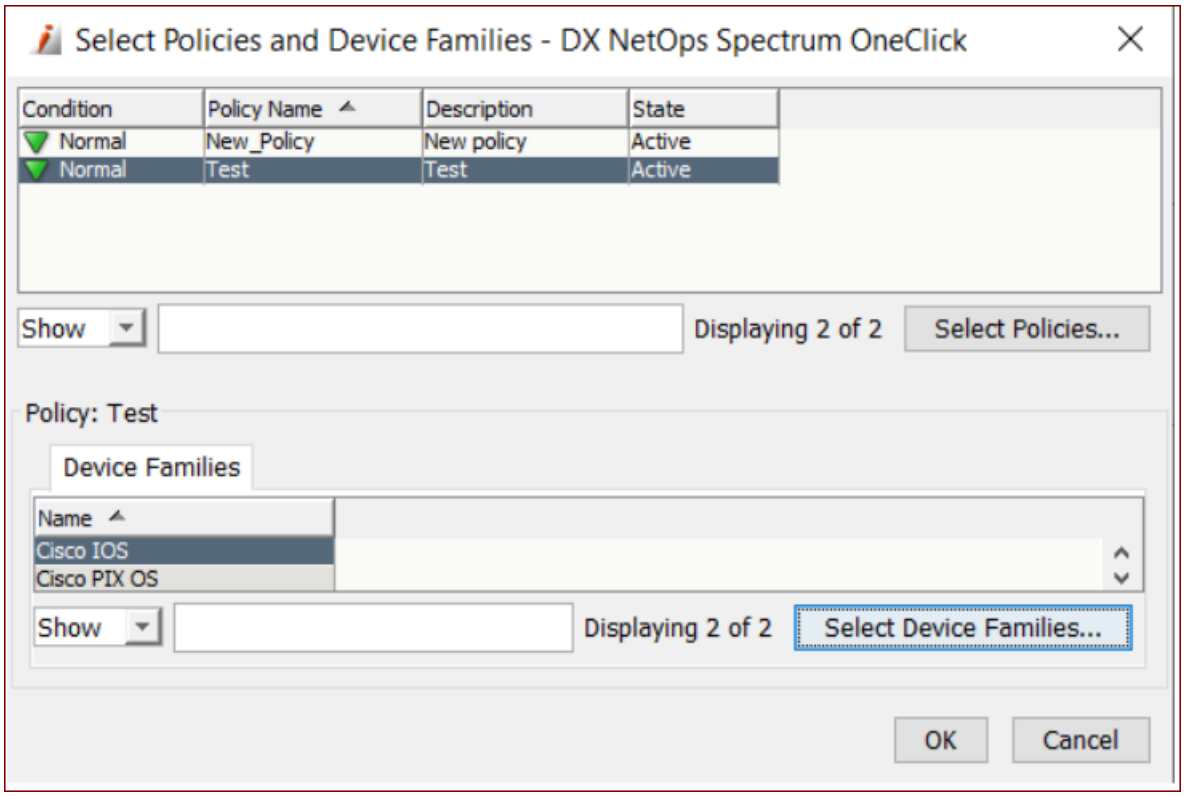
6. Click **OK**. The added policies appear in the table as follows:



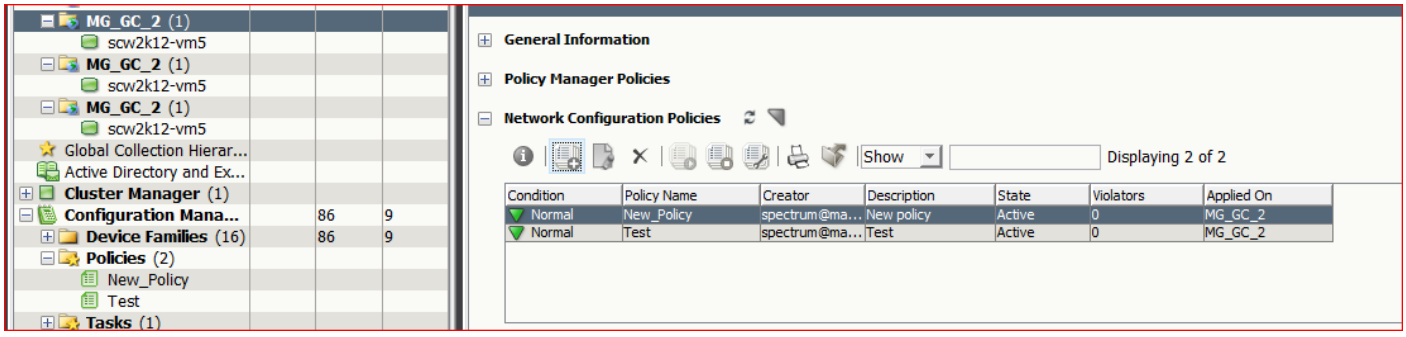
7. Select a policy from the table.
8. Click the **Select Device Families** button to select the device families that you want to associate with the selected policy.
9. Select the required device families from the **Not Applied To** area and move them to the **Applied To** area. The following screenshot shows the required information:



10. Click **OK**. The selected device families appear in the **Device Families** section of the dialog and get associated with the selected policy. The following screenshot shows the required information:



11. Click **OK**.
    The applied policies appear in the **Network Configuration Policies** table. The following screenshot shows the required information:



12. Repeat the steps to associate the remaining policies with the device families. The added policies will be enforced on all the global members based on the associated device families (in that global collection).

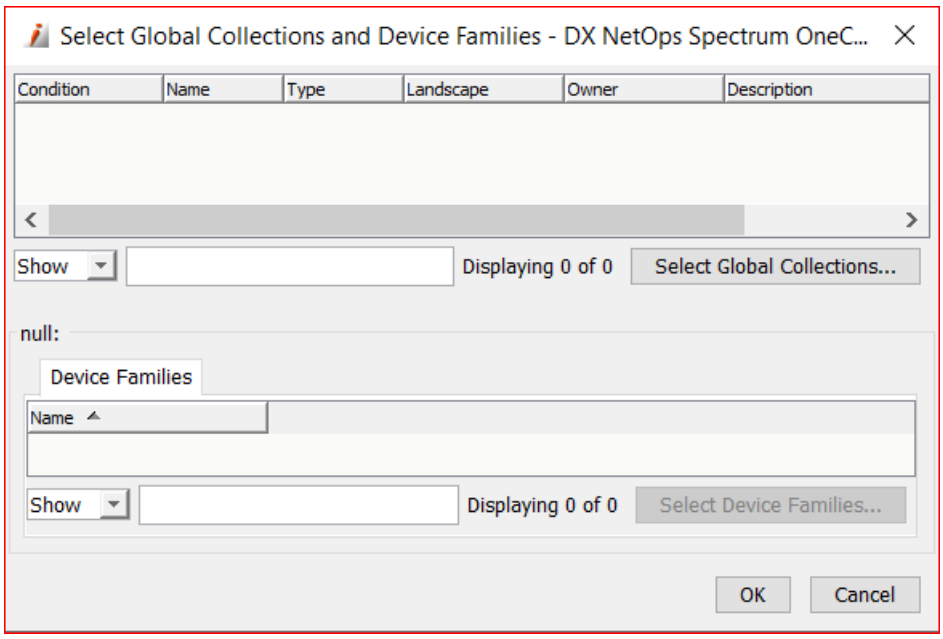
Apply a Single Policy to Multiple Global Collections

You can also apply a single policy to multiple global collections.

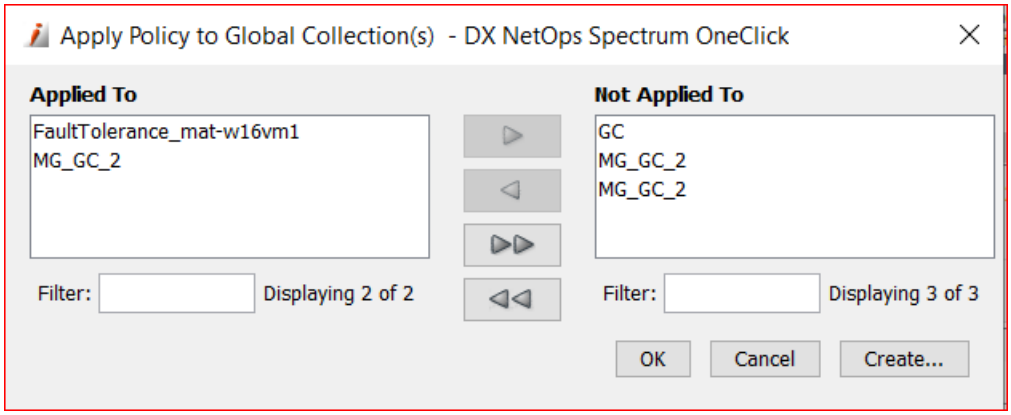
Follow these steps:

1. Navigate to the **Policies** node in the **Explorer** tab.
2. Select the **List** tab of the **Contents** panel.
   A list of policies appears.
3. Select the policy that you want to apply to multiple global collections.
4. Click the 'Apply policy to global collection(s)' icon.

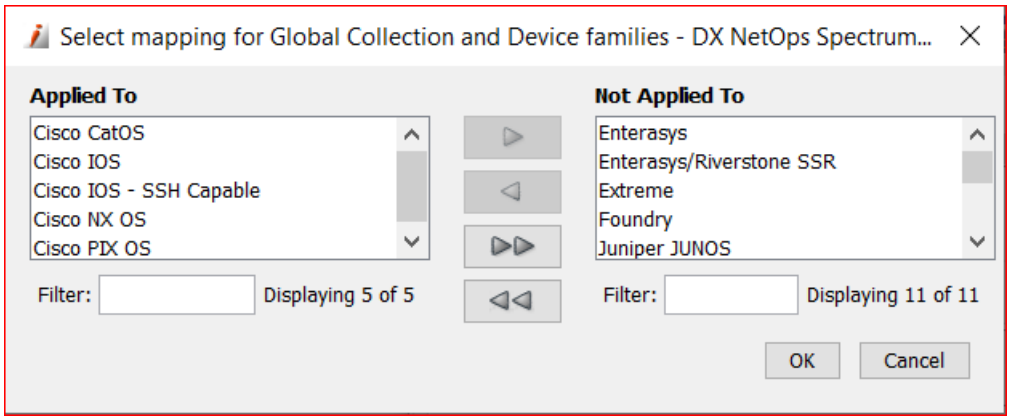
The **Select Global Collections and Device Families** dialog appears:



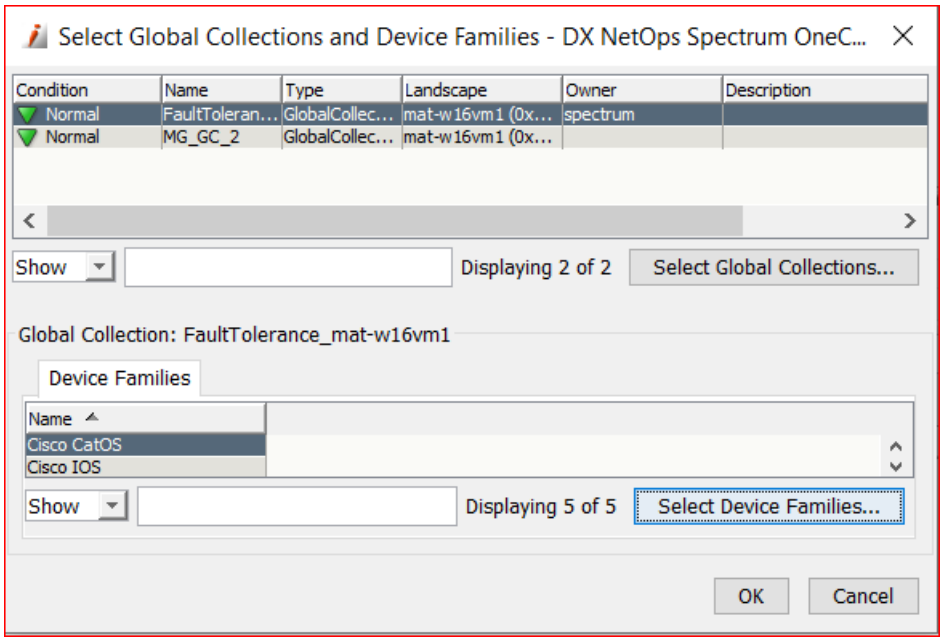
5. Click the **Select Global Collections** button to choose the global collections on which you want to apply the policy. The **Apply to Global Collection(s)** dialog appears.
6. Select the required global collections from the **Not Applied To** area and move them to the **Applied To** area. The following screenshot shows the required information:



7. Click **OK**. The selected global collections are added to the table.
8. Select the global collection in the table and then click the **Select Device Families** button. The **Select Mapping for Global Collection and Device Families** dialog appears. In this dialog, you select the required device families that you want to associate with the policy selected for the global collection.
9. Select the required device families from the **Not Applied To** area and move them to the **Applied To** area. The following screenshot shows the required information:



10. Click **OK**. The selected device families are added to the **Device Families** section of the dialog:



11. Follow the same steps to select device families for other global collections that you have added.
12. Click **OK**.

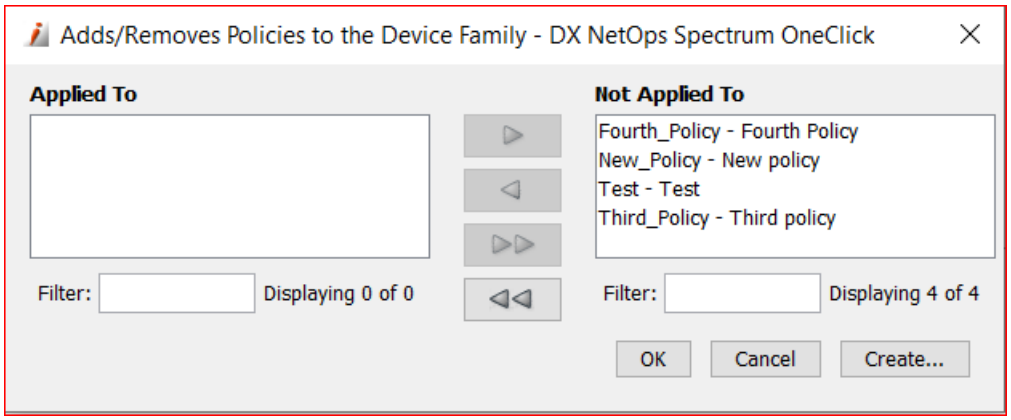
You have successfully applied a policy to multiple global collections. You have also selected the required devices families for the global collections. Therefore, the policy is applied to only those device families in the global collection, not to the other device families in that global collection.

Apply Multiple Policies to a Device Family

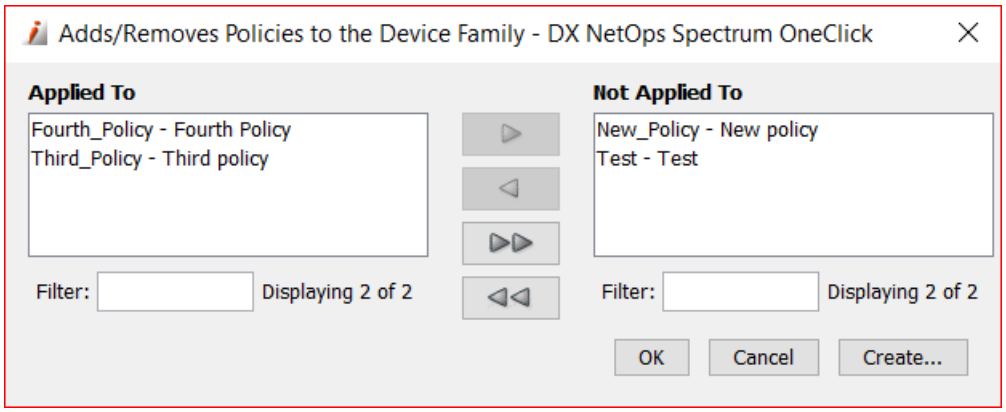
After policies are created, they can be applied to a device family. When multiple policies are applied to a device family, the policies are then enforced on all members of the selected device family.

Follow these steps:

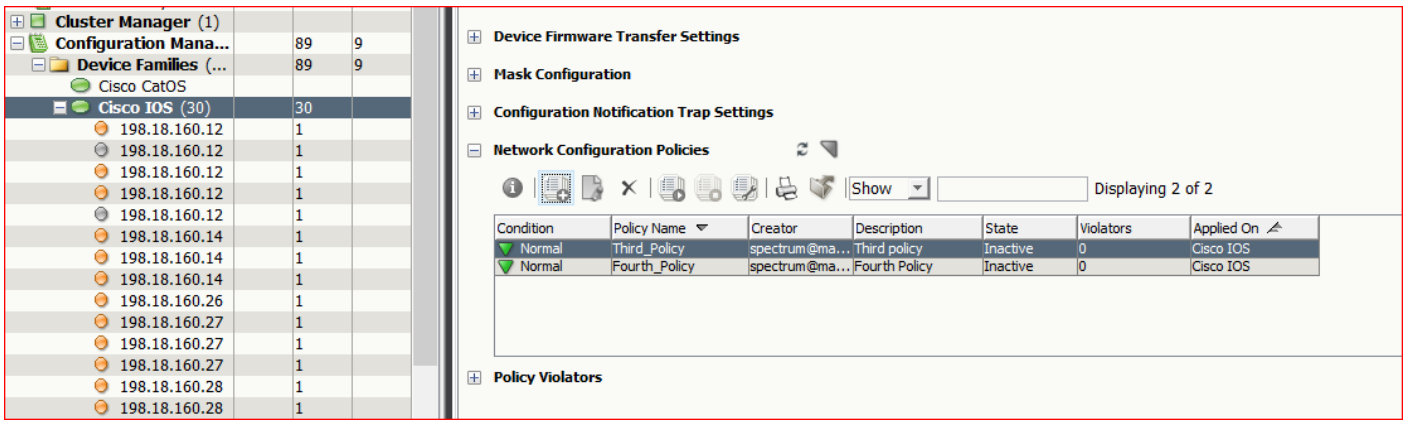
1. Select a device family in the **Explorer** tab.
   Information for the device family appears in the **Information** tab of the **Contents** panel.
2. Expand the **Network Configuration Policies** subview.
   The **Network Configuration Policies** table appears.
3. Click the 'Create policy' icon.
   The **Adds/Removes Policies to the Device Family** dialog appears:



4. Select the required policies (which you want to apply to the device family) from the **Not Applied To** area and move them to the **Applied To** area:



5. Click **OK**.
   The applied policies appear in the **Network Configuration Policies** table and will be enforced on all the members of the selected device family.



Apply a Single Policy to Multiple Device Families

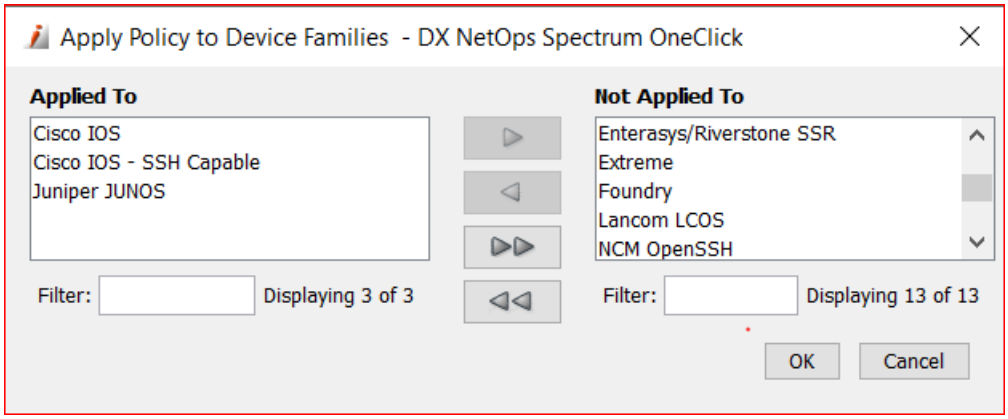
Similarly, you can apply a single policy to multiple device families.

Follow these steps:

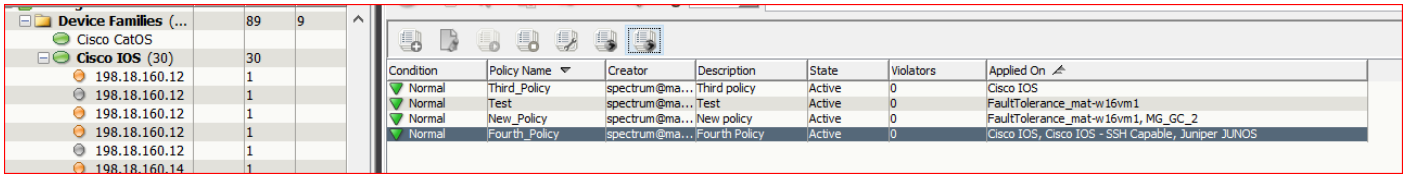
1. Navigate to the **Policies** node in the **Explorer** tab.
2. Select the **List** tab of the **Contents** panel.

A list of policies appears.

3. Select the policy that you want to apply to multiple device families.
4. Click the 'Apply policy to device families' icon. The **Apply Policy to Device Families** dialog appears. In this dialog, you select all the device families to which you want to apply the selected policy.
5. Select the required device families from the **Not Applied To** area and move them to the **Applied To** area.



6. Click **OK**. The policy is associated with the selected families.



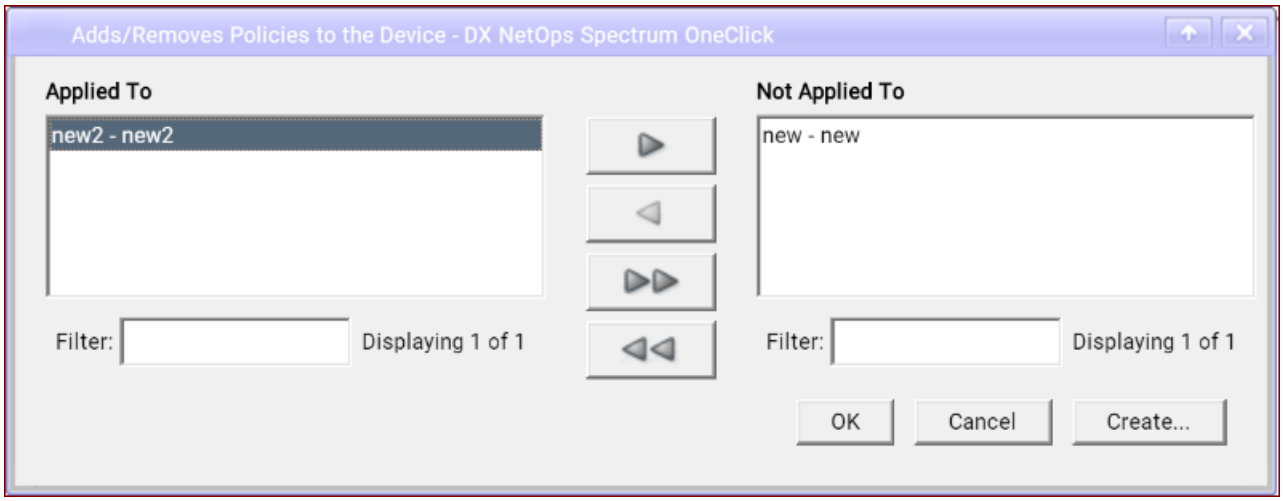
You have successfully applied a single policy to multiple device families.

Apply a Policy to an Individual Device

You can also apply a policy to an individual device.

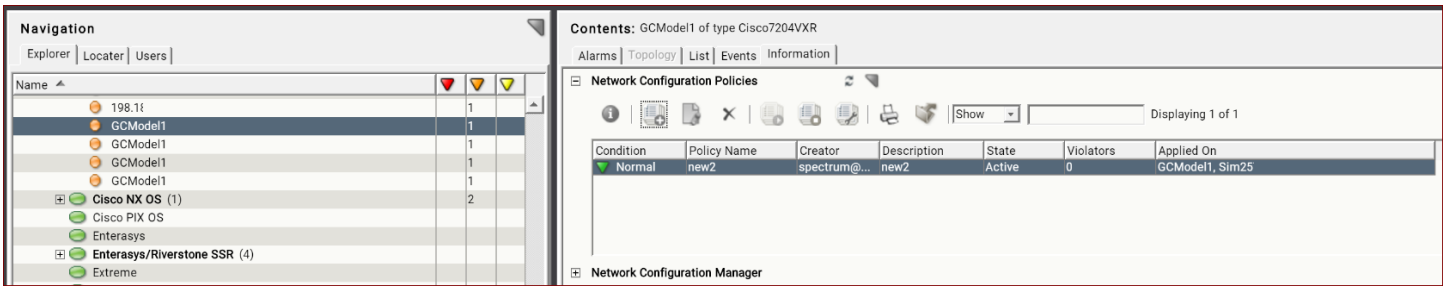
Follow these steps:

1. Select a device to which you want to apply a policy in the **Explorer** tab.
2. Click the **Information** tab in the **Contents** pane.
3. Expand the **Network Configuration Policies** subview.
4. Click the Create policy icon.
5. Move the required NCM policy that you want to apply from the **Not Applied To** area to **Applied To** area.



6. Click **OK**.

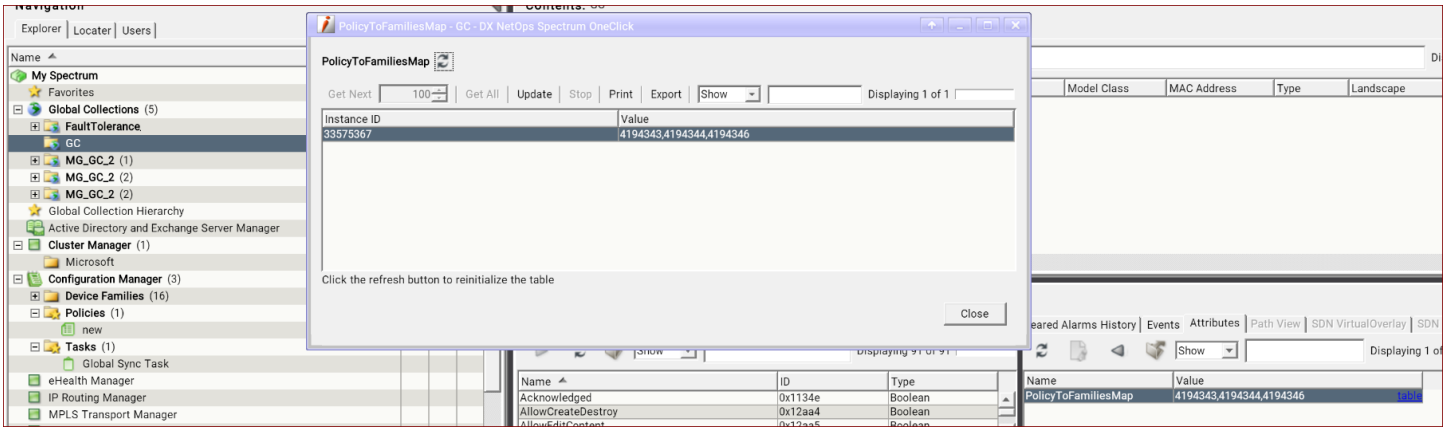
The policy is added to the Network Configuration Policies table.



If you see the **Applied On** column in this table, you can find the device on which you have applied the policy is also listed there. If you review the same table for other devices in this family, you will find that the policy is not applied to them; it is applied only to the selected device.

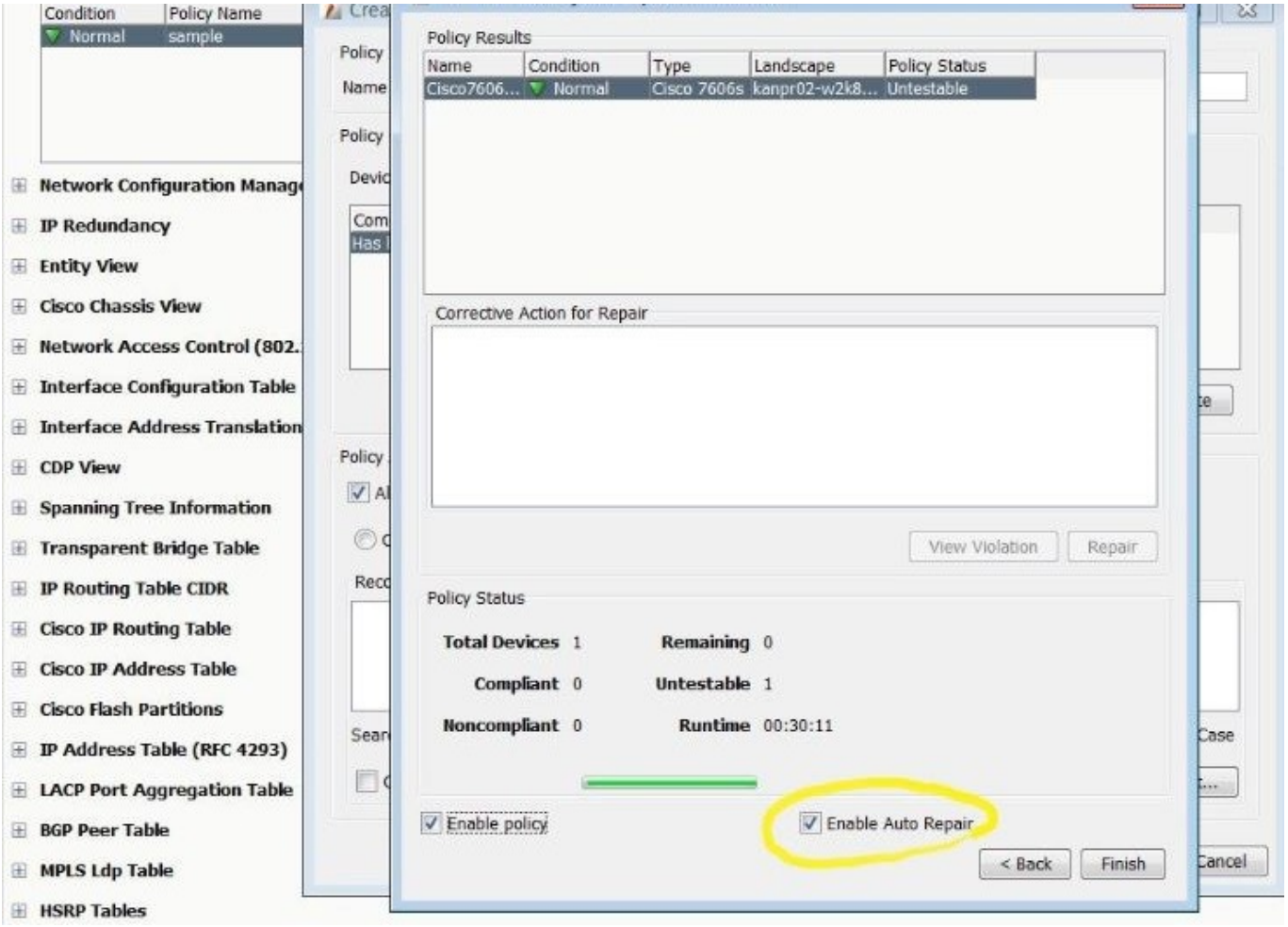
PolicyToFamiliesMap Attribute in Global Collection

The PolicyToFamiliesMap attribute stores mapping between the applied NCM policy (key) and the list of device families (value) in global collection. In each entry, the key is the home model handle of the NCM policy applied to that global collection. The value is the home model handles of the device families (comma separated). The following screenshot shows the required information:



Auto-Repair Functionality

Previously, Alarm Notifier's set script triggered the auto repair functionality and repaired all the violated policies configurations. From 10.3, auto-repair functions work independent of the Alarm Notifier. Select the check box next to the **Enable Auto Repair** option on the NCM Policy dialog. Here is a screenshot displaying the new Auto-Repair functionality for NCM devices.



Delete Policies

To delete a policy that you no longer need, right-click the policy under **Policies** in the **Explorer** tab and select **Delete**.

> **NOTE**
> Optionally, you can delete a policy by selecting a global collection, clicking the **Information** tab, and deleting the associated policy. You can also select an individual device or device family, click the **Information** tab, click **Network Configuration Policies**, and delete the associated policy.

Troubleshooting

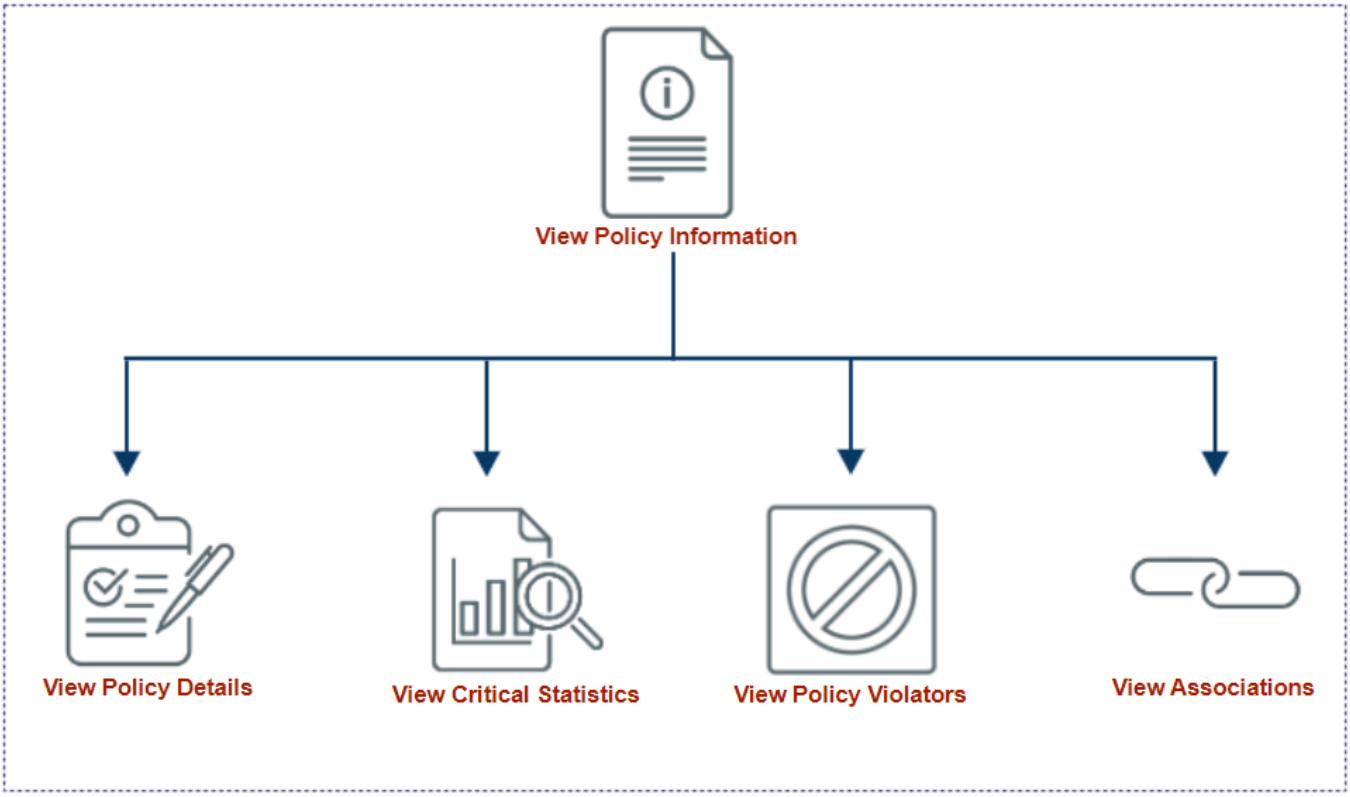
Review the following troubleshooting information:

- If a policy is violated, then the violation alarms are asserted on the device model, policy model, or both. If the same policy is not applied, then the corresponding alarms are cleared.
- If a policy that is applied to a global collection is violating for a device under that global collection, then the violation alarm is raised as expected. If the device is removed from the global collection, then the policy violation alarm is cleared. The same behavior holds good for the policy that is applied to a device family.
- If a policy is applied to a global collection and the expected violation alarm is not created on the device models, then verify that the PolicyToFamiliesMap attribute is updated properly with the policy and the families home model handles.

View Policy Information

This section describes how to view policy information and includes the following topics:

The following diagram shows the high-level tasks that help you view the information related to policies:



View Policy Details

You can view component details for Network Configuration Manager policies.

Follow these steps:

1. Select a device, device family, or global collection in the **Explorer** tab that has an associated policy.
2. Click the **Information** tab in the **Contents** panel.
   Information and configurations for the selected device, device family, or global collection appear.
3. Expand **Network Configuration Policies**, and click 'View the Component Detail for the selected model'.
   The **Component Detail** panel for the selected policy appears.

   NOTE
   You can also access this screen by selecting a policy from **Policies** in the **Explorer** tab.

View Critical Statistics for All Policies

You can view critical statistics for policies by selecting **Policies** under **Configuration Manager** in the **Explorer** tab and selecting the **List** tab in the **Contents** panel.

Statistics for all policies appear.

View Critical Statistics for All Policies Applied to a Device

You can view critical statistics for policies that are applied to a device.

Follow these steps:

1. Select a device, and then click the **Information** tab.
   Information about the device appears.
2. Select **Network Configuration Policies**.
   Statistics for all policies applied to a single device appear.

View Critical Statistics for Policies Applied to a Global Collection

You can view critical statistics for policies that are applied to a global collection.

Follow these steps:

1. Select an existing Global Collection from the **Explorer** tab.
2. Select the **Information** tab of the **Contents** panel.
   Information appears in the **Contents** panel.
3. Select **Network Configuration Policies**.
   Statistics for all policies that are applied to a collection appear.

View Critical Statistics for Policies Applied to a Device Family

You can view critical statistics for policies that are applied to a device family.

Follow these steps:

1. Select an existing device family from the **Explorer** tab.
2. Select the **Information** tab of the **Contents** panel.
   Information appears in the **Contents** panel.
3. Select **Network Configuration Policies**.
   Statistics for all policies that are applied to a device family appear.

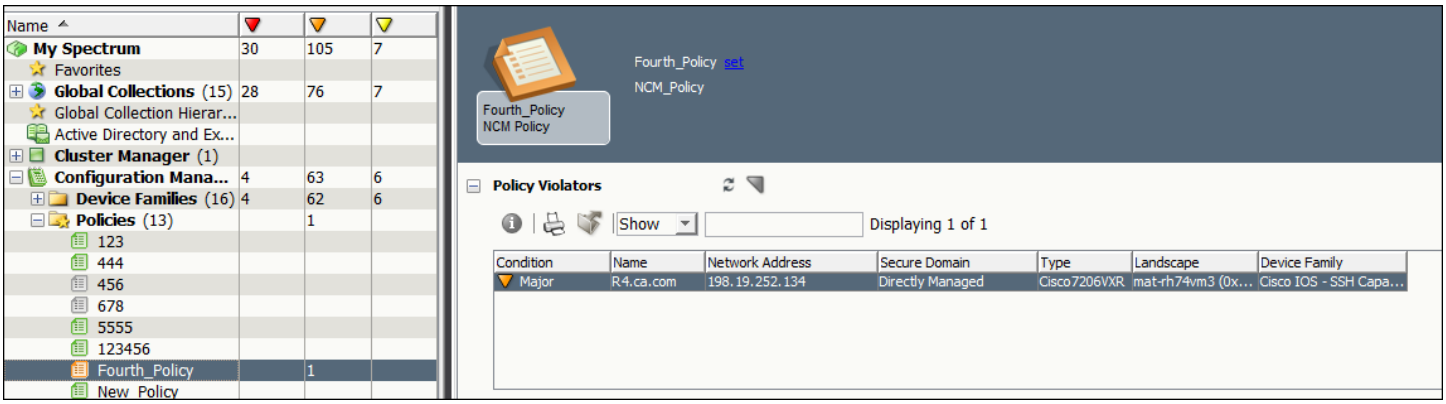
View Policy Violators

If any device violates the policy, you can view the list of such devices in the Policy Violators table at the policy level, global collection level, and at the device family level. With this information, you can quickly identify the violators and take appropriate actions.

(Policy Level) Follow these steps:

1. Navigate to the **Configuration Manager**, **Policies** node in the **Explorer** tab.
2. Select the required policy under the **Policies** node.
3. Click the **Information** tab in the **Contents** pane.
4. Expand the **Policy Violators** subview.
5. Review the list of policy violators in the table.

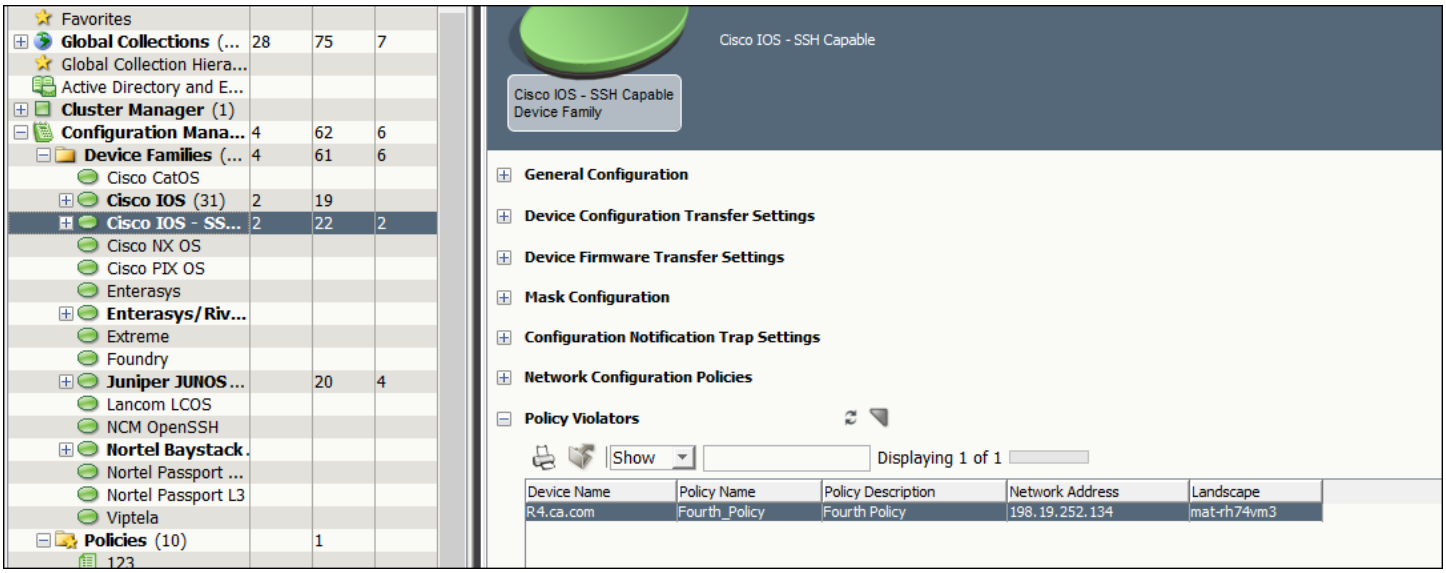
The following screenshot shows the required information at the policy level:



(Device Family Level) Follow these steps:

1. Navigate to the **Configuration Manager**, **Device Families** node in the **Explorer** tab.
2. Select the required device family under the **Device Families** node.
3. Click the **Information** tab in the **Contents** pane.
4. Expand the **Policy Violators** subview.
5. Review the list of policy violators in the table.

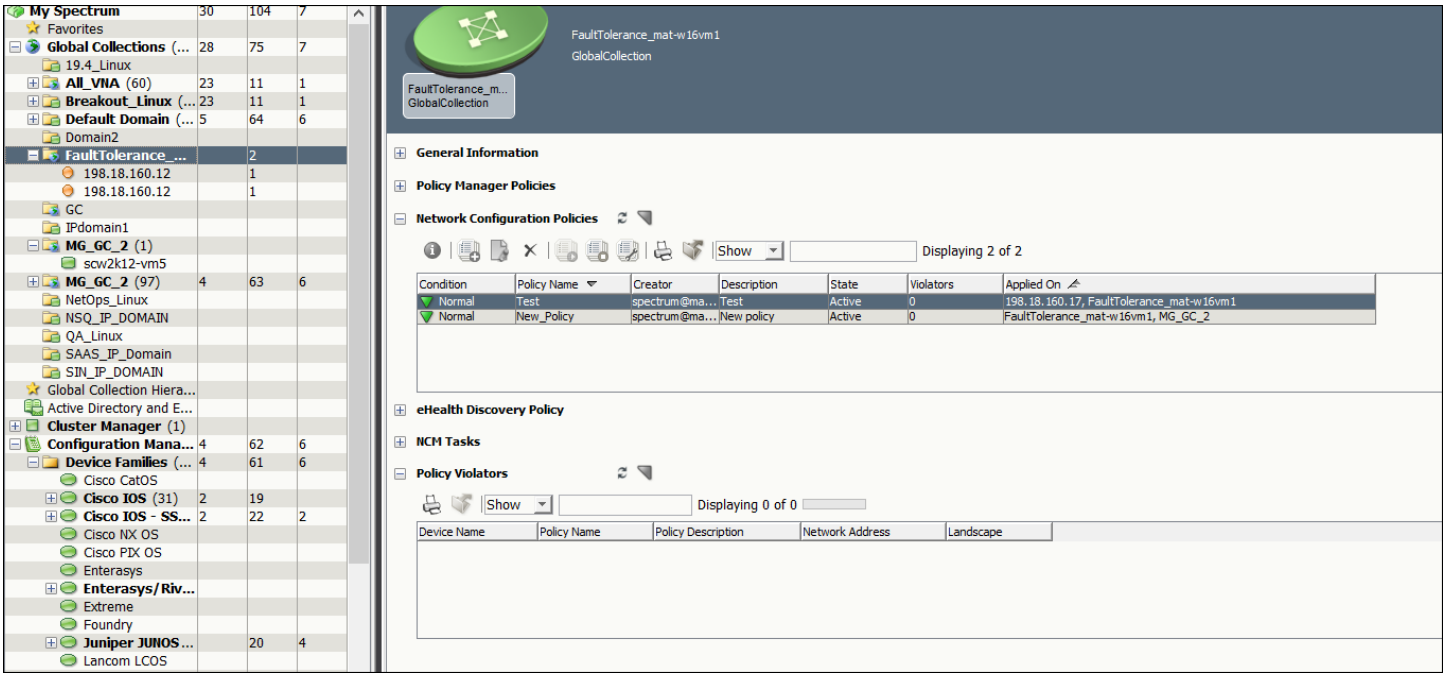
The following screenshot shows the required information at the device family level:



(Global Collection Level) Follow these steps:

1. Navigate to the **Global Collection** node in the **Explorer** tab.
2. Select the required global collection under the **Global Collection** node.
3. Click the **Information** tab in the **Contents** pane.
4. Expand the **Policy Violators** subview.
5. Review the list of policy violators in the table.

The following screenshot shows the required information at the global collection level:



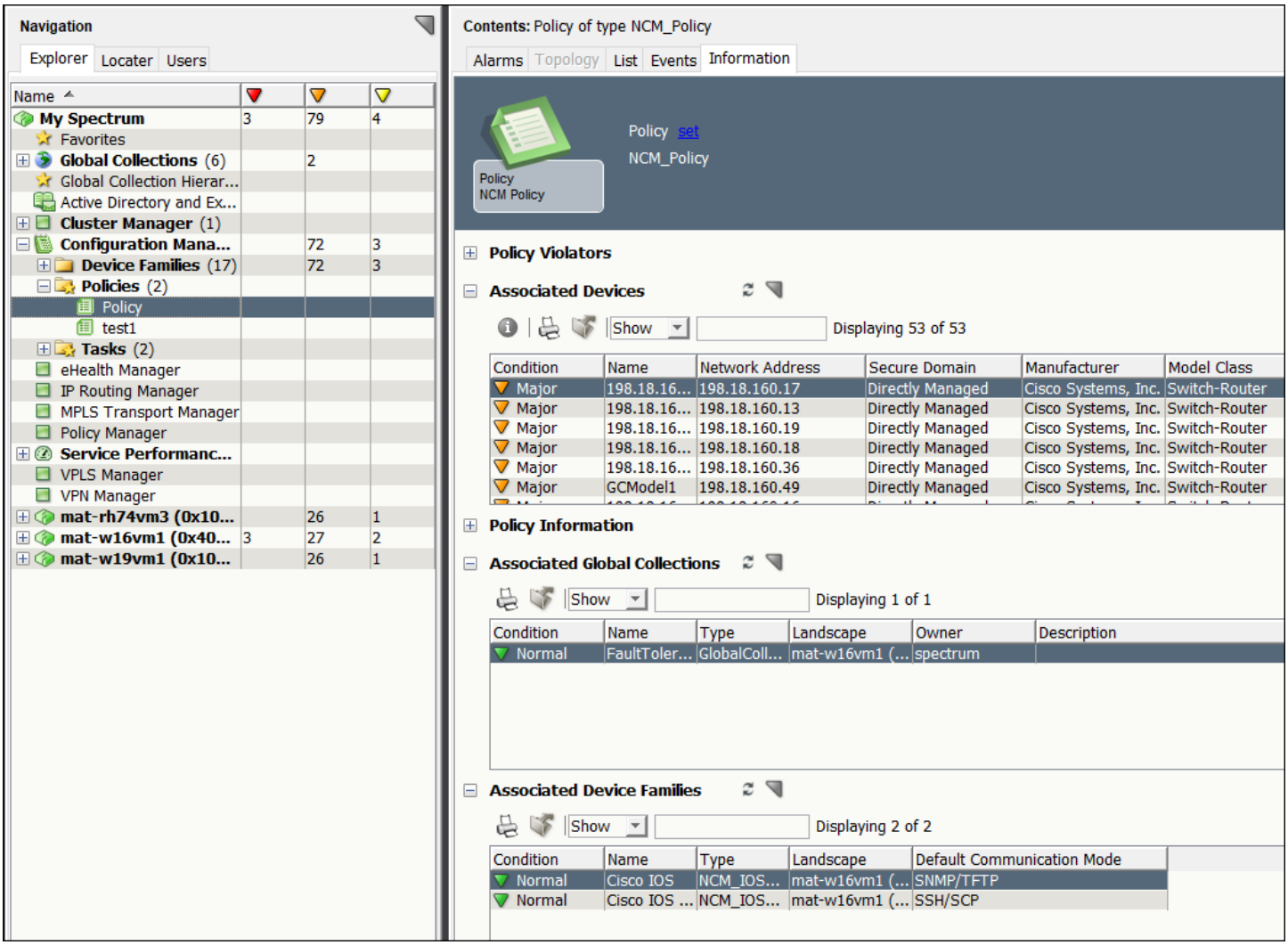
View Associated Devices, Device Families, and Global Collections

You can view the devices, device families, and global collections that are associated with a specific policy. This information helps you quickly get an overview of how the policy is associated with different items.

Follow these steps:

1. Navigate to the **Configuration Manager**, **Policies** node in the **Explorer** tab.
2. Select the required policy under the **Policies** node.
3. Click the **Information** tab in the **Contents** pane.
4. Expand the **Associated Devices, Associated Global Collections, Associated Device Families** subviews.
5. Review the list of associations in the tables, as required.

The following screenshot shows the required information:



Multi-line Block Policy Example

This section provides an example of how to use multi-line block policies. The same use case is implemented in two different ways: by comparing to specified contents, and by comparing it to another configuration.

> **NOTE**
> The content provided in this section is intended to provide a sample use case at a high level. For additional information on any of the concepts or items referenced in this section, please refer to the appropriate parent topic.

Scenario

Suppose you want to shut down certain interfaces that have been identified by the word "shutdown" appearing in their descriptions. You can identify such devices by defining multi-line block policies in the following ways:

- **By comparing to Specified Contents:**. You can search for all interfaces that do not contain "shutdown" in the description as the policy definition. This will highlight all the interfaces that *do* contain "shutdown" in the description as violators of the policy.
- B**y comparing to Specific Content with Script:** You can search for all interfaces that do not contain "shutdown" in the description as the policy definition. This will highlight all the interfaces that *do* contain "shutdown" in the description as violators of the policy.
- **By comparing it to Another Configuration:** You can monitor content by comparing newly captured configurations to a reference configuration every time a capture occurs. When "shutdown" is added to the description for an interface, it will be highlighted as a violator of the policy because it does not match the reference configuration.

After the devices are identified, the shutdown command can then be issued easily for those interfaces marked for shutdown as part of the recommended upload for corrective action.

Getting Started

Before you begin defining a policy, you must do the following:

- Identify what constitutes a block.
- Establish a reference configuration (if comparing to a reference configuration).

You can gather this information by reviewing captured host configurations for the device. Configurations are captured using the Global Sync task, the Sync task, and the Capture Configuration icon.

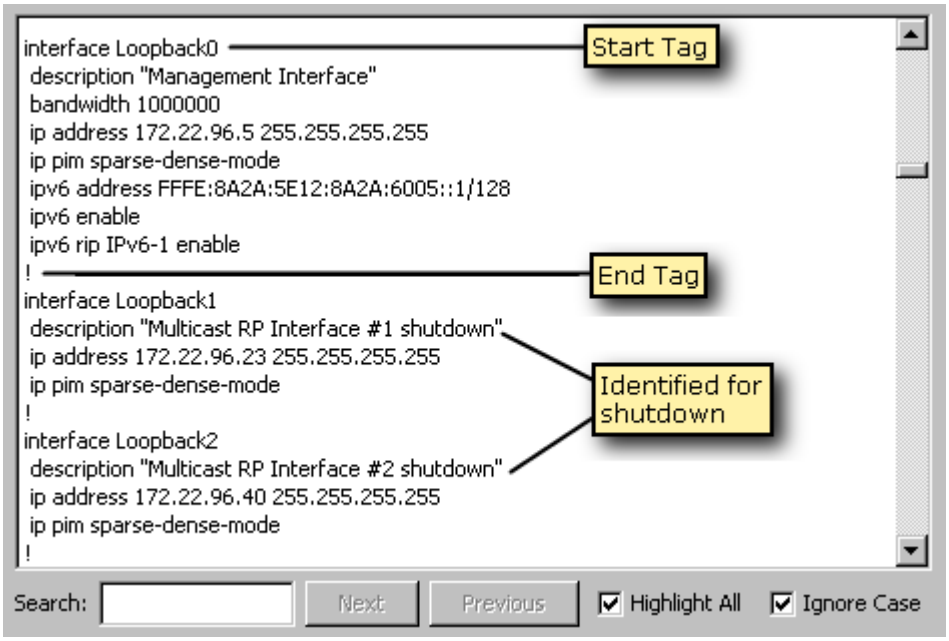
To view a captured host configuration for a device

1. Select a device in the **Explorer** tab.
2. Verify the **List** tab is selected in the **Contents** panel and select the **Host Configuration** tab in the **Component Detail** panel.
3. Click the row in the **Host Configuration** table for the captured host configuration that you want to view.
   The captured host configuration appears in the box below the table.

   NOTE
   For more information, see View Configuration History for a Single Device.

Identify What Constitutes a Block

The following image shows a portion of the configuration file for a device. You can see that each interface on this device has a similar format and is delimited by a start tag and an end tag. Also, notice the appearance of "shutdown" in the descriptions for a couple of interfaces.



In block policy terminology, each block in this example would be defined by:

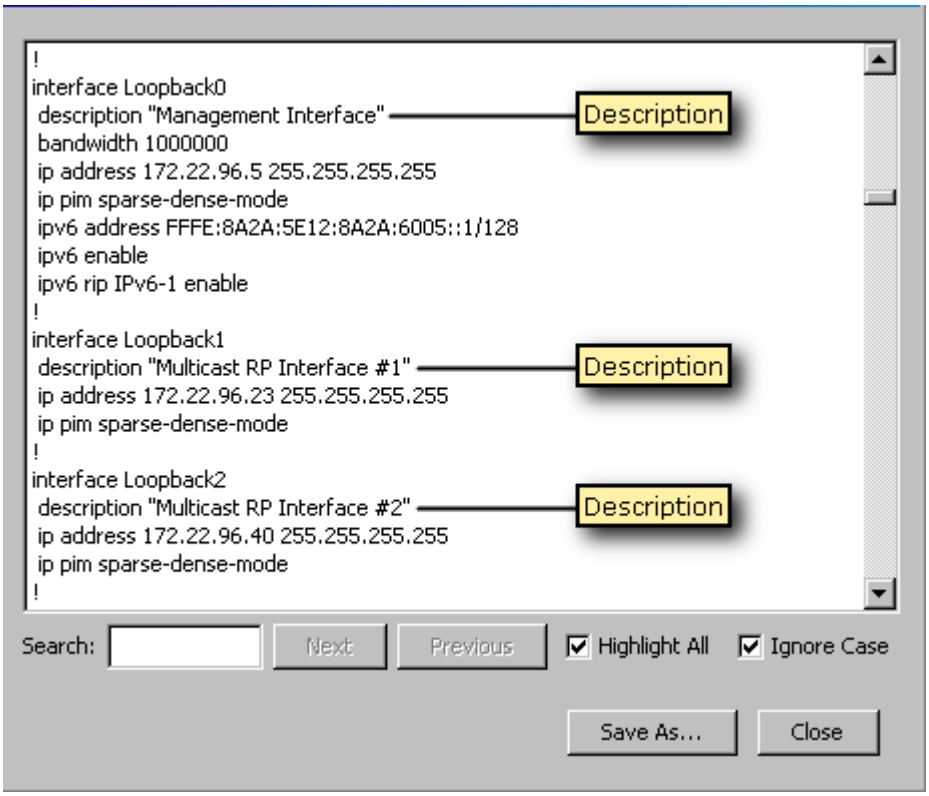
1. **Start Tag:** interface
2. **End Tag:** !

Establish a reference configuration (if comparing to a reference configuration)

Use the procedure outlined in "To view a captured host configuration for a device" to identify a host configuration that contains ideal settings for the device, and specify this configuration as its Reference Configuration. For information on setting a reference configuration, see Specify a Reference Configuration.

> **NOTE**
> You can also use the last captured configuration for comparison instead of the reference configuration.

The following image shows a portion of the configuration file that will be used as the reference configuration. Notice that this configuration does not contain "shutdown" in the description for any of the interfaces.



Defining the Policy

After you have established what constitutes a block and have specified a reference configuration (if applicable), the multi-line block policy can be defined.

Refer to the steps outlined in Create a Policy to create a multi-line block policy and subsequently invoke the Create NCM Block Policy dialog, which contains the following sections:

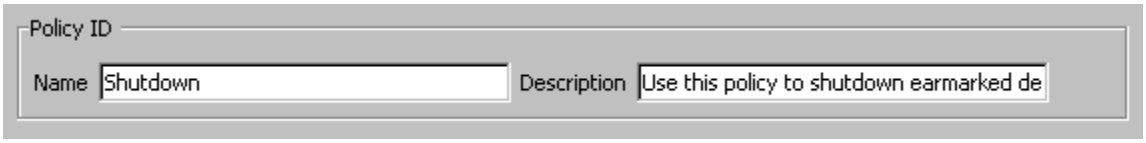
- Policy ID
- Policy Criteria
- Policy Actions

Each of these sections will be described separately.

> **NOTE**
> Additional detailed information for each of the fields mentioned in this section is available in the Create a Policy section.

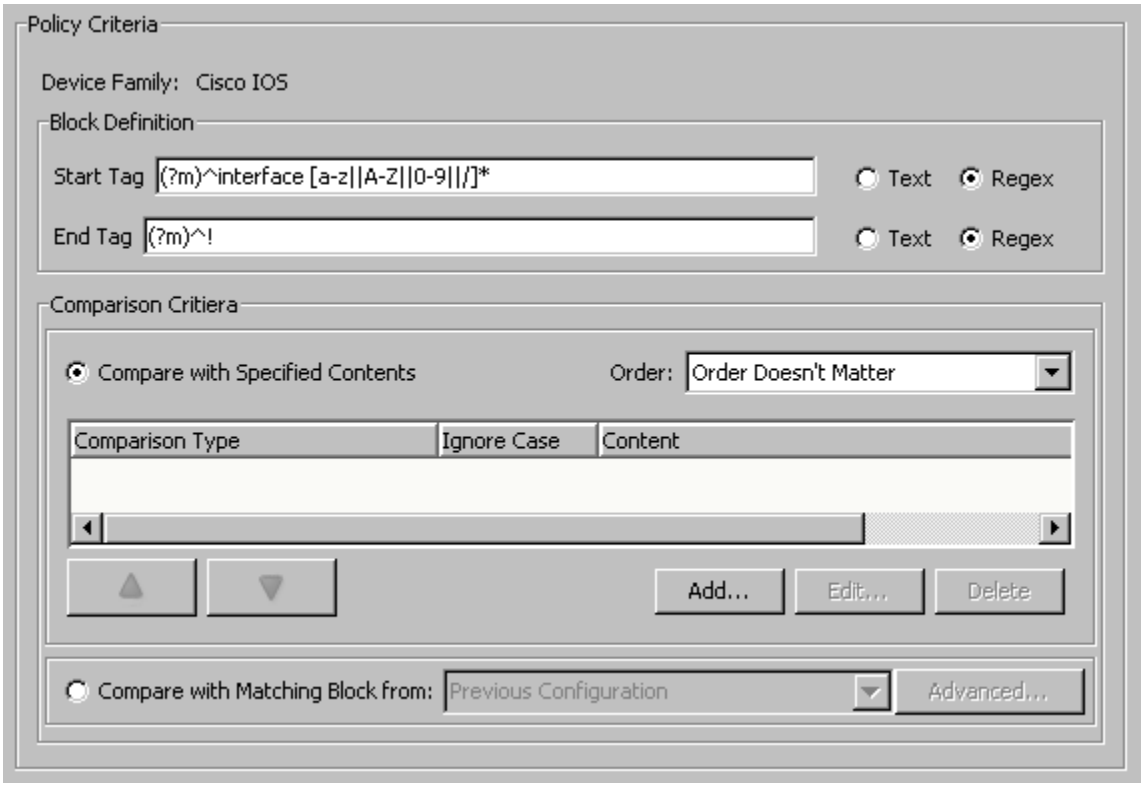
Policy ID

Policy ID information identifies the policy. Use these fields to name the policy according to standards in place at your site.



Policy Criteria

The Policy Criteria information defines the block-delimiting fields and the comparison criteria, which are described following the image. In 10.4.2, the **Device Family** option is not available. This option is applicable only for previous releases. The following screenshot is for a previous release.

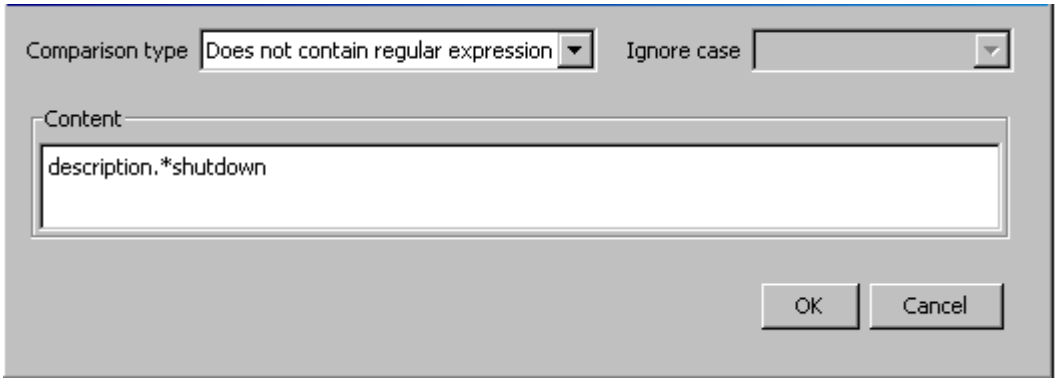


- **Block Definition.** The Start Tag and End Tag fields define strings that are used to identify the beginning and end of a block. In this example, regular expression values are used to designate that each block begins with the string "interface *name*" and ends with the character "!". These values are included as part of the block.

Comparison Criteria

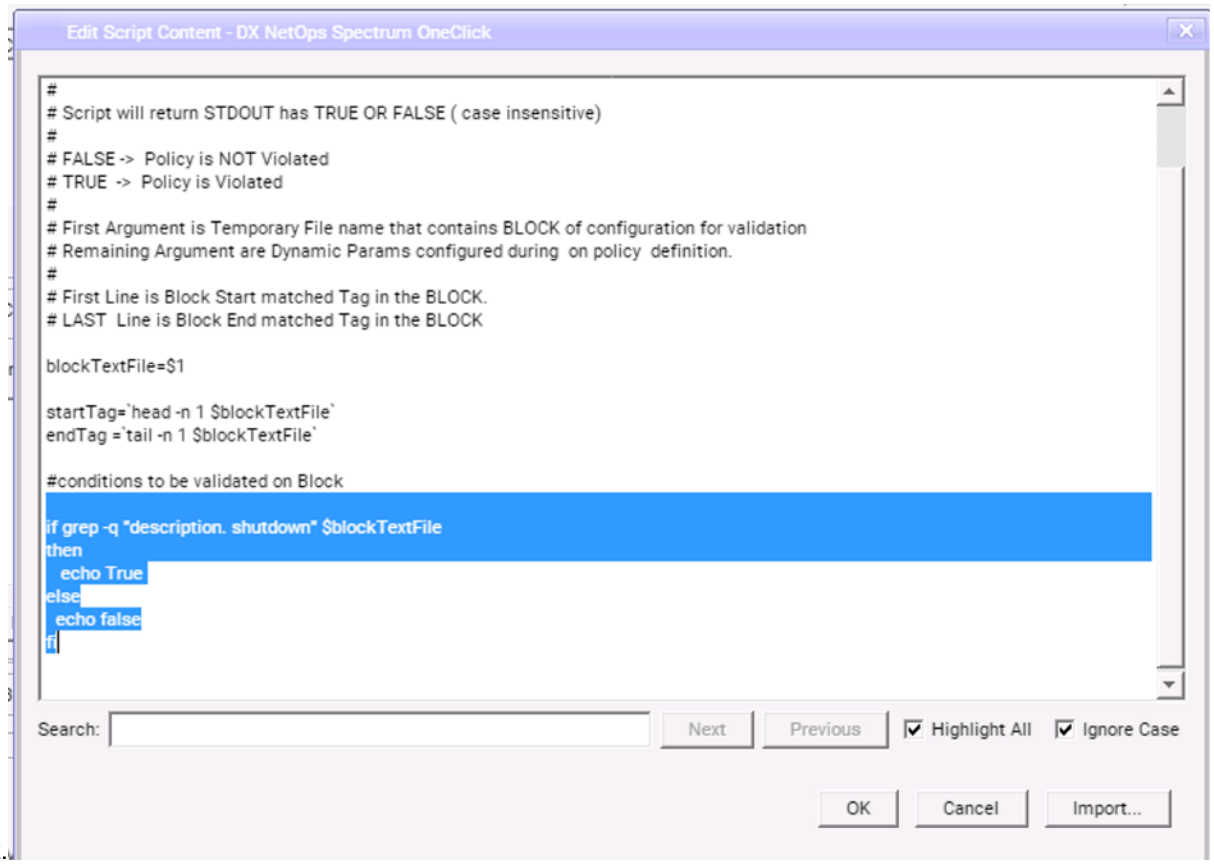
This section controls the method by which a newly captured configuration is evaluated against the policy. You can specify whether to compare the configuration against specific, user-defined criteria, specific content with a script, or against another configuration.

Compare with Specified content:



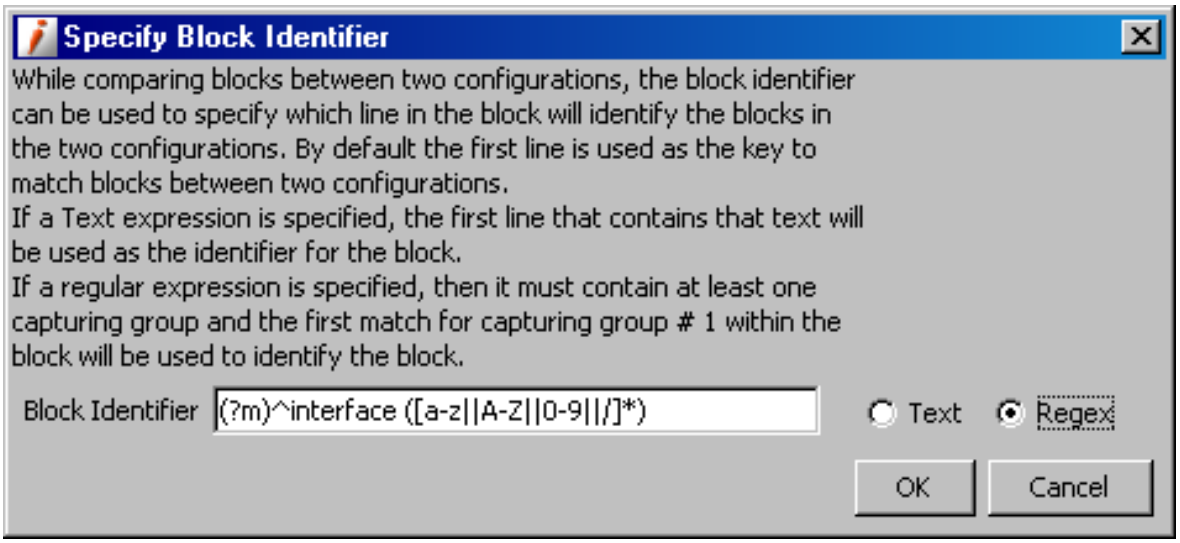
NOTE
Only one set of criteria can be included in a single policy. Both sets of criteria are shown here for demonstration purposes only.

Compare Specified content with Script:
Set the default script using the Set button and modify it to validate the condition. You can create a new script with a custom validation condition. In this example, we are modifying the script content at the policy level. In the following illustration, we can see the implementation of the same example we used for comparison with specified content, using the compare specified content with the script.



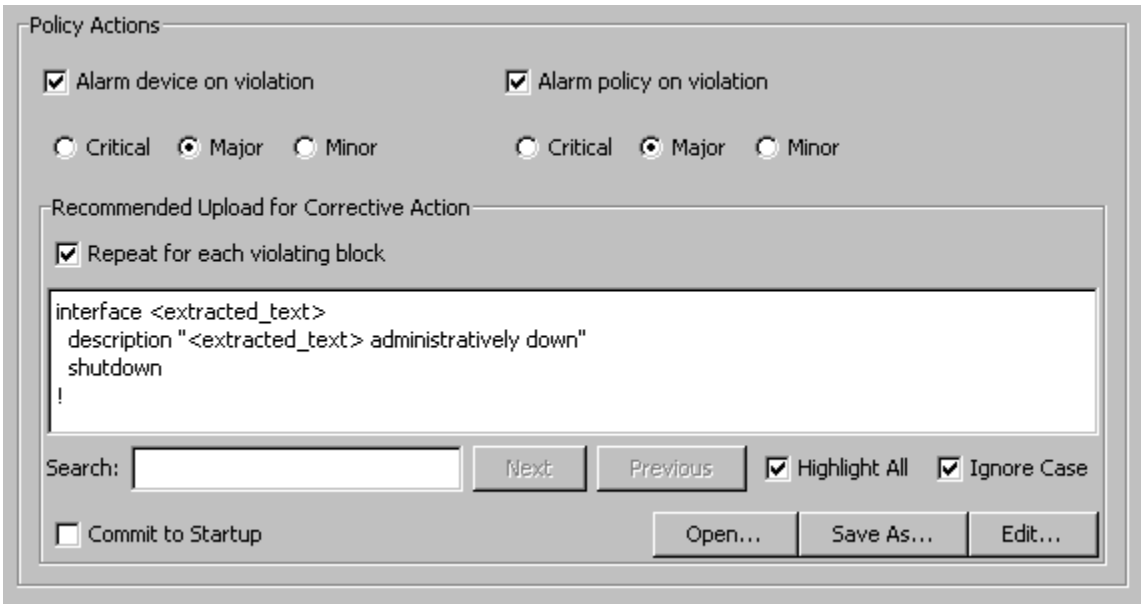
NOTE
The end of line charcter in each line must be in UNIX format.

Compare with Reference Configuration:
The option to 'Compare with Matching Block from Reference Configuration' indicates that each configuration that is captured after the policy is enabled will be compared to the configuration that has been designated as the Reference Configuration for the device. Click the Advanced button to specify the Block Identifier, which is used to match corresponding blocks between the current configuration and the reference (or previous, if specified) configuration. The following example will match up based on "interface *name'":*

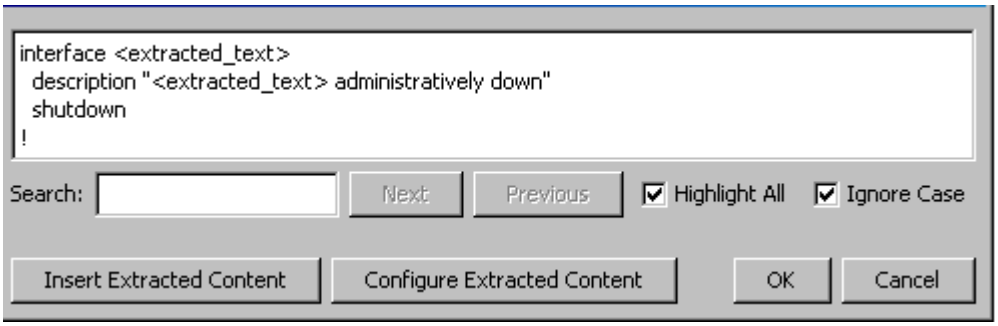


Policy Actions

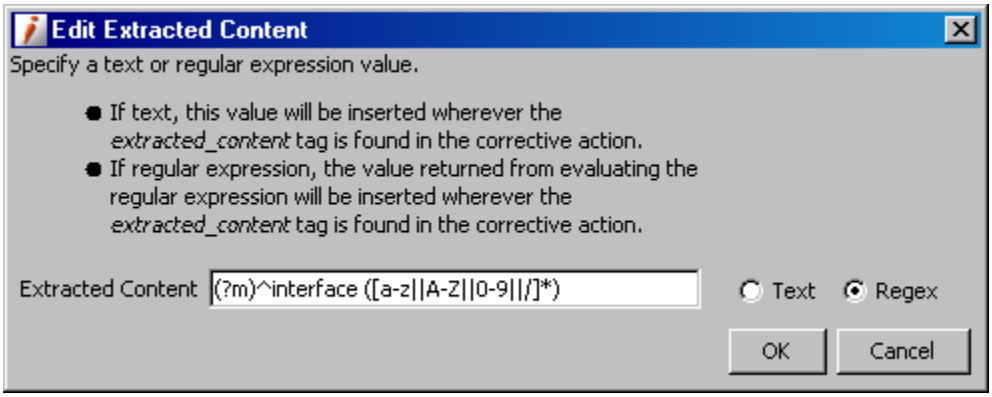
The Policy Actions options define how alarms should be generated and the corrective action to be uploaded should non-compliance occur. The following image shows this section (already populated) and will be described subsequently:



- Specify alarm preferences for when a violation occurs. Alarms can be associated with the device, the policy, or both.
- To define the 'Recommended Upload for Corrective Action,' click the Edit button to display the Edit Corrective Action dialog. In the box, enter the content that will be uploaded to the device. The following image shows content that will upload a modified description and the shutdown command to the device:



In this example, the <extracted_text> tag will be replaced by block-specific content when the policy runs. To insert this tag into your corrective action, use the Insert Extracted Content button. To configure what will be used to replace the tag, click the Configure Extracted Content button, which opens the following dialog:



In this example, the name of the interface will be extracted from each block and used to create corrective action content.

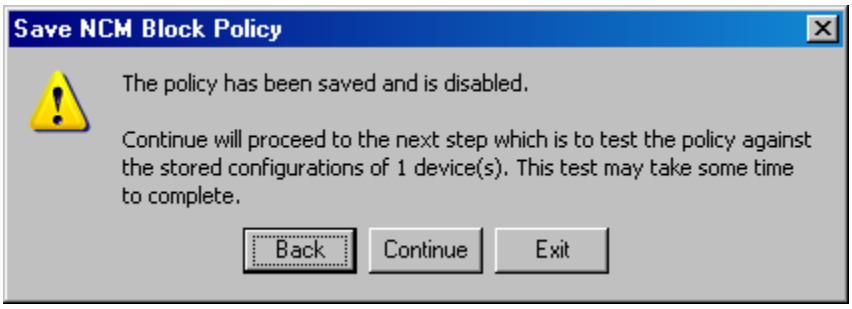
After the corrective action content is defined, it appears in the Recommended Upload for Corrective Action box. Select the 'Repeat for each violating block' option if you want this change to be made for each occurrence of a violation; if it is left blank, the change will only be made for the first occurrence.

Saving and Testing the Policy

After the policy is initially defined, it should be tested before being enabled to make sure it operates as expected.

To proceed with testing of the policy, click Save on the Create NCM Block Policy dialog to save your settings. On the ensuing Save dialog, shown in the next image, click Continue to test the policy.

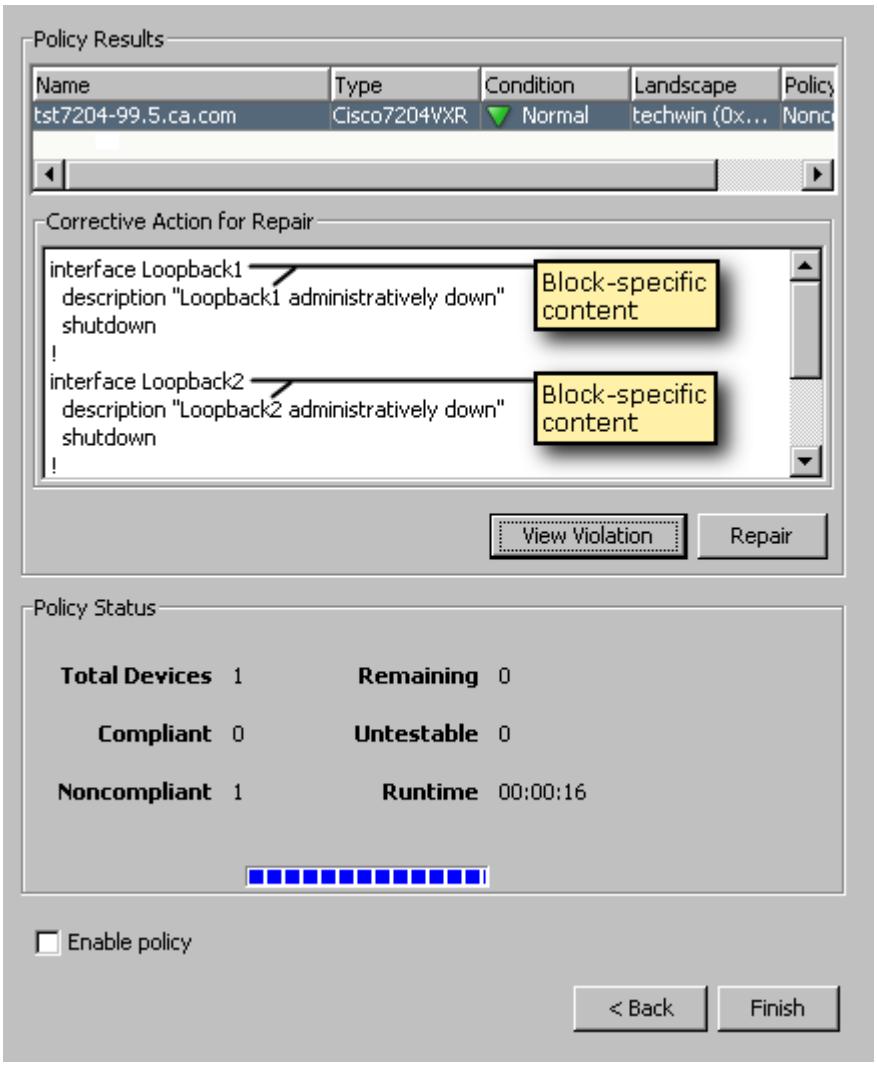
> **NOTE**
> You can also select Back to make additional changes to the policy definition. If you click Exit, the policy is saved but disabled.



The Test NCM Block Policy dialog opens, the test begins, and the status bar indicates its progress. During the test, current configurations are captured and compared to criteria specified in the policy. Blocks are matched based on the block identifier and the contents of corresponding blocks are compared.

NOTE
Depending on the number of devices included, the test may take a while to complete.

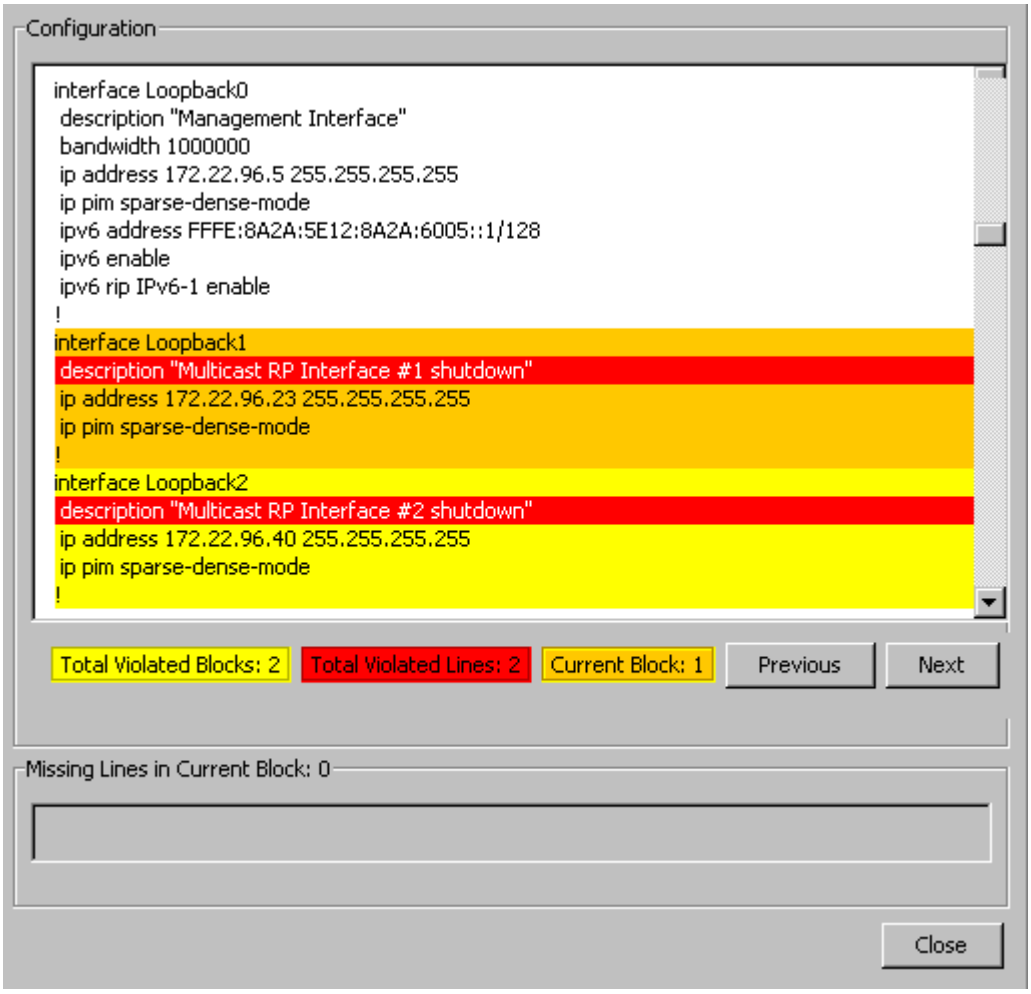
When testing of the policy is complete, the policy results are displayed, as follows:



When non-compliancy is detected, as in this example, the number of devices affected is reported in the Policy Status section, and, if it has been defined, corrective action is displayed as well. Notice that the <extracted_text> tag has been replaced with block-specific content.

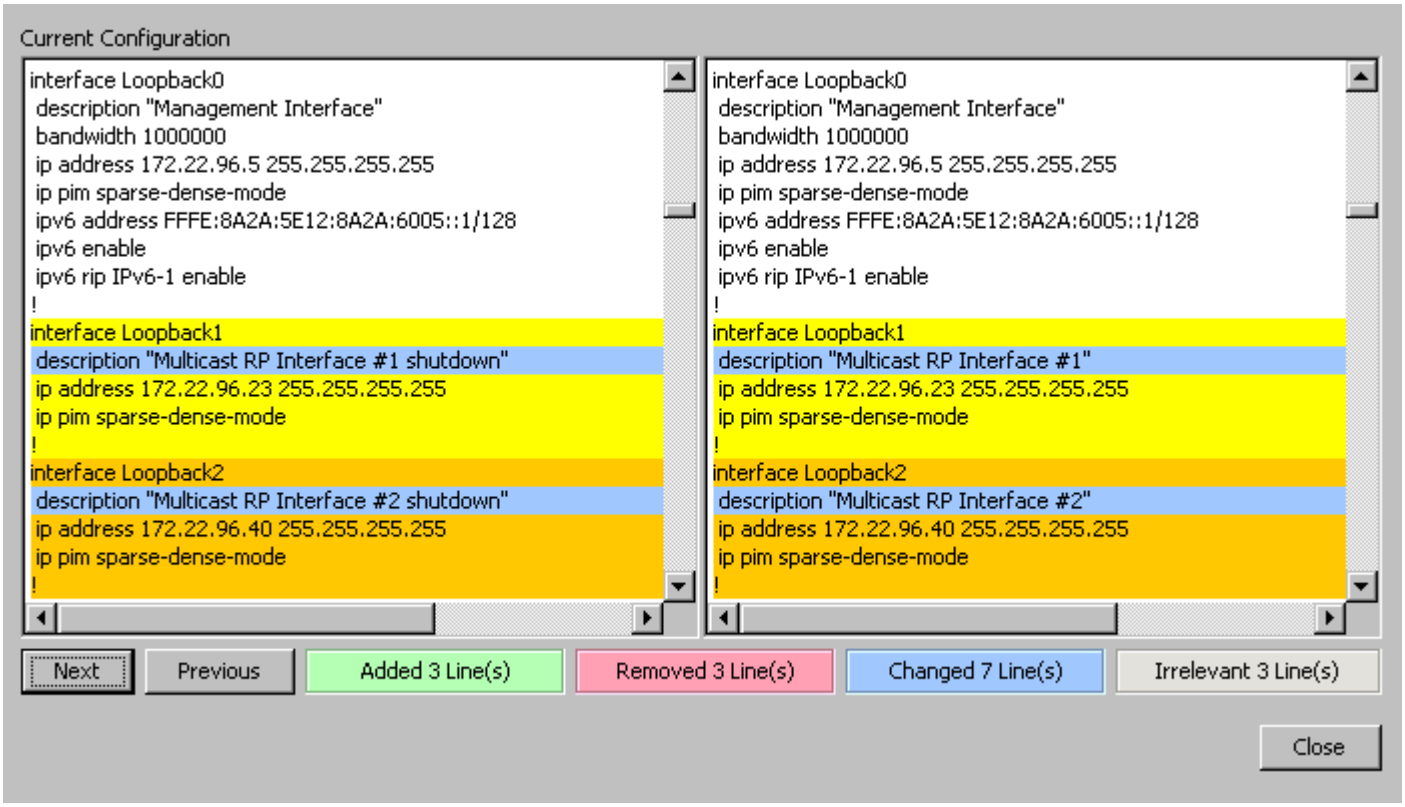
After the test is complete, you can do the following from the test dialog:

- To view the policy violations, click View Violation. A dialog appears that shows the violations.
  - If you used specified contents for comparison criteria, the View Violation Blocks dialog appears, as in the following image. Each block is distinguished by color and the violated lines are highlighted. In this example, the lines beginning with "description" and containing "shutdown" are identified as violations. You can scroll through the violations using the Next and Previous buttons.



If you used another configuration for comparison criteria, the View Policy Violation dialog appears, as in the following image. Each block is distinguished by color and the differences are highlighted. In this example, the description content for two of the interfaces does not match what was defined as the reference. You can scroll through the violations using the Next and Previous buttons.

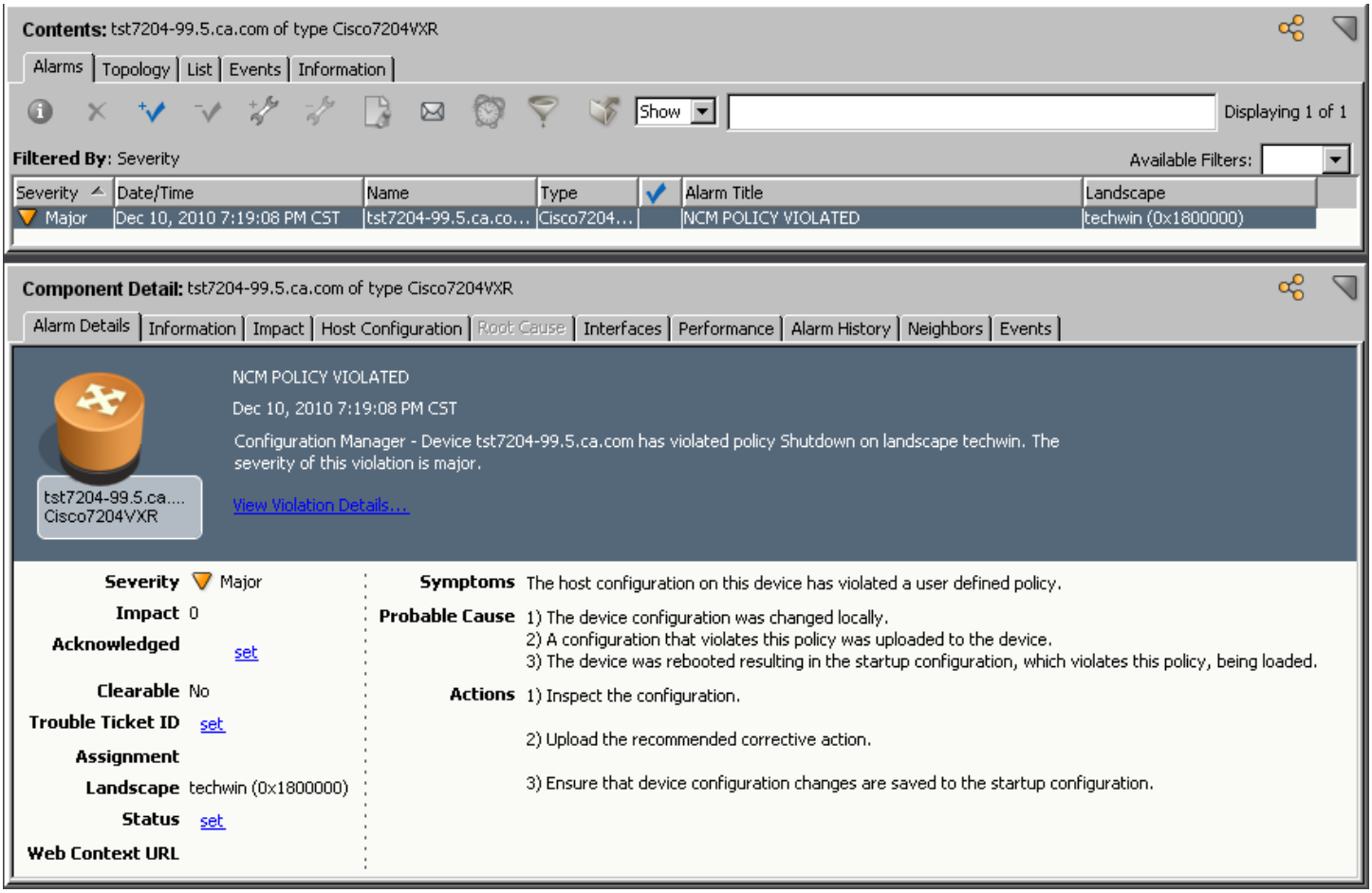
WARNING
When comparing against a reference configuration, be sure to review each of the differences found so that you do not inadvertently execute a corrective action where it does not apply.



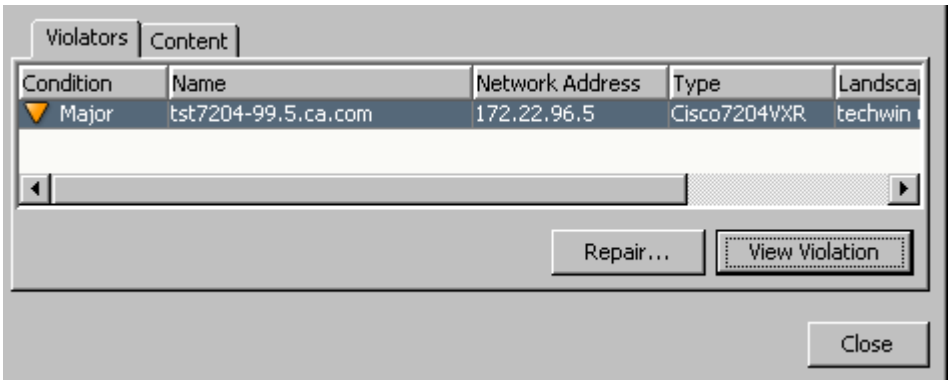
- To correct the policy violation and thereby make the device compliant, click Repair to upload the content as outlined in the 'Corrective Action for Repair' box. An Upload task is created and executed, with the results of the task displayed in the 'Upload Task Results' dialog.
- If you are satisfied with the results of the test, select the 'Enable policy' option to start automatic monitoring and alarm generation based on this policy; otherwise, you can click Back and modify the policy definition. If you click Finish, the policy will be saved but not enabled.

Monitoring Violations

After a policy has been enabled, it monitors configurations that are captured and will alert you to any violations based on the actions you have specified. An alarm generated by a policy violation has an alarm title of 'NCM Policy Violated.' The following image shows an alarm generated based on this example policy:



From the alarm details, you can click View Violation Details, which will open the Repair Devices in Violation dialog as follows:



From this dialog, you can use the available buttons to view violations or repair the non-compliant devices as described when testing the policy.

NOTE
You can also launch this dialog from the Network Configuration Policies table in the Contents panel for the non-compliant device. For more information, see Repair Non-Compliant Devices from the Policy Table.

Devices Supported by Network Configuration Manager

Cisco Devices

The DX NetOps Spectrum Network Configuration Manager supports the following Cisco devices. For a list of supported Catalyst devices, see Cisco Supported Catalyst Devices. For a list of PIX firewall devices, see Cisco Supported PIX Firewall Devices.

The table provides examples. For the most up-to-date information on device support, access the CA Device Certification database.

When a Perl script is the only means of communication with the device, the script method is provided.

Device Name	Sys OID	SNMP/TFTP Support	Telnet Support	SSH Support	Perl Support
CiscoDSC9216K9	1.3.6.1.4.1.9.1.521	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco677i	1.3.6.1.4.1.9.1.363	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco741	1.3.6.1.4.1.9.1.94	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco742	1.3.6.1.4.1.9.1.95	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco743	1.3.6.1.4.1.9.1.96	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco744	1.3.6.1.4.1.9.1.97	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco751	1.3.6.1.4.1.9.1.81	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco752	1.3.6.1.4.1.9.1.82	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco753	1.3.6.1.4.1.9.1.83	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco761	1.3.6.1.4.1.9.1.98	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco762	1.3.6.1.4.1.9.1.99	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco765	1.3.6.1.4.1.9.1.102	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco766	1.3.6.1.4.1.9.1.103	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco771	1.3.6.1.4.1.9.1.126	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco772	1.3.6.1.4.1.9.1.127	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco775	1.3.6.1.4.1.9.1.128	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco776	1.3.6.1.4.1.9.1.129	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes

Cisco801	1.3.6.1.4.1.9.1.212	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco802	1.3.6.1.4.1.9.1.213	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco802J	1.3.6.1.4.1.9.1.295	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco803	1.3.6.1.4.1.9.1.214	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco804	1.3.6.1.4.1.9.1.215	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco804J	1.3.6.1.4.1.9.1.296	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco805	1.3.6.1.4.1.9.1.245	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco806	1.3.6.1.4.1.9.1.384	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco811	1.3.6.1.4.1.9.1.395	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco813	1.3.6.1.4.1.9.1.396	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco826	1.3.6.1.4.1.9.1.322	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco826QuadV	1.3.6.1.4.1.9.1.321	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco827	1.3.6.1.4.1.9.1.284	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco827H	1.3.6.1.4.1.9.1.446	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco827QuadV	1.3.6.1.4.1.9.1.270	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco828	1.3.6.1.4.1.9.1.382	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco831	1.3.6.1.4.1.9.1.497	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco836	1.3.6.1.4.1.9.1.499	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco837	1.3.6.1.4.1.9.1.495	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco871	1.3.6.1.4.1.9.1.571	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco877	1.3.6.1.4.1.9.1.569	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco878	1.3.6.1.4.1.9.1.570	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco1000	1.3.6.1.4.1.9.1.40	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco1003	1.3.6.1.4.1.9.1.41	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes

Cisco1004	1.3.6.1.4.1.9.1.44	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco1005	1.3.6.1.4.1.9.1.49	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco1020	1.3.6.1.4.1.9.1.43	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco1401	1.3.6.1.4.1.9.1.206	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco1407	1.3.6.1.4.1.9.1.249	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco1417	1.3.6.1.4.1.9.1.250	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco1502	1.3.6.1.4.1.9.1.161	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco1503	1.3.6.1.4.1.9.1.160	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco1538M	1.3.6.1.4.1.9.1.224	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco1548M	1.3.6.1.4.1.9.1.225	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco1601	1.3.6.1.4.1.9.1.113	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco1602	1.3.6.1.4.1.9.1.114	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco1603	1.3.6.1.4.1.9.1.115	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco1604	1.3.6.1.4.1.9.1.116	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco1605	1.3.6.1.4.1.9.1.172	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco1701ADSLBRI	1.3.6.1.4.1.9.1.550	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco1710	1.3.6.1.4.1.9.1.200	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco1711	1.3.6.1.4.1.9.1.538	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco1712	1.3.6.1.4.1.9.1.539	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco1720	1.3.6.1.4.1.9.1.201	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco1721	1.3.6.1.4.1.9.1.444	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco1750	1.3.6.1.4.1.9.1.216	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco1751	1.3.6.1.4.1.9.1.326	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco1760	1.3.6.1.4.1.9.1.416	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes

Cisco1801	1.3.6.1.4.1.9.1.638	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco1811	1.3.6.1.4.1.9.1.641	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco1812	1.3.6.1.4.1.9.1.642	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco1841	1.3.6.1.4.1.9.1.620	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco2000	1.3.6.1.4.1.9.1.10	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco2102	1.3.6.1.4.1.9.1.15	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco2202	1.3.6.1.4.1.9.1.16	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco2500	1.3.6.1.4.1.9.1.13	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco2501	1.3.6.1.4.1.9.1.17	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco2502	1.3.6.1.4.1.9.1.18	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco2503	1.3.6.1.4.1.9.1.19	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco2504	1.3.6.1.4.1.9.1.20	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco2505	1.3.6.1.4.1.9.1.21	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco2506	1.3.6.1.4.1.9.1.22	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco2507	1.3.6.1.4.1.9.1.23	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco2508	1.3.6.1.4.1.9.1.24	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco2509	1.3.6.1.4.1.9.1.25	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco2501FRADFX	1.3.6.1.4.1.9.1.165	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco2501LANFRADFX	1.3.6.1.4.1.9.1.166	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco2502LANFRADFX	1.3.6.1.4.1.9.1.167	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco2510	1.3.6.1.4.1.9.1.26	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco2511	1.3.6.1.4.1.9.1.27	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco2512	1.3.6.1.4.1.9.1.28	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco2513	1.3.6.1.4.1.9.1.29	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes

Cisco2514	1.3.6.1.4.1.9.1.30	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco2515	1.3.6.1.4.1.9.1.31	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco2516	1.3.6.1.4.1.9.1.42	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco2517	1.3.6.1.4.1.9.1.67	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco2518	1.3.6.1.4.1.9.1.68	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco2519	1.3.6.1.4.1.9.1.69	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco2520	1.3.6.1.4.1.9.1.70	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco2521	1.3.6.1.4.1.9.1.71	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco2522	1.3.6.1.4.1.9.1.72	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco2523	1.3.6.1.4.1.9.1.73	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco2524	1.3.6.1.4.1.9.1.74	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco2525	1.3.6.1.4.1.9.1.75	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco2610	1.3.6.1.4.1.9.1.185	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco2610M	1.3.6.1.4.1.9.1.418	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco2610XM	1.3.6.1.4.1.9.1.466	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco2611	1.3.6.1.4.1.9.1.186	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco2611M	1.3.6.1.4.1.9.1.419	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco2611XM	1.3.6.1.4.1.9.1.467	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco2612	1.3.6.1.4.1.9.1.187	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco2613	1.3.6.1.4.1.9.1.195	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco2620	1.3.6.1.4.1.9.1.208	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco2620XM	1.3.6.1.4.1.9.1.468	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco2621	1.3.6.1.4.1.9.1.209	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco2621XM	1.3.6.1.4.1.9.1.469	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes

Cisco2650	1.3.6.1.4.1.9.1.319	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco2650XM	1.3.6.1.4.1.9.1.470	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco2651	1.3.6.1.4.1.9.1.320	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco2651XM	1.3.6.1.4.1.9.1.471	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco2691	1.3.6.1.4.1.9.1.413	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco2801	1.3.6.1.4.1.9.1.619	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco2811	1.3.6.1.4.1.9.1.576	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco2821	1.3.6.1.4.1.9.1.577	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco2851	1.3.6.1.4.1.9.1.578	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco3000	1.3.6.1.4.1.9.1.6	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco3101	1.3.6.1.4.1.9.1.32	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco3102	1.3.6.1.4.1.9.1.33	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco3103	1.3.6.1.4.1.9.1.34	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco3104	1.3.6.1.4.1.9.1.35	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco3202	1.3.6.1.4.1.9.1.36	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco3204	1.3.6.1.4.1.9.1.37	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco3220	1.3.6.1.4.1.9.1.553	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco3250	1.3.6.1.4.1.9.1.479	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco3620	1.3.6.1.4.1.9.1.122	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco3640	1.3.6.1.4.1.9.1.110	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco3660	1.3.6.1.4.1.9.1.205	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco3661Ac	1.3.6.1.4.1.9.1.338	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco3661Dc	1.3.6.1.4.1.9.1.339	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco3662Ac	1.3.6.1.4.1.9.1.340	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes

Cisco3662AcCo	1.3.6.1.4.1.9.1.342	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco3662Dc	1.3.6.1.4.1.9.1.341	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco3662DcCo	1.3.6.1.4.1.9.1.343	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco371098-HP001	1.3.6.1.4.1.9.1.625	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco371098-HP001	1.3.6.1.4.1.11.2.3.7.11.33.3.1.1	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco3725	1.3.6.1.4.1.9.1.414	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco3745	1.3.6.1.4.1.9.1.436	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco3825	1.3.6.1.4.1.9.1.543	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco3845	1.3.6.1.4.1.9.1.544	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco4000	1.3.6.1.4.1.9.1.7	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco4224	1.3.6.1.4.1.9.1.399	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco4500	1.3.6.1.4.1.9.1.14	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco4700	1.3.6.1.4.1.9.1.50	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco6015	1.3.6.1.4.1.9.1.299	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco6100	1.3.6.1.4.1.9.1.251	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco6130	1.3.6.1.4.1.9.1.252	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco6160	1.3.6.1.4.1.9.1.297	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco6200	1.3.6.1.4.1.9.1.192	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco6260	1.3.6.1.4.1.9.1.253	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco6400	1.3.6.1.4.1.9.1.180	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco6400Nrp	1.3.6.1.4.1.9.1.211	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco6400UAC	1.3.6.1.4.1.9.1.464	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco7000	1.3.6.1.4.1.9.1.8	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco7010	1.3.6.1.4.1.9.1.12	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes

Cisco7120Ae3	1.3.6.1.4.1.9.1.263	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco7120At3	1.3.6.1.4.1.9.1.262	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco7120E3	1.3.6.1.4.1.9.1.261	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco7120Quadt1	1.3.6.1.4.1.9.1.259	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco7120Smi3	1.3.6.1.4.1.9.1.264	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco7120T3	1.3.6.1.4.1.9.1.260	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco7140Dualae3	1.3.6.1.4.1.9.1.268	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco7140Dualat3	1.3.6.1.4.1.9.1.267	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco7140Duale3	1.3.6.1.4.1.9.1.266	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco7140Dualfe	1.3.6.1.4.1.9.1.277	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco7140Dualmm3	1.3.6.1.4.1.9.1.269	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco7140Dualt3	1.3.6.1.4.1.9.1.265	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco7140Octt1	1.3.6.1.4.1.9.1.276	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco7150Dualfe	1.3.6.1.4.1.9.1.355	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco7150Dualt3	1.3.6.1.4.1.9.1.357	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco7150Octt1	1.3.6.1.4.1.9.1.356	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco7202	1.3.6.1.4.1.9.1.194	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco7204	1.3.6.1.4.1.9.1.125	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco7204VXR	1.3.6.1.4.1.9.1.223	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco7206	1.3.6.1.4.1.9.1.108	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco7206VXR	1.3.6.1.4.1.9.1.222	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
UBR_7246	1.3.6.1.4.1.9.1.179	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco7301	1.3.6.1.4.1.9.1.476	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco7304	1.3.6.1.4.1.9.1.439	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes

Cisco7401ASR	1.3.6.1.4.1.9.1.403	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco7401VXR	1.3.6.1.4.1.9.1.376	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco7505	1.3.6.1.4.1.9.1.48	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco7507z	1.3.6.1.4.1.9.1.288	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco7506	1.3.6.1.4.1.9.1.47	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco7507	1.3.6.1.4.1.9.1.45	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco7507mx	1.3.6.1.4.1.9.1.290	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco7513	1.3.6.1.4.1.9.1.46	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco7513mx	1.3.6.1.4.1.9.1.291	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco7513z	1.3.6.1.4.1.9.1.289	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco7576	1.3.6.1.4.1.9.1.204	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco7603	1.3.6.1.4.1.9.1.401	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco7604	1.3.6.1.4.1.9.1.658	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco7606	1.3.6.1.4.1.9.1.402	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco7609	1.3.6.1.4.1.9.1.509	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco7613	1.3.6.1.4.1.9.1.528	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco9004	1.3.6.1.4.1.9.1.424	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco10005	1.3.6.1.4.1.9.1.437	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco10008	1.3.6.1.4.1.9.1.438	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco10400	1.3.6.1.4.1.9.1.272	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco10720	1.3.6.1.4.1.9.1.397	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco12004	1.3.6.1.4.1.9.1.181	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco12006	1.3.6.1.4.1.9.1.590	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco12008	1.3.6.1.4.1.9.1.182	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes

Cisco12010	1.3.6.1.4.1.9.1.348	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco12012	1.3.6.1.4.1.9.1.173	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco12016	1.3.6.1.4.1.9.1.273	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco12404	1.3.6.1.4.1.9.1.423	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco12406	1.3.6.1.4.1.9.1.388	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco12410	1.3.6.1.4.1.9.1.394	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco12416	1.3.6.1.4.1.9.1.385	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco3631Co	1.3.6.1.4.1.9.1.425	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoAGS+	1.3.6.1.4.1.9.1.11	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoAPEC	1.3.6.1.4.1.9.1.39	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoAPRC	1.3.6.1.4.1.9.1.38	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoAS5200	1.3.6.1.4.1.9.1.109	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoAS5300	1.3.6.1.4.1.9.1.162	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoAS5350	1.3.6.1.4.1.9.1.313	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoAS5350XM	1.3.6.1.4.1.9.1.679	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoAS5400	1.3.6.1.4.1.9.1.274	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoAS5400XM	1.3.6.1.4.1.9.1.668	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoAS5800	1.3.6.1.4.1.9.1.188	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoAS5850	1.3.6.1.4.1.9.1.308	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoCacheEngine	1.3.6.1.4.1.9.1.240	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoCrs1Fabric	1.3.6.1.4.1.9.1.739	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoCRS16S	1.3.6.1.4.1.9.1.613	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoCrs18LineCard	1.3.6.1.4.1.9.1.738	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoCRS8S	1.3.6.1.4.1.9.1.643	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes

CiscoCS500	1.3.6.1.4.1.9.1.9	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoFastHubBMMFX	1.3.6.1.4.1.9.1.178	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoFastHubBMMTX	1.3.6.1.4.1.9.1.177	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoFastHub216T	1.3.6.1.4.1.9.1.169	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoGS	1.3.6.1.4.1.9.1.1	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoIGESM	1.3.6.1.4.1.9.1.592	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoIGS	1.3.6.1.4.1.9.1.5	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoLocalDirector	1.3.6.1.4.1.9.1.244	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco MC3810	1.3.6.1.4.1.9.1.286	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cisco MC3810	1.3.6.1.4.1.9.1.157	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoME6340ACA	1.3.6.1.4.1.9.1.713	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoME6340DCA	1.3.6.1.4.1.9.1.714	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoME6340DCB	1.3.6.1.4.1.9.1.715	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoMicroWebServer2	1.3.6.1.4.1.9.1.176	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoMWR1900	1.3.6.1.4.1.9.1.398	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoMWR1941DC	1.3.6.1.4.1.9.1.520	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoOlympus	1.3.6.1.4.1.9.1.358	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoOpticalRegenerator	1.3.6.1.4.1.9.1.254	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoPro316C	1.3.6.1.4.1.9.1.148	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoPro316T	1.3.6.1.4.1.9.1.147	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoPro741	1.3.6.1.4.1.9.1.84	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoPro742	1.3.6.1.4.1.9.1.85	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoPro743	1.3.6.1.4.1.9.1.86	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoPro744	1.3.6.1.4.1.9.1.87	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes

CiscoPro751	1.3.6.1.4.1.9.1.76	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoPro752	1.3.6.1.4.1.9.1.77	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoPro753	1.3.6.1.4.1.9.1.78	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoPro761	1.3.6.1.4.1.9.1.88	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoPro762	1.3.6.1.4.1.9.1.89	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoPro765	1.3.6.1.4.1.9.1.92	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoPro766	1.3.6.1.4.1.9.1.93	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoPro1003	1.3.6.1.4.1.9.1.51	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoPro1004	1.3.6.1.4.1.9.1.52	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoPro1005	1.3.6.1.4.1.9.1.53	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoPro1020	1.3.6.1.4.1.9.1.54	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoPro1601	1.3.6.1.4.1.9.1.117	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoPro1602	1.3.6.1.4.1.9.1.118	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoPro1603	1.3.6.1.4.1.9.1.119	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoPro1604	1.3.6.1.4.1.9.1.120	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoPro2500PCE	1.3.6.1.4.1.9.1.55	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoPro2501	1.3.6.1.4.1.9.1.56	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoPro2502	1.3.6.1.4.1.9.1.130	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoPro2503	1.3.6.1.4.1.9.1.57	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoPro2504	1.3.6.1.4.1.9.1.131	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoPro2505	1.3.6.1.4.1.9.1.58	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoPro2506	1.3.6.1.4.1.9.1.132	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoPro2507	1.3.6.1.4.1.9.1.59	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoPro2508	1.3.6.1.4.1.9.1.133	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes

CiscoPro2509	1.3.6.1.4.1.9.1.60	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoPro2510	1.3.6.1.4.1.9.1.134	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoPro2511	1.3.6.1.4.1.9.1.61	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoPro2512	1.3.6.1.4.1.9.1.135	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoPro2513	1.3.6.1.4.1.9.1.136	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoPro2514	1.3.6.1.4.1.9.1.62	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoPro2515	1.3.6.1.4.1.9.1.137	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoPro2516	1.3.6.1.4.1.9.1.63	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoPro2517	1.3.6.1.4.1.9.1.138	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoPro2518	1.3.6.1.4.1.9.1.139	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoPro2519	1.3.6.1.4.1.9.1.64	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoPro2520	1.3.6.1.4.1.9.1.104	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoPro2521	1.3.6.1.4.1.9.1.65	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoPro2522	1.3.6.1.4.1.9.1.105	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoPro2523	1.3.6.1.4.1.9.1.140	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoPro2524	1.3.6.1.4.1.9.1.106	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoPro2525	1.3.6.1.4.1.9.1.141	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoPro3116	1.3.6.1.4.1.9.1.149	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoPro3620	1.3.6.1.4.1.9.1.123	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoPro3640	1.3.6.1.4.1.9.1.124	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoPro4500	1.3.6.1.4.1.9.1.66	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoPro4700	1.3.6.1.4.1.9.1.142	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoProtocolTranslator	1.3.6.1.4.1.9.1.4	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoRPM	1.3.6.1.4.1.9.1.199	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes

CiscoRPMPR	1.3.6.1.4.1.9.1.457	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoRpmXf	1.3.6.1.4.1.9.1.440	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoSC3640	1.3.6.1.4.1.9.1.189	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoSN5420	1.3.6.1.4.1.9.1.407	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoSN5428	1.3.6.1.4.1.9.1.475	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoSOHO76	1.3.6.1.4.1.9.1.354	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoSOHO91	1.3.6.1.4.1.9.1.498	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoSOHO97	1.3.6.1.4.1.9.1.496	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoSOHO77	1.3.6.1.4.1.9.1.353	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoSOHO96	1.3.6.1.4.1.9.1.500	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoTrouter	1.3.6.1.4.1.9.1.3	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoTS	1.3.6.1.4.1.9.1.2	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoWS3020Hpq	1.3.6.1.4.1.9.1.748	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoWS3030Del	1.3.6.1.4.1.9.1.749	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoWSC3750G-24PS	1.3.6.1.4.1.9.1.747	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoWSC6504E	1.3.6.1.4.1.9.1.657	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoWSC6509neba	1.3.6.1.4.1.9.1.534	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoWSX3011	1.3.6.1.4.1.9.1.112	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoWSX5302	1.3.6.1.4.1.9.1.168	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoWSX6302Msm	1.3.6.1.4.1.9.1.256	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoURM	1.3.6.1.4.1.9.1.373	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoURM2FE	1.3.6.1.4.1.9.1.374	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CiscoURM2FE2V	1.3.6.1.4.1.9.1.375	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
AP 1130	1.3.6.1.4.1.9.1.618	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes

LS_1010	1.3.6.1.4.1.9.1.107	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
LS_1015	1.3.6.1.4.1.9.1.164	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
UBR_7223	1.3.6.1.4.1.9.1.210	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
UBR_7246VXR	1.3.6.1.4.1.9.1.271	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
UBR_904	1.3.6.1.4.1.9.1.191	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
UBR_924	1.3.6.1.4.1.9.1.255	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
UBR_912C	1.3.6.1.4.1.9.1.292	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
UBR_912S	1.3.6.1.4.1.9.1.293	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
UBR_914	1.3.6.1.4.1.9.1.294	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
UBR_925	1.3.6.1.4.1.9.1.316	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
UBR_10012	1.3.6.1.4.1.9.1.317	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
UBR_7111	1.3.6.1.4.1.9.1.344	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
UBR_7111E	1.3.6.1.4.1.9.1.345	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
UBR_7114	1.3.6.1.4.1.9.1.346	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
UBR_7114E	1.3.6.1.4.1.9.1.347	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
UBR_905	1.3.6.1.4.1.9.1.351	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
350 AP	1.3.6.1.4.1.9.1.552	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
1100 AP	1.3.6.1.4.1.9.1.507	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
1210/1230 AP	1.3.6.1.4.1.9.1.525	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
1240 AP	1.3.6.1.4.1.9.1.685	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
1400 AP	1.3.6.1.4.1.9.1.533	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
1300 AP	1.3.6.1.4.1.9.1.565	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes

* IOS < 12.0 = OLD-CISCO-SYSTEM-MIB

** IOS > 12.2(18) with feature “K9”

Cisco Supported PIX Firewall Devices

The DX NetOps Spectrum Network Configuration Manager supports the following Cisco devices. For a list of supported Catalyst devices, see Cisco Supported Catalyst Devices.

The table provides examples. For the most up-to-date information on device support, access the CA Device Certification database.

When a Perl script is the only means of communication with the device, the script method is provided.

Device Name	Sys OID	SNMP/TFTP Support	Telnet Support	SSH Support	Perl Support
PIX Firewall	1.3.6.1.4.1.9.1.227	no	no	no	Telnet
PIX 506 Firewall	1.3.6.1.4.1.9.1.389	no	no	no	Telnet
PIX 515 Firewall	1.3.6.1.4.1.9.1.390	no	no	no	Telnet
PIX 520 Firewall	1.3.6.1.4.1.9.1.391	no	no	no	Telnet
PIX 525 Firewall	1.3.6.1.4.1.9.1.392	no	no	no	Telnet
PIX 535 Firewall	1.3.6.1.4.1.9.1.393	no	no	no	Telnet
PIX 501 Firewall	1.3.6.1.4.1.9.1.417	no	no	no	Telnet
PIX 515E Firewall	1.3.6.1.4.1.9.1.451	no	no	no	Telnet
PIX 506E Firewall	1.3.6.1.4.1.9.1.450	no	no	no	Telnet
cat6500FirewallSm	1.3.6.1.4.1.9.1.522	no	no	no	Telnet
PIX Firewall Security Module	1.3.6.1.4.1.9.1.674	no	no	no	Telnet
PIX 535sc Firewall	1.3.6.1.4.1.9.1.675	no	no	no	Telnet
PIX 525sc Firewall	1.3.6.1.4.1.9.1.676	no	no	no	Telnet
PIX 515Esc Firewall	1.3.6.1.4.1.9.1.677	no	no	no	Telnet
PIX 515sc Firewall	1.3.6.1.4.1.9.1.678	no	no	no	Telnet
PIX Firewall System Module	1.3.6.1.4.1.9.1.767	no	no	no	Telnet
PIX 515sy Firewall	1.3.6.1.4.1.9.1.768	no	no	no	Telnet
PIX 515Esy Firewall	1.3.6.1.4.1.9.1.769	no	no	no	Telnet
PIX 525sy Firewall	1.3.6.1.4.1.9.1.770	no	no	no	Telnet
PIX 535sy Firewall	1.3.6.1.4.1.9.1.771	no	no	no	Telnet

* IOS < 12.0 = OLD-CISCO-SYSTEM-MIB

** IOS > 12.2(18) with feature “K9”

Cisco CAT Supported Devices

The following table lists Cisco CAT devices supported by DX NetOps Spectrum Network Configuration Manager. The table provides examples. For the most up-to-date information on device support, access the CA Device Certification database.

Device Name	Sys OID	SNMP/TFTP Support	Telnet Support	SSH Support	Perl Support
Cat116T	1.3.6.1.4.1.9.1.150	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes

Cat116C	1.3.6.1.4.1.9.1.151	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat1116	1.3.6.1.4.1.9.1.152	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat1912C	1.3.6.1.4.1.9.1.175	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat2924XL	1.3.6.1.4.1.9.1.183	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat2924CXL	1.3.6.1.4.1.9.1.184	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat2924XLv	1.3.6.1.4.1.9.1.217	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat2640-48TT	1.3.6.1.4.1.9.1.717	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat2948gL3	1.3.6.1.4.1.9.1.275	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat2948gL3Dc	1.3.6.1.4.1.9.1.386	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat2960-24TC	1.3.6.1.4.1.9.1.694	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat2960-24TT	1.3.6.1.4.1.9.1.716	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat2960G-24TC	1.3.6.1.4.1.9.1.696	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat2960-48TC	1.3.6.1.4.1.9.1.695	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat297024	1.3.6.1.4.1.9.1.527	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat297024TS	1.3.6.1.4.1.9.1.561	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat2908xl	1.3.6.1.4.1.9.1.170	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat2912LREXL	1.3.6.1.4.1.9.1.370	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat2912MfXL	1.3.6.1.4.1.9.1.221	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat2912XL	1.3.6.1.4.1.9.1.219	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat2916mxl	1.3.6.1.4.1.9.1.171	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat2924CXLv	1.3.6.1.4.1.9.1.218	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat2924MXL	1.3.6.1.4.1.9.1.220	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat295012	1.3.6.1.4.1.9.1.323	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat295024	1.3.6.1.4.1.9.1.324	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes

Cat295024C	1.3.6.1.4.1.9.1.325	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat2950t24	1.3.6.1.4.1.9.1.359	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat2924LREXL	1.3.6.1.4.1.9.1.369	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat295012G	1.3.6.1.4.1.9.1.427	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat295024G	1.3.6.1.4.1.9.1.428	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat295048G	1.3.6.1.4.1.9.1.429	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat_3500	1.3.6.1.4.1.9.1.111	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat_3508GXL	1.3.6.1.4.1.9.1.246	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat_3512XL	1.3.6.1.4.1.9.1.247	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat_3524XL	1.3.6.1.4.1.9.1.248	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat_3524tXLEn	1.3.6.1.4.1.9.1.287	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat_3548XL	1.3.6.1.4.1.9.1.278	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat355012G	1.3.6.1.4.1.9.1.431	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat355012T	1.3.6.1.4.1.9.1.368	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat355024	1.3.6.1.4.1.9.1.366	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat355048	1.3.6.1.4.1.9.1.367	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat355024Dc	1.3.6.1.4.1.9.1.452	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat355024Mmf	1.3.6.1.4.1.9.1.453	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat355024PWR	1.3.6.1.4.1.9.1.485	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat3560_24PS	1.3.6.1.4.1.9.1.563	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat3560G-24PS	1.3.6.1.4.1.9.1.614	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat3560-24TS	1.3.6.1.4.1.9.1.633	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat3560G-24TS	1.3.6.1.4.1.9.1.615	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat3560_48PS	1.3.6.1.4.1.9.1.564	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes

Cat3560G-48PS	1.3.6.1.4.1.9.1.616	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat3560-48TS	1.3.6.1.4.1.9.1.634	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat3560G-48TS	1.3.6.1.4.1.9.1.617	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat37xxStack	1.3.6.1.4.1.9.1.516	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat3750Ge12Sfp	1.3.6.1.4.1.9.1.530	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat3750_24ME	1.3.6.1.4.1.9.1.574	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat3750G16TD	1.3.6.1.4.1.9.1.591	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat375024	1.3.6.1.4.1.9.1.511	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat375024T	1.3.6.1.4.1.9.1.514	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat375024TS	1.3.6.1.4.1.9.1.513	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat375048	1.3.6.1.4.1.9.1.512	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat4kGateway	1.3.6.1.4.1.9.1.318	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat4000NAM	1.3.6.1.4.1.9.1.575	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat4006	1.3.6.1.4.1.9.1.448	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat4503	1.3.6.1.4.1.9.1.503	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat4510	1.3.6.1.4.1.9.1.537	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat4232L3	1.3.6.1.4.1.9.1.300	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat4506	1.3.6.1.4.1.9.1.502	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat4507	1.3.6.1.4.1.9.1.501	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat4840gL3	1.3.6.1.4.1.9.1.312	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat4908gL3	1.3.6.1.4.1.9.1.298	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat4908gL3Dc	1.3.6.1.4.1.9.1.387	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat4948	1.3.6.1.4.1.9.1.626	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat494810GE	1.3.6.1.4.1.9.1.659	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes

Cat5kRsfc	1.3.6.1.4.1.9.1.257	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat6kSup720	1.3.6.1.4.1.9.1.557	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat6kGateway	1.3.6.1.4.1.9.1.573	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat6503	1.3.6.1.4.1.9.1.449	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat6513	1.3.6.1.4.1.9.1.400	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat6000	1.3.6.1.4.1.9.1.241	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat6006	1.3.6.1.4.1.9.1.280	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat6009	1.3.6.1.4.1.9.1.281	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat6506	1.3.6.1.4.1.9.1.282	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat6509	1.3.6.1.4.1.9.1.283	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat6kMsfc	1.3.6.1.4.1.9.1.258	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat6kMsfc2	1.3.6.1.4.1.9.1.301	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat6509Sp	1.3.6.1.4.1.9.1.310	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat8510_CSR	1.3.6.1.4.1.9.1.190	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat8510_MSR	1.3.6.1.4.1.9.1.230	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat8515_CSR	1.3.6.1.4.1.9.1.196	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat8515_MSR	1.3.6.1.4.1.9.1.231	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat8540_CSR	1.3.6.1.4.1.9.1.203	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat8540_MSR	1.3.6.1.4.1.9.1.202	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat9006	1.3.6.1.4.1.9.1.197	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat9009	1.3.6.1.4.1.9.1.198	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat295024GDC	1.3.6.1.4.1.9.1.472	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat295024S	1.3.6.1.4.1.9.1.430	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat295024SX	1.3.6.1.4.1.9.1.480	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes

Cat295024LREG	1.3.6.1.4.1.9.1.484	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat295024LRESt	1.3.6.1.4.1.9.1.482	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat29508LRESt	1.3.6.1.4.1.9.1.483	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat2955C12	1.3.6.1.4.1.9.1.489	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat2955S12	1.3.6.1.4.1.9.1.508	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat2955T12	1.3.6.1.4.1.9.1.488	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat29408TF	1.3.6.1.4.1.9.1.542	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat29408TT	1.3.6.1.4.1.9.1.540	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat295048SX	1.3.6.1.4.1.9.1.560	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat295048T	1.3.6.1.4.1.9.1.559	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat2950St24LRE997	1.3.6.1.4.1.9.1.551	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CatExpress500-24LC	1.3.6.1.4.1.9.1.725	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CatExpress500-12TC	1.3.6.1.4.1.9.1.727	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CatExpress500-24PC	1.3.6.1.4.1.9.1.726	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CatExpress500-24TT	1.3.6.1.4.1.9.1.724	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
CatWsCBS3040FSC	1.3.6.1.4.1.9.1.784	*CISCO-CONFIG-COPY-MIB	yes	**yes	yes
Cat2926	1.3.6.1.4.1.9.5.35	CISCO-CONFIG-COPY-MIB*	no	no	yes
Cat_2948G	1.3.6.1.4.1.9.5.42	CISCO-CONFIG-COPY-MIB*	no	no	yes
Cat2948ggetx	1.3.6.1.4.1.9.5.62	CISCO-CONFIG-COPY-MIB*	no	no	yes
Cat2980ga	1.3.6.1.4.1.9.5.51	CISCO-CONFIG-COPY-MIB*	no	no	yes
Cat_2980GSW	1.3.6.1.4.1.9.5.49	CISCO-CONFIG-COPY-MIB*	no	no	yes
Cat_4003	1.3.6.1.4.1.9.5.40	CISCO-CONFIG-COPY-MIB*	no	no	yes
Cat_4006	1.3.6.1.4.1.9.5.46	CISCO-CONFIG-COPY-MIB*	no	no	yes
Cat4503	1.3.6.1.4.1.9.5.58	CISCO-CONFIG-COPY-MIB*	no	no	yes

Cat_4506	1.3.6.1.4.1.9.5.59	CISCO-CONFIG-COPY-MIB*	no	no	yes
Cat4912	1.3.6.1.4.1.9.5.41	CISCO-CONFIG-COPY-MIB*	no	no	yes
Cat6knam	1.3.6.1.4.1.9.5.48	CISCO-CONFIG-COPY-MIB*	no	no	yes
Cat6503	1.3.6.1.4.1.9.5.56	CISCO-CONFIG-COPY-MIB*	no	no	yes
Cat6509neba	1.3.6.1.4.1.9.5.61	CISCO-CONFIG-COPY-MIB*	no	no	yes
Cat7603	1.3.6.1.4.1.9.5.53	CISCO-CONFIG-COPY-MIB*	no	no	yes
Cat7604	1.3.6.1.4.1.9.5.63	CISCO-CONFIG-COPY-MIB*	no	no	yes
Cat7606	1.3.6.1.4.1.9.5.54	CISCO-CONFIG-COPY-MIB*	no	no	yes
Cat7609	1.3.6.1.4.1.9.5.55	CISCO-CONFIG-COPY-MIB*	no	no	yes
Cat7613	1.3.6.1.4.1.9.5.60	CISCO-CONFIG-COPY-MIB*	no	no	yes
CiscoWSC6504E	1.3.6.1.4.1.9.5.64	CISCO-CONFIG-COPY-MIB*	no	no	yes
HubCat1400	1.3.6.1.4.1.9.5.6	CISCO-CONFIG-COPY-MIB*	no	no	yes
HubCat5000	1.3.6.1.4.1.9.5.7	CISCO-CONFIG-COPY-MIB*	no	no	yes
HubCat5002	1.3.6.1.4.1.9.5.29	CISCO-CONFIG-COPY-MIB*	no	no	yes
HubCat5500	1.3.6.1.4.1.9.5.17	CISCO-CONFIG-COPY-MIB*	no	no	yes
HubCat5505	1.3.6.1.4.1.9.5.34	CISCO-CONFIG-COPY-MIB*	no	no	yes
HubCat5509	1.3.6.1.4.1.9.5.36	CISCO-CONFIG-COPY-MIB*	no	no	yes
SwCat1200	1.3.6.1.4.1.9.5.5	CISCO-CONFIG-COPY-MIB*	no	no	yes
WS-C6006	1.3.6.1.4.1.9.5.38	CISCO-CONFIG-COPY-MIB*	no	no	yes
WS-C6009	1.3.6.1.4.1.9.5.39	CISCO-CONFIG-COPY-MIB*	no	no	yes
WS-C6506	1.3.6.1.4.1.9.5.45	CISCO-CONFIG-COPY-MIB*	no	no	yes
WS-C6509	1.3.6.1.4.1.9.5.44	CISCO-CONFIG-COPY-MIB*	no	no	yes
WS-C6509neb	1.3.6.1.4.1.9.5.47	CISCO-CONFIG-COPY-MIB*	no	no	yes
WS-C6513	1.3.6.1.4.1.9.5.50	CISCO-CONFIG-COPY-MIB*	no	no	yes

* CATOS < 8.4 = CISCO-STACK-MIB

Cisco NX OS Supported Devices

The following table lists Cisco NX OS devices supported by DX NetOps Spectrum Network Configuration Manager.

NOTE
Cisco NX OS devices are supported through scripts that utilize the Net::SSH::Expect modules. The Perl area must be set up with these modules for out-of box support for Cisco NX OS devices.

Device Name	Sys OID	SNMP/TFTP Support	Telnet Support	SSH Support	Perl Support
Cisco Nexus 1000V VSM	1.3.6.1.4.1.9.12.3.1.3.840	no	no	no	yes
Cisco Nexus 2000	1.3.6.1.4.1.9.12.3.1.3.820	no	no	no	yes
Cisco Nexus 5000	1.3.6.1.4.1.9.12.3.1.3.719	no	no	no	yes
Cisco Nexus 7000	1.3.6.1.4.1.9.12.3.1.3.612	no	no	no	yes

Enterasys Supported Devices

The following table lists Enterasys devices supported by DX NetOps Spectrum Network Configuration Manager. The table provides examples. For the most up-to-date information on device support, access the CA Device Certification database.

Device Name	Sys OID	SNMP/TFTP Support	Telnet Support	SSH Support	Perl Support
1G582-09	1.3.6.1.4.1.5624.2.1.35	ENTERASYS-CONFIG-MAN-MIB	no	no	yes
1G694-13	1.3.6.1.4.1.5624.2.1.36	ENTERASYS-CONFIG-MAN-MIB	no	no	yes
1H582-25	1.3.6.1.4.1.5624.2.1.59	ENTERASYS-CONFIG-MAN-MIB	no	no	yes
1H582-51	1.3.6.1.4.1.5624.2.1.34	ENTERASYS-CONFIG-MAN-MIB	no	no	yes
1G587-09	1.3.6.1.4.1.5624.2.1.60	ENTERASYS-CONFIG-MAN-MIB	no	no	yes
Matrix N	1.3.6.1.4.1.5624.2.1.51	ENTERASYS-CONFIG-MAN-MIB	no	no	yes
Matrix N1	1.3.6.1.4.1.5624.2.1.83	ENTERASYS-CONFIG-MAN-MIB	no	no	yes
Matrix N3	1.3.6.1.4.1.5624.2.1.53	ENTERASYS-CONFIG-MAN-MIB	no	no	yes
Matrix N5	1.3.6.1.4.1.5624.2.1.79	ENTERASYS-CONFIG-MAN-MIB	no	no	yes
Matrix N7	1.3.6.1.4.1.5624.2.1.52	ENTERASYS-CONFIG-MAN-MIB	no	no	yes
Matrix N Router	1.3.6.1.4.1.5624.2.1.70	ENTERASYS-CONFIG-MAN-MIB	no	no	yes
Matrix N Standalone	1.3.6.1.4.1.5624.2.1.77	ENTERASYS-CONFIG-MAN-MIB	no	no	yes

SecureStack A2H124-24	1.3.6.1.4.1.5624.2.1.87	ENTERASYS-CONFIG-MAN-MIB	no	no	yes
SecureStack A2H124-24FX	1.3.6.1.4.1.5624.2.1.91	ENTERASYS-CONFIG-MAN-MIB	no	no	yes
SecureStack A2H124-24P	1.3.6.1.4.1.5624.2.1.88	ENTERASYS-CONFIG-MAN-MIB	no	no	yes
SecureStack A2H124-48	1.3.6.1.4.1.5624.2.1.89	ENTERASYS-CONFIG-MAN-MIB	no	no	yes
SecureStack A2H124-48P	1.3.6.1.4.1.5624.2.1.90	ENTERASYS-CONFIG-MAN-MIB	no	no	yes
SecureStack A2H254-16	1.3.6.1.4.1.5624.2.1.95	ENTERASYS-CONFIG-MAN-MIB	no	no	yes
SecureStack B2G124-24	1.3.6.1.4.1.5624.2.2.314	ENTERASYS-CONFIG-MAN-MIB	no	no	yes
SecureStack B2G124-48	1.3.6.1.4.1.5624.2.2.315	ENTERASYS-CONFIG-MAN-MIB	no	no	yes
SecureStack B2G124-48P	1.3.6.1.4.1.5624.2.2.316	ENTERASYS-CONFIG-MAN-MIB	no	no	yes
SecureStack B2H124-48	1.3.6.1.4.1.5624.2.2.317	ENTERASYS-CONFIG-MAN-MIB	no	no	yes
SecureStack B2H124-48P	1.3.6.1.4.1.5624.2.2.318	ENTERASYS-CONFIG-MAN-MIB	no	no	yes
SecureStack B3G124-24	1.3.6.1.4.1.5624.2.1.100	ENTERASYS-CONFIG-MAN-MIB	no	no	yes
SecureStack B3G124-24P	1.3.6.1.4.1.5624.2.1.101	ENTERASYS-CONFIG-MAN-MIB	no	no	yes
SecureStack B3G124-48	1.3.6.1.4.1.5624.2.1.102	ENTERASYS-CONFIG-MAN-MIB	no	no	yes
SecureStack B3G124-48P	1.3.6.1.4.1.5624.2.1.103	ENTERASYS-CONFIG-MAN-MIB	no	no	yes
SecureStack C2G124-24	1.3.6.1.4.1.5624.2.2.283	ENTERASYS-CONFIG-MAN-MIB	no	no	yes
SecureStack C2G124-48	1.3.6.1.4.1.5624.2.2.284	ENTERASYS-CONFIG-MAN-MIB	no	no	yes
SecureStack C2G124-48P	1.3.6.1.4.1.5624.2.2.287	ENTERASYS-CONFIG-MAN-MIB	no	no	yes
SecureStack C2G134-24P	1.3.6.1.4.1.5624.2.2.350	ENTERASYS-CONFIG-MAN-MIB	no	no	yes
SecureStack C2G170-24	1.3.6.1.4.1.5624.2.2.360	ENTERASYS-CONFIG-MAN-MIB	no	no	yes
SecureStack C2H124-48	1.3.6.1.4.1.5624.2.2.220	ENTERASYS-CONFIG-MAN-MIB	no	no	yes
SecureStack C2H124-48P	1.3.6.1.4.1.5624.2.2.286	ENTERASYS-CONFIG-MAN-MIB	no	no	yes
SecureStack C2K122-24	1.3.6.1.4.1.5624.2.2.285	ENTERASYS-CONFIG-MAN-MIB	no	no	yes
SecureStack C3G124-24	1.3.6.1.4.1.5624.2.1.96	ENTERASYS-CONFIG-MAN-MIB	no	no	yes

SecureStack C3G124-24P	1.3.6.1.4.1.5624.2.1.97	ENTERASYS-CONFIG-MAN-MIB	no	no	yes
SecureStack C3G124-48	1.3.6.1.4.1.5624.2.1.98	ENTERASYS-CONFIG-MAN-MIB	no	no	yes
SecureStack C3G124-48P	1.3.6.1.4.1.5624.2.1.99	ENTERASYS-CONFIG-MAN-MIB	no	no	yes
XSR-1805	1.3.6.1.4.1.5624.2.1.32	ENTERASYS-CONFIG-MAN-MIB	no	no	yes
XSR-1850	1.3.6.1.4.1.5624.2.1.45	ENTERASYS-CONFIG-MAN-MIB	no	no	yes
XSR-1800	1.3.6.1.4.1.5624.2.1	ENTERASYS-CONFIG-MAN-MIB	no	no	yes

Enterasys/Riverstone SSR Supported Devices

The following table lists Enterasys/Riverstone SSR devices supported by DX NetOps Spectrum Network Configuration Manager. The table provides examples. For the most up-to-date information on device support, access the CA Device Certification database.

Device Name	Sys OID	SNMP/TFTP Support	Telnet Support	SSH Support	Perl Support
DEC 8000	1.3.6.1.4.1.36.2.15.30.1	CTRON-SSR-CONFIG-MIB	no	no	yes
DEC 8600	1.3.6.1.4.1.36.2.15.30.2	CTRON-SSR-CONFIG-MIB	no	no	yes
DEC 2000	1.3.6.1.4.1.36.2.15.30.3	CTRON-SSR-CONFIG-MIB	no	no	yes
OLI-8000	1.3.6.1.4.1.285.9.25	CTRON-SSR-CONFIG-MIB	no	no	yes
OLI-8600	1.3.6.1.4.1.285.9.26	CTRON-SSR-CONFIG-MIB	no	no	yes
OLI-2000	1.3.6.1.4.1.285.9.27	CTRON-SSR-CONFIG-MIB	no	no	yes
CPQ-8000	1.3.6.1.4.1.232.134.1.1	CTRON-SSR-CONFIG-MIB	no	no	yes
CPQ-8600	1.3.6.1.4.1.232.134.1.2	CTRON-SSR-CONFIG-MIB	no	no	yes
CPQ-2000	1.3.6.1.4.1.232.134.1.3	CTRON-SSR-CONFIG-MIB	no	no	yes
6-SSRM-02	1.3.6.1.4.1.52.3.9.33.4.1	CTRON-SSR-CONFIG-MIB	no	no	yes
RS-8000	1.3.6.1.4.1.5567.1.1.1	CTRON-SSR-CONFIG-MIB	no	no	yes
RS-8600	1.3.6.1.4.1.5567.1.1.2	CTRON-SSR-CONFIG-MIB	no	no	yes
RS-2000	1.3.6.1.4.1.5567.1.1.3	CTRON-SSR-CONFIG-MIB	no	no	yes
RS-2100	1.3.6.1.4.1.5567.1.1.4	CTRON-SSR-CONFIG-MIB	no	no	yes
RS-3000	1.3.6.1.4.1.5567.1.1.5	CTRON-SSR-CONFIG-MIB	no	no	yes
IA-1100	1.3.6.1.4.1.5567.1.1.22	CTRON-SSR-CONFIG-MIB	no	no	yes
IA-1200	1.3.6.1.4.1.5567.1.1.23	CTRON-SSR-CONFIG-MIB	no	no	yes
RS-1000	1.3.6.1.4.1.5567.1.1.8	CTRON-SSR-CONFIG-MIB	no	no	yes
IA-1500	1.3.6.1.4.1.5567.1.1.27	CTRON-SSR-CONFIG-MIB	no	no	yes
SSR-8000	1.3.6.1.4.1.52.3.9.20.1.3	CTRON-SSR-CONFIG-MIB	no	no	yes
SSR-8600	1.3.6.1.4.1.52.3.9.20.1.4	CTRON-SSR-CONFIG-MIB	no	no	yes
SSR-2000	1.3.6.1.4.1.52.3.9.33.1.1	CTRON-SSR-CONFIG-MIB	no	no	yes
SSR-2100	1.3.6.1.4.1.52.3.9.33.1.3	CTRON-SSR-CONFIG-MIB	no	no	yes

IA-1000	1.3.6.1.4.1.52.3.9.33.2.8	CTRON-SSR-CONFIG-MIB	no	no	yes
IA-2000	1.3.6.1.4.1.52.3.9.33.2.9	CTRON-SSR-CONFIG-MIB	no	no	yes
XP-2400	1.3.6.1.4.1.5624.2.1.42	CTRON-SSR-CONFIG-MIB	no	no	yes
RS-32000	1.3.6.1.4.1.5567.1.1.6	CTRON-SSR-CONFIG-MIB	no	no	yes
RS-38000	1.3.6.1.4.1.5567.1.1.9	CTRON-SSR-CONFIG-MIB	no	no	yes
SSR-32000	1.3.6.1.4.1.52.10.2	CTRON-SSR-CONFIG-MIB	no	no	yes
ER16	1.3.6.1.4.1.5624.2.1.23	CTRON-SSR-CONFIG-MIB	no	no	yes
BE2800	1.3.6.1.4.1.1456.3.2	CTRON-SSR-CONFIG-MIB	no	no	yes
Terayon Router	1.3.6.1.4.1.1456.3.3	CTRON-SSR-CONFIG-MIB	no	no	yes
5-SSRM-02	1.3.6.1.4.1.5624.2.1.24	CTRON-SSR-CONFIG-MIB	no	no	yes

Extreme Supported Devices

The following table lists Extreme devices supported by DX NetOps Spectrum Network Configuration Manager. The table provides examples. For the most up-to-date information on device support, access the CA Device Certification database.

Device Name	Sys OID	SNMP/TFTP Support	Telnet Support	SSH Support	Perl Support
Alpine 3802	1.3.6.1.4.1.1916.2.26	EXTREME-FILETRANSFER-MIB	no	no	yes
Alpine 3804	1.3.6.1.4.1.1916.2.20	EXTREME-FILETRANSFER-MIB	no	no	yes
Alpine 3808	1.3.6.1.4.1.1916.2.17	EXTREME-FILETRANSFER-MIB	no	no	yes
Altitude 300	1.3.6.1.4.1.1916.2.86	EXTREME-FILETRANSFER-MIB	no	no	yes
Altitude 350	1.3.6.1.4.1.1916.2.75	EXTREME-FILETRANSFER-MIB	no	no	yes
BlackDiamond 6800	1.3.6.1.4.1.1916.2.8	EXTREME-FILETRANSFER-MIB	no	no	yes
BlackDiamond 6804	1.3.6.1.4.1.1916.2.27	EXTREME-FILETRANSFER-MIB	no	no	yes
BlackDiamond 6808	1.3.6.1.4.1.1916.2.11	EXTREME-FILETRANSFER-MIB	no	no	yes
BlackDiamond 6816	1.3.6.1.4.1.1916.2.24	EXTREME-FILETRANSFER-MIB	no	no	yes
BlackDiamond 8806	1.3.6.1.4.1.1916.2.74	EXTREME-FILETRANSFER-MIB	no	no	yes
BlackDiamond 8810	1.3.6.1.4.1.1916.2.62	EXTREME-FILETRANSFER-MIB	no	no	yes
BlackDiamond 10808	1.3.6.1.4.1.1916.2.56	EXTREME-FILETRANSFER-MIB	no	no	yes
BlackDiamond 12802	1.3.6.1.4.1.1916.2.85	EXTREME-FILETRANSFER-MIB	no	no	yes
BlackDiamond 12804	1.3.6.1.4.1.1916.2.77	EXTREME-FILETRANSFER-MIB	no	no	yes

EnetSwitch 24Port	1.3.6.1.4.1.1916.2.23	EXTREME-FILETRANSFER-MIB	no	no	yes
Sentriant CE150	1.3.6.1.4.1.1916.2.83	EXTREME-FILETRANSFER-MIB	no	no	yes
Summit 1	1.3.6.1.4.1.1916.2.1	EXTREME-FILETRANSFER-MIB	no	no	yes
Summit 1iSX	1.3.6.1.4.1.1916.2.19	EXTREME-FILETRANSFER-MIB	no	no	yes
Summit 1iTX	1.3.6.1.4.1.1916.2.14	EXTREME-FILETRANSFER-MIB	no	no	yes
Summit 2	1.3.6.1.4.1.1916.2.2	EXTREME-FILETRANSFER-MIB	no	no	yes
Summit 3	1.3.6.1.4.1.1916.2.3	EXTREME-FILETRANSFER-MIB	no	no	yes
Summit 4	1.3.6.1.4.1.1916.2.4	EXTREME-FILETRANSFER-MIB	no	no	yes
Summit 4FX	1.3.6.1.4.1.1916.2.5	EXTREME-FILETRANSFER-MIB	no	no	yes
Summit 5i	1.3.6.1.4.1.1916.2.15	EXTREME-FILETRANSFER-MIB	no	no	yes
Summit 5iLX	1.3.6.1.4.1.1916.2.21	EXTREME-FILETRANSFER-MIB	no	no	yes
Summit 5iTX	1.3.6.1.4.1.1916.2.22	EXTREME-FILETRANSFER-MIB	no	no	yes
Summit 7iSX	1.3.6.1.4.1.1916.2.12	EXTREME-FILETRANSFER-MIB	no	no	yes
Summit 7iTX	1.3.6.1.4.1.1916.2.13	EXTREME-FILETRANSFER-MIB	no	no	yes
Summit 24	1.3.6.1.4.1.1916.2.7	EXTREME-FILETRANSFER-MIB	no	no	yes
Summit 24e2SX	1.3.6.1.4.1.1916.2.41	EXTREME-FILETRANSFER-MIB	no	no	yes
Summit 24e2TX	1.3.6.1.4.1.1916.2.40	EXTREME-FILETRANSFER-MIB	no	no	yes
Summit 24e3	1.3.6.1.4.1.1916.2.25	EXTREME-FILETRANSFER-MIB	no	no	yes
Summit 48	1.3.6.1.4.1.1916.2.6	EXTREME-FILETRANSFER-MIB	no	no	yes
Summit 48i	1.3.6.1.4.1.1916.2.16	EXTREME-FILETRANSFER-MIB	no	no	yes
Summit 48i1u	1.3.6.1.4.1.1916.2.28	EXTREME-FILETRANSFER-MIB	no	no	yes
Summit 200-24	1.3.6.1.4.1.1916.2.53	EXTREME-FILETRANSFER-MIB	no	no	yes
Summit 200-24fx	1.3.6.1.4.1.1916.2.70	EXTREME-FILETRANSFER-MIB	no	no	yes
Summit 200-48	1.3.6.1.4.1.1916.2.54	EXTREME-FILETRANSFER-MIB	no	no	yes

Summit 300-24	1.3.6.1.4.1.1916.2.61	EXTREME-FILETRANSFER-MIB	no	no	yes
Summit 300-48	1.3.6.1.4.1.1916.2.55	EXTREME-FILETRANSFER-MIB	no	no	yes
Summit 400-24	1.3.6.1.4.1.1916.2.59	EXTREME-FILETRANSFER-MIB	no	no	yes
Summit 400-24p	1.3.6.1.4.1.1916.2.64	EXTREME-FILETRANSFER-MIB	no	no	yes
Summit 400-24t	1.3.6.1.4.1.1916.2.63	EXTREME-FILETRANSFER-MIB	no	no	yes
Summit 400-48t	1.3.6.1.4.1.1916.2.58	EXTREME-FILETRANSFER-MIB	no	no	yes
Summit Px1	1.3.6.1.4.1.1916.2.30	EXTREME-FILETRANSFER-MIB	no	no	yes
Summit Ver2Stack	1.3.6.1.4.1.1916.2.93	EXTREME-FILETRANSFER-MIB	no	no	yes
Summit X250-24p	1.3.6.1.4.1.1916.2.89	EXTREME-FILETRANSFER-MIB	no	no	yes
Summit X250-24t	1.3.6.1.4.1.1916.2.88	EXTREME-FILETRANSFER-MIB	no	no	yes
Summit X250-24x	1.3.6.1.4.1.1916.2.90	EXTREME-FILETRANSFER-MIB	no	no	yes
Summit X250-48p	1.3.6.1.4.1.1916.2.92	EXTREME-FILETRANSFER-MIB	no	no	yes
Summit X250-48t	1.3.6.1.4.1.1916.2.91	EXTREME-FILETRANSFER-MIB	no	no	yes
Summit X450-24t	1.3.6.1.4.1.1916.2.66	EXTREME-FILETRANSFER-MIB	no	no	yes
Summit X450-24x	1.3.6.1.4.1.1916.2.65	EXTREME-FILETRANSFER-MIB	no	no	yes
Summit X450a-24t	1.3.6.1.4.1.1916.2.71	EXTREME-FILETRANSFER-MIB	no	no	yes
Summit X450a-24tDC	1.3.6.1.4.1.1916.2.80	EXTREME-FILETRANSFER-MIB	no	no	yes
Summit X450a-24x	1.3.6.1.4.1.1916.2.84	EXTREME-FILETRANSFER-MIB	no	no	yes
Summit X450a-24xDC	1.3.6.1.4.1.1916.2.82	EXTREME-FILETRANSFER-MIB	no	no	yes
Summit X450a-48t	1.3.6.1.4.1.1916.2.76	EXTREME-FILETRANSFER-MIB	no	no	yes
Summit X450a-48tDC	1.3.6.1.4.1.1916.2.87	EXTREME-FILETRANSFER-MIB	no	no	yes
Summit X450e-24p	1.3.6.1.4.1.1916.2.72	EXTREME-FILETRANSFER-MIB	no	no	yes
Summit X450e-48p	1.3.6.1.4.1.1916.2.79	EXTREME-FILETRANSFER-MIB	no	no	yes
SummitStack	1.3.6.1.4.1.1916.2.67	EXTREME-FILETRANSFER-MIB	no	no	yes

SummitWM 100	1.3.6.1.4.1.1916.2.68	EXTREME-FILETRANSFER-MIB	no	no	yes
SummitWM 200	1.3.6.1.4.1.1916.2.94	EXTREME-FILETRANSFER-MIB	no	no	yes
SummitWM 1000	1.3.6.1.4.1.1916.2.69	EXTREME-FILETRANSFER-MIB	no	no	yes
SummitWM 2000	1.3.6.1.4.1.1916.2.95	EXTREME-FILETRANSFER-MIB	no	no	yes

Foundry Supported Devices

The following table lists Foundry devices supported by DX NetOps Spectrum Network Configuration Manager. The table provides examples. For the most up-to-date information on device support, access the CA Device Certification database.

Device Name	Sys OID	SNMP/TFTP Support	Telnet Support	SSH Support	Perl Support
BigIronMG8Sw	1.3.6.1.4.1.1991.1.3.32.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
BigIronMG8Rt	1.3.6.1.4.1.1991.1.3.32.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
BigIronRX4Rt	1.3.6.1.4.1.1991.1.3.40.3.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
BigIronRX4Sw	1.3.6.1.4.1.1991.1.3.40.3.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
BigIronRX8Rt	1.3.6.1.4.1.1991.1.3.40.2.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
BigIronRX8Sw	1.3.6.1.4.1.1991.1.3.40.2.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
BigIronRX16Rt	1.3.6.1.4.1.1991.1.3.40.1.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
BigIronRX16Sw	1.3.6.1.4.1.1991.1.3.40.1.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
BigIronSXL3Sw	1.3.6.1.4.1.1991.1.3.37.1.3	FOUNDRY-SN-AGENT-MIB	no	no	yes
BigIronSXRt	1.3.6.1.4.1.1991.1.3.37.1.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
BigIronSXSw	1.3.6.1.4.1.1991.1.3.37.1.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
BIron4000Rt	1.3.6.1.4.1.1991.1.3.6.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
BIron4000SI	1.3.6.1.4.1.1991.1.3.6.3	FOUNDRY-SN-AGENT-MIB	no	no	yes
BIron4000Sw	1.3.6.1.4.1.1991.1.3.6.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
BIron8000Rt	1.3.6.1.4.1.1991.1.3.7.2	FOUNDRY-SN-AGENT-MIB	no	no	yes

BIron8000SI	1.3.6.1.4.1.1991.1.3.7.3	FOUNDRY-SN-AGENT-MIB	no	no	yes
BIron15000Rt	1.3.6.1.4.1.1991.1.3.14.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
BIron15000SI	1.3.6.1.4.1.1991.1.3.14.3	FOUNDRY-SN-AGENT-MIB	no	no	yes
BIron8000Sw	1.3.6.1.4.1.1991.1.3.7.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
BIron15000Sw	1.3.6.1.4.1.1991.1.3.14.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FastIronBBSw	1.3.6.1.4.1.1991.1.3.1.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
FastIron2Rt	1.3.6.1.4.1.1991.1.3.8.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
FastIron2Sw	1.3.6.1.4.1.1991.1.3.8.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FastIron3Rt	1.3.6.1.4.1.1991.1.3.16.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
FastIron3Sw	1.3.6.1.4.1.1991.1.3.16.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FastIronWGSw	1.3.6.1.4.1.1991.1.3.1.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FES2402Sw	1.3.6.1.4.1.1991.1.3.25.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FES2402Rt	1.3.6.1.4.1.1991.1.3.25.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
FES4802Rt	1.3.6.1.4.1.1991.1.3.26.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
FES4802Sw	1.3.6.1.4.1.1991.1.3.26.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FES9604Rt	1.3.6.1.4.1.1991.1.3.27.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
FES9604Sw	1.3.6.1.4.1.1991.1.3.27.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FES12GCFRt	1.3.6.1.4.1.1991.1.3.28.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
FES12GCFSw	1.3.6.1.4.1.1991.1.3.28.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FES2402POERt	1.3.6.1.4.1.1991.1.3.29.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
FES2402POESw	1.3.6.1.4.1.1991.1.3.29.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FES4802POERt	1.3.6.1.4.1.1991.1.3.30.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
FES4802POESw	1.3.6.1.4.1.1991.1.3.30.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX424Rt	1.3.6.1.4.1.1991.1.3.34.1.1.1.2	FOUNDRY-SN-AGENT-MIB	no	no	yes

FESX424Sw	1.3.6.1.4.1.1991.1.3.34.1.1.1.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX424PremRt	1.3.6.1.4.1.1991.1.3.34.1.1.2.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX424PremSw	1.3.6.1.4.1.1991.1.3.34.1.1.2.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX424P1XGPremSw	1.3.6.1.4.1.1991.1.3.34.1.2.2.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX424P1XGRt	1.3.6.1.4.1.1991.1.3.34.1.2.1.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX424P1XGSw	1.3.6.1.4.1.1991.1.3.34.1.2.1.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX424P1XGPremRt	1.3.6.1.4.1.1991.1.3.34.1.2.2.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX424P2XGRt	1.3.6.1.4.1.1991.1.3.34.1.3.1.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX424P2XGSw	1.3.6.1.4.1.1991.1.3.34.1.3.1.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX424P2XGPremRt	1.3.6.1.4.1.1991.1.3.34.1.3.2.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX424P2XGPremSw	1.3.6.1.4.1.1991.1.3.34.1.3.2.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX448Rt	1.3.6.1.4.1.1991.1.3.34.2.1.1.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX448Sw	1.3.6.1.4.1.1991.1.3.34.2.1.1.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX448PremRt	1.3.6.1.4.1.1991.1.3.34.2.1.2.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX448PremSw	1.3.6.1.4.1.1991.1.3.34.2.1.2.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX448P1XGSw	1.3.6.1.4.1.1991.1.3.34.2.2.1.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX448P1XGRt	1.3.6.1.4.1.1991.1.3.34.2.2.1.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX448P1XGPremRt	1.3.6.1.4.1.1991.1.3.34.2.2.2.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX448P1XGPremSw	1.3.6.1.4.1.1991.1.3.34.2.2.2.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX448P2XGRt	1.3.6.1.4.1.1991.1.3.34.2.3.1.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX448P2XGSw	1.3.6.1.4.1.1991.1.3.34.2.3.1.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX448P2XGPremRt	1.3.6.1.4.1.1991.1.3.34.2.3.2.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX448P2XGPremSw	1.3.6.1.4.1.1991.1.3.34.2.3.2.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX424FiberRt	1.3.6.1.4.1.1991.1.3.34.3.1.1.2	FOUNDRY-SN-AGENT-MIB	no	no	yes

FESX424FiberSw	1.3.6.1.4.1.1991.1.3.34.3.1.1.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX424FiberPremRt	1.3.6.1.4.1.1991.1.3.34.3.1.2.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX424FiberPremSw	1.3.6.1.4.1.1991.1.3.34.3.1.2.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX424FiberP1XGRt	1.3.6.1.4.1.1991.1.3.34.3.2.1.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX424FiberP1XGSw	1.3.6.1.4.1.1991.1.3.34.3.2.1.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX424FiberP1XGPremRt	1.3.6.1.4.1.1991.1.3.34.3.2.2.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX424FiberP1XGPremSw	1.3.6.1.4.1.1991.1.3.34.3.2.2.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX424FiberP2XGRt	1.3.6.1.4.1.1991.1.3.34.3.3.1.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX424FiberP2XGSw	1.3.6.1.4.1.1991.1.3.34.3.3.1.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX424FiberP2XGPremRt	1.3.6.1.4.1.1991.1.3.34.3.3.2.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX424FiberP2XGPremSw	1.3.6.1.4.1.1991.1.3.34.3.3.2.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX448FiberRt	1.3.6.1.4.1.1991.1.3.34.4.1.1.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX448FiberSw	1.3.6.1.4.1.1991.1.3.34.4.1.1.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX448FiberPremRt	1.3.6.1.4.1.1991.1.3.34.4.1.2.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX448FiberPremSw	1.3.6.1.4.1.1991.1.3.34.4.1.2.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX448FiberP1XGRt	1.3.6.1.4.1.1991.1.3.34.4.2.1.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX448FiberP1XGSw	1.3.6.1.4.1.1991.1.3.34.4.2.1.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX448FiberP1XGPremRt	1.3.6.1.4.1.1991.1.3.34.4.2.2.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX448FiberP1XGPremSw	1.3.6.1.4.1.1991.1.3.34.4.2.2.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX448FiberP2XGRt	1.3.6.1.4.1.1991.1.3.34.4.3.1.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX448FiberP2XGSw	1.3.6.1.4.1.1991.1.3.34.4.3.1.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX448FiberP2XGPremRt	1.3.6.1.4.1.1991.1.3.34.4.3.2.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX448FiberP2XGPremSw	1.3.6.1.4.1.1991.1.3.34.4.3.2.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX424POERt	1.3.6.1.4.1.1991.1.3.34.5.1.1.2	FOUNDRY-SN-AGENT-MIB	no	no	yes

FESX424POESw	1.3.6.1.4.1.1991.1.3.34.5.1.1.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX424POEPremRt	1.3.6.1.4.1.1991.1.3.34.5.1.2.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX424POEPremSw	1.3.6.1.4.1.1991.1.3.34.5.1.2.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX424POEP1XGSw	1.3.6.1.4.1.1991.1.3.34.5.2.1.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX424POEP1XGRt	1.3.6.1.4.1.1991.1.3.34.5.2.1.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX424POEP1XGPremRt	1.3.6.1.4.1.1991.1.3.34.5.2.2.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX424POEP1XGPremSw	1.3.6.1.4.1.1991.1.3.34.5.2.2.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX424POEP2XGRt	1.3.6.1.4.1.1991.1.3.34.5.3.1.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX424POEP2XGSw	1.3.6.1.4.1.1991.1.3.34.5.3.1.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX424POEP2XGPremRt	1.3.6.1.4.1.1991.1.3.34.5.3.2.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
FESX424POEP2XGPremSw	1.3.6.1.4.1.1991.1.3.34.5.3.2.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FWSX424Rt	1.3.6.1.4.1.1991.1.3.35.1.1.1.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
FWSX424Sw	1.3.6.1.4.1.1991.1.3.35.1.1.1.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FWSX424P1XGRt	1.3.6.1.4.1.1991.1.3.35.1.2.1.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
FWSX424P1XGSw	1.3.6.1.4.1.1991.1.3.35.1.2.1.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FWSX424P2XGRt	1.3.6.1.4.1.1991.1.3.35.1.3.1.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
FWSX424P2XGSw	1.3.6.1.4.1.1991.1.3.35.1.3.1.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FWSX448Rt	1.3.6.1.4.1.1991.1.3.35.2.1.1.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
FWSX448Sw	1.3.6.1.4.1.1991.1.3.35.2.1.1.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FWSX448P1XGRt	1.3.6.1.4.1.1991.1.3.35.2.2.1.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
FWSX448P1XGSw	1.3.6.1.4.1.1991.1.3.35.2.2.1.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FWSX448P2XGRt	1.3.6.1.4.1.1991.1.3.35.2.3.1.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
FWSX448P2XGSw	1.3.6.1.4.1.1991.1.3.35.2.3.1.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FIron2GCRt	1.3.6.1.4.1.1991.1.3.12.2	FOUNDRY-SN-AGENT-MIB	no	no	yes

FIron2GCSw	1.3.6.1.4.1.1991.1.3.12.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FIron2PlusRt	1.3.6.1.4.1.1991.1.3.9.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
FIron2PlusSw	1.3.6.1.4.1.1991.1.3.9.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FIron3GCRt	1.3.6.1.4.1.1991.1.3.17.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
FIron3GCSw	1.3.6.1.4.1.1991.1.3.17.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FIron400Sw	1.3.6.1.4.1.1991.1.3.22.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FIron400Rt	1.3.6.1.4.1.1991.1.3.22.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
FIron4802Rt	1.3.6.1.4.1.1991.1.3.21.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
FIron4802SI	1.3.6.1.4.1.1991.1.3.21.3	FOUNDRY-SN-AGENT-MIB	no	no	yes
FIron4802Sw	1.3.6.1.4.1.1991.1.3.21.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FIron800Rt	1.3.6.1.4.1.1991.1.3.23.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
FIron800Sw	1.3.6.1.4.1.1991.1.3.23.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FIron1500Rt	1.3.6.1.4.1.1991.1.3.24.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
FIron1500Sw	1.3.6.1.4.1.1991.1.3.24.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FIronSXRt	1.3.6.1.4.1.1991.1.3.36.1.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
FIronSXSw	1.3.6.1.4.1.1991.1.3.36.1.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FIronSXL3Sw	1.3.6.1.4.1.1991.1.3.36.1.3	FOUNDRY-SN-AGENT-MIB	no	no	yes
FIronSXPremL3Sw	1.3.6.1.4.1.1991.1.3.36.2.3	FOUNDRY-SN-AGENT-MIB	no	no	yes
FIronSXPremRt	1.3.6.1.4.1.1991.1.3.36.2.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
FIronSXPremSw	1.3.6.1.4.1.1991.1.3.36.2.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FI2PlusGCSw	1.3.6.1.4.1.1991.1.3.13.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
FI2PlusGCRt	1.3.6.1.4.1.1991.1.3.13.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
NetIronRt	1.3.6.1.4.1.1991.1.3.2.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
NetIron40GRt	1.3.6.1.4.1.1991.1.3.33.1	FOUNDRY-SN-AGENT-MIB	no	no	yes

NetIron400Rt	1.3.6.1.4.1.1991.1.3.10.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
NetIron800Rt	1.3.6.1.4.1.1991.1.3.11.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
NIron1500Rt	1.3.6.1.4.1.1991.1.3.15.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
NetIronMLX4Rt	1.3.6.1.4.1.1991.1.3.44.3.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
NetIronMLX16Rt	1.3.6.1.4.1.1991.1.3.44.1.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
NetIronMLX8Rt	1.3.6.1.4.1.1991.1.3.44.2.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
NetIronXMR16000Rt	1.3.6.1.4.1.1991.1.3.41.1.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
NetIronXMR8000Rt	1.3.6.1.4.1.1991.1.3.41.2.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
NetIronXMR4000Rt	1.3.6.1.4.1.1991.1.3.41.3.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
NetIronIMRRt	1.3.6.1.4.1.1991.1.3.39.1.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
NIron4802Rt	1.3.6.1.4.1.1991.1.3.31.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
NIron4802Sw	1.3.6.1.4.1.1991.1.3.31.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
ServerIron	1.3.6.1.4.1.1991.1.3.3.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
ServerIronXL	1.3.6.1.4.1.1991.1.3.3.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
SIron400Rt	1.3.6.1.4.1.1991.1.3.18.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
SIron400Sw	1.3.6.1.4.1.1991.1.3.18.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
SIron800Rt	1.3.6.1.4.1.1991.1.3.19.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
SIron800Sw	1.3.6.1.4.1.1991.1.3.19.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
SIron1500Rt	1.3.6.1.4.1.1991.1.3.20.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
SIron1500Sw	1.3.6.1.4.1.1991.1.3.20.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
SIronXLTCS	1.3.6.1.4.1.1991.1.3.3.3	FOUNDRY-SN-AGENT-MIB	no	no	yes
SIronLS100Rt	1.3.6.1.4.1.1991.1.3.42.9.1.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
SIronLS100Sw	1.3.6.1.4.1.1991.1.3.42.9.1.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
SIronLS300Rt	1.3.6.1.4.1.1991.1.3.42.9.2.2	FOUNDRY-SN-AGENT-MIB	no	no	yes

SIronLS300Sw	1.3.6.1.4.1.1991.1.3.42.9.2.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
SIronTM100Sw	1.3.6.1.4.1.1991.1.3.42.10.1.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
SIronTM100Sw	1.3.6.1.4.1.1991.1.3.42.10.1.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
SIronTM300Sw	1.3.6.1.4.1.1991.1.3.42.10.2.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
SIronTM300Sw	1.3.6.1.4.1.1991.1.3.42.10.2.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
TurboIronSXSw	1.3.6.1.4.1.1991.1.3.38.1.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
TurboIronSXRt	1.3.6.1.4.1.1991.1.3.38.1.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
TurboIronSXL3Sw	1.3.6.1.4.1.1991.1.3.38.1.3	FOUNDRY-SN-AGENT-MIB	no	no	yes
TurboIronSxPremSw	1.3.6.1.4.1.1991.1.3.38.2.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
TurboIronSXPremRt	1.3.6.1.4.1.1991.1.3.38.2.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
TurboIronSXPremL3Sw	1.3.6.1.4.1.1991.1.3.38.2.3	FOUNDRY-SN-AGENT-MIB	no	no	yes
TIron8SIXLG	1.3.6.1.4.1.1991.1.3.5.4	FOUNDRY-SN-AGENT-MIB	no	no	yes
TurboIronRt	1.3.6.1.4.1.1991.1.3.4.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
TurboIronSw	1.3.6.1.4.1.1991.1.3.4.1	FOUNDRY-SN-AGENT-MIB	no	no	yes
TurboIron8Rt	1.3.6.1.4.1.1991.1.3.5.2	FOUNDRY-SN-AGENT-MIB	no	no	yes
TurboIron8SI	1.3.6.1.4.1.1991.1.3.5.3	FOUNDRY-SN-AGENT-MIB	no	no	yes
TurboIron8Sw	1.3.6.1.4.1.1991.1.3.5.1	FOUNDRY-SN-AGENT-MIB	no	no	yes

Juniper Supported Devices

The following table lists Juniper devices supported by DX NetOps Spectrum Network Configuration Manager. The table provides examples. For the most up-to-date information on device support, access the CA Device Certification database.

Device Name	Sys OID	SNMP/TFTP Support	Telnet Support	SSH Support	Perl Support
EX3200	1.3.6.1.4.1.2636.1.1.1.2.30	no	no	yes	no
EX4200	1.3.6.1.4.1.2636.1.1.1.2.31	no	no	yes	no
EX8208	1.3.6.1.4.1.2636.1.1.1.2.32	no	no	yes	no
EX8216	1.3.6.1.4.1.2636.1.1.1.2.33	no	no	yes	no
IRM	1.3.6.1.4.1.2636.1.1.1.2.16	no	no	*yes	yes

J2300	1.3.6.1.4.1.2636.1.1.1.2.13	no	no	*yes	yes
J4300	1.3.6.1.4.1.2636.1.1.1.2.14	no	no	*yes	yes
J6300	1.3.6.1.4.1.2636.1.1.1.2.15	no	no	*yes	yes
M5	1.3.6.1.4.1.2636.1.1.1.2.5	no	no	*yes	yes
M7i	1.3.6.1.4.1.2636.1.1.1.2.10	no	no	*yes	yes
M10	1.3.6.1.4.1.2636.1.1.1.2.4	no	no	*yes	yes
M10i	1.3.6.1.4.1.2636.1.1.1.2.11	no	no	*yes	yes
M20	1.3.6.1.4.1.2636.1.1.1.2.2	no	no	*yes	yes
M40	1.3.6.1.4.1.2636.1.1.1.2.1	no	no	*yes	yes
M40e	1.3.6.1.4.1.2636.1.1.1.2.8	no	no	*yes	yes
M160	1.3.6.1.4.1.2636.1.1.1.2.3	no	no	*yes	yes
M320	1.3.6.1.4.1.2636.1.1.1.2.9	no	no	*yes	yes
T320	1.3.6.1.4.1.2636.1.1.1.2.7	no	no	*yes	yes
T640	1.3.6.1.4.1.2636.1.1.1.2.6	no	no	*yes	yes
TX	1.3.6.1.4.1.2636.1.1.1.2.17	no	no	*yes	yes

*Device must support SSH V2

Lancom Supported Devices

The following table lists Lancom devices supported by DX NetOps Spectrum Network Configuration Manager. Supported devices must be running firmware LCOS 7.58.0045 or above. The table provides examples. For the most up-to-date information on device support, access the CA Device Certification database.

When a Perl script is the only means of communication with the device, the script method is provided.

Device Name	Sys OID	SNMP/TFTP Support	Telnet Support	SSH Support	Perl Support
LANCOM 1721 VPN	1.3.6.1.4.1.2356.500.4.1721	no	no	no	Telnet/ TFTP
LANCOM 1751	1.3.6.1.4.1.2356.1000.1.1751	no	no	no	Telnet/ TFTP
LANCOM 7111	1.3.6.1.4.1.2356.500.2.7111	no	no	no	Telnet/ TFTP
LANCOM 8011	1.3.6.1.4.1.2356.500.2.8011	no	no	no	Telnet/ TFTP

Nortel Baystack Supported Devices

The following table lists Nortel Baystack devices supported by DX NetOps Spectrum Network Configuration Manager. The table provides examples. For the most up-to-date information on device support, access the CA Device Certification database.

Device Name	Sys OID	SNMP/TFTP Support	Telnet Support	SSH Support	Perl Support
BayStack450-24T	1.3.6.1.4.1.45.3.35.1	no	no	*yes	yes

BayStack380-24T	1.3.6.1.4.1.45.3.45.1	no	no	*yes	yes
BayStack420	1.3.6.1.4.1.45.3.43.1	no	no	*yes	yes
BayStack460-24T	1.3.6.1.4.1.45.3.49.1	no	no	*yes	yes
BayStack470-48T	1.3.6.1.4.1.45.3.46.1	no	no	*yes	yes
BayStack425-24T	1.3.6.1.4.1.45.3.57.2	no	no	*yes	yes
BayStack470-24T	1.3.6.1.4.1.45.3.54.1	no	no	*yes	yes
BayStack5510-24T	1.3.6.1.4.1.45.3.52.1	no	no	*yes	yes
BayStack5510-48T	1.3.6.1.4.1.45.3.53.1	no	no	*yes	yes
BayStack5520-24T-PWR	1.3.6.1.4.1.45.3.59.1	no	no	*yes	yes
BayStack5520-48T-PWR	1.3.6.1.4.1.45.3.59.2	no	no	*yes	yes
Nortel ERS 5530-24TFD	1.3.6.1.4.1.45.3.65	no	no	*yes	yes

* Device must support SSH V2

Nortel Passport Supported Devices

The following table lists Nortel Passport devices supported by DX NetOps Spectrum Network Configuration Manager. The table provides examples. For the most up-to-date information on device support, access the CA Device Certification database.

Device Name	Sys OID	SNMP/TFTP Support	Telnet Support	SSH Support	Perl Support
Passport1424T	1.3.6.1.4.1.2272.42	SWL2MGMT-MIB	no	no	yes
Passport1648	1.3.6.1.4.1.2272.43	SWL2MGMT-MIB	no	no	yes
Passport1612	1.3.6.1.4.1.2272.44	SWL2MGMT-MIB	no	no	yes
Passport1624	1.3.6.1.4.1.2272.45	SWL2MGMT-MIB	no	no	yes
Passport8610	1.3.6.1.4.1.2272.30	RAPID-CITY MIB	no	no	yes
Passport8606	1.3.6.1.4.1.2272.31	RAPID-CITY MIB	no	no	yes
Passport8110	1.3.6.1.4.1.2272.32	RAPID-CITY MIB	no	no	yes
Passport8106	1.3.6.1.4.1.2272.33	RAPID-CITY MIB	no	no	yes
Passport8610	1.3.6.1.4.1.2272.37	RAPID-CITY MIB	no	no	yes
IntrWanPE100	1.3.6.1.4.1.2272.40	RAPID-CITY MIB	no	no	yes
Passport8006	1.3.6.1.4.1.2272.280887558	RAPID-CITY MIB	no	no	yes
Passport8010	1.3.6.1.4.1.2272.280887562	RAPID-CITY MIB	no	no	yes

Network Configuration Manager Events

About Network Configuration Manager Events

Events are generated when a configuration change occurs on a device. All events for a particular device during a specified Correlation Event Period, which is set in Configuration Manager General Configuration, are combined. See Configure General Configuration for more information.

Information is correlated based on device traps, Syslog traps and events, Network Configuration Manager internals and any other trap that is mapped to the generic change event. This information varies by device family. Out-of-box support is provided for Cisco CatOS, Cisco IOS, Cisco IOS - SSH Capable, and Juniper JUNOS device families. See Configure Notification Trap Settings for more information on customizing traps for your installation.

Events Generated on the Device

Configuration Change

Event0082101b:{d "%w- %d %m-, %Y - %T"} Configuration Manager - Configuration change detected on device {m} of type {t} on landscape {S 3}. (event [{e}])

Event00821029:{d "%w- %d %m-, %Y - %T"} Configuration Manager - A configuration change notification was received from device {m}. (event [{e}])

Event0082105e:{d "%w- %d %m-, %Y - %T"} Configuration Manager - The running configuration of device {m} on landscape {S 2} is changed. (event [{e}])

Event0082105f:{d "%w- %d %m-, %Y - %T"} Configuration Manager - The running configuration of device {m} on landscape {S 2} is changed. An alarm will be generated on this model. (event [{e}])

Correlation of Configuration Change Events

Event0082105a:{d "%w- %d %m-, %Y - %T"} Configuration Manager - A configuration change notification was received from device {m} of type {t} on landscape {S 1}.

Device trap(s) provided:

1. Device User: {S 2}
2. From: {S 3}
3. On: {S 4}

The following information was provided by SPECTRUM:

Device User: {S 2}

1. Spectrum User: {S 5}
2. NCM Communication Mode: {S 6}
3. Capture Succeeded: {S 7}
4. Capture Error Message: {S 8}
5. Total Number of Line Changes: {I 9}
6. Relevant Number of Line Changes: {I 10}
7. Violated Policies: {S 11}
8. Compliant Policies: {S 12}
9. Model handle of current configuration model: {H 13}
10. Model handle of previous configuration model: {H 14}

Startup and Running Configurations Same/Differ

Event00821024:{d "%w- %d %m-, %Y - %T"} Configuration Manager - The startup configuration of device {m} on landscape {S 3} differs from its running configuration. (event [{e}])

Event00821025:{d "%w- %d %m-, %Y - %T"} Configuration Manager - The startup configuration of device {m} on landscape {S 3} differs from its running configuration. A minor alarm will be generated on this model. (event [{e}])

Event00821026:{d "%w- %d %m-, %Y - %T"} Configuration Manager - The startup configuration of device {m} on landscape {S 3} differs from its running configuration. A major alarm will be generated on this model. (event [{e}])

Event00821027:{d "%w- %d %m-, %Y - %T"} Configuration Manager - The startup configuration of device {m} on landscape {S 3} differs from its running configuration. A critical alarm will be generated on this model. (event [{e}])

Event00821028:{d "%w- %d %m-, %Y - %T"} Configuration Manager - The startup configuration of device {m} on landscape {S 3} is equal to its running configuration. (event [{e}])

Reference and Running Configuration Same/Differ

Event0082105b:{d "%w- %d %m-, %Y - %T"} Configuration Manager - The reference running configuration of device {m} on landscape {S 2} differs from its current running configuration. (event [{e}])

Event0082105c:{d "%w- %d %m-, %Y - %T"} Configuration Manager - The reference running configuration of device {m} on landscape {S 2} differs from its current running configuration. An alarm will be generated on this model. (event [{e}])

Event0082105d:{d "%w- %d %m-, %Y - %T"} Configuration Manager - The reference running configuration of device {m} on landscape {S 2} is equal to its running configuration. (event [{e}])

Device Compliant/Noncompliant with Policy

Event00821016:{d "%w- %d %m-, %Y - %T"} Configuration Manager - Device {m} of type {t} is compliant with policy {S 1} on landscape {S 3}. (event [{e}])

Event00821017:{d "%w- %d %m-, %Y - %T"} Configuration Manager - Device {m} of type {t} is not compliant with policy {S 1} on landscape {S 3}. (event [{e}])

Event00821051:{d "%w- %d %m-, %Y - %T"} Configuration Manager - Unable to verify policy compliance of host configuration on device {m} of type {t} on landscape {S 1}.
Specific error: {S 2} (event [{e}])

Event00821055:{d "%w- %d %m-, %Y - %T"} Configuration Manager - Device {m} of type {t} is no longer in violation of policy {S 1} because the device has been removed from global collection {S 2} on landscape {S 3}. (event [{e}])

Event00821056:{d "%w- %d %m-, %Y - %T"} Configuration Manager - Device {m} of type {t} is no longer in violation of policy {S 1} because the device has been removed from device family {S 2} on landscape {S 3}. (event [{e}])

Event00821057:{d "%w- %d %m-, %Y - %T"} Configuration Manager - Device {m} of type {t} on landscape {S 2} is no longer in violation of policy {S 1} because the policy has been deleted. (event [{e}])

Device Noncompliant with Policy Alarm Generating Events

Event00821020:{d "%w- %d %m-, %Y - %T"} Configuration Manager - Device {m} has violated policy {S 1} on landscape {S 3}. The severity of this violation is minor. (event [{e}])

Event00821021:{d "%w- %d %m-, %Y - %T"} Configuration Manager - Device {m} has violated policy {S 1} on landscape {S 3}. The severity of this violation is major. (event [{e}])

Event00821022:{d "%w- %d %m-, %Y - %T"} Configuration Manager - Device {m} has violated policy {S 1} on landscape {S 3}. The severity of this violation is critical. (event [{e}])

Capture Succeeded/Failed

Event00821000:{d "%w- %d %m-, %Y - %T"} Configuration Manager - Capture succeeded for host configuration file from device {m} of type {t} on landscape {S 1} initiated by user {S 2}. (event [{e}])

Event00821001:{d "%w- %d %m-, %Y - %T"} Configuration Manager - Capture failed for host configuration file from device {m} of type {t} on landscape {S 1} initiated by user {S 2}.
Specific error: {S 3} (event [{e}])

Event00821049:{d "%w- %d %m-, %Y - %T"} Configuration Manager - Capture succeeded for host startup configuration file from device {m} of type {t} on landscape {S 1} initiated by user {u}. (event [{e}])

Event00821050:{d "%w- %d %m-, %Y - %T"} Configuration Manager - Capture failed for host startup configuration file from device {m} of type {t} on landscape {S 1} initiated by user {u}. Specific error: {S 3} (event [{e}])

Upload Succeeded/Failed

Event00821002:{d "%w- %d %m-, %Y - %T"} Configuration Manager - Load succeeded for host configuration on device {m} of type {t} on landscape {S 1} initiated by user {S 2}. (event [{e}])

Event00821003:{d "%w- %d %m-, %Y - %T"} Configuration Manager - Load failed for host configuration on device {m} of type {t} on landscape {S 1} initiated by user {S 2}.
Specific error: {S 4} (event [{e}])

Upload Failed Alarm Generating Events

Event00821035:{d "%w- %d %m-, %Y - %T"} Configuration Manager - Load failed for host configuration on device {m} of type {t} on landscape {S 1} initiated by user {S 2}.
The severity of this failure is minor. (event [{e}])

Event00821036:{d "%w- %d %m-, %Y - %T"} Configuration Manager - Load failed for host configuration on device {m} of type {t} on landscape {S 1} initiated by user {S 2}.
The severity of this failure is major. (event [{e}])

Event00821037:{d "%w- %d %m-, %Y - %T"} Configuration Manager - Load failed for host configuration on device {m} of type {t} on landscape {S 1} initiated by user {S 2}.
The severity of this failure is critical. (event [{e}])

Write to Startup Succeeded/Failed

Event00821018:{d "%w- %d %m-, %Y - %T"} Configuration Manager - Successfully wrote the running configuration to the startup configuration on device {m} of type {t} on landscape {S 1} initiated by {S 2}. (event [{e}])

Event00821019:{d "%w- %d %m-, %Y - %T"} Configuration Manager - The attempt to write the running configuration to the startup configuration on device {m} of type {t} failed on landscape {S 1}. This operation was initiated by {S 2}. Specific Error: {S 3}. (event [{e}])

NCM Enabled/Disabled on Device

Event0082102f:{d "%w- %d %m-, %Y - %T"} Configuration Manager - NCM has been disabled for device {m}. (event [{e}])

Event00821030:{d "%w- %d %m-, %Y - %T"} Configuration Manager - NCM has been enabled for device {m}. (event [{e}])

NCM Disabled, Operation Not Performed

Event00821032:{d "%w- %d %m-, %Y - %T"} Configuration Manager - The requested NCM operation was not performed on model {m} because NCM is disabled on the models device family. (event [{e}])

Event00821033:{d "%w- %d %m-, %Y - %T"} Configuration Manager - The requested NCM operation was not performed on model {m} because NCM is disabled on this model. (event [{e}])

Event00821034:{d "%w- %d %m-, %Y - %T"} Configuration Manager - The requested NCM operation was not performed on model {m} because this model is a proxy model. (event [{e}])

Device Firmware Load

Event00821053:{d "%w- %d %m-, %Y - %T"} Configuration Manager - Firmware load completed successfully on device {m} of type {t} on landscape {S 1}. Firmware script was executed with command line parameters: {S 3} This operation was initiated by {u}. (event [{e}])

Event00821054:{d "%w- %d %m-, %Y - %T"} Configuration Manager - Firmware load failed on device {m} of type {t} on landscape {S 1}. Specific error: {S 2} Firmware script was executed with command line parameters: {S 3} (event [{e}])

Device Added/Removed from Device Family

Event00821058:{d "%w- %d %m-, %Y - %T"} Configuration Manager - Device {m} of type {t} has been added to device family {S 2} on landscape {S 1}. (event [{e}])

Event00821059:{d "%w- %d %m-, %Y - %T"} Configuration Manager - Device {m} of type {t} has been removed from device family {S 2} on landscape {S 1}. (event [{e}])

Events Generated on Policies

Policy Enabled/Disabled

Event00821014:{d "%w- %d %m-, %Y - %T"} Configuration Manager - Policy {m} been enabled by {u}. (event [{e}])

Event00821015:{d "%w- %d %m-, %Y - %T"} Configuration Manager - Policy {m} been disabled by {u}. (event [{e}])

Event00821023:{d "%w- %d %m-, %Y - %T"} Configuration Manager - Policy {S 1} has been disabled. Any alarms previously generated by violations of this policy have been cleared. (event [{e}])

Policy Modified

Event00821011:{d "%w- %d %m-, %Y - %T"} Configuration Manager - Policy {m} has been modified by {u}. (event [{e}])

Policy has Violators

Event00821012:{d "%w- %d %m-, %Y - %T"} Configuration Manager - Policy {m} has a violator on landscape {S 1}. (event [{e}])

Event00821013:{d "%w- %d %m-, %Y - %T"} Configuration Manager - Policy {m} no longer has a violator on landscape {S 1}. (event [{e}])

Violated Policy, Alarm Generating Events

Event0082101d:{d "%w- %d %m-, %Y - %T"} Configuration Manager - Policy {m} has a violator on landscape {S 1}. The severity of this violation is minor. (event [{e}])

Event0082101e:{d "%w- %d %m-, %Y - %T"} Configuration Manager - Policy {m} has a violator on landscape {S 1}. The severity of this violation is major. (event [{e}])

Event0082101f:{d "%w- %d %m-, %Y - %T"} Configuration Manager - Policy {m} has a violator on landscape {S 1}. The severity of this violation is critical. (event [{e}])

Events Generated on Tasks Global Sync, Capture, Upload and Write to Startup

Task Scheduled/Unscheduled

Event00821040:{d "%w- %d %m-, %Y - %T"} Configuration Manager - Task Scheduled - Task {m} of type {t} has been scheduled for {S 1} on landscape {S 2}. (event [{e}])

Event00821041:{d "%w- %d %m-, %Y - %T"} Configuration Manager - Task Unscheduled - Task {m} of type {t} has had the schedule removed on landscape {S 1}. (event [{e}])

Task Started, Stopped, Completed, Partially Completed

Event00821042:{d "%w- %d %m-, %Y - %T"} Configuration Manager - Task Started - Task {m} of type {t} has been started by {S 1} on {I 2} devices on landscape {S 3}. (event [{e}])

Event00821043:{d "%w- %d %m-, %Y - %T"} Configuration Manager - Task Stopping - Task {m} of type {t} was stopped by {S 1} on landscape {S 3}. (event[{e}])

Event00821045:{d "%w- %d %m-, %Y - %T"} Configuration Manager - Task Completed - Task {m} of type {t} has completed with all devices processed on landscape {S 1}. Out of a total of {I 2} devices, {I 3} succeeded and {I 4} failed. (event [{e}])

Event00821044:{d "%w- %d %m-, %Y - %T"} Configuration Manager - Task Partially Completed - Task {m} of type {t} completed with {I 5} unprocessed devices on landscape {S 1}. For a total of {I 2} deivces, {I 3} succeeded, {I 4} failed and {I 5} devices remained unprocessed. (event [{e}])

Task Partially Completed Alarm Generating Events

Event00821046:{d "%w- %d %m-, %Y - %T"} Configuration Manager - Task Partially Completed - Task {m} of type {t} completed with {I 5} unprocessed devices on landscape {S 1}. For a total of {I 2} devices, {I 3} succeeded, {I 4} failed and {I 5} devices remained unprocessed. A minor alarm has been generated. (event [{e}])

Event00821047:{d "%w- %d %m-, %Y - %T"} Configuration Manager - Task Partially Completed - Task {m} of type {t} completed with {I 5} unprocessed devices on landscape {S 1}. For a total of {I 2} devices, {I 3} succeeded, {I 4} failed and {I 5} devices remained unprocessed. A major alarm has been generated. (event [{e}])

Event00821048:{d "%w- %d %m-, %Y - %T"} Configuration Manager - Task Partially Completed - Task {m} of type {t} completed with {I 5} unprocessed devices on landscape {S 1}. For a total of {I 2} devices, {I 3} succeeded, {I 4} failed and {I 5} devices remained unprocessed. A critical alarm has been generated. (event [{e}])

Events Generated on the Configuration Manager Application

Event0082101a:{d "%w- %d %m-, %Y - %T"} Cannot connect to Configuration Manager - Cannot connect to the NCM secure communication daemon on landscape {S 1}. (event [{e}])

Event0082101c:{d "%w- %d %m-, %Y - %T"} Configuration Manager - Cannot create file in Archive Directory. Unable to archive a device configuration. Could not create a file in the archive directory {S 1} on landscape {S 2}. (event [{e}])

Event00821052:{d "%w- %d %m-, %Y - %T"} Configuration Manager - Connection to the ncmservice daemon has been restored. (event [{e}])

Global Unsolicited Notification

Event0082102b:{d "%w- %d %m-, %Y - %T"} Configuration Manager - The ability to respond to unsolicited notifications of configuration change has been globally disabled on all landscapes. (event [{e}])

Event0082102c:{d "%w- %d %m-, %Y - %T"} Configuration Manager - The ability to respond to unsolicited notifications of configuration change has been globally enabled on all landscapes. (event [{e}])

Events Generated on Device Families

Event0082102d:{d "%w- %d %m-, %Y - %T"} Configuration Manager - NCM enabled/disabled for device family {m}. (event [{e}])

Event0082102e:{d "%w- %d %m-, %Y - %T"} Configuration Manager - NCM has been enabled for device family {m}. (event [{e}])

Network Configuration Manager Privileges

This section lists Network Configuration Manager privileges for OneClick users. By default, each privilege is enabled.

> **NOTE**
> See the OneClick Administration section for details about configuring privileges.

- **Network Configuration Manager**
  Allows the administrator to configure the Network Configuration Manager application. This includes configurations performed in the Information tab views from the Configuration Manager node, Device Family nodes, and Network Configuration Manager configuration performed at the individual device level. This also includes the ability to schedule a global synchronization task.
- **Capture Host Configuration**
  Allows the operator to create a bulk capture task or an on-demand capture from the Host Configuration tab.
- **Hide Configuration Changes from Approval Requests**
  Allows the user to decide whether to include configuration content in a workflow approval request.
- **Include Global Collection in NCM Task**
  Allows the user to associate Network Configuration Manager tasks with Global Collections. By having access to a collection, a user will implicitly have access to all members within the collection. With this access, the user can perform any of the tasks, including uploading configurations and loading firmware, on these devices.
- **Load device firmware**
  Allows the operator to upload firmware to devices.
- **Manage NCM Tasks**
  Grants access to the Network Configuration Manager Tasks folder. Access to this folder provides global access to all Network Configuration Manager tasks on all DX NetOps Spectrum landscapes and the ability to start, stop, edit and delete them all.
- **Reload Device**
  Allows operator to reload firmware configuration to a device.
- **Repair Device**
  Allows the operator to upload the specified repair content for a policy with non-compliant devices.
- **Save Host Configuration to Startup**
  Allows the operator to create bulk Save to Startup tasks.
- **Schedule a Reload**
  Allows operator to schedule reload firmware configuration to a device.
- **Schedule NCM Tasks**
  Allows the operator to schedule bulk tasks.
- **Task Approver**
  Controls approval authorization for approval workflow.
  - **ServiceDesk**
    If Approval Workflow Mode is set to ServiceDesk and the user has this privilege, acquiring approval through Service Desk is optional.
  - **OneClick**
    If Approval Workflow Mode is set to OneClick and the user has this privilege, the user can approve his own tasks or tasks that are initiated by others.
- **Upload Host Configuration**
  Allows the operator to create a bulk Upload task or an automatic Upload from the Host Configuration tab.
- **Use Cached Device Authentication**
  Allows the operator to use the username and password as specified in the device family and single device override configurations. The user does not have to enter a username and password each time a task is initiated if this privilege is enabled. When initiating a task (for example, Upload or Save to Startup), the user will be prompted for device authentication if this privilege is disabled.

NOTE
When performing a bulk task, such as an Upload or Save to Startup, without this privilege, the operator is prompted for device authentication once. This same authentication data is then used for all devices for which the bulk operation is performed.

- **View Host Configuration**
  Allows the operator access to host configurations.
- **View NCM Policies**
  Grants access to the Network Configuration Manager Policies folder. Access to this folder provides global access to all Network Configuration Manager policies on all DX NetOps Spectrum landscapes and the ability to edit, enable, disable and delete them all.
  - **Create/Edit NCM Policies**
    Allows the operator ability to create a new policy or edit an existing policy. This privilege does not allow the user to enable a policy.
  - **Enable/Disable NCM Policies**
    Allows the operator to enable, disable, and delete a Network Configuration Manager Policy.
- **View Unmasked Configurations**
  Allows the operator to view content that is blacked-out by the View Mask. The View Mask is on a Device Family and can be overridden at the local device.

Network Configuration Manager Self Certification

From 10.3.1 release, the Network Configuration Manager feature is enhanced to support capturing and managing network configurations for your custom device families by using simple SSH commands. Previously, enabling NCM support for devices did not come with an out of the box NCM support. With this enhancement, users can now enable NCM support for any SSH enabled device by using custom device family and by providing commands in the OneClick view without having to use scripts. You can use the SSH commands to capture and upload the configurations for custom devices which are SSH capable. The SSH commands for respective devices can be obtained from an administrator of the network.

NOTE

We recommend running the bash -login, as a prerequisite, to create the .ssh folder which is required to run the NCM scripts.

The following NCM operations are supported using the SSH commands:

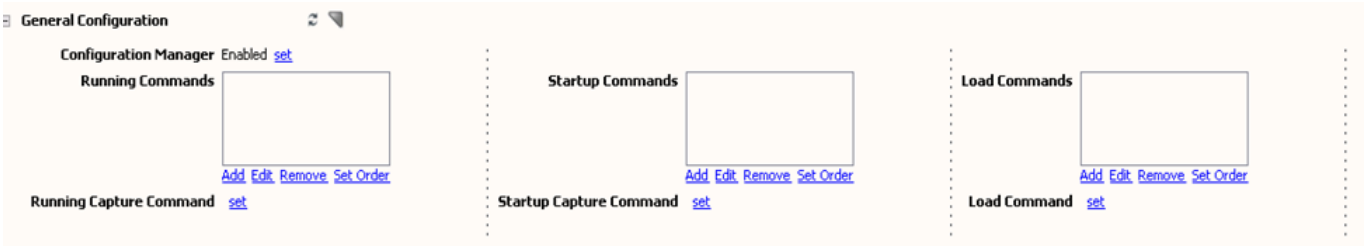
- Capture Startup Configuration
- Capture Running Configuration
- Upload Configuration

NOTE

The existing Perl script mode of configuration is also supported for enabling NCM support for custom devices. Using the SSH-based Perl scripts mode of configuration takes precedence over the SSH commands configuration. For more information about Perl Scripts functionality in NCM, see Using SSH-based Perl Scripts for Network Configuration Manager Operations.

General Configuration for SSH Commands

The following view is added in Spectrum OneClick to support SSH commands. When you select a custom device family in the Explorer tab, you can see the General Configuration section for the respective device family in the Information tab of the Contents panel. Using the General Configuration section, you can configure SSH commands to capture running, startup configurations and upload the modified or saved configurations for a custom device family.



NOTE

For custom devices (which are not supported out-of-the-box), create a custom device family and assign the devices to the family. A device can only belong to a single device family. For instructions, see Create a Custom Device Family section.

General Configuration View

Using the General Configuration view, you can add, edit, or remove commands for running, startup and upload configurations. These commands are also displayed in corresponding attributes of the device family in the Attributes table view of Component Detail panel.

- **Running Commands and Running Capture Command:** The **Running Commands** option allows you to perform the following tasks:
  - Add - To add a command for capturing the running configuration of a device family.
  - Edit- To edit a command.
  - Remove- To remove commands from the running commands box.
  - Set Order- To define the sequence for running the running configuration commands. When you select this option, Set Order pop-up window appears. Select a command and use the up and down arrows to set the order for that command.

  The **Running Capture Command** option allows you to set the command to capture the running conflagration of the selected device family. When you select the Set link, it converts into a drop-down list and allows you to select the command from the list.
- **Startup Commands and Startup Capture Command:** The **Startup Commands** option allows you to perform the following tasks:
  - Add- To add a command for capturing the startup configuration of a device family.
  - Edit- To edit a command.
  - Remove- To remove commands from the startup commands box.
  - Set Order- To define the sequence for running the startup configuration commands. When you select this option, Set Order pop-up window appears. Select a command and use the up and down arrows to set the order for that command.
  - The **Startup Capture Command** option allows you to set the command to capture the startup conflagration of the selected device family. When you select the Set link, it converts into a drop-down list and allows you to select the command from the list.
- **Load Commands and Load Config Start Sequence:** The **Load Commands** option allows you to perform the following tasks:

  - Add- To add a command for capturing the upload configuration of a device family.
  - Edit- To edit a command.
  - Remove- To remove commands from the upload commands box.
  - Set Order- To define the sequence for running upload configuration commands. When you select this option, Set Order pop-up window appears. Select a command and use the up and down arrows to set the order for that command.
- The **Load Config Start Sequence** option allows you to set the command sequence to upload the changed configurations. Spectrum executes sequence of commands until this value (this command is executed before uploading the configuration) and upload the changed configuration to the device then perform the remaining sequences.

Enable NCM Support for Custom Device Families

The following procedure provides end to end scenario to enable NCM support for custom devices (which are not supported out-of-the-box).

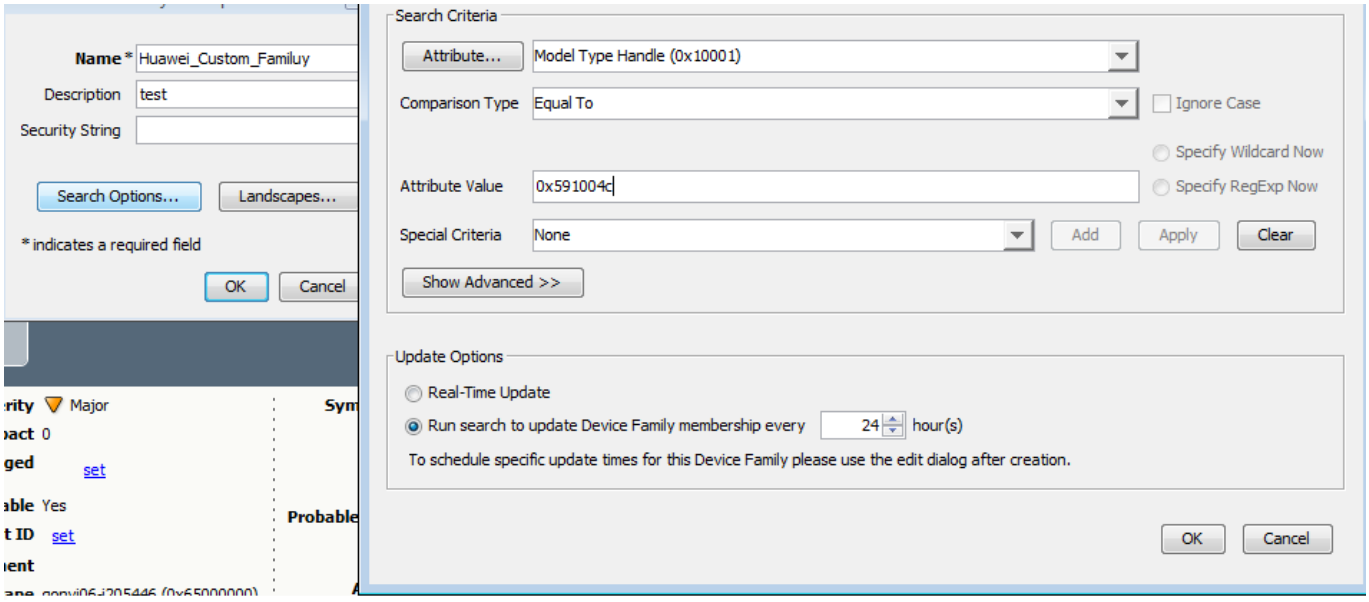
Prerequisites:

- List of commands for running, startup, and upload configurations for a specific device family.
- Create a Custom Device Family based on the model type.

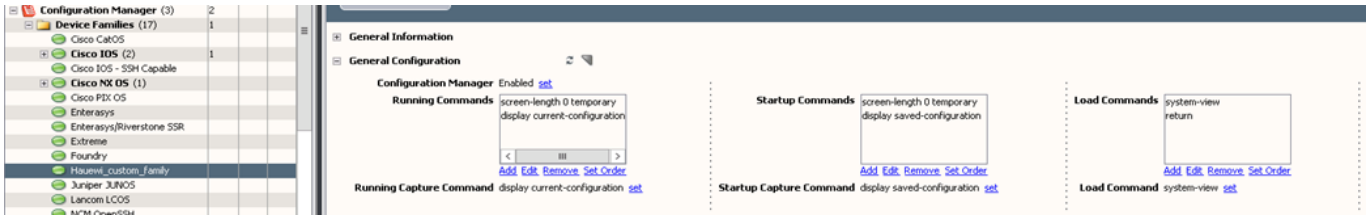
How to enable the support:

Following is an example to explain a step-by-step procedure to enable NCM support for Huawei device model type (0x591004c).

1. Ensure all the prerequisites are met.
2. Create a Custom Device Family based on the model type. In the Create Device Family window, select the Search Options to provide the search settings for the custom device family. This procedure is the same as the search settings for the global collection.
   The following screen explains the search criteria for Huawei model type (0x591004c).

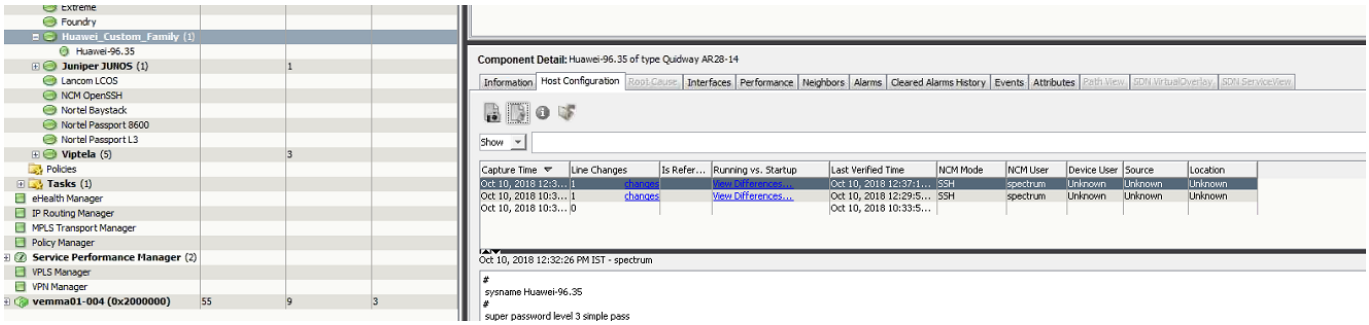


3. Provide the running, startup, and load commands list and corresponding values.



After successfully configuring the commands, Spectrum will enable the NCM support for the selected model type devices.
The default communication mode/primary communication mode is selected as SSH. To know more about communication mode, see Configure Device Family Communication Mode section.

4. Use the Host Configuration tab in the Component Details panel, to capture, and upload the configurations for selected host. The capture functionality gets the commands from the Information tab and executes them in the sequence that is defined there. For more details, see Network Configuration Manager Device-Level Tasks.



NOTE

To compare a startup configuration against a currently running configuration (Running vs Startup) for devices in your network, enable the **Verify Startup Equals Running Configuration** option in the Information tab at the Configuration Manager level.

Migrate devices from out-of-box NCM families

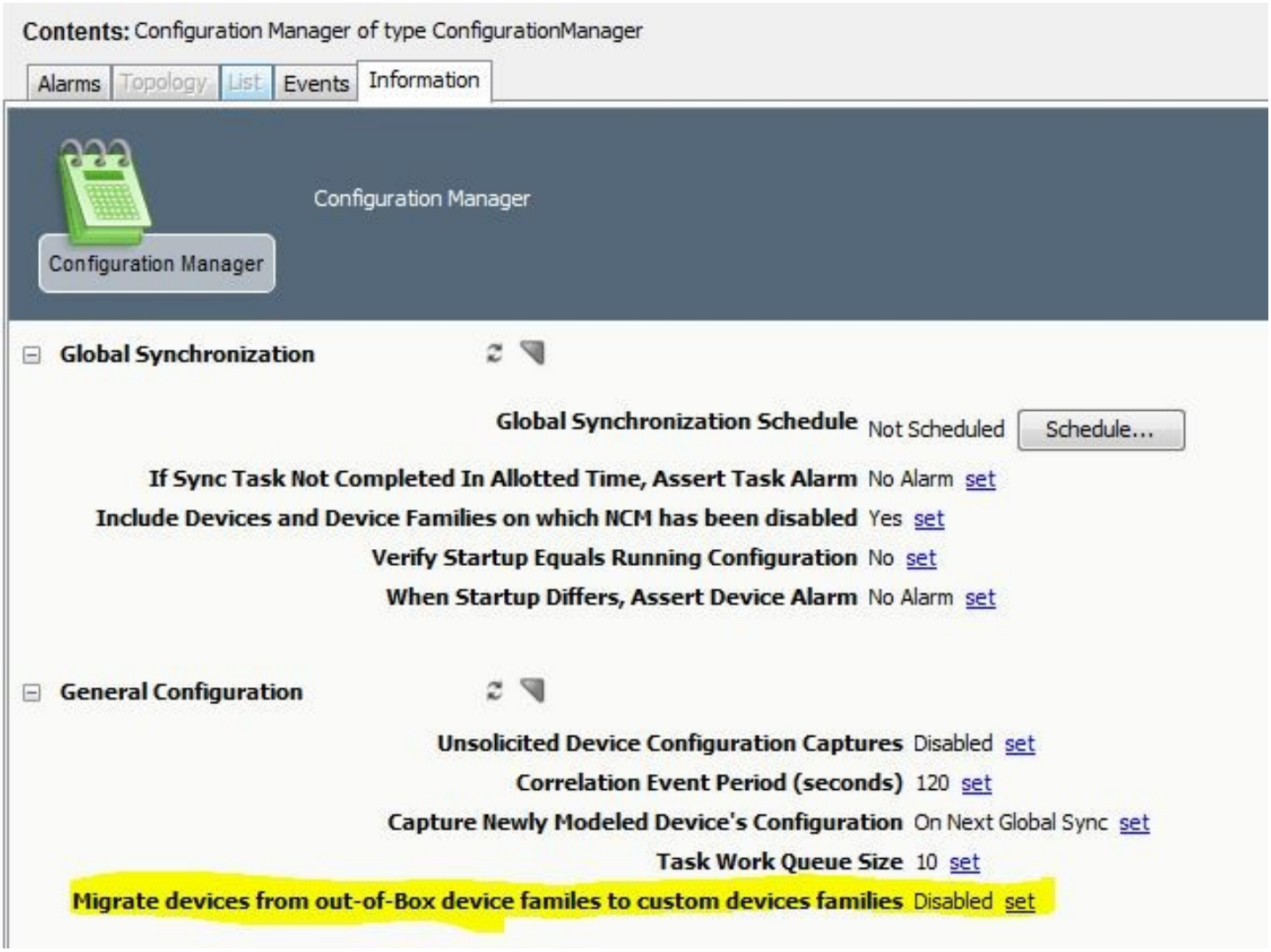
NOTE

Now migrate devices from out-of-box device families to custom device families, with Spectrum 10.3.2 !

This release introduces a new field to migrate out-of-box device families to the custom device families (as shown in the screenshot). To migrate devices from out-of-box device families to custom device families:

1. On the OneClick Console, navigate to the Configuration Manager>Information tab>General Configuration>Migrate devices from out-of-box device families to custom device families.
2. Set this option to 'enable', to use this feature. By default this feature is 'disabled'.

Optimize search settings for these devices by adding attributes and attribute value, when creating their device families. Changing the search criteria moves the existing devices back to the old families and the devices that satisfy the new search criteria are moved to the custom device families.



Non-Persistent Connections Manager

Contents

Getting Started with Non-Persistent Connections

This chapter provides an overview of the non-persistent connections functionality available in DX NetOps Spectrum OneClick and describes the types of links you can manage.

You can manage transient communications links using OneClick's non-persistent connections functionality. Non-persistent connections generally remain in an inactive state and are activated only under certain conditions. They can be activated manually as needed to serve as normal dialup type communications links, or they can be configured to activate automatically in either of the following two scenarios:

- To serve as a backup when failure of a primary connection occurs
- To provide load balancing or extra bandwidth for another overloaded communications link

Non-persistent connections let you see:

- When a link is activated
- Whether a link has failed to activate
- How long a particular link has been active

Events and alarm conditions are generated and logged to maintain accurate link usage statistics. Also, you can view the links as icons so that you can differentiate between normal connections and non-persistent connections and determine a connection's current status at a glance. Within DX NetOps Spectrum these non-persistent connections are represented by the Dialup_Link model type.

Non-Persistent Connection Types

When you create a Dialup_Link model, you choose the type of connection that best suits your needs. The Dialup_Link model type provides functionality for the following three distinct types of non-persistent connections:

- **Dial Backup Link (DBL)**
  A Dial Backup Link (DBL) provides a redundant link if a primary link fails. This type of link provides monitoring support for a one-to-one relationship between primary and secondary links and it provides support for multiple primary and secondary link configurations. Each DBL model can provide monitoring capabilities for multiple primary links using a common secondary interface. This is useful when many remote sites use one phone number at a central site (common interface) to support a non-persistent connection.
- **Bandwidth on Demand Link (BODL)**
  A Bandwidth on Demand Link (BODL) provides extra bandwidth for connections experiencing congestion problems. Links of this type may or may not be active in conjunction with the primary link. BODLs provide the same redundant capabilities as Dial Backup Links.
- **Primary on Demand Link (PODL)**
  A Primary on Demand Link (PODL) is not a redundant connection of any type. These links are activated manually when needed and function as a primary connection. There is no primary-secondary relationship involved.

Create Dialup_Link Models

This section explains how to create a Dialup_Link model to represent a non-persistent connection. Create one Dialup_Link model for each telephone number that remote routers can dial to create a non-persistent connection. Create Dialup_Link models manually. If the remote router can dial one number that one of several host routers or host router interfaces can pick up, one Dial_Up link model represents this connection.

Follow these steps:

1. In the Explorer tab of the OneClick Navigation panel, locate the Universe topology view where the devices involved in the non-persistent connection reside.
   The selected Universe topology view appears in the Topology tab of the Contents panel.
2. In the Topology tab toolbar, click the Create a Model by Type icon 
   The Select Model Type dialog opens.
3. Click the All Model Types tab.
   A list of model types opens.
4. Type **Dialup_Link** in the Filter text box to filter the list to show the Dialup_Link model type only.
5. Select the Dialup_Link model type and click OK.
   The Create Model Of Type Dialup_Link dialog opens.
6. In the Create Model Of Type Dialup_Link dialog, complete the following fields, which describe the device model type that you are modeling.

- **Name**
  Specifies a unique name for the Dialup_Link device you are modeling. This name appears in the Topology view with the icon.
  Limits: 1 to 16 ASCII characters
- **Security String**
  Specifies a DX NetOps Spectrum Security String for this Dialup_Link model.

  NOTE
  For more information, see the OneClick Administration section.
- **Dialup Link Type**
  Specifies the functional type of the Dialup_Link. Possible types are Dial Backup Link, Primary on Demand Link, and Bandwidth on Demand Link.
- **Dialup Protocol Type**
  Specifies the protocol type to use on the Dialup_Link. Possible protocol types include Analog, Switch-56, ISDN, Frame_Relay, and Other. The type appears in the Topology view with the icon.
- **Activation Grace Period (Min)**
  Specifies the time, in minutes, for the secondary link to become active after a primary link failure. If this time period expires before the secondary link is active, a red alarm is generated.
  Default: 3 minutes
- **Deactivation Grace Period (Min)**
  Specifies the time, in minutes, for an active secondary link to deactivate after the failed primary link reactivates. If the secondary link is still active after this grace period expires, a yellow alarm is generated.
  Default: 3 minutes
- **Active Time Until Yellow (Hours)**
  Specifies the number of hours a backup link can be active before a yellow alarm is generated.
- **Active Time Until Orange (Hours)**
  Specifies the number of hours a backup link can be active before an orange alarm is generated.
- **Active Time Until Red (Hours)**
  Specifies the number of hours a backup link can be active before a red alarm is generated.

7. Click OK to create a modeled device icon for the specified device and close the dialog.
   The Dialup_Link model appears in the selected Topology view.

   NOTE
   When first created, the model is in a normal inactive state and has a brown condition.
8. (Optional) Click the Edit mode button in the Topology tab toolbar to move the device icon to a more appropriate location in the Topology view.

   NOTE
   For more information, see the Modeling and Managing Your IT Infrastructure section.

Configuring Non-Persistent Connections

Configuring non-persistent connections involves identifying primary and secondary interfaces on the appropriate remote and host router or routers. Remote routers initiate dialup links. Host routers answer dialup calls from remote routers to establish the link. Management information is obtained by monitoring the remote router or routers.

Dial Backup Links (DBLs) and Bandwidth on Demand Links (BODLs) involve the redundant relationship between a primary link and a secondary link or links. The Dialup_Link model representing a DBL or a BODL monitors a primary link for a possible failure; it also monitors a related secondary link to verify its activation in a backup or supporting role if the primary link fails.

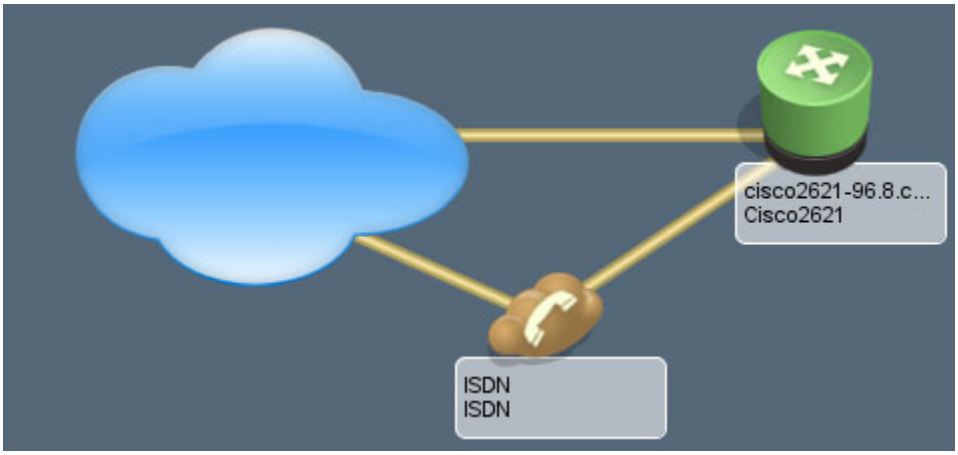
In general, the process to follow for configuring these types of links is as follows:

1. Configure the secondary interface or interfaces on the remote router or routers. This involves resolving the Dialup_Link model to the secondary interface.
   This is the interface that will serve as the dialing end of the dialup connection.
2. Configure the primary interface or interfaces on the remote router or routers.
   If all the primary interfaces go down, the dialup connection is used.
3. If known, configure the secondary interface or interfaces on the host router that will serve as the receiving end of the dialup connection.

> **NOTE**
> Primary on Demand Links (PODLs) are non-persistent connections that are used as transient primary links. You configure these links using this same process, however, since this type of link model does not need to monitor a primary interface, you can ignore the second step.

Basic Non-Persistent Connections

The Non-Persistent Connection solution requires that the primary and secondary interfaces reside on the same remote router. This leads us to the following most basic supported scenario in which only one end of the connection is known:

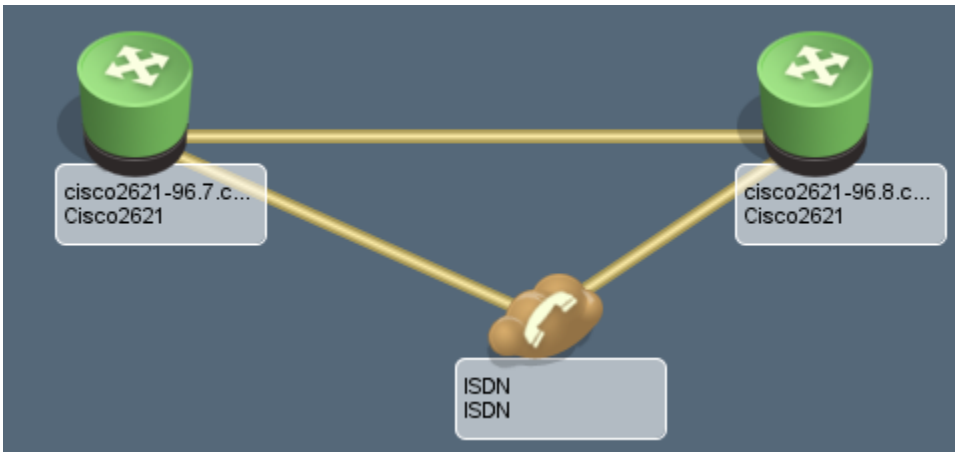


As illustrated by the previous image, in this case, only the remote (dialing) side of the redundant setup is known. As such, you would set up this scenario as described in the following procedure.

To model a basic non-persistent connection when only one end is known

1. Right-click the primary interface on the router and select 'Non-Persistent Connection Setup, Identify NPC Backup For This Interface.'
2. Right-click the backup interface which should be activated if the primary interface is unavailable and select 'Non-Persistent Connection Setup, NPC Backup For *<primary interface>*.'
   The Create Model of Type Dialup_Link dialog appears.
3. Create a Dialup_Link model.

Another basic non-persistent connection is one in which both the initiating and the receiving routers involved in the dialup connection are known, as illustrated by the following image:



To model a basic non-persistent connection when both ends are known

1. Complete all the steps in the previous procedure, "To model a basic non-persistent connection when only one end is known."
2. Right-click the interface model on the remote router that is initiating the dial backup link and select 'Non-Persistent Connection Setup, Start Non-Persistent Connection.'
3. Right-click the interface model on the host router interface that is on the receiving end of the dial backup link and select 'Non-Persistent Connection Setup, Connect With *<interface from step 2*>.'
   This creates a link between the Dialup_Link model and the host router interface on the receiving end of the non-persistent connection.

Dialer Map-Based Dial-on-Demand Routing (DDR) Configurations

This section discusses modeling a non-persistent connection that uses a Dial-on-Demand Routing (DDR) configuration that is map-based.

A DDR configuration that is map-based (often referred to as a legacy DDR configuration) associates a call specification for a single destination and a particular physical interface configuration. The physical interface contains all the configuration information that relates to receiving or making calls, and the bearer channels inherit the physical interface's configuration.

Modeling a Map-Based DDR Configuration

To model a DDR configuration that is map-based, create the appropriate number of Dialup_Link models and then configure the connection for each Dialup_Link model.

To model a map-based DDR configuration

1. Create one Dialup_Link model for each phone number that the remote router or routers could dial to create the non-persistent connections.

   NOTE
   If the remote router can dial one number that can be picked up by one of several host routers or host router interfaces, you can use just one Dial_Up link model to represent this connection.
2. Associate each Dialup_Link model you created with one of the secondary (dialing) interfaces on the remote router:
   a. Right-click the Dialup_Link model and select 'Non-Persistent Connection Setup, Start Non-Persistent Connection.'
   b. Right-click the associated secondary (dialing) interface on the remote router and select 'Non-Persistent Connection Setup, Connect With *<Dialup_Link model>.'*

3. Specify a primary interface for each Dialup_Link model. In the event that the primary interface goes down, the secondary link represented by the Dialup_Link model will activate:
   a. Select the Dialup_Link model and then click the Information tab in the Component Detail panel.
   b. Expand the Primary and Secondary Interface Information subview.
   c. In the Primary Interfaces table, click
      Add  .
   d. Select one or more primary interfaces in the resulting dialog and then click Add.
   e. Repeat for each Dialup_Link model that is part of this non-persistent connection configuration.
4. Associate each Dialup_Link model to the secondary (receiving) interfaces on the host router or routers.
   a. Right-click the Dialup_Link model and select 'Non-Persistent Connection Setup, Start Non-Persistent Connection.'
   b. Right-click the appropriate secondary (receiving) interface on the host router and select 'Non_Persistent Connection Setup, Connect With <*Dialup_Link model*>.'
   c. Repeat for each Dialup_Link model that is part of this non-persistent connection configuration.

Map-Based DDR Connections with Multilink Secondary Interfaces

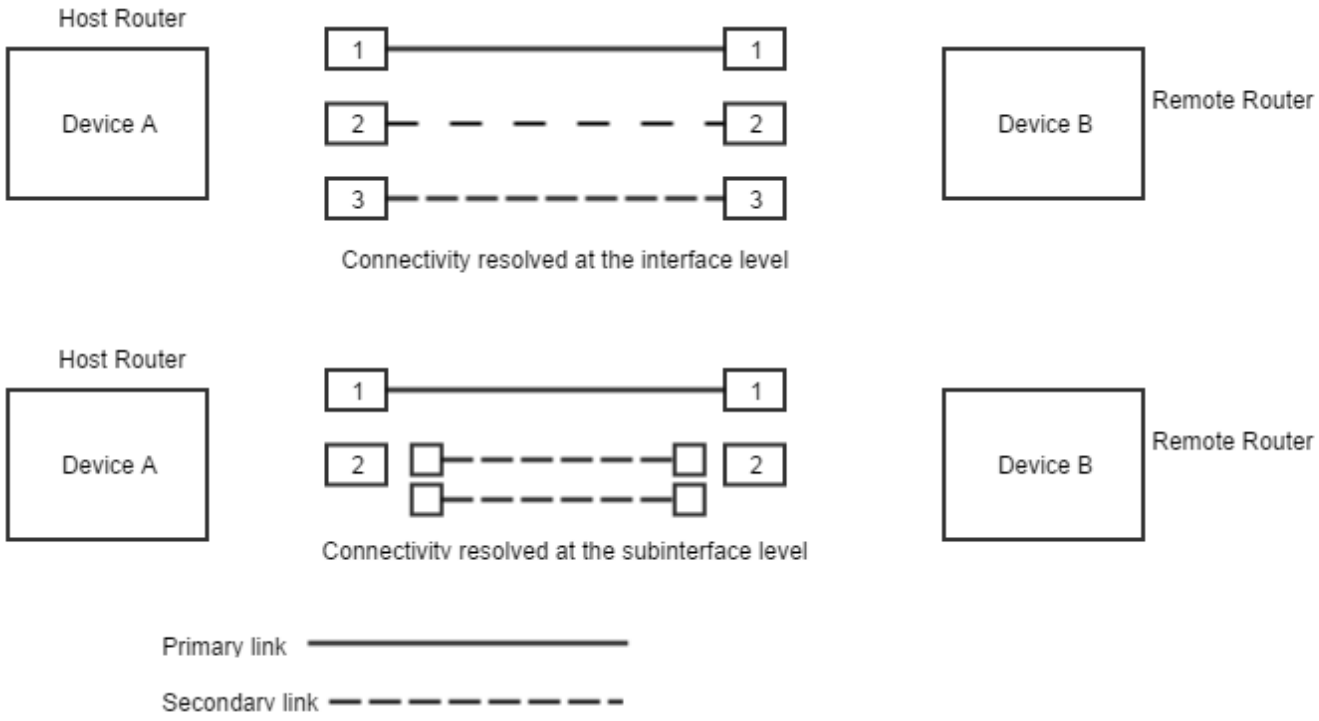
This section discusses modeling a non-persistent connection that uses one Dialup_Link model to specify a secondary link consisting of multiple interfaces that make up one logical link. This secondary link can be initiated by one or more ports on the remote router.

Modeling DDR Map-Based Connections with Multilink Secondary Interfaces

You can use one Dialup_Link model to support multiple secondary interfaces on a single remote router. For example, if your dialup connection uses a Multilink PPP over ISDN BRI configuration, you could associate a single Dialup_Link model with each PPP interface associated with the ISDN bearer channel connections.

The following diagram shows a remote router that has two connections to a host router. Since these two connections make up one logical link, you can use one Dialup_Link model to represent the link. Depending on the type of connectivity you are using, you can resolve the Dialup_Link model at the interface or the subinterface level.

Figure 76: Remote router



To model DDR map-based connections with multilink secondary interfaces on a remote router

1. Right-click the primary interface on the remote router and select 'Non-Persistent Connection Setup, Identify NPC Backup For This Interface.'
2. Right-click one of the the backup interfaces that should activate when the primary interface is unavailable and select 'Non-Persistent Connection Setup, NPC Backup For *<primary interface>*.'
   The Create Model of Type Dialup_Link dialog opens.
3. Create a Dialup_Link model.
4. Repeat Step 2 for the remaining secondary interfaces.
5. Right-click the Dialup_Link model and select 'Non-Persistent Connection Setup, Start Non-Persistent Connection.'
6. Right-click one of the receiving interfaces on the host router and select 'Non-Persistent Connection Setup, Connect with *<Dialup-Link model>*.'
7. Repeat Step 6 for each secondary (receiving) interface on the host router.

Secondary Groups

When a Dialup_Link model is associated with more than one interface on the same device, these interfaces will form a secondary group. DX NetOps Spectrum determines the status of the connection by looking at the state of the group. The DevSecGrpActiveCriteria attribute lets you change how DX NetOps Spectrum determines the state of the secondary group. This attribute has a default value of AnySecondaryIFActive (0), meaning that the group will be active if any member of the group is up and dormant only if all members of the group are down. The other possible value for this attribute is AllSecondaryIfsActive (1). If this is the value, the secondary group is considered active only if all members of the group are up, and is considered inactive if one member goes down. You can change the value of this attribute in the Dialup_Link model's Attributes tab.

NOTE
For more information about modifying attributes, see Modeling and Managing Your IT Infrastructure section.

Dialer Profile DDR Configurations

This section discusses modeling a non-persistent connection that uses a DDR configuration with dialer profiles.

In DDR Configurations that are dialer map-based, each physical interface can have only one set of configuration characteristics. When using a DDR configuration with dialer profiles, physical interfaces take on different characteristics based on incoming or outgoing call requirements. This is possible because dialer profiles separate logical configurations from the physical interfaces that receive or make calls. This lets interfaces such as ISDN, asynchronous modems, or synchronous serial connections be shared by multiple dialer profile configurations. The logical and physical interfaces are associated dynamically on a per call basis.

A dialer profile is made up of three main components:

- **Dialer Interface**
  Logical interfaces that represent a call to a particular destination. The dialer interface configuration contains all configuration settings specific to the destination. A router can have one or more dialer interfaces. Each interface references a group of physical interfaces called a dialer pool.
- **Dialer Pool**
  A group of physical interfaces that are associated with a dialer profile. A physical interface can belong to multiple dialer pools.
- **Physical Interface**
  Each physical interface is able to determine the dialer pool to which it belongs. Interfaces in a dialer pool are configured for encapsulation parameters and each dialer profile supports PPP and HDLC encapsulation.

Representing a Dialer Interface with DX NetOps Spectrum

DX NetOps Spectrum uses a logical interface model to represent a dialer interface; the logical interface model represents a call to or from a particular destination. This model is automatically created when DX NetOps Spectrum discovers a DDR dialer profile configuration on a device.

For example, if DX NetOps Spectrum discovers a dialer profile configuration on a Cisco router, the following logical interface model is used to represent a dialer interface. If there are multiple dialer interfaces configured on the router, multiple logical interfaces are created.

Modeling a Dialer Profile Connection

To model a connection that makes use of dialer profiles, create the appropriate number of Dialup_Link models and then configure the connection for each Dialup_Link model. Dialer profiles can be used on the remote router, on the host router, or both. Create one Dialup_Link model for each phone number that the remote router could dial to create the non-persistent connection.

If the remote router can dial one number that can be picked up by one of several host routers or host router interfaces, one Dial_Up link model is used to represent this connection.

To model a dialer profile DDR connection

1. Create a Dialup_Link model for each phone number the remote routers could dial.
2. Associate each Dialup_Link model with one of the secondary (dialing) interfaces on the remote router:
   a. Right-click the Dialup_Link model and select 'Non-Persistent Connection Setup, Start Non-Persistent Connection.'
   b. Right-click the associated secondary (dialing) interface on the remote router (or the logical interface representing the dialer profile) and select 'Non-Persistent Connection Setup, Connect With <*Dialup_Link model*>.'

NOTE
If you are making use of a dialer profile on the remote router, use the logical interface model representing the dialer. If you are not using a dialer profile on the remote router, use the interface model that represents the dialing interface.

3. Specify a primary interface for each Dialup_Link model. In the event that the primary interface goes down, the secondary link represented by the Dialup_Link model will activate:
   a. Select the Dialup_Link model and click the Information tab in the Component Detail panel.
   b. Expand the Primary and Secondary Interface Information subview.
   c. In the Primary Interfaces table,

      click Add 
   d. Select one or more primary interfaces in the resulting dialog and click Add.
   e. Repeat for each Dialup_Link model that is part of this non-persistent connection configuration.
4. Associate each Dialup_Link model to the secondary (receiving) interfaces on the host router or routers.
   a. Right-click the Dialup_Link model and select 'Non-Persistent Connection Setup, Start Non-Persistent Connection.'
   b. Right-click the appropriate secondary (receiving) interface on the host router and select 'Non_Persistent Connection Setup, Connect With <*Dialup_Link model*>.'
   c. Repeat for each Dialup_Link model that is part of this non-persistent connection configuration.

Monitoring a Dialer Profile Connection

The logical interface model may behave in different ways when a call is activated. This behavior is often dependent on the particular router supporting the given dialer profile.

The status indicator on the logical interface model icon may display OFF even when the connection is live. The logical interface model icon may also move from the physical layer of the device topology view and stack itself on the PPP encapsulation layer interface model. You can locate the model by drilling down from the physical interface, to the bearer channel layer, to the PPP encapsulation layer, and then to the dialer interface. All of these behaviors are normal, but may differ depending on the device modeled. Therefore, you should rely on the Dialup_Link model status and the views presented in Monitoring Non-Persistent Connections to monitor your connection.

Multiple Dial Backup Links

This section discusses modeling a non-persistent connection that lets you have variable or unspecified links between a remote router and a host router. There could be multiple remote routers linking to a single host router or a single remote router linking to multiple host routers.

A single Dialup_Link model can support configurations that include multiple secondary links. Only one Dialup_Link model is needed to monitor multiple secondary connections, because only a single active secondary connection can exist between the host router and the remote router at one time.

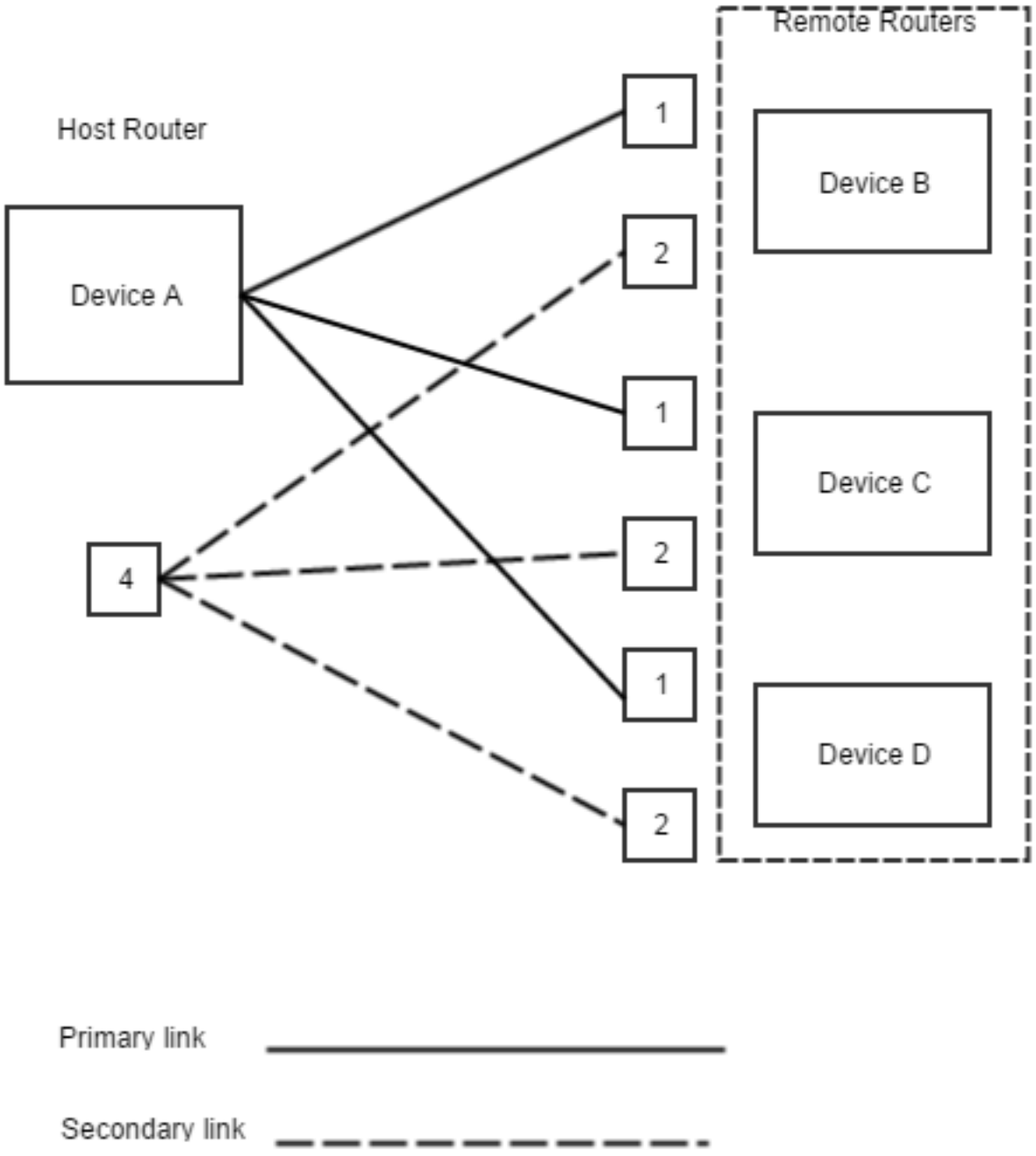
Multiple Dial Backup Links (MDBLs) are used in either of the following scenarios:

- Multiple remote routers each with a dial backup link that connects to a common host router.
- A single remote router with a dial backup link that could connect to one of several host routers.

How to Model Multiple Remote Routers to a Common Host Router

The following image shows three remote routers (devices B, C, and D) using one interface on a host router (Interface 4 of Device A) to provide redundant support for their primary links.

Figure 77: Multiple Remote Routers to a Common Host Router



There are three main tasks involved in creating this configuration:

1. Configure the secondary interfaces on the remote routers by resolving the the Dialup_Link model to the secondary interfaces on each remote router.
2. Configure the appropriate primary interfaces. When you configure a primary interface, you associate the secondary link to the primary link or links it is supporting. This is done in the Dialup_Link model's Primary Interfaces subview in the Component Detail panel.

NOTE
DX NetOps Spectrum automatically affiliates the secondary interface configured on a device with the primary interface configured on the device, so that if this primary interface fails, the proper secondary connection will be monitored.

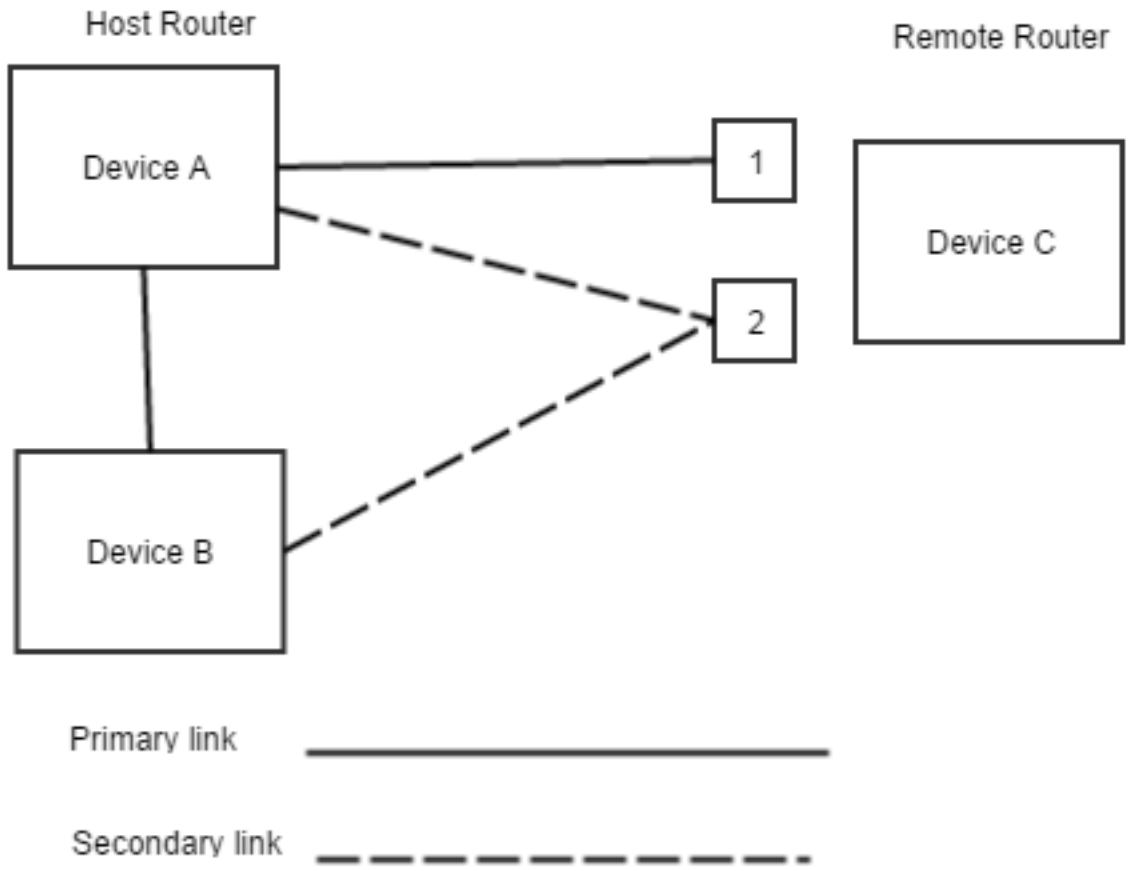
3. Configure the fixed secondary interface on the host router. This involves resolving the Dialup_Link model to the secondary interface of the host router.

How to Model a Single Remote Router to Multiple Host Routers or Host Router Interfaces

The following diagram shows a remote router (device C) which has secondary links established with two host routers (devices A and B) to provide redundant support for the primary link between device A and device C. When the secondary link becomes active, it will connect with either router A or router B. Before the connection is actually made, you cannot determine which of these routers the link will connect to.

The same scenario exists if the redundant link could connect to one of several interfaces on a host router or routers. In all of these cases, the connection to the host routers is resolved at the device level.

Figure 78: Single Remote Router to Multiple Host Routers



There are three main tasks involved in creating this configuration:

1. Configure the secondary interface on the remote router. This involves resolving the Dialup_Link model to the secondary interface.

2. Configure the appropriate primary interface. When you configure a primary interface, you associate the secondary link to the primary link it is supporting. This is done in the Dialup_Link model's Primary Interfaces subview in the Component Detail panel.
3. Configure the connection to the host router. This involves resolving the Dialup_Link model to the secondary interface on the host router. If it is unknown which host router or host router interface the remote router will connect to, resolve the dialup connection to the device level rather than the interface level.

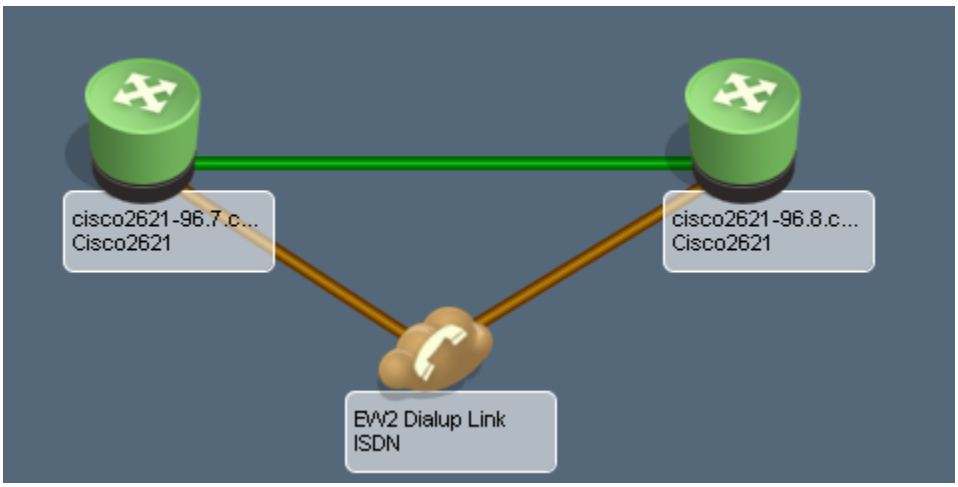
Monitoring Non-Persistent Connections

This section contains information about monitoring non-persistent connections.

Viewing Dialup_Link Model Information

Once you have configured the Dialup_Link model connections and specified the interface on the remote router for the primary connection, the Dialup_Link model is fully integrated into the current modeling scheme and reacts according to the type of link it represents. The Topology view containing the Dialup_Link model shows live pipes connecting the Dialup_Link model to the remote and host routers.

These live pipes will have a brown condition when the dialup link is inactive. If the modeled primary link and secondary links involve the same two devices, the topology configuration involving the host and remote routers should look similar to the following image:



You access information about the creation, current settings, and status of the Dialup_Link model by selecting the model and then clicking the Information tab in the Component Detail panel. The Information tab displays the following views for Dialup_Link models:

- General Information
- Thresholds and Watches
- Primary and Secondary Interface Information:
  - **Primary Interfaces**
    The Primary Interfaces list includes all currently configured primary interfaces on all devices associated with this Dialup_Link model.
  - **Secondary Interfaces**
    The Secondary Interfaces list includes all currently configured secondary interfaces on all of the devices associated with this Dialup_Link model.

The Primary and Secondary Interface Information view contains the Primary Interfaces view and the Secondary Interfaces view.

- **Primary Interfaces**
  Contains information about the primary interfaces set up on this Dialup_Link model. This is a left/right list that lets you select interfaces on the devices associated to the Dialup_Link model. Ideally, this list should contain only interfaces on the dialing/remote device.

  NOTE
  If more than one primary interface on a single remote router has been configured, these interfaces will be treated as a group. The status of the group of primary interfaces is determined by the healthiest group member. The backup interface will activate only if all of the primary interfaces in the group fail.
- **Secondary Interfaces**
  Contains information about the secondary interfaces setup on this Dialup_Link model.

The tables in both of these views display the following information by default:

- **Name**
  Specifies the name of the interface.
- **Condition**
  The contact status for the device, in addition to any alarm conditions in effect for the device model. Specifies the current condition of the interface.
- **Status**
  Indicates whether the interface is operational or non-operational. An interface may be non-operational for a variety of reasons including being administratively disabled. Some of the possible values include up, down, off, and dormant.
- **Type**
  Specifies the physical layer interface standard that the interface uses, such as ppp, ethernet, frameRelay, and so on.
- **Description**
  Describes whether the interface is physical or logical, and the interface ID, such as et.2.1.
- **Device Connected**
  The name and status (green for up or red for down) of the device that the current interface is connected to. The device name is a hyperlink that displays the Information tab for the connected device.
- **Port Connected**
  Specifies the name of the port on the device that the current port is connected to. The port name is a hyperlink that displays the Interfaces tab for the device that the current port is connected to.
- **QoS Policy**
  Specifies the QoS policy name that applies to this interface.
- **Index**
  Specifies the value of the index object in the standard RFC or proprietary MIB that uniquely identifies this interface within the device.
- **Board.Port**
  Identifies the board and port number on the device for the corresponding port. For example if the port is port 4 on a module in the device's third slot, the Board.Port value is 3.4.

Searching for Non-Persistent Connections

From the Locater tab, you can run the following two searches specific to non-persistent connections:

- **Active Non-Persistent Connections**
  The Active Non-Persistent Connections search returns a list of all Dialup_Link models that are currently active in a landscape.
- **All Non-Persistent Connections**

The All Non-Persistent Connections search returns a list of every Dialup_Link model in the landscape, even if they are currently inactive.

Each results list contains the following information:

- **Condition**
  Identifies the current condition of the Dialup_Link model.

  NOTE
  See Modeling and Managing Your IT Infrastructure section for more information about condition colors.
- **Name**
  Specifies the name of the Dialup_Link model.
- **Dialup From**
  Specifies the name of the remote router model.
- **Dialup To**
  Specifies the name of the host router model.
- **Link Activation Start Time**
  Indicates at what time the Dialup_Link model became active, or whether it is inactive. For example, when the Dialup_Link model was connected to the appropriate interfaces.
- **Link Activation Duration**
  Indicates the length of time that the connection has been active.
- **Dialup Link Type**
  Specifies the type of link between the two points. Possible types are Backup (DBL), Primary (PODL), and Bandwidth (BODL).
- **Dialup Protocol Type**
  Specifies the protocol type to used by the Dialup_Link. Possible protocol types include Analog, Switch-56, ISDN, Frame_Relay, and Other.

Determining the Status of Multiple Primary or Secondary Links

You can model non-persistent connections using multiple primary or secondary interfaces. When multiple primary interfaces are used, the status of the group is always determined using the following rule: If any of the interfaces are up, then the group is up; otherwise, the group is down.

Multiple secondary links use the DecSecGrpActiveCriteria attribute to determine the status of the group. If the value is AnySecondaryIFActive, and at least one of the secondary interfaces is up, then the group is considered to be up. Otherwise, the group is considered to be down.

Multiple primary links operate in the same way. If the value of the DecSecGrpActiveCriteria attribute is AllSecondaryIfsActive, all of the interfaces must be up for the group to be considered up. If not, then the group is considered to be down.

When one or more interfaces in the multiple link scenario is in maintenance mode, DX NetOps Spectrum will treat it as if that link does not exist, and base the status of the group on the remaining non-maintenance interfaces. The exception is if all interfaces are in maintenance mode, then the entire group is in maintenance mode.

Dialup_Link Condition Colors

The way in which the Dialup_Link model reacts to changes in the primary and secondary links depends upon the type of non-persistent connection this model represents. The following tables show the possible conditions for the Dialup_Link model representing a Dial Backup Link, Bandwidth on Demand Link, or a Primary On Demand Link.

Dial Backup Link Conditions

Primary Link	Secondary Link	Dialup_Link	Generated Events
Up	Up	Yellow	Both primary and secondary links are active.
Up	Down	Brown	Secondary link inactive.
Down	Up	Green	Secondary link active.
Down	Down	Red	Secondary link has failed to activate.
Maintenance Mode	Up	Green	Secondary link active.
Maintenance Mode	Down	Brown	Secondary link inactive.
Up	Maintenance Mode	Brown	Secondary link inactive.
Down	Maintenance Mode	Brown	Secondary link inactive.

Bandwidth On Demand Link Conditions

Primary Link	Secondary Link	Dialup_Link Condition	Generated Events
Up	Up	Green	None.
Up	Down	Brown	Secondary link is inactive.
Down	Up	Green	Secondary link is active.
Down	Down	Red	Secondary link has failed to activate.
Maintenance Mode	Up	Green	Secondary link active.
Maintenance Mode	Down	Brown	Secondary link inactive.
Up	Maintenance Mode	Brown	Secondary link inactive.
Down	Maintenance Mode	Brown	Secondary link inactive.
Maintenance Mode	Maintenance Mode	Brown	Secondary link inactive.

Primary On Demand Link Conditions

Primary Link	Dialup_Link Condition	Generated Events
Up	Green	No event is generated. Primary On Demand link is active.
Down	Brown	No event is generated. Primary On Demand link is inactive.
Up	Yellow	The maximum uptime value has been exceeded.
Maintenance Mode	Brown	No event is generated. Primary On Demand link is inactive.

There are two exceptions to these Dialup_Link Condition tables. These involve these special alarm conditions: Gray and Orange.

- A Gray condition is asserted on a Dialup_Link model when all secondary interfaces are considered unreachable.
- An Orange condition is asserted only in a multiple primary/secondary configuration when a primary link has failed but the secondary link is already active in support of another connection.

Alarms for a Dialup_Link model

The Thresholds and Watches view in a Dialup_Link model's Information tab lets you configure when alarms will be generated on the Dialup_Link model for certain conditions. To access the Thresholds and Watches view, click the Dialup_Link model, click the Information tab, and then expand the Thresholds and Watches subview.

- **Activation Grace Period (Minutes)**
  Lets you specify the time allowed, in minutes, for the secondary link to become active after a primary link failure. If this grace period expires before the secondary link is active, then a red alarm is generated. This field is only used by DBL-type link models. The value for this field can also be specified when creating a Dialup_Link model.
- **Deactivation Grace Period (Minutes)**
  Specifies the time allowed, in minutes, for an active secondary link to deactivate after the failed primary link reactivates. If the secondary link is still active after this grace period expires, then a yellow alarm is generated. This field is only used by DBL-type link models. The value for this field can also be specified when creating a Dialup_Link model.
- **Active Time Until Yellow (Hours)**
  Specifies the number of hours a backup link can be active before a yellow alarm is generated. The value for this field can also be specified when creating a Dialup_Link model.
- **Active Time Until Orange (Hours)**
  Specifies the number of hours a backup link can be active before an orange alarm is generated. The value for this field can also be specified when creating a Dialup_Link model.
- **Active Time Until Red (Hours)**
  Specifies the number of hours a backup link can be active before a red alarm is generated. The value for this field can also be specified when creating a Dialup_Link model.
- **Active Criteria**
  Specifies the active state for non-persistent connections that have more than one secondary interface, i.e. a connection that uses multiple physical connections to form one logical connection. When this multi-link type configuration is used, the Dialup_Link model will monitor all the secondary interfaces. If Active Criteria is set to Any Secondary IF Active then the dial-up link will be considered "active" if ANY of the secondary interfaces become active. If Active Criteria is set to All Secondary IFs Active then ALL the secondary interfaces must become active before the Dialup_Link model will be considered "active."

The following shows the alarms that DX NetOps Spectrum can generate on a Dialup_Link model.

> **NOTE**
> Events generated on a Dialup_Link model have an event message which contains information on the devices and interfaces that the Dialup_Link model is connected to.

Alarm Code	Alarm	Alarm Cause
0x022ffff8	Excess timer has been exceeded.	The link has been active for longer than the time specified by the Active Until Yellow, Orange or Red criteria.
0x022ffff9	Backup link is already active.	This alarm occurs when DX NetOps Spectrum attempts to activate a dialup link that is already active.
0x022ffffa	Both the primary and the secondary links are active.	Both the primary and the secondary links have been activated.
0x022ffffb	The secondary link failed to activate.	An attempt to activate the secondary link was unsuccessful.
0x022ffffc	The secondary link is inactive.	The primary link is functioning properly and the secondary link has returned to an inactive status.

0x022ffffd	The Dialup_Link model is unreachable.	This alarm is generated when all of the Dialup_Link model's neighbors are down. Neighbors include any interface model or device model to which the Dialup_Link model has a resolved connection.

Outsourcer Billing

DX NetOps Spectrum Outsourcer Billing is a powerful tool that enables service providers to produce accurate and meaningful data which can be further used as a basis for billing their customers for outsourced network services.

DX NetOps Spectrum Outsourcer Billing users should be familiar with the following:

- DX NetOps Spectrum
- Network management
- UNIX and/or Microsoft Windows
- The networks and devices you intend to bill for with DX NetOps Spectrum

Outsourcer Billing Users

Many end-customers would rather pay for managed network services as they use them, rather than buy, install, and maintain DX NetOps Spectrum themselves. DX NetOps Spectrum Outsourcer Billing helps the DX NetOps Spectrum equipped service provider bill these customers based upon the number of devices being managed or monitored by DX NetOps Spectrum in an outsourcing relationship.

Service providers who should consider using DX NetOps Spectrum Outsourcer Billing include:

- ASPs (Application Service Providers)
- CLECs (Competitive Local Exchange Carriers)
- DSPs (Data Service Providers)
- ISPs (Internet Service Providers)
- IXCs (IntereXchange Carriers)
- MSOs (Multiple Service Operators [e.g., cable, DSL, wireless, etc.])
- MSPs (Management Service Providers)
- RBOCs (Regional Bell Operating Companies)

Preliminary Information

DX NetOps Spectrum Outsourcer Billing is installed as part of the SpectroSERVER base installation.

Before you use DX NetOps Spectrum Outsourcer Billing for the first time you should gather some pertinent information about the network upon which you are utilizing Outsourcer Billing:

- The number of devices on the network.
- The types of devices on the network.
- Whether or not your customer has a distributed SpectroSERVER (DSS) environment (for example, more than one SpectroSERVER).
- Pricing schedules and volume discounts.

Such information will assist you in determining the best way to use DX NetOps Spectrum Outsourcer Billing to extract the most accurate information and eliminate billing inconsistencies.

Usage Fees

You will also need to design a usage fee schedule that meets your needs and those of your customers. You may or may not wish to incorporate volume discount pricing for your larger clients as well. This should be researched carefully to ensure an acceptable mix of value for your customers and profit for you.

Starting Outsourcer Billing

To invoke DX NetOps Spectrum Outsourcer Billing:

1. In a command prompt window, navigate to the <$SPECROOT>/SS-Tools directory.
2. Start the Billing executable using the syntax described below:

```
./Billing -landscape landscape [-detailed True|False] [-output file] [-overrideClassFile file] [-
mailQueueSize number]
[-mailTimeout seconds] [-throttle count] [-debug] [-help]
```

Parameters are described below.

- **landscape**
  The landscape handle of the default SpectroSERVER.
  Default value: 0x0
- **detailed**
  Toggles the option for a detailed report on (TRUE) or off (FALSE).
  Default value: False
- **output**
  The path to the desired output file. If no output file is specified, the output displays on the screen. If a file of the same name exists, it is overwritten.
- **overrideClassFile**
  A user-defined text file used to force a different class upon a model type.
- **mailQueueSize**
  The maximum number of pending SpectroSERVER requests.
  Default value: 1024
- **mailTimeout**
  The minimum time, in seconds, the MailService waits for a SpectroSERVER response before the request is canceled.
  Default value: 1800 (30 minutes)
- **throttle**
  The maximum number of models from which DX NetOps Spectrum Outsourcer Billing can read information simultaneously.
  Default value: 500
- **debug**
  Invokes DX NetOps Spectrum Outsourcer Billing in debug mode.
- **help**
  Displays DX NetOps Spectrum Outsourcer Billing syntax information.

WARNING
The amount of time that DX NetOps Spectrum Outsourcer Billing takes to complete a billing audit is dependent upon the numbers of devices in the network and models in the SpectroSERVER database. Audits run on large networks will take longer than those run on smaller networks.

Output

Invoking the application with no options (by typing only ./Billing at the command line) causes DX NetOps Spectrum Outsourcer Billing to read and display a report on the device models in all landscapes in the local SpectroSERVER's landscape database.

If you type an incorrect landscape handle using the -landscape option, DX NetOps Spectrum Outsourcer Billing prompts you for a valid landscape handle from a list of known SpectroSERVERs (VNMs) in the local SpectroSERVER's landscape database. Enter the number that corresponds to the landscape against which you wish to run DX NetOps Spectrum Outsourcer Billing. Pressing the Enter (or Return) key selects the default landscape.

For example, if you enter a landscape handle value of '0' the screen output would look similar to this:

```
% ./Billing -landscape 0
Invalid landscape 0x0
Please select from the following list of valid
landscapes. Enter the number (1, 2, 3, etc.)
corresponding to the VNM name and landscape handle,
and press the return key.
   1) storyville (0x8400000)
   2) cottonclub (0x8e00000)
   3) birdland (0xa600000)
Selection? [default: 1]
```

You then make your selection, which in this case was '3' for the SpectroSERVER 'birdland', you would see the following:

```
3
Please wait - working...
```

Progress Messages

DX NetOps Spectrum Outsourcer Billing provides report progress feedback in the form of informational status messages as it performs the following operations:

- Retrieves the landscape map
- Retrieves device models from the landscape
- Reads device models
- Retrieves device class information for each device
- Retrieves port counts for each device

General Output Summary

If the following syntax is issued from running DX NetOps Spectrum Outsourcer Billing in a distributed SpectroSERVER environment:

```
$ ./Billing -landscape <landscape_handle>
```

The following output will be generated:

```
Retrieving the landscape map
Retrieving device models from server birdland
Retrieving Class information for each device. Total device
count=78
Read 78 models
Retrieving port count and system information for each device
```

```
Read 78 models

Audit Summary
-------------
SpectroSERVER=birdland Port=48879 Landscape=0xa600000

Device Class      Type      Quantity
-----------------------------------------
Router             1            0
Router             2            0
Router             3            0
Switch             1            24
Switch             2            12
Switch             3            0
Hub                1            2
Hub                2            0
Hub                3            0
Class 1           n/a           117
Pingable          n/a           38
```

Detailed Output

Detailed output provides you with more granular information. It includes the summary output as described in General Output Summary as well as information about every device modeled in the SpectroSERVER's database. This information, which provides the basis for the accurate billing you require, includes:

- Device name
- Device vendor
- Device class
- Number of ports
- Device type value based on the device class and the number of ports:

Class	Type 1	Type 2	Type 3
Hub	<= 24 ports	25-120 ports	> 120 ports
Switch	<= 24 ports	25-120 ports	> 120 ports
Routers	<= 4 ports	5-12 ports	> 12 ports
Class 1	no types		
Pingable	no types		

If the following syntax is issued from running DX NetOps Spectrum Outsourcer Billing in a distributed SpectroSERVER environment:

```
$ ./Billing -landscape <landscape_handle> -detailed True
```

The following output will be generated:

```
Retrieving the landscape map
Retrieving device models from server birdland
Retrieving Class information for each device.
Total device count=78
Read 78 models
Retrieving port count and system information for each device
```

```
Read 78 models

Audit Summary
-------------
SpectroSERVER=birdland Port=48879 Landscape=0xa600000

Device Class      Type      Quantity
-------------------------------------------
Router             1            0
Router             2            0
Router             3            0
Switch             1            12
Switch             2            6
Switch             3            0
Hub                1            1
Hub                2            0
Hub                3            0
Class 1           n/a           59
Pingable          n/a           19

Detailed Report
---------------
Name,Vendor,Class,Type,Ports
corporate36,ABC MANAGEMENT TECHNOLOGIES,Switch,2,26
corporate20,ABC MANAGEMENT TECHNOLOGIES,Switch,2,26
corporate50,ABC MANAGEMENT TECHNOLOGIES,Switch,2,30
corporate39,ABC MANAGEMENT TECHNOLOGIES,Switch,2,48
corporate30,ABC,Switch,1,4
corporate14,ABC,Switch,1,4
corporate28,ABC MANAGEMENT TECHNOLOGIES,Switch,1,0
poseidon,,Switch,1,4
corporate27,ABC,Switch,1,4
corporate17,ABC,Switch,1,4
corporate18,ABC MANAGEMENT TECHNOLOGIES,Switch,2,25
corporate31,ABC MANAGEMENT TECHNOLOGIES,Switch,1,3
corporate59,ABC MANAGEMENT TECHNOLOGIES,Switch,1,3
corporate38,ABC MANAGEMENT TECHNOLOGIES,Switch,1,3
corporate56,ABC MANAGEMENT TECHNOLOGIES,Switch,1,4
sales134,ABC MANAGEMENT TECHNOLOGIES,Switch,2,29
corporate11,ABC MANAGEMENT TECHNOLOGIES,Switch,1,2
corporate7,ABC MANAGEMENT TECHNOLOGIES,Switch,1,4
corporate12,ABC MANAGEMENT TECHNOLOGIES,Hub,1,1
sales27,ABC MANAGEMENT TECHNOLOGIES,None,0,8
corporate15,,None,0,4
corporate26,ABC MANAGEMENT TECHNOLOGIES,None,0,25
corporate53,ABC MANAGEMENT TECHNOLOGIES,None,0,13
corporate55,ABC MANAGEMENT TECHNOLOGIES,None,0,27
corporate29,ABC,None,0,3
pc1,,None,0,0
sales8,,None,0,0
sales7,,None,0,0
sales9,,None,0,0
sales6,,None,0,0
```

```
pc2,,None,0,0
pc3,,None,0,0
pc4,,None,0,0
pc5,COMPAQ COMPUTER CORPORATION,None,0,2
pc6,ABC MANAGEMENT TECHNOLOGIES,None,0,2
ws1,3COM CORPORATION,None,0,2
pc7,ABC MANAGEMENT TECHNOLOGIES,None,0,2
ws2,COMPUTER PRODUCTS INTERNATIONAL,None,0,2
pc8,3COM CORPORATION,None,0,2
ws3,COMPUTER PRODUCTS INTERNATIONAL,None,0,2
ws4,COMPUTER PRODUCTS INTERNATIONAL,None,0,2
ws5,COMPUTER PRODUCTS INTERNATIONAL,None,0,2
ws6,COMPUTER PRODUCTS INTERNATIONAL,None,0,2
ws7,COMPUTER PRODUCTS INTERNATIONAL,None,0,2
pc9,INTEL CORPORATION - HF1-06,None,0,2
pc10,3COM CORPORATION,None,0,2
pc11,INTEL CORPORATION - HF1-06,None,0,2
pc12,INTEL CORPORATION - HF1-06,None,0,2
pc13,ABC MANAGEMENT TECHNOLOGIES,None,0,2
corporate10,ABC MANAGEMENT TECHNOLOGIES,None,0,0
192.168.32.99,CISCO SYSTEMS INC.,None,0,32
```

Advanced Operations

DX NetOps Spectrum Outsourcer Billing enables you to customize its operation to your needs using several advanced options.

Overriding Class Files for Custom Billing

There may be times when you wish to override the default device classifications. For example, you may have a need to define a brouter as either a switch or a router. Or, you may at some point need to define a completely new device classification. This is where the overrideClassFile parameter is used.

To override the default device class files

1. Create a text file containing the model type name and new class number in the following format:
   ```
   model_type_name new_class_number
   ```
   The file might look something like this:
   ```
   GnSNMPDev 5
   ```
2. Give the new file a name.
3. Invoke DX NetOps Spectrum Outsourcer Billing using the -overrideClassFile parameter with the newly created file's name as the value. For example:
   ```
   $ ./Billing -landscape <landscape_handle> -overrideClassFile newclass
   ```
   The output file is displayed on the screen and includes the new device class by name (e.g., Hub, Switch, etc.).

Adjusting the Mail Queue Size

By default DX NetOps Spectrum Outsourcer Billing can have 1024 pending SpectroSERVER requests queued at any one time. However, if your mail queue becomes full, you can configure the queue size so that more requests can be passed to DX NetOps Spectrum Outsourcer Billing using the mailQueueSize parameter.

To adjust the mail queue size when using DX NetOps Spectrum Outsourcer Billing

- Invoke DX NetOps Spectrum Outsourcer Billing using the -mailQueueSize parameter. For example:

```
$ ./Billing -landscape <landscape_handle> -mailQueueSize 2048
```

Now 2048 pending SpectroSERVER requests can be passed to DX NetOps Spectrum Outsourcer Billing.

NOTE
The value for the mailQueueSize parameter must be greater than or equal to 1. If an invalid value is entered (for example, 0 or a negative value), DX NetOps Spectrum Outsourcer Billing will terminate.

Adjusting the Mail Timeout Value

There may be instances when you wish for DX NetOps Spectrum Outsourcer Billing to wait longer than the default 1800 seconds for a response from the SpectroSERVER. For example, if in a DSS environment, a remote SpectroSERVER upon which you wish to run DX NetOps Spectrum Outsourcer Billing terminates abnormally, you may wish to increase the timeout value to compensate for the time required for that machine to come back up.

To adjust the mail timeout value when using DX NetOps Spectrum Outsourcer Billing

- Invoke DX NetOps Spectrum Outsourcer Billing using the -mailTimeout parameter. For example:

  ```
  $ ./Billing -landscape <landscape_handle> -mailTimeout 3600
  ```

  Now DX NetOps Spectrum Outsourcer Billing will wait for 3600 seconds before responding with an error message indicating a lack of response from the remote SpectroSERVER.

Adjusting the Throttle Count

There may be instances when DX NetOps Spectrum Outsourcer Billing is reading device information from some device models that have timed out, thus causing the application to appear sluggish. This can happen because by default DX NetOps Spectrum Outsourcer Billing reads information from 500 models at once. You can reduce that default number upon invoking DX NetOps Spectrum Outsourcer Billing.

To adjust the throttle count when using DX NetOps Spectrum Outsourcer Billing

- Invoke DX NetOps Spectrum Outsourcer Billing using the -throttle parameter. For example:

  ```
  $ ./Billing -landscape <landscape_handle> throttle 200
  ```

  Now DX NetOps Spectrum Outsourcer Billing will read information from just 200 models simultaneously.

  NOTE
  The value for the throttle parameter must be greater than or equal to 1. If an invalid value is entered (for example, 0 or a negative value), DX NetOps Spectrum Outsourcer Billing will terminate.

Gathering Debug Information

In the event that you require technical assistance with DX NetOps Spectrum Outsourcer Billing, you should run the application with the -debug option and capture the output to a file. This will provide the support personnel with needed low-level information for troubleshooting and analysis. For example:

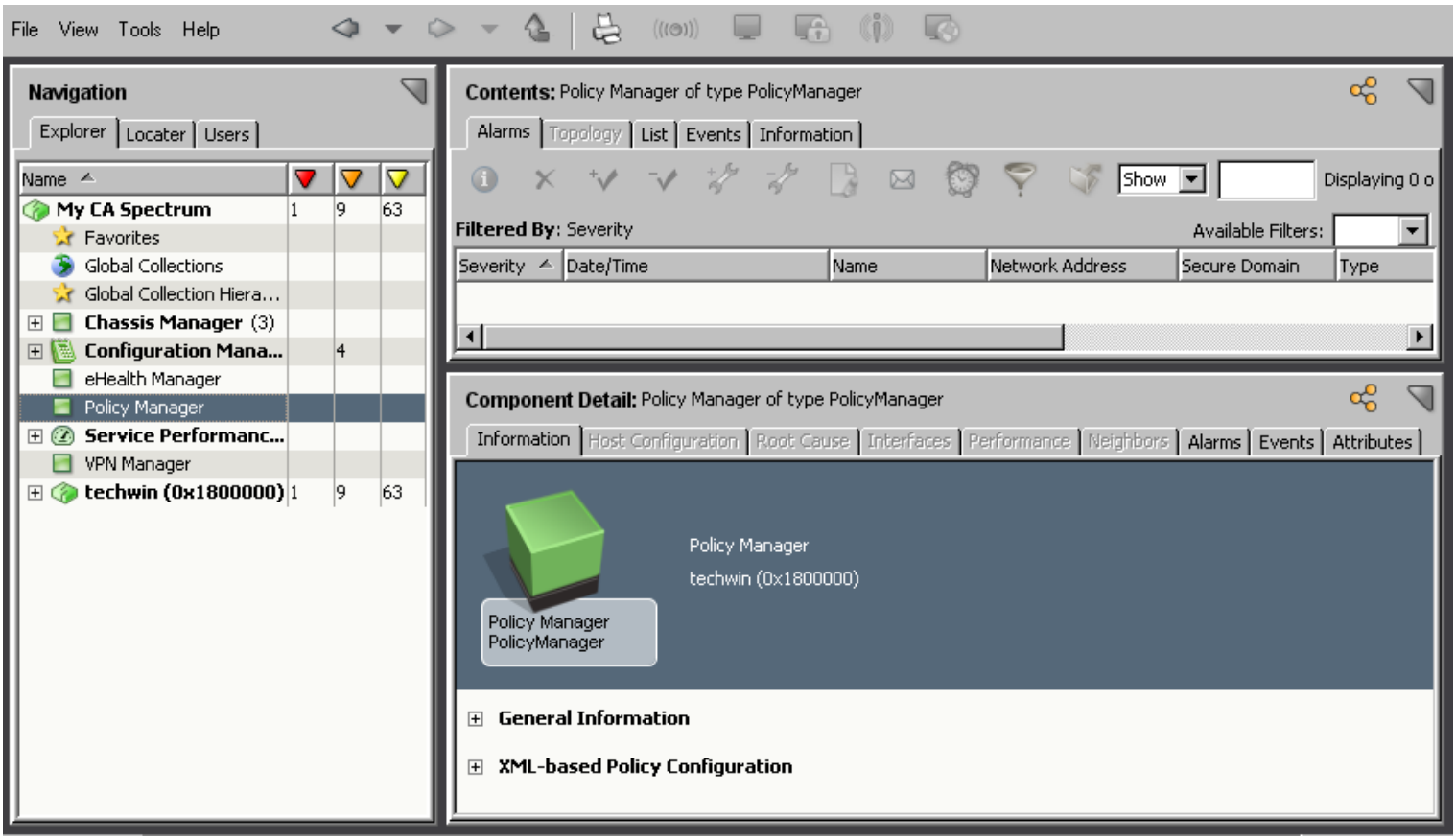
```
$ ./Billing -landscape <landscape_handle> -debug > debugfile
```

Policy Manager

DX NetOps Spectrum Policy Manager lets you apply network management policies across all models in a distributed SpectroSERVER environment. You can add, remove, or modify policy configurations while the SpectroSERVER is running and apply the changes immediately. Policy Manager automates the enforcement of management policies in DX NetOps Spectrum, eliminating the need to make manual updates as models are added or changed.

Access Policy Manager

To access Policy Manager, click Policy Manager in the Explorer tab. Policy Manager information is displayed in the Information tab in the Component Detail panel.



NOTE
All defined policies appear beneath Policy Manager in the Explorer tab and in the List tab in the Contents panel. For more information, see Viewing Policies.

The Component Detail panel contains the following subviews:

- **General Information** - The General Information subview contains general details about Policy Manager, including model class and security string.
- **XML-based Policy Configuration** - This subview provides details about maintaining legacy policies. For more information, see Legacy XML-based Policies.

Policy Manager Policies

Policies consist of DX NetOps Spectrum attribute settings that, when applied to a defined set of models, let DX NetOps Spectrum consistently implement a network management configuration. For example, you implemented a policy that defines how DX NetOps Spectrum manages alarm thresholds on all router port models. This policy is enforced on all existing router port models and is also implemented on any newly created router port models.

A policy comprises the following components:

- Policy definition
- Policy rules
- Rule settings

All policy components are configured and maintained in the OneClick Console.

> **NOTE**
> For legacy policies, you can also use XML files to configure and maintain policy definitions. For more information, see Legacy XML-based Policies.

Policy Definition

A policy is a set of prioritized policy rules. The priority handles situations in which a model resides in more than one of the global collections for the policy rules in one policy.

Policy Rules

A policy rule is a collection of rule settings and the global collections to which they apply. You can define multiple rules for one policy.

Rule Settings

Rule settings define the model attributes and attribute values that a policy maintains. Policy Manager provides predefined policy settings, including Passive Port Monitoring, Poll Unconnected Ports, Maintenance Mode, 10 Minute Polling, and Disable Redundancy. If the predefined settings do not meet your needs, you can define your own settings, using any of the available attributes. You can also include SpectroWatch settings that can activate or deactivate defined watches when a policy rule is triggered.

Creating Policies

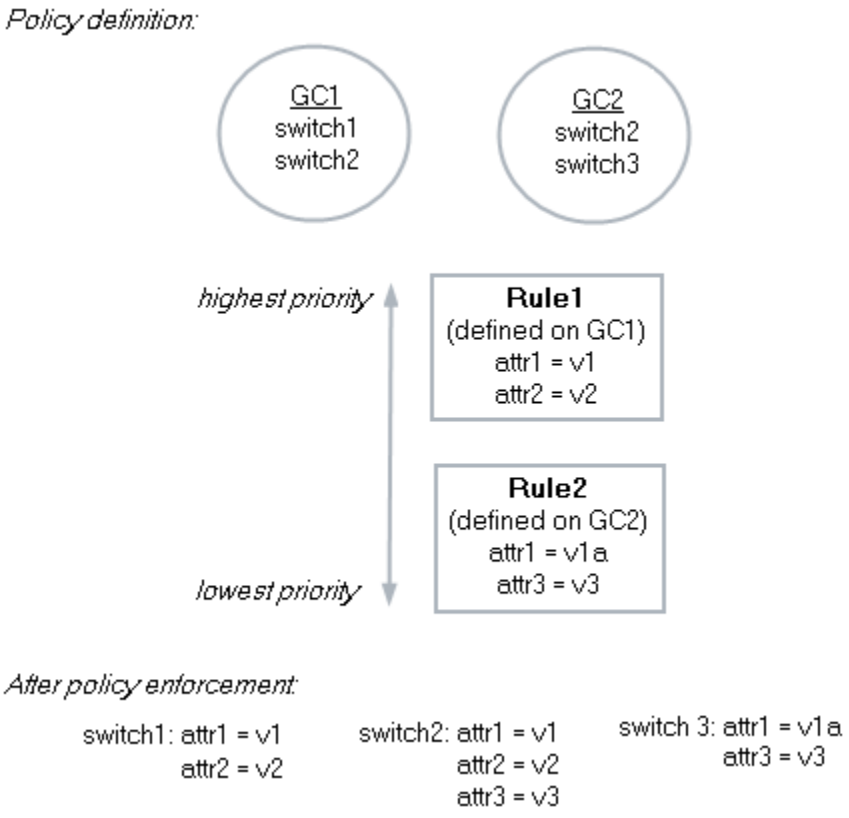
A policy consists of a policy definition, one or more policy rules, and rule settings. Policies must be enabled before they can be enforced.

> **NOTE**
> For legacy policies, use XML files to configure and maintain policy definitions. For more information, see Legacy XML-based Policies.

About Policies

A policy defines a group of related attributes for a set of policy rules. One or more rules are combined to create a policy. The settings that are applied to a model during policy enforcement can come from more than one rule. Rules are prioritized to handle situations in which a model resides in more than one of the global collections for the rules in a policy. Each rule is evaluated in prioritized order until all rules are exhausted. A model adopts the settings of the first rule whose criteria is met. If a model matches the criteria in a subsequent rule, only settings that have not yet been encountered are applied.

For example, the following diagram describes a policy that is defined on two different global collections, GC1 and GC2. Switch1 belongs to GC1 exclusively and only the settings in Rule1 are applied. Switch3 belongs to GC2 exclusively and only the settings in Rule2 are applied. Switch2 belongs to both GC1 and GC2. Because of rule priority, settings in Rule1, which includes attr1 and attr2, are applied first. Then, any settings in Rule2 that have not yet been applied to this model are applied, which is attr3 only. attr1 is ignored because it has already been applied to this model.



A policy must be enabled to be in effect. Events and alarms for the policy and affected device models track the activity and enforcement of a policy.

How to Successfully Plan Policy Definitions

When designing policies to be implemented at your site, consider the following guidelines:

- Use recommended policies. For more information, see Recommended Policy Settings.
- Use predefined settings. Typically, using templates is the best approach when setting up policies. Start with the collection of attributes in a template and then adjust the attributes and their values as needed.
- Define the policy to address a certain condition. If you identified a problem in your network, consider the attributes to monitor based on the nature of the problem.
- Develop a new policy in a test environment and then move into a production environment. For more information, see Exporting and Importing Policies.

Example: Port Fault Management Policy

Suppose you want to set up a policy for port status monitoring: you want to passively monitor all switch ports and actively monitor all router ports. From review of the section Recommended Policy Settings, you determine that the Port Fault Management Policy reflects this policy scenario. To implement this policy, you define a series of policy rules:

- Rule 1: Define a global collection specifying search criteria which identifies all switch ports. You can then use the predefined settings for passive monitoring in the Passive Port Monitoring template on all devices in this global collection. Adjust attribute values as necessary.
- Rule 2: Define a global collection specifying search criteria which identifies all router ports. Then, on all devices in the collection, use the predefined settings for port status monitoring using Live Pipes, as defined in the Live Pipes template. Adjust attribute values as necessary.

These two policy rules are combined to create the Port Fault Management Policy. The policy must then be enabled to take effect.

> **NOTE**
> The Port Fault Management Policy and other recommended policies are described in Recommended Policy Settings.

Restrictions in Policy Definitions

Consider the following restrictions when planning how to define your policies:

You cannot include the same attribute in more than one policy, regardless of whether the policy is enabled.

- Rules that apply to the same global collection cannot use the same setting target. One rule can apply to multiple global collections, but two *different* rules using the *same* setting target cannot apply to the *same* global collection.

These restrictions are in place to help prevent conflicts in your policy definitions.

Internal Attributes

Although Policy Manager is designed to modify and enforce DX NetOps Spectrum internal attributes, do not use Policy Manager to modify certain attributes.

Some attributes are used to control and customize DX NetOps Spectrum behavior and are documented for customizing. These attributes can be included in Policy Manager policies with expected results.

The values of other attributes change automatically in the DX NetOps Spectrum model (such as the Link_Condition attribute) or are intended as status only. These attributes change values as they are polled from the modeled device or in response to other attribute changes involved in computing the value. Overwriting these automatic attribute values can lead to unpredictable behavior. Do not use Policy Manager to modify these attributes.

External Attributes

Policy Manager is designed to enforce Spectrum internal attributes. You can specify external attributes (such as sysContact, sysLocation, or Firmware_version) in policies. However, the results differ from internal attributes, as follows:

- If the device is modeled with a read/write community string, the attribute value in the policy is written to the device.
- If the device is modeled with a read-only community string, the write fails.
- The SpectroSERVER has no write-lock. The external attributes can be modified through OneClick or the SpectroSERVER.
- You can modify the attribute on the device by other means, such as telnet/ssh to the device. The attribute value in the policy is enforced again the next time the policy is re-enabled.

Create a Policy in Policy Manager

You can create a policy using the OneClick Console. Define at least one rule when creating a policy. You can add more rules and settings later, as needed.

Follow these steps:

1. Click Policy Manager in the Explorer tab.
   Information about Policy Manager is displayed in the Contents panel and the Component Detail panel.
2. Click the List tab in the Contents panel.
   A list of existing policies is displayed.

3. Click  .
   The Configure Policy dialog opens.
4. Type a name for this policy in the Policy Name field.
5. Create a rule for the policy:
   a. Click  .
      The Configure Rule dialog opens.
   b. Type a name for this rule in the Rule Name field.
   c. Click Browse.
      The Select Global Collections dialog appears.
   d. Select the global collections for this policy, move them to the 'Applies to' list on the left, and then click OK.

      NOTE
      You can create global collections directly from the Select Global Collections dialog using the Create button. For information about creating and maintaining global collections, see Modeling and Managing Your IT Infrastructure .
   e. Define rule settings. Rule settings define the model attributes and attribute values that a policy uses. Use one or more of the following methods:
      - 
        - Create a new, custom setting.
      - 
        - Activate/deactivate a SpectroWatch.
      - 
        - Select predefined settings from a template.
   f. Click OK
      The rule is added to the list of Associated Rules and, when selected, the settings for the rule appear in the Rule Settings window.
6. Repeat step 5 to add more rules to the policy, as needed.

   NOTE

   ■ You can copy an existing rule by clicking the Copy an Existing Rule button.
   ■ Rules that apply to the same global collection cannot use the same setting target. Change the global collection designation when copying rules.
7. After all rules are defined, use the up and down arrows on the toolbar to adjust the priority of the rules in the list: the higher the rule is in the list, the higher the priority of the rule.
   The rules are adjusted in the list and the priority values are modified accordingly.
8. (Optional) Select 'Enable Policy on Creation' to enable and enforce the policy when it is created.

   NOTE
   You can also enable the policy later. For more information, see Enabling and Disabling Policies.
9. Click OK.
   The policy is created and the Configure Policy dialog closes. The new policy appears under Policy Manager in the Explorer tab and in the List tab of the Contents panel. If you enabled the policy on creation, the policy is enforced.

Creating a Custom Rule Setting

By creating custom rule settings, you can specify your own selection of model attributes and values in a policy rule.

> **NOTE**
> Policy Manager also provides templates that include predefined settings. For more information, see Add a Predefined Rule Setting.

Follow these steps:

1. On the Configure Rule dialog, click
   
   The Configure Attribute Setting dialog opens.
2. Type a name for this rule setting in the Setting Name field.
3. Specify an attribute using *one* of the following tasks:
   – Select an attribute from the Attribute drop-down list.
   – Click the Attribute button, select an attribute from the Attribute Selector dialog, and click OK.
   The selected attribute appears in the Attribute field.
4. Enter a value in the Attribute Value field using one of the following methods. The available methods vary depending on the attribute:
   – Accept the default value.
   – Select an attribute from a drop-down list.
   – Use the Browse button.
   – Enter a value manually.
5. Click OK.
   The selected attribute and its value are added to the Rule Settings list.
6. Repeat the steps 1 through 5 to add more custom rule settings.

Add a SpectroWatch Setting

By including SpectroWatch settings in your policy, you can activate or deactivate defined watches on a per-model basis when a policy rule triggers.

Watches can be set on any attribute of a model type, including both internal and external attributes. For example, a log watch can be set on ‘contact status’ or ‘total packets.’ Also, you can set multiple watches on a single attribute. For example, two threshold watches could be set on a device’s packet rate:

- One to generate a yellow alarm when the value exceeds 10,000
- Another to generate a red alarm when the value exceeds 15,000

> **NOTE**
>
> For more information about working with SpectroWatches, see Watches .

Follow these steps:

1. On the Configure Rule dialog, click
   
   The Configure SpectroWatch Setting dialog opens.
2. Type a name for this rule setting in the Setting Name field.
3. Select a SpectroWatch Value from the drop-down list.
   – **Active**

Activate the SpectroWatch when the rule is triggered.

- **Inactive**
  Deactivate the SpectroWatch when the rule is triggered.

4. Specify a watch using *one* of the following tasks:
   - Click Model Type, select a value from the Select Model Type dialog, and click OK.
   - Select a value from the Model Type drop-down list. The model type value of 'SpectroWatch' includes watches internal to DX NetOps Spectrum.

   The list is populated with available watches.

   NOTE
   The lists can take a moment to be populated.
5. Select the SpectroWatches you want to add to this policy rule.
6. Click OK.
   The selected SpectroWatches are added to the Rule Settings list.
7. Repeat steps 1 through 6 to add more SpectroWatch settings.

Add a Predefined Rule Setting

Policy Manager provides templates that contain predefined policy settings that you can add to your policy rule. Each template contains a number of related attributes and attribute values for a particular purpose. For example, the AlarmThresholdingSettingsTemplate includes alarm-related attributes that can be used to manage alarm thresholding on certain device or port models.

NOTE

Policy Manager also allows you to specify custom settings in your rules. For more information, see Creating a Custom Rule Setting.

Follow these steps:

1. On the Configure Rule dialog, click

   The Select Template dialog opens and lists all available templates. Predefined templates have a value of 'CA' in the Type field.

   NOTE
   Templates with a value of PolicyRule in the Type field are user-defined rules that have already been created for this policy. You can use these rules as templates when assigning the same settings for different global collections in the same policy.
2. Select the template that contains the attributes you want to use in the policy rule.

   NOTE
   To see the complete template description, roll over the description field for the template.

   The attributes that make up the selected template are listed in the Rule Settings section.
3. Click OK.
   The Select Template dialog closes and the attributes that make up the template you selected appear in the Rule Settings list. Any default attribute values are shown in the Value column.
4. Modify or define attribute values. Each setting requires an attribute value before the rule can be saved.
   a. Select an attribute in the Rule Settings list and click

   The Configure Attribute Setting dialog opens.

b. Enter a value in the Attribute Value field. Depending on the attribute, you can select from a drop-down list, use a Browse button, or enter a value manually.
c. Click OK.
The Configure Attribute Setting dialog closes and the attribute value appears in the Value column for that setting.
d. Repeat step 4 to modify or define attribute values for all rule settings as necessary.

5. Repeat steps 1 through 4 to add more predefined settings.

Enable and Disable Policies

For a policy to enforce its defined network management configurations, it must be enabled. Also, to edit or delete a policy or to export policy definitions, the policy must be disabled.

NOTE
Legacy XML-based policies can only be enabled or disabled by modifying and reloading the policy definition XML. For more information, see Legacy XML-based Policies.

Follow these steps:

1. Expand the Policy Manager node in the Explorer tab.

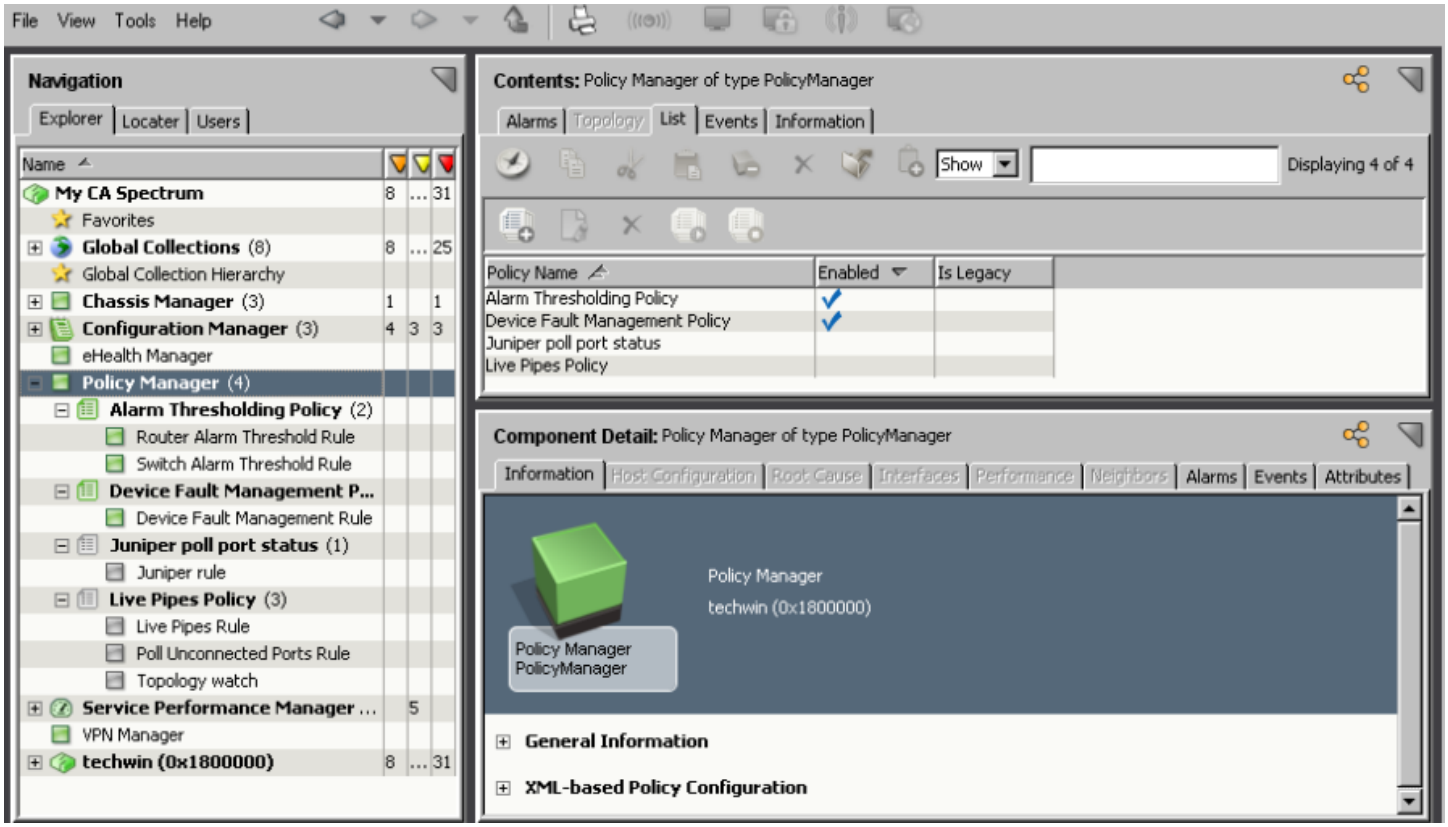
NOTE
Policies appear in the Explorer tab beneath the Policy Manager node. Enabled policies have a green icon and disabled policies have a gray icon.

2. Select the List tab in the Contents panel.
A list of available policies appears. A check in the Enabled column indicates an enabled policy.
3. Select the policies that you want to enable or disable.
4. Perform *one* of the following tasks:
– To enable the selected policy or policies,
click  .
– To disable the selected policy or policies,
Click  .
A check appears or disappears in the Enabled column accordingly.

NOTE
You can also enable or disable a policy using the Enabled field in the General Information subview for a specific policy.

Viewing Policies

Defined policies appear beneath Policy Manager in the Explorer tab and in the List tab in the Contents panel.



In the Explorer tab, the icons for enabled policies and their associated rules are green. Icons for disabled policies and rules are gray.

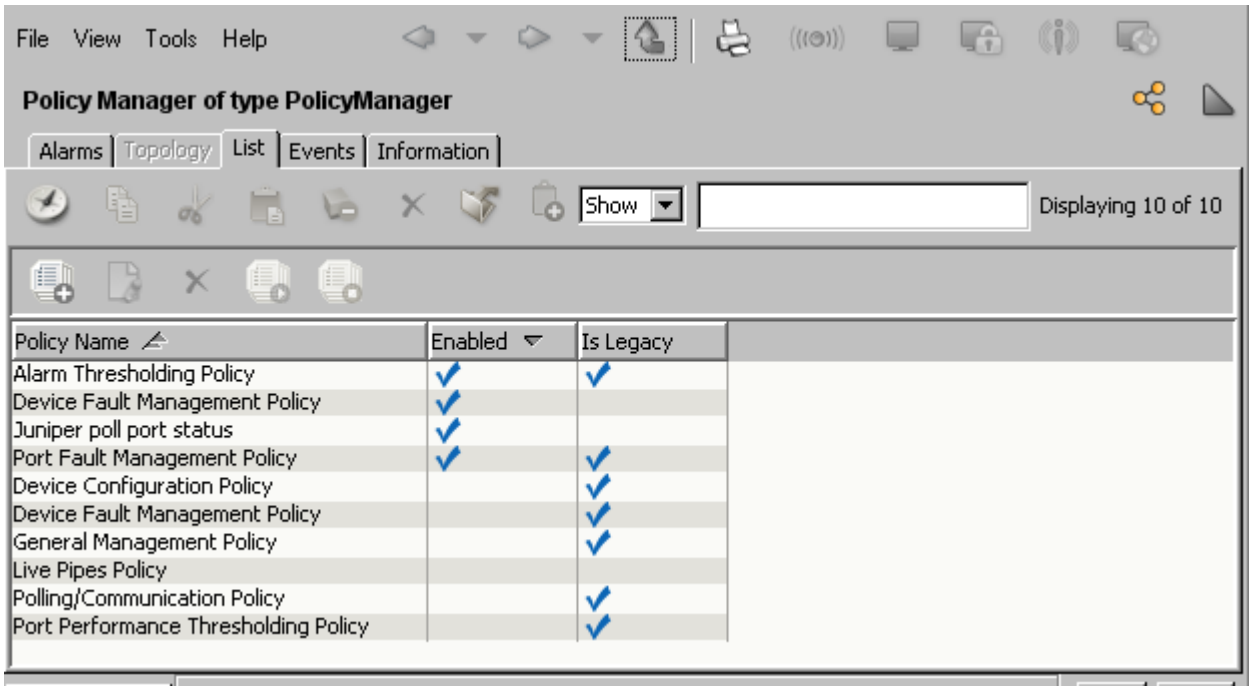
View All Policies

The following procedure describes how to view all existing policies.

Follow these steps:

1. Click Policy Manager in the Explorer tab.
   Information about Policy Manager is displayed in the Contents panel and the Component Detail panel.
2. Click the List tab in the Contents panel.
   A list of existing policies is displayed. From this view, you can create policies, and you can edit, delete, enable, and disable non-legacy policies.

   NOTE
   Legacy XML-based policies can only be edited, deleted, enabled, or disabled by modifying and reloading the policy definition XML. For more information, see Legacy XML-based Policies.



NOTE
You can also view all policies from the Locater tab: Policy Manager, All Policies.

View Policy Information

To view the details of a specific policy, select the policy in the Explorer tab under the Policy Manager node. The Contents and Component Detail panels are updated with information about the policy.

The List tab in the Contents panel contains information about the rules for the selected policy, including priority, name, global collections, and parent policy.

The Information tab in the Component Detail panel contains the following subviews:

- **General Information**
  Contains information about the policy, including security string and whether the policy has been enabled.
- **Global Collections**
  Lists the global collections to which this policy is applied.
- **Associated Rules**
  Lists the rules for the policy.

View Policies by Global Collection

You can view policies that are applied to a particular global collection.

Follow these steps:

1. In the Explorer tab, select the global collection for which you want to view Policy Manager policies.
   Information about the global collection is displayed in the Contents panel and the Component Detail panel.
2. Expand the Policy Manager Policies subview in the Information tab of the Component Detail panel.
   A list of existing policies identifies enabled policies and legacy policies.

View Policy Rule Information

To view the details of a policy rule, select the rule in the Explorer tab. Rules are located under policies in the Policy Manager node. The Contents and Component Detail panels are updated with information about the rule.

NOTE
To view all rules, use the Locater tab: Policy Manager, All Rules.

The List tab in the Contents panel contains information about the rules for the parent policy, including priority, name, global collections, and parent policy.

The Information tab in the Component Detail panel contains the following subviews:

- **General Information**
  Contains information about the rule, including priority, global collections to which it is applied, and security string.
- **Rule Settings**
  Lists the rule settings for this policy rule.
- **Affected Models**
  Lists the models affected by this policy rule.

NOTE
The policy must be enabled for this information to exist.

Search from the Locater Tab

Policy Manager provides multiple search criteria options for finding existing policies and rules in DX NetOps Spectrum. In addition to viewing policies from the Explorer tab, you can also search for specific policies and rules in the Locater tab**.**

NOTE
For more information about OneClick searches and configuration options, see Use OneClick .

Follow these steps:

1. Perform the following steps to display all policies or rules:
   a. Expand the Policy Manager folder in the Locater tab.
   b. Double-click the All Policies or All Rules option, as appropriate, to launch the search.
   c. Specify appropriate landscape information in the "Select Landscapes to Search" dialog and click OK.
   The Results tab displays search results appear in the Contents panel.
2. Perform the following steps for a criteria-based policy or rule search:
   a. Expand the Policy Manager folder in the Locater tab.
   b. Expand the Policies By or Rules By folder, as appropriate.
   c. Select the type of criteria-based search you want to run.
   d. Click the Search button.

   NOTE
   Depending on which search you select, you are prompted to enter values in a Search dialog before the search is executed.

   The Results tab displays search results in the Contents panel.

Editing Policies

You can change a policy name, the rules for the policy, and the rule settings. The following topics describe how to perform these tasks:

NOTE

- Only users with the appropriate privileges can edit policies. For more information, see Policy Manager Privileges.
- To modify the device or port models affected by a policy, edit the global collection and not the policy. For information about creating and maintaining global collections, see Modeling and Managing Your IT Infrastructure.

Edit a Policy

The following procedure describes how to edit a policy.

NOTE

- A policy must be disabled before it can be edited. For more information, see Enabling and Disabling Policies.
- Legacy XML-based policies can only be edited by modifying and reloading the policy definition XML. For more information, see Legacy XML-based Policies.

Follow these steps:

1. Click Policy Manager in the Explorer tab.
   Information about Policy Manager is displayed in the Contents panel and the Component Detail panel.
2. Click the List tab in the Contents panel.
   A list of policies is displayed.
3. Select the policy for editing and
   click 
   The Configure Policy dialog opens.
4. Modify the policy as needed and as described in Creating a Policy.
5. Click OK.
   The policy is updated and the Configure Policy dialog closes.
6. Enable the policy if needed.
   The updated settings are enforced.

Edit a Policy Rule

You can edit a specific policy rule and its settings without editing the entire policy. The parent policy must be disabled to edit one of its rules. The following procedure describes how to edit an existing policy rule.

- A policy must be disabled before its rules can be edited. For more information, see Enabling and Disabling Policies.
- Legacy XML-based policy rules can only be edited by modifying and reloading the policy definition XML. For more information, see Legacy XML-based Policies.

Follow these steps:

1. Select the rule to be modified in the Explorer tab.
   Note: You can also select the parent policy of the rule to be modified in the Explorer tab.
   Information about the selected rule and policy is displayed in the Contents panel and the Component Detail panel.
2. Click the List tab in the Contents panel.
   A list of the rules for the parent policy is displayed.
3. Select the rule for editing, and then perform one or more of the following tasks:
   – 

- Add a rule to the policy

– 

- Modify the selected rule

– 

- Copy the selected rule

The Configure Rule dialog opens.

4. Create or modify the rule as needed and as described in Creating a Policy.

   NOTE
   To modify an attribute value, click the modify the selected rule icon in the Rule Settings panel of the Configure Rule dialog.

5. Click OK.
   The rule is updated and the Configure Rule dialog closes.
6. Enable the parent policy if needed.
   The updated settings are enforced.

Editing a Rule Setting (Attribute Value)

You edit the attribute value by editing the rule itself. For more information, see Editing a Policy Rule.

Deleting Policies

The following procedure describes how to delete a policy. When you delete a policy, all rules and rule settings are also deleted.

NOTE

- A policy must be disabled before it can be deleted. For more information, see Enabling and Disabling Policies.
- Legacy XML-based policies can only be deleted by modifying and reloading the policy definition XML. For more information, see Legacy XML-based Policies.

Follow these steps:

1. Click Policy Manager in the Explorer tab.
   Information about Policy Manager is displayed in the Contents panel and the Component Detail panel.
2. Click the List tab in the Contents panel.
   A list of policies is displayed.
3. Select the policy to be deleted and

   click 
   The Delete Policy confirmation dialog opens.
4. Click Yes.
   The policy and all its associated rules and rule settings are deleted.

Delete a Policy Rule

The following procedure describes how to delete a policy rule. When you delete a policy rule, the rule settings for that rule are also deleted.

NOTE

- The parent policy must be disabled before one of its rules can be deleted. For more information, see Enabling and Disabling Policies.
- Legacy XML-based policy rules can only be deleted by modifying and reloading the policy definition XML. For more information, see Legacy XML-based Policies.

Follow these steps:

1. Select the rule to be deleted in the Explorer tab.

   NOTE
   You can also select the parent policy of the rule to be deleted in the Explorer tab.

   Information about the selected rule and policy is displayed in the Contents panel and the Component Detail panel.
2. Click the List tab in the Contents panel.
   A list of the rules for the parent policy is displayed.
3. Select the rule to be deleted and
   click 
   The Delete Rule confirmation dialog opens.
4. Click Yes.
   The rule and its rule settings are deleted. The rule priority values are adjusted accordingly.

Managing Policies

How to Check for Policy Enforcement

After a Policy Manager policy is enabled, you can verify results of its enforcement in the following ways:

- Check for generated events and alarms. An event is generated on the model when a rule cannot be enforced, such as when an attribute value cannot be written to a device. For more information, see Events and Alarms.
- Check the Affected Models subview of the rule. All models affected by the enforcement of the rule are listed. For more information, see Viewing Policy Rule Information.

Events and Alarms

DX NetOps Spectrum generates events and alarms to inform the user of Policy Manager activity.

An event is generated for conditions such as the following examples:

- When a policy is enabled or disabled.
- When the enforcement of a rule is successful.
- When a rule cannot be enforced, such as when an attribute value cannot be written to a device.
- When legacy XML-based policies are loaded or attempted to be loaded. Errors that occur during reloading are recorded in the event.

An alarm is generated for parsing errors that occur when legacy XML-based policies are reloaded.

Export Policies

You can export and import Policy Manager policies using DX NetOps Spectrum Modeling Gateway. This capability is useful when developing policies in a test environment and then moving them into a production environment. All related Policy Manager models, policies, rules, permissions, and templates are included.

NOTE
For more information about using the Modeling Gateway, see Modeling Gateway Toolkit .

Follow these steps:

1. Choose the SpectroSERVER from which to export Policy Manager policies, namely, where all policies exist. In a distributed SpectroSERVER environment, a policy may not exist on every SpectroSERVER, depending on the global collections it is associated with.

   NOTE
   You can also make policies exist temporarily on a SpectroSERVER by making all global collections for the policies exist on the SpectroSERVER.
2. On the chosen SpectroSERVER, modify the Modeling Gateway toolkit XML file to specify what to export:
   a. Open the following file for editing:
      `<$SPECROOT>/SS-Tools/.modelinggatewayresource.xml`
   b. Locate the ExportConfiguration tag and make the following edits below the tag:
      - Set the export_policy_manager and export_global_collections values to 'true'.

        NOTE
        You cannot select specific policies or global collections to export when using the Modeling Gateway. All policies and global collections are exported.
      - To avoid more exports, set all other values to 'false'.
   c. Save and close the file.
3. Export Policy Manager policies with the Modeling Gateway command-line tool, 'modelinggateway', located in the following directories.
   – On Linux:
      `<$SPECROOT>/SS-Tools>./modelinggateway -vnm vnm_name -e export_file`
   – On Windows:
      `<$SPECROOT>/SS-Tools>modelinggateway.bat -vnm vnm_name -e export_file`
   – ***vnm_name***
   is the name of the SpectroSERVER host
   – ***export_file***
   is the output file name
   The export process begins. Messages indicate the successful export of various models. Two files are created in the *<$SPECROOT>*/SS-Tools directory:
   – *export_file*.log - contains any error information
   – *export_file*.xml - contains exported Policy Manager data
4. Review the contents of *export_file*.xml to verify that all expected policies, rules, settings, and associations are included.

Import Policies

You can use the DX NetOps Spectrum Modeling Gateway to export and import Policy Manager policies. This capability is useful when developing policies in a test environment and then moving them into a production environment. All related Policy Manager models, policies, rules, permissions, and templates are included.

NOTE
For more information about using the Modeling Gateway, see Modeling Gateway Toolkit .

Follow these steps:

1. In the OneClick Console, disable and delete any policy that exists on the SpectroSERVER that you are going to replace with an imported policy. For more information, see Enabling and Disabling Policies and Deleting a Policy.

2. Review the contents of *export_file*.xml that was generated in the export procedure. Verify that all expected policies, rules, settings, and associations to be imported are included.
3. Import Policy Manager policies with the Modeling Gateway command-line tool, 'modelinggateway', located in the following directories.
   - On Linux:
     ```
     <$SPECROOT>/SS-Tools>./modelinggateway -vnm vnm_name -i export_file.xml
     ```
   - On Windows:
     ```
     <$SPECROOT>/SS-Tools>modelinggateway.bat -vnm vnm_name -i export_file.xml
     ```
   - ***vnm_name***
     is the name of the SpectroSERVER host
   - ***export_file.xml***
     is the name of the file containing the exported policy data

   The import process begins. Messages indicate the successful import of various models.
4. Verify in the OneClick Console that all policy information was correctly imported. For more information, see Viewing Policies.
5. Enable the policies as necessary. For more information, see Enabling and Disabling Policies.

Legacy XML-based Policies

In previous DX NetOps Spectrum releases, Policy Manager policies and rules could be developed and maintained using XML files exclusively. This DX NetOps Spectrum release continues to support these legacy XML-based policies with minimal integration into the OneClick Console interface. The topics in this section are provided to assist you during your migration from legacy policies to new OneClick Console-based policies.

> **WARNING**
> Using the DX NetOps Spectrum OneClick Console is the recommended and supported method for creating and maintaining policies in Policy Manager.

Maintaining XML-based Policies

Explanations and procedures describing how to create, modify, and maintain XML-based policies are provided in full in previous releases of DX NetOps Spectrum documentation. Please refer to the documentation provided in previous releases for details on maintaining your XML until you have fully migrated to OneClick Console-based policies.

Migrate from XML-based to OneClick Console-based Policies

Although legacy XML-based Policy Manager policies are still supported, you are encouraged to migrate your policies from XML to the OneClick Console-based format. The following process is a suggested workflow for this migration. We strongly recommend that you develop new policies in a test environment.

1. Identify a policy to be converted. Begin with your most basic policy.
2. Remove the policy from the XML file in the <*$SPECROOT*>/PolicyMgmt directory, then reload the policy.
   This step stops the policy from being enforced and deletes the policy.
3. Develop the same policy using the OneClick Console user interface, enable it, and test.

1. When the policy works as expected, use the Modeling Gateway to export the policy from the test environment and import it to the production environment.

Examples for Policy Settings

Configure the Device Fault Management Policy

This example shows you how to configure the Device Fault Management Policy using the OneClick Console. The policy settings in this example are predefined.

Follow these steps:

1. Click Policy Manager in the Explorer tab.
   Information about Policy Manager is displayed in the Contents panel and the Component Detail panel.
2. Click the List tab in the Contents panel.
   A list of existing policies is displayed.
3. Click 
   The Configure Policy dialog opens.
4. Type **Device Fault Management Policy** in the Policy Name field.
5. Click 
   The Configure Rule dialog opens.
6. Type **Device Fault Management Rule** in the Rule Name field.
7. Click Browse.
   The Select Global Collections dialog opens.
8. Create a global collection for the policy:
   a. Click Create.
      The Create Global Collection dialog opens.
   b. Type **All Devices** in the Name field.
   c. Click Search Options.
      The Search Options dialog opens.
   d. Click Show Advanced and click Add Existing.
      The Add Existing Search dialog opens.
   e. Expand the Devices folder, click All Devices, and click OK.
      The Add Existing Search dialog closes and the selected search criteria appears in the Expression field.
   f. Click OK.
      The Search Options dialog closes.
   g. Click OK.
      The Create Global Collection dialog closes. The All Devices global collection is created and added to the 'Applies to' list on the left.
9. Click OK.
   The Select Global Collections dialog closes and the All Devices global collection is added to this rule.
10. Specify the predefined settings for this policy:
    a. In the Rule Settings section of the Configure Rule dialog,
       click 
       The Select Template dialog opens.
    b. Select NoInvalidDLCIAlarms from the list of available templates.
       The Rule Settings list displays the policy settings that make up the No Invalid DLCI Alarms template.
    c. Click OK.
       The Select Template dialog closes and the settings are added to this rule.
    d. Select the first parameter, NoInvalidDLCIAlarms_1, and
       click 
       The Configure Attribute Setting dialog opens.

e. Set the Attribute Value to No, and click OK.
The Configure Attribute Setting dialog closes and the attribute value is defined.
f. Repeat the previous two steps for NoInvalidDLCIAlarms_2.
Values are now specified for all attributes.

11. Click OK.
The Configure Rule dialog closes and the rule is added to the policy.
12. Select 'Enable Policy on Creation' to enable and immediately enforce the policy when it is created.
13. Click OK.
The Configure Policy dialog closes. The Device Fault Management Policy is created, enabled, and appears in the policy list.

Configure the Alarm Thresholding Policy

This example shows you how to configure the Alarm Thresholding Policy for devices on your network. This example creates two alarm threshold policy settings: one that is applied to routers and one that is applied to switches.

The following attributes are used in this example:

- Value_When_Yellow (0x1000c)
- Value_When_Orange (0x1000d)
- Value_When_Red (0x1000e)
- Yellow_Threshold (0x10010)
- Orange_Threshold (0x10011)
- Red_Threshold (0x10012)

Follow these steps:

1. Click Policy Manager in the Explorer tab.
Information about Policy Manager is displayed in the Contents panel and the Component Detail panel.
2. Click the List tab in the Contents panel.
A list of existing policies is displayed.
3. Click 
The Configure Policy dialog opens.
4. Type **Alarm Thresholding Policy** in the Policy Name field.
5. Click 
The Configure Rule dialog opens.
6. Type **Router Alarm Threshold Rule** in the Rule Name field.
7. Click Browse.
The Select Global Collections dialog opens.
8. Create a global collection for the policy:
a. Click Create.
The Create Global Collection dialog opens.
b. Type **Routers** in the Name field.
c. Click Search Options.
The Search Options dialog opens.
d. Select 'Model Class (0x11ee8)' from the Attribute drop-down list.
e. Select 'Router' from the Attribute Value drop-down list.
f. Click OK.
The Search Options dialog closes.
g. Click OK.

The Create Global Collection dialog closes. The Routers global collection is created and added to the 'Applies to' list on the left.

9. Click OK.
   The Select Global Collections dialog closes and the Routers global collection is added to this rule.
10. Specify the predefined settings for this policy:
    a. In the Rule Settings section of the Configure Rule dialog,
       click 
       The Select Template dialog opens.
    b. Select AlarmThresholdingSettingsTemplate from the list of available templates.
       The Rule Settings list displays the policy settings that make up the Alarm Thresholding Settings template.
    c. Click OK.
       The Select Template dialog closes and the settings are added to this rule.
    d. Select the first parameter, AlarmThresholdingSettingsTemplate_1, and
       click 
       The Configure Attribute Setting dialog opens.
    e. Enter **2** for the Attribute Value and click OK.
       The Configure Attribute Setting dialog closes and the attribute value is defined.
    f. Repeat the previous two steps for the following settings:
       - AlarmThresholdingSettingsTemplate_2: **3**
       - AlarmThresholdingSettingsTemplate_3: **4**
       - AlarmThresholdingSettingsTemplate_4: **4**
       - AlarmThresholdingSettingsTemplate_5: **6**
       - AlarmThresholdingSettingsTemplate_6: **8**

       Values are now specified for all attributes.
11. Click OK.
    The Configure Rule dialog closes and the rule is added to the policy.
12. Click 
    The Configure Rule dialog opens.
13. Type **Switch Alarm Threshold Rule** in the Rule Name field.
14. Click Browse.
    The Select Global Collections dialog opens.
15. Create a global collection for the policy:
    a. Click Create.
       The Create Global Collection dialog opens.
    b. Type **Switches** in the Name field.
    c. Click Search Options.
       The Search Options dialog opens.
    d. Select 'Model Class (0x11ee8)' from the Attribute drop-down list.
    e. Select 'Switch' from the Attribute Value drop-down list.
    f. Click OK.
       The Search Options dialog closes.
    g. Click OK.
       The Create Global Collection dialog closes. The Switches global collection is created and added to the 'Applies to' list on the left.
16. Click OK.
    The Select Global Collections dialog closes and the Switches global collection is added to this rule.

17. Specify the predefined settings for this policy:
    a. In the Rule Settings section of the Configure Rule dialog,
    click 
    The Select Template dialog opens.
    b. Select AlarmThresholdingSettingsTemplate from the list of available templates.
    The Rule Settings list displays the policy settings that make up the Alarm Thresholding Settings template.
    c. Click OK.
    The Select Template dialog closes and the settings are added to this rule.
    d. Select the first parameter, AlarmThresholdingSettingsTemplate_1, and
    click 
    The Configure Attribute Setting dialog opens.
    e. Enter **1** for the Attribute Value and click OK.
    The Configure Attribute Setting dialog closes and the attribute value is defined.
    f. Repeat the previous two steps for the following settings:
    - AlarmThresholdingSettingsTemplate_2: **2**
    - AlarmThresholdingSettingsTemplate_3: **3**
    - AlarmThresholdingSettingsTemplate_4: **3**
    - AlarmThresholdingSettingsTemplate_5: **4**
    - AlarmThresholdingSettingsTemplate_6: **5**

    Values are now specified for all attributes.
18. Click OK.
    The Configure Rule dialog closes and the rule is added to the policy.
19. Select 'Enable Policy on Creation' to enable and immediately enforce the policy when it is created.
20. Click OK.
    The Configure Policy dialog closes. The Alarm Thresholding Policy is created, enabled, and appears in the policy list.

Recommended Policy Settings

Provided in this section are recommended policies that you can implement at your site. Each recommended policy is based on the settings in predefined settings templates. Each template configures the attributes in a different way. You select the policy setting that matches the way you want to implement your network management. You can also adjust the settings to suit your specific needs.

> **NOTE**
> Some attributes in these policy settings do not have predefined values. You can create your own settings for such attributes if necessary. If you do not want to enforce an attribute, simply remove it from the rule.

Port Fault Management Policy

The Port Fault Management Policy is used to maintain all port-level attributes related to fault management.

Policy Settings

This policy has four predefined settings templates:

- **Passive Port Monitoring**

These settings enable port status monitoring using only passive means. DX NetOps Spectrum listens for link down traps and generates alarms when needed. This method is the most efficient but least reliable means of port status monitoring. These settings are the default DX NetOps Spectrum settings.

- **Live Pipes**
  These settings enable port status monitoring using Live Pipes. DX NetOps Spectrum actively polls the status of ports in modeled connections. Colored pipes in all applications indicate the status of the connection. Trap-based monitoring is also enabled for expedited fault detection.
- **Poll Unconnected Ports**
  These settings enable port status monitoring for ports whose connectivity is not modeled in DX NetOps Spectrum. DX NetOps Spectrum actively polls the status of the port. Trap-based monitoring is also enabled for expedited fault detection.
- **Disabled Port Monitoring/No Alarms**
  These settings disable all port status monitoring methods and prevent any related alarms from being generated.

Attributes

The following attributes are used in the Port Fault Management Policy:

- **PollPortStatus**
  Attribute ID: 0x1280a
  Controls status polling of a port whose connectivity is not modeled.
- **ok_to_poll**
  Attribute ID: 0x11dd8
  Controls whether the pipe associated with this port is live. The status of the port is polled.
- **AlarmOnLinkDownTrap**
  Attribute ID: 0x11fc2
  Determines how DX NetOps Spectrum handles a Link Down trap on this particular port.
- **AssertLinkDownAlarm**
  Attribute ID: 0x12957
  Determines whether DX NetOps Spectrum generates a yellow alarm on the device model when a link-down trap is received for this port.
- **GeneratePortStatusAlarms**
  Attribute ID: 0x12a54
  Indicates whether a port status alarm is generated on this port.

Device Fault Management Policy

The Device Fault Management Policy is used to maintain all device-level attributes related to fault management.

Policy Settings

This policy has one predefined settings template:

- **No Invalid DLCI Alarms**
  These settings prevent DX NetOps Spectrum from generating red alarms on DLCI ports that have an ‘invalid’ state. Invalid DLCIs have a brown condition instead of a red condition.

Attributes

The following attributes are used in the Device Fault Management Policy:

- **PollPortStatus**
  Attribute ID: 0x12809
  Provides device-level control over the polling of port status for ports whose connectivity is not modeled.
- **support_ICMP**
  Attribute ID: 0x11d3d

Determines whether DX NetOps Spectrum attempts to contact a device using ICMP when SNMP contact is lost.

- **AlarmOnInvalidDLCIs**
  Attribute ID: 0x129ee
  Determines whether DX NetOps Spectrum generates red alarms on DCLI ports that have an 'invalid' state. When set to FALSE, invalid DLCIs have a brown condition instead of a red condition.

General Management Policy

The General Management Policy is used to maintain all device-level attributes related to general network management.

Policy Settings

This policy has two predefined settings templates:

- **Maintenance Mode**
  These settings suspend model management and put the model into Maintenance Mode. The model has a brown condition and no events or alarms are generated on the model. No SNMP requests are sent to the agent.
- **No Events Generated**
  These settings suspend event and alarm generation on the model. SNMP requests are sent to the agent.

Attributes

The following attributes are used in the General Management Policy:

- **isManaged**
  Attribute ID: 0x1295d
  Controls how DX NetOps Spectrum manages this model. When set to FALSE, DX NetOps Spectrum suspends management.
- **IsEventCreationEnabled**
  Attribute ID: 0x129f8
  Controls whether events are generated on the model. When set to FALSE, DX NetOps Spectrum stops generating events on the model, but SMNP and ICMP communication is still allowed.
- **Criticality**
  Attribute ID: 0x1290c
  Determines the relative significance of this device or port model. This value is used in determining the impact severity of a Contact Lost alarm. Any numeric value is supported.
- **DisableTrapEvents**
  Attribute ID: 0x11cd0
  Determines whether DX NetOps Spectrum escalates a trap into an event on a particular port model.
- **ContactStatusEventSwitch**
  Attribute ID: 0x11a56
  Determines whether DX NetOps Spectrum generates events when the Contact_Status of a device changes.

Polling/Communication Policy

The Polling/Communication Policy is used to maintain all device and port attributes for polling and communication with an SNMP agent.

Policy Settings

This policy has four predefined settings templates:

- **No Logging**
  Model statistics are not logged for the model.
- **1-Minute Polling**

The model is polled every 60 seconds.
- **5-Minute Polling**
  The model is polled every 300 seconds.
- **10-Minute Polling**
  The model is polled every 600 seconds.

Attributes

The following attributes are used in the Polling/Communication Policy:

- **PollingStatus**
  Attribute ID: 0x1154f
  Determines whether DX NetOps Spectrum polls the specified attributes of the model.
- **Polling Interval**
  Attribute ID: 0x10071
  Controls how often DX NetOps Spectrum polls this model.
- **Poll Log Ratio**
  Attribute ID: 0x10072
  Controls how often model statistics are logged. The actual interval is determined by multiplying the Polling_Interval by the Poll_Log_Ratio.
- **DCM Timeout (ms)**
  Attribute ID: 0x110c4
  Determines how long DX NetOps Spectrum waits to receive an SNMP response before sending a retry.
- **DCM Retry Count**
  Attribute ID: 0x110c5
  Determines the number of times that DX NetOps Spectrum attempts an SNMP get request before failing.
- **SNMP Community String**
  Attribute ID: 0x10024
  Specifies the SNMP password for communicating with an SNMP agent.
- **CommunityNameForSNMPSets**
  Attribute ID: 0x11a7f
  Specifies the SNMP password for performing an SNMP set. If this attribute is not filled in for a model, DX NetOps Spectrum uses the value of SNMP Community String.
- **Throttling**
  Attribute ID: 0x11f79
  Controls whether DX NetOps Spectrum restricts the amount of outstanding SNMP requests to a device. Throttling helps to alleviate problems involving SNMP agents that cannot handle large amounts of SNMP requests.
- **Throttle Count**
  Attribute ID: 0x11f39
  Determines how many outstanding SNMP requests are allowed when throttling is enabled for a device.
- **Agent_Port**
  Attribute ID: 0x10023
  Controls the port number for communicating with an SNMP agent.
- **Message Size**
  Attribute ID: 0x1197b
  Determines the largest packet size (in bytes) that DX NetOps Spectrum can send to an SNMP agent.

Alarm Thresholding Policy

The Alarm Thresholding Policy contains all the attributes that are related to the roll-up conditions and significance levels for models.

Policy Settings

This policy has one predefined settings template:

- **Alarm Thresholding Settings Template**
  These settings control the roll-up condition and significance level for a model.

Attributes

The following attributes are used in the Alarm Thresholding Policy:

- **Value_When_Yellow**
  Attribute ID: 0x1000c
  Specifies the significance level that the model inherits when its condition is yellow.
- **Value_When_Orange**
  Attribute ID: 0x1000d
  Specifies the significance level that the model inherits when its condition is orange.
- **Value_When_Red**
  Attribute ID: 0x1000e
  Specifies the significance level that the model inherits when its condition is red.
- **Yellow_Threshold**
  Attribute ID: 0x10010
  Specifies the threshold value that controls when the roll-up condition is yellow. The roll-up condition is yellow when its composite condition is greater than or equal to this value.
- **Orange_Threshold**
  Attribute ID: 0x10011
  Specifies the threshold value that controls when the roll-up condition is orange. The roll-up condition is orange when its composite condition is greater than or equal to this value.
- **Red_Threshold**
  Attribute ID: 0x10012
  Specifies the threshold value that controls when the roll-up condition is red. The roll-up condition is red when its composite condition is greater than or equal to this value.

Port Performance Thresholding Policy

The Port Performance Thresholding Policy contains all attributes for calculating and alarming on port performance.

Policy Settings

This policy has one predefined settings template:

- **Port Performance Thresholding Settings Template**
  These settings are used to calculate and alarm on port performance.

Attributes

The following attributes are used in the Port Performance Thresholding Policy:

- **% Utilization Threshold**
  Attribute ID: 0x1294b
  Specifies the threshold value for load on a port. When load is greater than or equal to this value, an alarm is generated.
- **% Utilization Reset**
  Attribute ID: 0x1294f
  Specifies the threshold value that controls when an alarm for load on a port is cleared. When load is less than this value, the alarm is cleared.
- **SET LEVEL IN LD**
  Attribute ID: 0x12d9f

Specifies the threshold value for receive load on a port. When receive load is greater than or equal to this value, an alarm is generated.

- **RESET LEVEL IN LD**
  Attribute ID: 0x12da0
  Specifies the threshold value that controls when an alarm for receive load on a port is cleared. When receive load is less than this value, the alarm is cleared.
- **SET LEVEL OUT LD**
  Attribute ID: 0x12da3
  Specifies the threshold value for the transmit load on a port. When the transmit load is greater than or equal to this value, an alarm is generated.
- **RESET LEVEL OUT LD**
  Attribute ID: 0x12da4
  Specifies the threshold value that controls when an alarm for transmit load on a port is cleared. When the transmit load is less than this value, the alarm is cleared.
- **SET LEVEL PR 64**
  Attribute ID: 0x12da7
  Specifies the threshold value for the packet rate on a port. When the packet rate is greater than or equal to this value, an alarm is generated.
- **RESET LEVEL PR 64**
  Attribute ID: 0x12da8
  Specifies the threshold value that controls when an alarm for the packet rate on a port is cleared. When the packet rate is less than this value, the alarm is cleared.
- **% Errors Threshold (micropercent)**
  Attribute ID: 0x1294d
  Specifies the threshold value for the error rate on a port. When the error rate is greater than or equal to this value, an alarm is generated. This attribute is in units of micropercent (1/1,000,000 of a percent). For example, 1 percent is entered as 1000000.
- **% Errors Reset (micropercent)**
  Attribute ID: 0x12951
  Specifies the threshold value that controls when an alarm for the error rate on a port is cleared. When the error rate is less than this value, the alarm is cleared. This attribute is in units of micropercent (1/1,000,000 of a percent). For example, 1 percent is entered as 1000000.
- **% Discarded Threshold (micropercent)**
  Attribute ID: 0x1294e
  Specifies the threshold value for the discard rate on a port. When the discard rate is greater than or equal to this value, an alarm is generated. This attribute is in units of micropercent (1/1,000,000 of a percent). For example, 1 percent is entered as 1000000.
- **% Discarded Reset (micropercent)**
  Attribute ID: 0x12952
  Specifies the threshold value that controls when an alarm for the discard rate on a port is cleared. When the discard rate is less than this value, the alarm is cleared. This attribute is in units of micropercent (1/1,000,000 of a percent). For example, 1 percent is entered as 1000000.

Device Configuration Policy

The Device Configuration Policy contains all attributes for how DX NetOps Spectrum automatically configures a device.

Policy Settings

This policy has one predefined settings template:

- **Disable Redundancy**

These settings control how DX NetOps Spectrum handles automatic device configuration. By default, DX NetOps Spectrum updates the model only when the primary address is accessible, even when a list of preferred redundant addresses exists.

Attributes

The following attributes are used in the Device Configuration Policy:

- **RedundancyEnabled**
  Attribute ID: 0x11d2c
  Specifies whether DX NetOps Spectrum updates the model when the primary address is not accessible and a list of redundant preferred addresses exists.
- **Rdnd_CheckGenAlarms**
  Attribute ID: 0x11dd6
  Controls whether DX NetOps Spectrum generates alarms when redundancy intelligence updates the network address.
- **If_IsAutoCnfgActive**
  Attribute ID: 0x11dd4
  Determines whether DX NetOps Spectrum automatically updates its modeling of interfaces when a change is detected on this device.
- **Create_Sub_Interfaces**
  Attribute ID: 0x11f3c
  Determines whether DX NetOps Spectrum models logical interfaces for this device.

  NOTE
  This setting applies only if this device supports RFC 1573.
- **DiscoverConnectionsAfterLinkUpEvent**
  Attribute ID: 0x11d25
  Controls whether DX NetOps Spectrum remodels the interfaces when this device sends a LINK UP or LINK DOWN trap.
- **DeviceDiscoveryAfterReconfig**
  Attribute ID: 0x11d27
  Determines whether DX NetOps Spectrum updates its knowledge of connections from device interfaces after a reconfiguration occurs.
- **IsMovable**
  Attribute ID: 0x11a80
  Controls whether DX NetOps Spectrum relocates the device model to a different topological location during the Discovery process.
- **IfModelNameOption**
  Attribute ID: 0x12a1e
  Controls the naming convention for interface models at the device level. The attribute ID is used to determine what suffix is appended to the model name for an interface model. Valid attribute IDs include:
  – 0x11f7e (ifAlias)
  – 0x1134b (ifDescr)
  – 0x11f6f (ifName)
  – 0x11348 (ifIndex)
- **Disposable_Precedence**
  Attribute ID: 0x114e2
  Determines the modeling precedence for this device. If a duplicate device is created with a higher precedence, the device model with the lower precedence is automatically destroyed.

Policy Manager Privileges

This section lists Policy Manager privileges for OneClick users.

NOTE
See OneClick Administration for more information about configuring privileges.

- **Policy Manager**
  Lets the administrator configure the Policy Manager application. Lets an operator view the Policy Manager application.
- **Explorer Add On Views/Policy Manager Hierarchy**
  Controls whether the Policy Manager node is displayed in the Navigation panel.
- **Policy Management**
  Controls access to the Policy Management privileges. Policy Management privileges are available for administrators with read/write privileges only. Deselecting the Policy Management privilege automatically deselects the following two privileges:
  - **Add/Edit/Delete Policies**
    Lets the administrator (AdministratorRW only) create, edit, and delete policies. This privilege does not let the user enable a policy.
  - **Enable/Disable Policies**
    Lets the administrator (AdministratorRW only) enable or disable a Policy Manager policy.
- **XML-based Policy Configuration/Reload Legacy Policies**
  Lets the administrator reload legacy XML-based policies.

How to Set up Policy Manager to Suppress Port Alarms

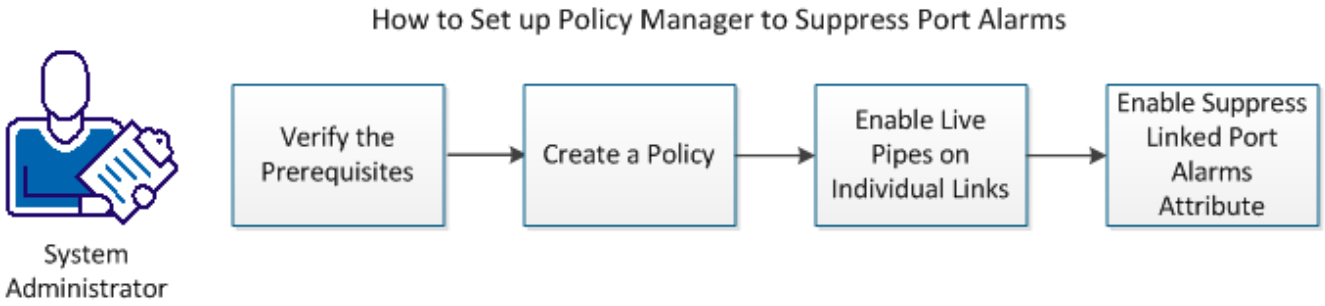
This scenario describes how a DX NetOps Spectrum administrator creates and uses a policy to suppress unwanted port alarms.

As a DX NetOps Spectrum system administrator you can suppress unwanted port alarms by creating a policy with the Live Pipes option in the Policy Manager. This policy suppresses duplicate alarms for a single problem and enables you to reduce network traffic and improve system performance.

The Live Pipes option allows you to enable port status monitoring for individual links. In the port status monitoring process, the new Live Pipes policy helps you to perform the following tasks:

- Suppress duplicate alarms on ports that have a broken connection or that are connected to a device unreachable by the SpectroSERVER.
- Suppress port alarms when the linked port model already has an alarm.

The following diagram shows how to set up Policy Manager to suppress port alarms:



Verify the Prerequisites

Before you create a policy to suppress port alarms, verify the following prerequisites:

You are logged in to the OneClick console as an administrator.

- You have the privileges and knowledge to create and enable a policy in the Policy Manager application.
- You understand the recommended policy settings and templates in the Policy Manager.

Create a Policy

Create a new policy in the OneClick console to suppress unwanted port alarms and to improve system performance by reducing network traffic.

> **NOTE**
> If you already have a policy with the Live Pipes option enabled, you can change the Suppress Linked Port Alarms attribute value to reduce port alarms.

A policy requires at least one rule. You can add more rules and settings later, as needed. All policy components are configured and maintained in the OneClick console. A policy contains the following components.

Policy DefinitionA set of prioritized policy rules. If a model resides in multiple global collections, policy rules are applied to the model based on the policy priority for a global collection.Policy RulesA collection of Rule Settings and the global collections to which the rule settings apply. You can define multiple rules for one policy.Rule SettingsDefine the model attributes and attribute values that a policy maintains.

Follow these steps:

1. Click Policy Manager in the Explorer tab of the OneClick console.
2. In the Contents panel, click the List tab, Create Policy and type a name.
   Example: SuppressPortAlarms
3. Click Create Rule and type a name.
4. Click Browse and select Global Collections for this policy.

   > **NOTE**
   > You can create global collections directly from the Select Global Collections dialog with the Create button.
5. Click Move to move the selected global collections to the Applies to list on the left, and click OK.
6. Click Predefined Settings in the Rule Settings box.
   The Select Template dialog opens.
7. Select Live Pipes from the Templates list and click OK.
   The Live Pipes settings are added to the Rule Settings.

   > **NOTE**
   > You cannot include the same attribute in more than one policy, regardless of whether the policy is enabled or not.
8. Click OK.
   The rule is added to the list of Associated Rules.
9. Select the rule and use the up and down arrows to adjust the rule priority in the list.

   > **NOTE**
   > The higher the rule is in the list, the higher the priority of the rule.

   The rules are adjusted in the list and the priority values are modified accordingly.
10. Select Enable Policy on Creation and click OK.
    The policy is created and enforced. The new policy appears in the List tab of the Contents panel.

Enable Live Pipes on Individual Links

To monitor the connection status between devices, enable Live Pipes on individual links from the Live Pipes subview. If an existing policy with the Live Pipes option is enabled for individual links, you can enable the Suppress Linked Port Alarms attribute in the Live Pipes subview to suppress unwanted port alarms.

Live Pipes for individual links are disabled by default. When an individual live pipe is enabled, the ok_to_poll attribute is set to TRUE for both ports in the link.

If Live Pipes are already enabled on individual links, continue to the next procedure.

Follow these steps:

1. Right-click the link that you want to enable as a live pipe, and select Enable Live Links.
2. Select the check box for the link to enable and click OK.
3. Click Set in the Live Pipes field and select Enabled from the drop-down list.
   The Enable Live Links dialog closes. The link that you selected is enabled as a live pipe.

Enable Suppress Linked Port Alarms Attribute

The Live Pipes subview lets you enable or disable the Suppress Linked Port Alarms attribute. Enable the attribute to suppress port alarms when the connected device is unreachable or the linked port model already has a red alarm.

Follow these steps:

1. Expand the Live Pipes subview in the Information tab of the OneClick console.
2. Select YES from the Suppress Linked Port Alarms drop-down list.
   The Suppress Linked Port Alarms attribute is enabled.

QoS Manager

DX NetOps Spectrum QoS Manager facilitates IP fault and performance management of networks that are configured for Quality of Service. QoS is deployed in IP data networks that carry traffic from services such as Voice Over IP (VoIP), order processing, and video conferences. QoS on routers lets the infrastructure serve the unique performance requirements of each service.
The QoS Manager performs the following tasks:

- Provides visibility into the health and performance of QoS traffic classes across the network.
- Automatically discovers QoS classes and policies on your network and maps them to the associated physical routers and interfaces.
- Creates models to represent each QoS policy, QoS traffic class, and QoS behavior.
- Creates traffic-class collections so you can view all identical traffic class models implemented on devices across the landscape.

QoS Manager also lets you drill down to per-class statistics such as packet drops, queue size, and pre-policy rates. You can view the individual performance statistics based on the QoS configuration type.

This section explains how to configure, discover, and manage the QoS elements in your network using DX NetOps Spectrum.

NOTE

Only Cisco devices that feature CISCO-CLASS-BASED-QOS MIB are supported.

Access the QoS Manager Interface

You can access the QoS Manager interface through the OneClick Console.
Follow these steps:

1. Launch the OneClick console from the OneClick homepage.

2. Expand the appropriate landscape in the Explorer tab and select QoS Manager.
   The QoS Manager interface appears.

   NOTE

   For information on using the OneClick Console, see Using OneClick .

Model Types

The following model types are available in the hierarchical view of the QoS Manager interface:

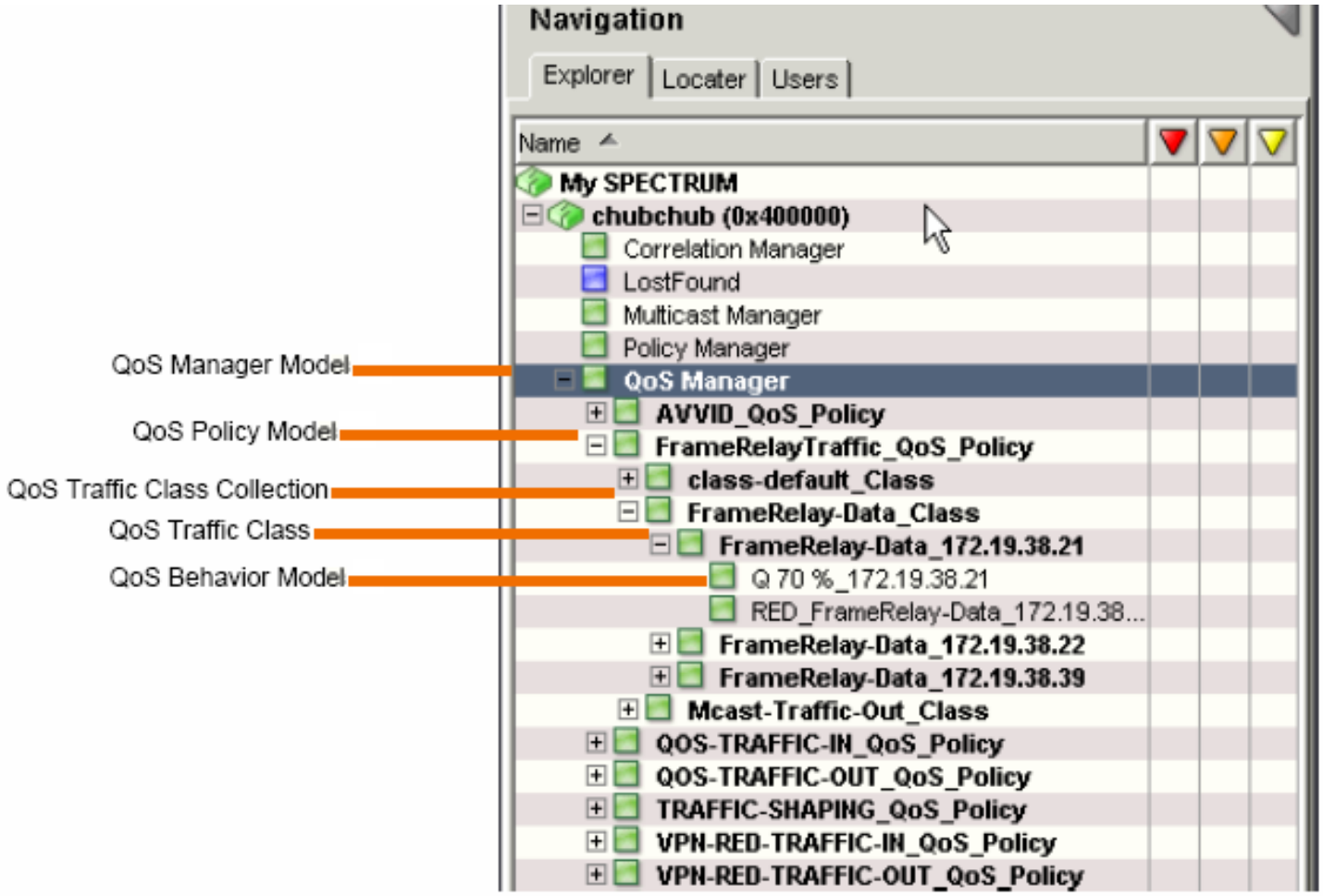
QosManager
Identifies the specific QoS Manager installation for a DX NetOps Spectrum landscape. Use this model type to configure alarm thresholds, launch QoS Discovery, and enable performance and report functionality.
Note: You need administrator privileges to use these configuration options.

QosPolicy
Defines the QoS policies that are set on your network. You can view general information, associated devices, and associated traffic classes of these models.

QosClassCollection
Lets you view all identical traffic class models across devices on your network. You can view general information, associated devices, and associated traffic classes of these models.

QosTrafficClass
Identifies the traffic classes that are defined on your network. You can view general information, associated devices, and associated behaviors of these models.

Behavior
Represents the behaviors per each hop defined on your network. There are four behavior model types: QosQueueing, QosRandomDetect, QosPolicing, and QosTrafficShaping. You can view the general information and performance statistics of these model types.
The following diagram shows the QoS Manager navigation hierarchy:



Search Options

The following search options are available on the OneClick Locater tab:

All Behaviors
Finds all QoS behaviors defined on selected landscapes.

All QoS Managers
Finds all QoS Manager models.

All QoS Policy Models
Finds all QoS policies defined on selected landscapes.

All QoS Services
Currently, this search is not enabled.

All Traffic Class Collections
Finds all Traffic Class Collection models defined on selected landscapes.

All Traffic Classes
Finds all Traffic Class models defined on selected landscapes.

Behavior By
Finds Behavior models by behavior, interface, name, or router.

QoS Policy By
Finds QoS Policy models by behavior, interface, name, or router.

Traffic Class By
Finds Traffic Class models by behavior, interface, name, or router.

NOTE
For more information on the Locater tab and performing a search, see Using OneClick.

QoS Manager Configuration

Access Configuration Parameters

The QoS Manager configuration parameters are provided in the Component Detail panel of the QoS Manager model. The Component Detail panel has three sections:

- General Information
- Configuration
- Traffic Behavior Alarm Thresholds

You should review these configuration parameters before you begin working with the QoS Manager to help ensure that the QoS Manager is set up to meet your network management needs.

NOTE
You need administrator privileges to make configuration changes.

QoS Manager Models

The following information can be accessed from the Information tab in the Component Detail panel for each QoS Manager model:

- **General Information**
  Displays the following general information about traffic class:
  - **Model Class**
    Indicates the model class for the QoS Manager model.

    NOTE
    For more information on a model type's model class, see SpectroSERVER and DX NetOps Spectrum Databases Overview..
  - **Creation Time**
    Indicates the date and time that this QoS Manager model was created.
  - **Security String**
    Indicates the security string for the QoS Manager model. If you have the required privileges, you can modify the security string by clicking Set, entering the security string into the available field, and clicking Save.
  - **Notes**
    Indicates the notes available for the QoS Manager. If you have the required privileges, you can add notes by clicking Set, entering the notes into the available field, and clicking Save.

Management Configuration

The Configuration section has the following parameters:

- **Enable Port Polling**
  Determines if the ports on a QoS-enabled device are polled.
  Default: Yes
- **Enable Performance Alarms**
  Lets you enable or disable the generation of performance alarms. Performance alarms are generated when a threshold defined in a Behavior model's Performance Component Detail has been violated.

Default: Yes

- **Enable Device Alarms**
  Enables or disables the generation of alarms on QoS-enabled devices.
  Default: Yes
- **Enable Statistic Polling**
  Enables the Performance Collection system.
  Default: No
- **Statistic Polling Interval**
  Controls how frequently statistical information is obtained for use in performance calculations. However, this polling interval will have no effect unless Enable Statistic Polling is set to Yes.
  Default: 300
- **Log Statistics to File**
  Enables you to send the performance statistics to a log file. No statistics will be captured and logged unless the Enable Statistic Polling parameter is set to Yes.
  Default: True
- **Minutes Per Log File Cycle**
  Specifies how often a new log file is created. No statistics can be logged unless the enable Statistic Polling parameter is set to Yes. The frequency at which statistics are written to the log file is defined by the Statistic Polling Interval described above.

Traffic Class Percent Dropped Thresholds

The Traffic Class Percent Dropped Thresholds section lets you define the critical threshold, major threshold, and minor threshold for violations based on percentage of packets dropped in any traffic class.

These thresholds indicate when critical, major, and minor alarms will be generated on a traffic class model. The default values are as follows:

- Five percent for a critical alarm
- Three percent for a major alarm
- One percent for a minor alarm

QoS Manager Traffic Behavior Alarm Thresholds

The Traffic Behavior Alarm Thresholds section lets you set the critical threshold, major threshold, and minor threshold for each type of traffic behavior, including policing percent dropped, queueing percent dropped, random early detect percent dropped, and traffic shaping percent dropped.

Each threshold sets percentage level of packets dropped that will cause a specific level of alarm. The default values are as follows:

- Five percent for a critical alarm
- Three percent for a major alarm
- One percent for a minor alarm

QoS Manager Discovery and Modeling

Device Discovery

DX NetOps Spectrum discovers and models the physical network infrastructure through OneClick Discovery, manual modeling, or the Modeling Gateway. Before using QoS Discovery, use one of these methods to model the physical components of your network in DX NetOps Spectrum.

NOTE

Only Cisco devices that feature CISCO-CLASS-BASED-QOS MIB are supported. For more information, see Certifications , OneClick Administration , and Modeling Gateway Toolkit.

QoS Manager requires that each QoS class have a unique name.

Reconfigure a QoS Device

If any of the QoS devices on your network were modeled before you installed QoS Manager, reconfigure them to create application models. Run the Reconfigure Model command on each of these models. Reconfiguration creates the CiscoCBQoSApp model, which supports the CISCO-CLASS-BASED-QOS MIB. This application provides necessary information to the QoS Manager.

Follow these steps:

1. Select the device model on the Contents panel.
2. Expand the Reconfiguration option on the Information tab in the Component Detail panel.
3. Click the Reconfigure Model button.
   The QoS application model is created for the device.

Run an On-Demand QoS Discovery

Use the QoS Discovery feature to discover and model the QoS policies, traffic classes, and behaviors defined on your network.

NOTE
If a QoS network element discovered and modeled by the QoS Manager is deleted from the network configuration, manually remove it from the modeled QoS hierarchy. It is not automatically removed by QoS Manager.

Follow these steps:

1. Select the QoS Manager model in the Explorer.
2. Select the Information tab in the Component Detail panel.
3. Select the Configuration option in the Component Detail panel.
4. Select the QoS Discovery option and click Run.
   The status bar indicates that the Discovery is running. Once the Discovery is complete, status returns to idle.

NOTE
Administrator privileges are required to run QoS Discovery.

Filter Traffic Class Names During Discovery

The QoS Manager lets you filter the Traffic Class during Discovery.

Follow these steps:

1. Select QoS Manager in the Explorer tab.
2. Select the Information tab in the Contents panel.
   The configuration options for QoS Manager appear.
3. Expand the QoS Discovery subview.
   The Discovery options that are available include the following settings:
   – **Traffic Class Filter Type**
   Determines if the Traffic Class in the 'Traffic Class Filter' field are included or excluded from modeling. Options include the following:

  - Exclusive
  - Inclusive
- **Traffic Class Filter**
  Lists the Traffic Classes to be included or excluded when the QoS Discovery is run. This field is used with the 'Traffic Class Filter Type' field.

  NOTE
  The Traffic Classes are not filtered and saved unless you add them in the Traffic Class Filter. Even if the Traffic Class Filter Type is inclusive and the Traffic Class Filter is empty, all Traffic Classes are discovered.

QoS Manager Model Naming Convention

During QoS Discovery, the QoS Manager assigns model names to policy, class collection, traffic class, and behavior models using the following conventions:

- **QoS Policy models**
  <*xxx*>_QosPolicy where <*xxx*> is the policy name assigned by the user who created the policy in the device's IOS Configuration.
- **Traffic Class Collection models**
  <*xxx*>_Class where <*xxx*> is the name of the traffic class assigned by the user who configured it.
- **Traffic Class models**
  <*xxx*>_QosPolicy_<yyy>_<*DeviceModelName*>, where:
  - <*xxx*> is the policy name assigned by the user who created the policy in the IOS Configuration of the device.
  - <*yyy*> is the name of the traffic class assigned by the user who configured it.
  - <*DeviceModelName*> is the device model name the user assigned to the device model using the traffic class.
- **Behavior models**
  Behavior models can be one of the following:
  - **Queuing Behavior Models**
    Q <*Bandwidth Units*> <*DeviceModelName*> where <*Bandwidth Units*> is the units of bandwidth used for queuing and <*DeviceModelName*> is the device model name the user assigned to the device model using the traffic class.
  - **RED Behavior Models**
    RED_<*ParentModelName*> where <*ParentModelName*> is the name of the parent traffic class model.
  - **Policing Behavior Models**
    Policing <Rate> <*DeviceModelName*> where <*Rate*> is the assigned policing rate and <*DeviceModelName*> is the device model name the user assigned to the device model using the traffic class.
  - **Traffic Shaping Behavior Models**
    TrafShaping <*Rate*> <*DeviceModelName*> where <*Rate*> is the assigned traffic shaping rate and <*DeviceModelName*> is the device model name the user assigned to the device model using the traffic class.

Change Policy and Class Model Names

You can change the default name that is assigned to a policy or class model.

Follow these steps:

1. Click the set link next to the model name on the Information tab.
2. Enter the new model name in the text box provided.
   The default name is changed.

Run QoS Discovery on Selected Models

You can configure the QoS Network Services Discovery from the OneClick views that display models. Then run QoS Network Services Discovery on selected models.

Follow these steps:

1. Select the models.
2. Click Tools, Utilities, Network Services Discoveries, QoS Discovery.
   The Discovery process is initiated. You can check the status in the Configuration subview.

Configuring QoS Discovery During Modeling

DX NetOps Spectrum lets you configure Network Services Discoveries, including QoS, during modeling. As a part of modeling configuration, you can specify which network service discoveries to run with the modeling process.

NOTE

For more information, see Modeling and Managing Your IT Infrastructure.

Model Types in QoS Manager

QoS Policy Models

The following information can be accessed from the Information tab in the Component Detail panel for each QoS policy model.

- **General Information**
  Displays the following general information about a QoS policy model:
  - **Condition**
    Defines the condition of the QoS policy model, which reflects the alarms that may be present on the model.
  - **Entity Condition**
    Indicates the calculated value of the model condition, which is based on the models that make up the QoS policy.
  - **Model Class**
    Specifies the model class of the QoS policy model.
  - **Policy ID**
    Indicates the policy identifier defined when the policy was created.
  - **Policy Type**
    Indicates the policy type selected when the policy was created.
  - **Security String**
    Defines the security string for this model. To change this value, click Set and enter the changes you want to make in the field provided. You must have administrator privileges to make changes to this field.
  - **Landscape**
    Defines the DX NetOps Spectrum landscape on which the policy exists.
  - **Description**
    Defines the description entered when the policy was created.
- **Associated Devices**
  Displays a table that shows all the devices that use this policy.
- **Associated Traffic Classes**
  Displays a table that shows all the traffic classes that are affiliated with this policy.

Traffic Class Collection Models

The following information can be accessed from the Information tab in the Component Detail panel for each traffic class collection model.

- **General Information**

Displays the following general information about a traffic class collection model:

- **Condition**
  Defines the condition of the traffic class collection model. This condition reflects the alarms that may be present on the model.
- **Entity Condition**
  Indicates the calculated value of the model condition, which is based on the models that make up the traffic class collection.
- **Traffic ID**
  Indicates the identifier defined when the traffic class collection was created. It uniquely identifies this traffic class collection.
- **Model Class**
  Defines the model class of the traffic class collection model.
- **Security String**
  Defines the security string for this model. To change this value, click Set and enter the changes you want to make in the field provided. You must have administrator privileges to make changes to this field.
- **Landscape**
  Defines the DX NetOps Spectrum landscape on which this traffic class collection exists.
- **Description**
  Indicates the description entered for this traffic class collection when it was defined.

- **Associated Devices**
  Displays a table that lists all of the devices that use the traffic classes contained in this traffic class collection.
- **Associated Traffic Classes**
  Displays a table that lists all of the traffic class models that make up this collection.

Traffic Class Models

The following information can be accessed from the Information tab in the Component Detail panel for each traffic class model.

- **General Information**
  Displays the following general information about a traffic class model:
  - **Condition**
    Defines the condition of the traffic class model. This condition reflects the alarms that may be present on the model.
  - **Entity Condition**
    Indicates the calculated value of the model condition, which is based on the models that make up the traffic class.
  - **Model Class**
    Defines the model class of the traffic class model.
  - **Traffic ID**
    Indicates the identifier defined when the traffic class was created. It uniquely identifies this traffic class. This is made up of the traffic class collection ID and the IP address of the interface that implements this traffic class.
  - **Security String**
    Defines the security string for this model. To change this value, click Set and enter the changes you want to make in the field provided. You must have administrator privileges to make changes to this field.
  - **Landscape**
    Defines the DX NetOps Spectrum landscape on which this traffic class exists.
  - **Match Statement**
    Defines the match statement defined for this traffic class. A match statement defines specific match criteria to identify packets for classification purposes.
  - **Description**
    Indicates the description entered when this traffic class was created.
  - **Drop Rate**

Specifies the rate of packets dropped aggregated across all interfaces supporting this traffic class.
  - **No Buffer Drops**
    Indicates the drop packet count, aggregated across all interfaces supporting this traffic class, which occurred due to a lack of SRAM buffers during output processing on this interface.
- **Associated Devices**
  Displays a table that lists all of the devices that use this traffic class.
- **Associated Behaviors**
  Displays a table that lists all of the behaviors that are applied to this traffic class.

Behavior Models

The following information can be accessed from the Information tab in the Component Detail panel for each behavior model.

- **General Information**
  Displays the following general information about a Behavior model:
  - **Condition**
    Defines the condition of the Behavior model. This condition reflects the alarms that may be present on the model.
  - **Entity Condition**
    Indicates the calculated value of the model condition, which is based on the interface models that use the behavior.
  - **Model Class**
    Indicates the model class of the Behavior model.
  - **Security String**
    Defines the security string for this model. To change this value, click Set and enter the changes you want to make in the field provided. You must have administrator privileges to make changes to this field.
  - **Behavior ID**
    Indicates the identifier defined when the behavior was defined. It uniquely identifies this behavior.
  - **Landscape**
    Defines the DX NetOps Spectrum landscape on which this behavior exists.
- **Performance**
  See Behavior Model Performance for the parameters provided in this selection.

QoS Alarms

QoS Manager generates alarms based on the configuration parameters you define. These alarms help define the status or condition of each of the modeled QoS components.

You can set a critical, major, and minor threshold for each type of behavior and traffic class model. Each threshold sets percentage level of packets dropped that will cause a specific level of alarm.

You can also enable the generation of alarms on QoS-enabled device models by setting the Enable Device Alarms parameter in the QoS Policy Manager model's Management Configuration to yes. The QoS Manager also generates alarms if there are problems logging statistical data.

The QoS Manager generates the following alarms:

- **0x4b30401**
  Defines the alarm generated on a QoS-enabled device model if one of the QoS Components exceeds a defined threshold.
- **0x4b30506**
  Defines the alarm generated on a behavior or traffic class model if the threshold set for a critical alarm is violated.
- **0x4b3050a**
  Defines the alarm generated on a behavior or traffic class model when the threshold set for a minor alarm is violated.
- **0x4b30000**

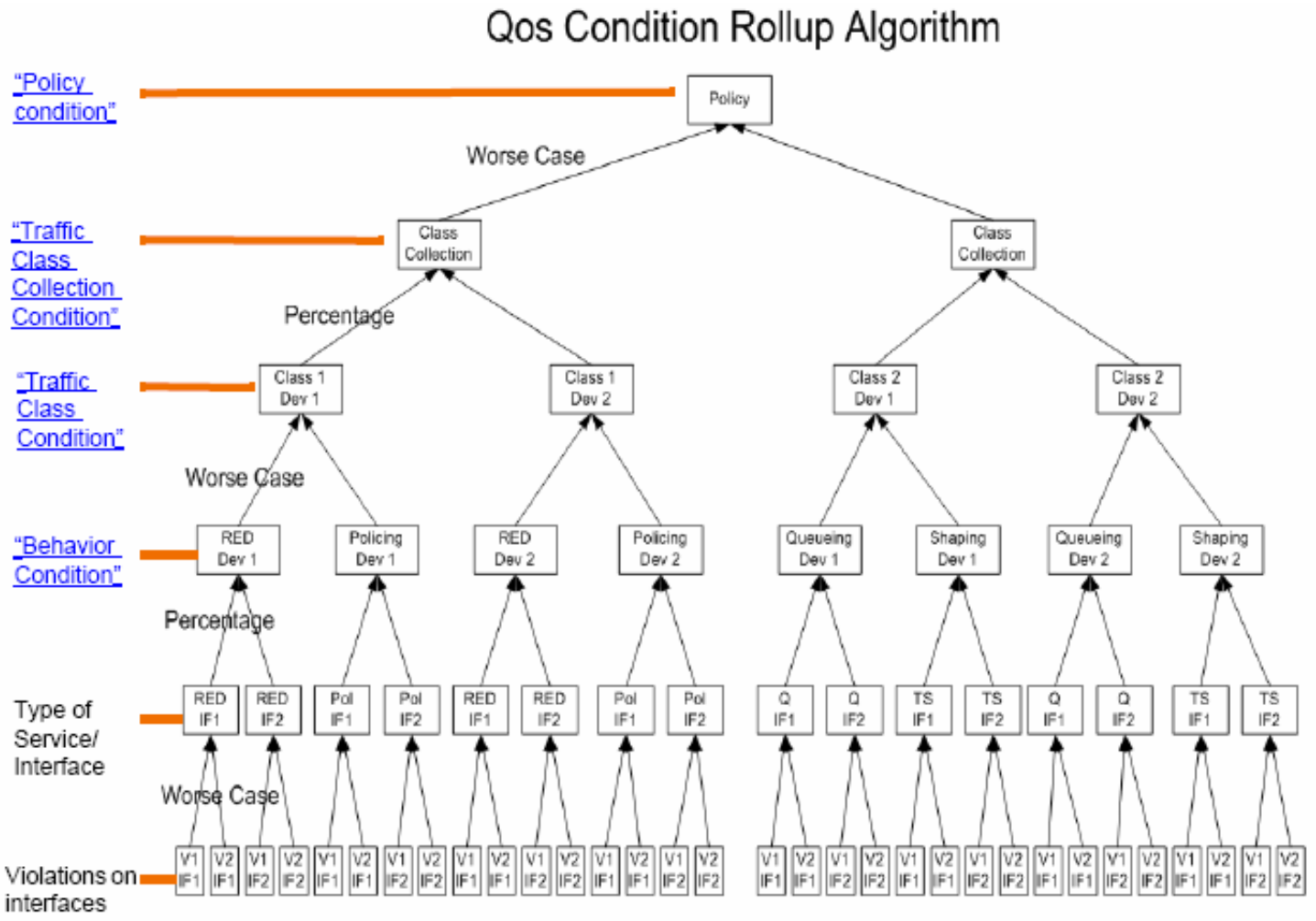
Indicates that the statistics log file could not be opened for writing. Because of this, at least one polling cycle's statistics will not be logged. This data is needed by other applications to determine the status of the network.

- **0x4b30001**
  Indicates that the statistics log file could not be closed. This may cause the statistical data to be lost and unused by other applications that may need it.

QoS Conditions

QoS Manager uses the QoS Condition Rollup algorithm to determine the entity condition of policy, traffic class collection, traffic class, and behavior models.

The following diagram illustrates the QoS Condition Rollup algorithm:



The conditions are as follows:

- **Policy**
  Is computed based on the status of the Class Collections contained in the policy. The condition of the Policy model is the worst condition found among all of the Traffic Class models monitored by the policy.
- **Traffic Class Collection**
  Is based on the status of the Traffic Class models contained in the collection. QoS Manager finds the condition that exists on the highest percentage of traffic classes contained in the collection. This predominant condition becomes the condition of the Traffic Class Collection model.

For example, if 45 percent of the traffic class models in the collection had a green condition, 10 percent had a red condition, and 20 percent had an orange condition, and 25 percent had a yellow condition, the condition of the collection would be green.

- **Traffic Class**
  Is based on the status of the Behavior models that are associated with the Traffic Class model. The condition of the traffic class model is the worst condition, or “worst case” found amongst all of the Behavior models in the traffic class.
- **Behavior**
  Is determined by the thresholds settings defined in QoS Manager traffic behavior alarm thresholds. These thresholds watch the percentage of dropped packets on a device for each specific service (queueing, random early detection, policing, or shaping) and designate what percentage level will generate a critical, major, and minor alarm condition. The percentage measurement used for these thresholds is derived from the average percentage of packets dropped for a particular service across all interfaces of the device. The percentage of packets dropped on each interface is equal to the worst percentage drop for a particular type of service on an interface.

QoS Performance

QoS Manager presents performance analysis information at both the traffic class and behavior level. In order for any of these statistics to be available, you must set the Enable Statistic Polling to Yes and set a value for the Statistic Polling Interval. These parameters can be found in the QoS Policy Manager model's Management Configuration.

View Traffic Class Performance Graphs

For each traffic class model, you can access graphed performance analysis based on Packet Drops and Pre-Policy Rate. The statistics for the graphs are derived from values in the CISCO-CLASS-BASED-QOS MIB.

To view traffic class performance graphs

1. Click the appropriate traffic class model in the OneClick Navigator.
2. Click the Performance tab in the Component Detail panel.
3. Select a graph from the drop-down list.
   The performance graph opens.

Behavior Model Performance

Performance information is available for each behavior model. The performance statistics provided depend on the type of service monitored by the behavior. These types of services are:

- queueing
- shaping
- policing
- random early detection

The statistics are derived from values in the individual service's Configuration and Statistics tables in the CISCO-CLASS-BASED-QOS MIB.

> **NOTE**
> The statistics provided for queueing (Mean Queue Depth), random early detection (Mean Queue Size), and shaping (Current Queue Size) are averages of the values gathered from each interface passing traffic for the given the service on the device. All other statistics are totals of values gathered from each interface passing traffic for the given service on the device.

Each behavior model has two ways to access performance information. The first method is to access the Information tab in the behavior model's Component Detail panel. This tab has a selection called Performance, which displays different statistics depending on the type of service the behavior model monitors. The data in this section does not refresh dynamically. To refresh the data, click the Refresh button.

This section also shows the critical, major, and minor percent violation thresholds for the behavior. These thresholds are set in the section QoS Manager traffic behavior alarm thresholds.

In addition, you can display the behavior model's performance information graphically by choosing the Performance tab in the behavior model's Component Detail panel. The graphs depend on the type of service the behavior model monitors. The data in these graphs refreshes dynamically.

Queueing Behavior Models

The following performance statistics are available on the Information tab for a queueing behavior model:

- **Configured Bandwidth**
  Indicates the configured bandwidth allocated to this traffic class.
- **Total Discarded Packets**
  Indicates the number of packets, associated with this class, that were dropped by queueing.
- **Average Queue Depth**
  Indicates the average queue depth in packets. This is an average of an average, because each value averaged by this statistic is based on the average queue depth for each interface.

The following performance graphs are available on the Performance tab for a queueing behavior model:

- **Queue Depth**
  Displays the value of the Average Queue Depth statistic over time.
- **Discard Rate**
  Displays the value of the Total Discarded Packets statistic over time.

Shaping Behavior Models

The following statistics are available in the Performance section of the Information tab for a shaping behavior model:

- **Configured Burst Size**
  Indicates the amount of traffic, in bits, in excess of the committed traffic-shaping rate that is instantaneously permitted.
- **Configured Extended Burst Size**
  Indicates the amount of traffic, in bits, in excess of the burst limit, which may be conditionally permitted.
- **Configured Traffic Shaping Rate**
  Displays the current adaptive traffic shaping rate.
- **Total Delayed Packets**
  Displays the total number of packets delayed.
- **Total Dropped Packets**
  Displays the total number of packets dropped.
- **Average Queue Size**
  Displays the traffic shaping queue depth in packets. Note that this is actually an average of an average, since each value averaged by this statistic is based on the average queue size for each interface.

The following performance graphs are available in the Performance tab for a shaping behavior model:

- **Queue Size**
  Displays the value of the Average Queue Size statistic over time.
- **Delays and Drops**
  Displays the values of the Total Delayed Packets and Total Dropped Packets statistics over time.

Policing Behavior Models

The following statistics are available in the Performance section of the Information tab for a policing behavior model:

- **Configured Burst Size**

Indicates the amount of traffic, in bytes, in excess of the committed policing rate that is permitted.

- **Configured Extended Burst Size**
  Indicates the amount of traffic, in bytes, in excess of the burst limit, which may be conditionally permitted by the policing feature. The probability that the traffic is not permitted increases as the received burst size increases.
- **Configured Policing Rate**
  Indicates the committed policing rate. This is the sustained rate permitted by policing.
- **Total Exceeded Packets**
  Indicates the number of packets treated as non-conforming by the policing service.
- **Total Violated Packets**
  Indicates the number of packets treated as violated by the policing service.

The Violated and Exceeded Packets graph is available in the Performance tab for a policing behavior model. This graph shows the values of the Total Exceeded Packets and the Total Violated Packets over time.

Random Early Detection Behavior Models

The following statistics are available in the Performance section of the Information tab for a random early detection behavior model:

- **Configured Decay Factor**
  Indicates the decay factor for the queue average calculation. The decay factor is equal to raising 2 to the power of N, where N could be up to 16. The smaller the number, the faster the decay.
- **Configured Mean Queue Size**
  Indicates the configured average queue size computed and used by the WRED algorithm.
- **Mean Queue Size**
  Indicates the average queue size computed and used by the WRED algorithm.
- **Total Random Drops**
  Indicates the number of packets dropped when the number of packets in the associated queue was greater than the minimum configured threshold and less than the maximum configured threshold.
- **Total Tail Drops**
  Indicates the number of packets dropped when the number of packets in the associated queue was greater than the maximum configured threshold.

The Mean Queue Size and Drops graph is available in the Performance tab for a random early detection behavior model. This graph shows the Mean Queue Size, Total Random Drops, and Total Tail Drops over time.

Remote Operations Suite

About Remote Operations Suite

Remote Operations Suite is the remotely deployed component of a DX NetOps Spectrum remote deployment. Remote Operations Suite is a stand-alone DX NetOps Spectrum management system that delivers all the standard DX NetOps Spectrum management capabilities. These capabilities include traditional fault-tolerant DX NetOps Spectrum management.

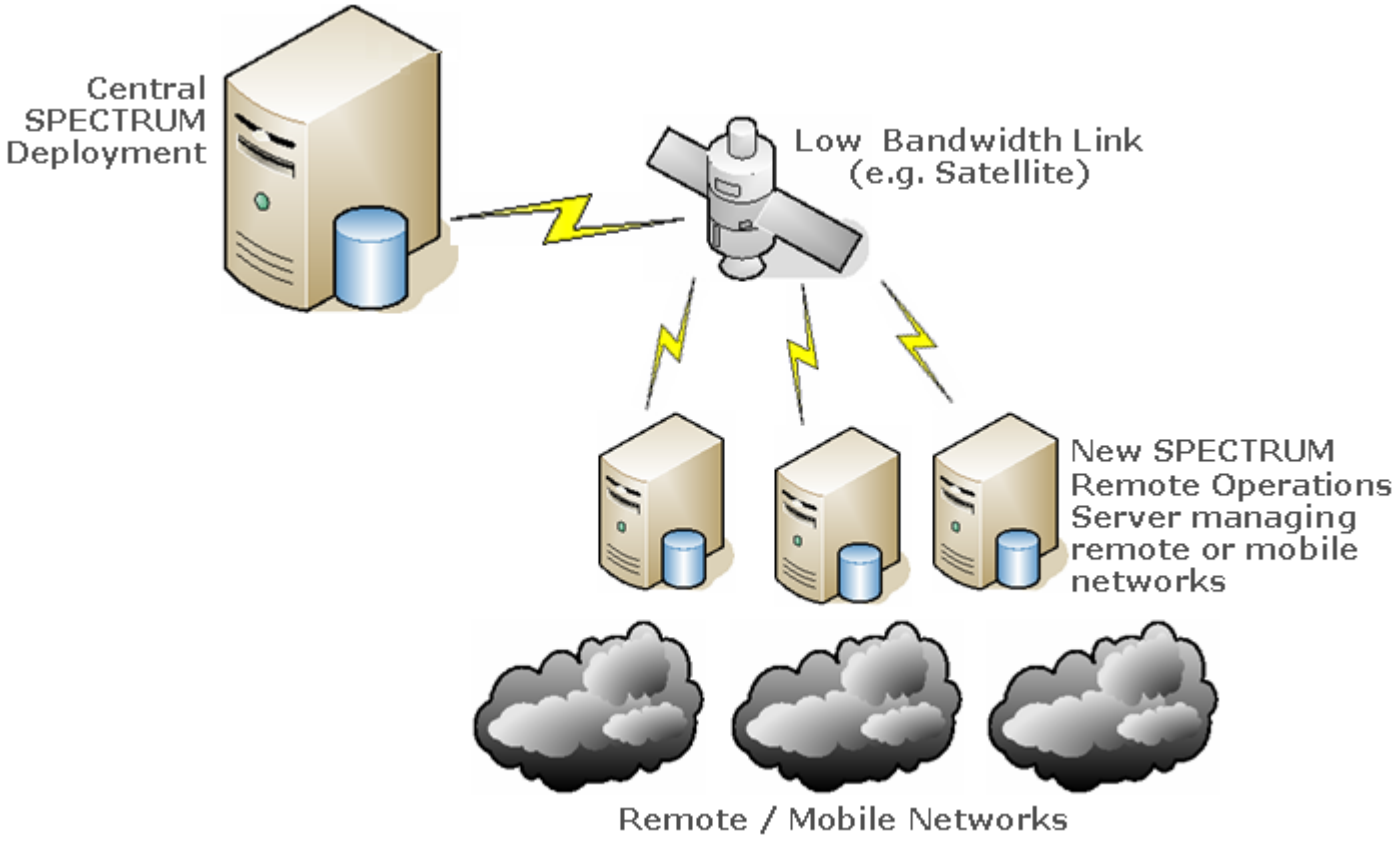
Remote Operations Suite supports operations in remote or mobile tactical environments. The remote component communicates with a designated Central SPECTRUM server to deliver real-time, or on-demand network topology. Remote Operations Suite also delivers the operational status of the managed remote site, mobile network over unreliable, or limited bandwidth communication links.

NOTE
In the environments where higher-speed communications links are more readily available, deploy the traditional DX NetOps Spectrum distributed architecture.

NOTE
Remote Operations Suite performs the following functions:

- Enables the network manager at the remote or mobile site to manage the local network infrastructure
- Delivers all the standard DX NetOps Spectrum management capabilities (for example, network discovery, root cause analysis, impact analysis, alarming/alerting)
- Connects to another DX NetOps Spectrum deployment situated in global network operations or central command to deliver real-time inventory and operational status of the remote network
- Enables an administrator to determine the information that the remote site sends to the central site to be low-impact on the limited bandwidth (and often costly) communications links

NOTE
The following architecture shows the Remote Operations Suite deployment:



NOTE
Remote Operations Suite uses the OneClick Console. For more information about OneClick, see the Using OneClick section.

Remote Operations Suite includes the following components:

- **Remote Operations Server**
  Is the remotely deployed DX NetOps Spectrum installation of a Remote Operations Suite deployment. The Remote Operations Server is a lightweight, stand-alone DX NetOps Spectrum Management system that performs traditional fault tolerant DX NetOps Spectrum management, including fault management and root cause analysis for the network

it manages. The Remote Operations Server communicates with its designated Central SPECTRUM Server through the installed Remote Operations Connector component. This architecture enables the delivery of network topology and operational alarm status over unreliable or limited bandwidth communication links.

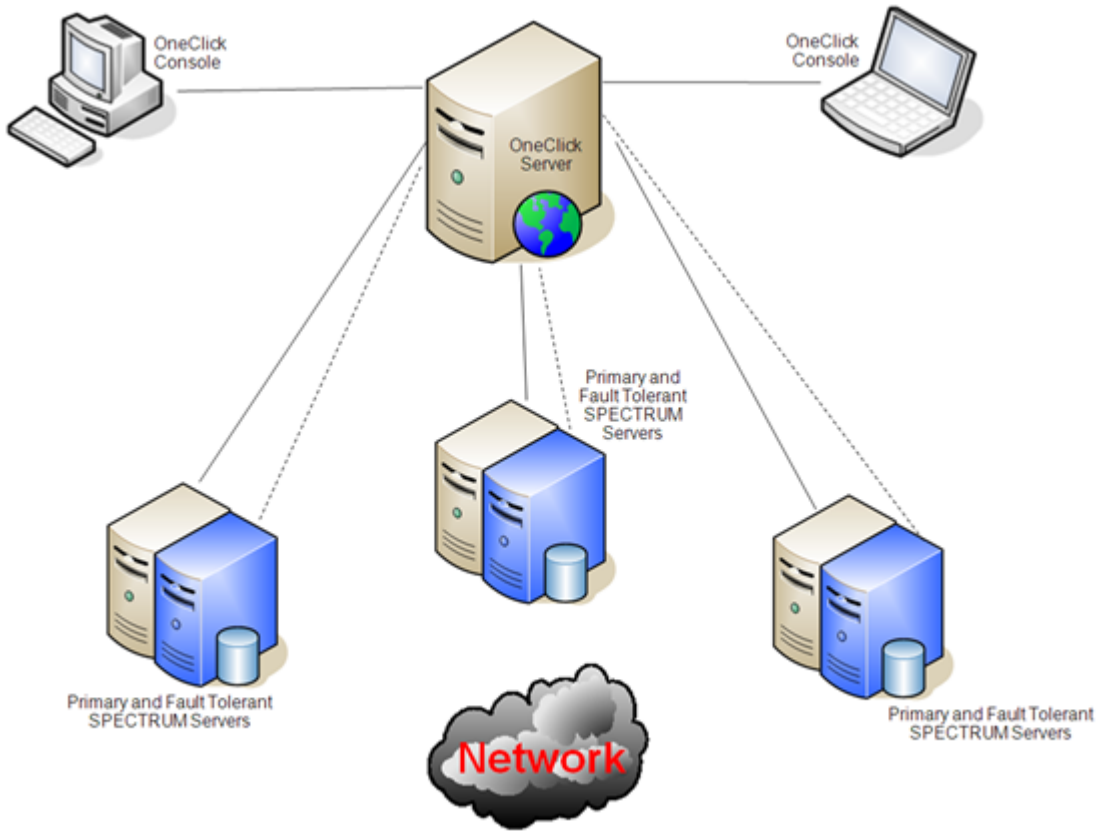
- **Central SPECTRUM Server**
  Is a separate installation of DX NetOps Spectrum that includes the Remote Operations Manager component. This component listens for new connections from Remote Operations Servers. A single Central SPECTRUM Server with the Remote Operations Manager component installed can integrate with multiple Remote Operations Servers. The integrated servers let you discover and monitor managed elements in the Universe hierarchy of the Remote Operations Server.
  The Remote Operations Manager component on the Central SPECTRUM Server communicates with the Remote Operations Connector component on all connected Remote Operations Servers. The remote server instructs connectors to forward all current topology data. Each Remote Operations Connector component receives the requested information and returns the topology data to the Remote Operations Manager component on the Central SPECTRUM Server, where the topology is modeled in the database.

Distributed Concepts in the Remote Operations Suite

The distributed capability of a Remote Operations Suite environment differs from the distributed capability of a traditional DX NetOps Spectrum environment. Remote Operations Suite has unique features to support distributed deployments.

A traditional DX NetOps Spectrum Distributed SpectroSERVER (DSS) environment deploys multiple DX NetOps Spectrum servers, which are sometimes referred to as SpectroSERVERs or landscapes. Each SpectroSERVER manages a different portion of the network. In this environment, you can see a combined or rolled-up view of the entire network using the OneClick Console as an aggregation point. OneClick provides monitoring and administration capabilities for each DX NetOps Spectrum server in the distributed environments. You can also designate primary and secondary DX NetOps Spectrum servers for the fault tolerance.

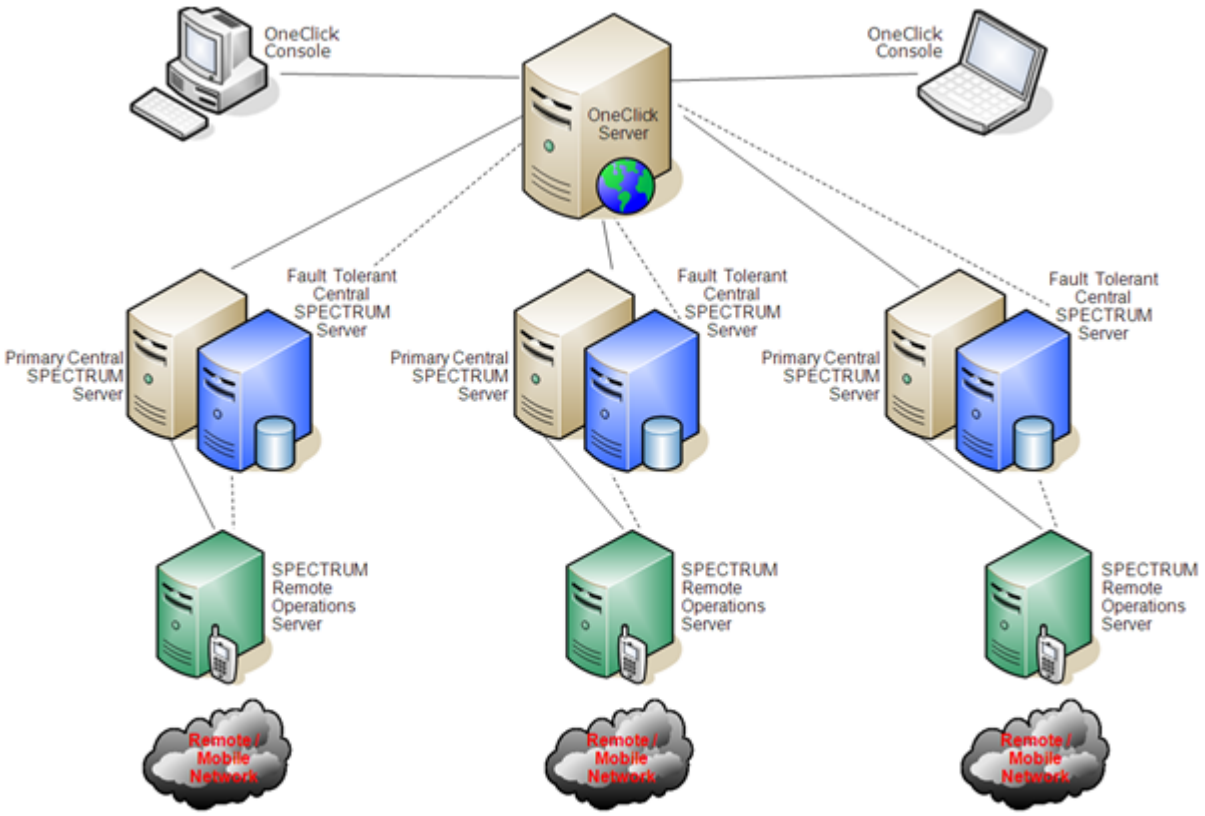
The following architecture shows a traditional DX NetOps Spectrum distributed deployment:



Remote Operations Suite also lets you deploy multiple Remote Operations Servers to manage a set of remote or mobile network devices. You can configure each Remote Operations Server to forward inventory information (for example, device and interface information) and the associated alarm data to a designated primary Central SPECTRUM Server. The primary Central SPECTRUM Server acts as the aggregation point for multiple Remote Operations Servers. Administrators at the central site have access to all of the information that has been forwarded from the Remote Operations Servers, by viewing each one on the primary Central SPECTRUM Server.

Fault tolerance features are also supported in a distributed Remote Operations configuration. You can configure each Remote Operations Server to connect to both a primary Central SPECTRUM Server and a backup Central SPECTRUM Server. All topology and alarm data is forwarded to the primary Central SPECTRUM Server. If the connection to the primary central SPECTRUM server is lost, the Remote Operations Server automatically sends a full topology and alarm update to the designated backup Central SPECTRUM Server. The Remote Operations Server then begins forwarding alarm information to the backup server. Once the connection to the primary Central SPECTRUM Server is restored, data forwarding to the backup Central SPECTRUM Server stops. The Remote Operations Server sends a full topology and alarm update to the primary Central SPECTRUM Server, and normal alarm forwarding resumes.

The following diagram shows a distributed Remote Operations deployment:



Management Visibility in a Remote Operations Environment

When a Remote Operations Server manages a network asset directly, you have access to all information that has been gathered about the device, regardless of the various methods DX NetOps Spectrum uses to collect that information. Direct management is therefore equivalent to management by a traditional DX NetOps Spectrum server. A difference appears in the available management data when the device is managed by a Remote Operations Server, but the information is viewed from a connected Central SPECTRUM Server.

When a Remote Operations Server forwards its topology to a designated Central SPECTRUM Server, the devices are represented in the Central SPECTRUM Server as “lightweight” models. Viewed locally by operators using the Remote Operations Server, the same models are "complete" models, providing complete local visibility. But the Central SPECTRUM Server only displays the essential information about these devices. You can still clearly identify the managed devices and their subcomponents. In addition, all alarms impacting the managed assets can be transferred from the Remote Operations Server to the Central SPECTRUM Server.

Less information is available from the Central SPECTRUM Server because Remote Operations Suite must operate over low-speed, unreliable connections. The communications that occur between the Remote Operations Server and the Central SPECTRUM Server are therefore optimized to conserve bandwidth.

The Remote Operations Server communicates only the information that is required to provide the Central SPECTRUM Server with a real-time view of its managed topology and operational alarm state.

Setting Up Remote Operations Suite

Install the Remote Operations Suite

Installing Remote Operations Servers resembles the procedure for installing DX NetOps Spectrum. The installers are similar. However, a few additional steps are required.

Designating a Main Location Server is required as part of the installation. Be sure to read Selecting a Main Location Server before you begin the installation.

Follow these steps:

1. Start the installation on Windows or Linux.
   The Install dialog opens.
2. Select the Install DX NetOps Spectrum option.
   The Introduction dialog opens.
3. Click Next to proceed.
   The License Agreement dialog opens.
4. Scroll through and read the license agreement, accept the agreement, and click Next.
   The Destination Host dialog opens.
5. Enter the name of the host system where you are installing DX NetOps Spectrum and click Next.
   The SRAdmin Authentication dialog opens.

   NOTE
   If the 'Unable to connect to DX NetOps Spectrum Remote Administration Daemon (SRAdmin)' dialog appears, install SRAdmin before continuing with the installation. To install SRAdmin, click Install on this dialog.
6. Enter a username and password as follows, and click Next:
   – For Linux installation, enter a username with root access. Or, you can use a sudoers file for root permissions.

     NOTE
     If you have root access when starting this installation, you are not prompted for a user name and password.
   – For a Windows installation, enter a username that has Administrator rights, and verify the domain name (if applicable).

   The Destination Location dialog opens.
7. Click Next to install DX NetOps Spectrum in the default directory. The default directory is C:\win32app\SPECTRUM on Windows and /usr/SPECTRUM on Linux.
   To install in a location other than the default directory, click Choose, select a location, and click Next.
   The installer reports that it is extracting installation information.
   The Select Destination Language dialog opens.
8. Select one of the supported languages to install in addition to English, and click Next.
   Localized CsEvFormat, CsPCause and EventTables will be installed for the selected language.
   The Select Options dialog opens.
9. Select Remote Operations Server as the Installation Type.
10. Select to install both the SpectroSERVER and OneClick from the list, and click Next.

    WARNING
    The Connector must be installed on a system with both OneClick and the SpectroSERVER installed.

    The Host Evaluation dialog opens.
11. Scroll down to verify that no warnings appear, and click Next to proceed.
    The DX NetOps Spectrum Installation Owner dialog opens.
12. Enter a username and password, and click Next. This username is used to create the initial DX NetOps Spectrum user (for the SpectroSERVER) and becomes the installation owner. For a OneClick installation, the username also determines the SpectroSERVERs to which the OneClick web server connects.
    For more information, see the Installing section.

NOTE
For first-time installations, the default DX NetOps Spectrum password for the installation owner is spectrum.

WARNING
When installing OneClick, be sure to specify a DX NetOps Spectrum username to which the administrative license is associated. This user needs access to all models in DX NetOps Spectrum (ADMIN access). We recommend that you specify the installation owner that you specified during the SpectroSERVER installations. This user must also exist on the installation host and does not have to be a Windows administrative user.

The Main Location Server dialog opens.

WARNING
Be sure to read Selecting a Main Location Server before you select a server. For more information about location servers and the Main Location Server, see the Distributed SpectroSERVER Administration section.

13. Enter a hostname for the Main Location Server and click Next. The Main Location Server should be installed on a separate server.

NOTE
DX NetOps Spectrum must be able to resolve the hostname, regardless of whether you provide a fully qualified hostname.

The Web Server Port Number dialog shows the default value.

14. (Optional) Enter a port number other than the default, and click Next.

NOTE
The default port is 80 for Windows and 8080 for Linux.

WARNING
During a typical SpectroSERVER installation, you are prompted to supply a landscape handle. The Remote Operations Server installation does not prompt for this value. Instead, the installer supplies a default value of 16. In some situations, multiple SpectroSERVERs are deployed with the Remote Operations Server. You can then change the landscape handles of selected SpectroSERVERs. For more information, see Set the Landscape Handle for the SpectroSERVER.

15. Click Next.
    The Review Settings dialog opens.
16. Scroll down to ensure all the settings are what you selected and click Next.
    The Installing DX NetOps Spectrum dialog appears. After DX NetOps Spectrum is installed, the status changes to 'Installation successful,' and the Next button is enabled.
17. Click Next.
    The Installation Complete dialog opens.
18. Click Done.
    The configuration dialog appears for a brief moment and closes.
    DX NetOps Spectrum is configured for your system.
19. Click Close on the initial Install dialog. Log out, and log back in.
    DX NetOps Spectrum is installed.
    Next, you must Configure the Remote Operations Connector.

Set the Landscape Handle for the SpectroSERVER

The Landscape Handle dialog that prompts you during DX NetOps Spectrum installation does not appear during Remote Operations Server installation. Instead, the handle setting for the SpectroSERVER defaults to 16. To use a different value for the landscape handle, you must set it. Unique landscape handles are crucial if you are configuring a distributed SpectroSERVER environment. Use the default handle setting for Multiple Remote Operations Servers if a Remote Operations Server does not report to another Remote Operations Server as its MLS.

Landscape handles can be assigned by using a utility named lh_set. For more information see the **Assign Landscape Handles** section in Setting up a Distributed SpectroSERVER Environment.

> **WARNING**
> Run the lh_set utility before you run the SpectroSERVER for the first time. Otherwise, DX NetOps Spectrum assigns a default landscape handle that is the same every time that DX NetOps Spectrum assigns it. As a result, duplicate landscape handles can be created when multiple landscapes are configured. Such landscapes can never be accessed simultaneously from the same application.

Follow these steps:

1. Navigate to the SS directory.
2. Enter the following command:
   ```
   ../SS-Tools/lh_set <landscape handle>
   ```
   You can specify the new landscape handle in either decimal or hexadecimal notation. If you use decimal notation, the lh_set utility converts your entry into a hexadecimal landscape handle.

   > **NOTE**
   > For more information, see the Installing section.

Selecting a Main Location Server

During Remote Operations Suite installation, you are prompted to select a Main Location Server. The *main location server (MLS)* is the primary SpectroSERVER used to coordinate the information and events from all other SpectroSERVERs connected in a DSS environment. Typically, a Remote Operations Server deployment does not use the same architecture as a typical DSS deployment.

In a Remote Operations Suite deployment, select the Remote Operations Server as the Main Location Server.

> **WARNING**
> Do not select the designated primary Central SPECTRUM Server as the MLS.

The Remote Operations Server OneClick points to the Remote Operations SpectroSERVER as the MLS. The interactivity to the full DX NetOps Spectrum environment comes from configuring the Remote Operations Server Connector on the server VNM model to point to a SpectroSERVER in the standard DX NetOps Spectrum environment.

If your environment uses Remote Operations Servers in a DSS deployment, select your MLS, but in such a case, the MLS should not be one of your Full Production SpectroSERVERs.

Configure the Remote Operations Server

When the installation completes and you have logged out of the server and logged back in, you are ready to enable the Remote Operations Connector component. This component links the Remote Operations Server to the Remote Operations Manager on the Central SPECTRUM Server. Configure the Remote Operations Connector with the information required to send the topology and alarm information to the Central SPECTRUM Server.

Follow these steps:

1. Log in to the OneClick server that is associated with the Remote Operations Suite deployment.
2. Launch the Spectrum Control Panel.
3. Start the SpectroSERVER process, which is equivalent to starting the Remote Operations Suite.
4. In OneClick, locate the new Remote Operations Server in the Explorer tab of the Navigation panel.
5. Expand the landscape, and expand the Universe.
6. Select the Virtual Network Manager (VNM) model.
   Information about the VNM model appears in the Component Detail panel.

We recommend configuring the Connector parameters in the following order:

a. Topology Forwarding
b. Alarm Forwarding
c. Central SPECTRUM Server - General

Configure Topology Forwarding

The Remote Operations Connector provides a representation of all modeling data in the Universe hierarchy of the associated Remote Operations Server. The Universe hierarchy includes containers, devices, ports, and applications. The Remote Operations Connector sends the topology information to the Remote Operations Manager component of the Central SPECTRUM Server. By default, the entire network topology under the Universe hierarchy is forwarded.

Configure the topology forwarding feature to selectively forward topology components. You can designate the components of the network topology that you want the Remote Operations Connector to forward to the Central SPECTRUM Server.

Follow these steps:

1. Locate the Remote Operations Connector configurations and expand the Topology Forwarding subview.
   The Topology Forwarding configuration options display.
2. Click set next to Hierarchy and select the aspects of the selected Remote Operations Server hierarchy you want to forward to the Central SPECTRUM Server:
   – Entire Universe hierarchy (including all containers)
   – Devices Only
   The selected value displays next to Hierarchy.

Click set next to Device Sub-Components and select the Remote Operations Server subcomponents to forward to the Central SPECTRUM Server:

- All Device Sub-Components (ports and applications)
- Interfaces Only
- Applications Only

NOTE
Remote Operations Connector always forwards all device models to the Central SPECTRUM Server. To forward alarms on port and application models to the Central SPECTRUM Server, configure the Remote Operations Connector to forward these models too. For example, if only device and interface models are sent to the Central SPECTRUM Server, then only device and interface alarms are forwarded. Alarms on application models are not forwarded.

The selected value appears next to Device Sub-Components.

NOTE
Do not click Update Remote Operations Topology after an initial configuration of the product. The topology data of the Remote Operations Server is sent to the Central SPECTRUM Server once you establish the initial connection.

Topology forwarding is configured.

Configure Alarm Forwarding

The Remote Operations Servers generates the alarms. The alarms are forwarded to the Remote Operations Manager component on the Central SPECTRUM Server by the Remote Operations Connector.

Configure the alarm forwarding feature to specify the severity of alarms that are forwarded to the Central SPECTRUM Server. You can also filter alarm forwarding, which is based on alarm probable cause IDs. The Remote Operations Connector only forwards alarms that are configured for sending to the Central SPECTRUM Server. Any alarm that does not match the alarm forwarding filter is not forwarded.

Follow these steps:

1. Locate the Remote Operations Connector configurations and expand the Alarm Forwarding subview.
   The Alarm Forwarding configuration options display.
2. Click the set link and select Yes, to select which alarm severities to send to the Central SPECTRUM Server. By default, the critical, major, minor, and maintenance alarm severity options are enabled.
   The value of each alarm displays next to the alarm.

   NOTE
   Suppressed, maintenance, and initial conditions on DX NetOps Spectrum models typically do not generate actual alarms. Fewer alarms improve the performance of the SpectroSERVER process during network outages. The forwarding of suppressed, maintenance, or initial alarms by the Remote Operations Server results in the generation of these alarms in the SpectroSERVER. Verify that the alarms have been generated by opening the Alarm Management subview of the VNM model. Examine the Disable Initial Alarms, Disable Suppressed Alarms, and Disable Maintenance Alarms settings. These alarm management attributes must be set to “No” to enable alarms of that severity to be forwarded by the Remote Operations Server.
3. Click Add under the Probable Cause Filter option to specify any alarm probable cause codes to forward to the Central SPECTRUM Server.
   The Add dialog appears.
4. Enter the probable cause code in the Enter PCause ID field and click OK.
   The probable cause code displays in the Probable Cause Filter list.

   NOTE
   If any probable cause codes are listed in the Probable Cause Filter list, then only alarms that match a probable cause ID in that list are forwarded to the Central SPECTRUM Server. All others are not forwarded. If the Probable Cause Filter list is empty, then no filtering of alarms that are based on probable cause code is performed.

Alarm forwarding is configured.

NOTE
For more information about alarms, see the Event Configuration section.

Configure and Connect to the Central SPECTRUM Server

Configure each Remote Operations Connector with the name of the Central SPECTRUM Server host to connect to the Central SPECTRUM Server. The Remote Operations Connector requires information about where to forward alarm and topology data.

NOTE
Alarm forwarding must be configured before you set the primary and backup Central SPECTRUM Server hostnames.

Follow these steps:

1. Locate the Remote Operations Connector configurations and expand the General subview.
   The General configuration options display.
2. Verify that the Remote Operations Server ID is correct. (The Remote Operations Server ID defaults to the local servername). If it is not correct:
   1. Select *Set*
   2. Enter the *Remote Operations Server ID*
   3. Select *Enter*.

   The Remote Operations Server ID displays next to Remote Operations Server ID.
3. (Optional) Click set next to the Backup Central SPECTRUM Server Host Name to enter the hostname of a backup Central SPECTRUM Server, and press Enter.

NOTE
This hostname must not be the fully qualified hostname.

The backup hostname displays next to the Backup Central SPECTRUM Server Host Name.

4. Click set next to the Primary Central SPECTRUM Server Host Name, enter the hostname of the primary Central SPECTRUM Server, and press Enter.

NOTE
This hostname must not be the fully qualified hostname. The server that you designate as the Primary Central SPECTRUM Server can be the MLS. For more information, see Selecting a Main Location Server. Ensure that the Remote Operations Server ID matches the Remote Operations Server local machine name.

The primary hostname displays next to the Primary Central SPECTRUM Server Host Name.

WARNING
Initial topology and alarm data are immediately sent to the Central SPECTRUM Server once the primary Central SPECTRUM Server hostname is set. Set the primary Central SPECTRUM Server hostname last.

Subsequent topology updates occur only when contact with the Central SPECTRUM Server is reestablished after a disconnect, or with an ondemand topology update.

Fault Tolerance

If primary and backup Central SPECTRUM Server host names are defined, the Remote Operations Server automatically starts forwarding topology and alarm information to the backup server if the connection to the primary server goes down. When this switchover happens, a full topology and alarm update is sent to the backup server and normal alarm forwarding begins.

NOTE
Be sure that the primary and backup Central SPECTRUM Servers are already in a fault tolerant environment before defining the host names for these servers.

Once the connection to the primary server is restored, the Remote Operations Server stops sending data to the backup server, a full topology, and alarm update is sent to the primary server, and normal alarm forwarding resumes.

NOTE
Manually clear the residual alarms. Delete any unnecessary RemoteOperationsManagedElement models that exist in the backup server after the Remote Operations Server stops forwarding data to it.

Using Remote Operations Suite

This chapter assumes that you are the administrator at your central site, using the Central SPECTRUM Server.

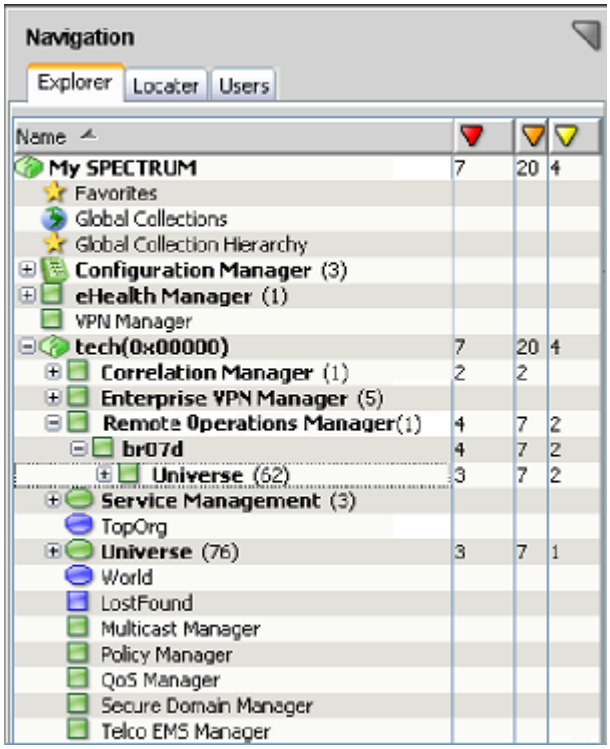
View Remote Operations Server Information

View information that is related to the connected Remote Operations Servers and subcomponents, from the Remote Operations Manager component on the Central SPECTRUM Server.

Follow these steps:

1. Expand Remote Operations Manager in the Explorer tab of the Navigation panel.
   Remote Operations Servers currently connected display.
2. Expand each Remote Operations Server to reveal its subcomponent hierarchy.

Information and alarm data associated with the selected Remote Operations Server or Remote Operations subcomponent display in the Contents and Component Detail panels.



Update Remote Operations Server Topology Data

The Remote Operations Connector does not keep the Central SPECTRUM Server up to date with topology and modeling changes that occur on the Remote Operations Server. The Remote Operations Connector forwards topology data to the Central SPECTRUM Server only when an initial connection to the Central SPECTRUM Server is made and when the connection is reestablished after a disconnect.

You can also manually update the Remote Operations Server topology data at any time, and instruct the Remote Operations Server or Remote Operations Servers, to send the topology data to the Central SPECTRUM Server.

> **NOTE**
> Each time the topology data is updated, if the *Model_Name* of any models in the Remote Operations Server has changed, the corresponding lightweight Remote Operations models are also updated. Each time the topology data is updated, the alarm status for all newly-forwarded and existing managed elements is also updated.

To update topology data for all connected Remote Operations Servers

1. Select the Remote Operations Manager in the Navigation panel.
   Information displays in the Contents panel.
2. Select the Information tab in the Component Detail panel and expand the Configuration subview.
   The Update Remote Operations Topology button is displayed.
3. Click Update Remote Operations Topology.
   The Start Remote Operations Topology Update dialog opens.
4. Click Yes to start the Remote Operations Topology Update.
   The Start Remote Operations Topology Update dialog displays if the Remote Operations topology update was successful.
5. Click OK.
   The topology data is updated for all connected Remote Operations Servers.

To update topology data for a single Remote Operations Server

1. Expand the Remote Operations Manager node in the Navigation panel.
   All connected Remote Operations Servers display.
2. Select the single Remote Operations Server for which the topology data should be updated.
   Information displays in the Contents panel.
3. Select the Information tab in the Component Detail panel and expand the Configuration subview.
   The Update Remote Operations Topology button is displayed.
4. Click Update Remote Operations Topology.
   The Start Remote Operations Topology Update dialog opens.
5. Click Yes to start the Remote Operations Topology Update.
   The Start Remote Operations Topology Update dialog displays if the Remote Operations topology update was successful.
6. Click OK.
   The topology data is updated for the selected Remote Operations Server.

   NOTE
   You can also update the topology data for a single Remote Operations Server from the computer where that Remote Operations Server resides.

To update topology data for a single Remote Operations Server on the machine where the Remote Operations Server resides

1. Select the Remote Operations Server for which the topology data should be updated in the Navigation panel.
   Information displays in the Contents panel.
2. Select the Information tab in the Component Detail panel and expand the Configuration subview.
   The Update Remote Operations Topology button is displayed.
3. Click Update Remote Operations Topology.
   The Start Remote Operations Topology Update dialog opens.
4. Click Yes to start the Remote Operations Topology Update.
   The Start Remote Operations Topology Update dialog displays if the Remote Operations topology update was successful.
5. Click OK.
   The topology data is updated for the selected Remote Operations Server.

RemoteOperationsManagedElement Model Type

The Central SPECTRUM Server can discover any or all managed network elements in the Universe hierarchy of the Remote Operations Server.

The Central SPECTRUM Server stores all of the Remote Operations-related data in the modeling database. It uses the model type RemoteOperationsManagedElement to model all network entities that are managed by a Remote Operations Server. All subcomponent relationships between two RemoteOperationsManagedElement models (for example, a "port" being a subcomponent of a "device") are modeled in the DX NetOps Spectrum database as associations, using the RemoteOperationsContains relation.

The RemoteOperationsManagedElement model type contains the following attributes:

- **RemoteOperationsServerId**
  Identifies the Remote Operations Server where the RemoteOperationsManagedElement is stored.
- **RemoteOperationsServerModelHandle**
  Identifies the RemoteOperationsManagedElement in its associated Remote Operations Server, using a unique model handle.
- **RemoteOperationsServerModelName**

Identifies the model name given to the element inside the Remote Operations Server.

- **RemoteOperationsServerModelType**
  Identifies the model type used to model the element inside the Remote Operations Server.

You can copy and paste a RemoteOperationsManagedElement model anywhere in DX NetOps Spectrum's topology views (Universe, World, or TopOrg), to arrange your network containment.

Search for Models

Search for models if you do not know the exact hierarchy where a particular model exists.

Follow these steps:

1. Select the Locater tab in the Navigation panel on the Central SPECTRUM Server and expand Remote Operations.
2. Double-click All Remote Operations Managed Elements.
   All of the models for the managed elements display in the Contents panel.
3. Double-click All Remote Operations Servers.
   All of the server models display in the Contents panel.
4. To search for managed elements that are based on the model name of the managed element or by the server name on which it resides, expand Remote Operations Managed Elements By and do the following actions:
   – Double-click Remote Operations Model Name to search by the name of the model.
   – Double-click Remote Operations Server to search by the name of the Remote Operations Server on which it resides.

   The Search dialog opens.
5. Enter the model name for the managed element, and click OK.
6. Enter the server name for the managed element, and click OK.
   The managed element displays in the Contents panel.

> **NOTE**
> Click Landscapes to specify the landscapes from which OneClick exports data in a distributed environment. Move the landscapes whose data you want to export data into Show Landscapes. Move any landscapes whose data you do not want to export into Hide Landscapes in the Landscape Filter.

Device Management

Remote Operations Suite limits the number of models that can be created in a DX NetOps Spectrum database. Specifically, the Remote Operations Server is limited to managing 100 device models. As a result the Remote Operations Server is limited to managing 100 network devices. The 100 device limit includes device models that have an IP address and are derived from the device model type. Devices that are modeled as hosts or as pingables are not included in this limit.

You can determine whether the Remote Operations Server can manage all of your device models using some simple arithmetic.

Example: 590 Devices

This example shows how the Remote Operations Server can manage 590 devices:

A customer network comprises 10 routers, 60 switches, 10 firewalls, 10 devices that DX NetOps Spectrum has automatically modeled using the Generic SNMP Device (GnSNMPDev) model type, approximately 500 servers and workstations, and many other devices DX NetOps Spectrum has modeled using the “pingable” device model type.

In this example, the models that would be included in the device limit would total only 90. The 10 routers, 60 switches, 10 firewalls, and the 10 devices that are modeled with GnSNMPDev would be included. The 500 workstations and servers and the other devices that are modeled as pingables do not affect the 100 device limit.

Fault Isolation

If the Central SPECTRUM Server loses contact with a Remote Operations Server, it generates a red alarm on the corresponding Remote Operations Server model. All RemoteOperationsManagedElement models that are associated with the disconnected Remote Operations Server model display a suppressed (gray) condition, indicating that their status is unknown.

The Central SPECTRUM Server cannot manage the RemoteOperationsManagedElement models in a suppressed state. When the Central SPECTRUM Server reestablishes contact with the Remote Operations Server, the red alarm is cleared. The state of all corresponding RemoteOperationsManagedElement models is reevaluated.

By default, the Central SPECTRUM Server waits for 30 seconds for a keepalive signal, or heartbeat, from the Remote Operations Server before generating the red alarm. You can change this default setting using the following parameter in the ".vnmrc" file of the Central SPECTRUM Server:

- **ros_heartbeat_timeout**
  Specifies the maximum number of seconds the Remote Operations Manager waits for a heartbeat from the Remote Operations Server, before a contact lost alarm is raised. This parameter must be configured on Central SPECTRUM Server.
  The heartbeat parameter has the following format:
  ros_heartbeat_timeout=# of seconds
  Default: 30 seconds

  NOTE
  When contact with a Remote Operations Server is lost, the previously existing alarms on all RemoteOperationsManagedElement models still appear in the OneClick Alarms tab.

Remote Operations Suite Alarms

When the Remote Operations Server forwards an alarm to the Central SPECTRUM Server, it includes two important pieces of information: the title of the alarm, and the actual text of the event that prompted alarm generation.

The restrictions of tactical deployments and geography can mean that the version of DX NetOps Spectrum that is installed and running on the Remote Operations Server is slightly different from the version that is running on the Central SPECTRUM Server. In such a case, the two servers do not have the same fault detection intelligence. Therefore, the Central SPECTRUM Server lacks the alarm support files. The alarm support files are required to generate the identical alarm type as was generated on the Remote Operations Server.

To ensure the Central SPECTRUM Server handles all alarm types that the Remote Operations Server sends, the Central SPECTRUM Server uses a set of Remote Operations-specific alarm types. One alarm type for each alarm severity. When an alarm is forwarded from the Remote Operations Server, the Central SPECTRUM Server generates the appropriate Remote Operations-specific alarm, which is based on the severity of the forwarded alarm.

This generic alarm is displayed in the OneClick Alarms tab like all other alarms. Instead of displaying a generic alarm title, the Alarm Details tab displays the title of the alarm that was forwarded from the Remote Operations Server. In addition, the Central SPECTRUM Server appends the actual event text that is forwarded from the Remote Operations Server to the event that generates the forwarded alarm. This event text is also displayed in the Alarm Details tab, like all other event text.

Example: Alarm Title

This example shows an alarm title from the Remote Operations Server:

```
DEVICE HAS STOPPED RESPONDING TO POLLS
```

Example: Probable Cause Text

This example shows the body of the Remote Operations critical alarm probable cause:

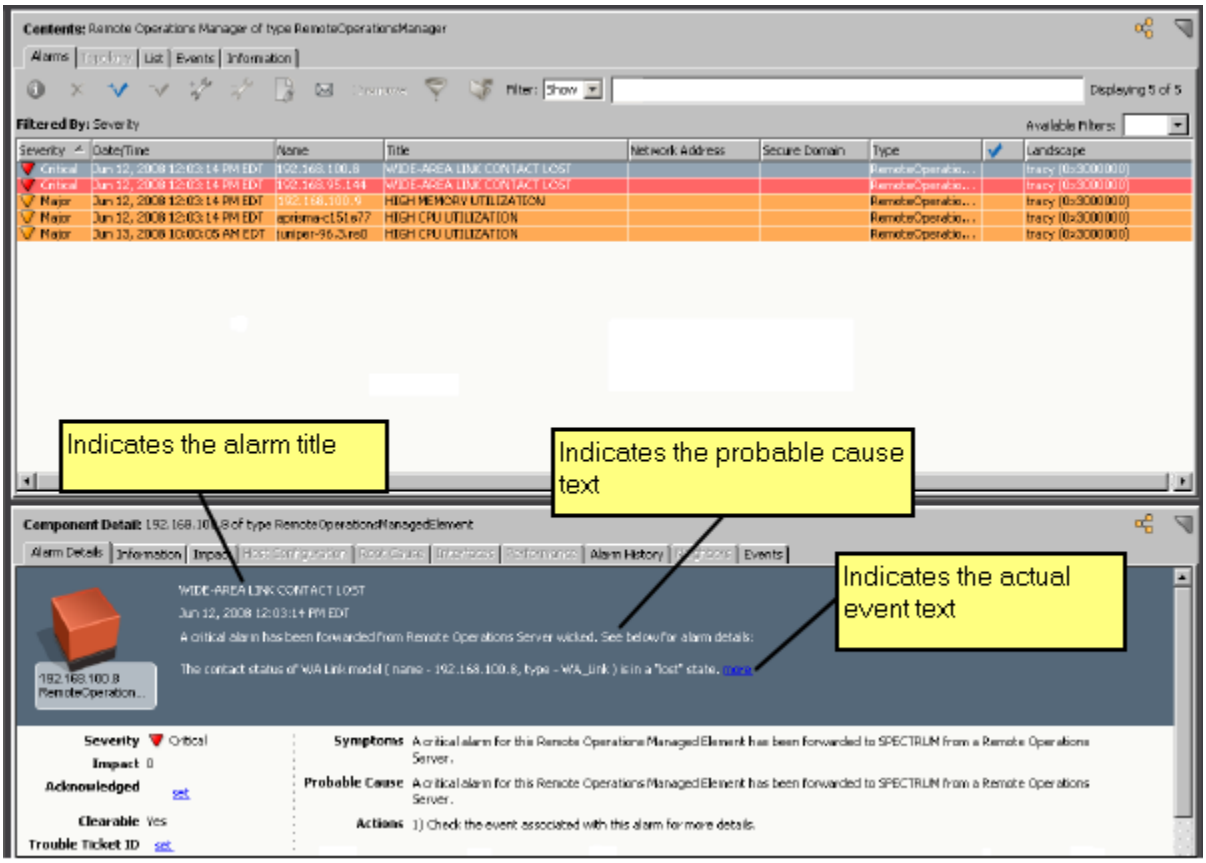
```
CRITICAL ALARM ON REMOTE OPERATIONS-MANAGED ELEMENT
SYMPTOMS:
A critical alarm for this Remote Operations-managed element has been forwarded to SPECTRUM from a SPECTRUM
 Remote Operations Server.
PROBABLE CAUSES:
A critical alarm for this Remote Operations-managed element has been forwarded to SPECTRUM from a SPECTRUM
 Remote Operations Server.
RECOMMENDED ACTIONS:
1) Check the event associated with this alarm for more details.
```

Example: Actual Event Text

In the following example, the event that is associated with the alarm displays the actual event text that was generated in the Remote Operations Server:

```
{d "%w- %d %m-, %Y - %T"} - A critical alarm has been forwarded to SPECTRUM from SPECTRUM Remote Operations
 Server TEAM_A.  See below for alarm details:
{d "%w- %d %m-, %Y - %T"} - Device Router1 of type Rtr_Cisco has stopped responding to polls and/or external
 requests.  An alarm will be generated.   (event [{e}])
(event [{e}])
```

The following image shows each part of the alarm as it appears in the OneClick Console:



NOTE
For more information about alarms, see the Event Configuration section.

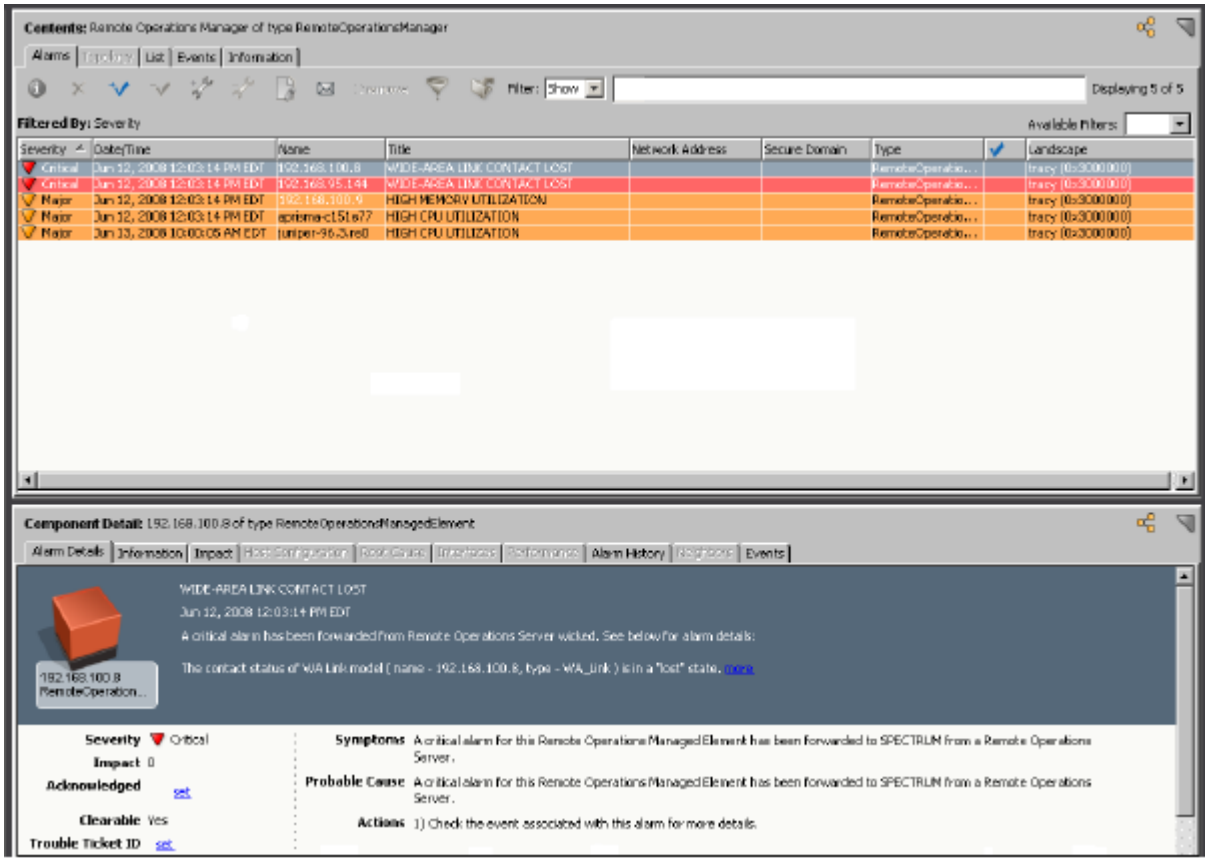
View Remote Operations-Related Alarms

You can view the Remote Operations-related alarms on all models in the Remote Operations hierarchy.

WARNING
Configure the Remote Operations Connector on each Remote Operations Server to forward alarms to the Central SPECTRUM Server. For more information about forwarding alarms to the Central SPECTRUM Server, see Configure Alarm Forwarding.

To view only the Remote Operations-related alarms on the Central SPECTRUM Server, select the Remote Operations Manager model in the Explorer tab of the Navigation panel. Click the Alarms tab in the Contents panel.

Information such as alarm symptoms and possible alarm causes displays in the Contents and Component Detail panels for all models in the Remote Operations hierarchy.



NOTE
If the topology data on the Central SPECTRUM Server is not synchronized with the topology data on the Remote Operations Server or subcomponent, the alarm data does not update. Perform an on-demand Topology synchronization to ensure that your topology and alarm data is current. For more information about synchronizing alarm data, see Update Remote Operations Server Topology Data.

Remote Operations Forwarding Events

When an alarm is forwarded to the Central SPECTRUM Server for processing, a Remote Operations Forwarding Event is generated on the corresponding lightweight Remote Operations model (RemoteOperationsManagedElement). The Remote Operations Forwarding Event contains the following varbinds:

- **RemoteOperationsServerId**
  Identifies the Remote Operations Server where the RemoteOperationsManagedElement is stored.
- **RemoteOperationsServerModelHandle**
  Identifies the RemoteOperationsManagedElement in its associated Remote Operations Server, using a unique model handle.
- **RemoteOperationsServerEventCode**
  Identifies the standardized DX NetOps Spectrum event/alarm code for the event.
- **RemoteOperationsServerEventSeverity**
  Identifies the severity of the event/alarm. Possible values include critical, major, minor, maintenance, suppressed, initial, and message.
- **RemoteOperationsServerEventType**
  Identifies the type of event. Possible values include message, alarm_set, and alarm_cleared.
- **RemoteOperationsServerEventCommentData**
  Displays a textual description of the event that caused an alarm to be generated or cleared.
- **RemoteOperationsServerEventTitle**
  Identifies the textual title (abstract) of the alarm.

> **NOTE**
> DX NetOps Spectrum does not log the Remote Operations Forwarding Event. DX NetOps Spectrum applies event rules to the Remote Operations Forwarding Event that determine the type of alarm or message event that is generated and logged. These event rules are stored in the following file: <*$SPECROOT*>/SS/CsVendor/Spectrum_ROS/EventDisp.

Remote Operations Alarming Events

A Remote Operations Forwarding Event is mapped to the Remote Operations Alarming Event by the event rules if the value of the RemoteOperationsServerEventType varbind is equal to "alarm_set". The following types of Remote Operations Alarming Events are defined:

- Remote Operations Critical Alarm Event
- Remote Operations Major Alarm Event
- Remote Operations Minor Alarm Event
- Remote Operations Maintenance Alarm Event
- Remote Operations Suppressed Alarm Event
- Remote Operations Initial Alarm Event

The value of the RemoteOperationsServerEventSeverity varbind determines which Remote Operations Alarming Event is generated. DX NetOps Spectrum logs the event. The event is used to generate the appropriate alarm. All varbind values in the Remote Operations Forwarding Event are copied to the new event.

Remote Operations Clearing Events

A Remote Operations Forwarding Event is mapped to the Remote Operations Clearing Event by the event rules if the value of the RemoteOperationsServerEventType varbind is equal to "alarm_cleared". This event clears any existing alarm on a model that has the same RemoteOperationsServerEventCode and RemoteOperationsServerEventSeverity values. All varbind values in the Remote Operations Forwarding Event are copied to the new event.

> **NOTE**
> When clearing events, the RemoteOperationsServerEventCode that is sent is not the actual event code that was generated in the Remote Operations Server. Rather, the code for the event that generated the original alarm is sent.

Remote Operations Message Events

A Remote Operations Forwarding Event is mapped to the Remote Operations Message Event by the event rules if the RemoteOperationsServerEventType varbind is equal to "message". This event is for informational purposes only and is not used to generate an alarm. However, this event is logged to DX NetOps Spectrum. All varbind values in the Remote Operations Forwarding Event are copied to the new event.

To create your own event and alarm types, modify the EventDisp file, which is located in <*$SPECROOT*>/SS/CsVendor/Spectrum_ROS.

Secure Domain Manager (SDM)

DX NetOps Spectrum Secure Domain Manager

Secure Domain Manager is a DX NetOps Spectrum network management solution that allows users to manage devices in secure networks. You can manage devices without deploying a local SpectroSERVER. Secure Domain Manager lets you manage your secure domains by securely tunneling SNMP and ICMP traffic through a secure connection. Only a single port is opened on the firewall, which allows the extended manageability without impacting security policies in place. This solution is transparent to end-users and client applications, eliminating the need to perform more administrative tasks.

Challenges to Managing Highly Secure Networks

Data networks are more secure than ever nowadays. The challenges that are involved in managing devices and applications in secure networks are correspondingly greater. These challenges include:

- Managing network elements in overlapping (or private) IP domains (NAT environments)
- Managing network elements behind firewalls that are configured to block SNMP and ICMP traffic
- Managing network elements across insecure network domains

The Secure Domain Manager product provides a unique solution to these management challenges.

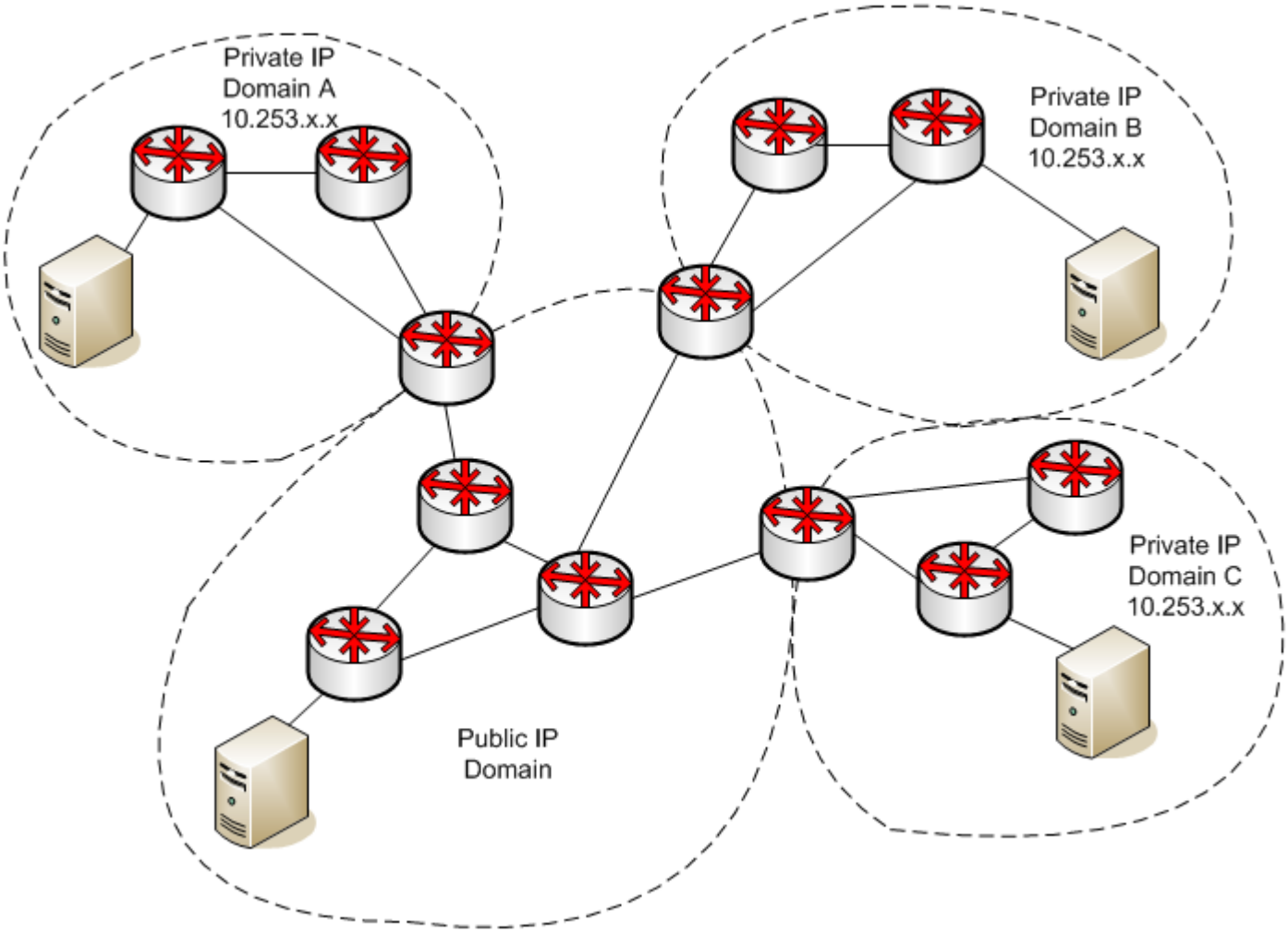
Benefits of Secure Domain Manager

The Secure Domain Manager solution enhances the existing management capabilities included with DX NetOps Spectrum in the following ways:

- Lets DX NetOps Spectrum communicate with all SNMP-compliant devices: SNMPv1, SNMPv2, and SNMPv3.
- Lets DX NetOps Spectrum communicate with devices located behind firewalls that block SNMP and ICMP traffic.
- Simplifies firewall configuration. A single "hole" is opened for traffic passing between two well-known hosts on a well-known port.
- Lets DX NetOps Spectrum pass SNMP and ICMP traffic securely through insecure networks.
- Lets DX NetOps Spectrum manage devices in overlapping IP domains (NAT environments) using a single SpectroSERVER.
- Provides enhanced Discovery capabilities to discover and model devices in secure environments, one IP address space at a time.

Overlapping IP Domains

The following diagram shows a NAT network that contains a public IP domain and three private IP domains that contain the same IP subnets.



Domains can represent the managed network of a company, a newly acquired division of a large enterprise, or a managed wireless hot spot in an airport terminal.

The following types of DX NetOps Spectrum customers face the overlapping IP challenge:

- **Managed service providers (MSPs)**
  MSPs use DX NetOps Spectrum to manage the networks of other organizations. The customers that MSPs manage invariably use IP ranges typically used for private IPs, 10.x.x.x or 172.16.x.x, for example. Therefore, the MSP must address the challenge of managing duplicate or overlapping IP addresses. In the past, this challenge was addressed by dedicating a DX NetOps Spectrum management server (the SpectroSERVER) to each customer that was using the same IP address space.
  This posed two issues. The first involved cost. A dedicated management server was necessary for each customer, regardless of the size of the managed environment and the number of overlapping IP addresses it used. The second issue involved administration. The MSP was burdened with maintaining more management systems. MSPs required less expensive and efficient alternatives to dedicated management systems, especially when the number of elements with overlapping IP addresses was small and did not warrant the expense of a dedicated management server.
- **Hotspot (Wi-Fi) access providers**
  Hotspot access providers provide Wi-Fi access in locations such as airport terminals, airport lounges, hotel rooms, and coffee shops. For each location, the same private IP address space is issued. This approach simplifies configuration, installation, and administration. A provider may have hundreds or thousands of hotspots. To deploy a new hotspot

quickly, each set of equipment that establishes the hotspot in a property is configured identically, including the IP address space. Once the hotspot is up and running, the challenge becomes managing it proactively to sustain an optimal level of service.

- **Enterprise managers**
  In an organization merger or acquisition scenario, an enterprise management staff must typically combine two entirely different and separately constructed IP networks, in many instances resulting in multiple overlapping IP addresses. In this scenario, the new IT organization must now deal with managing the combined network, especially the management of a network with the same IP address spaces. One solution to this challenge is to reassign IPs to every IP entity so there is no duplication of IP addresses. That solution involves a huge undertaking that presents many challenges.

Secure Domain Manager lets these customers overcome the challenge of managing overlapping IP domains in the following ways:

- Allows MSPs to deploy only a single, lightweight agent process on a host machine in each customer's remote network, thus removing the need to deploy and administer a full DX NetOps Spectrum installation.
- Allows the Hotspot access providers and large enterprises keep the overlapping private IP domains intact and manage the networks by using a lightweight agent process.

Firewalls Blocking SNMP and ICMP Traffic

Firewalls provide the security crucial to many network environments. Some challenges exist in managing a network behind a firewall. First, network administrators often configure firewalls to block SNMP and ICMP traffic, which provide the visibility into their network infrastructure, to unauthorized sources. Second, the configuration that is required to manage network elements through a highly secure firewall is complicated. Because, all the hosts and ports that are involved are identified and opened on the firewall to enable full management capabilities.

Secure Domain Manager allows network managers to overcome the challenge of managing networks through secure firewalls in the following ways:

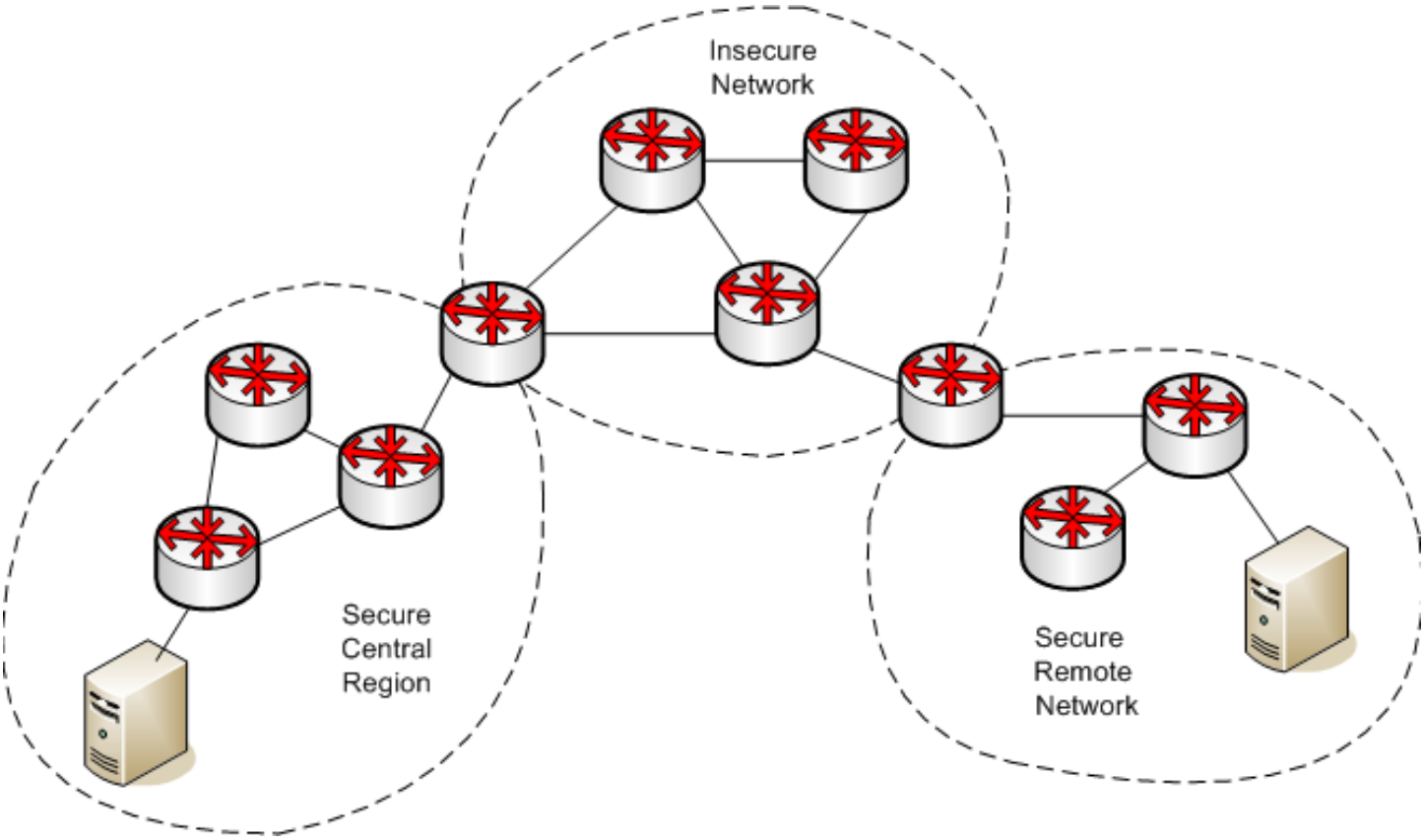
- Encoding UDP-based SNMP and the ICMP packets into a TCP/IP based protocol to overcome the restriction of the firewall on SNMP and ICMP traffic.
- Simplifying the configuration of the firewall by opening a single port that allows SNMP and ICMP traffic to flow between two well-defined hosts, on a well-defined port.

SNMP Traffic Passing Across Insecure Networks

SNMPv1 and SNMPv2 are insecure protocols: their data is not encrypted and can be viewed using a protocol analyzer. Therefore, it is undesirable to send this traffic across insecure networks. Allowing the SNMP traffic to cross insecure networks to reach a network you want to manage becomes challenging.

The following diagram shows a network management system on the host computer in the "Secure Central Region" that manages devices in a "Secure Remote Network." Management traffic must flow through the "Insecure Network" region. Network managers want to avoid exposing the data inside insecure protocol packets such as SNMPv1 and SNMPv2 in this portion of the network.

SNMP Across an Insecure Network



Secure Domain Manager lets network managers encrypt all management traffic that passes between the SpectroSERVER host and the host in the remote managed network. This solution securely passes insecure SNMP traffic across insecure networks. Data security is maintained when the traffic traverses the intermediate insecure networks.

How Secure Domain Manager Works

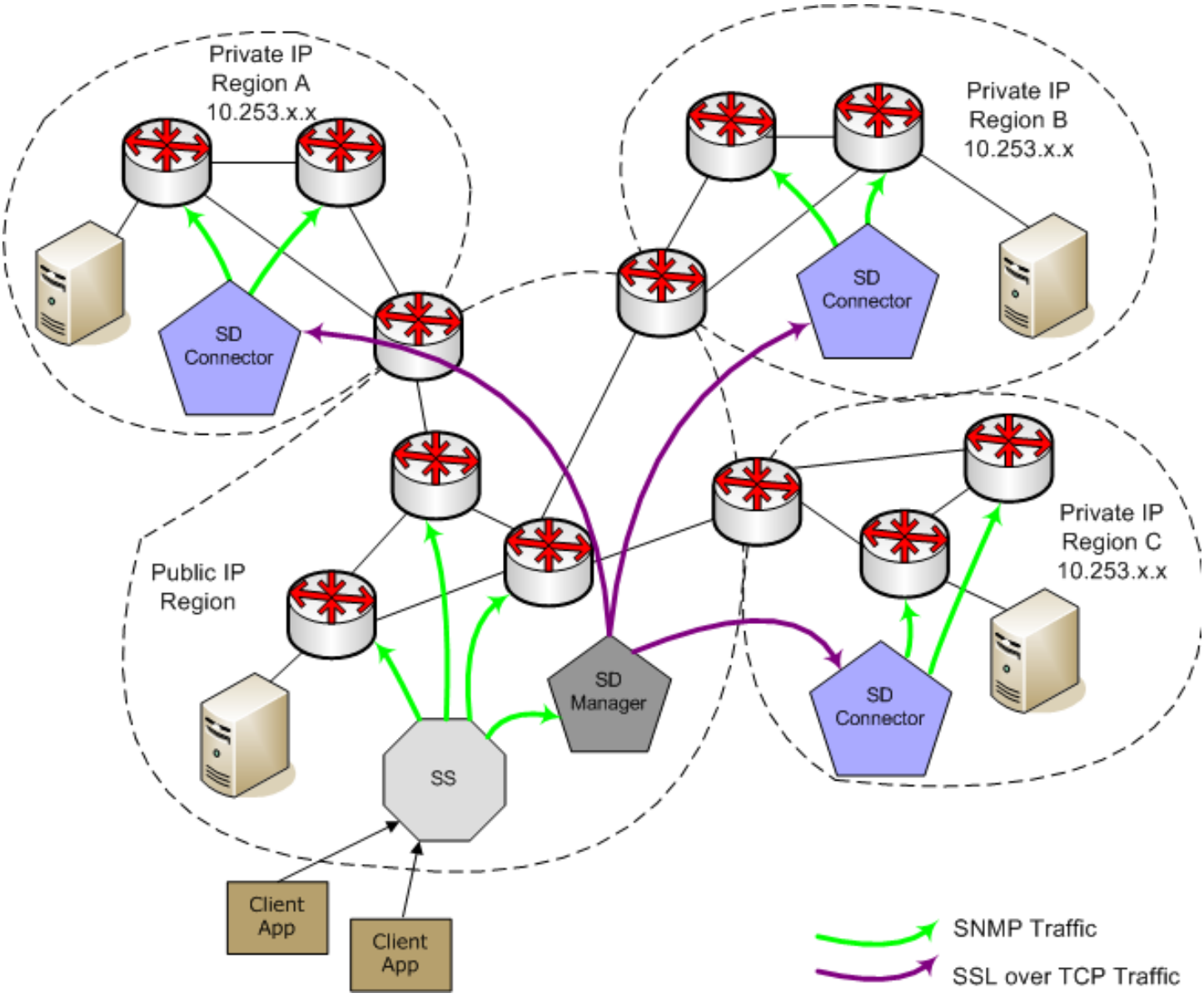
Secure Domain Manager supports SNMPv1, SNMPv2, and SNMPv3 communication. It consists of two different processes, SDManager (SDM) and SDConnector (SDC):

- **SDManager**
  SDManager is a server messaging library that is loaded by the SpectroSERVER.
- **SDConnector**
  SDConnector is a remote process responsible for communicating with the SDManager on the SpectroSERVER. It runs on a host machine that is located in a remote private network and it is capable of forwarding SNMP and ICMP messages on behalf of the SpectroSERVER (which would ordinarily be deployed in the private IP region) so that it can manage devices in the private network. SDConnector is configured using a configuration file (sdc.config) which can contain both primary and backup SpectroSERVER information. It is a part of the Secure Domain Manager solution.

The following diagram shows how these processes are deployed in a secure network environment.

> **NOTE**
>
> We recommend that you have a one to one mapping between SDC and SDM. In case of multiple SDMs connected to a single SDC, the SDC forwards traps to all the connected SDMs.

NAT Network Environment Using Secure Domain Manager



NOTE
Devices that are located in the same region as the SpectroSERVER are managed using SNMP, but without using Secure Domain Manager.

When the SpectroSERVER located in the public IP region must communicate with a device that is located in a remote secure region, the SpectroSERVER sends the request to the SDManager. The SDManager converts the SNMP data into a proprietary format and sends the data to the SDConnector located in the same region as the device. If the SDManager and SDConnector have been configured to run with SSL, the data is encrypted and sent through a secure tunnel to the SDConnector using SSL over TCP. When the SDConnector receives the data, it converts the data back to SNMP and sends a request to the appropriate device.

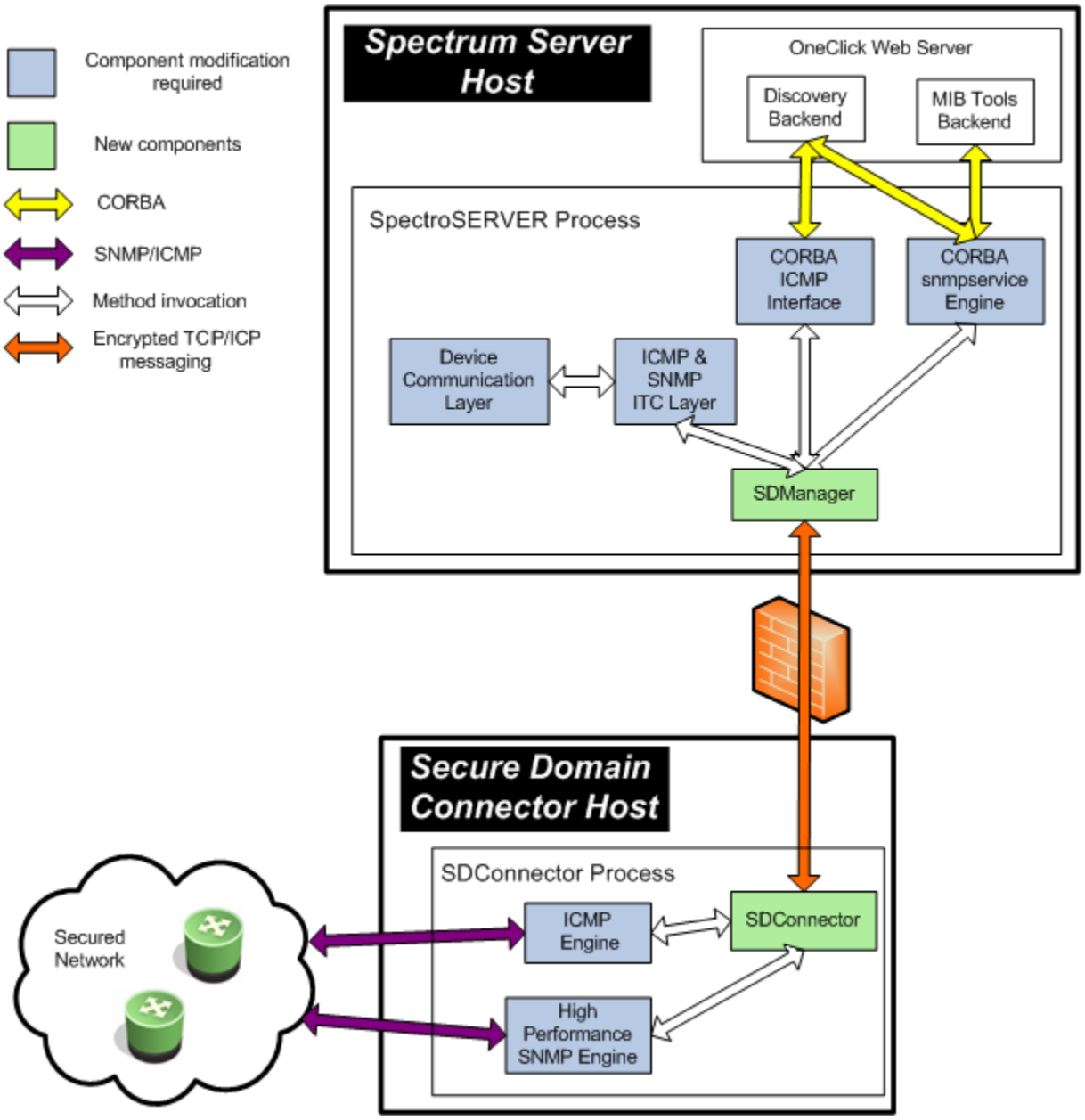
If the firewalls are deployed, it works the same way. The Network administrators must create a "hole" in each firewall that is dedicated to two well-known hosts. Devices that are located in regions that traverse more than one firewall are also manageable using this solution. To enable this communication, open a port on each firewall. The port must be a well-known port, which lets a pair of well-known hosts in adjacent regions to communicate using TCP.

When deploying Secure Domain Manager to manage overlapping IP domains, each SDConnector host machine must have a unique public IP address. The host must be able to communicate with all devices with which the SDConnector must communicate, including the SpectroSERVER host machine and all devices within the single private IP domain it manages. A likely candidate for this SDConnector host would be a machine behind the NAT that has a unique IP address that is statically assigned to it from the NAT. The SpectroSERVER uses the unique IP address of the SDConnector host machine as the additional discriminator to uniquely identify multiple devices that have the same private IP address.

NOTE
Certain DX NetOps Spectrum products such as Network Configuration Manager (NCM) and IP services management applications (including Multicast Manager and Enterprise VPN Manager) cannot manage overlapping IP addresses. However, you can still use Secure Domain Manager with these applications if you model their devices on the SDConnector rather than on the SpectroSERVER. Such configurations can have multiple SDConnectors deployed for each SpectroSERVER as long as the SDConnectors are not managing devices that are configured with overlapping IP addresses. Using this approach, you can still model devices on the local SpectroSERVER too, but only if they have not been configured with IP addresses that overlap with the IP addresses configured on the devices being managed through the SDConnectors.

Secure Domain Manager Architecture

The following diagram illustrates how the Secure Domain Manager operates:



Benefits of Secure Domain Manager

The Secure Domain Manager solution enhances the existing management capabilities included with DX NetOps Spectrum in the following ways:

- Lets DX NetOps Spectrum communicate with all SNMP-compliant devices: SNMPv1, SNMPv2, and SNMPv3.
- Lets DX NetOps Spectrum communicate with devices located behind firewalls that block SNMP and ICMP traffic.
- Simplifies firewall configuration. A single "hole" is opened for traffic passing between two well-known hosts on a well-known port.
- Lets DX NetOps Spectrum pass SNMP and ICMP traffic securely through insecure networks.
- Lets DX NetOps Spectrum manage devices in overlapping IP domains (NAT environments) using a single SpectroSERVER.
- Provides enhanced Discovery capabilities to discover and model devices in secure environments, one IP address space at a time.

Installing and Configuring Secure Domain Manager Processes

This section describes how to install and configure the Secure Domain Manager solution, a process that involves installing and configuring the SDConnector and SDManager.

How to Set Up Secure Domain Manager Processes

Setting up Secure Domain Manager involves installing and configuring Secure Domain Manager processes and then setting them up on the SpectroSERVER using OneClick.

Install and Configure Processes

Installing and configuring Secure Domain Manager processes requires the following steps:

1. Install the SDConnector process on a designated host.

   NOTE
   SDManager is installed when you install the core DX NetOps Spectrum product; however, it is only active if it is included in the bundle your company purchased.

   NOTE

   Information! During the install of the Secure Domain Manager, Microsoft Visual C++ 2013 Redistributable Update is also installed automatically as it is a required program to run SDConnector.
2. (Optional) Create and deploy SSL certificates for SSL encryption.
3. Set parameters in the configuration file for the SDConnector on the SDConnector host.
4. Set parameters in the configuration file for the SDManager on the SpectroSERVER.

Set Up Secure Domain Manager on SpectroSERVER

After you have installed and configured the SDConnector and SDManager processes, set up Secure Domain Manager on the SpectroSERVER host using OneClick. This setup involves the following steps:

1. Import the SDManager configuration file.
2. Model an SDConnector host.
3. Model the devices in secure domains that you want to manage.

Hardware Recommendations

Follow these recommendations to achieve an optimal Secure Domain Manager performance:

- To maintain an optimal SpectroSERVER modeling capacity, the SpectroSERVER/SDManager installation computer must have two CPUs: one dedicated to the SpectroSERVER and the other dedicated to servicing SDManager

functions. If SDManager and SpectroSERVER are required to share a single processor to manage network elements, SpectroSERVER modeling capacity reduces by 40%.

- We recommend that you have a host computer that is dedicated solely to running each SDConnector process you deploy. The SDConnector installation system requirements are the same as those for SpectroSERVER-only installations. Except for the special requirements of multiple disk configuration.

  NOTE
  For more information about installation requirements, see the Installing DX NetOps Spectrum section.
- An SDConnector has only one SDManager connected to it. Two SDManagers from fault-tolerant SpectroSERVERs can be connected to a single SDConnector, if that is a requirement for your setup. See Setting Up Processes in a Fault-Tolerant Environment for more information.

About SDConnector CPU and Memory Usage

The SDConnector uses half the CPU capacity that the SpectroSERVER uses to manage devices. If the SpectroSERVER computer uses 50 percent total CPU and all devices are managed using SDManager, the SDConnector uses approximately 25 percent of CPU capacity, on an equally powerful system. The major difference is that the SDConnector does not use much memory. If it is devoted as an SDConnector only, 512 MB of RAM suffices, although more RAM would be better.

Install the SDConnector Process

Before using Secure Domain Manager capabilities to manage devices and applications in secure networks with DX NetOps Spectrum, install a single SDConnector process on a host computer in the secure network. You must be an administrative user on your Windows system, or the root user on Linux systems when you install the SDConnector. To configure multiple SDC processes on a single server, refer to Configure Multiple SDC (Secure Domain Connector) processes on a single host computer.

NOTE
As a best practice before upgrading the SDConnector process on any platform, stop and if required kill the process. Stopping or killing the process ensures that the process runs properly after an upgrade.

To install SDConnector

1. On the nonSpectroSERVER host machine where you want to run SDConnector, launch the appropriate DX NetOps Spectrum installer for the platform.

   NOTE
   Only the SDConnector for the same operating environment as your SpectroSERVER is available for installation. If you have to install SDConnectors for other operating environments, contact CA Support. For information about launching the installer, see the Install DX NetOps Spectrum section.

   The Install dialog opens.
2. Select 'Install CA Secure Domain Connector.'
   The Introduction dialog opens.
3. Click Next to proceed.
   The License Agreement dialog opens.
4. Scroll through and read the license agreement, accept the agreement, and click Next.
   The Destination Location dialog opens.
5. Click Next to install the SDConnector in the default directory. The default directory is C:\Program Files\CA \SDMConnector on Windows and /usr/SDMConnector on Linux.
   To install the SDConnector in a location other than the default folder, click Choose, select a folder, and click Next. The Choose button only appears for a local installation (not for a nonlocal, remote installation).

NOTE
You cannot install the SDConnector into a directory that contains a space in the name.The Pre-Installation Summary dialog opens.

6. Click Install.
   The Installing SPECTRUM_SDM_Connector dialog opens. After the SDConnector is installed, the status changes to Install Complete and the Done button is enabled.
7. Click Done.
   The dialog closes.
8. Click Close on the initial Install dialog.
   SDConnector is installed on this host computer. The SDConnector is installed as a service, and starts automatically every time your system is restarted.

NOTE
You can also check the installation log located in the directory where SDConnector was installed to verify that the installation completed successfully.

Configure Multiple SDC (Secure Domain Connector) processes on a Single Server

Prerequisites for installing Secure Domain Connector (SDC)

- System requirements for SpectroSERVER and OneClick server
- Hardware recommendations for Secure Domain Manager (SDM) -Secure Domain Connector (SDC)

Configure multiple SDCs in a single machine (For Linux):

Follow these steps:

1. Extract the SDC_Install_<platform>.tgz file installed from the SpectroSERVER-side installation.
2. Stop Secure Domain Connector Service if already running on the linux server chosen for this setup.
3. Copy the extracted sdmc directory to SDConnector machine.
4. As `root` user, install SDC by running 'install.bin'.
5. Go to /opt/CA and make copies of the entire SDMConnector folder.

NOTE

Copy the SDMConnector folder as many times as the number of instances of SDMconnectors you would like to have.

6. Go to each SDC home (default /opt/CA/SDMConnector/) and change directory to bin.
7. Edit the sdc.rc file as below for each instance of SDMConnector:

```
icmp_listen_ip= <IP address for the instance>

snmp_comm_ip= <IP address for the instance>

snmp_comm_port= <Port for communication>

snmp_trap_ip= <IP address for the instance>
```

```
snmp_trap_port= <Port for SNMP traps>

snmp_trap_port_enabled= yes


snmp_trap_max_time_to_live_sec= 120


max_snmp_traps_per_sec= 300


snmpv3_default_auth_protocol= md5


snmpv3_default_priv_protocol= des


snmpv3_engine_id= <Leave this property blank>
```

8. Edit the **sdc.config** file, as follows:

```
-accept <IP address for the SDM>

-bind <IP address for the SDC instance>
```

9. Start each instance of SDMConnector using the command "./SdmConnectorService.exe --start".
10. Navigate to *<SPECROOT>*/SDM directory on the SpectroSERVER host machine and using a text editor, edit the **sdm.config** file to add a remoteconnect entry for each instance of SDMConnector as below:

```
-remoteconnect <IP address for the instance 1> -remoteconnect <IP address for the
 instance 2>
```

11. Import the SDM configuration from Oneclick.

You now have as many Secure Domain Connectors as you have configured on this server. All the SDC instances are bound to the SDM whose IP you have configured.

Configure multiple SDCs in a single machine (For Windows):

Follow these steps:

1. Extract the ***install.exe*** file from the SpectroSERVER-side installation (**Build Folder\sdmc\nt**) or use the **setupnt.exe > Install Secure Domain Connector**.
2. Stop the Secure Domain Connector Service (Navigate to services.msc on the Windows server and stop the service. (If the service is already running on the Windows server chosen for this setup).
3. Copy the extracted sdmc directory (**SDMConnector** folder) to SDConnector machine ***C:\Program Files\CA*** .
4. Navigate to **C:\Program Files\CA** and make copies of the entire SDMConnector folder.

NOTE

Copy the SDMConnector folder as many times as the number of instances of SDM connectors you would like to have.

5. Go to each SDC home (***C:\Program Files\CA\SDMConnector***) and change directory to ***C:\Program Files\CA\SDMConnector\bin*** .
6. In the same directory, edit the **sdc.rc** file for each instance of SDMConnector, as follows:

```
icmp_listen_ip= <IP address for the instance>

snmp_comm_ip= <IP address for the instance>

snmp_comm_port= <Port for communication>

snmp_trap_ip= <IP address for the instance>

snmp_trap_port= <Port for SNMP traps>

snmp_trap_port_enabled= yes

snmp_trap_max_time_to_live_sec= 120

max_snmp_traps_per_sec= 300

snmpv3_default_auth_protocol= md5

snmpv3_default_priv_protocol= des

snmpv3_engine_id= <Leave this property blank>

internal_name= <The name you specify here will be displayed in the Spectrum OneClick
 View>

display_name= <The name you specify here will be SDC connector/service name in the
 server you have installed>
```

7. Navigate to ***C:\Program Files\CA\SDMConnector\bin*** and install SDC using the **SdmConnectorService.exe --install.**
8. Edit the **sdc.config** file as follows:

```
-accept <IP address for the SDM>

-bind <IP address for the SDC instance>
```

9. Start each instance of SDMConnector by performing one of the following:
   – Use the command "./SdmConnectorService.exe --start"/
   – Double-click the SdmConnectorService.exe.
10. Navigate to the Spectroserver machine **<C:\win32app\Spectrum\SDM>**, edit the **sdm.config** file to add a remoteconnect entry for each instance of SDMConnector as specified below:

```
-remoteconnect <IP address for the instance 1> -remoteconnect <IP address for the
 instance 2>
```

11. Import the SDM configuration from Oneclick.
    You now have as many Secure Domain Connectors as you have configured on this server. All the SDC instances are bound to the SDM whose IP you have configured.

SDC TrapX Support

Introduction

TrapX is a Simple Network Management Protocol (SNMP) management application that receives and filters SNMP trap messages and forwards them to other management applications on other hosts and ports. With DX NetOps Spectrum TrapX, you can forward the traps to other management stations. 10.3.2 supports the new TrapX feature, that filters and forwards v1, v2, v3 traps to different products. The Secure Domain Connector (SDC) behaves as the TrapX, filters out v1, v2 and v3 traps, and pushes the filtered traps to the mentioned destination, converting the v2 to v1 and v3 to v1 translations.

DX NetOps Spectrum TrapX feature simplifies trap configuration and management and lets users focus on Information Technology (IT) resources on more strategic activities. CA TrapX is especially useful in environments where multiple management applications must receive trap messages from a diverse set of SNMP-capable devices that can issue messages to only a limited number of SNMP managers. Users can leverage TrapX for the following purposes:

- Filtering traps
- Forwarding traps to other trap receivers
- Forwarding traps to element managers
- Forwarding traps through TCP connections
- Extending fault tolerance for management software

Previously legacy TrapX did not support filtering or translating of v3 traps, but with the 10.3.2 release, the SDC TrapX supports filtering, translating and forwarding of v3 traps. This release also supports the v2 traps with 64-bit counters being translated to v1, which was a limitation with the legacy TrapX feature.

Considerations

Review the following considerations:

- From 10.4.2, you can convert an SDC setup into an SDC-TrapX setup by adding the trapX.config file. You can then add actions in it based on your requirements. You do not need to uninstall the existing environment. Note that the underlying approach has not been changed. For example, users can use their setup either as SDC or SDC-TrapX. They cannot use a combination of both. Similar to previous releases, users need to add -trapx -remoconnect <sdc ip> at SDM to process v3 traps for SDC-TrapX.

  NOTE

  - The trapX.config file should not be empty. It should have filters. The reason is that if it is empty (that is, no filters are added), then it will act as SDC and not SDC-TrapX.
  - If you have an SDC TrapX setup, you cannot convert that setup into an SDC setup by removing the trapX.config file because the NCM services will not be retrieved in this case.
  - You cannot convert an SDC High Availability (HA) setup to SDC TrapX HA setup by adding the trapx.config file. This functionality is not supported currently.
- To listen to the traps on the TCP SNMP port, you need to add the following line to the trapX.config file:
  `listen_for_tcp_traps on`
  An example is as follows:
  ```
  listen_for_tcp_traps on 161
  listen_for_tcp_traps off 161
  ```
  NOTE
  From 10.4.2, it is disabled (off) by default.
- Previously, multiple v3 profiles with the same username and unmanaged traps with that username were not processed by SpectroSERVER. With this release, multiple v3 profiles with the same username and unmanaged v3 traps with that username are processed by SpectroSERVER if the exact v3 local profile is present.
- If SDC is installed as TrapX then it does not function as a regular SDC, and does not model any device on a particular SDC-TrapX.

Supported Operating System

The SDC TrapX supports the following operating systems:

NOTE
DX NetOps Spectrum 10.4.2 has not been validated on Windows Server 2012. However, Broadcom will support any DX NetOps Spectrum product issues, if found. We reserve the right to have you upgrade to Windows Server 2016 (or later) if deemed necessary.

- Microsoft Windows Server 2012 R2 Standard Edition on a 64-bit processor
- Microsoft Windows Server 2016 Standard Edition on a 64-bit processor
- Microsoft Windows Server 2019 on a 64-bit processor
- Red Hat Enterprise Linux 7.x on a 64-bit processor
- Red Hat Enterprise Linux 8.0/8.1 on a 64-bit processor

Traps Management in DX NetOps Spectrum SDC TrapX

The following illustration shows how the DX NetOps Spectrum SDC TrapX can filter and forward traps to various devices.

Enabling SDC Support for TrapX

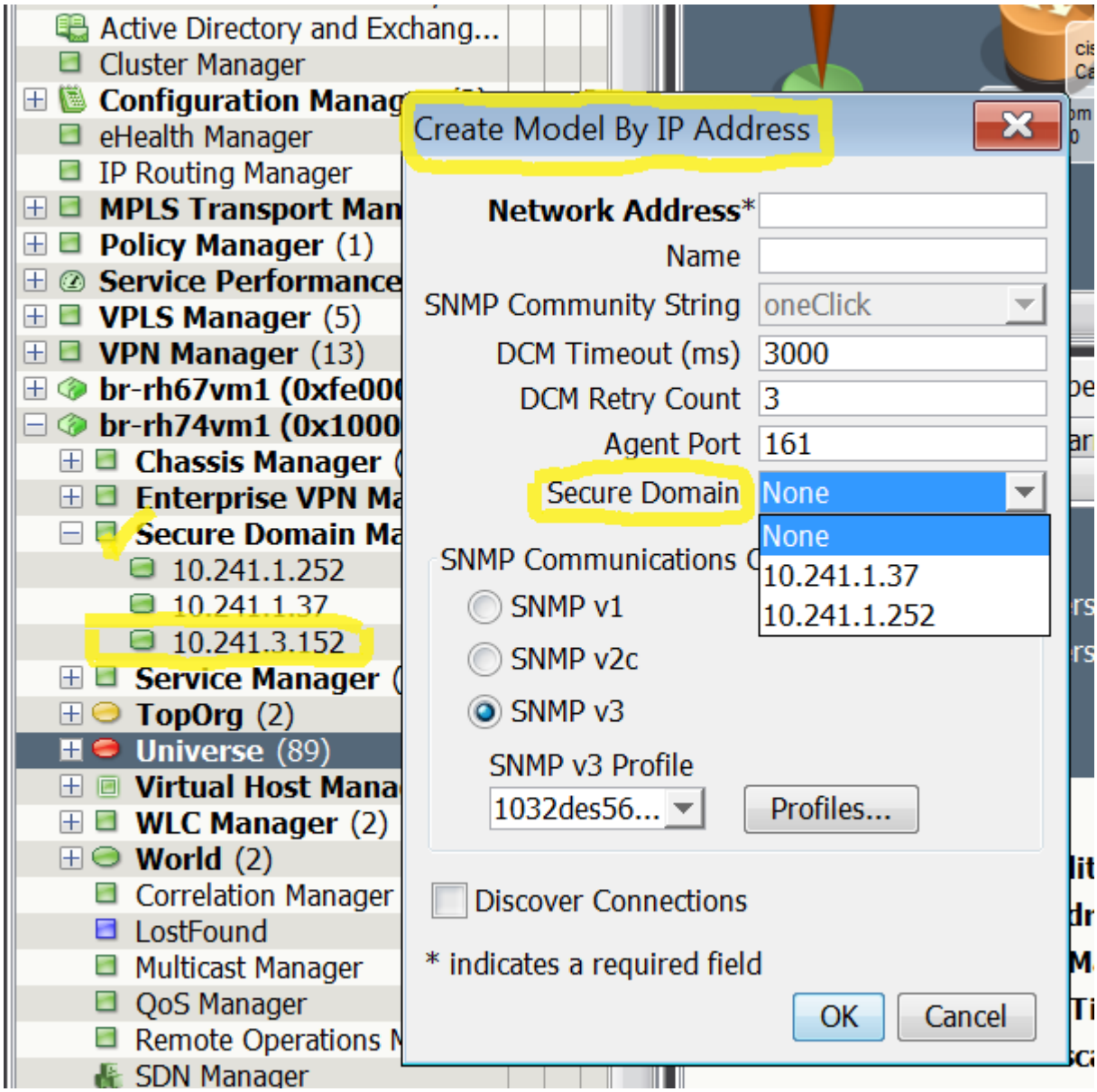
To convert the SDC feature into a TrapX capability, enable the TrapX checkbox during the Secure Domain Connector installation as shown here:

NOTE

Upgrading from SDC/TrapX, which is already installed, to TrapX/SDC through silent installation fails, and returns an error message in the "InstallationError.log" file under the SDMConnector/bin folder. For more information on the workaround, refer to the KB article on the CA support site.

Expected Behavior

- All v3 profiles created under the SDC domain during the SNMPv3 profile creation, are pushed to SDC for traps to be processed.
- If SDC is installed as TrapX then the same SDC cannot be used for managing devices (includes polling) under the Secure Domain. It is recommended to have a dedicated VM/physical box to deploy SDC TrapX.
- SDC profiles created under the Secure Domain Manager (SDM) are not modeled under the Model By IP, Discovery Console and the MIB Tools. For example, **10.241.3.152** is a SDC TrapX, listed under the 'Secure Domain Manager', as shown in the screenshot below, but **10.241.3.152** does not get modeled under the 'Secure Domain' list in the 'Model By IP Address' window.



SpectroSERVER and SDC TrapX Support on the Same Computer

Users can install SDC TrapX on a SpectroSERVER computer provided SDC TrapX and SpectroSERVER versions are the same.

NOTE

In this scenario, the SNMP communication port must be changed on SpectroSERVER and SDC. Both SpectroSERVER and SDC will be listening on 162.

For SpectroSERVER, change the value of the snmp_comm_port parameter in the $SPECROOT\SS\bin.vnmrc file.

For SDC, change the value of the snmp_comm_port parameter in the /opt/CA/SDMConnector/bin/sdc.rc file.

Configure Filters

To configure filters:

- Create the trapX.config file.
- Add the filters in the configuration file. For filter expressions for TrapX use the following format:

```
filter DateTime SrcIP Agent TrapType SpecificType Enterprise Action [Option]
```

Filter Name/Type	Description	Filter Value
DateTime (Optional)	A regular expression indicating the date and time that the trap was received by SDC TrapX.	For example, Fri May 11 09:23:34 EDT 2001. **Note**: This value does not necessarily indicate the time that the trap was sent by the device.
SrcIP (Optional)	An IP address-based regular expression that SDC TrapX uses to match the source IP, as in the IP packet header. The IP address from which the trap was received is not always equivalent to the agent IP address in the Trap PDU. The regular expression indicates that any IP address causes a match. You can specify a host name instead of an IP address for this field.	
AgentIP (Optional)	An IP address-based regular expression that SDC TrapX uses to match the IP address of the managed object that generated the trap (as in the Agent IP Address field in the Trap PDU). The IP address from which the trap was received is not always equivalent to the agent IP address in the Trap PDU. The regular expression indicates that any IP address causes a match. You can specify a host name instead of an IP address for this field. **Note**: Ensure that you add a backslash character (\) before any period (.) components that appear within the IP address. The period (.) is a special character in regular expression syntax.	
TrapType (Optional)	An integer-based regular expression that SDC TrapX compares to the Trap PDU's TrapType field.	Following are the Valid SNMP TrapType values: ■ coldStart(0) ■ warmStart(1) ■ linkDown(2) ■ linkUp(3) ■ authenticationFailure(4) ■ egpNeighborloss(5) ■ enterpriseSpecific(6)
SpecificType (Optional)	An integer-based regular expression that SDC TrapX compares to the Trap PDU's SpecificType field. Any integer is a valid value for this field.	
Enterprise (Optional)	An OID-based regular expression that SDC TrapX compares to the Trap PDU's enterprise field. **Note**: Ensure that you add a backslash character (\) before any period (.) components that appear within the OID. The period (.) is a special character in regular expression syntax.	

Action (Optional)	Action (Optional) A keyword that indicates the action that SDC TrapX performs, they include:
	• **file:** Logs the Trap PDU to a file specified by the Option field. SDC TrapX creates the file if it does not exist. This option is applicable only for SNMPv1 traps SNMPv2 traps, when performing SNMPv2c to SNMPv1 trap translation.
	• **forward:** Forwards the Trap PDU through UDP to a host specified by the Option field. Use this option if the trap receiver does not support TCP or if the TCP connection is broken.
	• **break:** This action causes no action to be taken for a given trap. No further filter processing is done on the current Trap PDU, and does not evaluate any remaining filters for the current Trap. Any option to this action is ignored.
	• **Tcp:** Forwards traps through a TCP connection without buffering the traps. This action drops the traps if SDC TrapX cannot connect to the remote trap receiver.
	• **Tcpbuff:** Forwards traps through a TCP connection. This action saves traps until SDC TrapX is able to connect to the remote trap receiver (or until the timeout limit is reached). Number of traps that are buffered depends on the queue size that is specified in the filter.
	Note: If you are using the tcp or tcpbuff actions and you receive the error message, “SDC TrapX: tcp forw detected a broken socket to: [port],” the TCP connection is broken or invalid. Use the forward action (which forwards traps through UDP).
	• **blind:** Forwards traps without parsing or decoding them first. This feature is useful for forwarding malformed traps or unsupported SNMP versions. It enables filtering only on the source IP address.
	• **exec:** This action executes a program or a script. The path to the script should be an absolute path, and the entire path and arguments must be enclosed in single quotes. For example: **exec '/path/to/script [arg1 [arg2...]] '**
	`exec '/tmp/trapScript.pl 0 123 4321'`
	`exec 'c:\temp\trapScript.pl 0 123 4321'`
	To run a script, give an absolute path of executable as well, for example:
	`filter * * * * * * exec 'C:\\win32app\\Spectrum\ \bin\\perl.exe C:\\abc.pl'`
	• **nat:** This action translates the agent-addr field in the trap to another IP address.
	Note: This action only does the address conversion. No forwarding is implied by this action. Therefore, place this rule before any forwarding rules that the nat action should be in effect. For example: **nat new-agent-addr** or **nat 1.2.3.4**
	• **tunnelfwd:** This action lets you forward traps through the SDC-SS tunnel. This action works only in that environment where SDC and SS are connected. For example, the -remoteconnect <sdc ip> option in sdm.config and the -accept <ss ip > option in sdc.config.
	An example of the tunnelfwd action is as follows:
	`filter * * * * * * tunnelfwd`
	Another example is as follows. This example can filter traps that are modeled on SDC:
	`filter * 1\.74\.24\.139 * * * * tunnelfwd` SDC TrapX will forward traps to all the connected SpectroSERVERs.	

NOTE

Restart the SDC services if you have modified the trapX.config file (for example, filters or global parameters).

NOTE

The TrapX feature from SDM supports the -remoteconnect parameter and from SDC supports the -accept parameter.

Enable Translation

You can globally enable translation of SNMP v2c traps to SNMPv1 traps by adding the translate_v2c_traps option to the trapX.config file. Follow these steps:

1. Add **translate_v2c_traps** in a trapX.config file.
2. If the device sends v2c traps without the snmpTrapAddress, enable the following option:
   agentaddr_is_srcaddr_translated_v1
   This option makes the v2c trap agent address the same as the source address. This option is used when the v2c to v1 translation and v3 to v1 translation is enabled. If this translation is not enabled, this option cannot be used.

To enable translation of SNMPv3 traps to SNMPv1 traps add translate_v3_traps:1:test to the conf. file. For example:

```
filter * * * * * * forward  138.42.86.54 translate_v3_traps:1:xyz
```

Here option 1 is to enable transition, it coverts v3 traps to v1 with the community string if this option not provided it will it take public.

Sample Filters

This section includes sample filters that you can add to the trapX.config file. To add the filters:

Ensure that you have first created the trapX.config file. You can use these examples to help design filters suitable for your environment. In these examples, asterisks (*) indicate placeholders for fields for which you do not want to filter on a specific value.

- **Match Trap PDUs from a Local Host:** These examples match all Trap PDUs from the local host, and effectively drop and suspend filter processing for them.
  ```
  filter * * 127\.0\.0\.1 * * * break
  filter * * ::1 * * * break
  ```

  NOTE

  By default, SDC trapX listens for traps on UDP port 162 and on TCP port 161 (if listening is enabled on the TCP port). You can configure these ports by changing the "snmp_trap_port=" value in the sdc.rc file for UDP and the trapX.config file for TCP.

- **Match Authentication Failure Traps:** This example matches all authenticationFailure (4) traps and forwards them to the system named concord at UDP port 162 (the default).
  ```
  filter * * * 4 * * forward concord
  ```
- **Match Private-Enterprise Traps:** This example matches all private-enterprise traps of SpecificType 3 through 8 and forwards them to the system named concord at UDP port 191.
  ```
  filter * * * 6 [3-8] * forward concord:191
  ```
- **Match Traps by Enterprise OID**: This example matches all traps that contain the enterprise OID 1.3.6.1.4.1.546.1.1 and forwards them to the system named ottoman at UDP port 162 (the default).
  ```
  filter * * * * * 1\.3\.6\.1\.4\.1\.546\.1\.1 forward ottoman
  ```

  NOTE

  A backslash character (\) appears before each period character (.) so that the period character is read correctly as part of the enterprise ID and not as a regular expression wildcard operation.

- **Match Traps by Date:** This example matches all traps that SDC TrapX received on Friday and forwards them to the system named ottoman.
```
filter "Fri" * * * * * forward ottoman
```
- **Match Traps by Source IP Address:** These examples match all traps that originated from the source IPv4 address 199.250.183.215 and forwards them to the system named ottoman.
```
filter * 199\.250\.183\.215 * * * * forward ottoman
filter * fe80::a00:20ff:fe8c:af7e * * * * forward ottoman
```
- **Match Traps by Agent IP Address:** These examples match all traps that were sent by a managed object with an IPv4 address of 199.250.183.215 and forwards them to the system named ottoman.
```
filter * * 199\.250\.183\.215 * * * forward ottoman
 filter * * fe80::a00:20ff:fe8c:af7e * * * forward ottoman
```
- **agentaddr_is_srcaddr_translated_v1:** This command works with translate_v2c_traps and translate_v3_traps, when translate_v2c_traps or translate_v3_traps are turned on, this makes the v2c trap or v3 trap agent address as the source address. By default it is commented out.

Varbind Filtering

The 10.4.1 release supports the SDC TrapX varbind filtering. This filtering supports the following operations that you can use with a combination of **AND** and **OR** statements:

- Equals
- NotEquals
- StartsWith
- EndsWith
- Contains
- NotContains

Some examples are as follows:

- **AND example:** In the following example, if all the operations are mentioned inside AND[] and if they are matched, then traps get processed:
```
filter * * * * * 1.3.6.1.6.3.1.1.5   forward 10.241.3.151 translate_v3_traps:0:public
"AND[1.3.6.1.2.1.2.2.1.2.1:Equals:FastEthernet0/0,1.3.6.1.4.1.9.2.2.1.1.20.1:EndsWith:p,1.3.6
```
- **OR example:** In the following example, if either one of the operations is inside OR[] and if it matches, then traps get processed:
```
filter * * * * * 1.3.6.1.6.3.1.1.5   forward 10.241.3.151 translate_v3_traps:0:public
"OR[1.3.6.1.2.1.2.2.1.2.1:Equals:FastEthernet0/0,1.3.6.1.4.1.9.2.2.1.1.20.1:EndsWith:p,1.3.6.
```
- **file example:** The following is a file example:
```
filter * * * * * 1.3.6.1.6.3.1.1.5  file /opt/CA/SDMConnector/bin/trapX.txt
 "AND[1.3.6.1.4.1.9.2.2.1.1.20.1:Equals:Link Down Trap]" 10
filter * * * * * *  file /opt/CA/SDMConnector/bin/trapX1.txt * 10
```
(10 represent file size; it is optional.)
- **break statement:** If you want to forward all the traps without filtering except a few traps with varbind filters, then you can review the following example. In this example, all the traps are forwarded to the destination 10.241.3.151 except traps 1.3.6.1.6.3.1.1.5 and 1.3.6.1.4.1.9.9.187.
1.3.6.1.6.3.1.1.5 traps are forwarded to the destination 10.241.3.151 when the varbind filter criteria matches and then the break statement does not execute further filters. This ensures that traps of these types are not forwarded to the destination again if the varbind filter criteria is a success or failure.

1.3.6.1.4.1.9.9.187 traps are forwarded to the destination 10.241.3.151 when the varbind filter criteria matches and then the break statement does not execute further filters. This ensures that traps of these types are not forwarded to the destination again if the varbind filter criteria is a success or failure.

```
filter * * * * * 1.3.6.1.6.3.1.1.5 forward 10.241.3.151 translate_v3_traps:0:public
"AND[1.3.6.1.6.3.1.1.4.1.0:Equals:1.3.6.1.6.3.1.1.5.3]"
filter * * * * * 1.3.6.1.6.3.1.1.5 break
filter * * * * * 1.3.6.1.4.1.9.9.187 forward 10.241.3.151 translate_v3_traps:1:public
"AND[1.3.6.1.6.3.1.1.4.1.0:Equals:1.3.6.1.4.1.9.9.187.0.2]"
filter * * * * * 1.3.6.1.4.1.9.9.187 break
```

- **tcp buff:** The following is a tcp buff example:

```
filter * * * * * 1.3.6.1.4.1.9.9.276.0.1.0.1 tcpbuff  10.241.3.151:5058
 translate_v3_traps:0:public 60 300 AND["1.3.6.1.2.1.2.2.1.7.436240384:equals:1"]
```

- **tcp forward:** The following is a tcp forward example:

```
filter * * * * * 1.3.6.1.4 tcp  138.42.246.37:1771  translate_v3_traps:0:public 300
 AND["1.3.6.1.2.1.2.2.1.1:Contains:10]"
```

Forward Traps through TCP Connections

TrapX can forward traps through TCP connections when you specify the host name (or IP address), port, and a connection timeout value. TrapX provides two actions for forwarding traps through TCP: tcp and tcpbuff. When you specify the tcp action for trap filtering in the TrapX file, TrapX does not buffer the traps. In that case, if the trap receiver is unavailable, TrapX drops the traps. When you specify the tcpbuff action, TrapX can queue the traps and then send them when the trap receiver restarts, providing better management of TCP connections than the tcp action provides. You can forward traps through TCP with or without buffering. To buffer the traps (save them if the trap receiver is unavailable), use the tcpbuff action. To filter traps without buffering them, use the tcp action. For example:

NOTE

Forwarding traps through TCP does not provide security, privacy, or authentication. It simply enhances the reliability of the trap reception. Listen_for_tcp_traps on 1771 and forward the traps on the TCP port in trapX.config file. For example, filter * * * * * * tcp 10.241.3.151:1771 translate_v3_traps 300

- **Filter Traps through TCP with buffering**: This example forwards traps through a TCP connection to a system with a hostname of violet on port 5058 with a buffer size equivalent to 60 traps and a timeout value of 300 seconds.

```
filter * * * * * *  tcpbuff  violet:5058  translate_v3_traps 60 300
```

  CA SDC TrapX buffer traps in a buffer that can hold a maximum of 60 traps for 300 seconds before dropping them.
- **Filter Traps through TCP without buffering**: This example forwards traps without buffering through a TCP connection to a system with a hostname of electrode on port 162 with a timeout value of 30 seconds.

```
filter * * * * * * tcp electrode:162 30
```

- **Forward Traps through UDP Connections:** This example forwards traps through a UDP connection to a system with a hostname of orange on port 5058. You can forward traps through UDP when the trap receiver does not support TCP.

```
filter * * * * * * forward orange:5058
```

- **Forward Traps Blindly** This example forwards traps to a system with a hostname of lemon on port 5058 without parsing or decoding. You can blindly forward traps without parsing or decoding.

```
filter * * * * * * blind lemon:5058
```

SDC TrapX Limitations

The limitations with this feature include:

- The multi-nic and SDC TrapX High Availability (HA) are not supported with this release.
- Installing SDC Trapx on an SDC computer is not supported.
- The filters that not supported with this release are listed as follows:

Filter Name	Value	Description
overloaded_header	overloaded_header	This flag overloads 16-byte agent IP address into SNMPv1 trap PDU, while translating v2c traps to v1. This flag is effective only when the translate_v2c_traps flag is on.
so_rcvbuf <buffer-size-in-bytes>	so_rcvbuf 128000	This command sets the size of the socket receive buffer. This applies to both the UDP and TCP listening sockets. The default is 128000 bytes.
tcp_receive_timeout <timeout-in-seconds>	tcp_receive_timeout 5	This command defines the timeout for recv() operations on the TCP listen socket, in seconds. Essentially, this keeps TrapEXPLODER from being indefinitely blocked by a rogue TCP connection. A value of 0 disables the timeout, allowing TCP recv() operations to block indefinitely. This option only has an effect if listen_for_tcp_traps is set to 'on'. The default is 5 seconds.
'aview' action	aview /opt/aview/var/traps aview c:\aview\var\traps	This action writes out traps in the aview format. This action does not work, if the Trap Exploder is running on an eHealth® machine with a Fault Manager license present.
'eh' action	eh 1.2.3.4:666 666 30	This action is used with CA eHealth® 5.0 to forward traps to the eHealth trap receiver. This action is deprecated with CA eHealth® 5.5 and higher.

Installation Files

Be aware of the following directories and files that are created during the installation process.

On the SpectroSERVER

The DX NetOps Spectrum installation process installs the following Secure Domain Manager directories and files in the <*$SPECROOT*>/SDM directory on the SpectroSERVER:

- **cert**
  This directory is the repository for the SSL certificates you create for the SDManager.
- **Logs**
  This directory contains output logs that are generated when you import a configuration file to the SpectroSERVER. Detail of the work that is performed. The work includes any errors that occur, are contained in the log file.
- **README**
  This file provides details on how to configure Secure Domain Manager on the SpectroSERVER host.

On the SDConnector Host

The SDConnector installation process installs the following directories and files in the SDMConnector directory on the SDConnector host:

- **bin**
  This folder contains the following items for working with the SDConnector:
  - **cert**
    This directory is the repository for the SSL certificates you create for the SDConnector.
  - **README**
    This file provides details on how to configure the SDConnector process on the SDConnector host.
  - **SdmConnectorService[.exe]**
    The executable file for the SDConnector.

Change Permissions (.ssh\config) on SDC Cygwin

In 10.4.2, Cygwin 3.1 with Perl modules is shipped with SDC on Windows. These modules are required for executing scripts to perform NCM operations on SDC-managed devices. On Linux, it must be installed manually.

> **NOTE**
> For more information about scripts, see the Perl Modules section in Network Configuration Manager Extension Utility. For more information about NCM operations on SDC-modeled devices, see NCM Enablement in Secure Domain.

You must change the permissions under .ssh\config from SYSTEM to Administrator on the SDC Cygwin to perform the configurations.

1. Open the command prompt.
2. Navigate to the "$SDCINSTALL$NT_TOOLS/SDCRE" directory.
   For example, "cd C:\Program Files\CA\SDMConnector\NT-TOOLS\SDCRE"
3. Execute `sdc.bat uninstall` .
   This command changes the permissions to Administrator.
4. Open and perform the required changes to the configuration ("$SDCINSTALL$\NT-TOOLS\SDCRE\home\SYSTEM \.ssh\config").
   For example, C:\Program Files\CA\SDMConnector\NT-TOOLS\SDCRE\home\SYSTEM\.ssh\config
5. Execute `sdc.bat install` .
   This command changes the permissions back to SYSTEM.

Working with Certificates

Certificates are loaded by default for both SDManager and SDConnector. This lets you use the SSL encryption to secure ICMP and SNMP (SNMPv1, SNMPv2c, and SNMPv3) data that is transmitted between SDManager and SDConnector hosts across nonsecured networks. If you do not want to use the SSL encryption for any SDManager-SDConnector connections in your network environment, include the nonsecure option in the configuration files for SDManager and SDConnectors. For more information about how to use the nosecure optionsee Configure SDConnector Process Settings.

Create Certificates

Secure Domain Manager uses digital certificates to ensure the security. The default certificates are provided with your DX NetOps Spectrum installation and site-specific certificates can be created using the CertGen tool.

Default Certificates

If you want to use the default certificates, do not perform any actions. All default files reside in the <*$SPECROOT*>/SDM/cert directory and include the following files:

- **SDMCA.pem**
  Certificate authority. Distribute this file to any computer that uses Secure Domain Manager or Secure Domain Connector in any capacity and can be treated as a trusted CA file.
- **SDMCAKey.pem**
  Private key of CA. It can be used to issue certificates but should not necessarily be distributed to any machines.
- **SDMCert.p12**
  Application certificate that is signed by SDMCA.pem. This is the certificate file that is used between SDManager and SDConnector. It should be carefully distributed to computers that deserve trust and used to assert the identity of those computers.
- **CertGen[.exe]**
  Program that is used to generate the site-specific certificate authority, key file, and certificate file. Run CertGen -h to review all certificate options available.
- **openssl[.exe]**
  OpenSSL open source implementation of the SSL protocol.

Site-Specific Certificates

If you want to create site-specific certificates, move the default certificate files (*.pem and *.p12) to another location on the hard drive. Perform the following procedures to create and deploy the custom certificates.

Create Site-Specific Certificates

Create site-specific certificates for better security. Create these certificates on a single computer that only qualified personnel can access. This computer can be the SDManager host.

> **WARNING**
> You must have administrator or root privileges to create the SSL certificates for Secure Domain Manager.

Follow these steps:

1. Run the following command to create a certificate authority certificate and the private key for the certificate authority certificate:
   ```
   CertGen -t ca -c US
   ```
   You only have to perform this step once to create the necessary certificate authority certificate for your organization. The following files are created:
   a. SDMCA.pem
   b. SDMCAKey.pem

   > **NOTE**
   > The default certificate authority and key file that come with Secure Domain Manager are read-only files. If you receive a permission error, check your user privileges or move SDMCA.pem and SDMCAKey.pem to another location and run the command again.
2. Run the following command to create a certificate for the SDManager:
   ```
   CertGen -t cert -c <Country Code>
   ```
   The SDMCert.01.p12 file is created.
3. (Optional) For the added security, use the -p option to generate the certificate with a password as follows:
   ```
   CertGen -t cert -p <password> -c <Country Code>
   ```
   Enter the password in the sdc.config file and sdm.config file.
4. Rename SDMCert.01.p12 to SDMCert.p12.

The new site-specific certificate is ready for use.

Deploy Site-Specific Certificates

After you create your certificate files, perform the following tasks:

- Deploy the certificate files on the SDManager hosts and on the SDConnector hosts.
- Restart the SpectroSERVER on the SDManager hosts and the SDConnector process on the SDC hosts.

To deploy certificates, copy the SDMCA.pem file that you created to the <*$SPECROOT*>/SDM/cert directory on the SDManager host computer and to the cert directory under the SDConnector installation on the SDConnector hosts that will connect to the SDManager host. Administrator, or root should own the SDMCert.p12 file.

WARNING
Retain the SDMCAKey.pem file on the computer where you plan to create more certificates. Restrict the file to authorized personnel only. This computer can be the SDManager host computer but is not a requirement.

After the certificates have been deployed, restart both the SpectroSERVER on the SDManager hosts and the SDConnector process on the SDC hosts. For information on restarting the SDConnector Process, see Start, Stop, and Restart the SDConnector Process on Windows or Start, Stop, and Restart the SDConnector Process on Linux.

Configure SDConnector Process Settings

This section describes the configuration options you can set in the SDConnector configuration file (sdc.config). This configuration file is read at startup and the options specified are applied at that time. Only one line of options is accepted in the sdc.config. The following is a sample line from an sdc.config file. It specifies that the SDConnector will accept connections from SDManager (192.168.0.2):

```
-accept 192.168.0.2
```

To configure SDConnector settings

1. Create (or open, if it already exists) a file named 'sdc.config' in the SDMConnector\bin directory on the SDConnector host machine using a text editor.
2. Add and specify details for the following options on one line in the file, according to your particular requirements:
   - **-accept remote_ipaddr:[local_port]**
     Accepts a connection from an SDManager running on a host at address <*ip*> at local port number < *port*>. Connections must originate from the IP address specified; otherwise, connection attempts are disregarded.
     If this option is specified, the SDManager that connects to this SDConnector must have the -remoteconnect option that specifies this SDConnector <*ip*> in its configuration file (sdm.config). Also, if this option is specified you cannot connect (-connect) to that SDManager.
   - **-buffersize <*size*>**
     Specifies the size of the send and receive socket buffer sizes in bytes.
     Default: 262,144 (256k, which should be sufficient in most deployments)
   - **-certdir <*dir*>**
     Specifies the directory for SSL certificates (application certificate, private key, and the certificate authority certificate) if they are not located in the default directory (/cert).
     If the -nosecure option is specified, certificates are not accessed.
   - **-certpassword <passwd>**
     Provides the certificate password. If you are using the default certificates that ship with Secure Domain Manager, then -certpassword need not be supplied. Otherwise, supply the certificate password using this option. If the password contains spaces, it must be enclosed in quotation marks ("). DX NetOps Spectrum assumes that the password for the application certificate will be encrypted.

NOTE
If you use -certpassword, it must be the first option declared in the config file.

- **-connect remote_ipaddr:[remote_port]**
  Connects to the SDManager running on a host at IP address <*ip*> and port <*port*>. If <*port*> is not specified, 6844 is assumed.
  If this option is specified, the SDManager to which this SDConnector connects must have the -remoteaccept option that specifies this SDConnector's IP address in its configuration file (sdm.config).
  If this option is specified, this SDConnector cannot accept (-accept) connections from or listen (-listen) for connections from the specified SDManager (sdm.config).
- **-keepalive <*n*>**
  Changes the default internal timeout (in seconds) when the SDManager or SDConnector sends out a small message to verify the network connection is still alive. If either the SDManager or an SDConnector does not hear from the other within three times the value of <*n*>, the connection is terminated.
  Default: 10 seconds
- **-listen [port]**
  By default, the SDConnector listens at port 6844 for connection requests from any SDManagers. However, if any -connect or -accept options are specified, then the SDConnector no longer listens by default.
  A port specified in a -listen option trumps a port specified in an -accept option. That is, if a port is specified in a -listen option, there will be no verification done of the source IP address for that port.
  Note: -listen and -listen6 are mutually exclusive.
- **-listen6 [local_port]**
  Accept connections from any IPv6 SDManager on the given port.

  NOTE
  -listen and -listen6 are mutually exclusive.
- **-loglevel fatal|error|warning|info|debug**
  Specifies the types of messages to log.
  Default: warning (includes error and fatal as well)
- **-maxlogsize <*n*>**
  Sets the maximum sdmLog.log size in megabytes.
  Default: 5M
  Minimum: 1M
- **-nosecure**
  Disables Secure Socket Layers (SSL) security, which is enabled by default. If the -nosecure option is used before any -connect or -accept entries, SSL is disabled for all connections. Otherwise, you can specify the -nosecure option after each -connect or -accept entry and it will pertain just to that entry.
  If SSL security is requested, the data stream is encrypted, and mutual cryptographic authentication is enforced. If either the SDManager or the SDConnector requests security, then security is mandatory on that connection.
- **-trappoll <*n*>**
  Forward traps to the SDManager every <*n*> seconds.
  Default: 15 seconds
- **-withfips**
  Specifies to run with FIPS mode. FIPS mode is off by default.

  NOTE
  If an empty sdc.config is created, SDConnector listens for connections from any SDManager on port 6844; the SDManager initiates the connection.

3. Save and exit the file.
   The SDConnector is configured.

   NOTE
   You must restart the SDConnector process every time you make updates to the sdc.config file.

Configure SDManager Process Settings

The SDManager configuration file (sdm.config) specifies the operational settings for the SDManager process. By default, the SDManager process is disabled. The SDManager process will not work until you create the sdm.config file and configure it according to your needs. After you configure the sdm.config file for the first time, or any time you revise its settings, you must import it into DX NetOps Spectrum to put SDManager settings into effect on the SpectroSERVER. For more information, see Import the SDManager Configuration File. You can configure the sdm.config either before or after the SpectroSERVER is started.

Only one line of options is accepted in the sdm.config. The following is a sample line of options for an sdm.config. It is specifying the connections (-remoteconnect) to two SDConnectors (172.24.148.196 and 172.19.32.199):

```
-remoteconnect 172.24.148.196 -remoteconnect 172.19.32.199
```

NOTE
If you use the -nosecure option to launch one or more of the SDConnector processes, you must specify the same -nosecure option for the corresponding -remoteconnect/-remoteaccept entry in the SDManager options, or simply specify -nosecure before all -remoteconnect/-remoteaccept entries to disable SSL for all connections.

To configure SDManager settings

1. Create (or open, if it already exists) a file named 'sdm.config' in the <*SPECROOT*>\SDM directory on the SpectroSERVER host machine using a text editor.
2. Add and specify details for the following options on one line in the file, according to your particular requirements:
   - **-apiclientport [port]**
     Sets the port to listen for API client connections. This parameter applies to the stand-alone SDManager process only.
   - **-buffersize <*size*>**
     Specifies the size of the send and receive socket buffer sizes in bytes.
     Default: 262,144 (256k, which should be sufficient in most deployments)
   - **-certdir <*dir*>**
     Specifies the directory for SSL certificates (application certificate, private key, and the certificate authority certificate) if they are not located in the default directory (/cert).
     If the -nosecure option is specified, certificates are not accessed.
   - **-certpassword <passwd>**
     Provides the certificate password. If you are using the default certificates that ship with Secure Domain Manager, then -certpassword need not be supplied. Otherwise, supply the certificate password using this option. If the password contains spaces, it must be enclosed in quotation marks ("). DX NetOps Spectrum assumes that the password for the application certificate will be encrypted.

     NOTE
     If you use -certpassword, it must be the first option declared in the config file.
   - **-clientServiceThreads <*n*>**
     Sets the number of threads per client that will process requests. This parameter applies to the stand-alone SDManager process only.
   - **-keepalive <*n*>**
     Changes the default internal timeout (in seconds) when the SDManager or SDConnector sends out a small message to verify the network connection is still alive.
     Default: 10 seconds
     If either the SDManager or an SDConnector does not hear from the other within three times the value of <*n*>, the connection is terminated.
   - **-loglevel fatal|error|warning|info|debug**
     Specifies the types of messages to log.

Default: warning (includes error and fatal as well)

- **-maxapiconnections <*n*>**
  Sets the maximum number of API client connections to <*n*>. This parameter applies to the stand-alone SDManager process only.
- **-maxlogsize <*n*>**
  Sets the maximum sdmLog.log size in megabytes.
  Default: 5M
  Minimum: 1M
- **-nosecure**
  Disables the Secure Socket Layers (SSL) functionality, which is enabled by default. If the -nosecure option is used before any -remoteconnect or -remoteaccept entries, SSL is disabled for all connections. Otherwise, you can specify the -nosecure option after each -remoteconnect or -remoteaccept entry and it will pertain just to that entry. If SSL security is requested, the data stream is encrypted, and mutual cryptographic authentication is enforced. If either the SDManager or the SDConnector requests security, then security is mandatory on that connection.
- **-remoteaccept remote_ipaddr[:local_port]**
  Accepts a connection from an SDConnector running on a host at address <*ip*> at local port number < *port*>. You must specify the SDConnector's public IP address.
  If this option is specified, the SDConnector that connects to this SDManager must have the -connect option that specifies this SDManager's IP address in its configuration file (sdc.config). Also, if this option is specified, you cannot connect (-remoteconnect) to the SDConnector (sdc.config).
- **-remotebackup (-remb) remote_ipaddr[:remote_port]**
  Specifies the backup SDConnector in a fault-tolerant Secure Domain Manager setup using the SDConnector's public IP address. For more information, See Setting Up Processes in a Fault-Tolerant Environment.
- **-remoteconnect remote_ipaddr[:remote_port]**
  Connects to the SDConnector running on a host at IP address <*ip*> and < *port*>. If < *port*> is not specified, 6844 is assumed. You must specify the SDConnector's public IP address.
  If this option is specified, the SDConnector to which this SDManager connects must have the -accept option that specifies this SDManager or the -listen option in its configuration file (sdc.config). Also, if this option is specified you cannot accept connections (-remoteaccept) from the specified SDConnector in this configuration file.
- **-withfips**
  Specifies to run with FIPS mode. FIPS mode is off by default. If changing the configuration from FIPS mode to non-FIPS, or vice versa, you must restart the application.

NOTE
If the sdm.config file is empty, the SDManager process is disabled.

3. Save and close the sdm.config file.
   The SDManager is configured.

Start, Stop, and Restart the SDConnector Process on Windows

Use the Services manager to start, stop, or restart the SDConnector process. The SDConnector process is listed under the name 'Secure Domain Connector.'

Start, Stop, and Restart the SDConnector Process on Linux

To start the SDConnector process, log in as root, open a command line console, and enter the following commands:
In 10.4.2, run the following command:

```
$ cd /etc/init.d
$ ./sdmconnector start
```

In 10.4.2.1, run the following command:

```
systemctl start sdmconnector
```

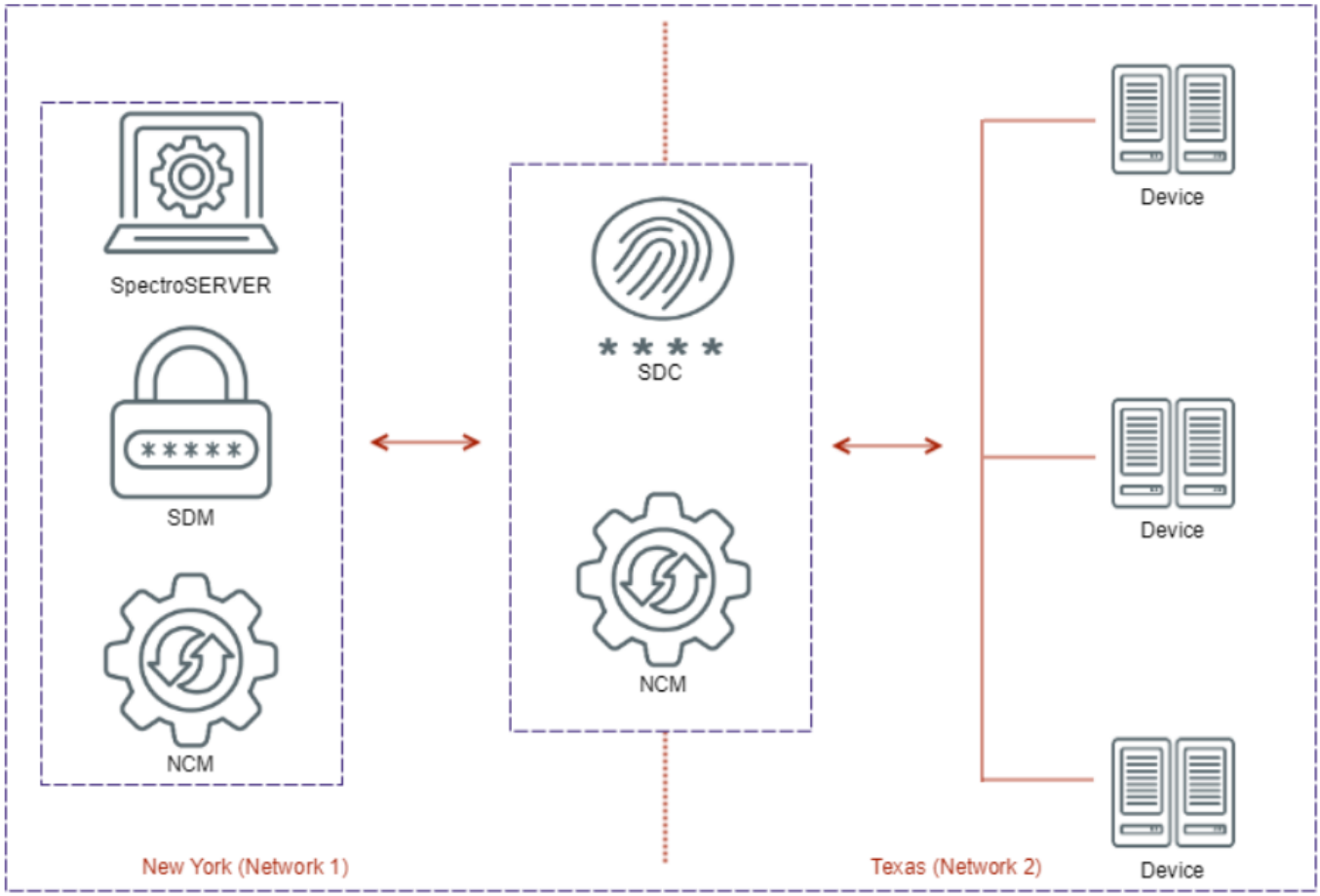
To stop the SDConnector process, issue the **./sdmconnector stop** command.

To restart the SDConnector process, issue the **./sdmconnector restart** command.

NCM Enablement in Secure Domain

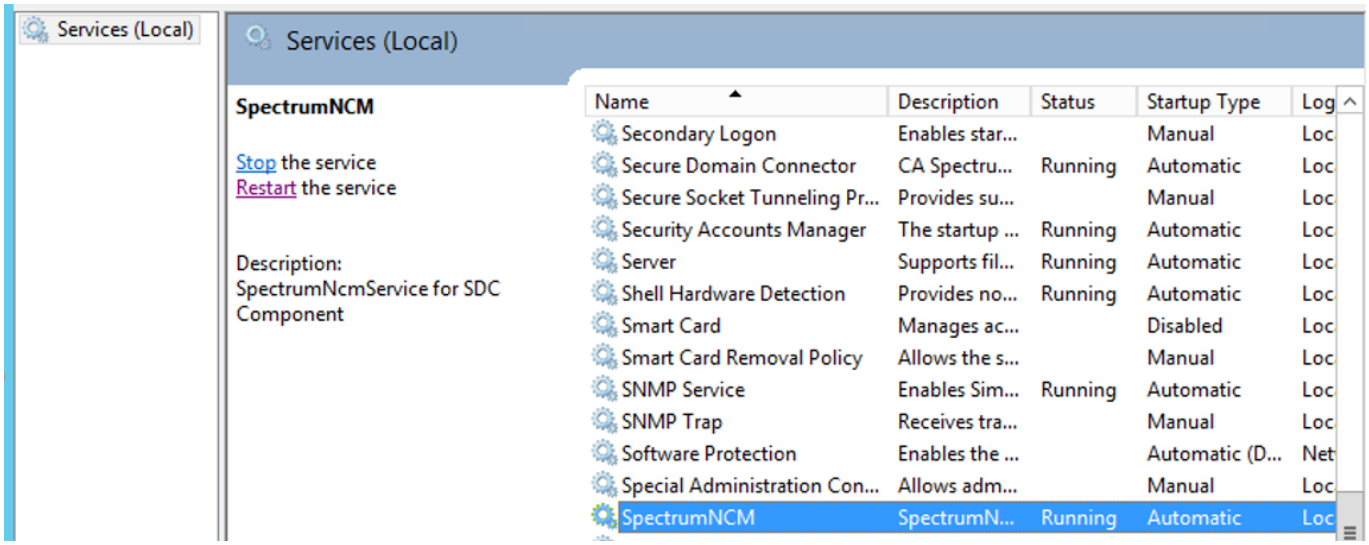
DX NetOps Spectrum supports Network Configuration Manager (NCM) operations (for example, load, sync, capture configurations) on the Secure Domain Connector (SDC)-managed models; that is, devices under a private network.

The following diagram shows the high-level representation:



When you install SDC, NCM is automatically installed as part of SDC; it is installed as a service.

- **Windows:** On Windows SDC, the SpectrumNCM service is installed for NCM. The following screenshot shows the service:



Note that the Secure Domain Connector service is also available.

NOTE
In 10.4.2, Cygwin with Perl modules is shipped with SDC on Windows. These modules are required for executing NCM scripts to perform NCM operations on SDC-managed devices. All the operations (such as capturing or writing a startup configuration, capturing or uploading a running configuration, uploading device firmware, reloading a device, and canceling the reload operation on a device) are supported. You can configure scripts within NCM for each of these operations and execute them on the SDC-managed devices. For more information about the NCM scripts, see the Perl Modules section in the Network Configuration Manager Extension Utility article.

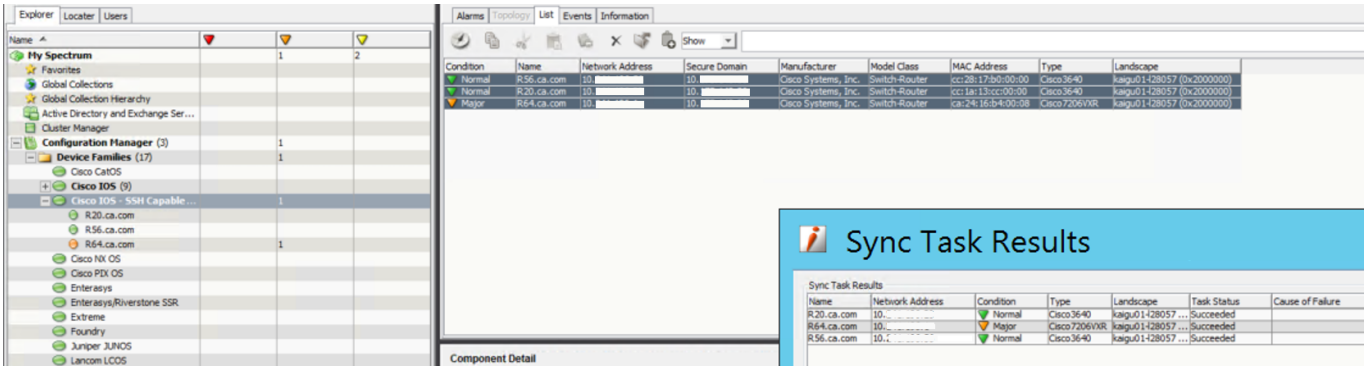
- **Linux:** On Linux SDC, use the following command to find out the SDC and NCM services:
  In 10.4.2, run the following command:
  ```
  [root@host ~]# /etc/init.d/sdmconnector status
  ```
  In 10.4.2.1, run the following command:
  ```
  [root@host ~]# systemctl status sradmin
  SdmConnectorService (pid 16467) is running...
  NCM Service is running pid - 16480
  [root@host ~]#
  ```

  NOTE
  In an upgraded 10.4.1 (or later) environment (for example, upgraded from 10.3.2 SS and SDC to 10.4.1), devices that have been modeled through SDC before the upgrade do not move on their own to the SSH-capable family. However, when the Reevaluate NCM Device Family action is performed, they move to the SSH-capable device family as expected.

SDC–NCM Support on Devices in Existing Device Family

Cisco devices that are modeled through SDC support SSH/SCP as a primary communication mode from the “Cisco IOS – SSH Capable” device family.

The following screenshot shows the required information:



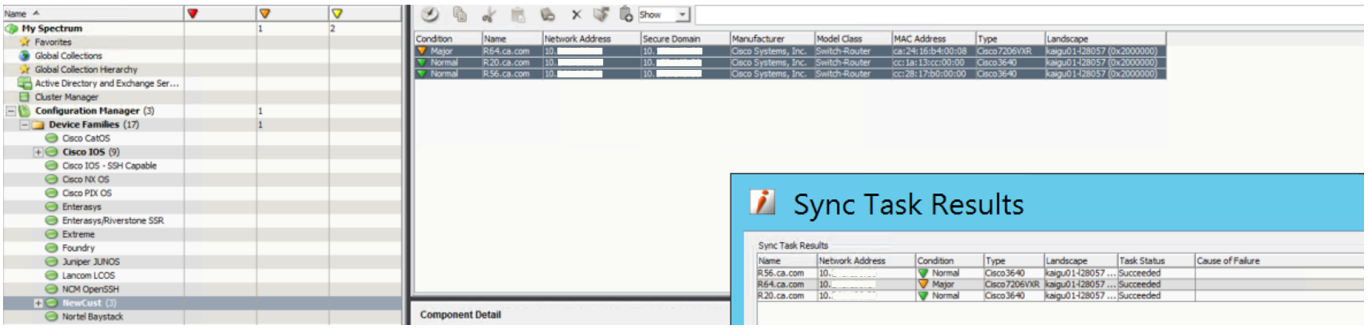
NOTE
In 10.4.2, you can use Perl scripts to execute the NCM operations on the devices modeled through SDC. For more information about how to use Perl scripts to perform various NCM operations, see Network Configuration Manager Extension Utility. In 10.4.1, such script operations were not supported on SDC-modeled devices.

NCM Self-Certification Supporting SDC–NCM

As part of NCM self-certification, it supports SSH as the only supported communication mode. In case of a new device family for NCM self-certification:

- Create a new device family.
- Add the new device modeled into it.
- Add the supported running, startup, and load commands from the newly created device family.

The following screenshot shows the required information:



Scenario: How to Sync Startup Configuration with Running Configuration

For a custom device family, you can use a Perl script to sync the startup configuration with the running configuration. For more information about using the script, see Network Configuration Manager Extension Utility.

Additionally, you can also use appropriate commands to sync the startup configuration with the running configuration. For example, in case of a Cisco device, the "write" command performs this job. Now, suppose you want to sync after you perform an upload task, there are two ways in which you can do this:

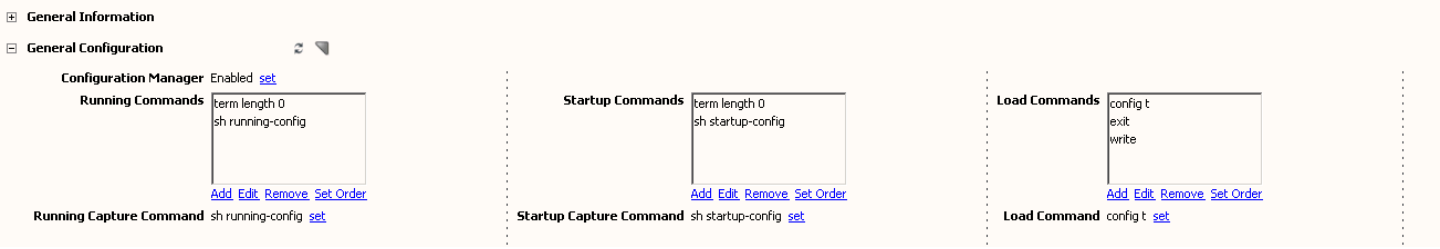
Using Command List

For the device family, navigate to **Information** tab **-> General Configuration**. Under "**Load Commands**", give the following command list:

```
config t
exit
write
```

This ensures that every time you do any upload task, the startup configuration is synced with the running configuration.

The following screenshot shows the required information:



Using Task

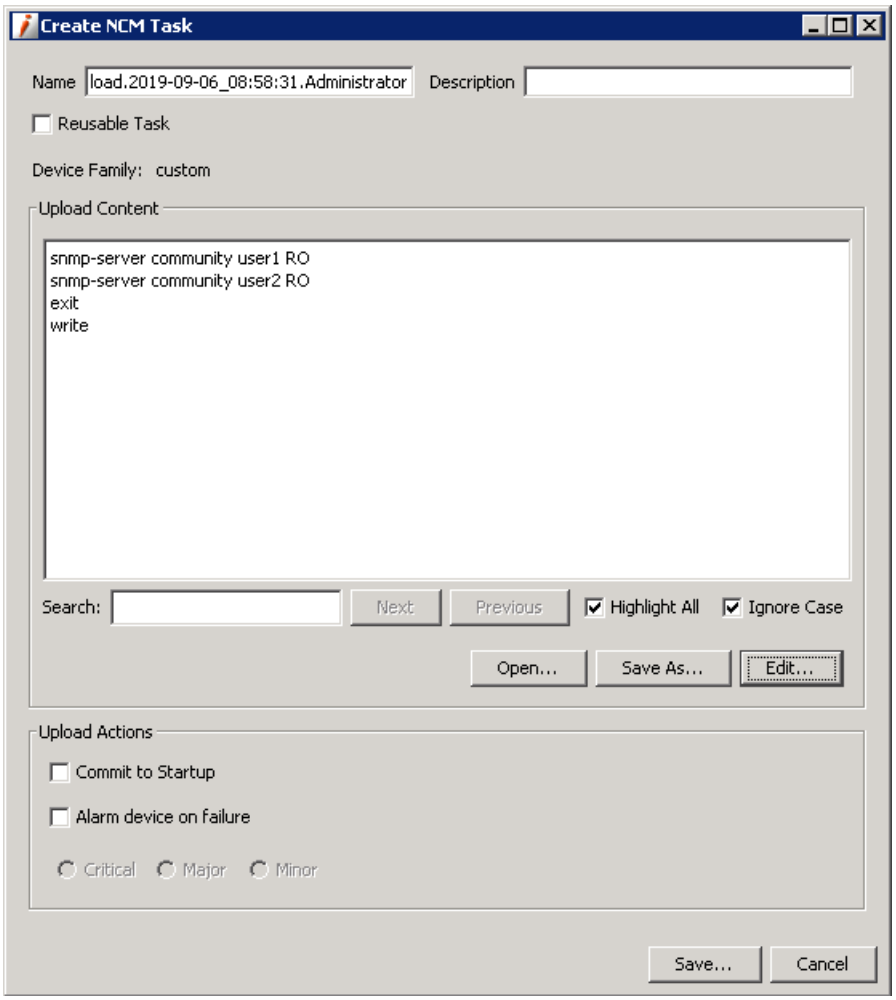
When you create an upload task, you add the following lines at the end of the upload content:

> **NOTE**
> Verify that the config t command is already added to the Load Commands list.

```
exit
write
```

This ensures that every time you run that upload task, the startup configuration is synced with the running configuration.

The following screenshot shows the required information:



NOTE
Commit to Startup is not applicable for self-certification, because it is taken care of by the proper sequence of the commands that users enter to perform the sync task. It is not done by enabling this option.

Configure the Secure Domain Time-out

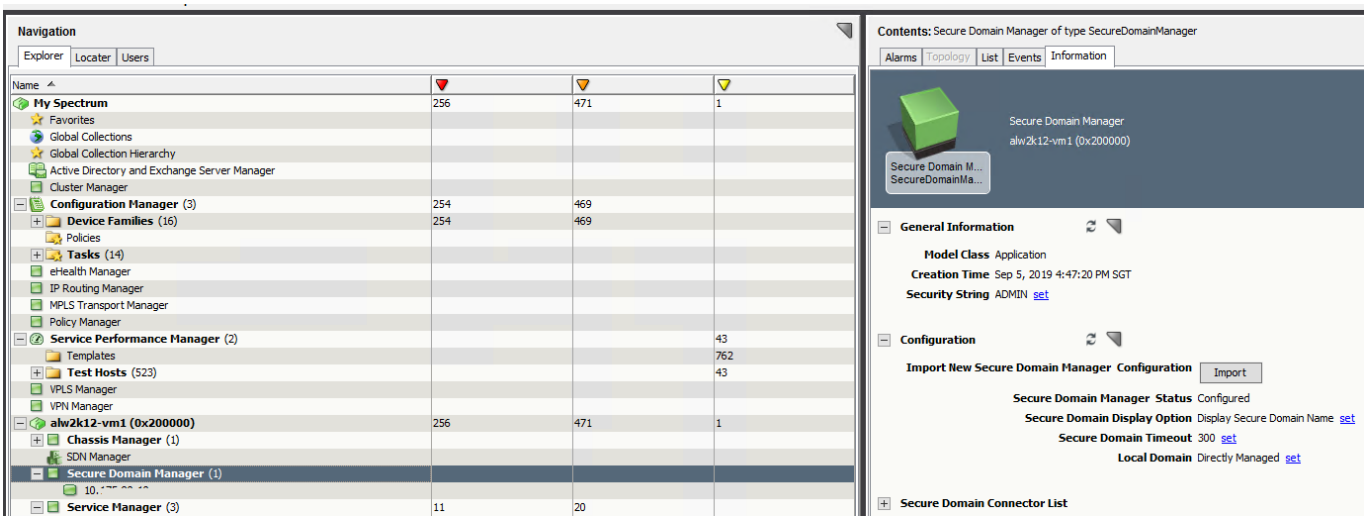
You can now configure the secure domain time-out setting based on your requirements. By default, the value is set to 300 seconds. You can specify any value between 60 seconds to 7200 seconds.

Follow these steps:

1. In the **Explorer** tab, navigate to **Secure Domain Manager**.
2. Click the **Information** tab in the right pane.
3. Expand the **Configuration** section.
4. Locate the **Secure Domain Timeout** option.
5. Click **set** and enter the required value.

You have successfully configured the time-out value.

The following screenshot shows the **Secure Domain Timeout** option:



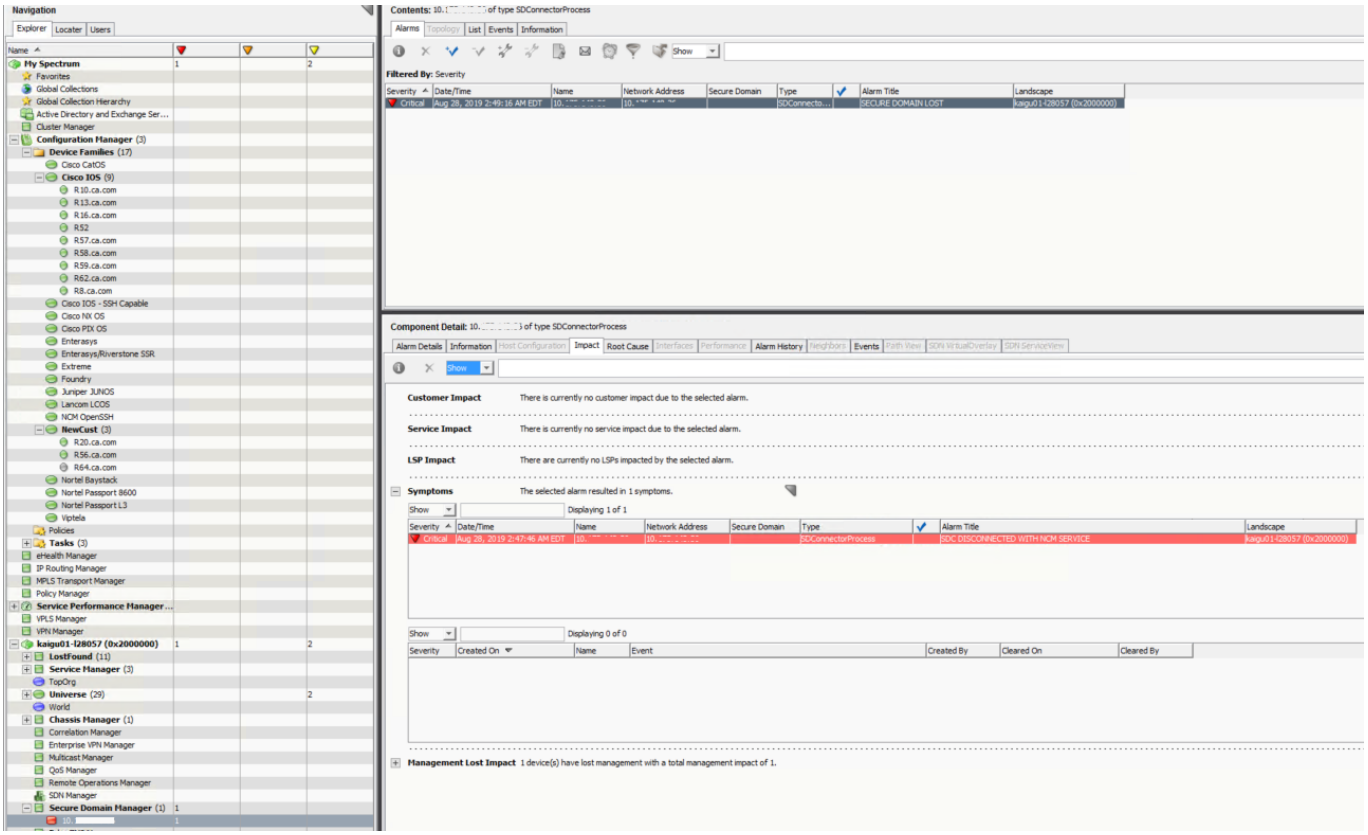
Scenario: NCM Down on SDC

If NCM is down on SDC, it creates a critical alarm on NCM down. In the case of the SDC service being down, it raises two critical alarms, where SDC down is the root cause and NCM down is the symptom.

Furthermore, if SDC is configured in a Fault-Tolerant (FT) environment, then the behavior is as follows:

- When NCM is down on the primary SDC, NCM on the secondary SDC starts working. But it will raise a major alarm on the DX NetOps Spectrum side stating that on the primary SDC, NCM is down and it has switched to the secondary SDC.
- If both NCMs are down, a critical alarm is raised.

The following screenshot shows the root cause as SDC and the symptom as NCM:



Telnet/FTP Support for SDC-Modeled Devices

In this release, for device families that support Telnet/FTP as a communication mode for SDC-modeled devices, you can perform the following NCM tasks on such devices:

- Capture
- Upload
- Sync
- Save to Startup

To use this functionality, you must provide the appropriate information in the **FTP Configuration** section.

NOTE

- This functionality is not supported on the SDC High Availability (HA) computers.
- Shared FTP and device-level FTP details are not supported.
- FTP server location must be in the same SDC system.
- Each SDC device must have its specific FTP server details.
- FTP server details must be provided in SDC.

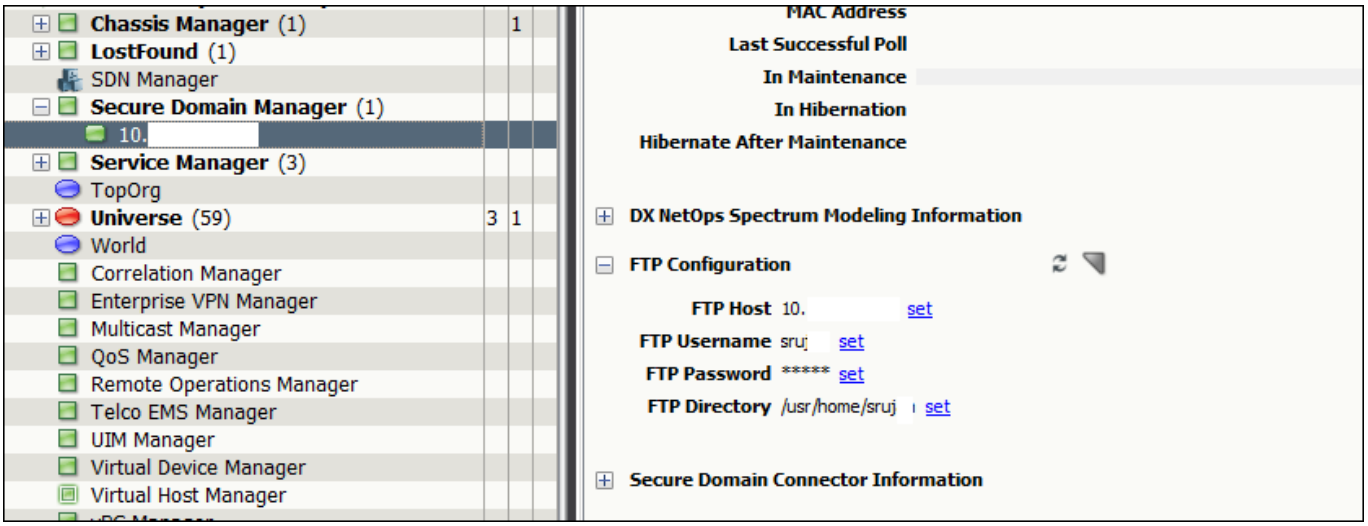
Follow these steps:

1. Navigate to **Secure Domain Manager** in the **Explorer** tab.
2. Select the required IP address under **Secure Domain Manager**.
3. Click the **Information** tab in the **Contents** pane.
4. Expand the **FTP Configuration** subview.
5. Provide information in the following fields; click set against each field, provide the value, and save it:

- **FTP Host**
  Specifies the FTP host IP address.
- **FTP Username**
  Specifies the FTP user name required to access the FTP host.
- **FTP Password**
  Specifies the password associated with the FTP user name.
- **FTP Directory**
  Specifies the FTP directory location.

6. Review the information.

You have successfully enabled the Telnet/FTP support for SDC-modeled devices. The following screenshot shows the required information:

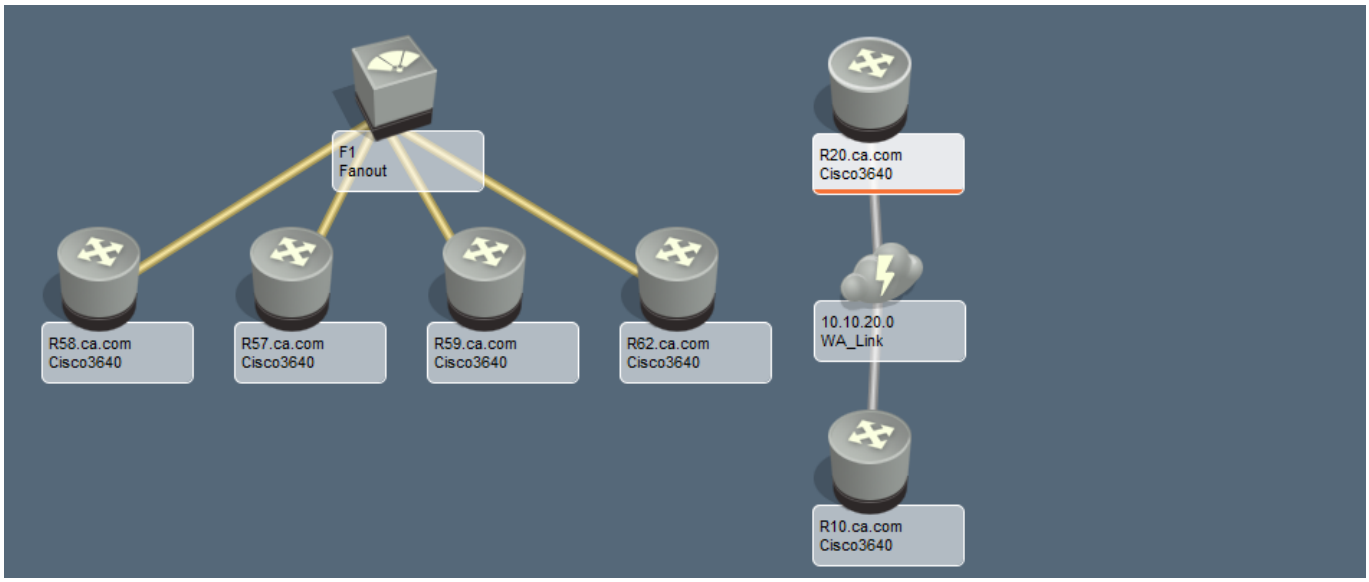


SDC Fault Isolation

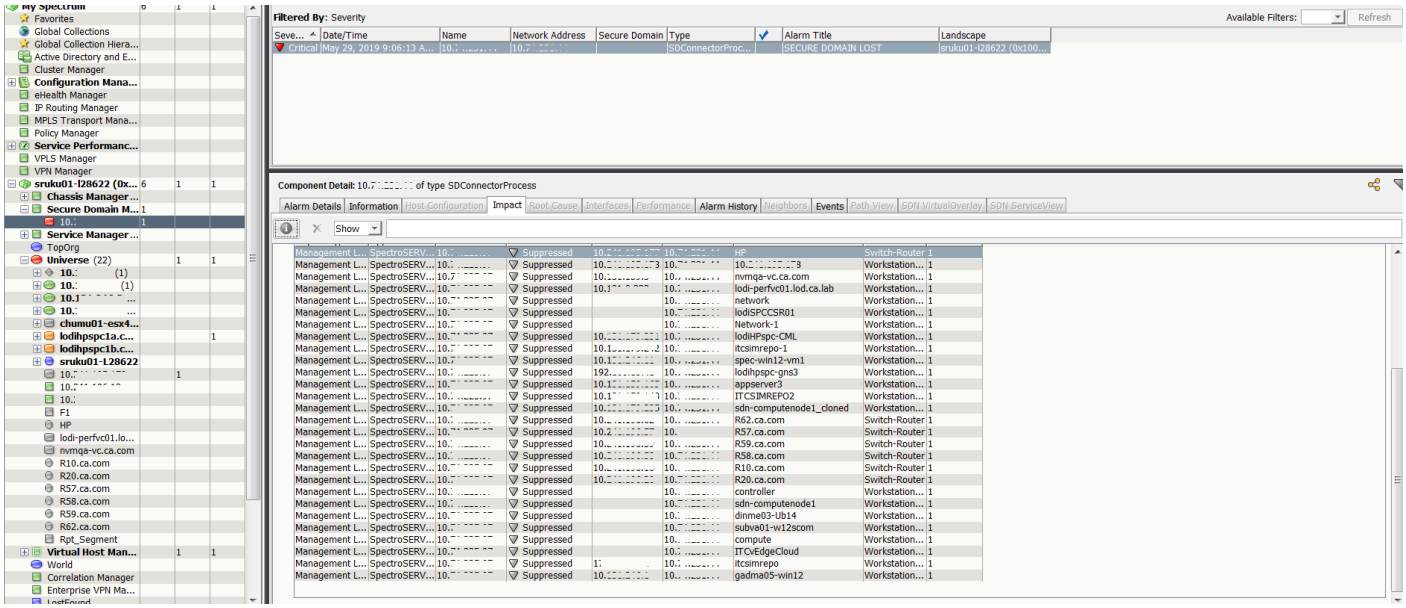
Secure Domain Connector (SDC) is used for managing devices. When SDC gets down, all the devices managed by SDC become unreachable. Previously, such devices would start generating individual alarms. Now, this behavior has changed in the 10.4.1 release. In this release, when SDC goes down, all devices modeled under SDC are suppressed and show the root cause as *SDC down*. Therefore, all related individual alarms are suppressed and users see only one alarm with the root cause as *SDC down*. It also helps users to efficiently manage devices and alarms.

The following information describes the behavior when SDC goes down.

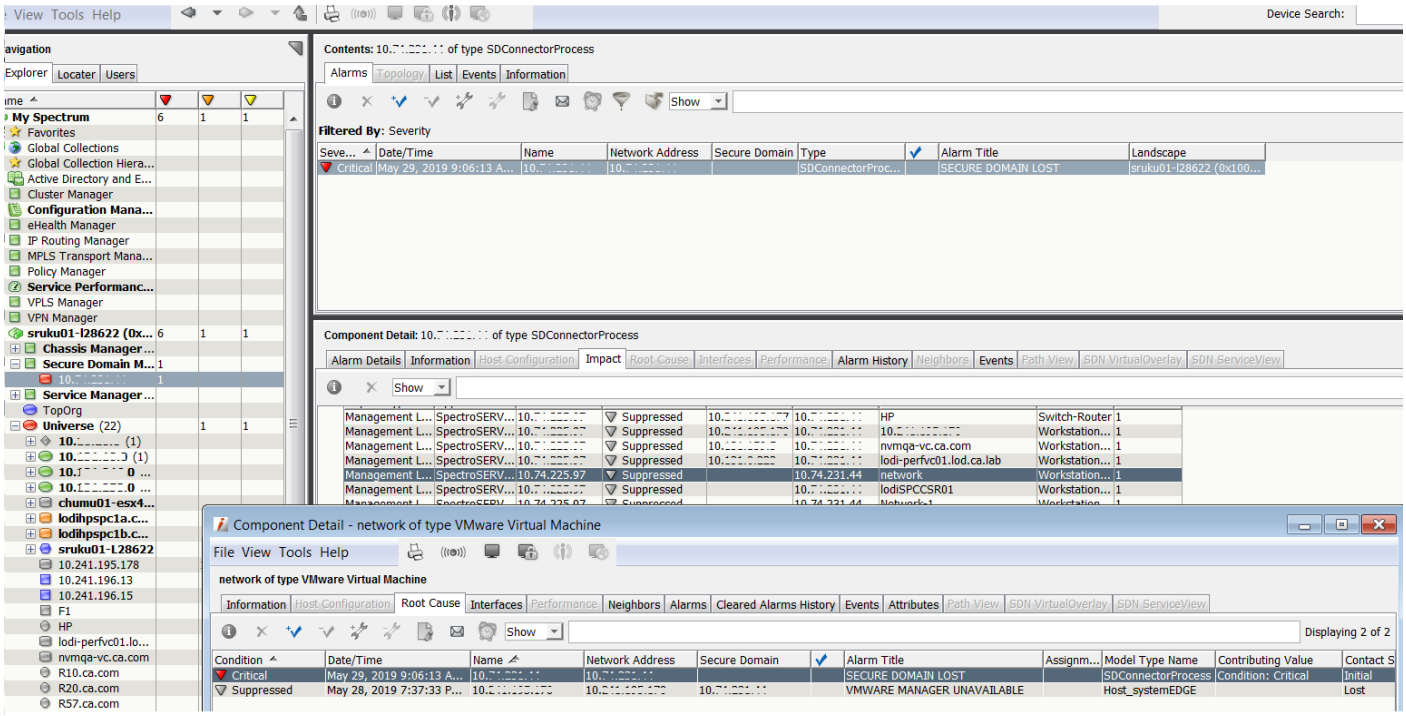
1. When SDC goes down, the SDC-modeled devices are suppressed. The following screenshot shows the required information:



In the SDC LOST alarm impact list, you can find a list of SDC-modeled devices that show the root cause as SDC down. The following screenshot shows the impact list:



The following screenshot shows the root cause:



2. If chassis down alarms are available on an SDC-managed device and SDC goes down, then those alarms on that device are also suppressed. However, when SDC comes up, SpectroSERVER again generates those alarms if the device is down.

   NOTE

   - If SDC-managed devices are in the maintenance mode and SDC goes down, then such devices are suppressed and are not shown in the impact list. And, if the device is removed from the maintenance mode, then it is shown in the impact list.
   - If a suppressed device is put in the maintenance mode and SDC goes down, then this device is not shown in the impact list.
   - Trap based alarms are not suppressed.
   - Alarms generated on System Resources (RFC 2790) on monitored processes and file systems are not suppressed.
   - It is not recommended to put virtual entities like Fanout and WA segment in maintenance mode.

Working with Secure Domain Manager

This chapter describes how to import the SDManager configuration file (sdm.config) into DX NetOps Spectrum and model SDConnector hosts and devices in secure domains. This chapter also describes OneClick tools that are use to locate Secure Domain Manager components. These components are used to ping devices in a secure domain. Ping the devices to view device MIBs, and to view information about the SDManager and SDConnector models.

Import the SDManager Configuration File

Import the sdm.config file into DX NetOps Spectrum before you can begin using OneClick with the Secure Domain Manager product and whenever you want to update the SDManager configuration. See Configure SDManager Process Settings for information about setting sdm.config parameters.

NOTE
You can import the SDManager configuration file before or after you create models for SDConnector hosts. If, however, you import an sdm.config file before you create models for SDConnector hosts, DX NetOps Spectrum automatically models the hosts as SDConnectorProcess model types. See Model SDConnector Hosts for more information about modeling options, including how to model SDConnectors as Pingable and Host_Device model types.

To import the SDManager configuration file

1. Click Secure Domain Manager in the Navigation panel in the OneClick Console.
2. Click the Information tab in the Component Detail panel and expand the Configuration subview.
3. Click Import.
   The Import Secure Domain Manager Configuration confirmation dialog opens.
4. Click Yes to confirm that you want to import the SDManager configuration file (sdm.config).
   The Import Secure Domain Manager Configuration dialog indicates whether the import started successfully. The dialog also provides information to check the output log to determine whether the import worked. The import log file in the SDM/Logs directory provides the troubleshooting information. This information is used to fix errors after an unsuccessful import.
5. Click OK.
   If the configuration file has been imported correctly, the Secure Domain Manager Status field displays "Configured." If an sdm.config file which contains no arguments to define how connections between SDManager and SDConnectors are established is imported, SDManager is disabled and the Secure Domain Manager Status field displays "Not Configured."

   NOTE
   If the sdm.config file is edited while the SpectroSERVER is not running, the SpectroSERVER automatically imports the new sdm.config file when it is started. You can verify whether the import was successful by checking the latest log file.

Model SDConnector Hosts

NOTE

When upgrading to 10.3.1 from 10.3 (directly) or 10.2.x (indirectly), if there were any alarms on the devices managed through Secure Domain Connector (SDC), those alarms will not be correlated to SDC lost alarm and only the newly generated alarms will be correlated, post upgrade.

Model SDConnector Host

Use the Model by Type option in the OneClick Topology view to model an SDConnector host computer as one of the three following model types:

- **SDConnectorProcess**
  The SDConnectorProcess model type is the default model type for SDConnectors. This model type does not allow you to manage the device status, but it does allow you to see the host computer that is represented in the OneClick Secure Domain Manager model hierarchy and provides access to the views discussed in SDConnector Model Information View.

  NOTE
  Use meaningful names for SDConnector host models that clearly identify the hosts. The model names appear in the Secure Domain Manager views in OneClick.
- **Host_Device**

Use the Host_Device model type if the host computer is running an SNMP agent.

- **Pingable**
  Use the Pingable model type if the host computer only supports ICMP.

If you use either the Host_Device or Pingable model type, you can monitor the status of the host computer. See SDConnector Modeling and DX NetOps Spectrum Fault Isolation for information about leveraging the DX NetOps Spectrum fault isolation capabilities by modeling SDConnector hosts as Host_Device or Pingable models.

Secure Domain Connector Status

In 10.2.2, you can ascertain the status of the SDC Host, if you have modeled the SDC host by any of the following methods:

- By importing from sdm.config
- By modeling SDC host as Host_Device model type
- By modeling SDC host as Pingable model type

As soon as the connection with the SDC host is lost, an alarm stating "Lost contact with SDC" is generated on whichever model type is used to model the SDC host. The alarm is asserted based on the attribute value returned for the following attributes which have been added to the SDC Host model:

SDConnectorStatus (0x12ad7) - Attribute value 0= Lost connection, and attribute value 1 = Established connection

SDConnectorStatusUpdateInterval (0x13381) - Default value = 30 seconds; Polls the SDC host model for connection status at an interval of 30 seconds.

SDConnector Modeling Considerations

By default, DX NetOps Spectrum automatically models an SDConnector host computer as an SDConnectorProcess model type if you have not created models for the computer before you initially import the SDManager configuration file.

If you prefer to model hosts as Pingables, or Host_Devices, model the hosts as your preferred type before the import. Alternately, destroy the SDConnectorProcess models after the import. Then, model the hosts as Pingables, or Host_Devices.

> **NOTE**
> If you use the Model By IP option to create a model representing the SDConnector host without first destroying the existing SDConnectorProcess model, DX NetOps Spectrum copies and pastes the SDConnectorProcess model into the topology view from which the Model By IP option was invoked.

Destroying the SDConnector host model in OneClick does not prevent the DX NetOps Spectrum from using the actual SDConnector for the device communication. The SDConnector can only be destroyed by reimporting the SDManager configuration (after editing the sdm.config file to remove the SDConnector).

If you accidentally destroy an SDConnectorProcess model, DX NetOps Spectrum recreates the model the next time you import the SDManager configuration file. If you destroy a Pingable or Host_Device model, DX NetOps Spectrum creates an SDConnectorProcess model the next time you import the SDManager configuration file. If you want to restore your Pingable or Host_Device model, explicitly recreate the model and then import the configuration file.

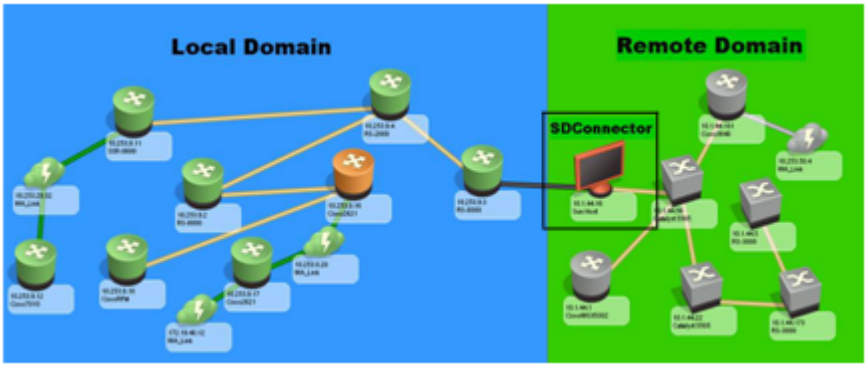
SDConnector Modeling and Fault Isolation

As described in Model SDConnector Hosts, when you model SDConnectors, you can choose one of the following model types:

- SDConnectorProcess
- Host_Device
- Pingable

We recommend modeling the SDConnector host as a Host_Device or Pingable model type. These model types let DX NetOps Spectrum fault isolation work correctly when a remote SDConnector process goes down, or loses its connection. DX NetOps Spectrum isolates the cause of an outage to the SDConnector host model, virtually eliminating unresolved fault alarms.

The SDConnector host is usually connected to a switch on the edge of a network. However, logically, it is the bridge between the public domain and secure domain regions. Model it accordingly. Place the SDConnector host model between the two models for the devices that are routing traffic between the public domain and secure domain regions. The following diagram illustrates this connection, showing the SDConnector as a Host_Device model.



Model Devices in Secure Network Domains

After you model an SDConnector host, model the network devices that you want to manage in the secure domain where the SDConnector host is located. Model the network devices one at a time using the OneClick Create Model By IP option, or Discovery. You can place the models anywhere in the Topology view. After you have successfully created models, DX NetOps Spectrum can communicate with them using the SDConnector process.

Create Model by IP

Use the OneClick Create Model By IP option to model each device in a secure domain.

> **NOTE**
> For more information about modeling in OneClick, see the Modeling and Managing Your IT Infrastructure section.

To model a device in a secure domain using the Model by IP option

1. Click the Model by IP option in the Topology view.
   The 'Create Model by IP Address' dialog opens.
2. Type the network address of the device you want to model in the Network Address field.
3. From the Secure Domain drop-down list, select the IP address of the host running the SDConnector. You also have the choice to select the name that has been configured for the SDConnector host in the secure domain where the device you are modeling is located.

   > **NOTE**
   > You can provide a secure domain name for the SDConnector host by changing the host model name in OneClick. See SDManager Model Information View for information about enabling the secure domain name as a selection option.
4. Select the SNMP version compatible with the device you want to manage in the SNMP Communications Options section.
5. Click OK.

NOTE

When you use the Modeling Gateway with devices located in a Secure Domain, you need to add the attribute details of the secure domain to the import to allow the Modeling Gateway to find models in question and discover the secure domain. Normally, devices can be found with just the device IP.

Example: (Device ip_dnsname="x.x.x.x" secdomain_ipname="y.y.y.y")"

OneClick Discovery

Use OneClick Discovery to discover and model all devices in a secure domain with an SDConnector host. Keep in mind the following points while discovering devices with overlapping IP addresses:

- Only one SDConnector can be used for each Discovery.
- Although you can use Layer 2 mapping, its effectiveness is dependent upon the accuracy of the Source Address and the Spanning Tree tables.
- The Protocol Options settings:
  - Do not use Layer 3 Autodiscovery mapping. Deselect IP Address Tables and IP Route Tables in the Protocol Options dialog.
  - Do not use Proprietary Discovery Protocols in Cisco, or the Nortel environments because they use IP addresses to convey neighbor relationships. Deselect Proprietary Discovery Tables in the Protocol Options dialog.
  - Do not use the Pingable mapping. Deselect ARP tables for Pingables in the Protocol Options dialog.

Discover Devices Using an SDConnector Host

Follow these steps:

1. Click Tools, Utilities, Discovery Console from the main menu.
   The Discovery Console opens.
2. Complete the Discovery configuration for the secure domain from which you want to model devices.

   NOTE
   For more information about configuring Discovery, see the Modeling and Managing Your IT Infrastructure section.
3. Click Advanced Options in the Configuration tab.
   The Advanced Options dialog opens.
4. In the Discovery Options section, from the Secure Domain drop-down list, select the IP address of the host running the SDConnector in this secure domain or the name that is specified for the secure domain.

   NOTE
   You can provide a secure domain name for the SDConnector host by changing the host model name in OneClick. See SDManager Model Information View for information about enabling the secure domain name to be a selection option.
5. Click OK.
   The Advanced Options dialog closes and your changes are saved.
6. Click Discover in the Discovery Console.
   The Discovery that you configured runs. After Discovery, view all the devices that are listed in the Secure Domain Connector Device Table of its corresponding SDConnector host icon.

   NOTE
   If you have already modeled the host machine running a remote SDConnector process using the SDConnectorProcess model and you perform a Discovery on the network region where the host exists, Discovery may create an additional model of the host using the Host_Device, or the Pingable model. Delete

this duplicate model once it has been created or you can filter this model out of the Discovery result that is set before it is created.About Maintaining Device Secure Domain Membership

About Maintaining Device Secure Domain Membership

In a NAT environment, multiple SDConnectors are used to manage the same IP ranges. When the duplicate IP ranges exist, DX NetOps Spectrum cannot determine the SDConnector that must manage each device. So, specify this information.

When discovering or modeling new devices in DX NetOps Spectrum, you can set the secure domain using the OneClick Model by IP view or OneClick Discovery. To update the secure domain for an existing device model, use the OneClick Attribute Editor to edit the Secure Domain Address attribute. This automatically updates the Secure Domain Name. When a new SDManager configuration file (sdm.config) is imported into DX NetOps Spectrum, any existing devices that were assigned to an old secure domain will still be assigned to it. Red alarms will likely be generated on these models.

Create Model by IP

Use the OneClick Create Model By IP option to model each device in a secure domain.

NOTE
For more information about modeling in OneClick, see the Modeling and Managing Your IT Infrastructure section.

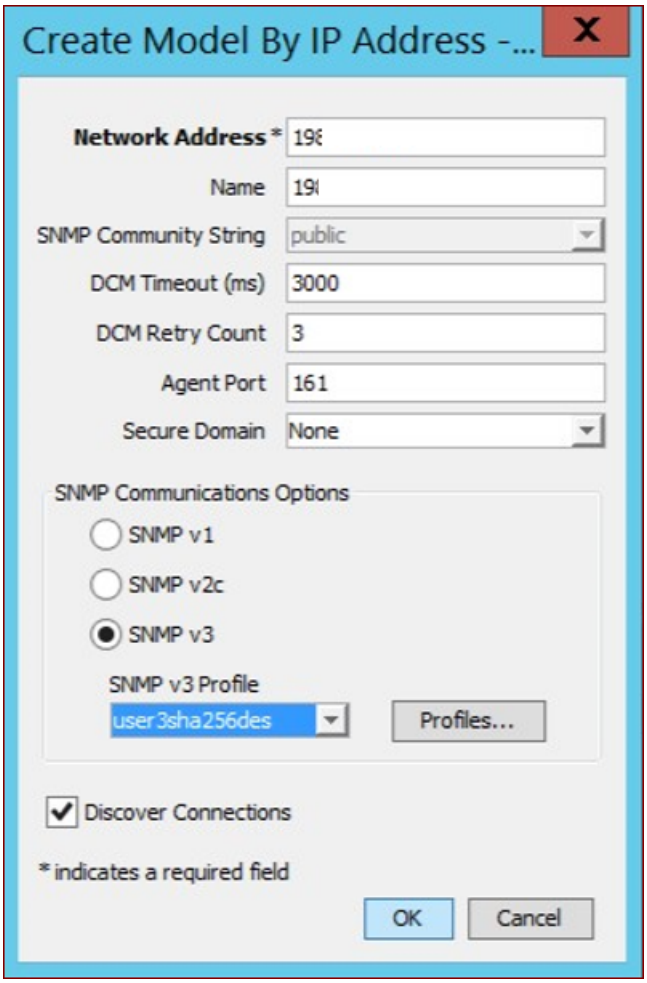
To model a device in a secure domain using the Model by IP option

1. Click the Model by IP option in the Topology view.
   The 'Create Model by IP Address' dialog opens.
2. Type the network address of the device you want to model in the Network Address field.
3. From the Secure Domain drop-down list, select the IP address of the host running the SDConnector. You also have the choice to select the name that has been configured for the SDConnector host in the secure domain where the device you are modeling is located.

   NOTE
   You can provide a secure domain name for the SDConnector host by changing the host model name in OneClick. See SDManager Model Information View for information about enabling the secure domain name as a selection option.
4. Select the SNMP version compatible with the device you want to manage in the SNMP Communications Options section. The SNMP v3 Profile drop-down is enabled only when you select the SNMP v3 option.

   NOTE
   In 10.4.2, the SNMPv3 profile creation supports SHA-256 and SHA-512 as authentication protocols. Therefore, you can now select those profiles, too.
5. Click OK.

The following screenshot shows the required information:



OneClick Discovery

Use OneClick Discovery to discover and model all devices in a secure domain with an SDConnector host. Keep in mind the following points while discovering devices with overlapping IP addresses:

- Only one SDConnector can be used for each Discovery.
- Although you can use Layer 2 mapping, its effectiveness is dependent upon the accuracy of the Source Address and the Spanning Tree tables.
- The Protocol Options settings:
  - Do not use Layer 3 Autodiscovery mapping. Deselect IP Address Tables and IP Route Tables in the Protocol Options dialog.
  - Do not use Proprietary Discovery Protocols in Cisco, or the Nortel environments because they use IP addresses to convey neighbor relationships. Deselect Proprietary Discovery Tables in the Protocol Options dialog.
  - Do not use the Pingable mapping. Deselect ARP tables for Pingables in the Protocol Options dialog.

Discover Devices Using an SDConnector Host

Follow these steps:

1. Click Tools, Utilities, Discovery Console from the main menu.
   The Discovery Console opens.

2. Complete the Discovery configuration for the secure domain from which you want to model devices.

   NOTE
   For more information about configuring Discovery, see the Modeling and Managing Your IT Infrastructure section.
3. Click Advanced Options in the Configuration tab.
   The Advanced Options dialog opens.
4. In the Discovery Options section, from the Secure Domain drop-down list, select the IP address of the host running the SDConnector in this secure domain or the name that is specified for the secure domain.

   NOTE
   You can provide a secure domain name for the SDConnector host by changing the host model name in OneClick. See SDManager Model Information View for information about enabling the secure domain name to be a selection option.
5. Click OK.
   The Advanced Options dialog closes and your changes are saved.
6. Click Discover in the Discovery Console.
   The Discovery that you configured runs. After Discovery, view all the devices that are listed in the Secure Domain Connector Device Table of its corresponding SDConnector host icon.

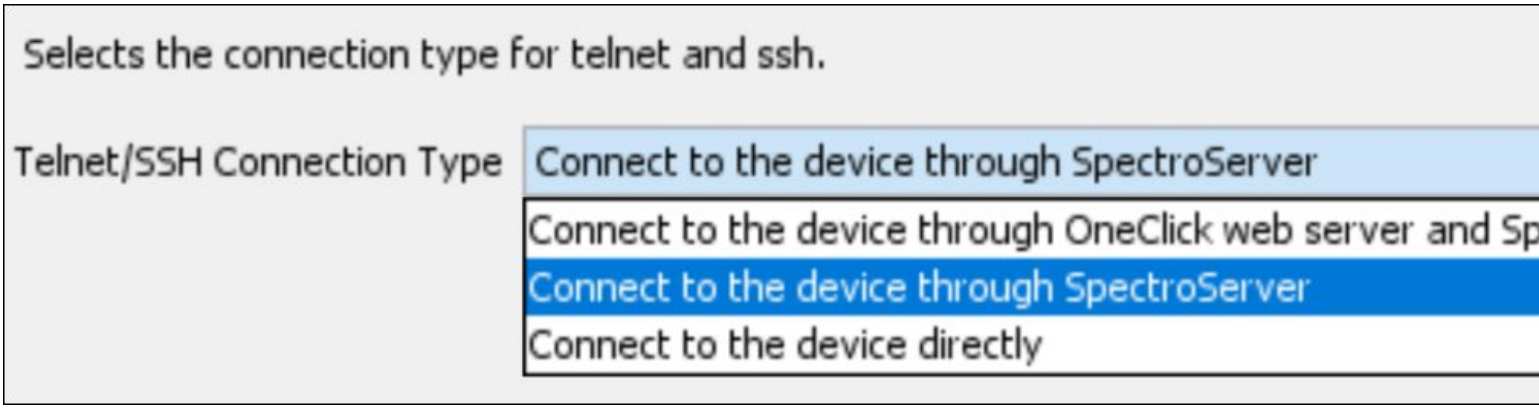
   NOTE
   If you have already modeled the host machine running a remote SDConnector process using the SDConnectorProcess model and you perform a Discovery on the network region where the host exists, Discovery may create an additional model of the host using the Host_Device, or the Pingable model. Delete this duplicate model once it has been created or you can filter this model out of the Discovery result that is set before it is created.

Telnet/SSH Connection Type

From 10.4.2.1, you can communicate from OneCLick console to devices managed through SDC using the telnet and SSH. Select **View, Preferences, Topology Tab, Telnet/SSH connection Type** to access the option.

Specifies the connection type that must be used to access the device from the OC Console for telnet and SSH. Select one of the following options:

- Connect to the device through OneClick web server and SpectroServer (**Default**)
  The communication goes from the OneClick Console workstation to the OC Tomcat server and is then routed to the SpectroSERVER and then to SDC machine which then accesses the remote device.
- Connect to the device through SpectroServer
  This connection method works when the OneClick console workstation is capable to connect to the SDC machine.
- Connect to the device directly
  Access the device from the OneClick console workstation directly.



About Maintaining Device Secure Domain Membership

In a NAT environment, multiple SDConnectors are used to manage the same IP ranges. When the duplicate IP ranges exist, DX NetOps Spectrum cannot determine the SDConnector that must manage each device. So, specify this information.

When discovering or modeling new devices in DX NetOps Spectrum, you can set the secure domain using the OneClick Model by IP view or OneClick Discovery. To update the secure domain for an existing device model, use the OneClick Attribute Editor to edit the Secure Domain Address attribute. This automatically updates the Secure Domain Name. When a new SDManager configuration file (sdm.config) is imported into DX NetOps Spectrum, any existing devices that were assigned to an old secure domain will still be assigned to it. Red alarms will likely be generated on these models.

Access Secure Domain Manager Searches

OneClick includes various predefined Secure Domain Manager search options.

To access Secure Domain Manager search options, expand the Secure Domain Manager folder in the Locater tab in the OneClick Console.

The predefined Secure Domain Manager searches available to you are displayed.

Check Device Accessibility in a Secure Domain

Determine whether the devices are accessible by using the OneClick Ping menu option to ping devices that are located in a secure domain.

> **NOTE**
> A successful ping does not display the number of bytes returned by the pinged device in a secure domain.

To check the device accessibility in a secure domain, right-click the device for which you want to assess accessibility in the OneClick Console and click Ping.

The Ping dialog opens, listing the results of the ping request. For example:

```
Secure reply from 10.254.1.5: icmp_seq=4. time =140. ms
```

If this device was not in a secure domain, the result would appear as follows:

```
64 bytes from 10.254.1.5: icmp_seq=4. time =140. ms
```

View a Device MIB in a Secure Domain

View a device MIB in a secure domain with MIB Tools. First, specify the SDConnector for the secure domain where the device is located. The following procedure describes how to specify an SDConnector.

> **NOTE**
> For more information about using MIB Tools, see the Certifications section.

Follow these steps:

1. Select the device that you want to investigate with MIB Tools.
2. Right-click the device and select Utilities, MIB Tools.
   MIB Tools open. The Contact Criteria is prepopulated with the selected SNMP contact information of the device. MIB Tools attempts to contact the device.
   If MIB Tools *cannot* contact the device an error message appears and the Contact Status indicator turns red.
   If MIB Tools *can* contact the device, the Contact Status indicator turns green.
   A status dialog also appears which shows the progress of retrieving and loading the MIB Tools database.

3. Click Advanced Options in the Contact Criteria section.
   The MIB Tools: Advanced Options dialog appears.
4. Select the applicable secure domain from the Secure Domain drop-down list.
5. Click OK.
   The Advanced Options dialog closes and your changes are saved.
6. Click Contact in the Contact Criteria section and verify that MIB Tools can contact the device successfully.
7. Close MIB Tools.
   MIB Tools closes and you have specified the SDConnector for the device.

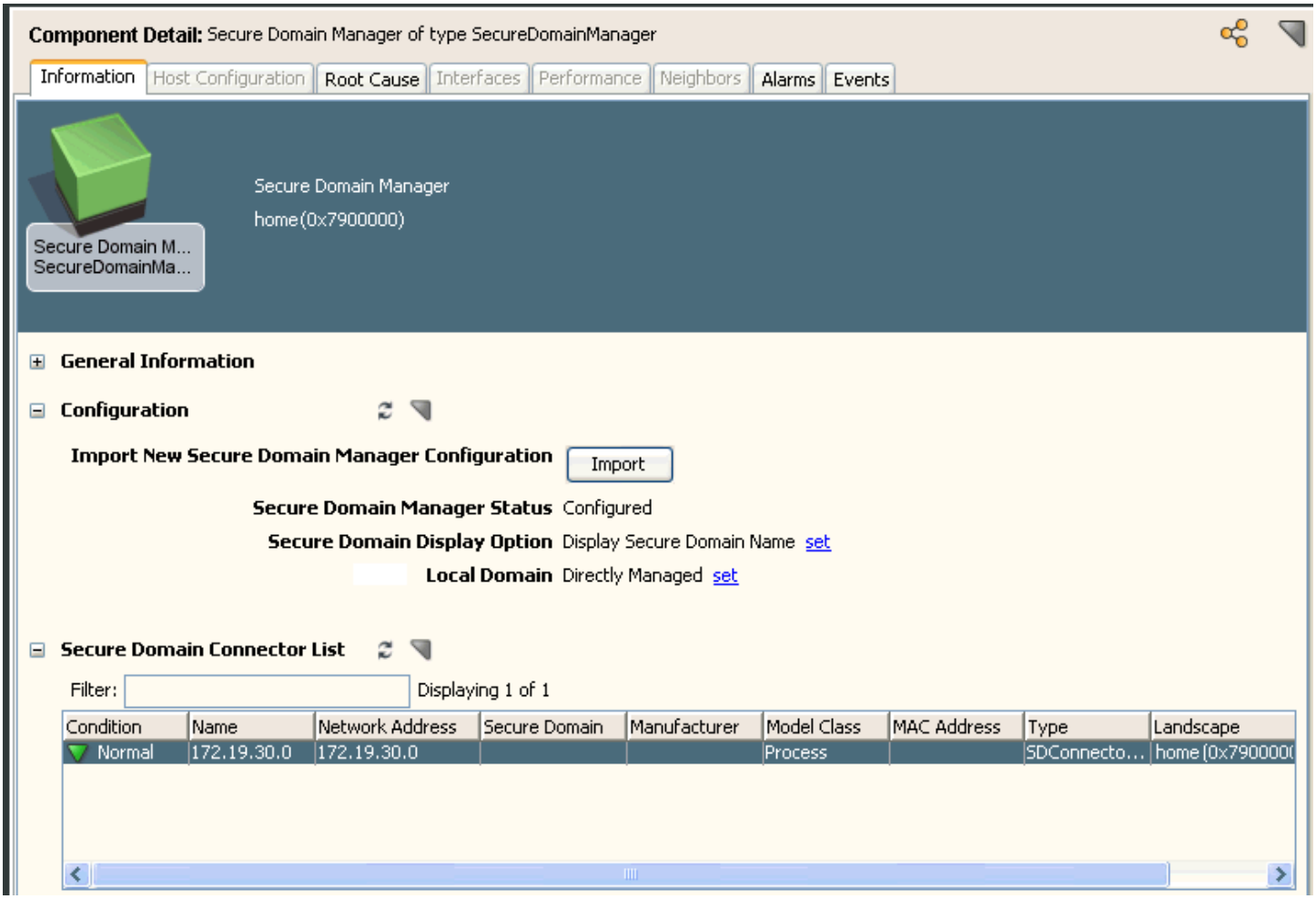
SDManager Model Information View

The Information tab in the Component Detail panel provides information about and configuration controls for the selected SDManager model in the following sections:

- **General Information**
  The General Information section provides standard information about the Secure Domain Manager model such as its model class and security string.
- **Configuration**
  The Configuration section includes the following content:
  - **Import**
    Imports the SDManager configuration file (sdm.config) into DX NetOps Spectrum.
  - **Secure Domain Manager Status**
    Indicates the configuration status of the SDManager as follows:
    - **Configured:** Indicates that the file has been successfully imported.
    - **Not Configured:** Indicates either that a custom or edited sdm.config file has never been imported, an sdm.config file with no arguments has been imported, or an sdm.config file that contains errors has been imported.
  - **Secure Domain Display Option**
    Specifies whether DX NetOps Spectrum displays the name that is used to identify the SDConnector host (and its domain) or the SDConnector host IP address. You can choose either "Display Secure Domain Name" or "Display Secure Domain Address" from the drop-down list. This determines which the type of SDConnector identifier is used throughout all OneClick views.
  - **Local Domain**
    Specifies the text that appears in the Secure Domain column for locally managed models (models that are not included in a secure domain).
    Default: Directly Managed

    NOTE
    The Secure Domain column appears in OneClick list views only when Secure Domain Manager is installed.
- **Secure Domain Connector List**
  Displays all the host machines currently running SDConnector processes in remote network regions.

The following image shows an example of the Component Detail panel for a selected SDManager model:



SDConnector Model Information View

The Information tab that is located in the Component Detail panel provides information about the selected SDConnector. The General Information and the SPECTRUM Modeling Information categories provide standard information about the SDConnector model. There is also a Secure Domain Connector section that includes the following subsection:

Secure Domain Connector Device Table
The Secure Domain Connector Device Table lists all the devices that are managed by the selected SDConnector. It also lets you print, export, and filter the list of devices. You can click the Name hyperlink of a device in this list to navigate directly to that device in the Topology view.

Setting Up Processes in a Fault-Tolerant Environment

This chapter describes how to set up SDConnectors to connect to SDManagers on primary and backup SpectroSERVERs in a fault-tolerant SpectroSERVER environment. This chapter also describes how to set up primary and backup SDConnectors.

Set Up SDManager in a Fault-Tolerant SpectroSERVER Environment

In a fault-tolerant SpectroSERVER environment, install the SDManager on both the primary SpectroSERVER and the backup SpectroSERVER. Each SDConnector that communicates with this SDManager is configured to connect to the primary and backup SpectroSERVERS. If the primary SpectroSERVER fails, the backup SpectroSERVER takes over communications with each SDConnector.

Follow these steps:

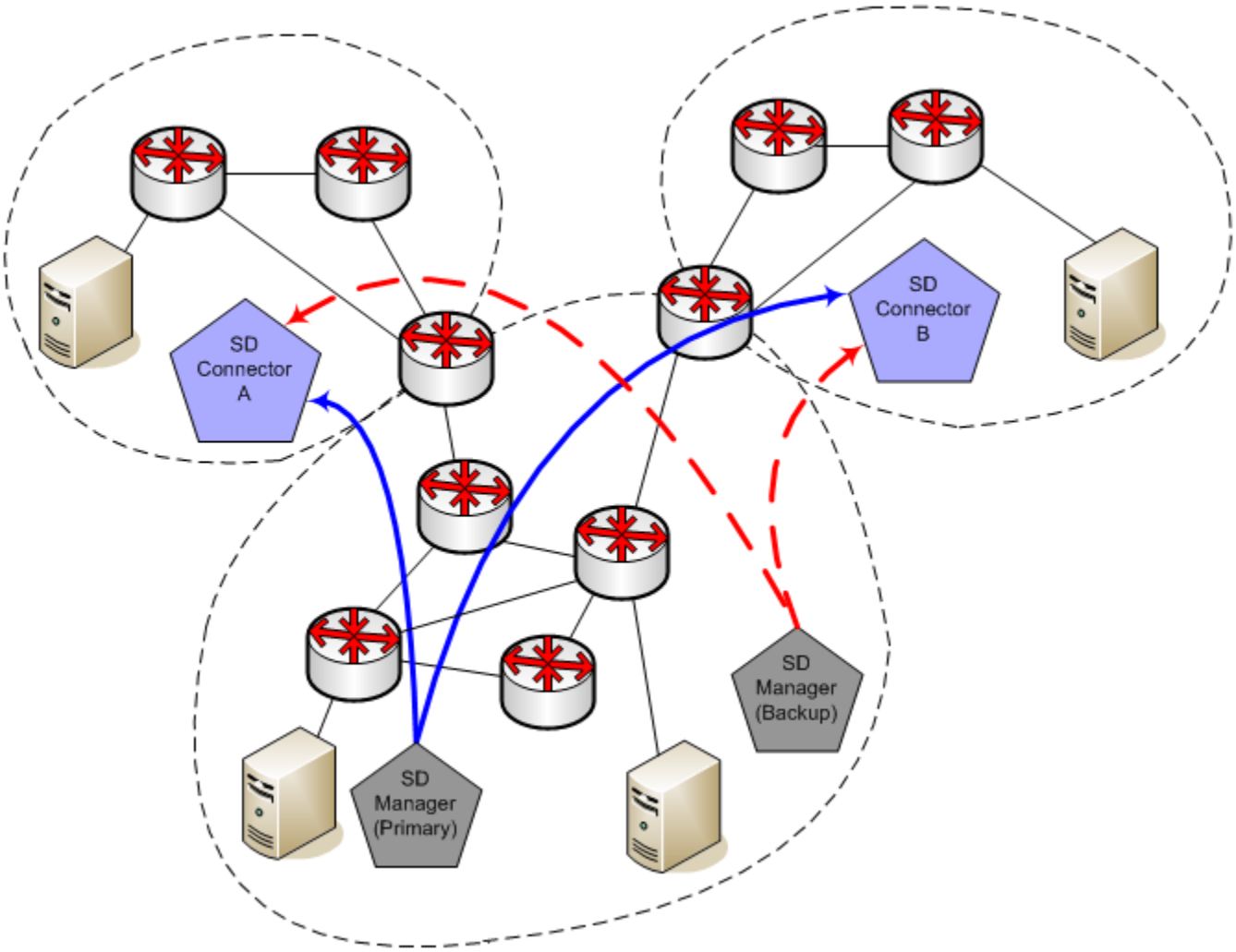
1. Deploy an SDConnector on each secure domain that you want to manage.

   NOTE
   See Installing and Configuring Secure Domain Manager Processes for detailed instructions about how to deploy SDConnectors.
2. Configure each SDConnector to accept connections from both the primary SpectroSERVER and the backup SpectroSERVER. For example, 172.24.1.2 and 172.24.3.4 respectively:

   ```
   -accept 172.24.1.2 -accept 172.24.3.4
   ```

Fault-Tolerant SpectroSERVERs (SDManagers)

The following diagram shows how two SDConnectors can be connected to both a primary SDManager and a backup SDManager:



Configuration setting for both SDManagers in the sdm.config:

```
-remoteconnect <IP of SDConnector A> -remoteconnect <IP of SDConnector B>
```

Configuration setting for both SDConnectors in the sdc.config:

```
-accept <IP of primary SDManager> -accept <IP of backup SDManager>
```

Set Up Fault-Tolerant SDConnectors

Secure Domain Manager supports the Backup functionality on a per-SDConnector basis. A backup SDConnector must be able to manage all the devices the primary SDConnector does, not merely a subset of them.

When you import a backup configuration into DX NetOps Spectrum, the backup SDConnector is not automatically modeled. If the primary SDConnector goes down, the backup functionality takes over *transparently*. There is no visible indication that the primary SDConnector is down. Also, because backups are not modeled they do not appear in the OneClick Console Model by IP or Discovery Configuration views or in MIB Tools.

Follow these steps:

1. Deploy both a primary and a backup SDConnector for each remote domain that you want to manage.

NOTE
See Installing and Configuring Secure Domain Manager Processes for detailed instructions about how to deploy SDConnectors.

2. Configure the SDManager to connect to the primary and backup SDConnectors by modifying the sdm.config file as shown in the following example:

```
-remoteconnect <IP of primary SDC> -remotebackup <IP of backup SDC>
```

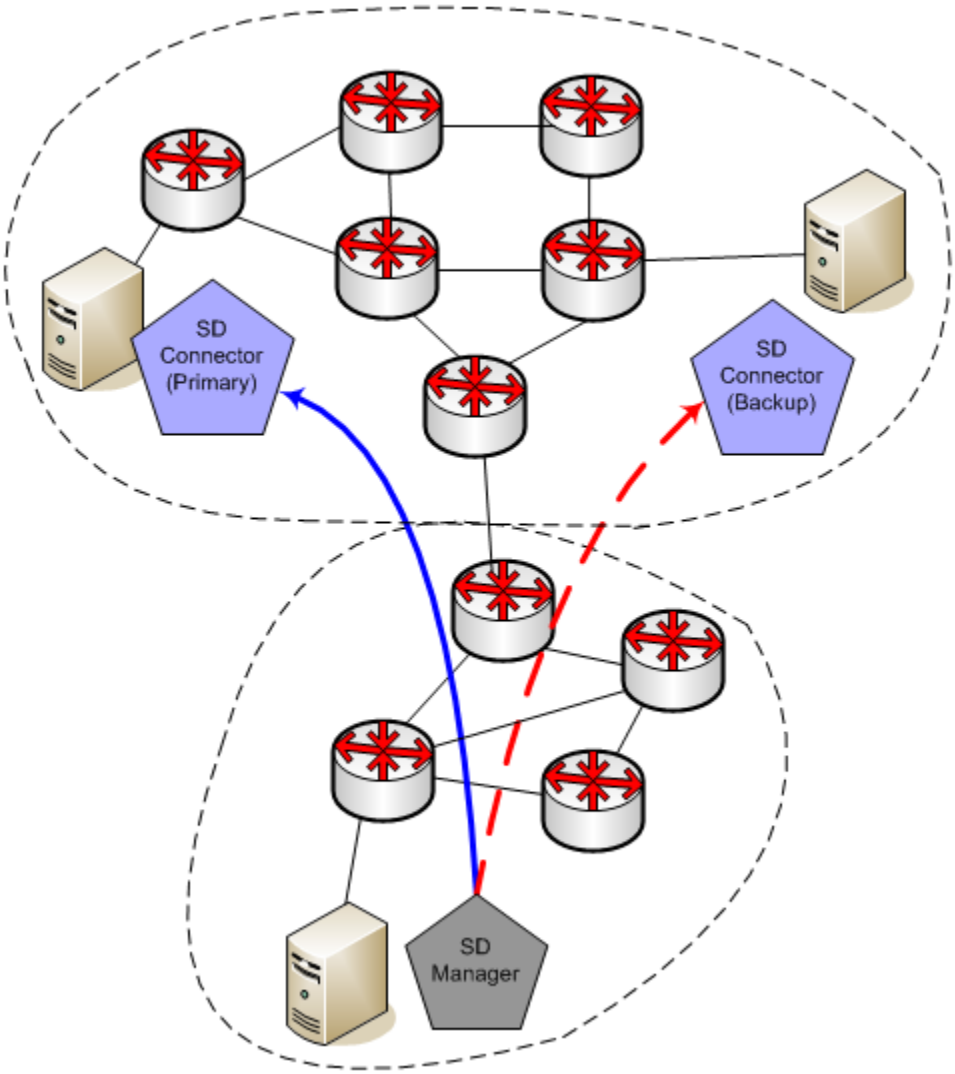
NOTE
Both primary and secondary SDCs always use different IP addresses in the sdm.config file. An example is as follows:

```
-remoteconnect 10.10.10.10 -remotebackup 20.20.20.20
```

Fault-Tolerant SDConnectors

Setting up Fault-Tolerant Secure Domain Connectors

The following diagram depicts two SDConnectors connected to a single SDManager:



Configuration setting for the SDManager in the sdm.config:

```
-remoteconnect <IP of primary SDConnector> -remotebackup <IP of backup SDConnector>
```

Configuration setting for both SDConnectors in the sdc.config:

```
-accept <IP of SDManager>
```

Secure Domain Connector (SDC) Status

From 10.2.2, you can view the status of Secure Domain Connectors (including FT/ Primary and Backup SDConnector). When the connection with SDC is lost or the SDC goes down, appropriate alarms are generated in DX NetOps Spectrum.

The SecureDomainStatus (**0x12ad7**) attribute has the following states indicating the status of the Secure Domain:

- **SecureDomainStatus** attribute Value = **1; Status:** Established
- **SecureDomainStatus** attribute Value = **2; Status:** Established, Primary SDC down
  Event: Secure Domain Connector established to secondary, but primary is down.
  Alarm Title: SECURE DOMAIN SWITCHED TO BACKUP
- **SecureDomainStatus attribute Value = 3; Status:** Established, Secondary/ Backup SDC down
  Event: Secure Domain connector established to primary, but secondary is down;
  Alarm Title: SECURE DOMAIN BACKUP IS DOWN
- **SecureDomainStatus attribute value = 4; Status: Lost**
  Event: Secure Domain connector connection lost;
  Alarm Title: SECURE DOMAIN LOST

Troubleshooting Secure Domain Manager

This section describes some potential Secure Domain Manager problems and their solutions.

Error Messages

This section provides information about Secure Domain Manager error messages. SDManager errors appear in the SDManager.out file; SDConnector errors appear on the terminal display.

Certificate Not Valid Error

Valid on Linux and Windows

Symptom:

The following SDConnector error message appears when a mismatch is found in certificates or security settings:

```
SdmEtpkiConnectEndpoint run() invalid socket security. No connection attempts will be made to the host.
Please verify that certificates and security configurations are correct.
```

Solution:

Verify that machines on which SSL is deployed have matching certificates.

Certificate or Key Error

Valid on Linux and Windows

Symptom:

The following error message appears in sdmLog.log file when a mismatch is found in a certificate or key across connector host and SpectroSERVER host:

```
SdmEtpkiConnectEndpoint run() ssock_handshake error.
```

Solution:

This error indicates a problem with the certificate/key used for SDM/SDC communication. If you are using certificates that are generated with earlier versions of Spectrum or using custom certificates, then you must regenerate the certificates using CertGen utility. If you are using your own custom certificates (not generated using CertGen), then you must ensure that the key size is minimum 1024-bits. The key of less than 1024-bit size does not work with Spectrum 9.4.3. Refer to Create Site-Specific Certificates and Deploy Site-Specific Certificates sections for the steps that are required to create and deploy the custom Site-Specific Certificates.

Port Conflicts

SDConnector Requires a Custom SNMP Trap Port

Valid on Linux and Windows

If there exists a need to change the trap port that the SDConnector listens for the SNMP traps on, configure a custom listening port.

> **NOTE**
> In the following procedure, port 951 is used as an example of a new custom listening port.

Follow these steps:

1. Configure SDConnector to listen for traps on a custom port by modifying the sdc.rc file as follows:
   ```
   snmp_trap_port = 951
   ```
2. Restart SDConnector process by rebooting the computer.
   The SDConnector now listens traps on port 951.

Installation Issues

When on some Windows installations the SDConnector service is not installed. Or, when it is installed, but not started. Then, manually install SDConnector service on Windows.

Follow these steps:

1. From a command prompt, navigate to the following folder:
   ```
   <SDC Install directory>/bin
   ```
2. Run the following command:
   ```
   SdmConnectorService.exe --install
   ```
3. Start the service from the Services window or run the following command:
   ```
   SdmConnectorService.exe --start
   ```

Service Performance Manager

Service Performance Manager Concepts

Service Performance Manager (SPM) lets you create, run, and manage performance tests that are supported by third-party products. The products from various vendors that DX NetOps Spectrum manages perform their own testing to address multiple network management requirements. The following goals are examples of the types of testing that are supported:

- Testing IT service delivery standards -- You can simulate transactions, such as HTTP transactions, login validations, or file transfers to establish delivery benchmarks. You can then measure service delivery to consumers and develop realistic service-level agreements.
- Capacity planning -- Run tests to determine whether service demands by consumers are underutilizing or exceeding current IT infrastructure capacity.
- Proactive fault management -- You can pinpoint service delivery degradation trends before they impact service consumers.

The topics in this section describe the Service Performance Manager components that let you configure and run performance tests.

Test Hosts

A *test host* is a model of a device or software agent that supports one or more performance tests for IT services. DX NetOps Spectrum creates a test host model of type RTM_TestHost during model discovery for each device that Service Performance Manager supports.

Some examples of test host devices and agents are a Cisco router, a Sun Workstation running iAgent, and a SpectroSERVER.

> **WARNING**
> Supported devices and agents must be modeled with read/write community strings to run Service Performance Manager tests. Attempts to run tests on test hosts (except for SpectroSERVERs) that are not modeled with a read/write community string cause an alarm on the test host model.

Configure the Application Model

You can configure the community string of the application model that Service Performance Manager uses. For example, you want to create the device model with a read-only community string. In such a case, you can configure the community string of the Service Performance Manager application model.

> **NOTE**
> Performance Agents and MIBs list tests that are supported by the performance agents and the associated DX NetOps Spectrum application model type. For more information, see Performance Agents and MIBs.

Follow these steps:

1. Click the Locater tab.
2. Expand the Application Models folder.
3. Double-click the By Device IP Address field, and enter the device IP address.
4. Filter and select the appropriate application model.
   For example, for Cisco IP SLA supporting hosts, enter CiscoRTTMonApp.
5. Click the Attributes tab.
6. Filter for the Community_Name.
7. Select Community_Name/0x10024 and select the right arrow.
   The attribute appears in the right panel.
8. Double-click to select and change the value.
   The community string of the application model is now configured.

Tests

A performance or response-time test is an IT service operation that returns a result. Retrieving a web page, downloading a file, and establishing a TCP connection are all examples of common service operations. A single test returns a result that can help you troubleshoot a performance issue. A group of tests return aggregate results that can help you evaluate service viability over a particular timeframe. You can also derive baseline standards that you can factor into infrastructure decision-making.

Service Performance Manager lets you create tests (model type RTM_Test) for your test hosts and discover tests that are configured on managed devices. You can schedule tests and also run them on demand. You can specify test thresholds and can incorporate tests into service management and service-level agreements.

Supported performance tests fall into the following three categories:

- Network response-time tests: Basic network response-time tests.
- Network service response-time tests: Measure the response times of essential network services.
- Network application response-time tests: Measure the response times of essential network applications.

Network Response Time Tests

Network response time tests are the most basic type of network response time tests. The following network response time tests are supported:

- **ICMP Ping**
  Tests the ICMP Echo Request messages from the test host to the destination address. If the test host and the source addresses are not same, then another ICMP Echo Request is issued from the test host to the source address. The resulting ICMP Echo Reply messages are used to determine the round-trip time between the source address and the destination address. A second metric, packet loss, is included in the results. When a series of ICMP Echo requests are made, which is typical of these tests, a coarse measure of packet loss is possible.
- **Jitter**
  Tests latency and losses between two endpoints. This test is designed to measure the quality of the network for applications that cannot tolerate loss or latency (such as VoIP).
- **Traceroute**
  Tests round-trip ICMP Echo from the host address to each layer with three hops in the discovered path.

Network Service Response Time Tests

Network service response time tests measure the response time of essential network services. The following network service response time tests are supported:

- **DHCP**
  Identifies the Dynamic Host Configuration Protocol IP address assignment.
- **DNS (Domain Name Service)**
  Translates the domain name to IP address.

Network Application Response Time Tests

Network application response time tests measure the response time of essential network applications. The following network application response time tests are supported:

- **Custom**
  Custom script execution.
- **FTP**

Indicates the File Transfer Protocol transaction time.

- **HTTP/HTTPS**
  Indicates the Hyper Text Transfer Protocol transaction time.
- **POP3**
  Indicates the Post Office Protocol transaction time.
- **SMTP**
  Indicates the Simple Mail Transfer Protocol transaction time.
- **SQL Query**
  Indicates the SQL query response time.
- **TCP**
  Indicates the Transmission Control Protocol connection time.
- **UDP Echo**
  Indicates the User Datagram Protocol transmissions echo round-trip delay.

Test Templates

Test templates contain the parameters for a particular type of test. You can selectively apply templates to multiple test host models. Or you can automatically apply templates to test host models that are added to the Global Collection models. DX NetOps Spectrum monitors Global Collection content and automatically creates the test that the template specifies on test hosts that are added to the Global Collection. Regardless of how a test template is applied to test hosts, the result is the same: the test that the template defines is created on the test hosts.

Event Alarms

Service Performance Manager generates various events representing significant occurrences that are related to response time testing. Some of these events produce alarms that notify you when a user-actionable event has occurred. For more information, see Event Codes.

You can also configure Service Performance Manager to generate an alarm whenever the duration of a response time test exceeds a predetermined threshold. Such alarms can be isolated to a specific link or path. For more information, see Specify Alarm Thresholds for a Test and Alarms and Events.

Performance Agents and MIBs

The following table lists performance agents and MIBs that are supported by Service Performance Manager. The tests that are supported for each agent or MIB are also given.

Agent/MIB	DX NetOps Spectrum Application Model Type	Tests
RFC2925	RFC2925App	ICMP Ping, Traceroute **Note:** You must verify with the vendor whether a particular device supports the RFC2925 MIB.
Cisco IOS IP SLAs Agent (Cisco IOS IP SLAs Agent is supported on Cisco routers running IOS 12.0 or greater.) **Note:** For full-mesh measurements between hubs, Cisco recommends using shadow routers that are dedicated for IOS IP SLAs. For more information, see the Cisco IOS IP SLAs documentation.	CiscoRTTMonApp	DHCP, DNS, FTP, HTTP, ICMP Ping, Jitter, TCP, UDP, ICMP_JITTER You can discover tests configured on Cisco agents with the SPM Test Discovery feature. For more information, see Discover Tests in the Network.

CA CA eHealth SystemEDGE Service Availability agent	Emp_SvRsp_App	Custom, DNS, FTP, HTTP, HTTPS, ICMP Ping, POP3, SMTP, SQL Query, TCP You can discover tests configured on SystemEDGE agents with the SPM Test Discovery feature. Tests for SystemEDGE agents are discovered in read-only mode only. For more information, see Discover Tests in the Network. **Note:** SystemEdge agents require the CA eHealth Service Availability module to run performance tests.
Network Harmoni SLAplus Agent OEM Vendors: Agilent Technologies, InfoVista, Peregrine Systems, Ericsson, Opticom, RedPoint, HP, Micromuse, Response Networks	HrmniSvcRspApp	DNS, HTTP, ICMP Ping
iAgent	SRISvcMonApp	DNS, FTP, HTTP, ICMP Ping, POP3
JUNOS Real Time Performance Monitor	JnprRFC2925ExtApp	ICMP Ping, Jitter, Traceroute
Other Support		
Cisco Ping MIB	CiscoPingApp	ICMP Ping
Wellfleet Ping MIB	WFPingApp	ICMP Ping

NOTE
It is only the current Cisco routers that support the RTTMON MIB. Older, pre-11.2 deployments support only the Cisco Ping MIB. Service Performance Manager does not support the Nortel Contivity Ping MIB (CONTIVITY-INFO-V1-MIB).

Service Performance Manager Features

Service Performance Manager supports multiple vendor and agent performance tests solutions and multiple implementation features.

Multiple Ways to Create Tests

You have the following options for creating tests with Service Performance Manager:

- Create a test from scratch.
- Create a version from a copy of another test.
- Use SPM Discovery to locate tests that are configured on test hosts.
- Apply test templates to Global Collection containers. When a test host is added to the container, a test with the settings in the template is created on the test host.

Test Scheduling

SPM offers scheduling capabilities to automate your performance testing. You can use preconfigured schedules that are provided by OneClick or you can create your own custom schedules. Scheduling tests lets you implement performance testing during peak and off-peak hours. You can then compare the results from each period and determine realistic performance standards.

For example, you want to schedule a test to run at a specific interval continually, such as 24 hours a day, 7 days a week. Or you can schedule tests to run during or after intervals when the infrastructure is in high demand. In either case, you can create test schedules that meet your requirements.

Automated Test Creation with Test Templates

Test templates let you create test configurations and apply them to multiple test hosts. Test templates leverage Global Collection container capabilities. By applying a template to a Global Collection container, you can automate the test creation process. The test that the template specifies is created for any test host that you add to the container that supports the test. Modifications to test parameters are also simple to perform. When you change template parameters, those changes extend to the parameters for all tests that are created from the template.

Single-Point Test Management

Service Performance Manager provides complete access and control over your test components on multiple landscapes from a single landscape. You can create, configure, locate, and manage tests from a single OneClick Console.

Service Performance Manager Tests and Service Level Agreement Management

Performance testing is indispensable for establishing IT service-obligation benchmarks and monitoring service performance. SPM response time test results provide a more accurate measure of service performance and viability than a notification that reports whether a service is up or down. Service consumers often consider service delivery speed the determining factor in whether a service meets their requirements.

For example, a service consumer considers a 20-second wait for a web page to load because it takes 5 seconds on average to retrieve and load a page. Such a consumer can assume that the sluggish web service is unavailable, even though it is accessible.

Service Performance Manager lets you specify response time thresholds to measure service performance in terms of latency. These thresholds support service-level agreements that ensure a specified response time and meet the requirements of the consumer. You can create and manage services and service-level agreements in DX NetOps Spectrum. For more information, see the Service Manager section.

User Roles

User access to SPM functionality depends on the rights that are granted to the user account. A DX NetOps Spectrum administrator can configure user roles and rights in OneClick. For more information, see the OneClick Administration section.

- **Service Performance Manager for Operators**
  When you are logged in to Service Performance Manager as an operator, you can view information for tests and test hosts and can run existing response time tests. As an operator, you cannot create, delete, edit, or copy any tests. Operators cannot change the state of a test on any test hosts.
- **Service Performance Manager for Administrators**
  When you are logged in to Service Performance Manager as an administrator, you have full access to all functionality, including creating, editing, deleting, and discovering tests.

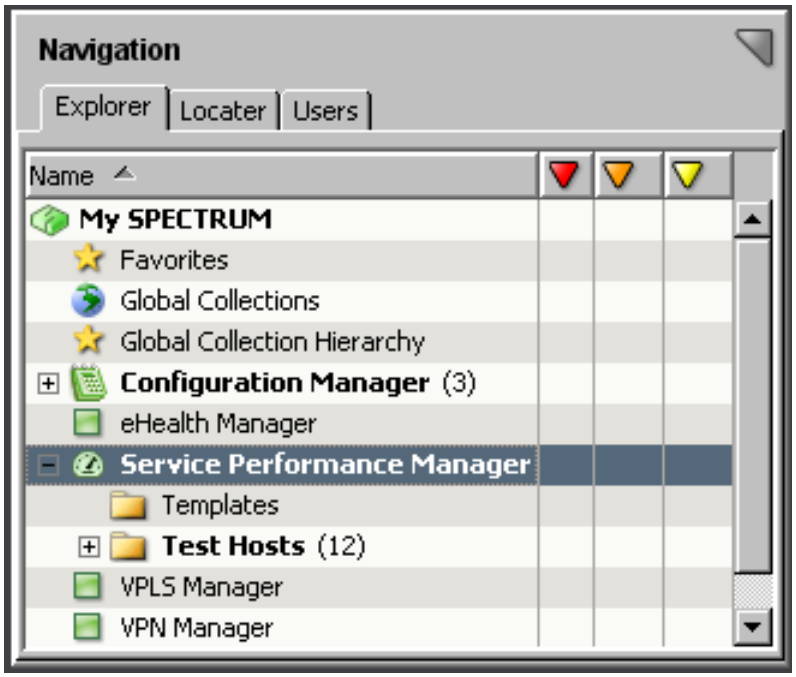
Access Service Performance Manager

This section describes the procedure to access Service Performance Manager components and the basic tasks that you can perform from the OneClick Console. This section provides information about configuring security for tests and test hosts.

You can access Service Performance Manager from the OneClick Console.

Follow these steps:

1. From the Explorer tab, select Service Performance Manager.
   The following image shows the navigation to access Service Performance Manager.



2. To display views, expand the Service Performance Manager node.
   The Templates and Test Hosts views appear.

Basic Tasks Overview

Once you have accessed Service Performance Manager in the OneClick Console Locater, perform the following basic tasks to work with its components:

- Locate a specific test component or group of components (tests, test hosts, test templates). For more information, see Finding Components.
- Create and edit tests, configure test parameters, schedule tests, specify test thresholds, manually run tests, and discover tests on test hosts. For more information, see Working with Performance Tests.
- Create and manage test templates and apply them to test hosts or to Global Collection containers that include test hosts. For more information, see Working with Test Templates.
- Create, run, and edit tests from the Command Line Interface. For more information, see Creating and Managing RTM Tests with the Command Line Interface (CLI).
- Generate reports on tests with Spectrum Report Manager and use the available result data. For more information, see Generating Reports on Test Data.

About Test Host and Test Security

A security string within the respective component controls security for test and test host models. You can use the security string to restrict access to test and test host models to authorized personnel only.

The security string originates from the device model. The test host model inherits the security string from the relevant device model. The test model later inherits the security string from the test host model.

Consider the following points when using test templates:

- When using test templates to create tests, the security string that is specified in the template is not propagated to the test model. The test model inherits its security string from the test host.
- When applying test templates to Global Containers, tests are created for the test hosts where you are authorized.

> **NOTE**
> You must have privileges to invoke detailed views of DX NetOps Spectrum models in OneClick Console and to modify security string settings for models. You can also set up DX NetOps Spectrum model security. For more information, see the OneClick Administration section.

Verify the following procedures to set or modify a security string for a test host or test:

- Secure a Test Host
- Secure a Test
- Overwrite a Security String

Secure a Test Host

You can secure a test host by specifying a security string for it.

Follow these steps:

1. In the Contents panel on the Information tab, expand Test Host Details for the test host that you want to secure.
2. Click set for the Security String parameter, and specify the security string.
   Permission to access the test host is now granted only to privileged users defined by the security string entered.

> **NOTE**
> All tests that are associated with a secured test host automatically inherit the host security string. You can override or remove a security string at a test level (or test host level). For more information, see Overwrite a Security String.

Secure a Test

You can secure a test by specifying a security string for it.

Follow these steps:

1. In the Component Detail panel on the Information tab, expand Test Details, General for the test to secure.
2. Click set for the Security String parameter.
3. Specify the security string.
   Permission to access the test is now granted only to the privileged users defined by the security string.

Override a Security String

You can override security strings for test hosts or tests (if necessary).

Follow these steps:

1. Access the Security String parameter for the component whose security string you want to override. For more information, see Secure a Test Host or Secure a Test.
2. Click set for the Security String parameter.
3. Edit or remove the security string.
   The security string is modified.

Finding Components

Service Performance Manager provides multiple options to find existing tests, test hosts, and test templates in DX NetOps Spectrum. For more information about search features, see the OneClick Administration section.

All Test Component Searches

This search category lets you search for all test hosts, tests, and test templates that are modeled in DX NetOps Spectrum. Using generated search results, you can select an item, view information about it, and can perform any operation that it supports.

Follow these steps:

1. Expand Service Performance Manager in the Explorer tab.
   The Templates and Test Hosts folders appear.
2. Expand the Templates folder.
   A list of all templates in the DSS environment is displayed.

   NOTE
   If you select the Templates folder, the list of templates also appears in the List tab of the Contents panel.
3. To display all test hosts, expand a test host.
   All test models under that host are displayed.

   NOTE
   If you select the Test Hosts folder, the list of test hosts also appears in the List tab of the Contents panel.
4. To display all tests, expand SPM folder in the Locater tab.
5. Double-click the All Tests option to launch the search.
6. Specify appropriate landscape information in the "Select Landscapes to Search" dialog and click OK.
   Search results appear in the Results tab of the Contents panel.

Criteria-Based Test Host Searches

You can search for test hosts that are modeled in DX NetOps Spectrum that meet the criteria you specify. This section describes the procedure to perform criteria-based test host searches and the supported criteria for test host searches.

How to Perform Criteria-Based Test Host Searches

You can perform a criteria-based search for test hosts.

Follow these steps:

1. Click the Locater tab in the Navigation panel.
2. Expand the SPM folder.
3. Expand the Tests Hosts By folder.
4. Select the type of criteria-based test host search to run.
5. Click the Search button.

NOTE
Depending on the search that you select, you can enter values (typically IP addresses) in a Search dialog before the search is executed.

Search results appear in the Results tab in the OneClick Contents panel. You can select an item that a search returned, view information about it, and can perform any operation that it supports.

Supported Criteria for Test Host Searches

You can perform a criteria-based test host searches based on the supported criteria for test host searches. Verify the following supported criteria for test host searches:

- **IP Address**
  Finds the test host with the Network Address or tests hosts that are associated with the address you specify in the Search dialog.

  NOTE
  The Search dialog does not support searching on partial IPs (for example, 10.253).
- **State**
  Finds test hosts in the following states:
  - **Active**
    Test hosts that have been activated.
  - **Contact Lost**
    Test hosts that have stopped responding to polls.
  - **Inactive**
    Test hosts that have not been activated or have been deactivated.
  - **Maintenance**
    Test hosts that are in maintenance mode.
- Test Discovery Support
  - **Supported**
    Finds test hosts that support test discovery.
  - **Unsupported**
    Finds the test host that does not support test discovery.
- **Test Type Support**
  Finds test hosts that support a particular test type.

Criteria-Based Test Searches

You can search for tests that are modeled in DX NetOps Spectrum and that meet criteria you specify by using the Tests By option. This section describes the procedure to perform criteria-based search for a test and the supported criteria for test searches.

Perform Criteria-Based Test Searches

You can perform a criteria-based search for a test.

Follow these steps:

1. Click the Locater tab in the Navigation panel.
2. Expand the SPM folder.
3. Expand the Tests By folder.
4. Select the type of criteria-based test search to run.
5. Click the Search button.

NOTE
Depending on the search that you select, you are prompted for additional values (such as IP addresses) before the search is executed.

Search results appear in the Results tab in the OneClick Contents panel. You can select an item that a search returned, view information about it, and can perform any operation that it supports.

Supported Criteria for Test Searches

You can perform a criteria-based test search. Verify the following supported criteria for test searches:

- **Configuration Parameters**
  Find tests that meet the following criteria as specified in the Search dialog:
  - Destination Address
  - Destination Address and Port Number
  - Source Address
  - Source Address and Destination Address
  - Source Address, Destination Address, and Port Number

  You can enter partial address strings (for example, 138.42) for the Source Address and Destination addresses. Searches with the destination port set to 0 return tests for which destination port is not applicable.
- **Discovery State**
  Find tests that meet the following criteria that are related to SPM Discovery:
  - **Discovery Read-Only**
    Tests that are discovered in Read-Only mode through SPM Discovery.
  - **Discovery Read/Write**
    Tests that are discovered in Read/Write mode through SPM Discovery.
  - **Other**
    Tests that were created by VPN Manager, other IP service applications, or the Command Line Interface.
  - **RTM Domain**
    Tests that were created using the user interface (and not through SPM Discovery).
  - **Stale Different Type**
    Tests that were discovered by Service Performance Manager with a corresponding test on the device of a different test type.
  - **Stale Entry Not Present**
    Tests that were discovered by Service Performance Manager that no longer have a corresponding test on the device. For more information, see Discover Tests in the Network.
- **Name**
  Find a test that is based on the Test Name you specify in the Search dialog.
- **Scheduled**
  Find all scheduled tests.
- **Status**
  Find a test using one of the following test status options:

  - Bad Community String
  - Bad Configuration
  - Device Disabled
  - Ready To Run
  - Running
  - Scheduled
  - Stopped
  - Timeout
  - User Disabled
- **Threshold Exceeded**
  See a list of all of the tests exceeding their thresholds.
- **Type**
  Find all tests of a particular type.

Working with Performance Tests

This section describes what tests are available and how to create them in your environment. Using DX NetOps Spectrum Command Line Interface, you can create and manage response time tests. For more information, see Creating and Managing RTM Tests with Command Line Interface (CLI).

Supported Test Types

This section includes information about specific test types that you can review before you create or can run the tests:

Custom Tests

Custom tests give you the flexibility to specify a custom script to run for the test. This test allows you to verify that the important services or other tasks are working efficiently.

> **NOTE**
> Custom tests are only for SystemEDGE hosts.

DHCP Tests

DHCP tests measure the round-trip time (latency) required to get an IP address. The DHCP server must be on the same subnet as the test host performing the DHCP test. To configure the test host to work with your DHCP server IP address, see the documentation for the device. For DHCP tests to work on a router, one of the neighboring routers must be a DHCP agent or relay. For more information, see the documentation for your device.

> **NOTE**
> DHCP test latency result values can exceed the timeout value for tests that are run on Cisco router test hosts. These values result from a known issue with Cisco IOS 12.2(2)T.

DNS Tests

DNS tests measure DNS lookup time. DNS-based host name-to-address translation must be enabled on the test host device performing the DNS test. You can verify whether DNS lookup is enabled and can enable it. For more information, see the documentation for the device.

DNS test results include the following metrics:

- Latency
- Packet Loss

FTP Tests

FTP tests measure the round-trip time to transfer a file.

FTP test results include the following metrics:

- Latency
- Packet Loss

> **NOTE**
> FTP tests that are run on Cisco test hosts using the RTTMON MIB require the username, password, and filename. However, FTP tests that are run on CA eHealth SystemEDGE Service Availability test hosts require a username and password.

HTTP Tests

HTTP tests measure the round-trip time to get a web page.

HTTP test results include the following metrics:

- Latency
- HTTP DNS Resolution Time
- HTTP TCP Connection Time
- HTTP Download Time

Considerations

- HTTP tests that are performed from Harmoni and iAgent test hosts generate only latency results.
- An HTTP test can fail on some Cisco systems using HTTP 1.1. For more information, see Firmware Issues.
- The HTTP version configuration setting is only available for HTTP tests that are run from Cisco test hosts. The Proxy URL setting is only available for HTTP tests that are run from Cisco and CA eHealth SystemEDGE Service Availability test hosts. For more information, see Configure Advanced Parameters.
- Service Performance Manager HTTP tests that require authentication are not supported on Cisco test hosts.

HTTPS Tests

HTTPS tests measure the round-trip time to get a web page over an encrypted Secure Sockets Layer (SSL) or Transport Layer Security (TLS) connection. An HTTPS test measures the same metrics as an HTTP test.

ICMP (Ping) Tests

ICMP (Ping) tests measure round-trip time from a source to a destination address.

ICMP test results include the following metrics:

- Latency
- Packet Loss

> **NOTE**
> Cisco, Juniper and Alcatel test hosts support Virtual Routing and Forwarding (VRF) Ping tests.

Considerations

- When using the Cisco IOS IP SLAs Agent, you can configure the ICMP Echo operation payload size by setting the request size. The router adds 36 bytes to the size specified. For example, if the request size is 28 bytes, the actual ICMP Packet size is 64 bytes (of IP packet). For more information, see the *Cisco IOS IP SLAs Agent Documentation*.
- The Harmoni agent does not support the configuration of sample count or packet size for ICMP Ping tests.

Jitter Tests

Jitter tests measure both latency and loss between a test host and a voice-enabled endpoint. However, they add a finer measure of the statistical behavior of a sequence of requests. The Mean Opinion Scoring (MOS) is also available from Cisco test hosts. The MOS provides a numerical measure of the quality of human speech at the receiver. Jitter test results include the following metrics, depending on the test host:

- Destination to Source Packet Loss
- Jitter Busies
- Jitter Egress
- Jitter Ingress
- Jitter Positive Destination to Source
- Jitter Positive Source to Destination
- Jitter Negative Destination to Source
- Jitter Negative Source to Destination
- Late Arrival Packet
- Latency
- Mean Opinion Score
- Missing in Action Packet
- Packet Loss
- Source to Destination Packet Loss

Jitter tests can be configured to target a destination port that is listening for Jitter traffic. For example, many Cisco devices running IOS IP SLAs use Port 16386. Failure to configure the port can result in test timeouts. Set the Destination Port parameter in the General options for Jitter tests. For more information, see Configure General Parameters.

POP3 Tests

POP3 tests measure POP3 response (transaction) time.

POP3 test results include the following metrics:

- Latency
- Packet Loss

SMTP Tests

SMTP Tests measures SMTP mail server response (transaction) time.

SMTP test results include the following metrics:

- Latency
- Packet Loss

SQL Query Tests

SQL Query tests confirm that SQL database servers are available by processing short queries that you specify.

NOTE
SQL Query tests are for SystemEDGE hosts only.

TCP Tests

TCP tests measure the time that is required to create a TCP connection.

TCP Connection test results include the following metrics:

- Latency
- Packet Loss

Trace Route Tests

Trace route discovers the layer three hops between the source and destination addresses. These tests also return a round-trip ICMP Echo measurement from the host address to each hop in the path.

Trace route test results include the following metrics:

- Latency
- Packet Loss
- For each hop, IP address and round-trip time

NOTE
Cisco, Juniper and Alcatel test hosts support Virtual Routing and Forwarding (VRF) trace route tests.

UDP Echo Tests

UDP Echo measures round-trip delay.

UDP Echo tests return Latency and Packet Loss results.

UDP tests must be configured to target a destination port that is listening for UDP traffic. For example, many Cisco devices running IOS IP SLAs use port 1967 and UNIX systems use port 7. Failure to configure the port can result in timeouts for UDP tests. Set the Destination Port parameter in the General options for UDP tests. For more information, see Configure General Parameters.

ICMP_JITTER

ICMP_JITTER tests measure end-to-end performance metrics like latency, round-trip time, jitter (inter-packet delay variance), and packet loss between a Cisco device (source) and any other IP device (destination).

ICMP_JITTER test results include the following metrics, depending on the test host:

- Latency
- Packet Loss
- Late Arrival Packet
- Jitter Busies
- Jitter Positive Source to Destination
- Jitter Positive Destination to Source
- Jitter Negative Source to Destination
- Jitter Negative Destination to Source
- Packet out of Sequence SD (Source to Destination)
- Packet out of Sequence DS (Destination to Source)
- Packet out of Sequence BOTH (SD and DS)
- Packet Skipped

Create Tests

You can create a performance test from scratch, or you can use an existing test as a starting point.

You can create tests on hosts that have not been activated. But you must activate test hosts before running the tests. For more information, see Activate and Deactivate Test Hosts.

You can also discover preconfigured tests on test hosts using SPM Discovery. Discovery models tests that have been created on test hosts using a method other than Service Performance Manager, such as the command line. For more information, see Discover Tests in the Network.

Create a Test

You can create a test of any type that the test host supports.

Follow these steps:

1. Expand Service Performance Manager in the Explorer tab.
   The Templates and Test Hosts folders appear.
2. Expand the Test Hosts folder.
   A list of all test hosts in the DSS environment is displayed.

   NOTE
   Expand a test host to see all test models that exist for that host.
3. Right-click the test host for which you want to create the test.
4. Select New Test and then select a test type.
   The New Test dialog opens. This dialog lets you configure test parameters.
5. Configure test settings and click OK.
   The new test is saved. Information about the new test appears on the Information tab in the Component Detail panel for the test host in the Test List table.

Create a Different Version of an Existing Test

You can create a test by saving a unique version of an existing test. The test can include the same settings, but a different name. You can save the new test to the base test host or to a different test host.

NOTE
You cannot copy tests between domains that are running different versions of DX NetOps Spectrum.

Follow these steps:

1. Expand Service Performance Manager in the Explorer tab.
   The Templates and Test Hosts folders appear.
2. Expand the Test Hosts folder.
   A list of all test hosts in the DSS environment is displayed. You can see a plus sign (+) for test hosts with existing test models.
3. Expand the test host whose test model you want to copy.
   A list of tests that are defined for that test is displayed.
4. Right-click the test, and select Copy Test.
   The Copy Test dialog opens, which lets you configure test parameters for the new test.

   NOTE
   Copied tests are disabled by default, and "_COPY" is appended to the test name.
5. (Optional) Rename the test.
6. Modify test settings, and click OK.
   The new test is saved. Information about the test appears on the Information tab in the Component Detail panel for the test host in the Test List table.

Discover Tests in the Network

SPM Test Discovery lets you discover and model performance tests that are configured on test hosts but not configured with Service Performance Manager.

Considerations

- Administrator role privileges are required to use SPM Test Discovery.
- Use a device that SPM Test Discovery supports.

When you run a test Discovery, you are prompted to select the Discovery mode. The following two Discovery modes determine how the tests are created in Service Performance Manager:

- Read-Only Discovery Mode
- Read/Write Discovery Mode

Read-Only Discovery Mode

When you discover tests in read-only mode, you cannot edit test configurations after the tests are modeled in DX NetOps Spectrum. Because, you do not have SNMP *set* privileges to tests. Test Discovery takes schedule information from the tests that are configured on the device and reads test results at the appropriate interval.

Considerations

- Administrator access to read-only tests is similar to that of the Operator role, but it includes full access to Threshold parameters. For more information, see Specify Alarm Thresholds for a Test.
- Read-only tests cannot be run manually; they must be scheduled.
- Sample Count is always 1.
- You can copy read-only tests to create tests that can be modified in Service Performance Manager.
- You can verify that data from read-only tests is available for response time reports, which you can generate with Spectrum Report Manager. Confirm that the test is in the Active state at the time of the read-only Discovery. You can also verify that the Events service indication in OneClick shows the Up status.

Read/Write Discovery Mode

When you discover tests in read/write mode, you can edit test configurations after the tests are modeled in DX NetOps Spectrum. SNMP *set* privileges to the tests are required. You can handle tests on these test hosts exactly like tests that are created manually in Service Performance Manager. Therefore, you can run tests on demand and can stop and edit. For more information, see Configure Tests.

Considerations

- If the test is activated on the device, the Schedule State field for the Schedule parameter field is set to Enabled.
- Sample Count is set to 1

Set the Test Name for Cisco IP SLA Tests

When discovering Cisco IP SLA tests in DX NetOps Spectrum for Cisco routers, the Tag value of the Cisco IP SLA test can be used as the test name in Service Performance Manager. Select a setting to enable this feature.

> **NOTE**
> This option is available for Cisco routers and for tests that were discovered using SPM Test Discovery only.

Follow these steps:

1. Select Service Performance Manager in the Explorer tab.
   Information about Service Performance Manager appears in the Information tab of the Contents panel.
2. Expand the General Information subview.
3. Modify the following field:
   Use Tag Field as Test Name for Cisco Test Host Discovery
   Specifies whether the Tag value is used as the test name during SPM Test Discovery.

   > **NOTE**
   > For tests created in DX NetOps Spectrum, this setting has no effect.

Run Discovery

You can run test Discovery on a single test host.

Follow these steps:

1. In the Explorer tab in the Navigation panel, under Test Hosts, right-click the test host where you want to run Discovery.
2. Select Discover Tests from the right-click menu.
   If the selected test host supports both Read-Only and Read/Write Discovery modes, the Discover Tests: Select Discovery Mode dialog opens.
   The Discover Tests option is not available in the menu if the host does not support Discovery.
3. Select the appropriate option for the test discovery mode you want to run: Read-Only or Read/Write.
   When Discovery completes, the Discover Tests Complete dialog indicates how many tests were created or updated.

You can also locate multiple test hosts that support test discovery and run test discovery on multiple hosts.

Follow these steps:

1. In the Locater tab in the Navigation panel, expand the SPM folder.
2. Run the Test Hosts By, Test Discovery Support, Supported search to locate Discovery-compatible test hosts.
   The Contents panel lists test hosts that support test Discovery.
3. Select one or more test hosts where you want to discover tests, and click the Discover Tests icon.

   > **NOTE**
   > The Discover Tests icon is disabled if a selected test host is not active, or if contact with the host is not established. For more information, see Activate and Deactivate Test Hosts.

   If the selected test host supports both Read-Only and Read/Write discovery modes, the Discover Tests: Select Discovery Mode dialog opens.

4. Select the appropriate option for the test Discovery mode you want to run: Read-Only or Read/Write.
   When Discovery completes, the Discover Tests Complete dialog indicates how many tests were created or updated.

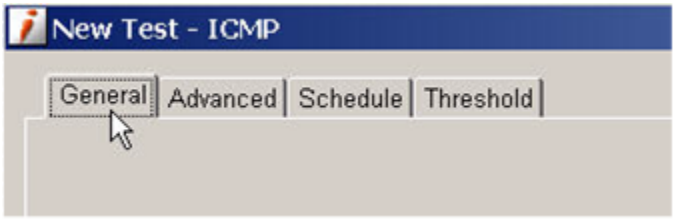
SPM Test Discovery Event Codes

The Events tab in the Component Detail panel provides results from the Discovery on the test host. The following list summarizes SPM Test Discovery event codes and descriptions:

- **SPM Test No Longer On Device Event (0x04560059)**
  Occurs when SPM Test Discovery fails to match an existing SPM Read-Only test to a table entry on the device, this event and a corresponding yellow alarm is generated on the test model.
- **SPM Test No Longer Running On Device Event (0x0456005a)**
  Occurs when SPM reads the test results and detects that the numberOfPktsSent object on the device has not increased. The operational state on the device is also read and found to be InActive. This event is generated, and the following actions occur:
  – No data is processed.
  – Schedule state on the SPM test is set to Disabled, and no more data is read.
- **SPM Duplicate Result Event (0x0456005b)**
  Occurs when SPM reads the test results and detects that the numberOfPktsSent object on the device has not increased. The operational state on the device is also read and found to be Active. This event is generated, and the following actions occur:
  – No data is processed.
  – Schedule state on the SPM test remains Enabled; therefore data is processed from the next scheduled test.
- **SPM Test Discovery Completion Event (0x0456005c)**
  Generated after an SPM Test Discovery has run. Indicates the mode (Read-Only or Read/Write) in which the Discovery was run. Can contain any of the following error output:
  – No Errors
  – No SPM Tests Were Discovered
  – Test type is invalid
  – Test name is null
  – Test timeout is 0
  – Test frequency is 0
  – Test packet size is 0
  – Test sample count is 0
  – Test port number is invalid
  – Test IP Address is invalid
  – Test URL is invalid
  – Test host name is invalid
  – Test user name is invalid
  – Test password is invalid
  – Test filename is invalid
- **SPM Test Type Mismatch Event (0x0456005d)**
  Occurs when SPM Test Discovery matches an SPM Read-Only test to a table entry on the device that is of the wrong test type. Generates this event and a corresponding yellow alarm on the test model.
- **SPM Stale Test Clear Event (0x0456005e)**
  Clears 0x4560059 or 0x456005d if subsequent SPM Test Discovery clears the condition.

Configure Tests

Whenever you issue a create, copy, or edit test command, Service Performance Manager displays a test configuration dialog. You can specify test parameters, set up a test schedule, and specify test thresholds. The following image shows example test configuration categories:



The configuration options depend on the type of test. Some of the options that are discussed in the following procedure do not apply to all types of test.

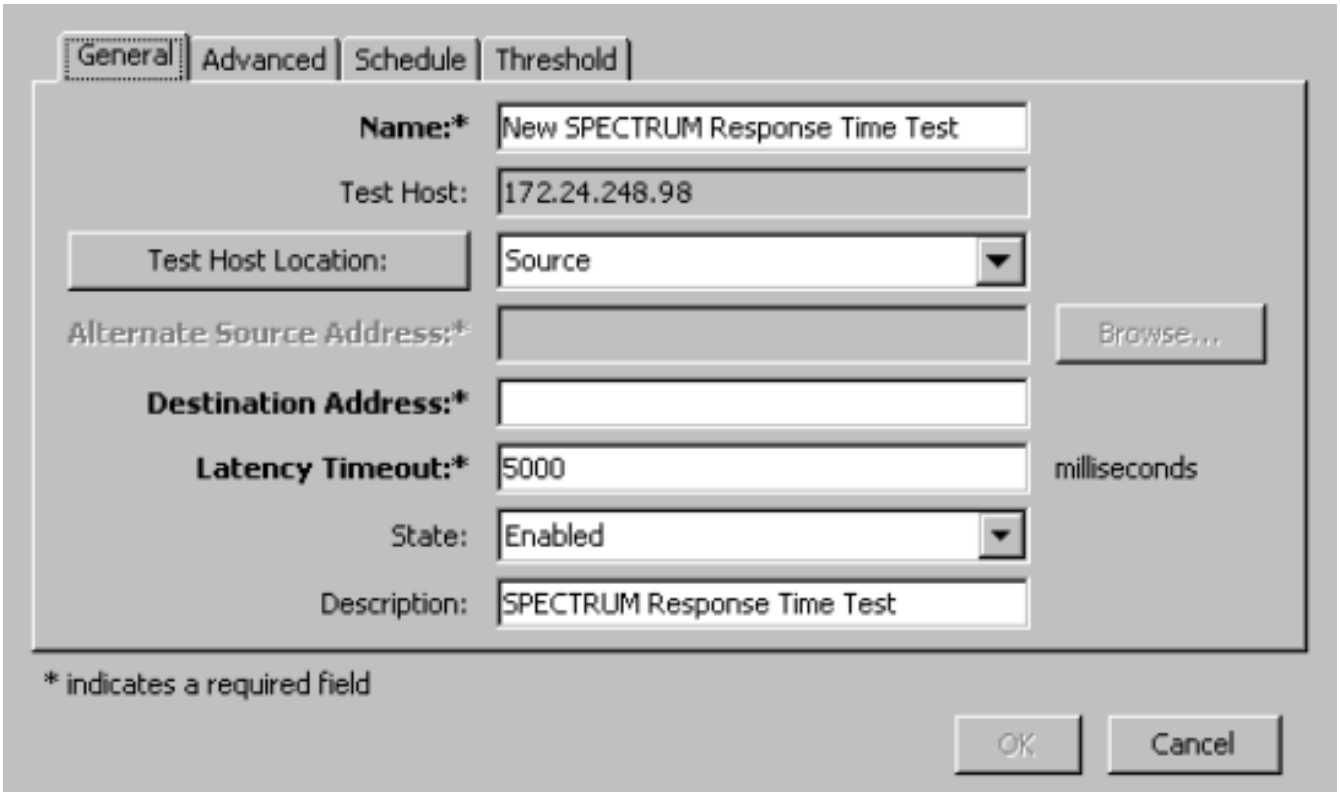
Follow these steps:

1. Enter values for the test in the parameter categories for your particular test.
   For example, test scheduling is disabled by default. If you are not interested in scheduling test runs, ignore the scheduling parameters. The same applies to the test thresholds.
2. Click OK in each parameter category to save your settings.

   NOTE
   The OK button is disabled if you do not enter required values.

Configure General Parameters

The General tab lets you configure required parameters for a test. The following image shows an example configuration dialog for an ICMP (Ping) test.



Standard General Parameters

The following general parameters are available for all performance tests that Service Performance Manager supports:

- **Name**
  Specifies the test name.
  Default: New SPECTRUM Response Time Test
- **Test Host**
  Indicates the IP address of the test host for the test.
- **Latency Timeout**
  Specifies the number of milliseconds for the response. If no response is received before this timeout occurs, DX NetOps Spectrum generates a timeout event. Any response that arrives after this timeout is ignored. Set the timeout higher than the threshold setting.
  Default: 5000 milliseconds

  NOTE
  The Harmoni agent does not support latency timeout configuration.
- **State**
  Enables and disables the test.
  Default: Enabled
- **Description**
  Specifies test annotations.
  Default: SPECTRUM Response Time Test

General Parameters for Specific Test Types

The following parameters may be included under the General tab, depending on the type of test:

- **Alternate Source Address**
  Specifies the IPv4 address or hostname of the test source location when it is not the test host. For example, specifies a mid-path location and extended path location for ICMP Ping test scenarios.

  NOTE
  IPv6 addresses are not supported.
- **Codec Type**
  Specifies the VOIP codec (audio compression/decompression) type to test: G.711 U-law, G.711 A-law and G.729A.

  NOTE
  You can set a voice quality threshold value (100 - 500) for the selected codec with the Mean Opinion Score parameter under the Threshold tab.
- **Connect String**
  Provides the string of commands that are used to connect to the database. For example,
  `jdbc:mysql://172.22.246.43/mysql?user=root&password=root`
- **Database Name**
  Specifies the name of the database.
- **Database Type**
  Specifies the type of database to test. Correct drivers must be installed on the SystemEDGE server.
- **Destination Address**
  Specifies the destination address for the test except those of type DHCP, DNS, HTTP. You can enter an IPv4 address or a host name.

  NOTE
  IPv6 addresses are not supported.
- **Destination DNS Server**
  Specifies the destination address of the DNS server for DNS tests. You can enter an IP address or a host name.
- **Destination Port**
  Specifies the port number where the service is running. For Mean Opinion Score (MOS) support in Jitter tests, the destination port can be an even-numbered port in the range 16384 through 32766 or 49152 through 65534.
- **Destination URL**
  Specifies the URL used in HTTP and HTTPS tests.
- **File Name**
  Specifies the file path that is used for the FTP test.
- **Lookup Name**
  Specifies the IP address, host name, or fully qualified domain name (FQDN) of the host in DNS tests.

  NOTE
  Some agents, such as iAgent, require the use of an FQDN rather than a host name.
- **Operation Type**
  Specifies the type of FTP operation to test.
  - **Login**
    This test logs in using the specified user name and password and then logs out.
  - **Get**
    This test logs in and reads the remote file that is specified in the File Name field (but does not perform a write operation), then logs out.
  - **Put**
    This test logs in and writes the local file that is specified in the File Name field out to the FTP server, then logs out. If the remote directory does not have write permissions, the test fails.

  Default: Login
  Note: Operation Type is available for FTP tests for SystemEDGE hosts only.
- **Password**

Specifies the password for FTP, HTTP, HTTPS, or POP3 test authentication. For SQL Query, this password is used for database access.

- **Query String**
  Specifies the query statement to execute.
- **Script Path**
  Specifies the name and location of a valid script.
- **SQL Database Server**
  Indicates the host name or IP address of the SQL database server.
- **SQL Driver**
  Specifies the name of the SQL driver. For example,
  `com.mysql.jdbc.Driver`
- **Test Host Location**
  Specifies the location of the test host on the path between the source and destination for ICMP Ping tests.
  Default: Source

  NOTE
  See About the Test Host Location Parameter for more information.
- **User Name**
  Specifies the user name for FTP, HTTP, HTTPS, or POP3 test authentication. For SMTP, you can use this email address to test. For SQL Query, this is the user name for database access.
- **Voice Test**
  Specifies whether to test voice quality for Jitter tests.

About the Test Host Location Parameter

Test host location refers to the path between the source and destination of a response time test. The location is critical to response time measurement.

Test host locations:

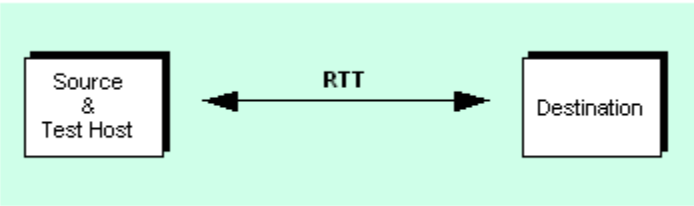
- **Source:** the test host is the source of the test.
- **Mid path:** the test host is located between the source and destination points.
- **Extended path:** test host is not located between the source and destination points.

NOTE
Only ICMP Ping tests support mid path or extended path tests.

Mid path and Extended path tests are useful only when the source or destination address is not a test host. In both of these test types, you can configure test host location.

Source Location

In the most common response time test scenario, the source, or starting point, of the test is also a test host, as shown in the following diagram:



In this case, the test host generates a transaction directly with the destination and measures the RTT.

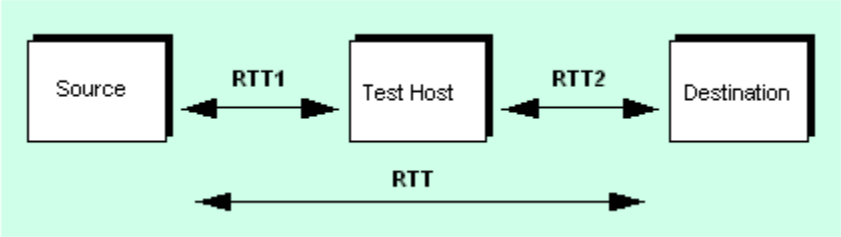
When the source of a test is a test host, response time measurements are the most direct and accurate. For this reason, you can set up tests with a test host as the source whenever possible.

Mid Path Location

In a mid path test configuration, a test host lies on the path between the source and destination points for which RTT is being measured. The source of the test is not capable of being a test host, so it cannot initiate or perform any response time measurements. Use the following calculation to measure the response time for a mid path test:

Response time for a mid path test = (response time (RTT1) from test host to the source) + (response time (RTT2) from test host to the destination)

This value is a relative value which approximates actual RTT. The following image is an example of a mid path test configuration:

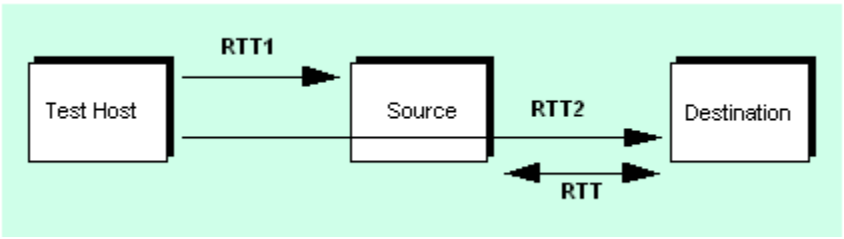


Extended Path Location

In an extended path test configuration, the source of the test is not capable of being a test host. In addition, no test host is located between the source and destination as in the Mid Path Location scenario. However, the source lies directly in the path between a test host and the destination. In this case, use the following calculation to measure the response time:

Response time = (response time (RTT2) from the test host to the destination) -- (response time (RTT1) from the test host to the source)

This value is a relative value which approximates actual RTT. The following image is an example of an extended path test configuration:



Configure Advanced Parameters

The Advanced tab lets you configure additional parameters for a given test type. Each test type has its own specific parameters.

Verify the following Advanced Parameters for all test types:

- **Alternate Packet Address**
  Specifies an alternate source IP address instead of using the default network address of the device that is discovered in DX NetOps Spectrum.

  NOTE
  The Alternate Packet Address field is available for all Cisco test hosts that support IOS IP SLAs Agent.
- **Alternate Packet Port**
  Specifies an alternate packet port instead of using the default packet port of the device that is discovered in DX NetOps Spectrum.

NOTE
The Alternate Packet Port field is available for all Cisco test hosts that support IOS IP SLAs Agent.

- **Delete Messages**
  Specifies whether to delete the messages that were downloaded during the test or to leave the messages on the test system.
  Default: False
  Note: Delete Messages is available for POP3 tests for SystemEDGE hosts only.
- **Download Content**
  Specifies whether to download all images, frames, scripts, and applets with the core HTML code from the website or URL.
  Default: False
  Note: Download Content is available for HTTP and HTTPS tests for SystemEDGE hosts only.
- **Download Type**
  Specifies whether the first or all messages are downloaded for POP3 tests.
  - **First**
    This option downloads only the first message for this user account.
  - **All**
    This option downloads all messages for this user account.

  Default: First
  Note: Download Type is available for POP3 tests for SystemEDGE hosts only.
- **Fail On Content Error**
  Specifies whether any errors encountered while downloading images, frames, scripts, and applets cause the test to fail.
  Default: False
  Note: Fail On Content Error is available for HTTP and HTTPS tests for SystemEDGE hosts only.
- **Filter Timeout Data**
  Specifies whether the Performance tab the OneClick Component Detail panel displays all data or a subset of data minus timeouts.
  Default: True
- **Frame Depth**
  The number of levels the test should traverse when downloading nested frames. The HTTP and HTTPS tests download all frames, images, external scripts, and applets during the page download. The measurement reflects the user experience when downloading a web page.
  Default: 3
  Note: Frame Depth is available for HTTP and HTTPS for SystemEDGE hosts only.
- **HTTP Version**
  Specifies the HTTP version (1.0 and 1.1) for HTTP tests.
  Default: 1.1
- **Mail Body Size**
  Specifies the size (in bytes) of the test message to send.
  Default: 1000
  Note: Mail Body Size is available for SMTP tests for SystemEDGE hosts only.
- **Minimum Matches**
  The minimum number of times the search expression must be found. If the search expression is not found at least as many times as you specify in this field, the test fails.
  Default: 1
  Note: Minimum Matches are available for HTTP and HTTPS tests for SystemEDGE hosts only.
- **Outgoing User Name**
  Specifies the outgoing user name for SMTP authentication for SystemEDGE hosts.
- **Outgoing Password**

Specifies the password for the outgoing user name for SMTP authentication for SystemEDGE hosts.

- **Packet Size**
  Specifies the value (in octets) that limits the size of the packets that are used in the test.

  NOTE
  The Harmoni agent does not support packet size configuration.
- **Proxy Password**
  Specifies the password for the Proxy User Name. The password is encrypted in the DX NetOps Spectrum database.
  Note: Proxy Password is available for HTTP tests for Cisco routers and HTTP and HTTPS tests for SystemEDGE hosts only.
- **Proxy Server**
  Specifies the host name (the name or IP address) of the proxy server to use if the system from which you are testing does not have direct Internet access.
  Note: Proxy Server is available for HTTP and HTTPS tests for SystemEDGE hosts only.
- **Proxy URL**
  Specifies the proxy URL for HTTP tests. For more information, see HTTP Tests.
- **Proxy User Name**
  Specifies a valid user name to be authenticated on the specified proxy server.
  Note: Proxy User Name is available for HTTP tests for Cisco routers and HTTP and HTTPS tests for SystemEDGE hosts only.
- **Sample Count**
  Specifies the number of times a test is performed during a test run.
  Default: 5
  - For Cisco IP SLA tests, the agent supports a sample count of '1' only for all tests other than Jitter tests. For more information, see Considerations for CA eHealth and Cisco IP SLA Tests.
  - RTTMON may occasionally perform more than the specified sample count repetitions for a Service Performance Manager test.
  - The Harmoni agent does not support the configuration of sample count.
- **Text Match**
  Specifies a regular expression or text string that you want to match on the pages you test.
  Note: Text Match is available for HTTP and HTTPS tests for SystemEDGE hosts only.
- **Type of Service**
  Specifies the Type of Service (TOS) parameter for test packets that are sent by the test host. For example, create a response time test with the TOS parameter that VOIP uses and test the performance of the network in routing the packets.
  The Type of Service octet in an IP datagram header enables packets with different TOS values to be routed differently.
- **Use SSL**
  Specifies whether to enable Secure Sockets Layer security in case the SMTP server requires SSL authentication.
  Default: False
  Note: Use SSL is supported for SMTP tests for SystemEDGE hosts only.
- **VRF Name**
  Specifies the VPN routing instance for the test (Ping and Traceroute for Cisco and Juniper test hosts).

Schedule a Test

The Schedule tab lets you schedule a test. Depending on which test you are configuring, you may or may not have access to the following Schedule parameters.

- **Schedule State**
  Enables and disables a test schedule.
  Default: Disabled
- **Schedule Time Interval**

Specifies the interval between scheduled test runs.
Default: 900 seconds

- **Schedule**
  Specifies one of the predefined schedules available from the drop-down list. For more information, see Create a Schedule.
  Default: 24/7

  NOTE
  Time zones for Schedule are local to the SpectroSERVER where the test host running the scheduled test is modeled.

- **Threshold Violation Interval**
  Specifies an alternate test interval that can be used during a threshold violation period. Decreasing the interval during a period of high latency can provide more precise data.
  Default: 300 seconds

  NOTE
  If the Threshold Violation Interval is exceeded, the test does not resume until the Threshold Violation Interval is reached, a lower or higher value than the scheduled time interval.

  WARNING
  Setting this value too low can create additional traffic and load on the router during a period of high latency to make the situation worse.

Create a Schedule

NOTE
You can create test schedules if the predefined schedules do not meet your particular requirements. The schedules that you create are not test-specific; they are available for all tests. For more information about creating schedules, see the Using OneClick section.

NOTE
If you create a schedule with Recurrence set to None, the test runs once on the start date. Then the schedule reverts to the default schedule, and the new schedule is disabled.

Follow these steps:

1. Click Create in the Schedule tab.
   The Create Schedule dialog opens.
2. Configure schedule settings, and click OK.
   The new schedule appears in the Schedule drop-down list.

Specify Alarm Thresholds for a Test

The Threshold tab lets you configure alarm thresholds to monitor response time measurements on a specific link or path. Threshold parameters vary among tests. When a threshold value is exceeded, DX NetOps Spectrum generates an event, an alarm, or both, based on the settings you specify.

Threshold parameters depend on the type of test. The following list describes Threshold parameter options:

- **Active Thresholds Schedule**
  Specifies one of the predefined schedules during which thresholds are in effect and which is available from the list. You can use this feature to measure tests against thresholds only during certain times of the day. For more information, see Create a Schedule.
  Default: 24/7

NOTE
Time zones for schedules are local to the SpectroSERVER where the test host running the scheduled test is modeled.

- **Threshold Events Asserted On**
  Specifies the model from the list where events are asserted.
- **Calculate the threshold On (From 10.4.2.1)**
  Specifies the criteria to calculate the threshold. You can specify the **minimum**, **average**, or **maximum** as threshold limit for Cisco devices and **average** for non-Cisco devices. The threshold calculation is applicable only for **time-related** test results. To see the alarms for threshold values, set Minor Alarm, Major Alarm, Critical Alarm values.

  NOTE

  - If Minor Alarm, Major Alarm, Critical Alarm values are 0 and only event value is set in latency, then only the result event is generated.
  - If Minor Alarm, Major Alarm, Critical Alarm values are set, the result event is generated and a new event generate alarm if set threshold values are breached.
- **Threshold Types**
  Identifies threshold settings for the threshold types that are related to a test.
- **Status**
  Enables and disables each threshold type for a test.
- **Event**
  Specifies a response time threshold value (greater than 0) that, if exceeded, generates an event.
  Depending on the threshold type, you can use one of the following standards to specify a threshold value:
  – A time interval in milliseconds (ms)
  – A percentage (%) of packet errors encountered
- **Minor Alarm**
  Specifies a response time threshold value (greater than 0) that, if exceeded, generates a minor alarm.
  Depending on the threshold type, you can use one of the following standards to specify a threshold value:
  – A time interval in milliseconds (ms)
  – A percentage (%) of packet errors that were encountered
- **Major Alarm**
  Specifies a response time threshold value (greater than 0) that, if exceeded, generates a major alarm.
  Depending on the threshold type, you can use one of the following standards to specify a threshold value:
  – A time interval in milliseconds (ms)
  – A percentage (%) of packet errors encountered
- **Critical Alarm**
  Specifies a response time threshold value (greater than 0) that, if exceeded, generates a critical alarm.
  Depending on the threshold type, you can use one of the following standards to specify a threshold value::
  – A time interval in milliseconds (ms)
  – A percentage (%) of packet errors encountered
- **Threshold Cycles**
  Specifies a value (greater than 0) for the number of consecutive test cycles that violate the threshold before DX NetOps Spectrum generates an event or alarm.
- **Clear Threshold**
  Specifies a response time threshold value (greater than 0) tha,t if not exceeded, clears an event or alarm. Depending on the threshold type, you can use one of the following standards to specify a threshold value:
  – A time interval in milliseconds (ms)
  – A percentage (%) of packet errors encountered
- **Clear Cycles**
  Specifies a value (greater than 0) for the number of consecutive test cycles that comply with a threshold before DX NetOps Spectrum clears an event or alarm.

Establish Baseline Data to Determine Valid Thresholds

The threshold values to use for a test are often too high or too low. You can establish baseline response times that you can then use to determine appropriate thresholds.

Use the following procedure to determine valid thresholds for a test.

Follow these steps:

1. Schedule multiple test runs over the interval to which you want to apply thresholds to tests.
2. Analyze test result data to determine realistic performance thresholds. For more information, see Viewing Service Performance Manager Information.
3. Configure test thresholds that are based on the results of your analysis.

Considerations for CA eHealth and Cisco IP SLA Tests

Service Performance Manager test scheduling is handled by the respective agent. The other management applications, such as CA eHealth, can discover the tests while providing an administrator a single point for RTM test configuration.

Cisco IP SLA tests only support a sample count of '1' for all test types other than Jitter. If a sample count other than 1 is used when configuring the test in Service Performance Manager, DX NetOps Spectrum schedules the test instead of the agent. This results in CA eHealth (and other management applications) not being able to discover the test, because CA eHealth only discovers tests that are scheduled by the agent.

The following test types describe the results that can be expected, if you specify a Sample Count value while configuring a test for a Cisco IP SLA agent:

- For all non-Jitter test types:
  - If Sample Count > 1, DX NetOps Spectrum schedules the test and CA eHealth (and other management applications) fails to discover the test.
  - If Sample Count == 1, DX NetOps Spectrum uses the agent to schedule the test and CA eHealth (and other management applications) discovers the test.
- For Jitter tests, DX NetOps Spectrum uses the agent regardless of the sample count and CA eHealth (and other management applications) discovers the test.

The Sample Count value is specified on the Advanced tab when configuring a test. For more information, see Configure Advanced Parameters.

Run Tests on an On-Demand Basis

You can run tests manually on an on-demand basis if the following criteria are met:

- The test is not scheduled or is scheduled and the schedule is disabled.
- The test is enabled.
- The test host for the test is activated. See Activate and Deactivate Test Hosts for more information.

Typically you manually run tests for diagnostic purposes. For example, you can run an HTTP test to determine a web server response to requests, or you can run a Jitter test to determine the quality of VoIP transmission.

Follow these steps:

1. In the Explorer tab under Service Performance Manager, expand the Test Hosts folder.
   Available test hosts appear.
2. Select the test host.
   Available tests appear in the Test List table on the Information tab of the Contents panel.
3. Select the test(s) you want to run and click the Run Test icon.
   Information about the test is updated in the Test List table.

NOTE
If the test information is not updated, click the Refresh icon.

Test result information is available in the Component Detail panel.

NOTE
To display the Component Detail panel from the Test List table, click the Information icon.

Activate and Deactivate Test Hosts

All test hosts are modeled in DX NetOps Spectrum in the active state. Tests can be configured on a test host in either the active or inactive state. However, tests can only be run on active test hosts.

NOTE
If the device model that is associated with a test host is in maintenance mode, an error is displayed and the test host is not activated. When the device model is taken out of maintenance mode, the test host is activated (if it was active when the device model went into maintenance mode).

Follow these steps:

1. In the Explorer tab under Service Performance Manager, expand the Test Hosts folder.
   The test hosts appear in the Contents panel on the List tab. The State field indicates whether test hosts are Active or Inactive.
2. In the Contents panel on the List tab, select the inactive test host(s) you want to activate, right-click the selection, and select Activate Test Host.
   The test host is activated.
3. To deactivate a test host, select the active test host(s) you want to deactivate, right-click the selection, and select Deactivate Test Host.

   NOTE
   Any scheduled tests stop running until the test host is reactivated.The test host is deactivated.

Manage Tests

If you have the required privileges to the test, you can edit parameter settings and can delete any test. The administrative-user and read/write privileges are required to make modifications.

WARNING
Editing and deleting tests or test templates should be performed with caution by qualified personnel. This guidance especially applies in cases when, for example, Service Performance Manager performance test results are monitored by SLAs (service level agreements) that are modeled in DX NetOps Spectrum or when tests are run for infrastructure performance analysis.

Edit a Test

You can edit all test parameter settings as required.

When you edit a test, the test stops and restarts after all changes have been completed. As a result, the test schedule can be disrupted. For example, if a test has a 60-minute interval between scheduled runs and it is edited 58 minutes after the last scheduled run, it runs for 1 hour and 58 minutes after it is restarted.

Follow these steps:

1. Locate a test to edit. In the Explorer tab under Service Performance Manager, expand the Templates or Test Hosts folders.
2. Expand templates or test hosts.

Available tests appear.
3. Right-click the test to edit, and select Edit Test.
4. Modify test parameters as required. For more information about configuration options, see Configure Service Performance Manager Tests.

Delete a Test

When you delete a test, you remove it permanently from DX NetOps Spectrum.

> **NOTE**
>
> If a Test is discovered in Read/Write mode, and if an admin deletes it, it is removed from Spectrum and also from the Test Host.

Follow these steps:

1. Access the Explorer tab under Service Performance Manager.
2. Locate a test to delete by expanding the Templates or Test Hosts folders, and then templates or test hosts. Available tests appear.
3. Right-click the test to delete, and select Delete.
   The Confirm Delete dialog opens.
4. Click Yes.
   The test is deleted.

Working with Test Templates

This section describes the procedure to create and manage Test Templates.

About Test Templates

A test template is a test configuration that you can apply to multiple test hosts in a single step to create tests on the test hosts. You configure and edit test templates the same way you do with tests. Where they are different is in their scope of application. When you create a test, you create it on a single test host. When you create a test template, however, you can apply it to multiple tests hosts in the following ways:

- Selectively apply a template to multiple test hosts that support the test that is specified by the template.
- Apply a template to any number of test hosts that support the tests that are specified by the templates and that are added to a Global Collection container.

Verify the following primary advantages of using a test template:

- You can implement automated bulk performance testing.
- All modifications to test template parameters are automatically applied to the tests on each test host to which the template is applied. For example, changing a threshold value in a template changes that threshold value for all tests that are derived from it.
- A test template allows you to specify that a test is either a test host or a destination type. A test host type specifies a variable test host but a constant destination. A destination type specifies a constant test host but variable destinations.

Create Test Templates

You can create performance test templates using either of the following methods:

- Create a test template from scratch
- Save a different version of an existing test template

You can use any method to create a test template. For more information, see Configure Service Performance Manager Tests. You can also configure the following template-specific parameters:

- **Template Type**
  Specifies Test Host or Destination. A test host type specifies a variable test host with a constant destination. A destination type specifies a constant test host with variable destinations. (The Destination configuration is not available for DHCP and HTTP test templates.)
- **Global Collections**
  (Optional) Specifies one or more Global Collection containers. OneClick automatically applies the test that is specified by the template to all test hosts that you add to the container if they support the test.

Create a New Test Template

You can create a test template from scratch. Use this method to create a test template if no other templates have been created.

Follow these steps:

1. In the Explorer tab under Service Performance Manager, right-click Templates, select Create Test Template, and select a test type.
   The Create Test Template dialog provides test template parameters. You can specify the Template type and one or more Global Collection containers to which to apply the template.
2. Configure test template settings, and click OK.
   The new template is saved and appears in the Templates folder.
3. For information about the new test template, select the new template.
   Information appears in the Information tab in the Contents panel.

Create a Different Version of an Existing Test Template

You can create a test template from a copy of another template.

> **NOTE**
> You cannot copy test templates between domains where different versions of DX NetOps Spectrum are running.

Follow these steps:

1. In the Explorer tab under Service Performance Manager, expand the Templates folder.
   The test templates appear.

   > **NOTE**
   > If no templates appear, create a test template.
2. Right-click the test template to copy, and select Copy Test Template.
   The Copy Test Template dialog lets you select settings for the new template.

   > **NOTE**
   > The test templates that you copy are disabled by default, and "_COPY" is appended to the test name. You can rename the template.
3. Modify test template settings, and click OK.
   The new template is saved and appears in the Templates folder.
4. (Optional) Select the new template.
   Information appears in the Information tab in the Contents panel.

Apply a Test Template to Test Hosts

You can selectively apply a test template to a group of test hosts that support the template test type. You also have the option to automate the process by placing test hosts in a Global Collection container and applying a test template to the container. Apply a test template to a Global Collection container so that tests are automatically created for all test hosts that support the test type as they are added to the container. For more information, see the Modeling and Managing Your IT Infrastructure section.

Selectively Apply a Test Template to One or More Test Hosts

You can manually apply a test template to a selection of test hosts. You can precisely specify only those test hosts for which you want to create tests.

Follow these steps:

1. In the Explorer tab under Service Performance Manager, select the Templates folder.
   The test templates appear in the Contents panel on the List tab.

   NOTE
   If no templates appear, create a template. For more information, see Create a New Test Template.
2. In the Contents panel on the List tab, select the template(s) to apply.
3. Right-click the selection, and select Apply Test Template.
   The Select Test Host dialog lists test hosts that support the template test type.
4. Select one or more test hosts to which to apply the test template, and click OK.
   OneClick creates the test for the selected test hosts. The tests that were created from the template appear beneath the template in the Explorer tab.

   NOTE
   The List tab in the Contents panel also lists tests that were created from a test host template type when the template is selected.

   Two naming formats are supported for tests that were created from test templates. The test names can indicate the test target IP address, or they can indicate the test target model name. The target can be the test host or a particular device, depending on the template type.
   Use the following default format:

   ```
   Template Name_IP Address
   ```

   – **Template Name**
     Specifies the name of the template that you applied to a test host.
   – **IP Address**
     Specifies the IP address for the test target.

   You can use the SPM Template Naming option from the OneClick Administration Pages to change the naming convention to the following format:

   ```
   Template Name_Model Name
   ```

   – **Template Name**
     Specifies the name of the template that you applied to a test host.
   – **Model Name**
     Specifies the model name for the test target.

   The SPM Template Naming setting does not change the naming format for tests that have already been created from a template.

Apply a Test Template to a Global Collection Container

Applying test templates to Global Collection containers lets you automate the process of creating tests of different types for multiple test hosts. For example, assume a Global Collection container includes a group of test hosts that only

supports ICMP Ping tests, a group that only supports HTTP tests, and a group that supports both. Assume also that numerous test hosts that belong to each group are sometimes added to the container.

In this case, two different test templates can be applied to the container to create the tests for the test hosts that are included in it. One template specifies an ICMP Ping test and the other specifies an HTTP test. ICMP Ping tests are created for the test hosts that support ICMP Ping tests, and HTTP tests are created for the test hosts that support HTTP tests.

This example understates the potential complexity of a “real-life” performance testing implementation. However, it does illustrate the ease with which you can use test templates to set up tests on multiple test hosts. This method is an alternative to setting them up individually. For a more information, see Example Implementation Scenario.

Follow these steps:

1. In the Create Test Template, Copy Test Template, or Edit Test Template dialog, click the Browse button next to the Global Collection parameter.
   The Select Collections dialog opens.
2. Select the Global Collections to which to apply the template, and click OK. If you prefer to create a Global Collection for your test hosts, take the following steps:
   a. Click Create.
   b. Select settings for the new Global Collection.
   c. Proceed with the selection process.

Example Implementation Scenario

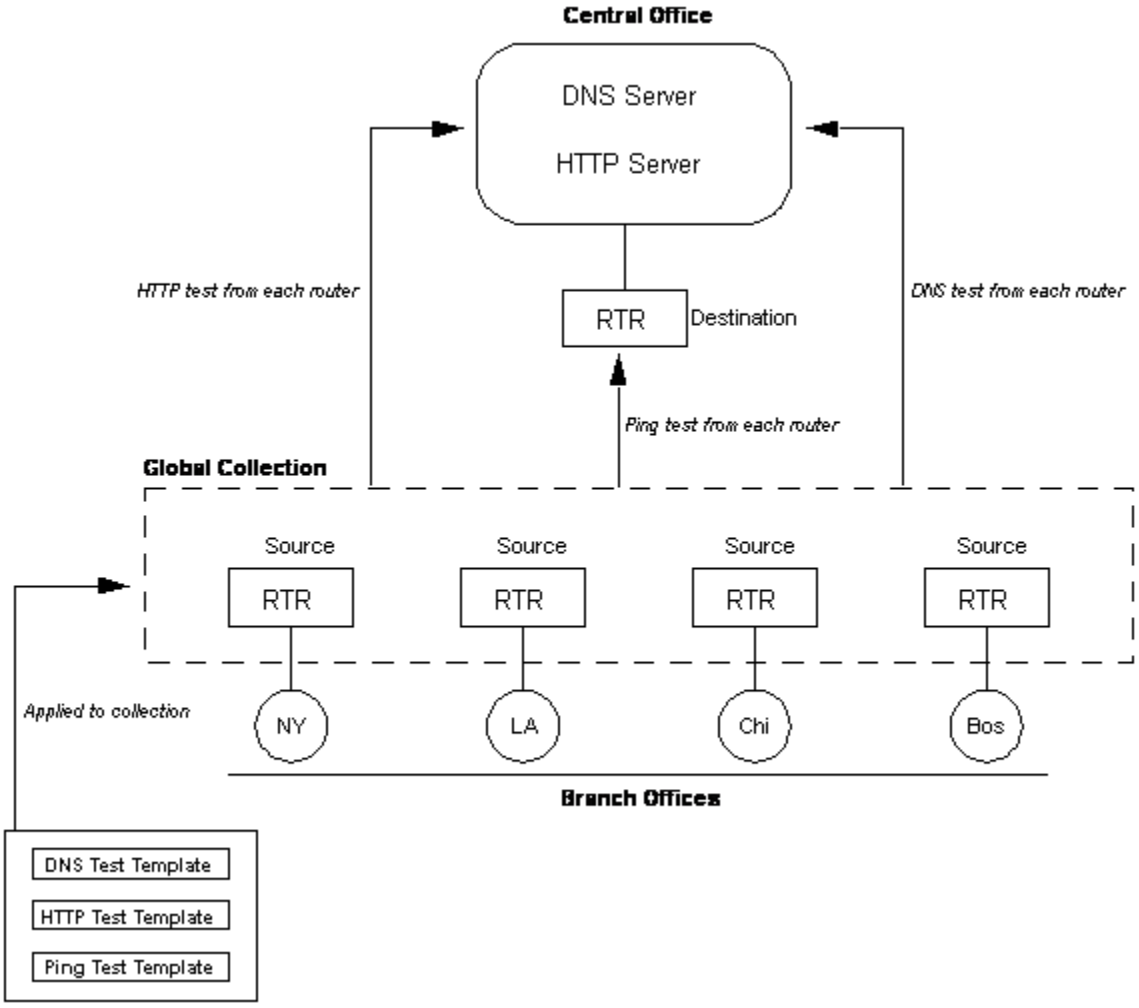
This section illustrates how test templates can be applied to a Global Collection to implement automated bulk performance testing between a central office location and branch office locations.

Scenario

- The central location of an organization provides domain name lookup and web services to numerous remote branch locations that connect to the central location on an intermittent basis.
- You can confirm that remote connections to the central location are maintained and the remote branch locations have continuous access to DNS and HTTP services that are provided by the central location.
- To determine whether service delivery from the main location to the remote locations remains viable, run Ping, DNS, and HTTP tests from remote location edge routers (that support the tests) on a regularly scheduled basis whenever they are brought online (connect to the central location edge router).
- Create a Global Collection that is configured to include all edge router models for remote locations as they are brought online.
- Three test templates are applied to a Global Container:
  – An ICMP Ping test template that is configured to create a Ping test on remote-location edge routers that tests connectivity to the central-location router.
  – A DNS test template that is configured to create a DNS lookup test on remote-location edge routers that tests DNS lookup time from the DNS server at the central location.
  – An HTTP test template that is configured to create an HTTP download test on remote-location edge routers that tests the round-trip time to download a web page from the HTTP server at the central location.

Test Template Scenario

The following image illustrates the example scenario.



About Destination Template Types

Test templates can use the Test Host or the Destination template. A Test Host template specifies a variable test host with a constant destination. A Destination template specifies a constant test host with variable destinations.

When applying a Destination test template to Global Collections, DX NetOps Spectrum populates the Destination Address from the type of model to which the test template is applied. Both port and device models are supported as destinations. These models must exist in or must be added to the target Global Collection for the test template to be applied.

- **Device models**
  When a Destination template is applied to a device model, the network address of the device is used as the Destination Address.
- **Port models**
  When a Destination template is applied to a port model, the IP address of the port is used as the Destination Address.

The Destination Address can be modified after the test is created.

Manage Test Templates

When you edit a template, all the tests that are created from the template are updated with the modified settings. When you remove the association between a template and a Global Collection or delete a template, all tests that are created from the template are deleted.

> **WARNING**
> Editing and deleting tests or test templates should be performed with caution by qualified personnel. This guidance especially applies in cases when, for example, Service Performance Manager performance test results are monitored by SLAs (service level agreements) that are modeled in DX NetOps Spectrum or when tests are run for infrastructure performance analysis.

Edit a Test Template

You can edit all test template parameter settings as required. The changes apply to all tests that you created with the template. You can also remove the association between a template and a Global Collection when you edit a template.

Note: You can remove the association between a template and a Global Collection. Delete the value of the Global Collections parameter in the Edit Test Template dialog. When you remove the association, all tests on the test hosts in the collection are deleted.

Follow these steps:

1. In the Explorer tab under Service Performance Manager, expand the Templates folder.
   The test templates appear.

   > **NOTE**
   > If no templates appear, create a template. For more information, see Create a New Test Template.
2. Right-click the test template that you want to edit and select Edit Test Template.
   The Edit Test Template dialog opens.
3. Modify the template settings and click OK.
   The changes are applied to the template and to all tests that are created from it.

Delete a Test Template

When you delete a template, you also delete all tests that are created from that template.

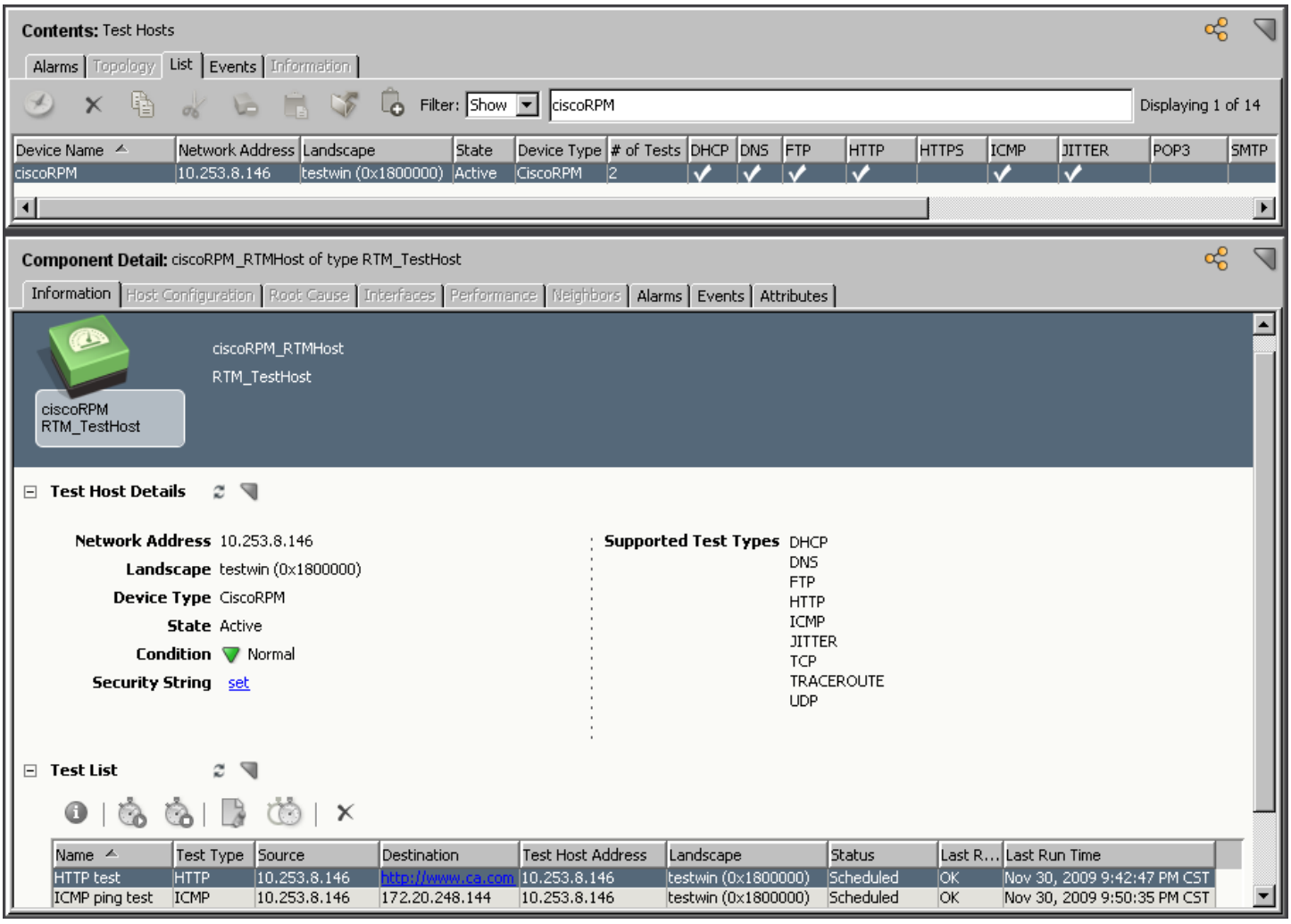
Follow these steps:

1. In the Explorer tab under Service Performance Manager, expand the Templates folder.
   The test templates appear.
2. Right-click the test template that you want to delete and select Delete.
   A confirmation dialog opens.
3. Click Yes.
   The test template is deleted.

Test Host Information

The OneClick Console displays summary and detailed information about test hosts that are modeled in DX NetOps Spectrum and lets you perform operations on tests for test hosts. OneClick provides views of test hosts, test templates, tests, and test results. You can also see detailed information about events and alarms for Service Performance Manager components.

The following image shows an example view of a test host:



Click Refresh in a OneClick view to see current information. You can refresh and customize views and dock and undock panels. You can set up table column preferences to display only the information types that you want to view. For more information, see the Using OneClick section.

The Contents panel displays information about the selected test host model, the landscape where it is modeled, its state (Active or Inactive), and the test types that it supports. From the Contents panel, you can create tests for the test host, discover tests for the test host, and activate and deactivate the test host.

Test Host Information in the Component Detail View

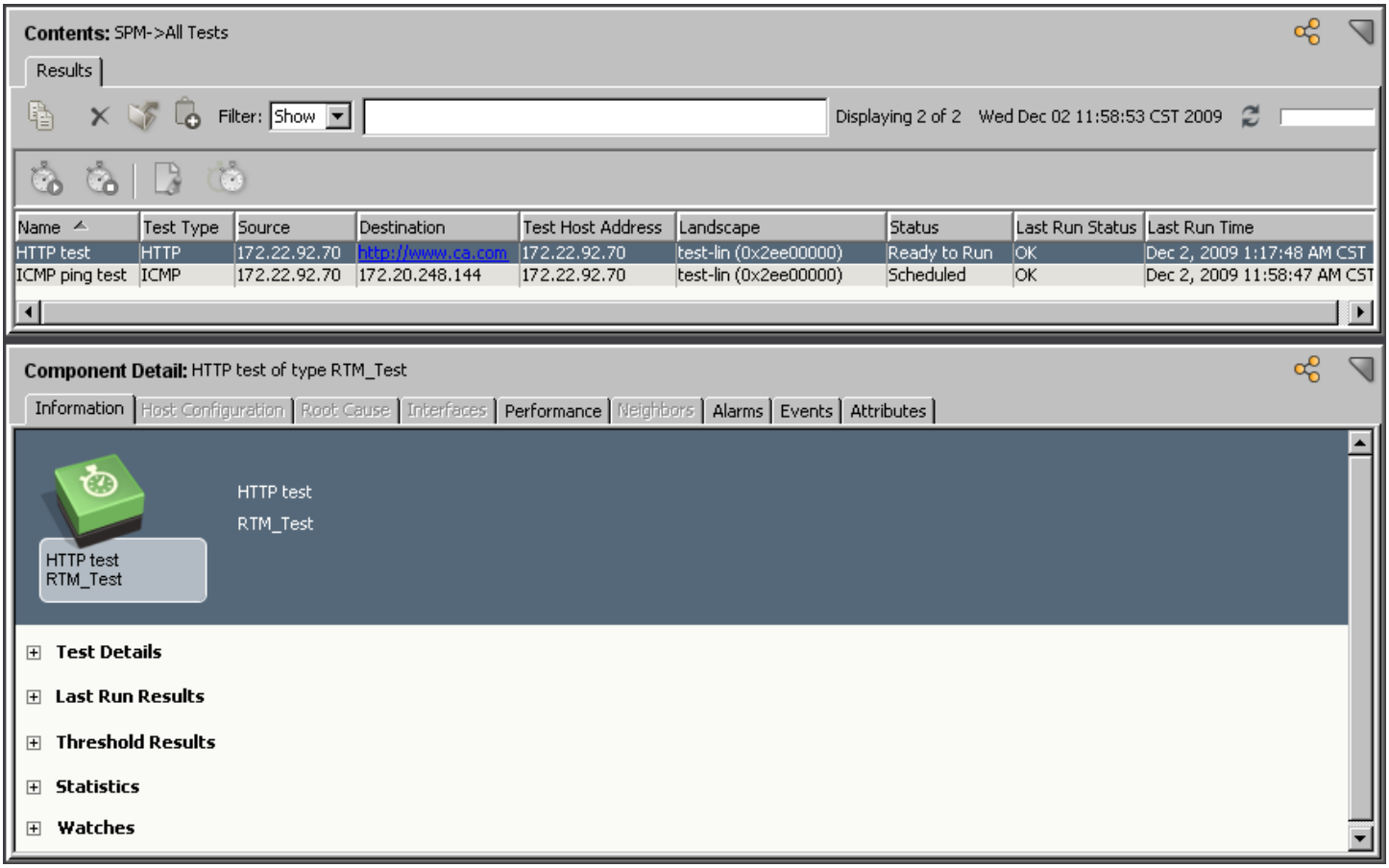
The Component Detail panel provides detailed information about a selected test host. The Information tab of the Component Detail panel includes two categories of information about the test host:

- **Test Host Details**
  Provides detailed information about the test host and lets you set the test host model security string.
- **Test List**
  Displays information about tests for the selected test host. It also lets you run tests, stop tests, manage tests, and invoke the Component Detail view for tests.

  NOTE
  For information about accessing information about alarms and events for Service Performance Manager components, see Alarms and Events.

Test Information

OneClick Console displays summary and detailed information about tests that are modeled in DX NetOps Spectrum and lets you perform operations on tests. The following image shows an example view.



The Contents panel table displays information about the selected test model, the landscape where it is modeled, the IP addresses of the test source, test destination, and test host for the test. It also provides additional information, such as the tests that are scheduled, running, or stopped, the last time the test was run, and the resulting status of the test (did or did not violate a threshold). Command icons for running, stopping, and managing tests are provided too.

Test Information in the Component Detail Panel

The Information tab in the Component Detail panel provides detailed information about a selected test:

- **Test Details**
  Provides information about test configuration settings, including scheduling and test threshold settings.
- **Last Run Results**
  Provides information about the most recent test run, including test results such as latency and packet loss values. Reported metrics vary depending on test host and test type.
- **Threshold Results**
  Provides information about threshold violations (event, minor, major, critical) resulting from the test. Threshold types vary depending on test host and test type.

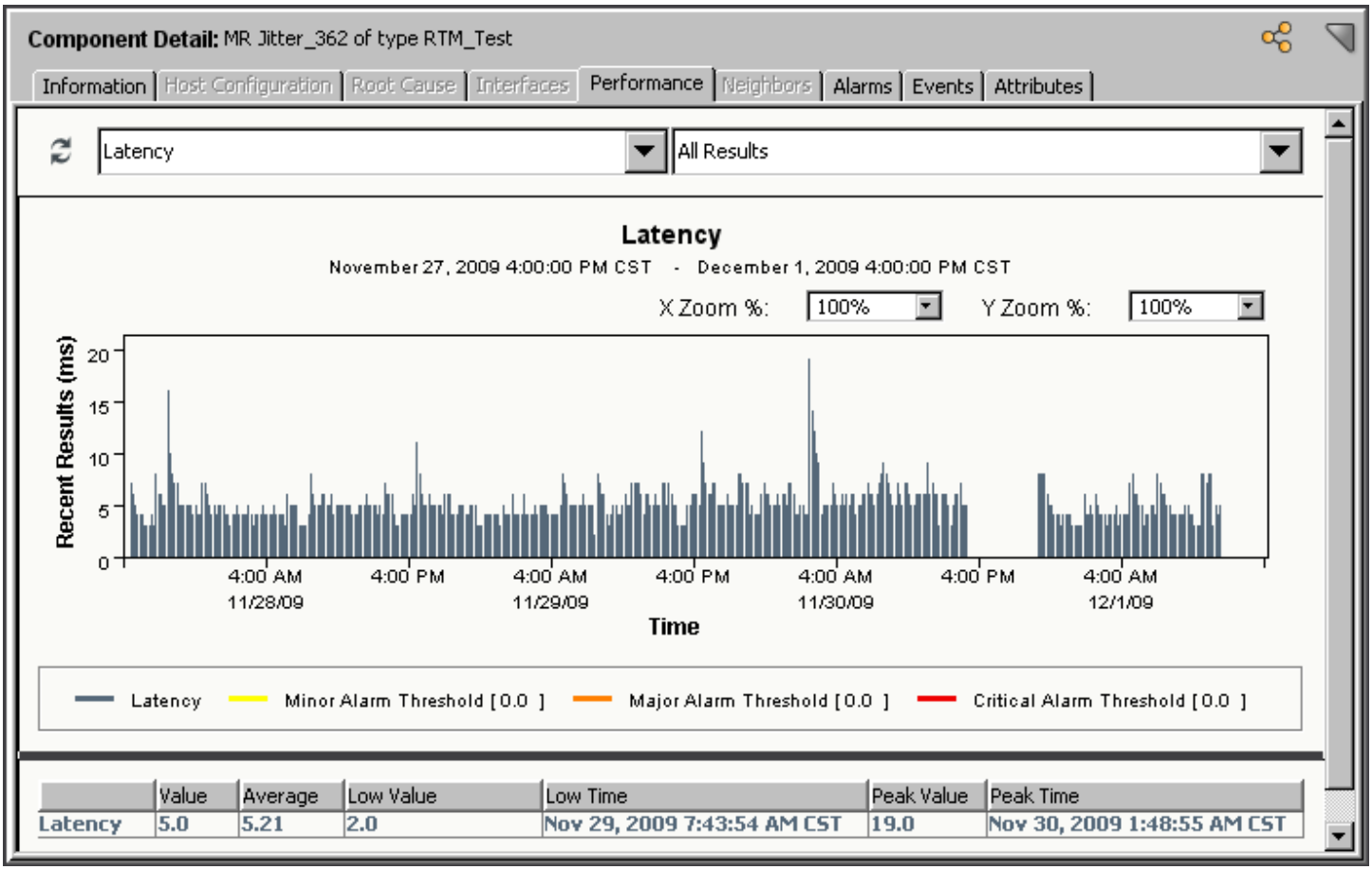
  NOTE
  Regarding the sample count and the % of Threshold values, RTTMON can perform more than the repetitions that are specified for a test. Service Performance Manager includes this count in the results and calculates

the average, maximum, and minimum results for latency and packet loss. Therefore, the value for percentage of packet loss can be other than a multiple of (1 / sample count) * 100.For example: If you set sample count to 5, typically the percentage of packet loss is a value of 0%, 20%, 40%, ...100%. If the agent performs more than 5 repetitions, you can see 0%, 16.66%, 33.33%, ...100%.

- **Statistics**
  Provides statistical test result information for tests that are run for SystemEDGE test hosts.
  Note: This section appears for tests that are created for SystemEDGE test hosts only.
- **Watches**
  Provides information about any watches that have been defined on the test model.

Test Performance Information in the Component Detail Panel

The Performance tab in the Component Detail panel shows graphical representations of test results. The following image shows an example test performance graph.



NOTE
The Archive Manager for the test landscape must be running for Service Performance Manager to display performance test results.

Specify the Interval

You can specify the following time intervals for which you want to display performance results:

- 1 hour
- 3 hours
- 6 hours
- 12 hours
- 1 Day
- 1 Week
- 4 Weeks
- All Results

> **NOTE**
> The All Results option retrieves all result data that is stored in the DX NetOps Spectrum Distributed Data Manager (DDM) database, up to a maximum of 365 days. By default, the DDM stores up to 45 days of data. See the Database Management section for more information.
>
> **WARNING**
> Retrieving large numbers of events can affect performance.

To specify the interval, select an interval from the list.

View Test Data

You can view data from a selected data point in a graph.

To view test data, position your mouse pointer over the end of a data point line in the graph. A descriptive label (including date, time, and value) is displayed for that data point.

You can also view additional test data (averages, high and low values) in tabular form under the graph.

Adjust an Axis

You can adjust the dimensions of the X and Y axes as necessary to meet your viewing requirements. For example, you can adjust the X axis to display latency outliers that are not visible from the default view.

To adjust the dimension of an axis, select a zoom percentage from the list for the X or Y axis, or supply a value.

> **NOTE**
> 100% is the minimum allowable zoom percentage.

Timeout Data Setting Affects a Performance Graph

When you configure a test, you can specify that data generated as a result of a test timeout is filtered out of test results. This setting is enabled by default. Timeouts and test failures can cause gaps in the graph. When timeout filtering is enabled, you cannot discern these gaps. Examples of test failures include lost contact to device, device in maintenance mode, or device failure.

A setting of False, however, causes the following effects:

- Skewed performance graph auto-scaling: Timeout values can so greatly exceed the response time values that exact response times on the performance graph can be difficult to discern.
- Skewed average calculation: Timeout values can so greatly exceed response times that average calculations are inaccurate.
- Skewed data from data export: Timeout values can be prevalent in the exported data. As a result, depending on your post-processing mechanisms, results can be undesirable.

Alarms and Events

You can view alarms for Service Performance Manager components from multiple points in the OneClick Console. Use the Explorer tab to view events and alarms for Service Performance Manager components.

Follow these steps:

1. Select Service Performance Manager in the Explorer tab and click the Alarms tab in the Contents panel.
   All alarms related to Service Performance Manager events appear in the Alarms tab.
2. Select an alarm in the Contents panel.
   The Alarm Details and Events tabs in the Component Detail panel display detailed information for the selected alarm.

You can also view events and alarms for Service Performance Manager components from the Locater tab.

Follow these steps:

1. List the components whose event and alarm information you want to view. For more information, see Finding Components.
2. In the Contents panel, select a component.
3. In the Component Detail panel, click the Alarms or Events tab to view information.

Service Performance Manager Result Data

You can export Service Performance Manager result data that is compiled over an extended period using the SPM result logger. SPM result logger output files include a model handle, timestamp, and a list of test-specific results per line. Once logging is enabled, Service Performance Manager produces text files with result data in SSLogger format. All result files are created and saved in an output directory that you create. New log files are created after an interval that you specify. For information about configuring data export parameters, see the OneClick Administration section.

Data Logging Event Codes

The following table lists Service Performance Manager data logging event codes:

Event Code	Event	Contains Statistics For...
0x04560000	SPM Result Event	Latency and packet loss (DHCP tests, which do not contain data for packet loss, are not included.)
0x0456002e	SPM Result Event (HTTP)	HTTP response time, DNS resolution time,TCP connect time, and HTTP download time
0x04560010	SPM Result Event (Jitter)	Jitter response time, Jitter source to destination time, Jitter destination to source time, Jitter MIA, Jitter late arrival, and Jitter busies
0x0456003e	SPM Result Event (Traceroute)	Latency

Result Events

For SPM result events, the preceding data is followed by statistical data particular to the RTM_Test. The event code dictates which of the statistics are relevant.

> **NOTE**
> Irrelevant statistics for a given test type (for example, Jitter statistics for a ping test) are reported as 0.

The remaining 12 reported statistics in a logged event correspond to, in order:

- Latency
- Packet loss
- HTTP response time
- DNS resolution time
- TCP connect time
- HTTP download time
- Jitter response time
- Jitter source to destination time
- Jitter destination to source time
- Jitter MIA
- Jitter late arrival
- Jitter busies

Data Export Sample

The following sample illustrates the structure of logged Service Performance Manager test result data:

```
0x830201d,6PM-7AM_6.6.0.2,1054176719,0x4560000,1,0.0,0,0,0,0,0,0.0,0.0,0.0,0.0,0.0
0x8302010,Jitter_6.6.0.2,1054176724,0x4560010,0,0.0,0,0,0,0,5,0.0,0.0,0.0,0.0,0.0
0x8302012,FTP_6.6.0.2,1054176730,0x4560000,843,0.0,0,0,0,0,0,0.0,0.0,0.0,0.0,0.0
0x8302017,TCP_6.6.0.2,1054176748,0x4560000,1,0.0,0,0,0,0,0,0.0,0.0,0.0,0.0,0.0
0x830201b,24x7_6.6.0.2,1054176767,0x4560000,1,0.0,0,0,0,0,0,0.0,0.0,0.0,0.0,0.0
0x6a00f8a,TCP_6.6.0.0,1054176941,0x4560000,6,0.0,0,0,0,0,0,0.0,0.0,0.0,0.0,0.0
0x6c009de,Traceroute_6.6.0.1,1054176872,0x456003e,40,0.0,0,0,0,0,0,0.0,0.0,0.0,0.0,0.0
0x6c009d6,FTP_6.6.0.1,1054176872,0x4560000,858,0.0,0,0,0,0,0,0.0,0.0,0.0,0.0,0.0
0x6c009e0,TCP_6.6.0.1,1054176872,0x4560000,12,0.0,0,0,0,0,0,0.0,0.0,0.0,0.0,0.0
0x8302011,HTTP_6.6.0.2,1054176837,0x456002e,0,0.0,753,5,32,716,0,0.0,0.0,0.0,0.0,0.0
0x830201f,6-11M-F_6.6.0.2,1054176870,0x4560000,1,0.0,0,0,0,0,0,0.0,0.0,0.0,0.0,0.0
0x6c009e6,24x7_6.6.0.1,1054176927,0x4560000,5,0.0,0,0,0,0,0,0.0,0.0,0.0,0.0,0.0
0x6c009e8,6PM-7AM_6.6.0.1,1054176937,0x4560000,5,0.0,0,0,0,0,0,0.0,0.0,0.0,0.0,0.0
0x6c009d5,HTTP_6.6.0.1,1054176939,0x456002e,0,0.0,19,0,7,12,0,0.0,0.0,0.0,0.0,0.0
0x6a00f8b,ICMP_6.6.0.0,1054177035,0x4560000,1,0.0,0,0,0,0,0,0.0,0.0,0.0,0.0,0.0
0x6c009df,UDP_6.6.0.1,1054176943,0x4560000,3,0.0,0,0,0,0,0,0.0,0.0,0.0,0.0,0.0
0x8302019,UDP_6.6.0.2,1054176906,0x4560000,7,0.0,0,0,0,0,0,0.0,0.0,0.0,0.0,0.0
0x6c009eb,6-11M-F_6.6.0.1,1054176960,0x4560000,8,0.0,0,0,0,0,0,0.0,0.0,0.0,0.0,0.0
0x6c009dd,ICMP_6.6.0.1,1054176978,0x4560000,4,0.0,0,0,0,0,0,0.0,0.0,0.0,0.0,0.0
```

Test Modification Event

Service Performance Manager test modification events (event code 0x0456000a) contain the following statistical data, in addition to the common data described in Data Logging Event Codes:

- Test name
- Destination address
- Packet size
- Source address
- Test interval
- Sample count

Using the Command Line Interface (CLI) to Manage Tests

This section describes the procedure to use the Command Line Interface.

CLI Response-Time Testing

Response time tests are represented in DX NetOps Spectrum as models of model type RTM_Test. You can use the DX NetOps Spectrum Command Line Interface (CLI) to create, run, and edit response time tests and get test results as an alternative method to using the user interface in OneClick. For more information, see Working with Performance Tests. You can issue test management commands from the CLI command line, or you can embed CLI commands in scripts, which you can run on an ad hoc or scheduled basis.

Important! Before you attempt to create and manage tests with the CLI, understand Service Performance Manager concepts, CLI commands and command syntax, and DX NetOps Spectrum modeling concepts.

Create Tests in CLI

You can create any of the response time test models that DX NetOps Spectrum supports using the CLI create command. For more information, see Required Parameters for Creating Tests.

The following example script shows how to create a regularly scheduled Ping test. It enables a latency threshold, a latency value that, if exceeded, triggers a minor alarm, and a latency value that, if not exceeded, triggers an alarm clear.

```
#!/bin/ksh
cd $SPECROOT/vnmsh
connect
./create model mth=0x4560000 \
attr=0x1006e,val="Ping_Test_1" \ # Test Name
attr=0x4560005,val=0 \ # Test Type is ICMP_Ping
attr=0x45600f1,val=10.253.9.8 \ # TestHost address
attr=0x456001f,val=10.253.9.12 \ # Destination IP
attr=0x4560022,val=1 \ # Schedule is enabled
attr=0x4560014,val=600 \ # Test interval is 10 minutes
attr=0x4560035,val=1 \ # Latency Threshold is enabled
attr=0x456009b,val=100 \ # Latency Minor Threshold Set value
attr=0x4560017,val=100 \ # Latency Threshold Clear value
attr=0x4560027,val=2 # Clear Cycles
```

The following example script shows how to create an HTTP test. Note the formatting that is required for URL values.

```
#!/bin/ksh
cd $SPECROOT/vnmsh
connect
./create model mth=0x4560000 \
attr=0x1006e,val="Http_Test_1" \ # Test Name
attr=0x4560005,val=5 \ # Test Type is HTTP
attr=0x45600f1,val=10.253.9.8 \ # TestHost address
attr=0x456000f,val="http:\/\/www.ca.com\/about.htm" # Destination URL
attr=0x456008d,val="http:\/\/proxyServer" # Proxy URL
attr=0x4560022,val=1 \ # Schedule is enabled
```

Discover Tests in CLI

You can use the CLI update command to run SPM Test Discovery to discover and model performance tests that have been configured on test hosts but not with Service Performance Manager. Specify the discover tests action code (0x4560007) and the test host model handle. The following example script shows how to run SPM Test Discovery for a single test host.

NOTE
For more test action codes, see Test Action Codes.

```
#!/bin/ksh
# this will discover tests for the test host with the given model handle.
cd $SPECROOT/vnsmh
connect
./update action=0x4560007 mh=<testhostMH>
```

Run Tests in the CLI

You can run a response-time test using the CLI update command. Specify the run test action code (0x4560009) and the test model handle. For more information, see Test Action Codes.

The following example script shows how to run a single test. You can run multiple tests by specifying multiple tests in the script.

```
#!/bin/ksh
# this will run a test with the given model handle.
cd $SPECROOT/vnsmh
connect
./update action=0x4560009 mh=testMH
```

Edit Tests in the CLI

You can edit a response time test using an update command in the CLI. Change the values of test parameters by specifying new parameter values and issuing an update test action code (0x4560008).

The following example script shows how to modify a test schedule interval and test timeout values.

```
#!/bin/ksh

# change the schedule interval from 15 minutes to 30 minutes (1800 seconds)
# change the test timeout from 5000 ms to 1000 ms
# have the test not filter its timeout data

cd $SPECROOT/vnmsh
connect
./update mh=<testMH> attr=0x4560014,val=1800
./update mh=<testMH> attr=0x4560025,val=1000
./update mh=<testMH> attr=0x45600d6,val=FALSE
./update action=0x4560008 mh=$i
```

Get Test Results in CLI

You can get various test status indications and test results using the CLI show command. This command also returns the parameters with the values that you want.

The following example script shows how to get test status and test results. For more information, see Test Status and Test Results Parameters.

```
#!/bin/ksh
# obtain the latestResult status and latency/packet loss values for a given test
cd $SPECROOT/vnmsh
connect
./show attributes -e attr=0x4560004 attr=0x4560015 attr=0x456007d mh=<testMH>
```

Test Parameters Used in the CLI

The topics in this section list and describe parameters for creating and scheduling tests and setting test thresholds. You can configure RTM_Test model parameters using the user interface in OneClick. For more information, see Configure Tests.

Parameters for Creating Tests

Required Parameters

For All Test Types

The following parameters are required for all response time tests:

- **Model_Name (0x1006e)**
  Specifies the name of the test. Only one test per host can exist with the same name. Duplicate test names are appended with "_COPY".
- **Test_Type (0x4560005)**
  Specifies the type of test to create. Once a test is created, this value cannot be changed. If it is changed, it is reset when the update takes place.
  a. 0 = ICMP
  b. 1 = UDP
  c. 2 = Trace Route
  d. 3 = TCP
  e. 4 = DNS
  f. 5 = HTTP
  g. 6 = POP3
  h. 7 = DHCP
  i. 8 = FTP
  j. 9 = SMTP
  k. 10 = Jitter
  l. 13 = Custom
  m. 14 = SQL Query
  Default: None
- **Test_Host_Address (0x45600f1)**
  Specifies the IP address of a test host. This address must match the network address of the associated device model. Once a test is created, this value cannot be changed. If the value changes, it is reset when the update takes place.

For Specific Test Types

The following parameters are required when creating certain response time tests, depending on the test type:

- **Connect_String (0x456010a)**

Specifies a string of commands to connect to the database. This parameter is required for some SQL Query tests, depending on database type.
Example:

```
jdbc:mysql://172.22.246.43/mysql?user=root&password=root
```

- **Database_Name (0x4560108)**
  Specifies the name of the database. This parameter is required for some SQL Query tests, depending on the database type.
- **Database_Type (0x456010b)**
  Specifies the type of database to test. This parameter is required for SQL Query tests. Correct drivers must be installed on the CA eHealth SystemEDGE computer.
  a. 0 = Oracle
  b. 1 = Microsoft SQL
  c. 2 = Other
- **Dest_File_Name (0x456000d)**
  Specifies the destination filename. This parameter is required for FTP tests.
- **Dest_Host_Name (0x456000a)**
  Specifies the destination test hostname. For DNS tests, it is the lookup name; for FTP and POP3 tests, it is the address where the transaction is performed. For Custom tests, it is the name and location of a valid script.
- **Dest_IP_Address (0x456001f)**
  Specifies the destination IP address. This parameter is used for DNS, ICMP, Jitter, SMTP, TCP, Trace Route, and UDP tests.
- **Dest_Password (0x456000e)**

  WARNING
  The Dest_Password parameter is used for FTP, HTTP, HTTPS, POP3, SMTP, and SQL Query tests. You cannot use the CLI or the REST API for this parameter, as this password value is encrypted in the DX NetOps Spectrum database. Use the OneClick console to enter this value.
- **Dest_Port_Number (0x4560011)**
  Specifies the port number where the service is running for FTP, Jitter, POP3, SMTP, TCP, and UDP tests. For Mean Opinion Score (MOS) support in Jitter tests, the destination port must be an even-numbered port in the range 16384 - 32766 or 49152 - 65534.
  Note: Dest_Port_Number is supported for FTP, POP3, and SMTP tests only for CA eHealth SystemEDGE hosts.
- **Dest_URL (0x456000f)**
  Specifies the destination URL required for HTTP tests. Enclose the URL between double quotes (“ ”), and use escape slashes (\) before forward slashes (/). See Create Tests in CLI for a format example.
- **Dest_User_Name (0x456000b)**
  Specifies the destination user name, which is required for FTP and POP3 tests and optional for HTTP and HTTPS tests. For SMTP tests, this required parameter is the email address that you want to test. For SQL Query tests, this required parameter is the username for database access.
- **Query_String (0x456010c)**
  Specifies the query statement to execute. This parameter is required for SQL Query tests.
- **SQL_Driver (0x4560109)**
  Specifies the name of the SQL driver, which is required for some SQL Query tests, depending on the database type.
  Example:

```
com.mysql.jcbc.Driver
```

Optional Parameters

You can use optional parameters to specify various optional test parameters.

NOTE
Configure Advanced Parameters describes how to specify these parameters using the Service Performance Manager user interface in OneClick.

- **CodecType (0x45600e7)**
  Specifies the type of codec that is used by the router to perform audio compression and decompression. This parameter is important for calculating the Mean Option Score (MOS).
  a. 0 = None
  b. 1 = G.711 U-law
  c. 2 = G.711 A-law
  d. 3 = G.729A
  Default: 0
- **DeleteMessages (0x45600f5)**
  Specifies whether to delete the messages that were downloaded during the test or to leave the messages on the test system.
  Default: False
  Note: DeleteMessages is supported for POP3 tests for CA eHealth SystemEDGE hosts only.
- **DownloadContent (0x45600fd)**
  Specifies whether to download all images, frames, scripts, and applets, with the core HTML code from the website or URL.
  Default: False
  Note: DownloadContent is supported for HTTP and HTTPS tests for CA eHealth SystemEDGE hosts only.
- **DownloadType (0x45600f4)**
  Specifies whether the first or all messages are downloaded for POP3 tests.
  a. 1 = Download only the first message for this user account.
  b. 2 = Download all messages for this user account.
  Default: 1
  Note: DownloadType is supported for POP3 tests for CA eHealth SystemEDGE hosts only.
- **EchoAdminSourceAddress (0x45600b0)**
  Specifies the IP address or hostname of the test when it is not the test host. For more information, see About the Test Host Location Parameter.

  NOTE
  EchoAdminSourceAddress is supported for Cisco hosts only.
- **EchoAdminSourcePort (0x45600b1)**
  Specifies the port number that is used by the tested application (Jitter, TCP, UDP only).
  Limits: Must be less than 65536
  Note: EchoAdminSourcePort is supported for Cisco hosts only.
- **FailOnContentError (0x45600fe)**
  Specifies whether any errors that are encountered while downloading images, frames, scripts, and applets cause the test to fail.
  Default: False
  Note: FailOnContentError is supported for HTTP and HTTPS tests for CA eHealth SystemEDGE hosts only.
- **FILTER_TIMEOUT_DATA (0x45600d6)**
  Specifies whether the RTM_Test generates result events for timeouts.
  Default: True
- **FrameDepth (0x45600fa)**
  Specifies the number of levels the test should traverse when downloading nested frames. The HTTP and HTTPS tests download all frames, images, external scripts, and applets during the page download. The measurement reflects your experience when downloading a web page.
  Default: 3

Note: FrameDepth is supported for HTTP and HTTPS for CA eHealth SystemEDGE hosts only.

- **MailBodySize (0x45600f6)**
  Specifies the size (in bytes) of the test message to send.
  Default: 1000
  Note: MailBodySize is supported for SMTP tests for CA eHealth SystemEDGE hosts only.
- **MinMatches (0x45600fc)**
  Specifies the minimum number of times that the search expression can be found. The test fails if the search expression is not found for the specified number of times.
  Default: 1
  Note: MinMatches is supported for HTTP and HTTPS tests for CA eHealth SystemEDGE hosts only.
- **OperationType (0x45600f3)**
  Specifies the type of FTP operation to test.
  a. 1 = Login
  b. 2 = Get
  c. 3 = Put
  Default: 1
  Note: OperationType is supported for FTP tests for CA eHealth SystemEDGE hosts only.
- **OtherUserName (0x45600f8)**
  Specifies a username. For SMTP tests for CA eHealth SystemEDGE hosts, this is the outgoing username for SMTP authentication. For HTTP tests for Cisco routers and HTTP and HTTPS tests for CA eHealth SystemEDGE hosts, it is a valid username to be authenticated on the specified proxy server.
- **OtherPassword (0x45600f9)**

  WARNING
  The OtherPassword parameter is used for the Outgoing Password value for SMTP tests for CA eHealth SystemEDGE hosts. It is also used for the Proxy Password value for HTTP and HTTPS tests for Cisco routers and CA eHealth SystemEDGE hosts. Because this password value is encrypted in the DX NetOps Spectrum database, you cannot use CLI for this parameter. Use the OneClick Console to enter this value.
- **Packet_Size (0x4560067)**
  Specifies the value (expressed in octets) that limits the size of the packets that are used in the test.
  Default: 64
- **Proxy_URL (0x456008d)**
  Specifies a proxy URL or server.
  For Proxy URL, enclose the URL between double quotes (“ ”), and use escape slashes (\ ) before forward slashes (/).
  See Create Tests in CLI for a format example using a URL.
  For Proxy Server (for SystemEDGE hosts only), use the format <*server*>[:*port*].
  Note: Proxy_URL is supported for HTTP and HTTPS tests only.
- **Sample_Count (0x4560068)**
  Specifies the number of times during a test run that the test is performed.
  Default: 5
- **Source_IP_Address (0x45600f2)**
  Specifies the source IP address.

  NOTE
  Source_IP_Address is supported for Mid_Path/Extended_Path Ping tests only.
- **State (0x4560003)**
  Specifies the test state:
  a. 1 = Enabled
  b. 2 = Disabled
  Default: 1
- **Test_Host_Position (0x4560030)**
  Specifies test host location.

a. 0 = End point
b. 1 = Mid path
c. 2 = Extended path
Default: 0

- **TextMatch (0x45600fb)**
  Specifies a regular expression or text string to match on the pages you test.
  Note: TextMatch is supported for HTTP and HTTPS tests for CA eHealth SystemEDGE hosts only.
- **Thresh_Model (0x4560024)**
  Specifies the test entity where threshold events are asserted.
  a. 0 = Test
  b. 1 = Source
  c. 2 = Destination
  d. 3 = Host
  Default: 0
- **THRESH_SCHED_MH (0x4560090)**
  Specifies the Model Handle of a Schedule model. This parameter is used with the ThreshSchedule_Type parameter. Not specifying a value results in a 7x24 schedule.
- **ThreshSchedule_Type (0x4560090)**
  Specifies preconfigured threshold schedules for periods during which you want to see Service Performance Manager threshold alarms. Not specifying a value results in a 7x24 schedule.
  a. 0 = 7x24
  b. 1 = 7A-6P
  c. 2 = 6P-7A
  d. 3 = MF 8A-8P
  e. 4 = MF 6A-11P
  Default: 0
- **TypeOfService (0x4560099)**
  Specifies the Type of Service (TOS) octet in an IP datagram header that enables packets with different TOS values to be routed differently.
  Limits: Must be less than 256
  Default: 0
- **UseSSL (0x45600f7)**
  Specifies whether to enable Secure Sockets Layer security in case the SMTP server requires SSL authentication.
  Default: False
  Note: UseSSL is supported for SMTP tests for CA eHealth SystemEDGE hosts only.

Required Parameters for Scheduling Tests

Use these parameters to specify test timeout and test schedule management values.

> **NOTE**
> Schedule a Test explains the procedure to configure test schedule parameters using the Service Performance Manager user interface in OneClick.

- **ACTIVE_SCHED_MH (0x456008f)**
  Specifies the model handle of a test schedule model. Use this parameter as an alternative to TestSchedule_Type. Not specifying a value results in a 7x24 schedule.
- **Schedule_State (0x4560000)**
  Specifies whether the test schedule is in effect.
  a. 0 = Disables
  b. 1 = Enabled

Default: 0

- **Sched_Frequency (0x4560014)**
  Specifies the interval in seconds between scheduled test runs.
  Default: 5000
- **Test_Timeout (0x4560025)**
  Specifies the number of milliseconds before a test connection to an unresponsive test host times out.
  Default: 5000
- **TestSchedule_Type (0x456008b)**
  Specifies preconfigured test schedules. Use this parameter as an alternative to ACTIVE_SCHED_MH. Not specifying a value results in a 7x24 schedule.
  a. 0 = 7x24
  b. 1 = 7A-6P
  c. 2 = 6P-7A
  d. 3 = MF 8A-8P
  e. 4 = MF 6A-11P
  Default: 5
- **Thresh_Frequency (0x456001a)**
  Specifies an interval in seconds after which a test runs while in a threshold condition.
  Default: 300

Threshold Type Parameters

You can specify threshold management parameters for all test types. Thresholds are expressed either in terms of response, or transaction time, or packet error or loss, depending on the type of test for which the threshold is specified.

Latency Threshold Parameters

Latency_Thresh_State (0x4560035)

Specifies whether the latency threshold is enabled or disabled for a test.

1. 0 = Disabled
2. 1 = Enabled

Default: 0

Latency_MinorSetValue (0x456009b)

Specifies the latency threshold period in milliseconds that must be exceeded before a minor alarm is generated.

Default: 500

Latency_MajorSetValue (0x456009c)

Specifies the latency threshold period in milliseconds that must be exceeded before a major alarm is generated.

Default: None

Latency_CriticalSetValue (0x456009d)

Specifies the latency threshold period in milliseconds that must be exceeded before a critical alarm is generated.

Default: None

Thresh_Set_Value (0x4560016)

Specifies the latency threshold period in milliseconds that must be exceeded before an event is generated.

Default: None

Thresh_Clear_Value (0x4560017)

Specifies the latency threshold period in milliseconds the test must not exceed before an event is cleared.

Default: 500

Thresh_Set_Delay (0x4560026)

Specifies the number of consecutive cycles the test must run in violation of the threshold before an event or alarm is generated.

Default: 1

Thresh_Clear_Delay (0x4560027)

Specifies the number of consecutive cycles the test must run without exceeding the threshold before an event or alarm is cleared.

Default: 1

Packet Loss Threshold Parameters

PL_Thresh_State (0x4560034)

Specifies whether the packet loss threshold is enabled or disabled for a test.

1. 0 = Disabled
2. 1 = Enabled

Default: 0

PacketLoss_MinorSetValue (0x456009e)

Specifies the packet lost percentage that must be exceeded before a minor alarm is generated.

Default: 20percent

PacketLoss_MajorSetValue (0x456009f)

Specifies the packet lost percentage that must be exceeded before a major alarm is generated.

Default: None

PacketLoss_CriticalSetValue (0x45600a0)

Specifies the packet lost percentage that must be exceeded before a critical alarm is generated.

Default: None

PL_Set_Value (0x456002c)

Specifies the packet lost percentage that must be exceeded before an event is generated.

Default: None

PL_Clear_Value (0x456002e)

Specifies the packet lost percentage that must not be exceeded before a minor alarm is generated.

Default: 20percent

PL_Set_Delay (0x456002d)

Specifies the number of consecutive cycles the test must run in violation of the threshold before an event or alarm is generated.

Default: 1

PL_Clear_Delay (0x456002f)

Specifies the number of consecutive cycles the test must run without exceeding the threshold before an event or alarm is cleared.

Default: 1

HTTP DNS Threshold Parameters

Statistic_1_Thresh_State (0x4560036)

Specifies whether the HTTP DNS threshold is enabled or disabled for a test.

1. 0 = Disabled
2. 1 = Enabled

Default: 0

Statistic_1_MinorSetValue (0x45600a1)

Specifies the resolution time threshold period in milliseconds that must be exceeded before a minor alarm is generated.

Default: 500

Statistic_1_MajorSetValue (0x45600a2)

Specifies the resolution time threshold period in milliseconds that must be exceeded before a major alarm is generated.

Default: None

Statistic_1_CriticalSetValue (0x45600a3)

Specifies the resolution time threshold period in milliseconds that must be exceeded before a critical alarm is generated.

Default: None

Statistic_1_Set_Value (0x4560039)

Specifies the resolution time threshold period in milliseconds that must be exceeded before an event is generated.

Default: None

Statistic_1_Clear_Value (0x456003a)

Specifies the resolution time threshold the test must not exceed before an event is cleared.

Default: 500

Statistic_1_Set_Delay (0x456003b)

Specifies the number of consecutive cycles the test must run in violation of the threshold before an event or alarm is generated.

Default: 1

Statistic_1_Clear_Delay (0x456003c)

Specifies the number of consecutive cycles the test must run without exceeding the threshold before an event or alarm is cleared.

Default: 1

HTTP TCP Threshold Parameters

Statistic_2_Thresh_State (0x456003e)

Specifies whether the HTTP TCP threshold is enabled or disabled for a test.

1. 0 = Disabled
2. 1 = Enabled

Default: 0

Statistic_2_MinorSetValue (0x45600a4)

Specifies the connection time threshold period in milliseconds that must be exceeded before a minor alarm is generated.

Default: 500

Statistic_2_MajorSetValue (0x45600a5)

Specifies the connection time threshold period in milliseconds that must be exceeded before a major alarm is generated.

Default: None

Statistic_2_CriticalSetValue (0x45600a6)

Specifies the connection time threshold period in milliseconds that must be exceeded before a critical alarm is generated.

Default: None

Statistic_2_Set_Value (0x4560041)

Specifies the connection time threshold period in milliseconds that must be exceeded before an event is generated.

Default: None

Statistic_2_Clear_Value (0x4560042)

Specifies the connection time threshold that must not exceed before an event is cleared.

Default: 500

Statistic_2_Set_Delay (0x4560043)

Specifies the number of consecutive cycles the test must run in violation of the threshold before an event or alarm is generated.

Default: 1

Statistic_2_Clear_Delay (0x4560044)

Specifies the number of consecutive cycles the test must run without exceeding the threshold before an event or alarm is cleared.

Default: 1

HTTP Download Threshold Parameters

Statistic_3_Thresh_State (0x4560046)

Specifies whether the HTTP download threshold is enabled or disabled for a test.

1. 0 = Disabled
2. 1 = Enabled

Default: None

Statistic_3_MinorSetValue (0x45600a7)

Specifies the download time threshold period in milliseconds that must be exceeded before a minor alarm is generated.

Default: 500

Statistic_3_MajorSetValue (0x45600a8)

Specifies the download time threshold period in milliseconds that must be exceeded before a major alarm is generated.

Default: None

Statistic_3_CriticalSetValue (0x45600a9)

Specifies the download time threshold period in milliseconds that must be exceeded before a critical alarm is generated.

Default: None

Statistic_3_Set_Value (0x4560049)

Specifies the download time threshold period in milliseconds that must be exceeded before an event is generated.

Default: None

Statistic_3_Clear_Value (0x456004a)

Specifies the download time latency threshold the test must not exceed before an event is cleared.

Default: 500

Statistic_3_Set_Delay (0x456004b)

Specifies the number of consecutive cycles the test must run in violation of the threshold before an event or alarm is generated.

Default: 1

Statistic_3_Clear_Delay (0x456004c)

Specifies the number of consecutive cycles the test must run without exceeding the threshold before an event or alarm is cleared.

Default: 1

Jitter Source to Destination Packet Loss Threshold

Statistic_1_Thresh_State (0x4560036)

Specifies whether the Jitter source to destination packet loss threshold is enabled or disabled for a test.

1. 0 = Disabled
2. 1 = Enabled

Default: 0

Statistic_1_MinorSetValue (0x45600a1)

Specifies the percentage of packets that must be lost before a minor alarm is generated.

Default: 20percent

Statistic_1_MajorSetValue (0x45600a2)

Specifies the percentage of packets that must be lost before a major alarm is generated.

Default: None

Statistic_1_CriticalSetValue (0x45600a3)

Specifies the percentage of packets that must be lost before a critical alarm is generated.

Default: None

Statistic_1_Set_Value (0x4560039)

Specifies the percentage of packets that must be lost before an event is generated.

Default: None

Statistic_1_Clear_Value (0x456003a)

Specifies the percentage of packets lost that cannot be exceeded before an event or alarm is cleared.

Default: 20percent

Statistic_1_Set_Delay (0x456003b)

Specifies the number of consecutive cycles the test must run in violation of a threshold before an event or alarm is generated.

Default: 1

Statistic_1_Clear_Delay (0x456003c)

Specifies the number of consecutive cycles the test must run without exceeding the threshold before an event or alarm is cleared.

Default: 1

Jitter Destination to Source Packet Loss Threshold

Statistic_2_Thresh_State (0x456003e)

Specifies whether the Jitter destination to source packet loss threshold is enabled or disabled for a test.

1. 0 = Disabled
2. 1 = Enabled

Default: 0

Statistic_2_MinorSetValue (0x45600a4)

Specifies the packet lost percentage that must be exceeded before a minor alarm is generated.

Default: 20percent

Statistic_2_MajorSetValue (0x45600a5)

Specifies the packet lost percentage that must be exceeded before a major alarm is generated.

Default: None

Statistic_2_CriticalSetValue (0x45600a6)

Specifies the packet lost percentage that must be exceeded before a critical alarm is generated.

Default: None

Statistic_2_Set_Value (0x4560041)

Specifies the packet lost percentage that must be exceeded before an event is generated.

Default: None

Statistic_2_Clear_Value (0x4560042)

Specifies the packet lost percentage that must not be exceeded before a minor alarm is generated.

Default: 20percent

Statistic_2_Set_Delay (0x4560043)

Specifies the number of consecutive cycles the test must run in violation of the threshold before an event or alarm is generated.

Default: 1

Statistic_2_Clear_Delay (0x4560044)

Specifies the number of consecutive cycles the test must run without exceeding the threshold before an event or alarm is cleared.

Default: 1

Jitter MIA Threshold Parameters

Statistic_3_Thresh_State (0x4560046)

Specifies whether the Jitter MIA threshold is enabled or disabled for a test.

1. 0 = Disabled
2. 1 = Enabled

Default: 0

Statistic_3_MinorSetValue (0x45600a7)

Specifies the percentage of missing in action packets that must be exceeded before a minor alarm is generated.

Default: 20percent

Statistic_3_MajorSetValue (0x45600a8)

Specifies the percentage of missing in action packets that must be exceeded before a major alarm is generated.

Default: None

Statistic_3_CriticalSetValue (0x45600a9)

Specifies the percentage of missing in action packets that must be exceeded before a critical alarm is generated.

Default: None

Statistic_3_Set_Value (0x4560049)

Specifies the percentage of missing in action packets that must be exceeded before an event is generated.

Default: None

Statistic_3_Clear_Value (0x456004a)

Specifies the percentage of missing in action packets that must not be exceeded before an event or alarm is cleared.

Default: 20percent

Statistic_3_Set_Delay (0x456004b)

Specifies the number of consecutive cycles the test must exceed the threshold before an event or alarm is generated.

Default: 1

Statistic_3_Clear_Delay (0x456004c)

Specifies the number of consecutive cycles the test must not exceed the threshold before an event or alarm is generated.

Default: 1

Jitter Late Arrival Threshold Parameters

Statistic_4_Thresh_State (0x456004e)

Specifies whether the Jitter late arrival threshold is enabled or disabled for a test.

1. 0 = Disabled
2. 1 = Enabled

Default: 0

Statistic_4_MinorSetValue (0x45600aa)

Specifies the percentage of late packets that must be exceeded before a minor alarm is generated.

Default: None

Statistic_4_MajorSetValue (0x45600ab)

Specifies the percentage of late packets that must be exceeded before a major alarm is generated.

Default: 20percent

Statistic_4_CriticalSetValue (0x45600ac)

Specifies the percentage of late packets that must be exceeded before a critical alarm is generated.

Default: None

Statistic_4_Set_Value (0x4560051)

Specifies the percentage of late packets that must be exceeded before an event is generated.

Default: None

Statistic_4_Clear_Value (0x4560052)

Specifies the percentage of late packets that must not be exceeded before an event or alarm is cleared.

Default: None

Statistic_4_Set_Delay (0x4560053)

Specifies the number of consecutive cycles the test must run in violation of the threshold before an event or alarm is generated.

Default: 20percent

Statistic_4_Clear_Delay (0x4560054)

Specifies the number of consecutive cycles the test must run without exceeding the threshold before an event or alarm is cleared.

Default: 1

Jitter Busies Threshold Parameters

Statistic_5_Thresh_State (0x4560056)

Specifies whether the Jitter busies threshold is enabled or disabled for a test.

1. 0 = Disabled
2. 1 = Enabled

Default: 0

Statistic_5_MinorSetValue (0x45600ad)

Specifies the percentage of busy failures that must be exceeded before a minor alarm is generated.

Default: 20

Statistic_5_MajorSetValue (0x45600ae)

Specifies the percentage of busy failures that must be exceeded before a major alarm is generated.

Default: None

Statistic_5_CriticalSetValue (0x45600af)

Specifies the percentage of busy failures that must be exceeded before a critical alarm is generated.

Default: None

Statistic_5_Set_Value (0x4560059)

Specifies the percentage of busy failures that must be exceeded before an event is generated.

Default: None

Statistic_5_Clear_Value (0x456005a)

Specifies the percentage of busy failures that must not be exceeded before an event is generated.

Default: 20

Statistic_5_Set_Delay (0x456005b)

Specifies the number of consecutive cycles the test must run in violation of the threshold before an event or alarm is generated.

Default: 1

Statistic_5_Clear_Delay (0x456005c)

Specifies the number of consecutive cycles the test must run without exceeding the threshold before an event or alarm is cleared.

Default: 1

Jitter Positive Source to Destination Threshold Parameters

Statistic_6_Thresh_State (0x45600ba)

Specifies whether the Jitter positive source to destination threshold is enabled or disabled for a test.

1. 0 = Disabled
2. 1 = Enabled

Default: 0

Statistic_6_MinorSetValue (0x45600b4)

Specifies the percentage of positive Jitter that must be exceeded before a minor alarm is generated.

Default: 25

Statistic_6_MajorSetValue (0x45600b5)

Specifies the percentage of positive Jitter that must be exceeded before a major alarm is generated.

Default: None

Statistic_6_CriticalSetValue (0x45600b6)

Specifies the percentage of positive Jitter that must be exceeded before a critical alarm is generated.

Default: None

Statistic_6_Set_Value (0x45600b3)

Specifies the percentage of positive Jitter that must be exceeded before an event is generated.

Default: None

Statistic_6_Clear_Value (0x45600b7)

Specifies the percentage of positive Jitter that must not be exceeded before an event is generated.

Default: 25

Statistic_6_Set_Delay (0x45600b8)

Specifies the number of consecutive cycles the test must run in violation of the threshold before an event or alarm is generated.

Default: 1

Statistic_6_Clear_Delay (0x45600b9)

Specifies the number of consecutive cycles the test must run without exceeding the threshold before an event or alarm is cleared.

Default: 1

Jitter Positive Destination to Source Threshold Parameters

Statistic_7_Thresh_State (0x45600ce)

Specifies whether the Jitter positive destination to source threshold is enabled or disabled for a test.

0 = Disabled

1 = Enabled

Default: 0

Statistic_7_MinorSetValue (0x45600bd)

Specifies the percentage of positive Jitter that must be exceeded before a minor alarm is generated.

Default: 25

Statistic_7_MajorSetValue (0x45600be)

Specifies the percentage of positive Jitter that must be exceeded before a major alarm is generated.

Default: None

Statistic_7_CriticalSetValue (0x45600bf)

Specifies the percentage of positive Jitter that must be exceeded before a critical alarm is generated.

Default: None

Statistic_7_Set_Value (0x45600bc)

Specifies the percentage of positive Jitter that must be exceeded before an event is generated.

Default: None

Statistic_7_Clear_Value (0x45600c0)

Specifies the percentage of positive Jitter that must not be exceeded before an event is generated.

Default: 25

Statistic_7_Set_Delay (0x45600c1)

Specifies the number of consecutive cycles the test must run in violation of the threshold before an event or alarm is generated.

Default: 1

Statistic_7_Clear_Delay (0x45600c2)

Specifies the number of consecutive cycles the test must run without exceeding the threshold before an event or alarm is cleared.

Default: 25

Jitter Negative Source to Destination Threshold Parameters

Statistic_8_Thresh_State (0x45600cc)

Specifies whether the Jitter negative source to destination threshold is enabled or disabled for a test.

1. 0 = Disabled
2. 1 = Enabled

Default: 0

Statistic_8_MinorSetValue (0x45600c5)

Specifies the percentage of negative Jitter that must be exceeded before a minor alarm is generated.

Default: 25

Statistic_8_MajorSetValue (0x45600c6)

Specifies the percentage of negative Jitter that must be exceeded before a major alarm is generated.

Default: None

Statistic_8_CriticalSetValue (0x45600c7)

Specifies the percentage of negative Jitter that must be exceeded before a critical alarm is generated.

Default: None

Statistic_8_Set_Value (0x45600c5)

Specifies the percentage of negative Jitter that must be exceeded before an event is generated.

Default: None

Statistic_8_Clear_Value (0x45600c9)

Specifies the percentage of negative Jitter that must not be exceeded before an event is generated.

Default: 25

Statistic_8_Set_Delay (0x45600ca)

Specifies the number of consecutive cycles the test must run in violation of the threshold before an event or alarm is generated.

Default: 1

Statistic_8_Clear_Delay (0x45600cb)

Specifies the number of consecutive cycles the test must run without exceeding the threshold before an event or alarm is cleared.

Default: 1

Jitter Negative Destination to Source Threshold Parameters

Statistic_9_Thresh_State (0x45600d5)

Specifies whether the Jitter negative destination to source threshold is enabled or disabled for a test.

1. 0 = Disabled
2. 1 = Enabled

Default: None

Statistic_9_MinorSetValue (0x45600cf)

Specifies the percentage of negative Jitter that must be exceeded before a minor alarm is generated.

Default: 25

Statistic_9_MajorSetValue (0x45600d0)

Specifies the percentage of negative Jitter that must be exceeded before a major alarm is generated.

Default: None

Statistic_9_CriticalSetValue (0x45600d1)

Specifies the percentage of negative Jitter that must be exceeded before a critical alarm is generated.

Default: None

Statistic_9_Set_Value (0x45600ce)

Specifies the percentage of negative Jitter that must be exceeded before an event is generated.

Default: None

Statistic_9_Clear_Value (0x45600d2)

Specifies the percentage of negative Jitter that must not be exceeded before an event is generated.

Default: 25

Statistic_9_Set_Delay (0x45600d3)

Specifies the number of consecutive cycles the test must run in violation of the threshold before an event or alarm is generated.

Default: 1

Statistic_9_Clear_Delay (0x45600d4)

Specifies the number of consecutive cycles the test must run without exceeding the threshold before an event or alarm is cleared.

Default: 1

Packet out of Sequence SD Threshold Parameters

POOS_SD_ThreshState (0x4560119)

Specifies whether the Packets out of Sequence SD threshold is enabled or disabled for a test.

1. 0 = Disabled
2. 1 = Enabled

Default: 0

POOS_SD_MinorSetValue (0x4560113)

Specifies the percentage of Packets out of Sequence SD that must be exceeded before a minor alarm is generated.

Default: 25

POOS_SD_MajorSetValue (0x4560114)

Specifies the percentage of Packets out of Sequence SD that must be exceeded before a major alarm is generated.

Default: None

POOS_SD_CriticalSetValue (0x4560115)

Specifies the percentage of Packets out of Sequence SD that must be exceeded before a critical alarm is generated.

Default: None

POOS_SD_SetValue (0x4560112)

Specifies the percentage of Packets out of Sequence SD that must be exceeded before an event is generated.

Default: None

POOS_SD_ClearValue (0x4560116)

Specifies the percentage of Packets out of Sequence SD that must not be exceeded before an event or alarm is cleared.

Default: 25

POOS_SD_SetDelay (0x4560117)

Specifies the number of consecutive cycles the test must run in violation of the threshold before an event or alarm is generated.

Default: 1

POOS_SD_ClearDelay (0x4560118)

Specifies the number of consecutive cycles the test must run without exceeding the threshold before an event or alarm is cleared.

Default: 1

Packet out of Sequence DS Threshold Parameters

POOS_DS_ThreshState (0x4560124)

Specifies whether the Packets out of Sequence DS threshold is enabled or disabled for a test.

1. 0 = Disabled
2. 1 = Enabled

Default: 0

POOS_DS_MinorSetValue (0x456011e)

Specifies the percentage of Packets out of Sequence DS that must be exceeded before a minor alarm is generated.

Default: 25

POOS_DS_MajorSetValue (0x456011f)

Specifies the percentage of Packets out of Sequence DS that must be exceeded before a major alarm is generated.

Default: None

POOS_DS_CriticalSetValue (0x4560120)

Specifies the percentage of Packets out of Sequence DS that must be exceeded before a critical alarm is generated.

Default: None

POOS_DS_SetValue (0x456011d)

Specifies the percentage of Packets out of Sequence DS that must be exceeded before an event is generated.

Default: None

POOS_DS_ClearValue (0x4560121)

Specifies the percentage of Packets out of Sequence DS that must not be exceeded before an event or alarm is cleared.

Default: 25

POOS_DS_SetDelay (0x4560122)

Specifies the number of consecutive cycles the test must run in violation of the threshold before an event or alarm is generated.

Default: 1

POOS_DS_ClearDelay (0x4560123)

Specifies the number of consecutive cycles the test must run without exceeding the threshold before an event or alarm is cleared.

Default: 1

Packet out of Sequence BOTH Threshold Parameters

POOS_BOTH_ThreshState (0x456012f)

Specifies whether the Packets out of Sequence BOTH threshold is enabled or disabled for a test.

1. 0 = Disabled
2. 1 = Enabled

Default: 0

POOS_BOTH_MinorSetValue (0x4560129)

Specifies the percentage of Packets out of Sequence BOTH that must be exceeded before a minor alarm is generated.

Default: 25

POOS_BOTH_MajorSetValue (0x456012a)

Specifies the percentage of Packets out of Sequence BOTH that must be exceeded before a major alarm is generated.

Default: None

POOS_BOTH_CriticalSetValue (0x456012b)

Specifies the percentage of Packets out of Sequence BOTH that must be exceeded before a critical alarm is generated.

Default: None

POOS_BOTH_SetValue (0x4560128)

Specifies the percentage of Packets out of Sequence BOTH that must be exceeded before an event is generated.

Default: None

POOS_BOTH_ClearValue (0x456012c)

Specifies the percentage of Packets out of Sequence BOTH that must not be exceeded before an event or alarm is cleared.

Default: 25

POOS_BOTH_SetDelay (0x456012d)

Specifies the number of consecutive cycles the test must run in violation of the threshold before an event or alarm is generated.

Default: 1

POOS_BOTH_ClearDelay (0x456012e)

Specifies the number of consecutive cycles the test must run without exceeding the threshold before an event or alarm is cleared.

Default: 1

Packet Skipped Threshold Parameters

PSKIPPED_ThreshState (0x456013a)

Specifies whether the Packets Skipped threshold is enabled or disabled for a test.

1. 0 = Disabled
2. 1 = Enabled

Default: 0

PSKIPPED_MinorSetValue (0x4560134)

Specifies the percentage of Packets Skipped that must be exceeded before a minor alarm is generated.

Default: 25

PSKIPPED_MajorSetValue (0x4560135)

Specifies the percentage of Packets Skipped that must be exceeded before a major alarm is generated.

Default: None

PSKIPPED_CriticalSetValue (0x4560136)

Specifies the percentage of Packets Skipped that must be exceeded before a critical alarm is generated.

Default: None

PSKIPPED_SetValue (0x4560133)

Specifies the percentage of Packets Skipped that must be exceeded before an event is generated.

Default: None

PSKIPPED_ClearValue (0x4560137)

Specifies the percentage of Packets Skipped that must not be exceeded before an event or alarm is cleared.

Default: 25

PSKIPPED_SetDelay (0x4560138)

Specifies the number of consecutive cycles the test must run in violation of the threshold before an event or alarm is generated.

Default: 1

PSKIPPED_ClearDelay (0x4560139)

Specifies the number of consecutive cycles the test must run without exceeding the threshold before an event or alarm is cleared.

Default: 1

Test Action Codes

Use action codes with the CLI update command to discover tests, save changes to tests, run tests, and control timeout debugging information.

- **Discover tests (0x4560007)**
  Specifies to run SPM Test Discovery to discover and model performance tests that are configured on test hosts and not with Service Performance Manager. Use this action code with the RTM_TestHost model.
- **Update test (0x4560008)**
  Specifies to update test after making attribute changes.
- **Run test (0x4560009)**
  Specifies to run the test.
- **Update and Run test (0x456000a)**
  Specifies to run a combination of the Update and Run actions.
- **Include Timeout Debugging Information (0x456000e)**
  Specifies to include diagnostic information with SPM Timeout Event description.
- **Turn Off Timeout Debugging Information (0x456000f)**
  Specifies not to include diagnostic information with SPM Timeout Event description.

Test Status and Test Results Parameters

SPM LatestStatus (0x4560004) Attribute Values

You can use the latest status attribute with the CLI show command to get status information for the most recently run test.

- **1 = Ok**
  Test ran successfully.
- **2 = Threshold**
  Test result has exceeded its Set Value.
- **3 = Timeout**
  Test timed out.
- **4 = Failed**
  Test failed to create a response time test table entry due to configuration issue.
- **5 = Initial**
  Test has never run.
- **6 = Bad_comm**
  Test failed to create a response time test table entry because of an invalid community string.
- **7 = Running**
  Test is running.
- **8 = Stopped**
  Test has stopped running.
- **9 = Threshold_minor**
  Test result has exceeded its Minor Set Value.
- **10 = Threshold_major**

Test result has exceeded its Major Value.
- **11 = Threshold_critical**
  Test result has exceeded its Critical Value.

Test Results

Use the CLI show command to get results for a particular test.

- **Result_Timestamp (0x456005e)**
  The time test last completed.
- **Latest_Result (0x4560015)**
  Average Response Time (scalar).
- **TRACEROUTE_Result (0x4560075)**
  TraceRoute Result. List of IP Address Latency Result pairs.
- **PL_Result (0x456007d)**
  % Packet Loss.
- **DNS_Latest_Result (0x4560037)**
  Average HTTP DNS Lookup Time (scalar).
- **TCP_Latest_Result (0x456003f)**
  Average HTTP TCP Connection Time (scalar).
- **DL_Latest_Result (0x4560047)**
  Average HTTP Page Download Time (scalar).
- **JPLSD_Latest_Result (0x456007e)**
  % Jitter Packets Loss between Source and Destination.
- **JPLDS_Latest_Result (0x456007f)**
  % Jitter Packets Loss between Destination and Source.
- **JBUS_Latest_Result (0x4560080)**
  % Jitter Packets Busy.
- **JMIA_Latest_Result (0x4560081)**
  % Jitter Packets Missing in Action.
- **JLATE_Latest_Result (0x4560082)**
  % Jitter Packets Arriving Late.
- **PosJitterSD_LatestResult (0x45600b2)**
  Average Positive Jitter between Source and Destination.
- **PosJitterDS_LatestResult (0x45600bb)**
  Average Positive Jitter between Destination and Source.
- **NegJitterSD_LatestResult (0x45600c4)**
  Average Negative Jitter between Source and Destination.
- **NegJitterDS_LatestResult (0x45600cd)**
  Average Negative Jitter between Destination and Source.
- **MOS_Latest_Result (0x45600e5)**
  Mean Opinion Score Value (0 - 500). It provides a numerical measure of the quality of human speech at the destination end of the circuit.

Troubleshooting Service Performance Manager

This chapter identifies error messages that may be generated during Service Performance Manager operations and describes corrective action where feasible. Other maintenance and optional configuration issues are also addressed.

Firmware Issues

Certain router firmware revisions can exhibit instability. CA follows published interfaces to the SNMP agents, and rely on the device vendors to fully support these interfaces. Before deploying Service Performance Manager, it is advisable that users review device and firmware documentation from the vendor and apply any updates as appropriate.

Cisco IOS

When managing Jitter tests on Cisco IP SLA hosts, certain Cisco IOS versions do not allow codec type changes once the test has been run.

Cisco IOS 12.0(9)

Cisco IOS 12.0(9) has an issue that causes the router to reload upon the first SNMP SET performed to validate that the MIB is writable.

Cisco IOS 12.0(9) has an issue that causes the router to reload upon an SNMP Get of the supported test types table, which occurs during model activation.

Cisco IOS 12.0(7)T2

Cisco IOS 12.0(7)T2 has an issue which causes the test that is described in Traceroute Tests to fail. The problem is that the device does not report hop data correctly in the result tables of the CISCO-RTTMON-MIB, which causes Service Performance Manager to put erroneous data in the result event for average response time. To address this issue, upgrade your firmware to Cisco IOS 12.1(17).

Cisco IOS below 12.2

Cisco IOS below 12.2 has an issue which causes HTTP version 1.1 tests to fail with a “Request Timed Out” error message. Upgrading to Cisco IOS 12.2 or later versions will fix this issue.

Workaround: Changing the HTTP version from 1.1 to 1.0 may correct the timeout error message. See Configure Advanced Parameters for more information.

Cisco IOS 12.2(2)T

Cisco IOS 12.2(2)T has an issue that causes the router to intermittently report incorrect operation error codes such as DHCP response time tests that have timed out are reported as OK. If you encounter this issue during a run of a DHCP test, Service Performance Manager can report a latency value for the DHCP test, greater than the timeout value set for the test. To address this issue, Cisco recommends upgrading your firmware to Cisco IOS 12.2(15)T2.

Cisco IOS 12.2(11)T

Cisco devices running IOS firmware 12.2(11)T and higher correctly function as test hosts in Service Performance Manager. Previously, configured tests would not operate correctly due to changes in the RTTMon MIB.

Cisco IOS 12.3(4)

When running Service Performance Manager tests, a router running Cisco IOS version 12.3(10a) that is configured for SAA/RTR may crash because of a memory leak in the SAA/RTR process. This issue has been resolved in IOS version 12.3 (11) TO4.

Cisco IOS 12.3(5) and below

In Cisco IOS 12.3(5) and lower versions, changing the packet size on a Jitter Tests can cause the IOS to crash and reboot the router. This issue is addressed in Cisco IOS 12.3(5.013).

Cisco IOS 12.2(18)SXF3 and 12.2(18)SXF4

Cisco Routers running IOS 12.2(18)SXF3 with version 12.2(18)SXF4 can crash, when Service Performance Manager tests are run. Because of Cisco bug CSCin62031, the router can crash. Routers running these IOS versions cannot be modeled as test hosts capable of running Cisco IPSLA. To prevent these routers from being modeled as test hosts capable of running Cisco IPSLA, add the following commands to the router configuration before modeling:

```
snmp-server view NoRTTMON internet included
snmp-server view NoRTTMON ciscoRttMonMIB excluded
snmp-server community TEST view NoRTTMON RO
```

If these configuration commands are added after modeling, any attempt to run Service Performance Manager tests fails and do not let the router to crash.

Juniper (all JUNOS devices)

On a Juniper host device, running a response time test with a name longer than 32 characters returns an error. If you see such an error, recreate the test with a shorter name. When test templates are used, be aware that the model name or IP address is appended onto the template name. Be sure to leave enough characters when using templates so that the final, full test name reaches or falls below the 32 character limit.

Riverstone RS-8000 FW 9.0.0.4

Response time tests that are run from Riverstone RS-8000 (firmware 9.0.0.4) test host devices can return Bad Configuration errors. If you see these errors, verify that no two tests (configured for the Riverstone test host) have names with the same character length. If necessary, rename any such tests. For more information, see Configure General Parameters.

Timeout Errors

In most cases, lack of access is the cause of timeout errors during Service Performance Manager tests. For example, when Network Address Translation (NAT) is enabled, it can deny access to networks that are not in the list of networks to translate. Thus response time tests for HTTP or ICMP echo from an unlisted range of IPs would result in timeouts.

The solution in this case is to add the test network to the NAT list and rerun the tests. Other tests, such as DNS, DHCP, and UDP require the test host device to be properly configured for the service to be tested.

Timeout errors can also be caused by setting the latency timeout parameter to an invalid value. Valid values for test timeouts can vary from one test type to another, and from one device to another. For example, the CISCO-RTTMON-MIB provides the following guideline in describing its timeout value:

```
To prevent unwanted closure of connections, be sure to set this value to a realistic connection timeout.
```

You can verify the following common solutions for timeout errors:

- Verify that you have SNMP read/write access on the test host device.
- Verify that you have access from the test host device to the service being tested (such as ICMP, HTTP).
- Verify that you have set reasonable values for a test timeout.

NOTE
HTTP tests using certain Cisco Routers that run HTTP version 1.1 can fail with a Request Timed Out error. For more information, see Firmware Issues.

ICMP Ping Tests and Extreme Summit Test Host Devices

Ping tests that are run from Extreme Summit test host devices can result in timeouts. When you execute a ping from certain Extreme Summit devices, the RFC2925 branch of their MIB agents incorrectly reports a response time of 0. Service Performance Manager interprets this issue as a timeout, when in fact the device has replied to the ping. The current workaround is to use a different device as a test host.

Traceroute Tests and iAgent Test Hosts

Service Performance Manager always reports a timeout when traceroute tests are run on an iAgent test host. This issue is resolved by iAgent version 16.2.

Add Debugging Information to a Timeout Event

When attempting to determine the cause of a timeout event, you can collect diagnostic information for use by CA Support. Use the Command Line Interface to send an action to a test model so that it includes debugging information with the timeout event.

Using a CLI command, you can add debugging information to a timeout event and also disable debugging.

Follow these steps:

1. Open the Command Line Interface and enter the following command:
   ```
   update action=0x456000e mh=<RTM_Test Model_Handle>
   ```
   where *RTM_Test Model_Handle* is the Model_Handle attribute for the test model.
2. Add the following information to the SPM Timeout Event description:
   ```
   Additional Info: id xxxxx error: yy
   ```
   The information is added to the SPM Timeout Event description.
3. To disable debugging, use the following command:
   ```
   update action=0x456000f mh=<RTM_Test Model_Handle>
   ```
   where *RTM_Test Model_Handle* is the Model_Handle attribute for the test model.
   Debugging is disabled.

NOTE
For more information, see the Command Line Interface section.

Scheduling Tests in Geographically Distributed Environments

In a geographically distributed environment, the following network components can potentially be in different time zones:

- SpectroSERVERs
- OneClick consoles
- Web servers
- Test host devices

Test scheduling in Service Performance Manager is based on the time zone of the SpectroSERVER where the test host is modeled. Test results are based on the time zone of the OneClick console. To address the time discrepancy, select the 24/7 option when scheduling.

SpectroSERVER Crashed while Deleting the RTM_Hosts

Symptom:

When RTM_Hosts are deleted from Service Performance Manager, SpectroSERVER crashes.

Solution:

RTM Test Hosts are automatically created when a device is modeled and it supports the RTTMON application You must not delete these models. These models lets you know the device that supports the RTTMON application.

Delayed SpectroSERVER Activation Resulting from External Reads

For agents that support performance test types through MIB objects, DX NetOps Spectrum must perform external reads of these MIB objects at SpectroSERVER activation time to obtain the list of supported test types. This action can delay the SpectroSERVER activation time. You can configure the spm_wait_activate parameter in the .vnmrc file to delay the external reads until the SpectroSERVER is activated.

Set the default value of spm_wait_activate from No to Yes to prevent Service Performance Manager from performing external reads until the SpectroSERVER is activated.

> **NOTE**
> The agent test host remains inactive until the SpectroSERVER is activated.

Event Codes

This section describes the Event Code descriptions.

About SPM Timeout Event

Service Performance Manager determines when it is appropriate to generate an SPM Timeout Event and its associated alarm:

- SPM Timeout Event and its related alarm are suppressed, if the destination IP of the test represents a DX NetOps Spectrum device model that does not have ICMP contact. The alarm appears as a symptom of the DEVICE HAS STOPPED RESPONDING TO POLLS alarm of the DX NetOps Spectrum device model.
- SPM Timeout Event and its related alarm are not generated, if the test host does not have SNMP contact at the time that the RTM results are read for a given Service Performance Manager test.

Event Code Descriptions

The following table lists Service Performance Manager events by event code, event name, the model types that event can be asserted on, the alarm that is generated or cleared (if any), and the alarm severity.

Event Code	Event Name	Model Type Asserted On	Alarm Generated or Cleared	Alarm Severity
0x04560000	SPM Result Event	RTM_Test	None	N/A
0x04560001	SPM Timeout Event. For more information, see About SPM Timeout Event.	RTM_Test	0x04560001	Yellow
0x04560002	SPM Test Host Configuration Failed Event	RTM_Test	0x04560002	Yellow
0x04560003	SPM Threshold Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	None	N/A
0x04560004	SPM Threshold Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x04560003	Yellow
0x04560005	SPM Threshold Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x04560003	Orange
0x04560006	SPM Threshold Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x04560003	Red
0x04560007	SPM Threshold Cleared Event	RTM_Test, RTM_TestHost, Source, or Destination	Clears 0x04560003	N/A
0x04560008	SPM Test Creation Event	RTM_Test	None	N/A
0x0456000a	SPM Test Modification Event	RTM_Test	None	N/A
0x0456000b	SPM Packet Loss Threshold Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	None	N/A
0x0456000c	SPM Packet Loss Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x0456000b	Yellow
0x0456000d	SPM Packet Loss Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x0456000b	Orange
0x0456000e	SPM Packet Loss Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x0456000b	Red
0x0456000f	SPM Packet Loss Cleared Event	RTM_Test, RTM_TestHost, Source, or Destination	Clears 0x0456000b	N/A
0x04560010	SPM Result Event (Jitter)	RTM_Test	None	N/A

0x04560011	SPM Jitter Packet Loss Source to Destination Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	None	N/A
0x04560012	SPM Jitter Packet Loss Source to Destination Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x04560011	Yellow
0x04560013	SPM Jitter Packet Loss Source to Destination Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x04560011	Orange
0x04560014	SPM Jitter Packet Loss Source to Destination Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x04560011	Red
0x04560015	SPM Jitter Packet Loss Source to Destination Cleared Event	RTM_Test, RTM_TestHost, Source, or Destination	Clears 0x04560011	N/A
0x04560016	SPM Jitter Packet Loss Destination to Source Threshold Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	None	N/A
0x04560017	SPM Jitter Packet Loss Destination to Source Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x04560016	Yellow
0x04560018	SPM Jitter Packet Loss Destination to Source Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x04560016	Orange
0x04560019	SPM Jitter Packet Loss Destination to Source Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x04560016	Red
0x0456001a	SPM Jitter Packet Loss Destination to Source Threshold Cleared Event	RTM_Test, RTM_TestHost, Source, or Destination	Clears 0x04560016	N/A
0x0456001b	SPM Jitter Packet MIA Threshold Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	None	N/A
0x0456001c	SPM Jitter Packet MIA Threshold Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x0456001b	Yellow
0x0456001d	SPM Jitter Packet MIA Threshold Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x0456001b	Orange
0x0456001e	SPM Jitter Packet MIA Threshold Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x0456001b	Red
0x0456001f	SPM Jitter Packet MIA Threshold Cleared Event	RTM_Test, RTM_TestHost, Source, or Destination	Clears 0x0456001b	N/A
0x04560020	SPM Jitter Packet Late Arrival Threshold Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	None	N/A
0x04560021	SPM Jitter Packet Late Arrival Threshold Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x04560020	Yellow

0x04560022	SPM Jitter Packet Late Arrival Threshold Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x04560020	Orange
0x04560023	SPM Jitter Packet Late Arrival Threshold Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x04560020	Red
0x04560024	SPM Jitter Packet Late Arrival Threshold Cleared Event	RTM_Test, RTM_TestHost, Source, or Destination	Clears 0x04560020	N/A
0x04560025	SPM Jitter Busies Threshold Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	None	N/A
0x04560026	SPM Jitter Busies Threshold Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x04560025	Yellow
0x04560027	SPM Jitter Busies Threshold Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x04560025	Orange
0x04560028	SPM Jitter Busies Threshold Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x04560025	Red
0x04560029	SPM Jitter Busies Threshold Cleared Event	RTM_Test, RTM_TestHost, Source, or Destination	Clears 0x04560025	N/A
0x0456002a	SPM Test Pause (Test Host Down) Event	RTM_Test	None	N/A
0x0456002b	SPM Test Restart (Test Host Recontacted) Event	RTM_Test	None	N/A
0x0456002c	SPM Test Entering Management Mode Event	RTM_Test	None	N/A
0x0456002d	SPM Test Exiting Management Mode Event	RTM_Test	None	N/A
0x0456002e	SPM Result Event (HTTP)	RTM_Test	None	N/A
0x0456002f	SPM HTTP DNS Resolution Threshold Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	None	N/A
0x04560030	SPM HTTP DNS Resolution Threshold Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x0456002f	Yellow
0x04560031	SPM HTTP DNS Resolution Threshold Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x0456002f	Orange
0x04560032	SPM HTTP DNS Resolution Threshold Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x0456002f	Red
0x04560033	SPM HTTP DNS Resolution Threshold Cleared Event	RTM_Test, RTM_TestHost, Source, or Destination	Clears 0x0456002f	N/A

0x04560034	SPM HTTP TCP Connect Threshold Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	None	N/A
0x04560035	SPM HTTP TCP Connect Threshold Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x04560034	Yellow
0x04560036	SPM HTTP TCP Connect Threshold Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x04560034	Orange
0x04560037	SPM HTTP TCP Connect Threshold Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x04560034	Red
0x04560038	SPM HTTP TCP Connect Threshold Cleared Event	RTM_Test, RTM_TestHost, Source, or Destination	Clears 0x04560034	N/A
0x04560039	SPM HTTP Page Download Threshold Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	None	N/A
0x0456003a	SPM HTTP Page Download Threshold Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x0456003c	Yellow
0x0456003b	SPM HTTP Page Download Threshold Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x0456003c	Orange
0x0456003c	SPM HTTP Page Download Threshold Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x0456003c	Red
0x0456003d	SPM HTTP Page Download Threshold Cleared Event	RTM_Test, RTM_TestHost, Source, or Destination	Clears 0x0456003c	N/A
0x0456003e	SPM Result Event (Traceroute)	RTM_Test	None	N/A
0x0456003f	SPM Traceroute Threshold Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	None	N/A
0x04560040	SPM Traceroute Threshold Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x0456003f	Yellow
0x04560041	SPM Traceroute Threshold Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x0456003f	Orange
0x04560042	SPM Traceroute Threshold Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x0456003f	Red
0x04560043	SPM Traceroute Threshold Cleared Event	RTM_Test, RTM_TestHost, Source, or Destination	Clears 0x0456003f	N/A
0x04560044	SPM Test SNMP Set Failure Event	RTM_TestHost	0x04560044	Yellow
0x04560045	SPM Test SNMP Set Failure Cleared Event	RTM_TestHost	Clears 0x04560044	N/A

0x04560046	SPM Test Timeout Cleared Event	RTM_Test	Clears 0x04560001	N/A
0x04560047	SPM Test Host Configuration Failed Cleared Event	RTM_Test	Clears 0x04560002	N/A
0x04560048	SPM Too Many Probes On Test Host Event	RTM_TestHost	0x04560048	Yellow
0x04560049	SPM Too Many Probes on Test Host Cleared Event	RTM_TestHost	Clears 0x04560048	N/A
0x0456004a	SPM HTTP Result Event	RTM_Test	None	N/A
0x04560054	Bad Ping Result	RTM_Test	None	N/A
0x04560055	Bad Jitter Result	RTM_Test	None	N/A
0x04560056	Bad HTTP Result	RTM_Test	None	N/A
0x04560057	Bad Traceroute Result	RTM_Test	None	N/A
0x04560058	Bad HTTP Result	RTM_Test	None	N/A
0x04560059	SPM Test No Longer On Device Event	RTM_Test	0x04560059	Yellow
0x0456005a	SPM Test No Longer Running On Device Event	RTM_Test	None	N/A
0x0456005b	SPM Duplicate Result Event	RTM_Test	None	N/A
0x0456005c	SPM Test Discovery Completion Event	RTM_TestHost	None	N/A
0x0456005d	SPM Test Type Mismatch Event	RTM_Test	0x0456005d	Yellow
0x0456005e	SPM Stale Test Cleared Event	RTM_Test	Clears 0x04560059 0x0456005d	N/A
0x0456005f	SPM Positive Jitter Source to Destination Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	None	N/A
0x04560060	SPM Positive Jitter Source to Destination Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x0456005f	Yellow
0x04560061	SPM Positive Jitter Source to Destination Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x0456005f	Orange
0x04560062	SPM Positive Jitter Source to Destination Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x0456005f	Red
0x04560063	SPM Positive Jitter Source to Destination Cleared Event	RTM_Test, RTM_TestHost, Source, or Destination	Clears 0x0456005f	N/A
0x04560064	SPM Positive Jitter Destination to Source Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	None	N/A

0x04560065	SPM Positive Jitter Destination to Source Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x04560064	Yellow
0x04560066	SPM Positive Jitter Destination to Source Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x04560064	Orange
0x04560067	SPM Positive Jitter Destination to Source Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x04560064	Red
0x04560068	SPM Positive Jitter Destination to Source Cleared Event	RTM_Test, RTM_TestHost, Source, or Destination	Clears 0x04560064	N/A
0x04560069	SPM Negative Jitter Source to Destination Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	None	N/A
0x0456006a	SPM Negative Jitter Source to Destination Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x04560069	Yellow
0x0456006b	SPM Negative Jitter Source to Destination Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x04560069	Orange
0x0456006c	SPM Negative Jitter Source to Destination Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x04560069	Red
0x0456006d	SPM Negative Jitter Source to Destination Cleared Event	RTM_Test, RTM_TestHost, Source, or Destination	Clears 0x04560069	N/A
0x0456006e	SPM Negative Jitter Destination to Source Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	None	N/A
0x0456006f	SPM Negative Jitter Destination to Source Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x0456006e	Yellow
0x04560070	SPM Negative Jitter Destination to Source Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x0456006e	Orange
0x04560071	SPM Negative Jitter Destination to Source Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x0456006e	Red
0x04560072	SPM Negative Jitter Destination to Source Cleared Event	RTM_Test, RTM_TestHost, Source, or Destination	Clears 0x0456006e	N/A
0x04560073	SPM Too Many Probes Event	RTM_Test	None	N/A
0x04560074	SPM Bad Community String Event	RTM_Test	None	N/A
0x04560075	SPM Invalid Destination Address Event	RTM_Test	0x04560075	Yellow
0x04560076	SPM Invalid Destination Address Cleared Event	RTM_Test	Clears 0x04560075	N/A

0x04560077	SPM Invalid Test Host Event	RTM_Test	0x04560077	Yellow
0x04560078	SPM Invalid Test Type Event	RTM_Test	0x04560078	Yellow
0x04560079	SPM RTM_TestHost No Device Model Event	RTM_TestHost	0x04560079	Yellow
0x0456007a	SPM RTM_TestHost No Device Model Cleared Event	RTM_TestHost	Clears 0x04560079	N/A
0x0456007b	SPM Result Failure Event	RTM_Test	None	N/A
0x0456007c	SPM Mean Opinion Score (MOS) event	RTM_Test, RTM_TestHost, Source, or Destination	None	N/A
0x0456007d	SPM Mean Opinion Score (MOS) event	RTM_Test, RTM_TestHost, Source, or Destination	0x456007c	Yellow
0x0456007e	SPM Mean Opinion Score (MOS) event	RTM_Test, RTM_TestHost, Source, or Destination	0x456007c	Orange
0x0456007f	SPM Mean Opinion Score (MOS) event	RTM_Test, RTM_TestHost, Source, or Destination	0x456007c	Red
0x04560080	SPM Mean Opinion Score (MOS) event	RTM_Test, RTM_TestHost, Source, or Destination	Clears 0x456007c	N/A
0x04560081	Juniper Jitter Result Event	RTM_Test, RTM_TestHost, Source, or Destination	None	N/A
0x04560082	Bad Juniper Jitter Result	RTM_Test, RTM_TestHost, Source, or Destination	None	N/A
0x04560083	Juniper RTT Jitter Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	None	N/A
0x04560084	Juniper RTT Jitter Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x04560083	Yellow
0x04560085	Juniper RTT Jitter Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x04560083	Orange
0x04560086	Juniper RTT Jitter Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x04560083	Red
0x04560087	Juniper RTT Jitter Cleared Event	RTM_Test, RTM_TestHost, Source, or Destination	0x04560083	N/A
0x04560088	Juniper Egress Jitter Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	None	N/A

0x04560089	Juniper Egress Jitter Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x04560088	Yellow
0x0456008a	Juniper Egress Jitter Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x04560088	Orange
0x0456008b	Juniper Egress Jitter Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x04560088	Red
0x0456008c	Juniper Egress Jitter Cleared Event	RTM_Test, RTM_TestHost, Source, or Destination	Clears 0x04560088	N/A
0x0456008d	Juniper Egress Jitter Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	None	N/A
0x0456008e	Juniper Egress Jitter Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x0456008d	Yellow
0x0456008f	Juniper Egress Jitter Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x0456008d	Orange
0x04560090	Juniper Egress Jitter Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x0456008d	Red
0x04560091	Juniper Egress Jitter Cleared Event	RTM_Test, RTM_TestHost, Source, or Destination	Clears 0x0456008d	N/A
0x04560092	ICMP_Jitter Result Event	RTM_Test	None	N/A
0x04560093	SPM ICMP_Jitter Bad Result Event	RTM_Test	None	N/A
0x04560094	SPM ICMP_Jitter Packet out of sequence source to destination Threshold Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	None	N/A
0x04560095	SPM ICMP_Jitter Packet out of sequence source to destination Threshold Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x04560094	Yellow
0x04560096	SPM ICMP_Jitter Packet out of sequence source to destination Threshold Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x04560094	Orange
0x04560097	SPM ICMP_Jitter Packet out of sequence source to destination Threshold Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x04560094	Red
0x04560098	SPM ICMP_Jitter Packet out of sequence source to destination Threshold Cleared Event	RTM_Test, RTM_TestHost, Source, or Destination	Clears 0x04560094	N/A

0x04560099	SPM ICMP_Jitter Packet out of sequence destination to source Threshold Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	None	N/A
0x0456009a	SPM ICMP_Jitter Packet out of sequence destination to source Threshold Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x04560099	Yellow
0x0456009b	SPM ICMP_Jitter Packet out of sequence destination to source Threshold Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x04560099	Orange
0x0456009c	SPM ICMP_Jitter Packet out of sequence destination to source Threshold Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x04560099	Red
0x0456009d	SPM ICMP_Jitter Packet out of sequence destination to source Threshold Cleared Event	RTM_Test, RTM_TestHost, Source, or Destination	Clears 0x04560099	N/A
0x0456009e	SPM Jitter Packet Out of Sequence BOTH Threshold Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	None	N/A
0x0456009f	SPM Jitter Packet Out of Sequence BOTH Threshold Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x0456009e	Yellow
0x04560100	SPM Jitter Packet Out of Sequence BOTH Threshold Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x0456009e	Orange
0x04560101	SPM Jitter Packet Out of Sequence BOTH Threshold Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x0456009e	Red
0x04560102	SPM Jitter Packet Out Of Sequence BOTH Threshold Cleared Event	RTM_Test, RTM_TestHost, Source, or Destination	Clears 0x0456009e	N/A
0x04560103	SPM Jitter Packet Skipped Threshold Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	None	N/A
0x04560104	SPM Jitter Packet Skipped Threshold Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x04560103	Yellow
0x04560105	SPM Jitter Packet Skipped Threshold Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x04560103	Orange

0x04560106	SPM Jitter Packet Skipped Threshold Exceeded Event	RTM_Test, RTM_TestHost, Source, or Destination	0x04560103	Red
0x04560107	SPM Jitter Packet Skipped Threshold Cleared Event	RTM_Test, RTM_TestHost, Source, or Destination	Clears 0x04560103	N/A

Standards-Based Protocol Reference

This section describes the standard protocols that DX NetOps Spectrum supports. When a device is modeled in OneClick, DX NetOps Spectrum automatically creates child application models for the standards supported by that device. Information about the standards is available in various OneClick subviews. From OneClick, you can view standard MIB information through model attributes, create attribute watches, and edit attribute values.

The standards that are described in this section are organized according to their functionality.

Bridging

This section describes the bridging standards applications that DX NetOps Spectrum supports. Bridging applications include models such as Spanning Tree and PPP Bridging.

Bridges are generally more flexible and intelligent than repeaters because they interconnect separate LAN or WAN data links and they learn the addresses of nodes that can be reached over each data link. Traffic can then be relayed selectively across each bridge. The bridging function operates in the MAC (Media Access Control) sublayer and is transparent to layers above the MAC sublayer.

Bridges can interconnect networks using different transmission techniques or MAC methods. A bridge can connect a LAN data link to a WAN telecommunications operation. Multiple bridges can be used to interconnect a series of networks. A pair of bridges with a telecommunications entity located between them can interconnect two different LAN locations.

Individual LAN data links that are interconnected by a bridge are considered to be a single subnetwork. Subnetwork station addresses must be unique and must use the same station address format. An Extended LAN is actually a LAN subnetwork constructed of bridges and is different from a single physical LAN. Operating layers above the MAC sublayer view, the Extended LAN acts as if it were a single LAN data link.

A bridge can implement a frame-filtering mechanism, or filtering bridge, to receive all frames that are transmitted over each attached data link. Based on each frame's destination address, the bridge determines if each frame can be transmitted across the bridge to any other attached data links. A bridge can therefore isolate network traffic generated on a LAN data link from other LAN data links in the Extended LAN. Broadcast traffic generated on one LAN is transmitted across a bridge to other data links to which it is attached; traffic generated by any station is received by all stations on the Extended LAN.

PPP Bridging

PPP (Point-to-Point Protocol) is used for managing the bridge network control protocol on subnetwork interfaces using the family of Point-to-Point protocols.

- RFC 1474
  - **Name:** PPP Bridge
  - **Model Type Name:** PPP_Bridge_App
  - **Name:** PPP_Bridge
  - **Model Type Name:** PPP_BdgApp1474

QBridge

QBridge, or IEEE 802.1Q, defines VLAN tagging, used to allow multiple bridged networks to transparently share the same physical network link without leakage of information between networks.

- RFC 2674
  - **Name:** Q Bridge
  - **Model Type Name:** qbridge_app_05

Source Routing

The Source Routing application contains utilization statistics derived from source routing. This information may be present in bridging token ring packets, and may affect a specific token ring interface. The Source Routing data is collected from the source routing information potentially present in any token ring packet. This information can be present in a transparent bridging or a mixed bridging environment, and can only be valid in a pure source route bridging environment.

- RFC 1525
  - **Name:** Source Routing
  - **Model Type Name:** rfc1525App

Spanning Tree Bridges

A spanning tree bridge learns appropriate routes from frames and verifies that all bridges are using the same network topology. Spanning tree bridges are used to form tree structures in which any two stations on an extended LAN are connected by one active path. A spanning tree bridge is transparent to ordinary stations on the interconnected LANs. To create and maintain the spanning tree, each bridge periodically multicasts Hello packets to all other bridges on the extended LAN. These packets are used to calculate the spanning tree and to verify that all bridges are using the same topology. Redundant links are not used. If a bridge or link failure occurs, the Hello packet transmissions allow the bridges to quickly calculate a new spanning tree.

- RFC 1493
  - **Name:** Spanning Tree
  - **Model Type Name:** Span_Tree_App

Static Bridging

Static bridging deals with destination-address filtering, letting you create an entry that stays in the static routing table without aging out or being removed if a device is powered off.

The Static Group, one of the five groups within the Bridge MIB, contains objects that describe the entity's state with respect to destination address filtering. If destination address filtering is not supported, this group is not implemented. This group is applicable to any type of bridge that performs destination address filtering.

- RFC 1493
  - **Name:** Static
  - **Model Type Name:** Static_App

Transparent Bridges

Transparent bridges operate in a manner that is transparent to network hosts. A transparent bridge learns the network's topology by analyzing the source address of incoming frames from all attached networks. From this process, it builds a table, which it then uses as a basis for traffic forwarding. When a frame is received on one of the bridge's interfaces, the bridge looks up the frame's destination address in its internal table. If the table contains an association between the destination address and any of the bridge's ports, aside from the one on which the frame was received, the frame is forwarded out to the indicated port. If no association is found, the frame is sent to all ports except the inbound port. Broadcasts and multicasts are also flooded in this way.

- RFC 1493
  - **Name:** Transparent
  - **Model Type Name:** Transparnt_App

Broadband

This section describes the broadband standards applications that DX NetOps Spectrum supports.

ADSL

ADSL (Asymmetric Digital Subscriber Line) is a continuously available connection over an existing phone line.

- RFC 2662
  - **Name:** ADSLLineApp
  - **Model Type Name:** ADSLLineApp

DOCSIS

DOCSIS (Data Over Cable Service Interface Specifications) is an international standard that defines the communications and operation support interface requirements for a data over cable system. It permits the addition of high-speed data transfer to an existing Cable TV system.

- Draft DOCS-BPI-MIB
  - **Name:** DOCSISBPIApp
  - **Model Type Name:** DOCSISBPIApp
- Draft DOCS-BPI2-MIB
  - **Name:** DOCSISBPI2App
  - **Model Type Name:** DOCSISBPI2App
- RFC 2669
  - **Name:** DOCSISCblDvApp
  - **Model Type Name:** DOCSISCblDvApp
- RFC 2670
  - **Name:** DOCSISIFApp
  - **Model Type Name:** DOCSISIFApp
- Draft DOCS-QOS-MIB
  - **Name:** DOCSISQOSApp
  - **Model Type Name:** DOCSISQOSApp

Device and System Identity

This section describes the device and system identity standards applications that DX NetOps Spectrum supports.

Entity

RFC 2737, Entity MIB, provides insight into logical and physical entities managed by an SNMP agent.

- RFC 2737
  - **Name:** EntityMIBApp
  - **Model Type Name:** RFC2737App

Host Resources

DX NetOps Spectrum supports the following Host Resources standards applications:

- RFC 2790
  - **Name:** Host Resources
  - **Model Type Name:** rfc2790App
- RFC 1514
  - **Name:** Host Resources
  - **Model Type Name:** rfc1514App

System Application

The system application RFC provides objects for fault, configuration, and performance management of applications from a systems perspective.

- RFC 2287
  - **Name:** SystemLvlApp
  - **Model Type Name:** RFC2287App

IP Protocols and Services

The section describes the standards DX NetOps Spectrum supports for application models associated with IP protocols and services.

DHCP

DHCP (Dynamic Host Configuration Protocol) is a computer networking protocol used by devices (DHCP clients) to obtain configuration information for operation in an Internet Protocol network.

- RFC 2131
  - **Name:** dhcpApp
  - **Model Type Name:** dhcpApp

ICMP

ICMP (Internet Control Message Protocol) uses IP datagrams and is typically generated in response to errors in IP datagrams or for diagnostic or routing purposes.

- RFC 792
  - **Name:** ICMP
  - **Model Type Name:** ICMP_App

    NOTE
    If the firmware on your device supports both the draft and the RFC version of the 2933 MIB, only the RFC2933App is modeled.

IGMP

IGMP (Internet Group Management Protocol) provides support for managing the membership of IP multicast groups. IGMP is used by IP hosts and adjacent multicast routers to establish multicast group memberships.

- Draft RFC 2933

  - **Name:** IGMP Draft
  - **Model Type Name:** Draft2933App
- RFC 2933
  - **Name:** IGMP
  - **Model Type Name:** RFC2933App

IP

IP (Internet Protocol) is commonly used as a delivery service. IP is a connectionless service responsible for moving datagrams from one device to another. As a delivery agent, IP watches to verify that the datagrams are within shipping regulations. To deliver datagrams, IP deals with two issues: addressing and fragmentation.

- RFC 1354
  - **Name:** IP
  - **Model Type Name:** IP1_App

IPM

The IP Multicast Routing (IPM) standard is used for managing IP Multicast Routing for IPv4, independent of the specific multicast routing protocol in use.

- RFC 2932
  - **Name:** IPMRouteStd
  - **Model Type Name:** RFC2932App

IP Tunnel

IP Tunnel MIB describes managed objects used for managing tunnels of any type over IPv4 networks.

- RFC 2667
  - **Name:** IPTunnel
  - **Model Type Name:** RFC2667App

MPLS

MPLS (Multi-Protocol Label Switching) is a standards-approved technology for speeding up network traffic flow and making it easier to manage.

- MplsVpn Draft 3
  - **Name:** MplsVpnApp
  - **Model Type Name:** MplsVpnApp
- MplsVpn Draft 4
  - **Name:** MplsVpnD4App
  - **Model Type Name:** MplsVpnD4App

MSDP

MSDP (Multicast Source Discovery Protocol) is a standard that describes a mechanism to connect multiple IPv4 protocol independent multicast sparse-mode (PIM) domains together.

- IETF Draft 3
  - **Name:** Msdp_D3_App
  - **Model Type Name:** Msdp_D3_App
- IETF Draft 7

  - **Name:** Msdp_D7_App
  - **Model Type Name:** Msdp_D7_App

PIM

PIM (Protocol Independent Multicast) describes objects used for managing the PIM protocol for IPv4, which is applicable to IPv4 routers that implement PIM.

- RFC 2934
  - **Name:** PIM-MIB
  - **Model Type Name:** rfc2934App

SNMP

SNMP is the standard protocol used to monitor IP gateways and the networks to which they attach.

- RFC 1213
  - **Name:** MIB-I
  - **Model Type Name:** SNMP1_Agent
- RFC 1213
  - **Name:** MIB-II
  - **Model Type Name:** SNMP2_Agent
- RFC 4293
  - **Name:** SNMP2_v6_Agent
  - **Model Type Name:** SNMP2_v6_Agent
- RFC 2271, RFC 3411
  - **Name:** SNMP Management Frameworks
  - **Model Type Name:** RFC2271App
- RFC 3413
  - **Name:** SNMP Applications
  - **Model Type Name:** RFC3413App
- RFC 3584
  - **Name:** SNMP Coexistence
  - **Model Type Name:** RFC3584App
- RFC 3415
  - **Name:** View-based Access Control Model
  - **Model Type Name:** RFC3415App
- RFC 1213
  - **Name:** System
  - **Model Type Name:** System1_App

TCP

TCP (Transmission Control Protocol) refers to the connection-oriented transport (communications) protocol used in the Internet suite.

- RFC 2012
  - **Name:** TCP
  - **Model Type Name:** TCP1_App

UDP

UDP (User Datagram Protocol) is part of the transport layer of the OSI model and uses the services of IP to deliver data.

- RFC 2013
- – **Name:** UDP
  – **Model Type Name:** UDP1_App

LWAPP

LWAPP (Lightweight Access Point Protocol) is a Cisco proprietary protocol defining how Wireless Termination Points communicate with Access Controllers. Wireless Termination Points and Access Controllers may communicate either by means of Layer 2 protocols or by means of a routed IP network.

- RFC 5412
  – • **Name:** LWAPP
    • **Model Type Name:** Cisco_LWAPP_App

CAPWAP

The CAPWAP (Controlling And Provisioning of Wireless Access Points) protocol is a generic protocol defining AC and WTP control and data plane communication via a CAPWAP protocol transport mechanism. CAPWAP Control messages, and optionally CAPWAP Data messages, are secured using Datagram Transport Layer Security (DTLS) [RFC4347].

- RFC 5415
  – • **Name:** CAPWAP
    • **Model Type Name:** CapwapApp

LAN

This section describes the following LAN standards applications that DX NetOps Spectrum supports:

Character Stream

The Character Stream application standard provides objects for the management of character stream devices, specifically to interface ports that carry a character stream, whether physical or virtual, serial or parallel, synchronous or asynchronous.

- RFC 1316
  – **Name:** Character Stream
  – **Model Type Name:** RFC1316App

Ethernet

The Ethernet standard provides objects for managing ethernet, such as media.

- RFC 1284
  – **Name:** Ethernet IF App
  – **Model Type Name:** EthernetIfApp

FDDI

FDDI (Fiber Distributed Data Interface) provides speed and reliability to a LAN and is often used as a backbone technology as well as a means of connecting high-speed computers in a local area. DX NetOps Spectrum supports the following FDDI applications:

- RFC 1512
  - **Name:** FDDI
  - **Model Type Name:** rfc1512App
  - **Name:** SMT_1
  - **Model Type Name:** GenFDDISmt_II
  - **Name:** MAC_1.1
  - **Model Type Name:** GenFDDIMac_I
  - **Model Type Name:** GenFDDIMac_II

Link Aggregation

Link aggregation is supported through the IEEE8023-LAG-MIB, developed by IEEE for managing 802.3ad. LACP (Link Aggregation Control Protocol) dynamically detects the links that can be aggregated into a Link Aggregation Group (LAG) and aggregates when links are available.

- 802.3ad
  - **Name:** LinkAggregation
  - **Model Type Name:** 802dot3adApp

    NOTE
    You can verify LACP_IF_Port for link aggregation interface information. For more information, see Manage Link Aggregation.

    .

Power Over Ethernet

Power over Ethernet (POE) provides objects that allow management of power ethernet power sourcing equipment.

- RFC 3621
  - **Name:** Power Ethernet
  - **Model Type Name:** RFC3621App

  For devices that support POE, their corresponding device model's POE interfaces can be identified through the POE column in the Interfaces table. You can disable the POE interface identification by changing the value of the ‘EnablePOEMapping’ attribute on the selected device model.

  NOTE
  For more information about setting attribute values, see Modeling and Managing Your IT Infrastructure .

RS-232

RS-232 is one of several common data terminal equipment (DTE) or data circuit-terminating equipment (DCE) interface standards. There are three RS-232 applications:

- RFC 1317

  - **Name:** RS-232
  - **Model Type Name:** RFC1317App
  - **Name:** RS-232 sync
  - **Model Type Name:** RFC1317sync
  - **Name:** RS-232 async
  - **Model Type Name:** RFC1317async

Token Ring

The standard for the Token Ring protocol is IEEE 802.5. FDDI also uses a Token Ring protocol.

- RFC 1231
  - **Name:** Token Rng IF App
  - **Model Type Name:** TokenRingIfApp

Wireless LAN

802.11 is a family of IEEE standards for wireless local area networks (WLANs).

- 802.11
  - **Name:** WirelessLAN
  - **Model Type Name:** 802dot11App

Performance

This section describes the standards DX NetOps Spectrum supports for the following performance-related application models:

RMON

The RMON (Remote Network Monitoring) MIB is based on RFC1757 (Ethernet) and RFC1513 (Token Ring). It is divided into the following groups:

- Statistics
- History
- Alarm
- Host
- HostTopN
- Matrix
- Filter
- Packet Capture
- Event
- Token Ring

Each group defines a set of objects to be monitored. In addition, each group stores data and statistics collected by the agent on the device, which may have multiple network interfaces.

The RMONApp model accesses and presents RMON data from all network interfaces supported by a device. DX NetOps Spectrum supports the following RMON application models:

- RFC 1757, RFC 1513

  - **Name:** RMON
  - **Model Type Name:** RMONApp
- RFC 1757
  - **Name:** E Probe
  - **Model Type Name:** RMONEthProbe
- RFC 1513
  - **Name:** T R Probe
  - **Model Type Name:** RMONTRProbe

Traceroute/Lookup

The Traceroute/Lookup standard defines objects for performing remote ping, traceroute, and lookup operations at a remote host.

- RFC 2925
  - **Name:** RFC2925App
  - **Model Type Name:** RFC2925App

Routing

This section describes the following routing standards applications that DX NetOps Spectrum supports:

BGP4

BGP (Border Gateway Protocol) is the core routing protocol of the Internet.

- RFC 1269
  - **Name:** BGP4
  - **Model Type Name:** BGP4_App

IP Routing

The IP routing applications contain information used by DX NetOps Spectrum to resolve the interconnections of devices through the network at the IP level (Layer 3).

- RFC 1213, RFC 2096, RFC 1354
  - **Name:** IP Routing
  - **Model Type Name:** IPRtrApp
  - **Name:** IP Routing
  - **Model Type Name:** IP2RtrApp

IS-IS Routing

The Intermediate System to Intermediate System (IS-IS) routing protocol is used to build routing tables for IP networks.

- RFC 4444
  - **Name**: IS-IS Routing
  - **Model Type Name**: RFC4444App

OSPF

The OSPF (Open Shortest Path First) protocol is a hierarchical interior gateway protocol (IGP) for routing.

- RFC 1253
  - **Name:** OSPF
  - **Model Type Name:** OSPF2RtrApp

Repeater

The Repeater standard provides support for objects used to manage IEEE 802.3 repeaters, sometimes referred to as hubs.

- RFC 1516
  - **Name:** SNMP-Repeater
  - **Model Type Name:** rfc1516App

RIP2

RIP2 (Routing Information Protocol 2) is a distance-vector routing protocol.

- RFC 1724
  - **Name:** RIP2
  - **Model Type Name:** RFC1724App

VRRP

The VRRP (Virtual Router Redundancy Protocol) application allows several routers on a multi-access link to use the same virtual IP address. One router is elected as the master with the other routers acting as backups. This allows host systems to be configured (manually or using DHCP) with a single default gateway rather than running an active routing protocol. This protocol also supports the ability to load-share traffic when both routers are up.

- RFC 2338
  - **Name:** VRRP
  - **Model Type Name:** VRRPApp

SAN

This section describes the SAN (Storage Area Network) standards applications that DX NetOps Spectrum supports.

Fibre Channel

Fibre Channel is a gigabit-speed network technology. It is the standard connection type for SAN in enterprise storage.

- Draft FC MGMT-MIB
  - **Name:** FcMgmt
  - **Model Type Name:** DraftFcMgmtApp
- Draft RFC 2837
  - **Name:** FcFabricElement
  - **Model Type Name:** Draft2837App
- RFC 2837
  - **Name:** FcFabricElement
  - **Model Type Name:** RFC2837App

The DX NetOps Spectrum DraftFcMgmtApp application model type supports versions 3.0 and 4.0 of the Fibre Channel Management Framework Integration MIB, which was developed by the Fibre Alliance in order to provide an integrated management environment for an enterprise class storage network. If a device that supports this MIB is modeled with DX NetOps Spectrum, a DraftFcMgmtApp model is created for that device.

Security-Based Protocols

Port-Based Network Access Control

The IEEE 802.1x standard is designed to enhance the security of wireless local area networks (WLANs). It provides an authentication framework for wireless LANs, allowing a user to be authenticated by a central authority.

- 802.1x
  - **Name:** PortNetAccControl
  - **Model Type Name:** 802dot1xApp

RADIUS

DX NetOps Spectrum supports the following RADIUS (Remote Authentication Dial-In User Service) standards applications:

- RFC 2618
  - **Name:** RADIUS Authentication Client
  - **Model Type Name:** RFC2618App
- RFC 2620
  - **Name:** RADIUS Accounting Client
  - **Model Type Name:** RFC2620App

User-Based Security Model

This standard provides a description of the User-based Security Model (USM) for SNMP. It defines the elements of procedure for providing SNMP message-level security and provides objects for remotely monitoring and managing the configuration parameters for this security model.

- RFC 3414
  - **Name:** User-based Security Model
  - **Model Type Name:** RFC3414App

WAN

Wide Area Network (WAN) data link technology is used to implement point-to-point connections between devices. This section describes the foloowing WAN standards applications that DX NetOps Spectrum supports:

ATM

The ATM (Asynchronous Transfer Mode) Client application allows you to monitor virtual channel links and change channel statistics.

- RFC 1695
  - **Name:** ATM_Client
  - **Model Type Name:** ATMClientApp
  - **Name:** ATM_Switch
  - **Model Type Name:** ATMSwitchApp

DS1

DS1 is a high-speed baseband transmission link.

- RFC 1406

  - **Name:** DS1
  - **Model Type Name:** DS1App1406
  - **Model Type Name:** DS1_1406App
- RFC 1232
  - **Name:** DS1
  - **Model Type Name:** RFC1232App

Frame Relay

Frame relay is a packet-based interface standard that has been optimized for the transport of protocol-oriented data. The frame relay interface specification provides a signaling and data transfer mechanism. Frame relay's ability to statistically multiplex means that paths or virtual circuits are defined throughout the network, but no bandwidth is allocated to the paths until data needs to be transmitted. Frame relay provides multiple logical connections within a single physical connection.

RFC 1315 is the MIB for Frame Relay DTEs (Data Terminal Equipment).

- RFC 1315
  - **Name:** Frame Relay
  - **Model Type Name:** FR_1315App
  - **Name:** Frame Relay
  - **Model Type Name:** rfc1315App

    NOTE
    The FR_1315App replaces the rfc1315App when Frame Relay Manager is installed.

RFC 2115 is the MIB for Frame Relay DTEs using SMIv2.

- RFC 2115
  - **Name:** FrameRelayApp
  - **Model Type Name:** rfc2115App

PNNI

PNNI (Private Network-to-Network Interface) is a suite of network protocols that can be used to discover an ATM network topology, create a database of topology information, and route calls over the discovered topology.

- PNNI-MIB
  - **Name:** PNNI_App
  - **Model Type Name:** PNNI_App

PPP

The PPP (Point-to-Point Protocol) application provides a method for transmitting datagrams over serial point-to-point links.

- Draft IETF-PPP-MIB
  - **Name:** PPP
  - **Model Type Name:** PPPLinkApp

SONET

SONET (Synchronous Optical Network) applications let you manage SONET/SDH (Synchronous Digital Hierarchy) interfaces. The following SONET standards are supported by DX NetOps Spectrum:

- RFC 1595

  - **Name:** SONET
  - **Model Type Name:** rfc1595App
- RFC 2558
  - **Name:** SONET
  - **Model Type Name:** rfc2558App

Accessing Standard Protocol Information in OneClick

This section describes how to access standards-based application information in OneClick. You can access application models and the information related to them from the Information tab of the device or by using the search functionality in the Locater tab.

Access Standards-Based Information Using the Information Tab

Depending on the device, you have modeled, you might have various OneClick subviews available to you from the Information tab in the Component Detail panel. From these subviews, you can view application model information.

To access standards-based information from a specific device model

1. Select the device from which you want to view RFC information.
2. In the Component Details panel, in the Information tab, expand the subview containing the information that you want to review.

   NOTE
   Various subviews are available depending on what MIB standards the device supported when it was modeled.

Locater Search

You can use the search functionality in the Locater tab to find application models and to subsequently view application model information. Search results appear in the Results tab of the Contents panel. Detailed information for application models that are selected in the results list appears in the Component Detail panel.

Search for Application Models

If you are not sure which device to access for the application models you want to view, you can use the search functionality of the Locater tab to list every application model that was created and to which you have access rights.

NOTE
If you are operating a Distributed SpectroSERVER (DSS) environment, some searches require you to select which landscapes to include in your search from the 'Select Landscapes to Search' dialog.

To access application models using Locater search

1. Click the Locater tab in the Navigation panel.
2. In the Locater tab, in the Name column, click 'Application Models' to expand it, and then do one of the following:
   - Double-click 'All Application Models.'
   - Click 'All Application Models,' and then click the 'Create a new search' button.
3. Enter any additional information if prompted, depending on the type of search you are running, and click OK.

   NOTE
   If additional input is not required, the search runs immediately and the search results appear in the Results tab of the Contents panel.
4. To narrow your search, enter further criteria in the Filter box at the top of the results list in the Results tab.

For example, if you were looking for the Repeater application, you could enter **1516** in the Filter box, which would find the Repeater model by its Model Type Name, 'rfc1516App.' Alternatively, you could enter **repeater** in the Filter box, which would find the Repeater model by its Name, '<*device_name*>_SNMP-Repeater.'
The Results tab refreshes to show only those models that match the criteria in the Filter box.

5. In the Results tab list, select the application model you want more information about.
The Component Detail view displays details about the selected application model.
6. Click any of the available tabs in the Component Detail view to review information specific to the selected application model.

NOTE
For more information about using Locater search functionality, see Using OneClick.

Manage Link Aggregation

IEEE 802.3ad-based link aggregation lets you aggregate two or more links together to form Link Aggregation Groups (LAGs), such that a MAC client considers LAG as a single link. DX NetOps Spectrum supports LACP (Link Aggregation Control Protocol) with the following features:

- Locater Search
- Spotlight View
- Threshold Alarm

LACP OneClick View

The LACP Port Aggregation table contains Link Aggregation Control configuration information of every Aggregation Port associated with a device.

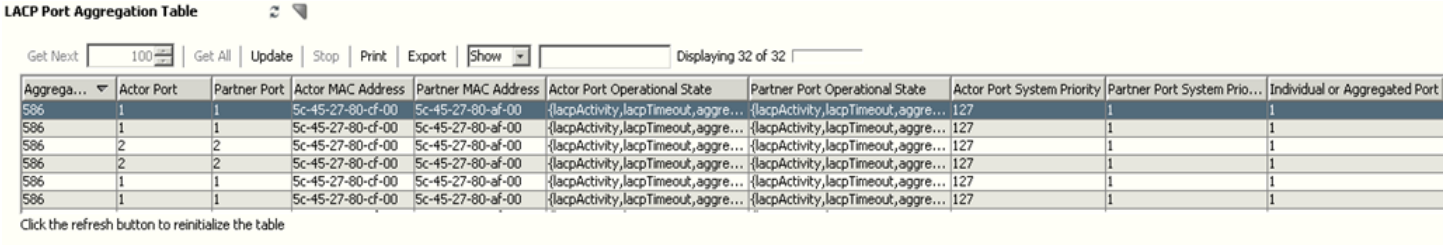
The following attributes are available in the LACP Port Aggregation Table:

- **Aggregator ID**
Indicates the ID number as the base port number or lowest number of an aggregator. Each aggregator must have an ID number. For example, an aggregator with ports 12, 16, and 17 must be assigned the ID number 12 because that is the base port number.
- **Actor Port**
Indicates a port number that is assigned to the port by the actor, encoded as an unsigned integer.
- **Partner Port**
Indicates a port number that is associated with the link assigned to the port by the partner, encoded as an unsigned integer.
- **Actor MAC Address**
Defines the value of the system ID for the system that contains the Aggregation Port.
Default: Six-digit MAC address.
- **Partner MAC Address**
Represents the current value of the Aggregation Port protocol.
Default: Six-digit MAC address.
- **Actor Port Operational State**
Indicates the operational values of Actor_State, transmitted by the actor in the Link Aggregation Control Protocol Data Units (LACPDU).
- **Partner Port Operational State**
Indicates a string of 8 bits, corresponding to the current values of Actor_State in the most recently received LACPDU transmitted by the protocol Partner. In the absence of an active protocol partner, this value might reflect the manually configured aAggPortPartnerAdminState value.
- **Actor Port System Priority**
Defines the priority value that is associated with the Actor System ID.

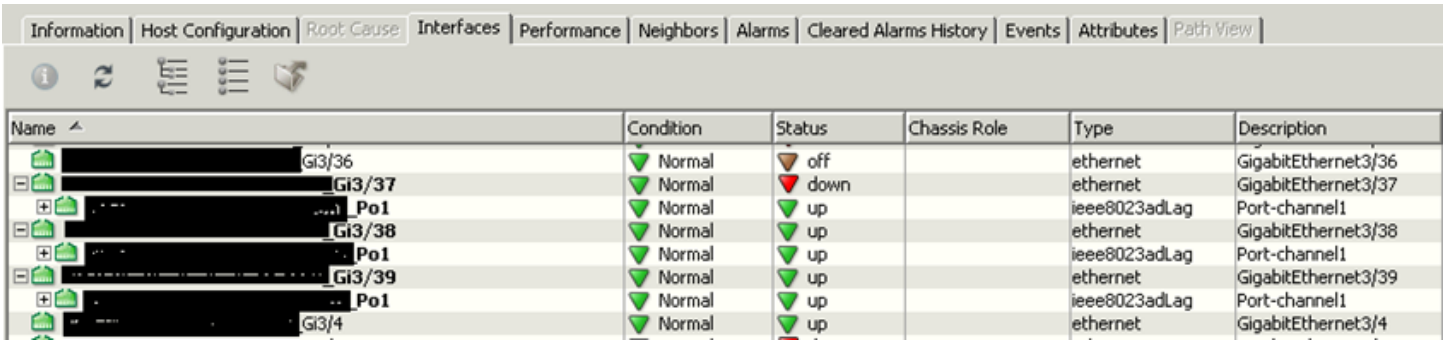
Default: Two-digit MAC address.

- **Partner Port System Priority**
  Indicates the current administrative value of the port priority for the protocol partner.
- **Individual or Aggregated Port**
  Represents a Boolean value of the Aggregation Port. This value indicates whether the Aggregation Port can Aggregate (TRUE) or can only operate as an Individual link (FALSE).

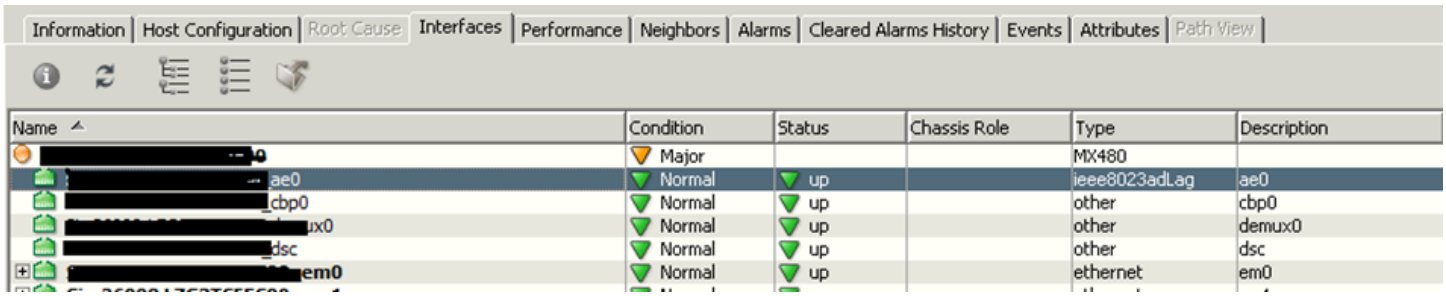
The following image displays the LACP Port Aggregation Table that is available in OneClick:



The port channel aggregation is displayed in the Interfaces tab of OneClick. The port channel is displayed as a sub-interface of aggregated Ethernet interfaces in the Interface tab. The LACP configured port channels are of type ieee8023adLag.
The following image displays the Interfaces tab of a device configured for LACP.



In case of VLAN tagging enabled on top of LACP configuration, the port channel is displayed separately and not as a sub-interface of aggregated Ethernet interfaces in the Interface tab of OneClick.
The following image displays the Interface tab in case of VLAN tagging enabled:



LACP Locater Search

You can use the search functionality in the Locater tab to find the devices that are configured with LACP and the application models that are associated with LACP. Search results appear in the Results tab of the Contents panel. Detailed information for application models that are selected in the results list appears in the Component Detail panel. Access LACP Locater search from the Locater tab of the Navigation

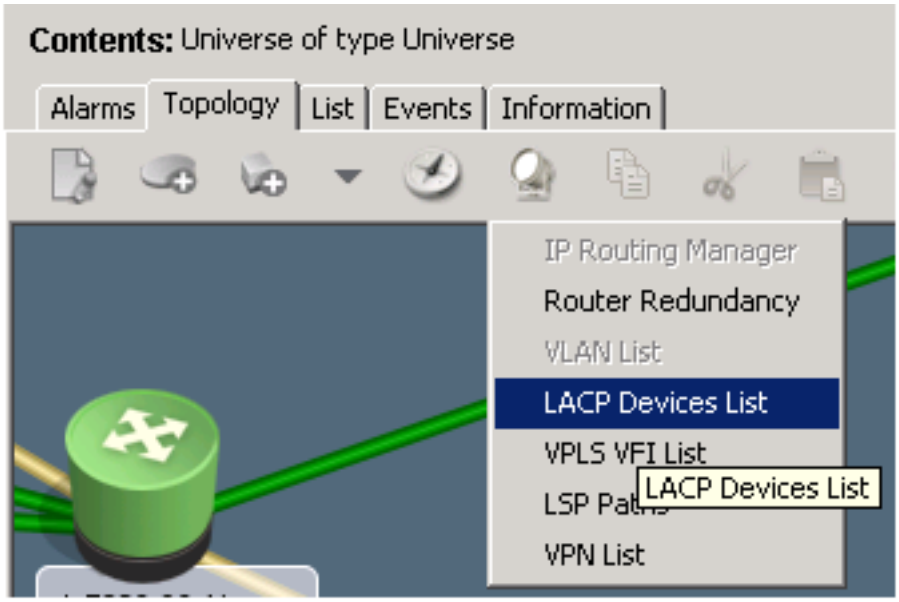
Follow these steps:

1. Open the OneClick Console.
2. From the Navigation Panel, Click the Locater tab.
   The Search Options window opens.
3. Select Devices, Supports Application, and LACP.
   The LACP Locater Search results are displayed in Contents panel.

Spotlight View

You can use the Spotlight feature in OneClick to view the devices that are configured with LACP on your network.

The following image displays the LACP Devices List option that is available in Spotlight View:



You can spotlight the listed items if they have been enabled and configured on your network. You can click the Spotlight View button that is available on the Topology toolbar and can select the LACP Devices List option to display the LACP configured devices.

LACP Threshold Computation

Spectrum generates various LACP threshold violation alarms on the LACP port channel model, such as critical, major, and minor. All the physical interfaces that are aggregated to the logical port of LACP participate in threshold calculation. You can use this threshold alarm to know the usage of the physical interfaces in an aggregate group of LACP.

The LACP threshold is calculated based on the IfOperStatus of the aggregated interfaces. To access IfOperStatus, you must have the admin link (based on external attribute, ifAdminStatus) status as UP.

```
down/totallinks
```

The following scenarios are considered during the threshold comparisons:

- If the LACP_IF_Port attributes are set to a value other than 0 but not greater than 100, the following attributes are used during the threshold comparisons:

  - LACP_Critical_Threshold = 0x2202c0
  - LACP_Major_Threshold = 0x2202bf
  - LACP_Minor_Threshold = 0x2202be
- If the LACP_IF_Port attributes are set to 0 or less than 0 or greater than 100, the following VNM attributes values are used during the threshold comparisons:
  - Critical_Threshold = 0x0001321e
  - Major_Threshold = 0x0001321d
  - Minor_Threshold = 0x0001321c
- If all the LACP port level attributes are set to 0 and the VNM LACP threshold attributes are set to greater than 100, no threshold is calculated.

Condition Correlation of LACP ports

Starting from the 10.2.2 release, the LACP threshold violation alarms functionality is enhanced to avoid multiple independent alarms (on port and port channel) for bad links. With this enhancement, the alarms generated on the ports that are part of an LACP aggregation are correlated and suppressed. Only the LACP threshold violation alarms are generated on the port channels and the alarms generated on the ports are suppressed and shown as symptoms.

Troubleshooting

Symptom
The LACP Port Aggregation Table does not contain Link Aggregation Control configuration information of every Aggregation Port associated with a device.

Solution

1. Verify the values in the dot3adAggPortAggregateOrIndividual table in IEEE8023-LAG-MIB. This indicates whether a port is attached to an LACP channel. If the port is attached to an LACP channel, the value can be TRUE, otherwise the value can be FALSE.
2. Verify the values in the dot3adAggPortAttachedAggID table in IEEE8023-LAG-MIB. Spectrum uses this table to get the aggregated port details.

Symptom
For Cisco devices, the LACP port channel type is displayed wrongly. It is displayed as propVirtual instead of ieee8023adLag in Interfaces tab of OneClick.
SolutionConfigure the 'IsLACP_PropVirtual=true in. vnmrc' file, then delete and rediscover the devices. If there are multiple devices, use the LACP migration script.

Symptom The Lacp_HASPART associations are not created correctly for LACP configured port channels. **Solution** Verify whether LACP is correctly configured on the device and whether the values are correct in the dot3adAggPortAttachedAggID table in IEEE8023-LAG-MIB. Spectrum uses this table to get the aggregated port details.

Symptom The LACP threshold violation alarms are not getting generated correctly. **Solution** Verify whether the Lacp_HASPART associations are correctly created for LACP configured port channels. There should be Lacp_HASPART association that are created between:

i) the device (left model handle) and port channel (right model handle).
ii) the port channel (left model handle) and the aggregated physical interfaces (right model handle).

Symptom The link information tab is not displaying all the links correctly. **Solution** Verify whether the CDP/LLDP is configured correctly for the device. For example, following are the OIDs Spectrum checks in case of Juniper devices in which LLDP is configured.

1.0.8802.1.1.2.1.3.7.1.2 (lldpLocPortIdSubtype)
1.0.8802.1.1.2.1.3.7.1.3 (lldpLocPortId)

1.0.8802.1.1.2.1.3.7.1.4 (lldpLocPortDesc)

Diagnostics/Logs

In case, the above troubleshooting did not help to resolve the problem, enable the following logging.

1. Enable the LACP logging on SpectroSERVER by adding the configuration lacp_debug=true in the .vnmrc file. This change requires a SpectroSERVER restart.
   After enabling the above logging, reproduce the issue and collect the VNM.out file.
2. Take an attributes dump of the LACP application model associated with the problematic device. To find the associated LACP application model, click Locater Search -> Application Models -> By Device Name. After listing the application models associated with the device, filter the list by 802dot3adApp.

Reconfigure LACP-Enabled Interface Models

When you upgrade 9.3 to 10.0, the LACP modeling requires deletion and rediscovery of the affected models. To eliminate the problem of manually deleting and re-discovering the effected models, run the RecreateLACPModels.pl script, which migrates the LACP configured devices without any problem.

Click to download the RecreateLACPModels.pl script attached to this page.

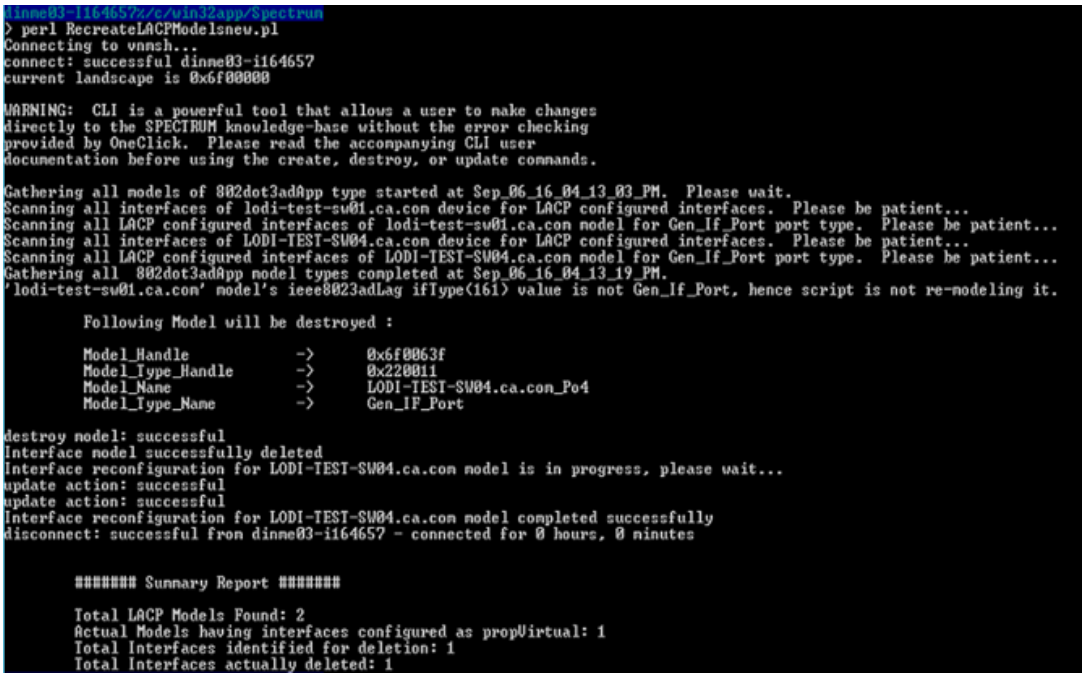
Use the script to:

1) Identify the LACP configured devices by reading the LACP mib (IEEE8023-LAG-MIB).
2) Delete the LACP configured interfaces/port channels and model them correctly.

After the migration, follow these steps to run the script:

1. Copy the downloaded script to $SPECROOT
2. Set the IsLACP_PropVirtual=true in the vnmrc file, which is located at $SPECROOT\SS
3. To Run the script manually, connect to the bash prompt where the SpectroSERVER installed
4. Go to the $SPECROOT Directory (cd /c/win32app/Spectrum)
5. Run the script (RecreateLACPModels.pl) using the Perl

For your reference, following is a sample output of the RecreateLACPModels.pl command.



RFC Reference

The following is a list of all RFCs that are supported by DX NetOps Spectrum. RFC support is typically provided in the form of application models, which allows it to be dynamically applied to any device model in DX NetOps Spectrum that is identified as supporting the particular RFC, including GnSNMPDev.

RFC Number	RFC Name	RFC Description
RFC1286	RFC1286-MIB	Definitions of Managed Objects for Bridges.
RFC1493	BRIDGE-MIB	Definitions of Managed Objects for Bridges.
RFC1474	PPP-BRIDGE-NCP-MIB	Definitions of Managed Objects for Bridges.
RFC2674	Q-BRIDGE-MIB	Management Information Base for Network Management of TCP/IP-based Internets: MIB-II.
RFC1525	SOURCE-ROUTING-MIB	Definitions of Managed Objects for Source Routing Bridges.
RFC1289	RFC1289-phivMIB	DECnet Phase IV MIB Extensions.
RFC1559	DECNET-PHIV-MIB	DECnet Phase IV MIB Extensions.
RFC2662	ADSL-LINE-MIB	Definitions of Managed Objects for the ADSL Lines.
RFC2669	DOCS-CABLE-DEVICE-MIB	DOCSIS Cable Device MIB Cable Device Management Information Base for DOCSIS Compliant Cable Modems and Cable Modem Termination Systems.
RFC2670	DOCS-IF-MIB	Radio Frequency (RF) Interface Management Information Base for MCNS/ DOCSIS Compliant RF Interfaces.

RFC2737	ENTITY-MIB	The MIB module for representing multiple logical entities supported by a single SNMP agent.
RFC1514	HOST-RESOURCES-MIB	Host Resources MIB.
RFC2790	HOST-RESOURCES-MIB	Host Resources MIB.
RFC1565	APPLICATION-MIB	Network Services Monitoring MIB.
RFC2248	NETWORK-SERVICES-MIB	Network Services Monitoring MIB.
RFC2788	NETWORK-SERVICES-MIB	Network Services Monitoring MIB.
RFC1566	MTA-MIB	Mail Monitoring MIB.
RFC2249	MTA-MIB	Mail Monitoring MIB.
RFC2789	MTA-MIB	Mail Monitoring MIB.
RFC1567	DSA-MIB	X.500 Directory Monitoring MIB.
RFC2605	DSA-MIB	X.500 Directory Monitoring MIB.
RFC1628	UPS-MIB	UPS Management Information Base.
RFC2287	SYSAPPL-MIB	Definitions of System-Level Managed Objects for Applications. The MIB module defines management objects that model applications as collections of executables and files installed and executing on a host system. The MIB presents a system-level view of applications; objects in this MIB are limited to those attributes that can typically be obtained from the system itself without adding special instrumentation to the applications.
RFC792		Internet Control Message Protocol (ICMP).
RFC2933	IGMP-STD-MIB	The MIB module for IGMP Management.
RFC1158	RFC1158-MIB	Management Information Base for Network Management of TCP/IP-based internets: MIB-II.
RFC1213	RFC1213-MIB	Management Information Base for Network Management of TCP/IP-based internets: MIB-II.
RFC1354	RFC1354-MIB	Management Information Base for Network Management of TCP/IP-based internets: MIB-II.
RFC2011	IP-MIB	Management Information Base for Network Management of TCP/IP-based internets: MIB-II.
RFC2012	TCP-MIB	Management Information Base for Network Management of TCP/IP-based internets: MIB-II.
RFC2013	UDP-MIB	Management Information Base for Network Management of TCP/IP-based internets: MIB-II.
RFC2096	IP-FORWARD-MIB	Management Information Base for Network Management of TCP/IP-based internets: MIB-II.

RFC4293	IP-MIB	Management Information Base for Network Management of TCP/IP-based internets: MIB-II.
RFC2465	IPV6-TC	Management Information Base for Network Management of TCP/IP-based internets: MIB-II.
RFC2452	IPV6-TCP-MIB	Management Information Base for Network Management of TCP/IP-based internets: MIB-II.
RFC1573	IANAifType-MIB	The MIB module which defines the IANAifType textual convention, and thus the enumerated values of the ifType object defined in MIB-II's ifTable.
RFC1907	SNMPv2-MIB	The MIB module for SNMPv2 entities.
RFC2233	IF-MIB	The MIB module to describe generic objects for network interface sub-layers. This MIB is an updated version of the MIB-II ifTable, and incorporates the extensions defined in RFC1229.
RFC2863	IF-MIB	The MIB module to describe generic objects for network interface sub-layers. This MIB is an updated version of the MIB-II ifTable, and incorporates the extensions defined in RFC1229.
RFC2213	INTEGRATED-SERVICES-MIB	The MIB module to describe the Integrated Services Protocol.
RFC2932	IPMROUTE-STD-MIB	The MIB module for management of IP Multicast routing, but independent of the specific multicast routing protocol in use.
RFC2667	TUNNEL-MIB	The MIB module for management of IP Tunnels, independent of the specific encapsulation scheme in use.
RFC2934	PIM-MIB	The MIB module for management of PIM routers.
RFC2271	SNMP-FRAMEWORK-MIB	An Architecture for Describing SNMP Management Frameworks.
RFC3411	SNMP-FRAMEWORK-MIB	An Architecture for Describing SNMP Management Frameworks.
RFC3413	SNMP-TARGET-MIB	Defines five types of SNMP applications that make use of an SNMP engine as described in STD 62, RFC3411.
RFC3584	SNMP-COMMUNITY-MIB	Describes the coexistence between Version 1, Version 2, and Version 3 of the Internet-Standard Network Management Framework.
RFC3415	SNMP-VIEW-VASED-ACM-MIB	View-Based Access Control Model (VACM) for the Simple Network Management Protocol.
RFC1316	RFC1316-MIB	Definitions of Managed Objects for Character Stream Devices using SMIv2.

RFC1658	CHARACTER-MIB	Definitions of Managed Objects for Character Stream Devices using SMIv2.
RFC1284	RFC1284-MIB	Definitions of Managed Objects for the Ethernet-like Interface Types.
RFC1398	RFC1398-MIB	Definitions of Managed Objects for the Ethernet-like Interface Types.
RFC1623	EtherLike-MIB	Definitions of Managed Objects for the Ethernet-like Interface Types.
RFC1643	EtherLike-MIB	Definitions of Managed Objects for the Ethernet-like Interface Types.
RFC2665	EtherLike-MIB	Definitions of Managed Objects for the Ethernet-like Interface Types.
RFC1285	RFC1285-MIB	FDDI Management Information Base.
RFC1512	FDDI-SMT73-MIB	FDDI Management Information Base.
RFC3621	POWER-ETHERNET-MIB	An extension to the Ethernet-like Interfaces MIB with a set of objects for managing Power Sourcing Equipment (PSE).
RFC1317	RFC1317-MIB	Definitions of Managed Objects for RS-232-like Hardware Devices using SMIv2.
RFC1659	RS-232-MIB	Definitions of Managed Objects for RS-232-like Hardware Devices using SMIv2.
RFC1318	RFC1318-MIB	Definitions of Managed Objects for Parallel-printer-like Hardware Devices using SMIv2.
RFC1660	PARALLEL-MIB	Definitions of Managed Objects for Parallel-printer-like Hardware Devices using SMIv2.
RFC1231	RFC1231-MIB	IEEE 802.5 MIB using SMIv2.
RFC1743	TOKENRING-MIB	IEEE 802.5 MIB using SMIv2.
RFC1748	TOKENRING-MIB	IEEE 802.5 MIB using SMIv2.
RFC1271	RFC1271-MIB	Remote Network Monitoring Management Information Base Version 2 using SMIv2.
RFC1757	RMON-MIB	Remote Network Monitoring Management Information Base Version 2 using SMIv2.
RFC2021	RMON2-MIB	Remote Network Monitoring Management Information Base Version 2 using SMIv2.
RFC2925	DISMAN-PING-MIB	Definitions of Managed Objects for Remote Ping, Traceroute, and Lookup Operations.
RFC4560	DISMAN-PING-MIB	Definitions of Managed Objects for Remote Ping, Traceroute, and Lookup Operations.
RFC1269	RFC1269-MIB	Definitions of Managed Objects for the Fourth Version of the Border Gateway Protocol (BGP-4) using SMIv2.
RFC1657	BGP4-MIB	Definitions of Managed Objects for the Fourth Version of the Border Gateway Protocol (BGP-4) using SMIv2.
RFC1243	RFC1243-MIB	AppleTalk Management Information Base II.
RFC1742	APPLETALK-MIB	AppleTalk Management Information Base II.
RFC1248	RFC1248-MIB	OSPF Version 2 Management Information Base.

RFC1252	RFC1252-MIB	OSPF Version 2 Management Information Base.
RFC1253	RFC1253-MIB	OSPF Version 2 Management Information Base.
RFC1850	OSPF-MIB	OSPF Version 2 Management Information Base.
RFC1368	SNMP-REPEATER-MIB	Definitions of Managed Objects for IEEE 802.3 Repeater Devices.
RFC1516	SNMP-REPEATER-MIB	Definitions of Managed Objects for IEEE 802.3 Repeater Devices.
RFC2108	SNMP-REPEATER-MIB	Definitions of Managed Objects for IEEE 802.3 Repeater Devices.
RFC3289	DIFFSERV-DSCP-TC	The Textual Conventions defined in this module should be used whenever a Differentiated Services Code Point is used in a MIB.
RFC1724	RIPv2-MIB	The MIB module to describe the RIP2 Version 2 Protocol.
RFC2837	FIBRE-CHANNEL-FE-MIB	The MIB module for Fibre Channel Fabric Element.
RFC2618	RADIUS-AUTH-CLIENT-MIB	The OID assigned to Remote Access Dialin User Service (RADIUS) MIB work by the IANA.
RFC2620	RADIUS-ACC-CLIENT-MIB	The MIB module for entities implementing the client side of the RADIUS accounting protocol.
RFC2574	SNMP-USER-BASED-SM-MIB	The management information definitions for the SNMP User-Based Security Model.
RFC3414	SNMP-USER-BASED-SM-MIB	The management information definitions for the SNMP User-Based Security Model.
RFC1696	Modem-MIB	Modem Management Information Base (MIB) using SMIv2.
RFC2787	VRRP-MIB	Definitions of Managed Objects for the Virtual Router Redundancy Protocol (VRRP).
RFC1695	ATM-MIB	Definitions of Managed Objects for ATM Management.
RFC2514	ATM-TC-MIB	Definitions of Managed Objects for ATM Management.
RFC2515	ATM-MIB	Definitions of Managed Objects for ATM Management.
RFC1232	RFC1232-MIB	Definitions of Managed Objects for the DS1, E1, DS2, and E2 Interface Type.
RFC1406	RFC1406-MIB	Definitions of Managed Objects for the DS1, E1, DS2, and E2 Interface Type.
RFC2495	DS1-MIB	Definitions of Managed Objects for the DS1, E1, DS2, and E2 Interface Type.
RFC1233	RFC1233-MIB	Definitions of Managed Objects for the DS3/E3 Interface Type.

RFC1407	RFC1407-MIB	Definitions of Managed Objects for the DS3/E3 Interface Type.
RFC2496	DS3-MIB	Definitions of Managed Objects for the DS3/E3 Interface Type.
RFC1315	RFC1315-MIB	Management Information Base for Frame Relay DTEs Using SMIv2.
RFC2115	FRAME-RELAY-DTE-MIB	Management Information Base for Frame Relay DTEs Using SMIv2.
RFC1595	SONET-MIB	Definitions of Managed Objects for the SONET/SDH Interface Type.
RFC2558	SONET-MIB	Definitions of Managed Objects for the SONET/SDH Interface Type.
RFC5412		Lightweight Access Point Protocol (LWAPP)
RFC5415		Controlling And Provisioning of Wireless Access Points (CAPWAP) Protocol Specification
RFC5416		Controlling and Provisioning of Wireless Access Points (CAPWAP) Protocol Binding for IEEE 802.11

VPLS Manager Solution

This section provides concepts, processes, and procedures for installing and configuring VPLS Manager. It also describes how to discover and model the VPLS environment, and how to monitor the performance of the VFIs and Sites.

Getting Started with VPLS

About VPLS Manager

VPLS Manager is a DX NetOps Spectrum add-on application that provides management tools to service providers deploying VPLS technology in their environment. Service providers who offer Layer 2 MPLS VPN service to their customers can benefit from using VPLS Manager to monitor their environment. In combination with the MPLS VPN Manager application, VPLS Manager provides seamless management of both Layer 3 (BGP) and Layer 2 MPLS VPNs within the same navigation and viewing framework. The primary benefits of the product are the discovery, modeling, and monitoring of Layer 2 networks, services, and resources. As an integrated application of the DX NetOps Spectrum product family, VPLS Manager benefits from the DX NetOps Spectrum ability to manage the entire network.

NOTE
For more information about the MPLS VPN Manager application, see MPLS VPN Manager .

Fault management is a key capability provided by VPLS Manager. VPLS Manager constantly monitors the resources used to provide VPLS to customers and intelligently computes the condition of the VPLS environment. For example, VPLS Manager monitors the devices, interfaces, configurations, and circuits with which the VPLS service is implemented. Monitoring is accomplished through polling and intelligent trap handling.

Another key capability is performance management of your VPLS environment. Performance management is accomplished by monitoring the traffic on VPLS-enabled interfaces and displaying them in real time.

VPLS Manager is designed to be a distributed SpectroSERVER (DSS) enabled application. This design provides the application significant scalability and capacity, supporting management of the largest networking environments.

Who Should Use VPLS Manager

VPLS Manager is designed for service providers who are running a VPLS core environment and must monitor their client network traffic for performance and accuracy within this network. VPLS Manager supports service provider environments that use Juniper equipment in their MPLS core and edge networks.

MIBs and Devices Supported by VPLS Manager

VPLS Manager manages Virtual Private LAN Service (VPLS) as described in the draft "RFC 4761 VPLS using BGP for Auto-Discovery and Signaling." These MIBs provide access to the configuration information for Layer 2 VPNs configured on PE router interfaces.

VPLS Manager supports the following tables in the Juniper Enterprise VPN MIB:

- Table of Configured VPNs (jnxVpnTable)
- Table of VPN Interfaces (jnxVpnIfTable)

VPLS Manager functionality is supported by Juniper on JunOS 6.1 or later.

System Requirements

VPLS Manager is an add-on application that works within DX NetOps Spectrum. In addition to a running SpectroSERVER installation, VPLS Manager requires the following:

- SNMP access to all PE devices in the VPLS service.
- Juniper VPN MIB implemented and populated on your MPLS hardware for Juniper.

  NOTE

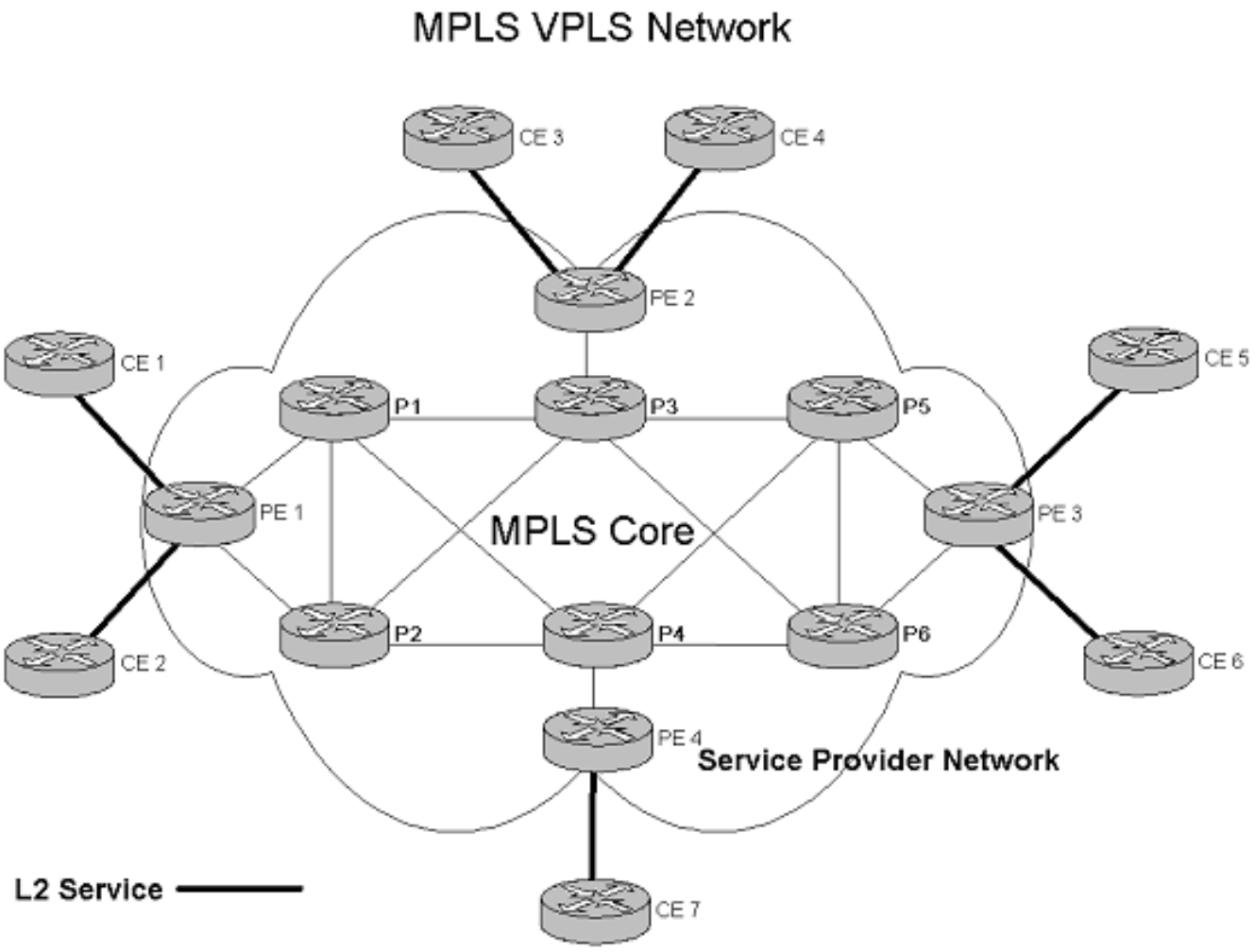
  DX NetOps Spectrum comes with all required MIBs.
- Management module for Juniper routers.

  NOTE

  The device models in this module have the necessary application models needed for discovery of your VPLS environment.

How VPLS Manager Works

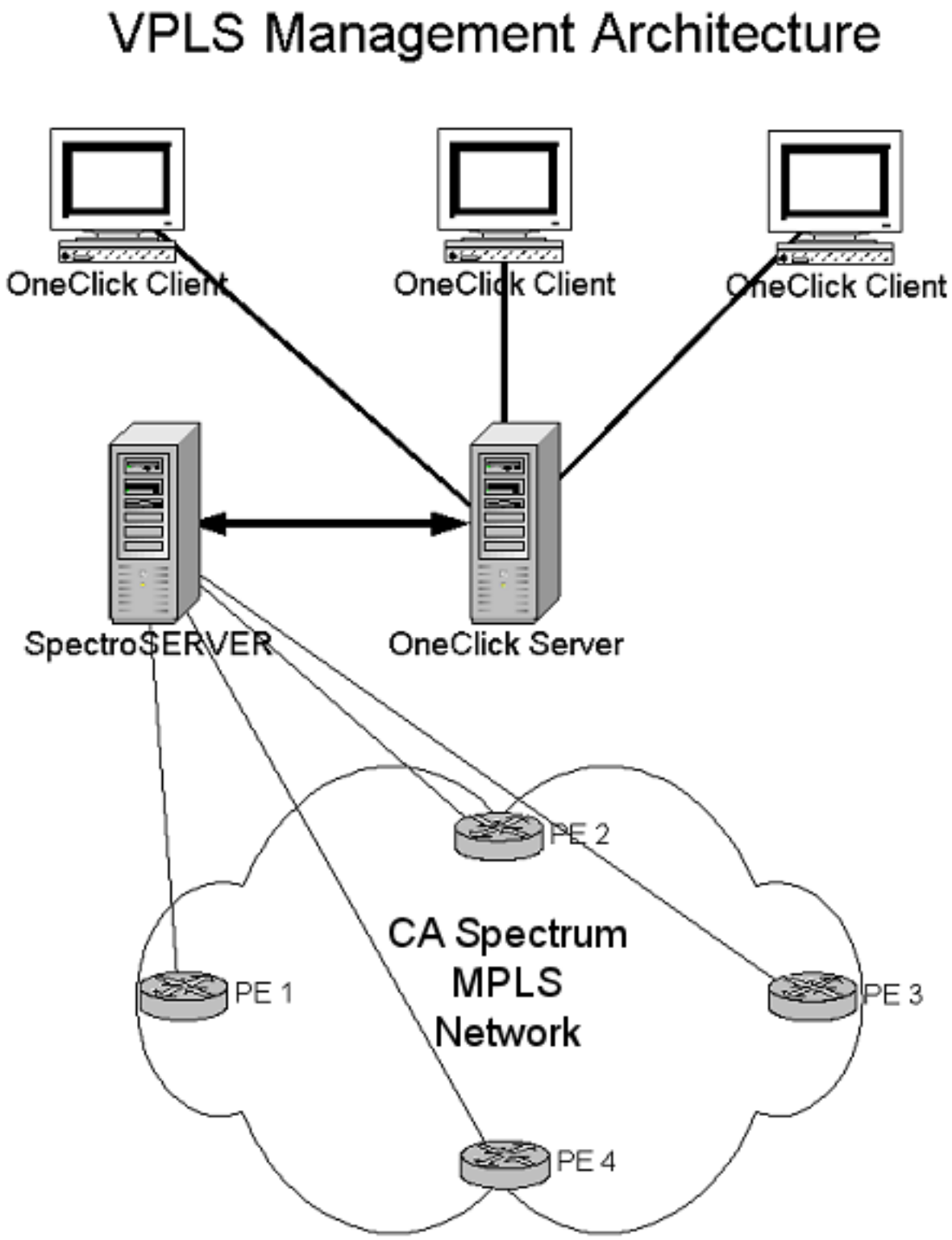
As a service provider, your goal is to use your MPLS VPLS core network to transport data packets from one part of your customer's network to another. For example, your MPLS network may transfer data packets from your customer's New York office to their London office. In this situation, your customer cannot model and monitor their network traffic through your MPLS VPLS network, and you, as a service provider, typically cannot model and monitor the client networks in New York or London. The following diagram shows that in this scenario the only interface between the customer network and your MPLS VPLS network is the relationship between their Customer Edge (CE) routers and your Provider Edge (PE) routers:



As shown by the architectural diagram, the major components of this environment are the following entities:

- Provider’s core MPLS network (P routers)
- Provider’s edge routers (PE routers)
- Customer edge routers (CE routers)

VPLS Manager focuses on the information provided by the PE routers and does not utilize nor provide any information on what exists in the MPLS core. As shown in the following deployment diagram, the management of the customer edge equipment is not part of VPLS Manager:


VPLS Management Architecture
OneClick Client
OneClick Client
OneClick Client
SpectroSERVER
OneClick Server
PE 2
CA Spectrum
MPLS
Network
PE 1
PE 3
PE 4

As shown in the diagrams above, VPLS Manager monitors your MPLS VPLS network environment as follows:

1. The PE routers use Layer 2 services to gather information from the CE routers, including the discovery of configuration and performance information from the supported MIBs. In addition, traps are supported where they are available.
2. The SpectroSERVER communicates with your PE routers when executing Discovery, modeling, polling, and fault management operations. All database operations with respect to modeling are handled by the SpectroSERVER.
3. The OneClick server periodically polls the SpectroSERVER database for management information, then displays this information in the OneClick clients.

VPLS Manager

Install VPLS Manager

VPLS Manager is included in your DX NetOps Spectrum extraction key. When you install DX NetOps Spectrum, the VPLS Manager components are automatically installed and available for use. For best results, you can adjust the configuration settings appropriately after installation is complete.

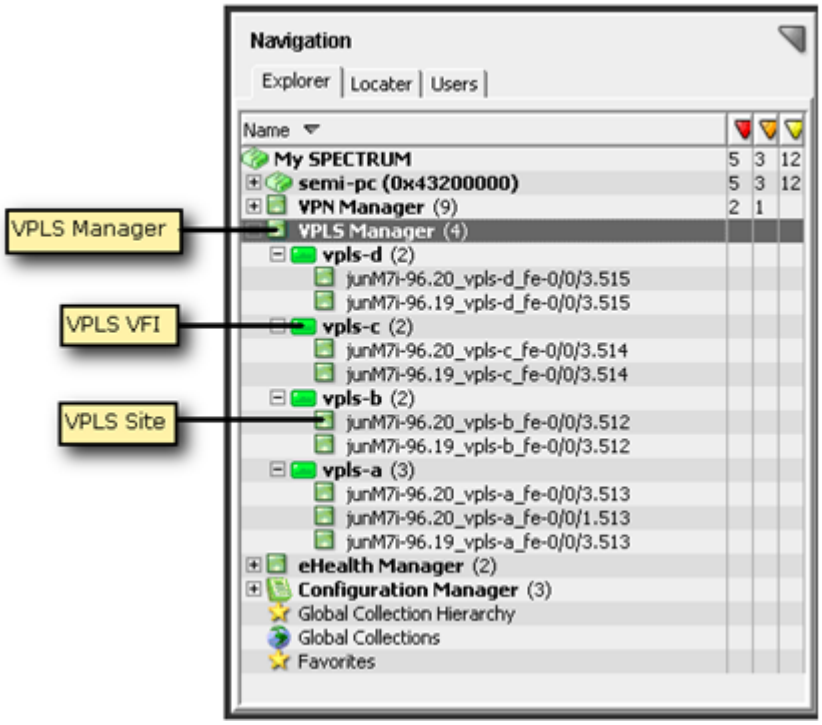
To install VPLS Manager properly, the administrator must install DX NetOps Spectrum.

For existing DX NetOps Spectrum installations, perform an in-place installation to help ensure the VPLS Manager components are installed. In a fault tolerant environment, VPLS Manager must be installed on each SpectroSERVER to be fault tolerant.

> **NOTE**
> Perform this procedure on all SpectroSERVERs in a distributed environment for which you want to use VPLS Manager. For specific installation instructions, see *the* Installation *section*.

Accessing VPLS Manager

You can access VPLS Manager from the Explorer tab of the Navigation panel. Expanding the VPLS Manager node displays all VFIs managed by the VPN Manager. Expanding each VFI displays the VPLS sites contained in the VPLS environment. The following is an example of the VPLS Manager hierarchy in the Navigation panel.



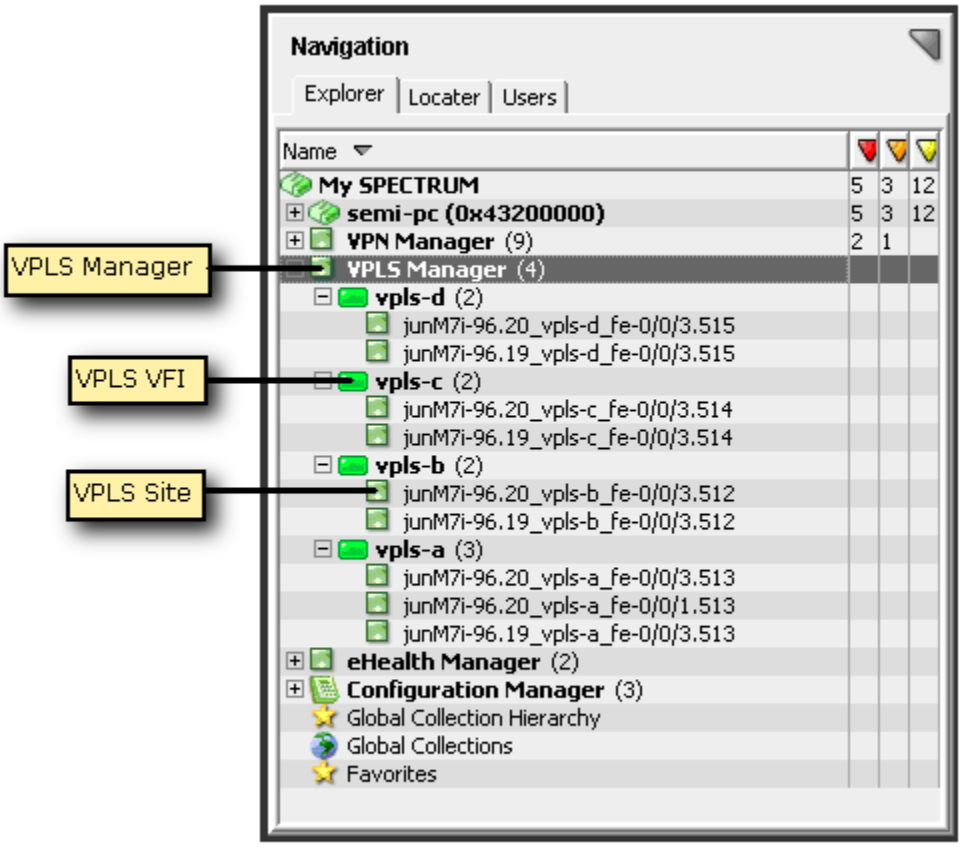
The Contents panel displays the Alarm list for the modeled element that you have selected in the Navigation panel. The Component Detail panel displays the Alarm details for the Alarm selected in the Alarm list shown in the Contents panel. If the Alarm list is empty, the Component Detail panel displays the Information view for the modeled element selected in the Navigation panel.

Viewing VPLS Manager Data

This section describes where to find VPLS Manager data in DX NetOps Spectrum, the model types, and additional options for viewing the available data. The content in this section is intended for all VPLS Manager users.

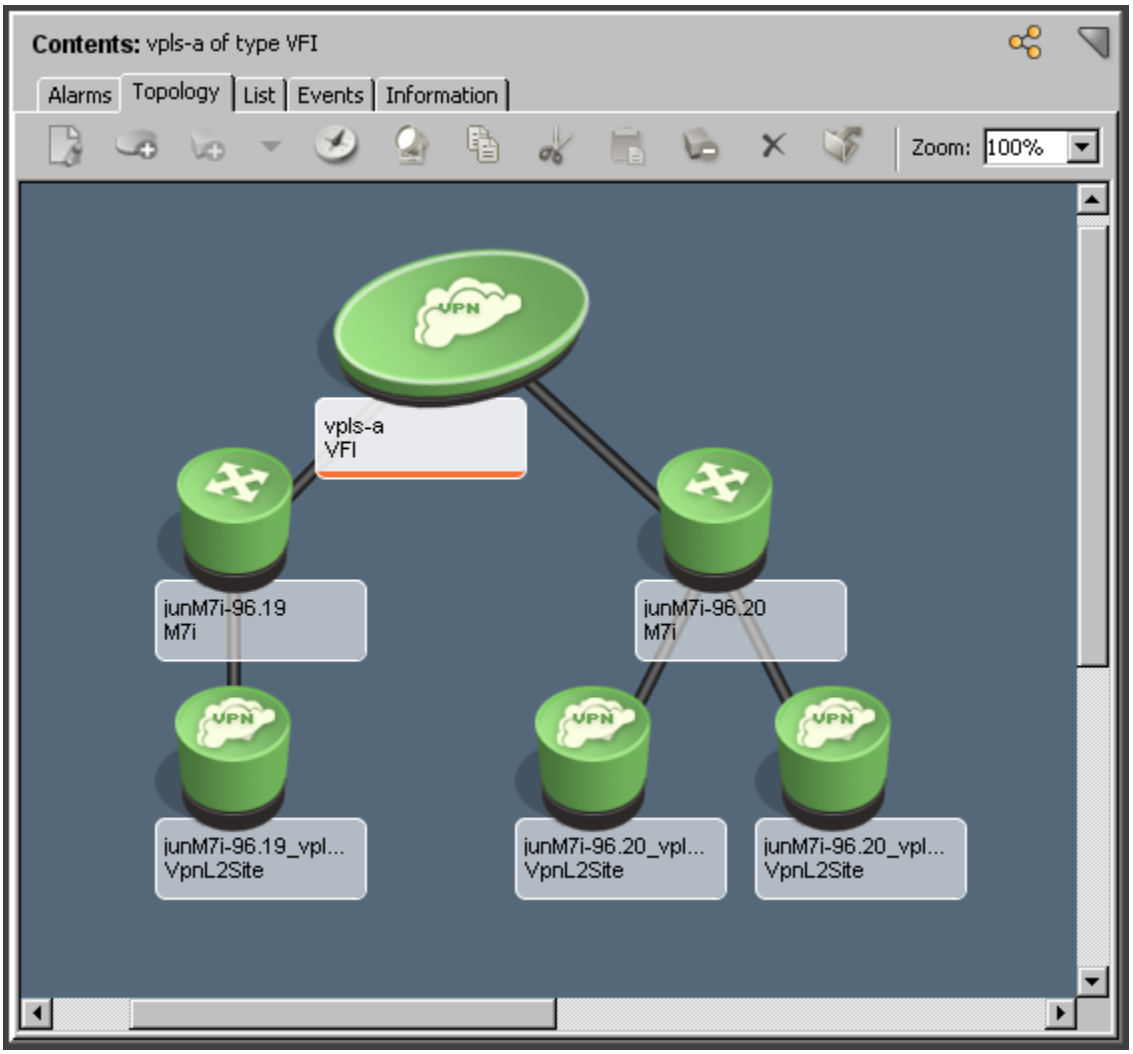
Navigation

OneClick is the main tool for interacting with VPLS Manager models. Using OneClick, you can configure the application, as well as view and search model information. All VPLS Manager navigation is contained within the VPLS Manager hierarchy in the Navigation panel, as shown in the following example:



Topology

VPLS Manager contains a topology view that shows how your VPLS entities are connected to other devices in your network environment. For example, the following topology view shows a VPLS environment with the VFI model (vpls-a) connected to two PE routers (junM7i-96.19 and junM7i-96.20). Then, these PE routers are connected to the VFI Site models, as shown:



VPLS Manager Model Types

During VPLS Discovery, several DX NetOps Spectrum models are created to represent different aspects of the MPLS/BGP VPN MIB in VPLS Manager. DX NetOps Spectrum uses these models to build your VPLS environment topology and reflect the current status of each entity in that environment.

The VPLS Manager topology includes the following model types:

- **VPLS Manager**
  Represents the VPLS Manager application. By default, DX NetOps Spectrum creates this model when the application is installed with DX NetOps Spectrum. This model cannot be destroyed. This application model must be present for VPLS Discovery to successfully discover MPLS-VPLS information.
  Model Type: VplsManager
- **VFI**
  Represents each unique VPLS VFI. A *Virtual Forwarding Instance (VFI)* is a logical collection of VPLS Sites that are part of the same virtual packet forwarding instance. These VPLS Sites share a common Layer 2 forwarding database, much the same way that Layer 3 devices may share a common routing table. The Model Class attribute for the VFI model is set to Transport Service.

Model Type: VFI

- **VPLS Site**
  Represents each unique VPLS Site modeled during VPLS Discovery. A *VPLS Site* represents the service delivered to a customer over an interface in your VPLS environment. DX NetOps Spectrum creates a VPLS Site model on the same SpectroSERVER on which its associated PE router is modeled. The Model Class attribute for the VPLS Site model is set to Transport Service. VPLS Manager assumes that each VPLS Site is connected to a given PE by a single interface. By default, DX NetOps Spectrum does not create VpnL2Site models for loopback and tunnel interfaces.
  Model Type: VpnL2Site

VPLS Site Model Names

DX NetOps Spectrum generates unique names for VPLS Site models during VPLS Discovery. The naming convention is as follows:

```
RouterName_VFI_IfName
```

To help ensure the name is unique, the naming convention is based on the following three parts:

- *RouterName* -- The PE router associated with the VPLS Site
- *VFI* -- The associated VFI
- *IfName* -- The IfName of the interface to which the VPLS Site is connected

View Associated VPLS Sites

For an individual VFI model, you can view a list of associated VPLS Sites. Using the list, you can quickly see the condition of related VPLS Sites that can affect the performance of the selected VFI.

To view a list of associated VPLS Sites

1. Open VPLS Manager in the Navigation panel.
   The main details page for VPLS Manager opens in the Contents panel.
2. Select the VFI model
   Details about the selected model display in the Contents panel.
3. Click the Information tab.
4. Expand the Associated Sites subview.
   A table lists all of the VPLS Site models associated with the selected VFI model.

View Associated PE Routers

For an individual VFI or VPLS Site model, you can view a list of associated PE routers. Using the list, you can quickly see the condition of all related routers that can affect the performance of the selected model.

To view a list of associated PE routers

1. Open VPLS Manager in the Navigation panel.
   The main details page for VPLS Manager opens in the Contents panel.
2. Select the VFI or VPLS Site model.
   Details about the selected model display in the Contents panel.
3. Click the Information tab.
4. Expand the Associated Edge Routers subview.
   A table lists all of the PE routers associated with the model selected in step 2.

Spotlighting VFIs

The spotlighting feature in OneClick lets you isolate and visualize model relationships within your network that are not readily visible from the Topology view. For example, the Topology view does not visually distinguish VFIs and VPLS Sites, making it more difficult to picture these relationships within the context of your network. With spotlighting, these model relationships are accentuated, showing you where they appear in the network topology.

Using the spotlighting feature, you can select a VFI to view in the Topology view. Viewing VFI information from this view can help you more easily understand which devices are related to the selected VFI. From this view, you can also see if any alarming devices are impacting the performance of the VFI.

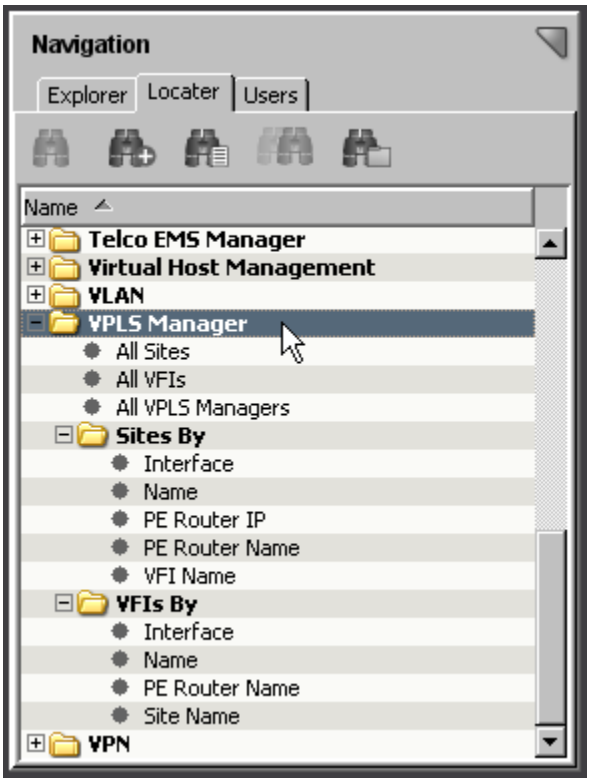
> **NOTE**
> For more information about how to use spotlighting, see Using OneClick.

VPLS Manager Searches

Using the search feature can help you find details about your VPLS environment that help you monitor the performance. Searches can locate specific components of your MPLS-VPLS core, such as locating VPLS Sites by PE router IP or all VFIs within a specific landscape. These types of searches can help you investigate information related to a specific customer, because you can use details associated with your customer SLAs in your searches, such as specific device or site name.

VPLS searches support cross-server device modeling. You can specify one or more, or all SpectroSERVERs to search for a particular VPLS Manager model, sets of VFIs, or VPLS Sites that are managed by a specific SpectroSERVER. You can use the search results to access a number of views that present management, performance, and configuration information.

The search options are grouped under the VPLS Manager folder, as shown:



For example, if you know the IP address or name of a specific PE router, you can search for all VPLS Sites that use it. Using the list of VPLS Sites affected by the router can be useful when performing scheduled maintenance. You can change your VPLS Sites to use a different router, or be sure to place the VPLS Site model in maintenance mode to avoid unnecessary alarms in DX NetOps Spectrum. Also, you can give prior notice to any customers affected by those VPLS Sites.

NOTE
For more information about Locater searches, see Using OneClick .

Discovery and Modeling (VPLS)

Before you can use VPLS Manager to manage your VPLS network, you must run VPLS Discovery. VPLS Discovery discovers each VFI and VPLS Site currently configured on devices modeled in DX NetOps Spectrum. Before running VPLS Discovery, you can configure the VPLS Discovery options to achieve the best results for your environment.

Discovery Prerequisites

For VPLS Discovery to complete successfully, devices must meet the following prerequisites:

- VPLS devices must support the correct MPLS-VPLS MIBs.
- DX NetOps Spectrum discovers and models the physical network infrastructure via Discovery, manual modeling, or the Modeling Gateway. Before using the VPLS Discovery functionality, you must first model the physical components of your network in DX NetOps Spectrum using one of these methods.

  NOTE
  For instructions about using these mechanisms to model your network, see the Modeling and Managing Your IT Infrastructure and the Modeling Gateway Toolkit .
- The devices must have MPLS-VPLS properly configured.

Configure VPLS Manager for Discovery

Before you run VPLS Discovery, you can configure the VPLS Discovery options. Making these selections helps ensure that DX NetOps Spectrum finds and manages only the VPLS devices you want to monitor.

To configure VPLS Manager for VPLS Discovery

1. Open VPLS Manager in the Navigation panel.
   The main details page for VPLS Manager opens in the Contents panel.
2. Click the Information tab in the Contents panel.
3. Expand the Configuration, VPLS Discovery subview and make your selections from the following configuration options:
   - **Enable Dynamic Discovery**
     Determines whether to start the VPLS Discovery automatically when a new PE router is modeled. Starting VPLS Discovery automatically helps to keep the VPLS information current when new devices are added to the network.

     NOTE
     As the VPLS Manager application is running, VPLS sites may be created or destroyed when certain traps are received.

     Default: No
   - **VFI Name Filter Type**
     Determines if the VFI names in the 'Global VFI Name Filter' field are included or excluded from modeling. This feature can save unnecessary resources by limiting the number of VPLS sites that require monitoring. Options include the following:

- Exclusive
- Inclusive

– **Global VFI Name Filter**
Lists the VFI names to be included or excluded from modeling. This field is used together with the 'VFI Name Filter Type' field.

VPLS Discovery options are configured.

Discover and Model Your VPLS Environment

To view and monitor your VPLS environment in DX NetOps Spectrum, you must model your VPLS devices. Running VPLS Discovery is the most comprehensive method of modeling your VPLS environment, but it also requires the greatest amount of system resources. To accommodate your modeling needs, you can choose from the following options when discovering and modeling the VPLS devices in your environment:

- Run a full VPLS Discovery
- Run VPLS Discovery on selected models only
- How to Update the New VPLS VFI Interface
- Filter the full VPLS Discovery results
- Configure DX NetOps Spectrum modeling to include VPLS Discovery

Run VPLS Discovery

To discover and model your entire VPLS environment, you can run a VPLS Discovery. Before you run an on-demand VPLS Discovery, be sure that you meet the prerequisites and that you configured your VPLS Discovery options.

Follow these steps:

1. Open VPLS Manager in the Navigation panel.
   The main details page for VPLS Manager opens in the Contents panel.
2. Click the Information tab in the Contents panel.
3. Expand the Configuration, VPLS Discovery subview.
4. Click Run in the Discovery Status field.
   A dialog opens to request the landscapes on which you want to run VPLS Discovery.
5. Select the landscapes and click OK.
   VPLS Discovery runs. When complete, the VPLS Discovery Status field lists the status. Also, a tooltip on this field lists the number of discovered VPLS devices (single SpectroSERVER) or servers (distributed SpectroSERVER).

Run VPLS Discovery on Selected Models

In the available OneClick views for VPLS Manager, you can select a set of models and can run VPLS Discovery for those models only. This ability can help you minimize the DX NetOps Spectrum resources required when troubleshooting or verifying changes to the status of only specific devices.

To run VPLS Discovery on selected models

1. Select the models in OneClick.
2. Click Tools, Utilities, Network Services Discoveries, VPLS Discovery.
   The VPLS Discovery process is initiated for the selected models only. You can check the status in the Configuration, VPLS Discovery subview.

How to Update the New VPLS VFI Interface

When a new VPLS enabled device is added in your VPLS environment, you model it in your OneClick landscape. If you run the VPLS discovery, the new device is created as a child of the configured VPLS VFI under the VPLS Manager in

the Navigation pane. But the new interface is not updated in the Interfaces tab in the Component Detail pane for the new device under the VPN VFI. Manually update the new interface.

Follow these steps:

1. In the Navigation pane, select VPLS Manager.
2. In the Contents pane, select List.
   A list of all the VPLS VFIs is displayed.
3. Delete the configured VPLS VFI.
4. In the Component Detail pane, select Information and run the VPLS Discovery.
   The VPLS VFI model that was deleted is created with the same number of child devices.
5. Select Interfaces, and verify that the new interface is added.

The new interface is updated for the configured VPN VFI.

Filter VPLS Discovery Results

If you do not want to monitor all VFIs, you can apply a filter that includes or excludes selected VFIs from discovery and modeling. This feature can help save resources by reducing the number of VFIs DX NetOps Spectrum polls.

To filter VPLS Discovery results

1. Open VPLS Manager in the Navigation panel.
   The main details page for VPLS Manager opens in the Contents panel.
2. Click the Information tab in the Contents panel.
3. Expand the Configuration, VPLS Discovery subview.
   VPLS Discovery options display.
4. Click Set in the VFI Name Filter Type field and select one of the following options:
   – Exclusive
   – Inclusive
5. Click Add in the Global VFI Name Filter field.
   The Add dialog opens, prompting you to enter the VFI name.
6. Enter the VFI name and click OK.
   The VFI name is added to the Global VFI Name Filter list. Depending on the VFI Name Filter Type that you select, VPLS Discovery is filtered to include or exclude the listed VFIs.

Configure DX NetOps Spectrum Modeling to Include VPLS Discovery

DX NetOps Spectrum lets you configure modeling to include network services discoveries, such as VPLS Discovery. As a part of modeling configuration, you can specify which network service discoveries to run with the DX NetOps Spectrum modeling process.

NOTE
For more information, see the Modeling and Managing Your IT Infrastructure *section.*

Configuring VPLS Manager

This section describes how to configure VPLS Manager options. These are tasks that are typically performed by the VPLS Manager administrator.

VPLS Manager Model Configuration

After VPLS Discovery, you must configure VPLS Manager so that it manages your VPLS environment appropriately. The VPLS Manager application offers configuration options for VPLS Manager models, the individual VFI models,

and for VPLS Site models. In addition to the typical configuration options for DX NetOps Spectrum models, the VPLS Manager model lets you specify parameters for all the VFIs managed by the selected VPLS Manager model. Using these configuration options, you can optimize how DX NetOps Spectrum monitors your VPLS environment. Configuration options include the following:

- Port polling
- Traps for creating and deleting models

NOTE
By default, both options are enabled.

Configure Port Polling

With traps, the polling mechanism is used to determine the health of the resources that make up your VPLS environment. Although enabled by default, you can disable polling to reduce network traffic. However, disabling polling causes the loss of significant functionality in VPLS Manager. Port polling must be enabled to update the condition of VPN Site models.

NOTE
Only an administrator performs this task.

To configure port polling in your VPLS environment

1. Open VPLS Manager in the Navigation panel.
   The main details page for VPLS Manager opens in the Contents panel.
2. Click the Information tab in the Contents panel.
3. Expand the Configuration, Management Configuration subview.
   VPLS Manager configuration options display.
4. Click the 'set' link in the Enable Port Polling field.
   The value for the selected option becomes editable.
5. Select the desired value for the field and click Save.
   VPLS Manager is configured to poll devices in your VPLS environment according to your selection.

Configure Traps to Create and Delete Models (VPLS)

If your VPLS devices are properly configured to send traps to the SpectroSERVER host, you can use this trap data to create or delete VFI and VPLS Site models automatically. Traps that are sent from the devices in your VPLS environment can ensure that devices in your VPLS environment are not unmanaged. Likewise, traps can also help to keep the information in DX NetOps Spectrum accurate by eliminating VPLS devices that no longer exist.

Some environments do not support the use of traps, and you can choose to disable them. However, we recommend that you enable traps when possible, because traps (with polling) provide the best response to network faults and outages.

NOTE
Only an administrator performs this task.

To configure traps to create and delete VPLS device models automatically

1. Open VPLS Manager in the Navigation panel.
   The main details page for VPLS Manager opens in the Contents panel.
2. Click the Information tab in the Contents panel.
3. Expand the Configuration, Management Configuration subview.
   VPLS Manager configuration options display.
4. Click the 'set' link in the following fields, as needed:
   - **Create on Trap**
     Creates a VFI or VPLS Site when the mplsVrfIfUp/ mplsL3VpnVrfUp or jnxVpnIfUp trap is received and the device model already exists.

Default: Yes

- **Delete on Trap**
  Deletes an existing VFI or VPLS Site model when the mplsVrfIfDown/ mplsL3VpnVrfDown or jnxVpnIfDown is received and the device no longer exists in DX NetOps Spectrum.
  Default: Yes

The value for the selected option becomes editable.

5. Select the desired value for the field and click Save.
   The traps are configured to create or delete devices in your VPLS environment according to your selection.

VFI Model Configuration

After VPLS Discovery, you must configure VPLS Manager so that it manages your VPLS environment appropriately. The VPLS Manager application offers configuration options for VPLS Manager models, the individual VFI models, and for VPLS Site models. In addition to the typical configuration options for DX NetOps Spectrum models, the VFI model lets you specify parameters for the selected VFI and the VPLS Sites it manages. Using these configuration options, you can optimize how DX NetOps Spectrum monitors your VPLS environment. Configuration options include the following:

- Impact thresholds
- Alarms for VFIs or VPLS Sites

Set Impact Thresholds

The path of network traffic across your VPLS environment depends on which VPLS Sites are available for use. Network traffic can move most efficiently when all VPLS Sites for a VFI are available to your customers. Therefore, monitoring the performance of your VFIs can help ensure you provide your customers the correct level of service. To monitor the performance, you can set impact thresholds that trigger alarms when a percentage of the VPLS Sites for a VFI are no longer available.

To set impact thresholds for your VFIs

1. Open VPLS Manager in the Navigation panel.
   The main details page for VPLS Manager opens in the Contents panel.
2. Select a VFI in the Navigation panel.
   Details for the selected VFI display in the Contents panel.
3. Click the Information tab in the Contents panel.
4. Expand the Configuration Information subview.
   VFI configuration options display.
5. Click the 'set' link in the following fields, as needed:
   - **Critical Threshold %**
     Specifies the percentage of down (unreachable) VPLS Sites that cause DX NetOps Spectrum to generate a critical alarm for the model.
     Default: 15
     Limits: 0-100
     Note: A value of zero causes DX NetOps Spectrum to always generate a critical alarm when any VPLS Site becomes unreachable.
   - **Major Threshold %**
     Specifies the percentage of down (unreachable) VPLS Sites that cause DX NetOps Spectrum to generate a major alarm for the model.
     Default: 10
     Limits: 0-100
     Note: A value of zero causes DX NetOps Spectrum to always generate a major alarm when any VPLS Site becomes unreachable.
   - **Minor Threshold %**

Specifies the percentage of down (unreachable) VPLS Sites that cause DX NetOps Spectrum to generate a minor alarm for the model.
Default: 5
Limits: 0-100
Note: A value of zero causes DX NetOps Spectrum to always generate a minor alarm when any VPLS Site becomes unreachable.

The value for the selected option becomes editable.

6. Enter the desired value for the field and click Save.
   The impact thresholds are configured to generate alarms when the threshold values for the selected VFI are breached.

Enable or Disable Alarms

To alert you to problems within your VPLS environment, DX NetOps Spectrum can generate alarms for VFIs or VPLS Sites. For example, an event is triggered when contact is lost to one of these models. When alarms are enabled, DX NetOps Spectrum generates a corresponding alarm. Although both alarm types are enabled by default, you can disable these alarms for non-critical devices to reduce the number of unnecessary alarms.

To enable or disable alarms

1. Open VPLS Manager in the Navigation panel.
   The main details page for VPLS Manager opens in the Contents panel.
2. Select a VFI in the Navigation panel.
   Details for the selected VFI display in the Contents panel.
3. Click the Information tab in the Contents panel.
4. Expand the Configuration Information subview.
   VFI configuration options display.
5. Click the 'set' link in the following fields, as needed:
   - **Enable VFI Alarms**
     Determines whether DX NetOps Spectrum generates a VFI alarm when the condition of the VFI model changes. When this option is enabled and traps are properly configured, a contact lost event triggers an alarm that is asserted against the VFI model. When this option is disabled, a contact lost event causes the VFI model to turn red, but no alarm is created.
     Default: Yes
   - **Enable Site Alarms**
     Determines whether DX NetOps Spectrum generates a VPLS Site alarm when the condition of a VPLS Site managed by the selected VFI changes to critical. When this option is enabled, DX NetOps Spectrum generates an alarm. However, DX NetOps Spectrum does not generate an alarm when this option is disabled.
     Default: Yes

   The value for the selected option becomes editable.
6. Select the desired value for the field.
   The alarms for the selected VFI model are configured according to your selection.

Distributed SpectroSERVER Configuration

In a DSS environment, you can model your PE routers from any SpectroSERVER. DX NetOps Spectrum creates an L2Site model on the same SpectroSERVER on which its associated PE router is modeled. However, DX NetOps Spectrum creates all VFI models on the main location server (MLS). All the discovered VFIs appear in the VPLS Manager area of the OneClick console. The local SpectroSERVERs must have a connection to the MLS to support this distributed configuration.

WARNING
To create a distributed VPLS Manager environment, all VFI models must reside on the SpectroSERVER that is on the MLS. When you change the MLS, the VFI and VpnL2Site models become invalid and are deleted by each SpectroSERVER in the DSS environment. To change the MLS properly, see the procedure for designating a new MLS. For more information, see the Distributed SpectroSERVER Administration section.

Distributed SpectroSERVER Considerations

Review the following considerations before configuring a Distributed SpectroSERVER:

- When the same PE router is modeled on multiple landscapes, multiple site models are created. The site models have the same name (unless the device models are named differently) but reside on different landscapes. If an outage occurs, you can see multiple similar alarms because multiple sites are experiencing an outage.
- If you do not want to receive alarms on one set of the site models, use proxy models. Otherwise, DX NetOps Spectrum raises alarms on each site model.
- If you model the same device in multiple landscapes, name the device differently. Using different device names lets you easily differentiate the similar alarms in multiple landscapes.

Applying Global Configurations

In a DSS environment, you can apply configuration settings globally using the VPLS Manager model on the MLS. By default, the configuration options you set on the main VPLS Manager model are applied to the VPLS Managers on remote SpectroSERVERs. Setting a configuration option globally provides an efficient method for setting your preferences across all SpectroSERVERs.

If needed, you can override a global configuration on a remote VPLS Manager. For example, if you have a lab environment that is managed by a separate SpectroSERVER, you can turn off VFI alarms generated by the devices managed on that SpectroSERVER. Turning these off helps eliminate unnecessary alarms.

NOTE
Use caution when making overrides. Whenever possible, update the model information about the MLS as a best practice and use local overrides for any changes required on a specific SpectroSERVER.

Override a Global Configuration

By default, the configuration options you set on the main VPLS Manager model are applied to the VPLS Managers on remote SpectroSERVERs in a DSS environment. However, you can override these settings on a remote VPLS Manager. For example, if you have a lab environment that is managed by a separate SpectroSERVER, you can eliminate unnecessary alarms by turning off VFI alarms generated by the devices managed on that SpectroSERVER.

Follow these steps:

1. Search for the VPLS Manager you want to configure.
   The VPLS Manager model appears in your search results in the Contents panel.
2. Select the VPLS Manager model.
3. Click the Information tab in the Component Details panel.
4. Expand the Configuration, Management Configuration subview.
5. Click the 'set' link for the configuration option to override.
   The Local Override Panel displays. The Global value reflects the value set on the MLS VPLS Manager model.
6. Enter a value that is different from the global value, uncheck the 'Use global value' checkbox, and click Save.
   Your changes are saved, and the new configuration settings override the global settings for the selected VPLS Manager model.

NOTE
When a local override is being used, the Management Configuration subview displays an asterisk appended to the non-global attribute value.

Managing Models, Traps, and Alarms

This section describes the concepts and procedures for managing models, traps, and alarms generated for VPLS Manager devices.

VPLS Manager Alarms

To alert you to problems within your monitored networks, DX NetOps Spectrum generates alarms. When monitoring your VPLS environment, your VPLS Manager models display an alarm state for the following conditions:

- Site Down
- VFI Condition (Initial, Good, Minor, Major, Critical)

Trap Support

NOTE

Alcatel traps are not supported by the VPLS Manager for this release.

The following table lists the Juniper traps supported by VPLS Manager. Receipt of either a jnxVpnIfUp trap or a jnxVpnIfDown trap typically results in a change of the VFI model condition. Events are created based on changes in condition.

NOTE
Each device must be configured to send SNMP traps to the DX NetOps Spectrum Virtual Network Machine (VNM).

Trap	Result of Receiving Trap
jnxVpnIfUp	If the VFI model already exists, a change in status is reported. Otherwise, a new VFI model is created.
jnxVpnIfDown	If both the VFI model and the VPLS Site on the device sending this trap exist, a change in status is reported. Otherwise, the VFI model is deleted.
jnxVpnPwUp	If no other related outages are present, the status of the VPLS Site is upgraded to good.
jnxVpnPwDown	If the VPLS Site is no longer provisioned in the device, the VPLS Site is deleted.

VPLS Site Model Deletion

When you no longer need to monitor a VPLS Site, you can manually select the VPLS Site model and delete it. VPLS Site models are also deleted in the following situations:

- When you delete a VFI model, DX NetOps Spectrum automatically deletes all VPLS Site models associated with this VFI model.
- When you delete a PE router model, DX NetOps Spectrum deletes all of the associated VPLS Site models. When the last VPLS Site is deleted, DX NetOps Spectrum also deletes the VFI model associated with those VPLS Site models.
- If a PE router model sends a jnxVpnIfDown SNMP trap and the same VRF entry has been removed from the device's VRF Table, DX NetOps Spectrum deletes the VPLS Site model.

Monitoring Status and Performance

This section explains the information you need to monitor the performance of your VPLS environment and check your adherence to customer SLAs using VPLS Manager. This section is intended for general VPLS Manager operators.

Monitoring Performance and SLAs

For an ISP, continuity of service is crucial for each customer, which is why client SLAs are often established. Although all service outages are not avoidable, the customer relies on the ISP to keep the outages to a minimum, adhering to the terms agreed upon in their SLA.

Monitoring a VPLS environment for adherence to customer SLAs can be difficult. VPLS Manager makes this task easier by monitoring the performance of the VPLS devices and providing a way to view the relationship of those devices to your PE routers. By knowing which customers are impacted by a specific VFI, you can monitor SLAs when monitoring VPLS environment performance in one of the following ways:

- **Analyze the condition of your VFIs** -- Using fault monitoring, VPLS Manager continually updates you about the latest condition of your VPLS devices. These conditions can trigger traps and alarms. All alarms generated from the VPLS environment can provide insight into performance glitches.
- **Search for a customer's VFIs to monitor their health** -- Using search, you can locate the VFIs or VPLS Sites used by a specific customer. After locating these device models, you can check their current condition or performance details to help ensure they meet your established SLA.
- **View real-time performance details** -- For any VFI or VPLS Site model, you can view an on-demand performance report.
- **Run reports** -- VPLS Manager provides various reports to help you monitor the health of your VPLS environment, such as event or availability reports.

Some of these monitoring tasks are proactive (such as searching for specific VFIs) and some are the result of an alarm triggered by a device in your VPLS environment.

Although SLA information is not maintained in VPLS Manager, you can use the information you have about your SLAs, such as the name of a VFI used by a client, to determine which customers are impacted by an alarm involving a VPLS device.

How the Condition of VPLS Devices is Calculated

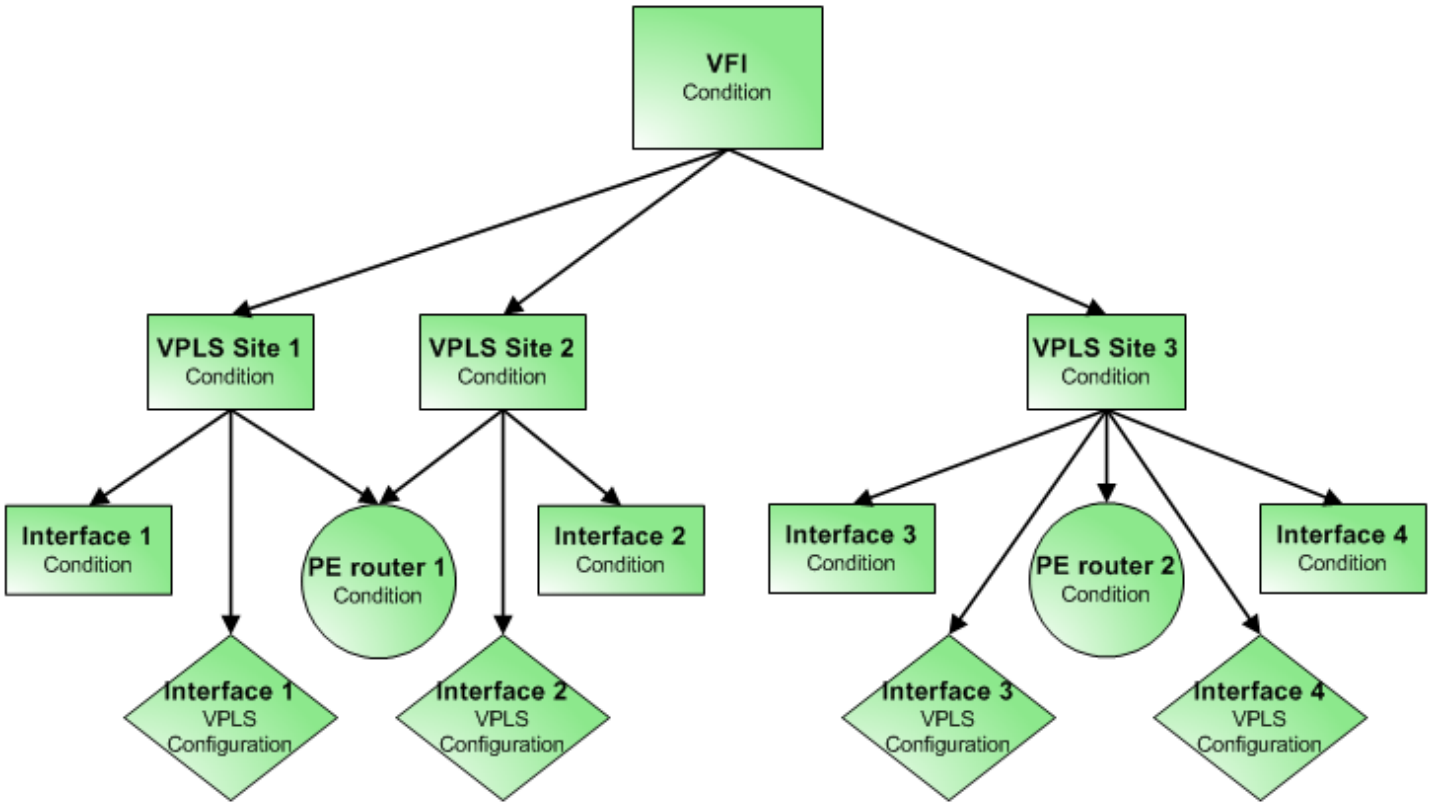
The overall condition of a VFI is of critical importance to providers of VPLS services. Knowing the condition of a VFI can help ensure that you meet your customer SLAs or determine when performance tuning in your VPLS environment is needed.

DX NetOps Spectrum constantly monitors the status of the resources used to provide the VPLS service. The goal of VPLS Manager is to determine the health of the VPLS service by understanding its relationship to physical and logical network entities. VPLS Manager calculates VFI condition based on analysis of the VFI's aggregate VPLS Site conditions.

The process for calculating VFI condition is as follows:

1. The PE routers, interfaces, and interface VPLS configurations are polled at a user-defined interval.
2. When one of these monitored resources experiences an outage, this condition is reflected in the health of the VPLS Site model. Specifically, VPLS Site condition is computed, based on these factors:
   - Contact status of the PE router
   - Condition of the VPLS-enabled interface (IPLS)
   - Configuration of the VPLS-enabled interface
   - Value of mplsVpnVrfOperStatus
   - Value of ifOperStatus for the physical interface
   - Receipt of a jnxVpnIfUp, jnxVpnPwUp, jnxVpnIfDown, or jnxVpnPwDown trap
3. The status of all VPLS Sites is then rolled up into the health of the VFI.

The following diagram demonstrates how condition is calculated for VPLS Sites and rolled up to the VFI models:



For example, DX NetOps Spectrum calculates the condition of VPLS Site 1 as a combination of the Interface 1 condition, PE router 1 condition, and the VPLS configuration of Interface 1. For VPLS Site 3, the condition is affected by two interfaces Interface 3 and Interface 4. Therefore, the condition of VPLS Site 3 is a combination of the PE router 2 condition and the condition and VPLS configuration of the two interfaces.

The condition of the VFI model is determined by the condition of all of the L2VpnSites contained in the VFI. The following list explains how each condition of a VFI is determined:

- **Initial**
  No L2VpnSites are modeled or all VPLS Site models are “Initial.” The percentage of VPLS Sites that are “Initial” is greater than the Minor threshold.
- **Maintenance**
  All the L2VpnSite models are in maintenance mode.
- **Minor**
  The percentage of L2VpnSites down (Site Rollup condition) is greater than the Minor threshold and less than the Major threshold.
- **Major**
  The percentage of L2VpnSites down is greater than the Major threshold and less than the Critical threshold.
- **Critical**
  The percentage of L2VpnSites down is greater than the Critical threshold.
- **Good**
  The percentage of L2VpnSites down is less than the Minor threshold.

You can set threshold values for each VFI model that control whether the VPLS Manager model generates VFI condition alarms.

View Real-Time Performance Data

To measure the performance of your VPLS environment, you must analyze the devices and interfaces used by the VFIs. Their status helps to determine the overall health of your VPLS environment and to decide if you must make changes to improve the performance. Real-time performance monitoring is accomplished by on-demand polling of VPLS-enabled interfaces. The following interface statistics are polled:

- Bytes In Rate
- Bytes Out Rate

To view real-time performance data for a VPLS device

1. Search for the VPLS device you want to monitor.
   The VPLS device model appears in your search results in the Contents panel.
2. Select the VPLS device model.
   Details about the selected model appear in the Component Details panel.
3. Click the Performance tab in the Component Details panel.
4. Select one of the following on-demand reports from the drop-down field:
   – Total Traffic
   – Component Traffic In
   – Component Traffic Out

   NOTE
   Performance data for a selected VFI is an aggregation of the performance data of the component VPLS Sites in a VFI.

   Performance data for the selected report is displayed in the real-time graph. Although DX NetOps Spectrum does not store this data, the graph is updated every 10 seconds to reflect the current performance information.

Reports

Reports provide information that helps you and your customers make informed decisions about your IT assets. Viewing a report lets you analyze specific areas of your VPLS environment to determine if changes are necessary to improve performance.

VPLS Manager uses Spectrum Report Manager to generate reports. Using Spectrum Report Manager you can create the following types of reports:

- Asset reports
- Availability reports

NOTE
For more information about report options and procedures to generate these reports using Spectrum Report Manager, see Report Manager.

Asset Reports

Asset reports generate information about the inventory of assets in the IT infrastructure, including information about VPLS environment assets. Asset reports can help you determine the health of your VPLS environment by focusing on the status of a model's condition over time.

The asset reports available for VPLS Manager include the following:

- **Site Health History** -- A tabular report showing the status of a single VPLS Site during a specified time period
- **Site Health Percentage** -- A pie chart for a single VPLS Site showing all statuses reported and the percentage of time spent in each status during a specified time period
- **VFI Health History** -- A tabular report showing the condition of a single VFI during a specified time period
- **VFI Health Summary** -- A pie chart for a single VFI showing all conditions reported and the percentage of time spent in each condition during a specified time period

Availability Reports

Availability reports provide historical information about up time and down time for assets in the IT infrastructure. Availability reports can help you determine the health of your VPLS environment by highlighting which VFIs and VPLS Sites performed the best and worst over time.

> **WARNING**
> The availability reports cannot function properly when duplicate site or VFI names are modeled in DX NetOps Spectrum. To help ensure these reports work correctly, be sure you provide unique names for all site and VFI models.

The availability reports available for VPLS Manager include the following:

- **Top N Most Available Sites** -- A tabular report of best performing VPLS Sites in respect to up-time accumulated during a specified time period
- **Top N Least Available Sites** -- A tabular report of the worst performing VPLS Sites in respect to up-time accumulated during a specified time period
- **Top N Most Available VFIs** -- A tabular report of the best performing VFIs in respect to up-time accumulated during a specified time period
- **Top N Least Available VFIs** -- A tabular report of the worst performing VFIs in respect to up-time accumulated during a specified time period

Watches

This section contains information about how to work with Watches.

Working With Watches

A *watch* is a mechanism for adding thresholds for model attributes. Watches let you monitor network elements, such as routers, with a high level of detail. They also provide current data that can be used with other DX NetOps Spectrum tools in network analysis.

Watch administration comprises two main components: an intelligent circuit in the SpectroSERVER that performs monitoring functions, and a OneClick administration interface. Watch administration is available under the Thresholds And Watches view on the Information tab for a model.

You can dynamically apply watches on any type of attribute and can monitor attributes against thresholds to generate events and alarms. You can also copy the properties of a watch to another watch. You can thus retain the first watch information while adding new information to the second watch.

You can set watches on the internal and external attributes of any model type. You can also set multiple watches for an attribute. For example, you can set two packet rate threshold watches on a device. One threshold generates a yellow alarm when the value exceeds 10,000, and another threshold generates a red alarm for values above 15,000.

NOTE
We recommend familiarizing yourself with DX NetOps Spectrum administration before setting up watches. For more information, see Database Management, OneClick Administrator , and Performance Administration. Familiarity with the OneClick interface is also helpful.

The watch feature lets you perform the following tasks:

- Dynamically apply watches on any type of attribute.
- Monitor attributes against thresholds and generate events and alarms.
- Execute scripts when watches are violated or when they are reset.
- Generate reports about watches.

NOTE
Watches that you create in one release of DX NetOps Spectrum can be migrated to later releases.

Thresholds And Watches Subview

You can create, configure, and administer watches in OneClick. View and configure watches from a table in the Thresholds And Watches subview.

NOTE
You can access the Thresholds and Watches subview from the Information tab for a model.

The Watches table displays information for each watch defined on that model. The Watch Status column displays the watch condition with color codes as follows:

- **Gray**
  Indicates that the watch is inactive. The watch is not currently running because it has not been activated.
- **Blue**
  Indicates the initial state of the watch. The watch is activated but has yet to run for the first time.
- **Green**
  Indicates that the watch is active and running without any violation.
- **Yellow**
  Indicates that the watch threshold is violated.
- **Red**
  Indicates that the watch failed to evaluate. The text explains the reason.

The toolbar buttons let you do the following:

- Activate
- Deactivate
- Create
- Edit
- Copy
- Delete
- Display watch information
- Print watch information
- Export the Watches table

Create and Edit a Watch

You can create watches to monitor selected attributes of network models.

Follow these steps:

1. Click the Information tab for a model, and expand the Thresholds And Watches subview.
2. Expand the Watches subview.
   The Watches table appears.
3. On the toolbar in the Watches subview, click Create a new watch 
   The Create Watch dialog opens.

   NOTE
   The Expression tab is selected by default.
4. Enter a name for the new watch and select a data type from the Data Type list.

   NOTE
   This data type is the watch destination attribute. The watch expression must evaluate to this data type.
5. Build the expression for the watch by combining operators, functions, and model type attributes:
   - Click a button to the left of the expression area to insert an operator or function into the watch expression at the cursor.
   - Click the Attributes button and select the required attribute to insert an attribute.

     NOTE
     Set the PollingStatus attribute to True to activate a watch.
6. Click the Properties tab and specify the following settings:
   - **Active By Default**
     Specifies whether the watch is active by default for all models that inherit the watch. If this option is not set, activate the watch manually for the desired models.

     WARNING
     Setting Active By Default for a polled watch can adversely affect SpectroSERVER performance.
   - **Evaluate On Demand**
     Evaluates the watch expression only when the watch attribute is read.
   - **Evaluate On Change**
     Evaluates the watch expression if any attribute in the watch expression changes. The attributes must have either the Memory or Database flag set.
   - **Evaluate By Polling**
     Evaluates the watch after each poll interval.
   - **Poll Interval**
     Specifies the polling frequency in seconds. This field is enabled only if you select Evaluate By Polling.

     NOTE
     A watch does not become active if the poll interval is set to 0.
   - **Create watch on model type *model type***
     Specifies the model type where the watch is created. Click Browse to select a different parent model type.

     NOTE
     The watch is created on the model type of the selected model by default. However, you can select a different parent model type for the watch to let other derived model types inherit the watch.
   - **Make Inheritable**
     Makes the watch available for models of all model types that are derived from the selected model type.
7. Click the Threshold tab and select the Attach a Threshold check box if you want to attach a threshold value to the watch.
   The Comparison, Notification, and Script options are enabled.

8. Specify the following Comparison options. The Comparison options specify the threshold settings that specify the boundaries or limits to regulate watch notifications:
   - **Threshold violated if value**
     Specifies the operator for threshold violations. For example, select "greater than". The threshold is considered violated if the sample returns a value that exceeds the value in the threshold attribute. For the threshold violation value, specify a constant or click Attributes to select an attribute.
   - **Threshold reset if value**
     Determines when a threshold status of Violated is reset to Normal. For each subsequent sample after a threshold violation, this reset value is compared to the watch expression using the comparison primitive in the opposite direction. That is, if the comparison primitive is >= (greater than or equal to), the status is reset to Normal as soon as a sample returns a value that falls below this reset value. The reset value is not applicable when the Threshold Comparison is set to == or != because the watch is considered violated if the value is either greater than or less than the watch expression, irrespective of the direction. In such a situation, this option is disabled; otherwise, this entry is mandatory. For the threshold reset value, specify a constant or click the Attributes button to select an attribute.
9. Specify the following Notification options. The Notification options let you specify the type of notification that is sent when the threshold is violated.
   - **No Notification**
     Prevents notification generation when a threshold is violated.
   - **Generate Event(s)**
     Specifies the events to be generated by the DX NetOps Spectrum event management system when the watch is violated or reset. Selecting this option enables fields for event codes as follows:
     - when threshold is violated -- Specifies the event that is generated when the threshold is violated.
     - when threshold is reset -- Specifies the event that is generated when the threshold is reset.
     - when deactivated/all instances are reset -- Specifies the event that is generated when all instances of this watch are reset or when a violated watch is deactivated.

     You can further configure these events to generate or clear alarms and to participate in Event Rules and Procedures. Events are not required for all fields; events are only generated for those fields that contain valid event codes.
   - **Generate Alarm**
     Specifies an alarm to be generated directly if the watch threshold is violated. A generic event (0x480004) is also generated and associated with the alarm. If the watch resets and the alarm is cleared, a generic reset event (0x480005) is generated. Selecting this option enables the following fields:
     - Severity -- Specifies the alarm severity as Minor, Major, or Critical.
     - Description -- Specifies the alarm cause description. A generic description is provided by default. The Browse button lets you select a different cause description or create a new one. Newly created descriptions are stored on the OneClick server in the directory $INSTALL/custom/Events/CsPCause. If you have multiple OneClick servers, copy the files manually.
     - User clearable -- When set, lets you manually clear the watch alarm.
     - Reset watch upon user clearing alarm -- Resets the watch if the alarm is manually cleared even if the watch has not reached its reset value. The alarm can be generated again on a subsequent violation. This option is only available if User Clearable is selected.
10. Specify the following Script options. The Script options let you specify a script to execute if the watch threshold is violated or cleared.
    - **Execute a Script when threshold**
      Enables script execution. Select the threshold condition that triggers script execution.
    - **Execute for each instance**
      If enabled and the watch expression contains list attributes, the script is executed for each instance that meets the threshold condition.
    - **Script File**

Specifies the script file on the SpectroSERVER to execute. If no directory path is specified, the default directory, <*$SPECROOT*>/SS-Tools/SwScript, is used. You can change the default directory by setting sw_script_path in the file <*$SPECROOT*>/SS/.watchrc.

NOTE
Scripts are executed by the user who starts the SpectroSERVER. Therefore, that user requires the privileges to access and execute the scripts. If a permission problem is detected during watch creation or modification, an error message appears. An event is generated if privileges are changed later and the problem is detected in an attempt to execute the script.

11. Click the Landscapes button to specify the destination landscapes for the watch. In the OneClick environment, you can distribute watches to multiple landscapes in a distributed server environment. New watches are created in all landscapes by default.
    A dialog opens with a list of landscapes.
12. Move the landscapes in which you do not want to create the watch to the right.

    NOTE
    If a landscape is not available when the watch is created, you can add the watch to that landscape later. Use the Edit Watch dialog and click Landscapes.
13. Click OK.
    The watch is created.

Create and Edit an Alarm Cause Description

You can create and edit cause descriptions for alarms that are generated for threshold violations. A notification of a threshold violation is delivered through the DX NetOps Spectrum event and alarm facilities. These features support notification delivery by telephone, pager, or by email using an external script interface.

Follow these steps:

1. Select a watch and
   click edit 
   The watch opens in the Edit Watch dialog.
2. Select the Threshold tab, and select Generate Alarm.
   The alarm options are displayed.
3. Click Browse.
   The Select Alarm Cause Code dialog opens. The alarm details appear at the bottom.
4. To create a new alarm cause,
   click create 
   Or, to edit it, select one from the list and click
   edit 

   A dialog opens to let you create or edit the alarm cause.

   NOTE
   The Cause Code field is read-only.
5. Enter information in the fields as required, and click OK.
   The alarm cause description is created or modified.

Copy an Alarm

Use the Copy feature to create multiple alarm cause descriptions that differ only minimally from each other.

Follow these steps:

1. Select a watch in the Watches table and
   click Edit 
   The watch opens in the Edit Watch dialog.
2. Select Generate Alarm on the Threshold tab and click Browse.
   The Select Alarm Cause Code dialog opens.
3. Select an alarm cause description from the list and
   click Copy 
   The Copy Alarm Cause dialog opens.
4. Enter the cause code.
5. (Optional) Edit other alarm properties.
6. Click OK.
   The new alarm cause description is created.

Edit a Watch

You can edit a watch to change its expression, properties, and thresholds.

Follow these steps:

1. Click the Information tab for a model, and expand the Thresholds And Watches subview.
2. Expand the Watches subview.
3. Select the watch in the Watches table and
   click Edit 
   The watch opens in the Edit Watch dialog.
4. Change the watch expression, properties, and thresholds as required, and click OK.

   NOTE
   For more information about the watch expression, properties, and thresholds, see the Create and Edit a Watch section.

   The watch is edited.

Copy a Watch

Use the Copy feature to create multiple watches that differ only minimally from each other.

Follow these steps:

1. Select a watch in the Watches table, and click Copy
   Watch 
   The watch opens.
2. Enter a name for the new watch, change any other information as needed, and click OK.
   The watch is copied and a new watch is created.

Delete a Watch

You can delete a watch if you no longer require the associated threshold monitoring.

Follow these steps:

1. Select the watch in the Watches table and
   click Delete 
   If the watch exists in multiple landscapes, a dialog opens. The left side displays the landscapes where the watch exists and from which it will be deleted.
2. (Optional) To preserve the watch on a landscape, move that landscape to the right.
3. Click OK.
   The watch is deleted from the landscapes that are listed on the left.

Display Watch Information

You can display watch information to see a report about the watch. You can also print and export the information to a text file or HTML file.

Follow these steps:

1. Select a watch in the Watches table and
   click information 
   The Watch Report dialog displays the watch information.
2. (Optional) Click Print to print the watch information.
3. (Optional) Click Export to save the information to a text or HTML file.
   The watch is printed or exported.

Generate a Report on Multiple Watches

You can generate a report about multiple watches from the OneClick Administration page.

Follow these steps:

1. Open the OneClick home page in a browser, and click the Administration tab.
   The Administration Pages opens.
2. Click the Watch Reports link.
   The Generate Watch Report dialog opens. The Select Landscape list and the Select Model Type(s) list are displayed.
3. Select the landscape and model types for which you want to generate a report, and click Generate Report.
   The report is generated.

Inheritable Watch Edits

When you edit an inheritable watch, the watch definition on the selected model type is modified by default. You can select a different parent model type for the watch. Use the Browse button in the Model Type section of the Properties tab.

If a watch definition is modified for a model type that was derived from the watch's originating model type, the changes only take effect for models of that type and its derived types. The model type where the watch originated is not changed.

For example, a watch is originally created on Gen_IF_Port and is inheritable. Serial_IF_Port is derived from Gen_IF_Port, so Serial_IF_Port models inherit the watch. If the watch definition is modified on the Gen_IF_Port model type, the changes are propagated to Serial_IF_Port models. However, if the watch definition is modified on Serial_IF_Port, the new version overrides the definition on Gen_IF_Port. Only Serial_IF_Port models are affected, while Gen_IF_Port models retain the original watch definition. As a result of the inheritance rules, you can redefine or override inheritable watches differently for derived model types.

Manage and Configure Events

Events

To illustrate DX NetOps Spectrum advanced event processing, consider the following scenario. For testing purposes, you want to temporarily allow a high child count. You want to generate an alarm only if the high child count condition persists for more than one minute, and to clear any such alarms when the watch is reset. Using an EventPair Event Rule, DX NetOps Spectrum can perform the following tasks:

- Generate an initial event when the ChildLimit watch is violated.
- Suppress the alarm unless that initial event is *not* followed by a reset event within 60 seconds.
- Generate another event when all instances of this watch are either reset or deactivated.

> **NOTE**
> For more information about event rules and creating events, see Event Configuration.

Configure Events

The following procedure is an example of event configuration. Building from the example in the previous topic, the first event is "Initial child count too high" (0xffffffff), and the second event is "Reset" (0xfffffffe). If the child count threshold is violated and is not reset within 60 seconds, a "Child count too high" alarm (event/alarm 0xfffffffd) is generated.

Follow these steps:

1. Select Utilities, Event Configuration from the Tools menu.
   The Event Configuration page appears with the Navigation, Contents, and Details panels.
2. Click
   
   (Create Event) in the Navigation panel.
   The Create Event dialog appears, with Event Code populated with a default value.
3. (Optional) Edit the event code and enter an event message and click OK.
   The event is created and its details appear in the Contents and Details panels on the right.
4. Create and save the initial Child count too high event (0xffffffff).
   This event triggers the EventPair rule that conditionally generates an alarm and a Reset event (0xfffffffe). The watch uses the Reset event to evaluate the persistence of the child count condition and to clear any alarms that the EventPair rule generates when the watch is reset.
5. Create and save an event "0xfffffffd" and a corresponding alarm that is used to generate a notification if the child count is persistently high.
6. Create and save an Event Pair rule for the initial "Child count too high" event (0xffffffff) that generates the "0xfffffffd" event and alarm if the Reset event (0xfffffffe) is not received within 60 seconds.

   > **WARNING**
   > If the SpectroSERVER is updated while an event is being processed using an Event Rule, the processing is not completed. All event rules that are processing when this functionality is invoked are flushed.
7. Click File, Save All.
   The new events are saved.
8. Open a watch for editing.
9. On the Threshold tab, select Generate Event(s) under Notification.
   The event code fields appear.
10. Enter the initial "Child count too high" event (0xffffffff) that you previously created in the 'when threshold is violated' field.
11. Enter the alarm-clearing "Reset" event (0xfffffffe) that you previously created in the 'when threshold is reset' field, and click OK.
    The event is associated with the watch.

If the LAN_802_3 container contains 5 or more models, the initial "Child count too high" event is generated, and the event rule is evaluated. An alarm is generated only if the reset event is not received within 60 seconds of the initial event. If the watch is reset, any "Child count too high" alarms that the watch generates are cleared.

The EventDisp entries (if needed) for this example are as follows:

```
# make sure child count too high (0xffffffff) is
# reset (0xfffffffe) within 60 seconds
# otherwise generate alarming event (0xfffffffd)0xffffffff E 50 R Aprisma.EventPair, 0xfffffffe, 0xfffffffd,
 60
# generate an alarm on persistent child count too high0xfffffffd E 50 A 1,0xfffffffd
# if child count drops below threshold, clear any existing alarms0xfffffffe E 50 C 0xfffffffd
```

Event Codes

Event codes are generated in the DX NetOps Spectrum Event Log for each event.

NOTE
An event is not generated for watch creation.

The following list contains event codes and their descriptions:

- **0x0048000a**
  Watch could not be activated for a model
- **0x00480000**
  Modification of a watch
- **0x00480001**
  Destruction of a watch
- **0x00480002**
  Activation of a watch
- **0x00480003**
  Deactivation of a watch
- **0x00480004**
  Violation of a watch threshold
- **0x00480005**
  Resetting of threshold status for a watch (with variable reset value)
- **0x00480006**
  Violation of an instance of a watch
- **0x00480007**
  Resetting of threshold status for an instance of a watch (with variable reset value)
- **0x00480010**
  Bad probable cause message
- **0x00480012**
  Logging interval conflict
- **0x00480013**
  Error executing a script (for threshold watches only)
- **0x00480014**
  Resetting a watch threshold for a model, with reason

Event Format Files

You can use the existing event format files Event00480004 and Event00480005 (located in <*$SPECROOT*>/SG-Support/CsEvFormat) as templates to create your own watch-related event format files.

NOTE
For more information, see Event Configuration.

Event Variables

The events for watch threshold violation (0x00480004) and reset (0x00480005) have variable bindings. This applies both to watches that generate alarms directly and watches that generate events. The clear event has a single variable binding, ID #2, containing the watch name.

The following table shows the binding variables and their contents:

Variable Binding ID	Contents
2	Watch name
4	Threshold reference value
5	Threshold reset value (only for reset events)
6	Comparison string value
7	Calculated watch value
8	Instance object identifier (OID) if the watch has instances, otherwise empty
9	Internal; strings to complete the respective event message
10	Internal; strings to complete the respective event message
11	Internal; strings to complete the respective event message

Watch Expressions

A watch expression defines what is monitored by the watch. The watch expression can be as simple as the name of an attribute or as complex as a formula. A watch expression includes attributes and constant values that are related to each other through mathematical symbols and Boolean operators, which are referred to as primitives.

Expressions are evaluated proceeding from left to right in the following order of precedence for primitives:

Precedence	Primitives
1	.# .
2	( )
3	DELTA ( ) COUNTER_DELTA ( )
4	!
5	| &
6	* /
7	+ -
8	= = != >= <= < >

Primitives

Primitives are typically used in expressions that have a left side and a right side. For example, in A+B, + is the primitive defining the relationship between the left side, A, and the right side, B. Parentheses are used to indicate the order of evaluation for expressions with multiple components.

The available primitives are as follows:

- +
- -
- *
- /
- ==
- !=
- >
- <
- >=
- <=
- ! (not)
- & (and)
- ,
- |
- (
-)
- TRUE
- FALSE
- TIME
- DELTA
- MIN
- MAX
- INTEGER
- REAL
- UNSIGNED
- COUNTER_DELTA
- UNSIGNED64

You can group primitives by function and applicability. You can apply the following primitives only between text strings or in numerical expressions:

- = = equal to
- >= greater than or equal to
- <= less than or equal to
- != not equal to
- > greater than
- < less than

NOTE
These primitives do not support the attribute types Object ID, IP Addr, and Boolean.

You can apply the following primitives only for numerical expressions:

- + addition
- - subtraction
- * multiplication
- / division

These primitives do not support the attribute types Text String, Object ID, IP Addr, and Boolean. Supported types are evaluated in the following order:

- Real
- Time Ticks, Date, Gauge, Counter
- Enumeration Integer

If elements are from two different levels, they are evaluated as belonging to the higher level. For example, 5 + 5 is evaluated as an integer, while 5 + 5.1 is evaluated as a real.

> **NOTE**
> DX NetOps Spectrum model types use the attribute type Counter to store unsigned long integers. If you use the unsigned primitive illegally in an expression, for example, "Text String + UNSIGNED(5)", the resulting error message is of type Counter.

You can apply the following primitives only to Boolean expressions:

- ! logical not
- & and
- | or

The following primitives are composed of words instead of symbols:

- **TIME**
  Represents the current time as the number of seconds since January 1, 1970, 00:00 Greenwich Mean Time.
- **DELTA**
  Calculates the change of an attribute over the intervals of the sampling frequency. For example, a watch on an integer attribute named int1 at 30-second intervals might produce DELTA values as follows:

Interval	Values
0	100, 0
30	1000 900
60	1000 0
90	-1000 -2000
120	2000 3000

The DELTA primitive supports the following attribute types:

- Integer
- Enumeration
- Real
- Short
- Time Ticks
- Date
- Gauge
- Counter

In cases where the value is never expected to go down (such as with Time Ticks and Counter), use COUNTER_DELTA.

You can instance attributes in DELTA expressions as follows:

- DELTA (If_In_Octets.2)
- DELTA (If_Out_Octets.#)
- DELTA (TIME) is also supported

> **NOTE**
> DELTA (TIME) is the difference between now and the last evaluation of the watch; it is useful only for calculations involving internal attributes. To generate a rate calculation involving external attributes, use the

device's time, for example, DELTA (XfIDELTA (Sys_Up_Time), where Sys_Up_Time is the time counter on the device.

- **COUNTER_DELTA**
  Calculates the change of an attribute whose value is assumed to be an increasing unsigned integer. The value may be reset to zero, but the result of this primitive is always a positive unsigned integer. Negative attribute values are treated as large unsigned values.
  The COUNTER_DELTA primitive supports the same attribute types as DELTA, and you can also instance attributes within COUNTER_DELTA expressions as described for DELTA. COUNTER_DELTA (TIME) is also supported.
- **ATTR**
  Indicates that the hexadecimal number in parentheses following the primitive is an attribute identifier. This primitive lets an attribute to be uniquely identified when duplicate names exist as shown in the following examples:
  ATTR (<*attr_id*>)
  ATTR (<*attr_id*>).<*instance id*>
  ATTR (<*attr_id*>).#
  The <*attr_id*> value is “0x” or “0X” followed by between one and eight hexadecimal numerals.
- **TRUE**
  Defines the Boolean constant True.
- **FALSE**
  Defines the Boolean constant False.
- **MAX**
  Identifies the larger of two expressions separated by a comma and enclosed by parentheses, for example, (attribute x +1, attribute y-5). The expressions may consist of any combination of operands, attributes, or primitives that results in a numeric value (non-text, string, octet, bool).
- **MIN**
  Identifies the smaller of two expressions separated by a comma within parentheses, for example, (attribute x+1, attribute y-5). The expressions may consist of a combination of operands, attributes, or primitives that results in a numeric value (non-text, string, octet, bool).

You can use the following primitives (not shown on the pop-up selector menu) to indicate the instances of a list attribute that are monitored by a watch.

- **.**
  Indicates that the following entry identifies a specific instance of a list attribute.
  For example, If_In_Octets.2 specifies the second instance of the attribute If_In_Octets. In the case of an IP address table, the identifier might be an entire address instead of a single digit.
  A watch fails if the specified instance of an attribute does not exist even for one of the models for which the watch is activated. In cases where the instance exists for some models but not for others, the watch succeeds for those models where it is present and fails for others.
- **.#**
  Indicates that the current Instance Specifier value shown in the Watch Detail View determines which instances are monitored.
  For example, if the Instance Specifier value is ALL, the expression If_In_Octets.# applies the watch to all instances of the If_In_Octets attribute. However, if the Instance Specifier value is RANGE (1-3), only instances 1, 2, and 3 are monitored.
  In cases where a range of instances is specified, the watch fails even if one of the instances in the range is not present for one of the models. Conversely, the watch is successful for any model where all of the instances in the range are present.

Data Types

The following table shows the data type values that you can assign to the various attribute types.

Note: Although DX NetOps Spectrum does not support watches of attributes with the data type OCTET STRING, you can watch such attributes using the TEXT STRING data type.

Expression Result Type	Acceptable Destination Attribute Types
BOOLEAN	Any Numeric Type
TEXT STRING	Text String
INTEGER	Integer, Enumeration, Real
ENUMERATION	Integer, Enumeration, Real
REAL	Integer, Enumeration, Real
DATE	Real, Date, Time Ticks, Counter, Gauge, Counter64
TIME TICKS	Real, Date, Time Ticks, Counter, Gauge, Counter64
COUNTER	Real, Date, Time Ticks, Counter, Gauge, Counter64
GAUGE	Real, Date, Time Ticks, Counter, Gauge, Counter64
OBJECT Id	OBJECT Id
IP ADDRESS	IP ADDR
COUNTER64	COUNTER64, Real

For example, given an integer attribute named Int1, the expression Int1 = 50.5 (requires a cast) is allowed, while Int1 = “a string” is not allowed.

The rules are most flexible for assigning values to Boolean and Text String attributes. Any expression can be evaluated to a 0 and 1 and can be written to a Boolean attribute. For example, given Bool1 as a Boolean attribute, the expression Bool1 = 500 * 50 + 450 assigns TRUE to Bool1. The expression Bool1 = (500 * 50 + 450) * 0 assigns FALSE to Bool1. Similarly, any expression can be evaluated to a text string. Given a Text String attribute named str1, the expression str1 = 500 + 50 assigns the string “550” to str1.

> **NOTE**
> The Text String attribute type accepts text strings in quotation marks, and arithmetic expressions that are not contained in quotation marks, such as 500 + 50, which results in a string of “550.”

Constants

Like attribute values and operators, you can enter constants directly into the expression field. You can enter the following types of constants in expression formulas:

- UnsignedInteger, represented by a sequence of one or more digits with a positive value.
- Real, represented by a sequence of zero or more digits followed by a dot, followed by a sequence of one or more digits. For example, .7 or 1.7 or 23.24 (but not 7).
- Signed Integer, represented by a sequence of one or more digits with either a positive or negative value.
- Text String, represented by any sequence of characters that are enclosed by double quotation marks. For example, "a string" or "5.25" or " ".

Casting Operators

Casting forces one data type into another. When you perform casting, you risk losing some portion of the value being cast. Casting is unnecessary when the range of the source data type fits into the range of the destination data type.

For example, you can assign an integer to a real number without casting because -1.79769 e+308 **<=** -2,147,483,648 and 2,147,483,647 **<=** 1.79769 e+308. However, you must use casting when assigning an integer to a counter because the ranges do not overlap.

Casting works correctly for the overlapping ranges of the involved data types. For example, casting the integer 5 to a counter behaves as expected. But negative numbers do not fit inside of the range of Counter. As a result, casting -5 to a Counter produces the unsigned (positive) result of 4,294,967,291 because of the way computers represent numbers. Conversely, casting a counter value larger than 2,147,483,647 into an integer produces a negative number.

You can use the following operators to cast the result of an expression in parentheses to a selected data type.

Operator	Result of Casting
UNSIGNED	Normally, 27 is treated as a signed integer; however, “UNSIGNED (27)” is interpreted as the unsigned integer 27. If a negative number appears in the expression following this operator, it is interpreted as unsigned. Example: UNSIGNED (-5) is interpreted as 5. The number, -5 is UNSIGNED with a value of: 4294967291.
INTEGER	Any real number in the expression following this operator is rounded up or down to the nearest integer. Examples: INTEGER (2.4) is interpreted as 2 and INTEGER (2.6) is interpreted as 3.
REAL	Example: REAL (3) is interpreted as 3.0.

Data Type of a Literal Number

A numeric literal in a watch expression is compared against a series of ranges to determine its data type. The Type column in the table below is the actual data type of the numeric literal, which is directly interchangeable with the other types that are listed in that column cell.

The following table shows how the data type of a literal number is determined:

Minimum	Maximum	Type
-2,147,483,648	2,147,483,647	Integer, Boolean*, Enumeration
0	4,294,967,295	Counter, Date**, Gauge, TimeTicks
0	18,446,744,073,709,551,615	Counter64
-1.79769 e+308	1.79769 e+308	Real

* You can also use the constants TRUE and FALSE for Boolean data types. Any non-zero value is equivalent to TRUE.

**The number of seconds from Jan 1, 1970 00:00:00 UTC(0).

Attributes And Instance Identifiers

The attributes that you enter in watch expression formulas must always begin with a letter and consist of a combination of letters, digits and underscore characters (_). Enclose attributes with names identical to primitives (TRUE, FALSE, TIME, DELTA, COUNTER_DELTA) in single quotes. You can also specify attributes by attribute ID using the ATTR primitive.

A watch expression can consist of a single attribute. To create a multi-attribute expression, select an operator from the button palette, then select another attribute and so on.

If a list attribute is specified in the expression, you must include instance information. To specify an instance, append a '.' followed by the instance ID number (for example, **ifInOctets.2**). Or append '.#' (for example, ifInOctets.#), and select either All or Range in the Instance field. For Range, specify the low and high IDs in the From and To fields. List attributes are indicated with "[]" in the Type column in the Attribute Selector dialog.

In addition to the attributes that you enter as part of watch expressions, you can also dynamically create the attributes that are required to store watch information and destination attributes.

Instance Identifiers

You can use instance identifiers to specify a particular object when you are using list attributes in a watch expression formula.

List attributes require instance IDs to fully identify an object. You can specify a specific instance, a range of instances, or all the possible instances of an attribute in a watch expression. When an OID is built, it uses up to two different sub-OIDs: the standard SpectroSERVER OID mechanism or the Watch Instance Specifier. The SpectroSERVER creates an OID from the OID prefix and the OID reference (optional). When you specify an instance for the watch expression on the Expression page, it is appended to the OID as follows:

```
OID = OID prefix + OID reference + Watch instance
```

If you use the Instance Specifier to specify a range, a result for each instance in the range is computed. If the range specified is larger than the actual instances, results are only computed for the actual instances. That is, if you specify a range of 1-10, and there are only 3 instances, only 3 results are computed.

For example, you want to set a watch on If_In_Octets for a router. You supply a range of 1-3 as the Instance Specifier. Because the OID prefix that the DX NetOps Spectrum database uses for If_In_Octets is 1.3.6.1.2.1.2.2.1.10, the following OIDs are assigned:

- OID = 1.3.6.1.2.1.2.2.1.10 + .1
- OID = 1.3.6.1.2.1.2.2.1.10 + .2
- OID = 1.3.6.1.2.1.2.2.1.10 + .3

If you use ALL as the Instance Specifier, all of the instances for the If_In_Octets object are dynamically determined. You can also specify an instance in the watch expression, or you can mix specific instances of an object with RANGE or ALL Instance Specifiers. Using the Instance _of primitive (“.”) to specify an instance of an object overrides the Instance Specifier for that object.

For example, in the following formula, the Instance_of primitive (followed by a value) is used to specify instance number 3 of the If_In_Errors object, and the .# primitive indicates that the current Instance Specifier should be used for the If_In_Octets object.

```
If_In_Errors.3 / If_In_Octets.#
```

Assume that the current Instance Specifier is ALL, and both If_In_Errors and If_In_Octets are list attributes with the OIDs 1.3.6.1.2.1.2.2.1.14 and 1.3.6.1.2.1.2.2.1.10 respectively. The division operation for each instance would be performed, using the following OIDs:

```
If_In_Errors OID = 1.3.6.1.2.1.2.2.1.14 + .3
If_In_Octets OID = 1.3.6.1.2.1.2.2.1.10 + .1
If_In_Errors OID = 1.3.6.1.2.1.2.2.1.14 + .3
If_In_Octets OID = 1.3.6.1.2.1.2.2.1.10 + .2
If_In_Errors OID = 1.3.6.1.2.1.2.2.1.14 + .3
If_In_Octets OID = 1.3.6.1.2.1.2.2.1.10 + .n
```

Notice how the Instance_of primitive overrides the Instance Specifier for the If_In_Errors attribute OID. Instead of each attribute using each possible instance as a result of selecting ALL, the Instance_of primitive allows specification of a particular instance. However, the Instance Specifier still determines the number of instances of a particular watch.

The INSTANCE_ID of the model and the Instance Specifier is applied to each list attribute in the watch expression. If the watch expression contains attributes that use different instances, an error is returned when you attempt to add the watch.

For example, if you enter one formula that contains an attribute that uses board number as an instance and another formula that uses board port as an instance, an error is returned.

Watch Definition Attributes

When you create a watch for a model, an attribute is created to store the watch details. The attribute has a computer-generated name. It is created using the active database Developer ID (registered or default) in the Watch Definition group under the model type to which the model belongs.

Watch Destination Attributes

When you create a watch for a model, a destination attribute is also created with the same name and data type as the watch. This attribute resembles any other model type attribute. When it is read, the watch expression is automatically evaluated. The result of the evaluation is the result of the read operation.

Threshold Reference and Reset Compatibility

For threshold watches, the threshold and reset expressions must be of a type that is compatible with the watch expression.

The following table shows the result types and the corresponding acceptable types:

Result Type	Acceptable Types
BOOLEAN	Boolean
TEXT STRING	Text String
INTEGER	Integer, Enumeration, Real
ENUMERATION	Integer, Enumeration, Real
REAL	Real
DATE	Real, Date, Time Ticks, Counter, Gauge, Counter64
TIME TICKS	Real, Date, Time Ticks, Counter, Gauge, Counter64
COUNTER	Real, Date, Time Ticks, Counter, Gauge, Counter64
GAUGE	Real, Date, Time Ticks, Counter, Gauge, Counter64
OBJECT Id	OBJECT Id
IP ADDRESS	IP_ADDR
COUNTER64	Counter64, Real

Alarm Script and Watch Type Examples

You can refer the alarm script and watch type examples that are available in this section.

Alarm Script Example

The following sample is an example of a UNIX script file that you can create as a notification. You can designate scripts for execution whenever an alarm is set or cleared in conjunction with a threshold watch. This example script is named *sw_alrm_script* and is included in the SS-Tools/SwScript directory when you install DX NetOps Spectrum.

```
#!/bin/sh
############################################################
This is an example script that can used by users to perform important tasks upon threshold violation of a
 watch (Watch Status is #VIOLATED) or when an already violated watch becomes normal (Watch Status #is NORMAL).
#
```

```
# In each of the above cases, the user-specified script is executed (specified during watch creation) and
 provides twenty arguments. The arguments in ascending order are:
#
# 1) DATE   - The date on which the watch was violated.
# 2) TIME        - The time at which the watch was violated.
# 3) MTYPE_NAME     - The model type of the watch.
# 4) MODEL_HANDLE   - The modelhandle of the model.
# 5) MODEL_NAME     - The model of the watch.
# 6) INSTANCE       - The instance (port, board, etc) for which the watch is either violated or reset.
# 7) ALARM_ID      - The ID of the alarm.
# 8) CONDITION     - The CONDITION of the alarm.
# 9) CAUSE_CODE    - The cause code of the alarm.
# 10) REPAIR_PERSON   - The repair person assigned to troubleshoot.
# 11) ALARM_STATUS  - The status of the alarm.
# 12) SCRIPT_TYPE  - This is set to "VIOLATED" if the script is getting executed upon violation and it is set
 to # "NORMAL" if the script is getting executed upon reset.
# 13) WATCH_NAME  -  The name of the watch for which the script is executed.
# 14) WATCH_CREATOR  - The name of the creator of the watch.
# 15) WATCH_SRC  - The formula for the watch expression.
# 16) WATCH_SRC_VAL  - The value of the watch expression formula.
# 17) WATCH_REF  - The formula for threshold reference.
# 18) WATCH_REF_VAL  - The value of the threshold reference.
# 19) WATCH_RES  - The formula for threshold reset.
# 20) WATCH_RES_VAL  - The value of the threshold reset.
################################################################
## Save the argument values into variables that could be used later.
## Remember that DATE is the first argument and WATCH_RES_VAL is the last argument.
# IMPORTANT NOTE:
## Pay particular attention to the variable, "SCRIPT_TYPE". If the watch has a violated status, SCRIPT_TYPE
 is set to "VIOLATED". If a watch has normal status, SCRIPT_TYPE is set to "NORMAL". You can be use this to
 decide what to do when the watch is violated and when it is normal.
while [ "$#" -ne 0 ]
do
case "$#"
 in
 20) DATE=`echo "$1"`;;
 19) TIME=`echo "$1"`;;
 18) MTYPE_NAME=`echo "$1"`;;
 17) MODEL_HANDLE=`echo "$1"`;;
 16) MODEL_NAME=`echo "$1"`;;
 15) INSTANCE=`echo "$1"`;;
 14) ALARM_ID=`echo "$1"`;;
 13) CONDITION=`echo "$1"`;;
 12) CAUSE_CODE=`echo "$1"`;;
 11) REPAIR_PERSON=`echo "$1"`;;
 10) ALARM_STATUS=`echo "$1"`;;
 9) SCRIPT_TYPE=`echo "$1"`;;
 8) WATCH_NAME=`echo "$1"`;;
 7) WATCH_CREATOR=`echo "$1"`;;
 6) WATCH_SRC=`echo "$1"`;;
 5) WATCH_SRC_VAL=`echo "$1"`;;
 4) WATCH_REF=`echo "$1"`;;
 3) WATCH_REF_VAL=`echo "$1"`;;
```

```
 2) WATCH_RES=`echo "$1"`;;
 1) WATCH_RES_VAL=`echo "$1"`;;
  esac
  shift
done
## Compose a message that can be used, for example, to send mail.
## The composed message can be used for any other purpose as well.
message()
{
echo "Watch Status Notification"
 echo ""
 echo "Watch Status is $1"
 echo ""
echo "Date:             $DATE"
echo "Time:             $TIME"
echo "MType Name:  $MTYPE_NAME"
echo "Model Handle:      $MODEL_HANDLE"
echo "Model Name:  $MODEL_NAME"
echo "Instance:        $INSTANCE"
echo "Alarm ID:   $ALARM_ID"
echo "Condition:  $CONDITION"
echo "Cause Code:  $CAUSE_CODE"
echo "Repair Person:     $REPAIR_PERSON"
echo "Alarm Status:      $ALARM_STATUS"
echo "Watch Name:  $WATCH_NAME"
echo "Watch Creator:  $WATCH_CREATOR"
echo "Watch Expression: $WATCH_SRC"
echo "Watch Expression Value: $WATCH_SRC_VAL"
echo "Watch Reference:  $WATCH_REF"
echo "Watch Reference Value: $WATCH_REF_VAL"
echo "Watch Reset:  $WATCH_RES"
echo "Watch Reset Value: $WATCH_RES_VAL"}
## Mail the composed message to the interested users.
## <USER LIST> is the list of mail recipients.
message "$SCRIPT_TYPE" | mail <USER LIST>
```

Watch Type Examples

First Watch Scenario

The following parameters establish a watch on Hub_CSI_IRBM. This watch checks the total number of frames transmitted and collisions received every 60 seconds. It generates a minor alarm with the message "Too many collisions" if the threshold value of one million is exceeded. The Threshold Reset Value indicates that the threshold status is reset from Violated to Normal (and the alarm is cleared) when the total number of collisions falls to 500,000.

You can create watches for this scenario as follows:

1. Create an inactive watch of type Counter and name it HubColls_v. Assign the total number of collisions of 1000000 in the expression.
2. Create an inactive watch of type Counter and name it HubColls_r. Assign the total number of collisions of 500000 in the expression.

3. Create an active watch of type Counter and name it HubColls and assign it values as shown in the following watch examples.

Examples: Watches

The first watch consists of the following parameters:

- Name: HubColls_v
- Data Type: Counter
- Expression:
  - Expression: 1000000
- Instance: None
- Properties
  - Default Activation: Inactive
- Threshold: None

The second watch consists of the following parameters:

- Name: HubColls_r
- Data Type: Counter
- Expression:
  - Expression: 500000
- Instance: None
- Properties
  - Default Activation: Inactive
- Threshold: None

The third watch consists of the following parameters:

- Name: HubColls
- Data Type: Counter
- Expression
  - Expression: Hub_Trans_Coll+Hub_Rec_Colls
- Instance: None
- Properties
  - Default Activation: Active
  - Evaluated: By Polling
  - Poll Interval: 00:01:00
- Threshold
  - Attach a Threshold (checked) Threshold Violated if value: > HubColls_v (create a counter watch named HubColls_v Inactive, expression is 1000000.)
  - Threshold reset if value: HubColls_r (create Counter watch named HubColls_r Inactive, expression is 500000.)
  - The attributes, HubColls_v and HubColls_r must be created before the creation of the HubColls watch.
  - Generate Alarm (checked) Severity: Minor Alarm Desc: CollsExceeded (Create a new alarm named CollsExceeded with the message "Too Many Colls".)

Second Watch Scenario

A network administrator wants to monitor the level of disk utilization on a server, but wants to exclude CD-ROM drives from monitoring. CD-ROM drives typically show 0% utilization if no CD is present, but they show close to 100% utilization if a full CD is in the drive. However, such information is not useful to the administrator.

To check disk utilization, the administrator wants to use RFC2790App, which returns device types as OIDs. Each OID maps to a different device type. Any expression that must check against a storage type in this MIB must use its OID for comparison. (You can compare strings to OIDs).

The following table shows RFC2790App Device Type/OID Mapping.

Device Type	OID
hrStorage	1.3.6.1.2.1.25.2
hrStorageTypes	1.3.6.1.2.1.25.2.1
hrStorageOther	1.3.6.1.2.1.25.2.1.1
hrStorageRam	1.3.6.1.2.1.25.2.1.2
hrStorageVirtualMemory	1.3.6.1.2.1.25.2.1.3
hrStorageFixedDisk	1.3.6.1.2.1.25.2.1.4
hrStorageRemovableDisk	1.3.6.1.2.1.25.2.1.5
hrStorageFloppyDisk	1.3.6.1.2.1.25.2.1.6
hrStorageCompactDisk	1.3.6.1.2.1.25.2.1.7
hrStorageRamDisk	1.3.6.1.2.1.25.2.1.8
hrStorageFlashMemory	1.3.6.1.2.1.25.2.1.9
hrStorageNetworkDisk	1.3.6.1.2.1.25.2.1.10

The administrator must create two new watches on the RFC2790App model. The first watch sets up a Boolean value that is used in the second watch to exclude CD-ROM drives (OID 1.3.6.1.2.1.25.2.1.7) from monitoring. The second watch monitors the disk utilization (expressed as a percentage) of other types of storage devices on the host. In this example, an alarm is sent if utilization exceeds 90%.

Examples: Watches

The first watch consists of the following parameters:

- Name: isNotCDROM
- Data Type: Boolean
- Expression: Expression: hrStorageType.# != 1.3.6.1.2.1.25.2.1.7
- Instance: All
- Properties
  - Default Activation: Active
  - Evaluate: On Demand
- Threshold: None

The second watch consists of the following parameters:

- Name: PctDiskUsed
- Data Type: Real
- Expression
  - Expression: REAL(hrStorageUsed.#)/REAL(hrStorageSize.#) * 100 * REAL(isNotCDROM.#)
  - Instance: All
- Properties
  - Default Activation: Active
  - Evaluate: By Polling
  - Poll Interval: 30 seconds
- Threshold

  - Threshold violated if value >= <*set to the acceptable level of disk utilization*>
- Alarm
  - Alarm Severity: Minor
  - Alarm Description: DiskUtilAlarm
  - Alarm is user clearable
  - Watch is not reset upon user clearing of alarm.
- Script: None

Evaluate-On-Change Watches

The following watches are examples of evaluate-on-change (EoC) watches. These watches determine when a view was edited. As a result, the administrator can run a search that shows exactly when LANs were edited and when new models were added to them. Both attributes are evaluated on change, and neither attribute is likely to change frequently.

Examples: Evaluate-on-Change Watches

The first watch consists of the following parameters:

- Name: Child_Count_Watch
- Data Type: Integer
- Expression
  - Expression: Child_Count
  - Instance: None
- Properties
  - Default Activation: Active
  - Evaluate: By EoC
  - Poll Interval: None
- Threshold: None

The second watch consists of the following parameters:

- Name: Edit_Count
- Data Type: Integer
- Expression
  - Expression: EDIT_COUNT
  - Instance: None
- Properties
  - Default Activation: Active
  - Evaluate: By EoC
- Threshold: None

Polled Threshold Watches

In the following example, a Polled Threshold watch is created to generate a minor alarm each time the number of children in a LAN_802_3 exceeds 5. The watch clears the alarm when the count of children falls below 5.

First, create the attribute watch ChildLimit_v (an arbitrary name) and give it an expression of "5". Creating a watch depicts the creation of this watch in detail. Then create a polled threshold watch named Child_Limit.

Examples: Polled Threshold Watches

The first watch consists of the following parameters:

- Name: ChildLimit_v
- Data Type: Counter
- Expression: 5
- Properties: None
- Threshold: None

The second watch consists of the following parameters:

- Name: ChildLimit
- Data Type: Integer
- Expression
  - Expression: Child_Count
- Instance: None
- Properties
  - Default Activation: Active
  - Evaluated: By Polling
  - Poll Interval: 00:01:00
- Threshold
  - Attach a Threshold (checked)
  - Threshold Violated if value: > ChildLimit_v
  - Threshold reset if value: <= ChildLimit_v
  - Generate Alarm (checked)
  - Severity: Minor
  - Alarm Description: ExceedChildLimit (Create a new alarm named ExceedChildLimit with the message "Only 5 children allowed.")

First On-Demand Watch Scenario

On-Demand watches are evaluated whenever information is requested from them.

The following examples of simple On-Demand watches work with a polled Threshold watch that generates an alarm when a Cisco router (model type: Rtr_CiscoIGS) starts running out of memory. The attribute, freeMem, is less than 1500 and clears when it exceeds 2500.

Examples: Simple On-Demand Watches Working With a Polled Threshold Watch

The first watch consists of the following parameters:

- Name: MemLow
- Data Type: Integer
- Expression
  - Expression: 1500
  - Instance: None
- Properties
  - Default Activation: Inactive
  - Evaluated: On Demand
- Threshold: None

The second watch consists of the following parameters:

- Name: MemHigh
- Data Type: Integer
- Expression

  - Expression: 2500
  - Instance: None
- Properties
  - Default Activation: Inactive
  - Evaluated: On Demand
- Threshold: None

The third watch consists of the following parameters:

- Name: Mem_Good
- Data Type: Integer
- Expression
  - Expression: freeMem
  - Instance: None
- Properties
  - Default Activation: Active
  - Evaluate: By Polling
  - Poll Interval: 1 minute
- Threshold
  - Threshold violated if value: <= MemLow
  - Threshold reset if value: > MemHigh
- Alarm
  - Alarm Severity: Major
  - Alarm Description: Rtr+Memory
- Script: None

Second On-Demand Watch Scenario

On-Demand watches are evaluated whenever information is requested from them.

The following watches are designed to show the network administrator colored alarm conditions as the performance of a Cisco router (model type: Rtr_ Cisco) progressively degrades. A minor alarm has a value of 50 -59; a major alarm has a value of 60 - 69, and the value of a critical alarm exceeds 70.

Examples: On-Demand Watches

The first watch consists of the following parameters:

- Name: True_Ref
- Data Type: Boolean
- Expression
  - Expression: 1
  - Instance: None
- Properties
  - Default Activation: Inactive
  - Evaluate: On Demand
- Threshold: None

The second watch consists of the following parameters:

- Name: Minor_Busy
- Data Type: Boolean
- Expression

  - Expression: ((busyPer >= 50) & (busyPer <60))
  - Instance: None
- Properties
  - Default Activation: Active
  - Evaluate: By Polling
  - Poll Interval: 1 minute
- Threshold
  - Threshold violated if value: > True_Ref
  - Threshold reset if value: <= True_Ref
- Alarm:
  - Alarm Severity: Minor
  - Alarm Description: RtrBusy1
  - Script: None

The third watch consists of the following parameters:

- Name: Major_Busy
- Data Type: Boolean
- Expression
  - Expression: ((busyPer >= 60) & (busyPer < 70))
  - Instance: None
- Properties
  - Default Activation: Active
  - Evaluate: By Polling
  - Poll Interval: 1 minute
- Threshold
  - Threshold: Threshold violated if value: == True_Ref
  - Threshold reset if value: != True_Ref
- Alarm
  - Alarm Severity: Major
  - Alarm Description: RtrBusy 2
  - Script: None

The fourth watch consists of the following parameters:

- Name: Critical_Busy
- Data Type: Boolean
- Expression
  - Expression: (busyPer >= 70)
  - Instance: None
- Properties
  - Default Activation: Active
  - Evaluate: By Polling
  - Poll Interval: 60
- Threshold
  - Threshold violated if value: == True_Ref.
  - Threshold reset if value: ! = True_Ref.
- Alarm:

  - Alarm Severity: Critical
  - Alarm Description: RtrBusy3
  - Script: None

Third On-Demand Watch Scenario

On-Demand watches are evaluated whenever information is requested from them.

For this scenario, a network administrator wants to see an alarm anytime a Frame Relay link fails, causing the dial-backup lines to activate. The dial backup is ISDN Interface Type 21 (the basic ISDN service). The following watches verify whether traffic is received on any ISDN port. If traffic is flowing, a minor alarm is placed on the Cisco router (model type: Rtr_Cisco).

Examples: On-Demand Watches

The first watch consists of the following parameters:

- Name: True_Ref
- Data Type: Boolean
- Expression
  - Expression: 1
  - Instance: None
- Properties
  - Default Activation: Inactive
  - Evaluate: On Demand
- Threshold: None

The second watch consists of the following parameters:

- Name: ISDN_Backup
- Data Type: Boolean
- Expression
  - Expression: ((COUNTER_DELTA (ifInOctets.#) > 0) & (ifType.# == 21))
  - Instance: All
- Properties
  - Default Activation: Active
  - Evaluate: By Polling
  - Poll Interval: 1 minute
- Threshold
  - Threshold violated if value: == True_Ref
  - Threshold reset if value: != True_Ref
- Alarm:
  - Alarm Severity: Minor
  - Alarm Description: Backup_Active
- Script: None

Fourth On-Demand Watch Scenario

On-Demand watches are evaluated whenever information is requested from them.

For this scenario, the watch monitors the ifOutOctets value on ISDN interface models to see whether it increases or is non-zero. This value indicates whether the interface is active. You can modify the polling, logging, alarm severity, and alarm text to suit your requirements. The second watch uses a true or false (1/0) indicator: 1 indicates that the interface is sending ifOutOctets, and 0 indicates that the interface is not sending.

Examples: On-Demand Watches

The first watch consists of the following parameters:

- Name: isISDN
- Data Type: Boolean
- Expression
  - Expression: (X_ifType == 21)
  - Instance: None
- Properties
  - Default Activation: Active
  - Evaluate: On Demand
  - Inheritable: False
- Threshold
  - Threshold: None

The second watch consists of the following parameters:

- Name: ISDN_Up
- Data Type: Integer
- Expression
  - Expression: MAX(0, MIN(1, INTEGER(((COUNTER_DELTA (X_OutOctets) * INTEGER(isISDN)) > 0))))
  - Instance: None
- Properties
  - Default Activation: Active
  - Evaluate: By Polling
  - Poll Interval: 0+00:05:00
  - Inheritable: False
- Threshold:
  - Threshold violated if value == 1
  - Threshold reset if value != 1
- Alarm
  - Alarm Severity: Minor
  - Alarm Description: ErrorTholdAlarm
  - Alarm is user clearable
  - Watch is not reset upon user clearing of alarm.
- Script: None

First Usability and Testing Watch Scenario

The example watches that are described here are used for usability and testing.

For this scenario, you create two watches, one to create an attribute named WatchLoad_v with an expression value of .75, and a second watch to use the CPULoadRate attribute. In the following example, you create an Alarm Description named WatchLoad_Alarm that says, for example, “The CPU load has exceeded 75%.”

Examples: Usability and Testing Watches

The first watch consists of the following parameters:

- Name: WatchLoad_v
- Data Type: Real
- Expression

  - Expression: .75
  - Instance: None
- Properties
  - Default Activation: Inactive
  - Evaluate: On Demand
- Threshold: None

The second watch consists of the following parameters:

- Name: CPU_Load
- Data Type: Real
- Expression
  - Expression: CPULoadRate
  - Instance: None
- Properties
  - Default Activation: Active
  - Evaluate: By Polling
  - Poll Interval: 1 minute
- Threshold
  - Threshold violated if value: > WatchLoad_v
  - Threshold reset if value: < = WatchLoad_v
- Alarm
  - Alarm Severity: Minor
  - Alarm Description: WatchLoad_Alarm
  - Script: None

Second Usability and Testing Watch Scenario

The example watches that are described here are used for usability and testing.

In this example, you create an EoC watch that generates a minor alarm when Composite Condition attribute of a container model exceeds 4. You first create a reference attribute named conditionCheck_ref and use that attribute in a watch named ConditionCheck. Then use the Alarm Description dialog box to create an Alarm Description named “ConditionCheckAlarm” (or another appropriate string).

The reference attribute has the following parameters:

- Name: ConditionCheck_ref
- Data Type: Integer
- Expression
  - Expression: 4
  - Instance: None
- Properties: None
- Threshold: None

Example: Usability and Testing Watch

The watch consists of the following parameters:

- Name: ConditionCheck
- Data Type: Integer
- Expression

  - Expression: Composite_Condition
  - Instance: None
- Properties
  - Default Activation: Active
  - Evaluate: EoC
- Threshold
  - Threshold violated if value: > ConditionCheck_ref
  - Threshold reset if value: <= ConditionCheck_ref
- Alarm:
  - Alarm Severity: Minor
  - Alarm Description: ConditionCheckAlarm
- Script: None

Third Usability and Testing Watch Scenario

The example watches that are described here are used for usability and testing.

In this example, if the load value falls below 10% for longer than 90 minutes, you create an alarm. To configure this watch, change the watch expression to use your desired external attributes, and set the threshold to the number of consecutive polls equal to the desired time duration, divided by the polling interval.

Examples: Usability and Testing Watches

The first watch consists of the following parameters:

- Name: Watch_Load_Under_10_Pct
- Data Type: Integer
- Expression
  - Expression: MAX (0, MIN (1, (<Load under 10% test expression here>)))
  - Instance: None
- Properties
  - Default Activation: Inactive
  - Evaluation: On demand
- Threshold: None

The second watch consists of the following parameters:

- Name: Watch_TimeTicker_LoadUnder10Pct
- Data Type: Integer
- Expression
  - Expression: (Watch_TimeTicker_LoadUnder10Pct + 1) * Watch_Load_Under_10_Pct
  - Instance: None
- Properties
  - Default Activation: Inactive
  - Evaluation: On demand
- Threshold: *The desired number of consecutive polls*

Fourth Usability and Testing Watch Scenario

The example watches that are described here are used for usability and testing.

The first watch monitors the CPU for a sustained referenced usage value (80% in this example).

The second watch triggers an alarm if the CPU usage remains at a certain level (80%) for a sustained period of time. This watch calculates this time period using the threshold value (3), multiplied by the polling interval (5 minutes). Therefore, this watch violates the threshold and triggers an alarm if the CPU usage exceeds 80% for 15 minutes. You can adjust these values to suit your requirements.

Examples: Usability and Testing Watches

The first watch consists of the following parameters:

- Name: CPU_Duration_Over_80
- Data Type: Integer
- Expression
  - Expression: MAX(0, MIN(1, INTEGER((INTEGER(cpqHoCpuUtilMin.#) >= 80))))
  - Instance: All
- Properties
  - Default Activation: Active
  - Evaluate: On Demand
  - Inheritable: False
- Threshold: None

The second watch consists of the following parameters:

- Name: CPU_Time_Duration
- Data Type: Integer
- Expression
  - Expression: ((CPU_Time_Duration.# + 1) * CPU_Duration_Over_80.#)
  - Instance: All
- Properties
  - Default Activation: Active
  - Evaluate: By Polling
  - Poll Interval: 0 + 00:05:00
  - Inheritable:False
- Threshold
  - Threshold violated if value >= 3
  - Threshold reset if value < 3
- Alarm
  - Alarm Severity: Minor
  - Alarm Description: ErrorTholdAlarm
  - Alarm is user clearable
  - Watch is not reset upon user clearing of alarm.
  - Script: None

WLC Manager

Overview

The WLC Manager lets you discover, model, and monitor Wireless LAN controller (WLC) module components, regardless of the device it is present in, allows you to centrally manage and configure a number of access points (APs) in a simplified manner within the network environment. WLC Manager provides AP (Wireless Access Point) Names and Device discovery and modeling, AP connectivity status, SNMP trap handling, monitoring of WLC conditions, and performs calculation of and alarming on AP conditions.

WLC Manager discovers all the Access Points (AP) associated with a particular WLC Controller. If the intermediate L2 and L3 devices are discovered connections are automatically made. DX NetOps Spectrum creates AP models representing each unique Access Point. These AP models can then be polled for their current operational status (Up or Down).

In addition, WLC Manager supports CAPWAP and CISCO LWAPP (Lightweight Access Point Protocol), a protocol that can control multiple WiFi Wireless Access Points at once. This helps reduce the amount of time that is spent on configuring, monitoring, or troubleshooting a large network. The system also allows network administrators to closely analyze the network. This system is installed in a central server that gathers data from RF devices from different brands and settings. The server can command a selected group of devices to apply given settings simultaneously.

This enables you to monitor the status of the paths between the sites in each of the modeled APs. You can configure threshold alarms to alert you when path changes exceed the configured tolerance.

NOTE

- Do not delete a WLC device, while it is still being modeled. We recommend that you wait until the WLC model is activated and the modeling is completed, before you delete the WLC, so that all the AP models created (associated with the WLC you want to delete) are also deleted.
- If you delete a WLC device before all its associated Access Points (AP) model creation is complete, all the sub-modules or models may not be deleted and can be found in the Lost/Found Container.

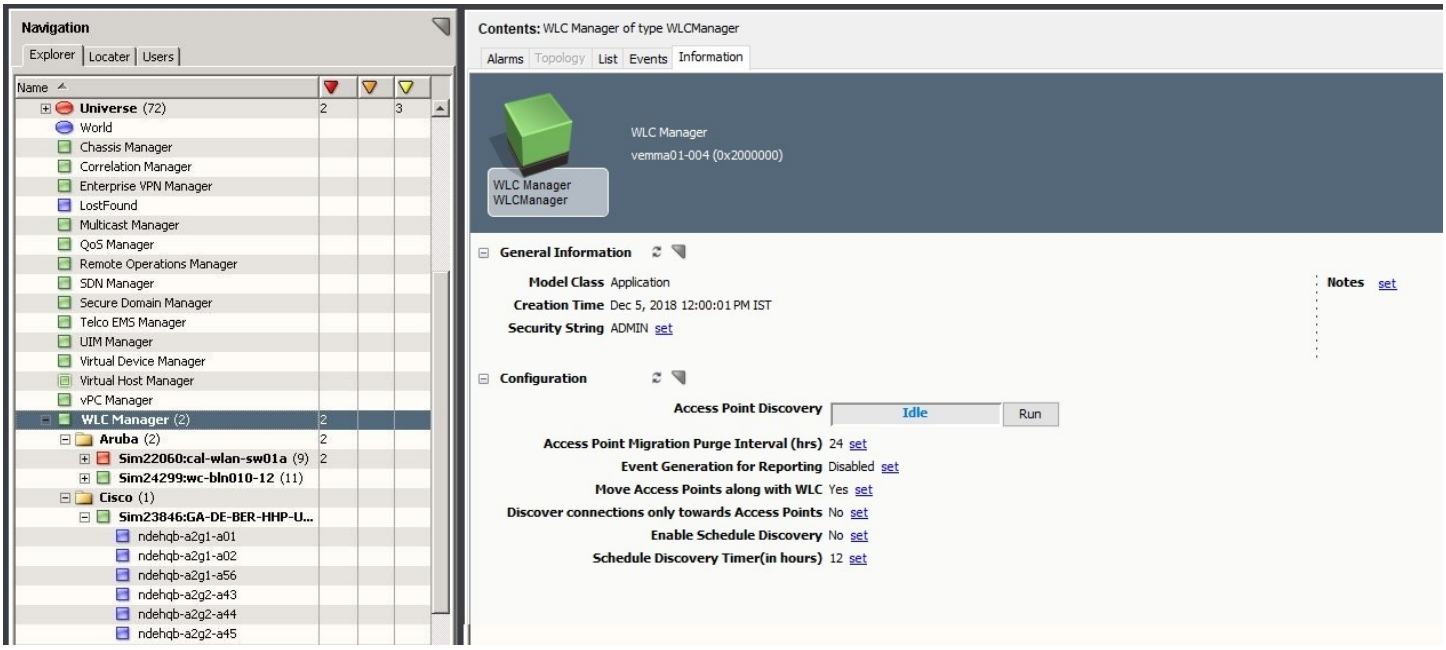
NOTE

- DHCP IP changes are now supported for APs; duplicate models were being created as dynamic IPs were created when WLC/ AP down/ reboot happened.
- Now models are created on unique MAC ID thus negating the possibility of duplicate models.
- AP name changes are now supported and the updated AP name will reflect after the next polling cycle of the WLC.

WARNING

From 10.2, you can no longer manually discover and model APs by model type (AP-Pingable in earlier releases). Navigate to the **WLC Controller > Information Tab > Configuration sub-view and Run AccessPoint discovery.**
If you are upgrading to 10.2, all existing AP-Pingable models will be modeled as AccessPoint models after the upgrade.

To view the WLC Manager in the OneClick Console Topology view, select WLC Manager from the Explorer Hierarchy



WLC Manager provides searches that let you quickly find a particular AP or all APs. Search results contain a list of the current Wireless Access Points that are configured within the environment and their status.
You can drill into a single AP to see the current list of Wireless interfaces participating within that AP and the status of the Access Point(s).

Moving Access Points (APs) with WLC

From 10.2.2, the WLC Manager includes the option to move access points (APs) with the WLC. When the WLC is moved to any other container, all the APs attached to it are moved to that respective WLC container. This feature automates the movement of APs across various containers. By default, this value is set to **Yes.** If you want to retain the APs attached in the WLC container, then set this value to **No**, then access points (APs) remain in the same container, although the WLC is moved to a different container.

Wireless Reports

From r10.2, DX NetOps Spectrum lets you generate reports for events that are raised on Wireless APs, by enabling the Event Generation for Reporting option, at the WLC Manager level.

NOTE

The default value for the **Event Generation for Reporting** field is **Disabled.**

To enable these Wireless reports, from the OneClick console> Explorer Hierarchy, navigate to the **WLC Manager** > **Information Tab** > **Event Generation for Reporting** field and select **Enabled** from the corresponding drop down option.

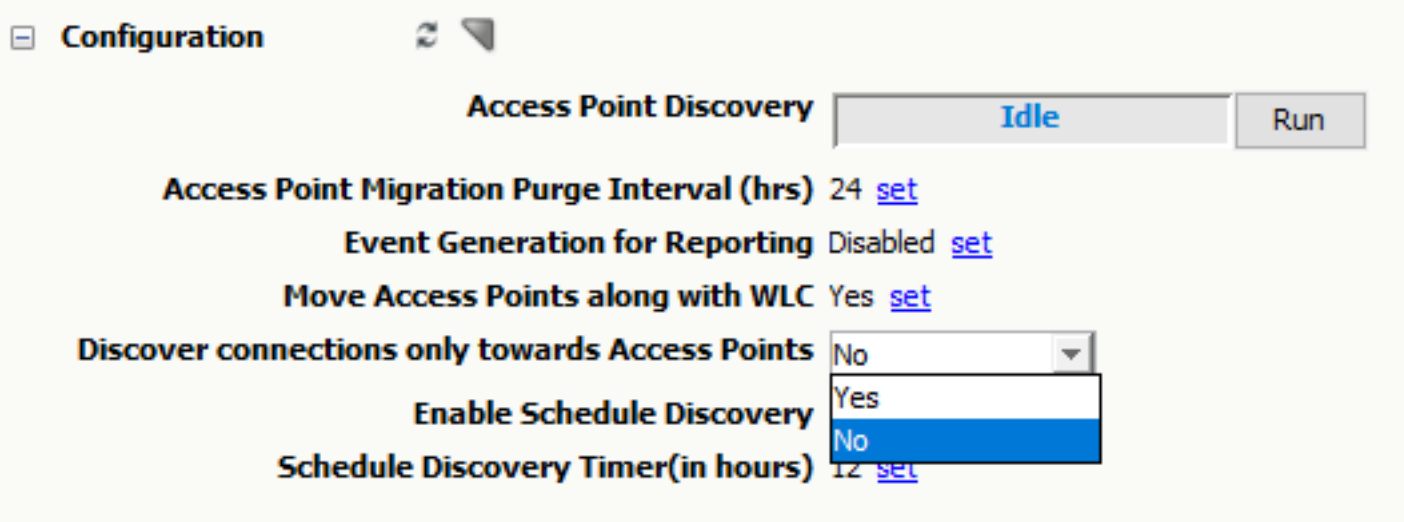
Once you set this to **Enabled,** the events that are raised on the selected WLC Controller, are registered in the SRM DB, parsed and the event information is saved in the MySQL database named 'reporting' which Report Manager uses to store data. This database contains all the tables that are required to store the data that is used by SRM application to generate reports. At startup, the Report Manager retrieves the data from the primary Archive Manager for each SpectroSERVER through OneClick and stores the data in the SRM databases.

NOTE

For more information about the Wireless Reports, refer to the **wirelessaps** table in the reporting Database.

Discover Connections Only toward Access Points

In the 10.3 and earlier releases, when the Access Point Discovery is run, the WLC Manager discover connections on all the subnet switches corresponding to the access points (APs). From 10.3.1, the WLC Manager includes an option to discover connections only toward the access points (APs) from their right upstream switches, which skips creation of other neighboring connections for the subnet switches. This feature uses MAC address (SAT Table) or IP address (CDP Table) of the Access Point to discover a connection toward its upstream switch.



By default, this value is set to No (in releases prior to 10.4.2). If you want to see the connections only toward access points, then set this value to **Yes**.

> **NOTE**
> In 10.4.2, this option is enabled by default; that is, the parameter value is already set to Yes. In previous releases, the option was not enabled by default.

Raise Access Point-Related Alarms Only on Associated Access Points

In 10.4.1, if any applicable alarm is raised on an access point (AP), then the alarm is asserted to that AP. Similarly, if any alarm is raised on a WLC controller, then the alarm is asserted to the WLC controller. This functionality helps in the proper organization of the alarms based on the associated AP or WLC controller, allowing you to manage your devices more efficiently. Previously, all alarms were getting asserted to the WLC controller irrespective of the fact where the alarm had been raised: AP or WLC controller.

> **NOTE**
> If the required AP is not accessible for asserting the alarm, then the related AP alarm is asserted to the associated WLC controller.

Access Point (AP) Attributes

The details in the **Access Point > Information Tab** is populated with relevant values, only when SNMP APs make the association with its WLC. (The values are actually exposed from the WLC MIBs)

> **NOTE**
>
> By default, the information displayed in **Access Point > Information Tab** is not exposed by SNMP APs. This is because, Standalone/Autonomous **Access Points** (i.e APs not associated to any WLC), do not have this information in its MIBs, and this information is available only from associated WLCs.

To view the details / values in the *Access Point > Information Tab*:

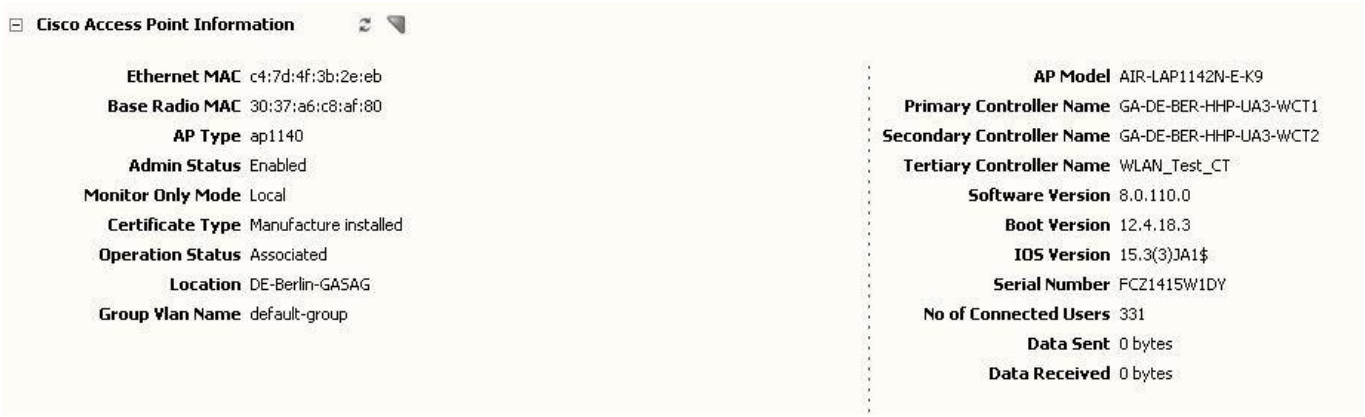
- Model the parent **Wireless Controller** device, by enabling Access Point Discovery, using the **Create Wireless Access Points** option.

NOTE

For the **Access Point Information** sub view to populate, a minimum of one polling cycle is required.

Cisco Access Point Information

The following attributes are displayed when you navigate to **Access Point > Information Tab** > **Access Point Information** sub view for an identified and modeled AP, under the **WLC Manager** nodes.



Access Point Information sub view displays the following information:

- Ethernet MAC
- Base Radio MAC
- AP Type
- Admin Status
- Monitor Only Mode
- Certificate Type
- Operation Status
- AP Model
- Location
- Group Vlan Name
- Primary Controller Name
- Secondary Controller Name
- Tertiary Controller Name
- Software Version
- Boot Version
- IOS Version
- Serial Number
- No of connected Users
- Data Sent
- Data Received

Aruba Access Point Information

The following attributes are displayed when you navigate to **Access Point > Information Tab** > **Aruba Access Point Information** sub view for an identified and modeled AP, under the **WLC Manager** > **Aruba** nodes.

The Aruba Access Point Information sub view displays the following information:

- MAC Address
- IP Address
- Name
- Group Name
- Up Time
- AP Type
- IP Sec Mode
- Location
- AP Model
- Operation Status
- Active Controller IP Address
- Standby Controller IP Address
- Access Point Connected as
- Software Version
- Serial Number
- No of Connected Clients
- Data Sent
- Data Received

Access Point Migration

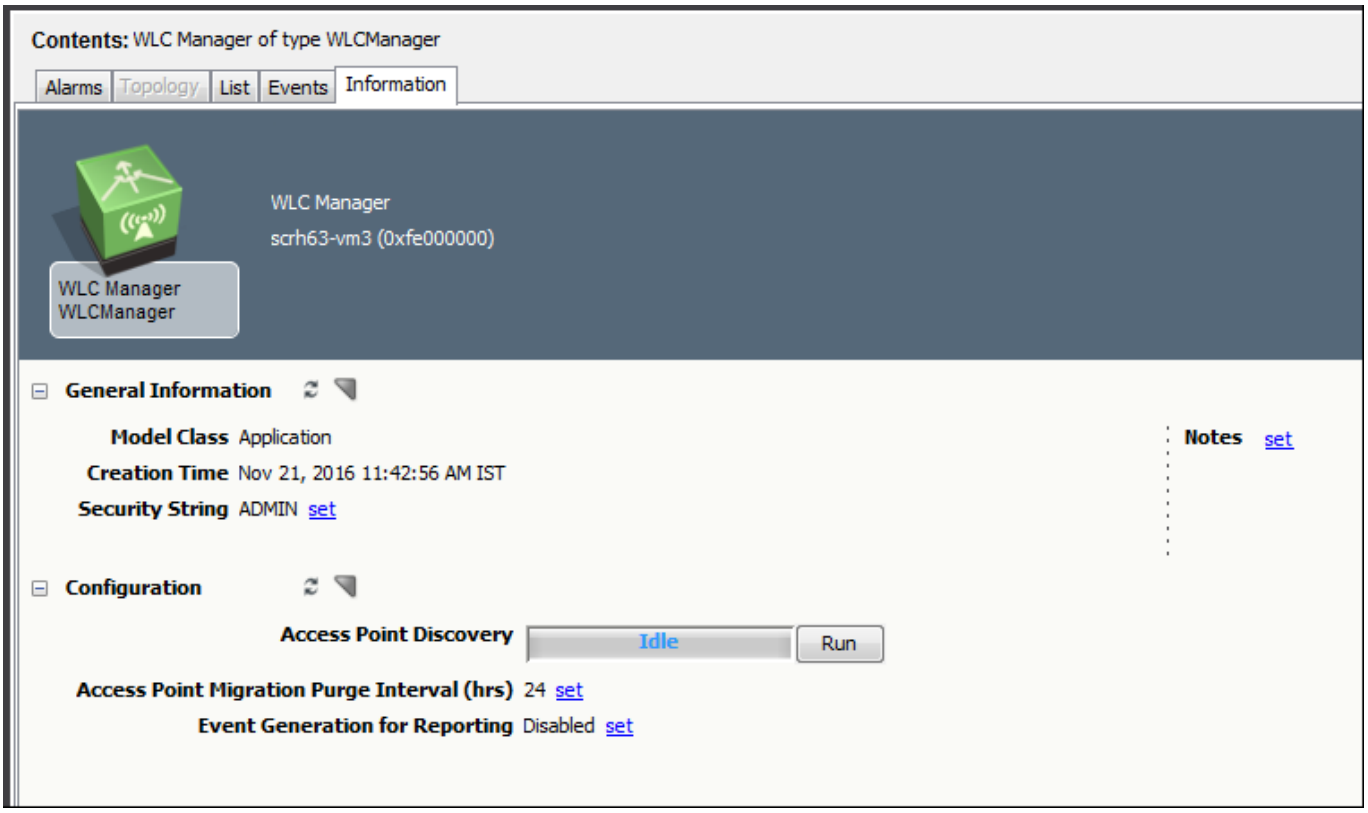
DX NetOps Spectrum identifies the mobility/migration of Access Points across WLCs. In DX NetOps Spectrum when you remove an AP physically from one WLC and connect to another WLC, the Access Point model will be moved from the previous WLC's (source WLC) hierarchy, to the WLC you have subsequently connected to (Destination WLC). This happens during the next polling cycle, of the destination WLC.
On successful migration, an AP migration event (**Attribute ID: 0x6760001**) will be raised on the Destination WLC.

NOTE

AP model is not deleted and recreated during migration (unless you explicitly do so), but same existing AP model is updated with its parent WLC relation.

The **Access Point Migration Purge Interval (hrs)** value is set to 24 (hrs), by default.

You can configure the same from **WLC Manager** > **Information Tab** > Configuration:



NOTE

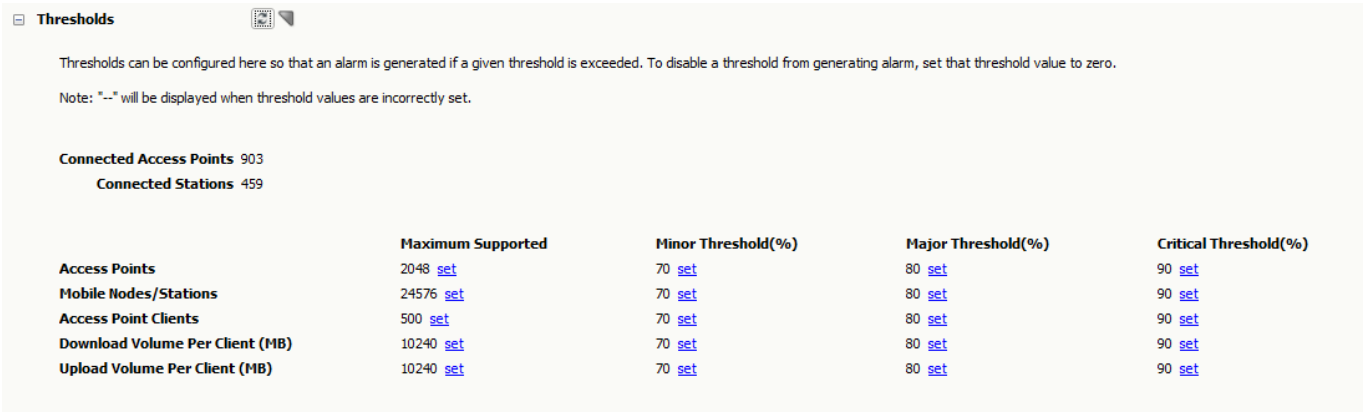
If Access Point is physically removed from one WLC and is not to attached/migrated to any other WLC within Purge interval, then AP Migration event will not be raised.

Configuring WLC related Thresholds

You can configure minor, major and critical thresholds from the **OneClick Explorer View** > **Wireless Device/ Controller level** > **Information Tab**> **Cisco Light Weight Access Point Protocol/ Aruba Controller Information** > **Thresholds** to generate alarms related to WLC/ AP/ Clients.
Please refer to the list below to view where the corresponding alarms are asserted.

NOTE

Threshold violation alarms will not work if the devices are modeled using snmp v1. It is recommended to model them using snmp v2 or v3.


Thresholds

Thresholds can be configured here so that an alarm is generated if a given threshold is exceeded. To disable a threshold from generating alarm, set that threshold value to zero.

Note: "--" will be displayed when threshold values are incorrectly set.

Connected Access Points 903
Connected Stations 459

	Maximum Supported	Minor Threshold(%)	Major Threshold(%)	Critical Threshold(%)
Access Points	2048 set	70 set	80 set	90 set
Mobile Nodes/Stations	24576 set	70 set	80 set	90 set
Access Point Clients	500 set	70 set	80 set	90 set
Download Volume Per Client (MB)	10240 set	70 set	80 set	90 set
Upload Volume Per Client (MB)	10240 set	70 set	80 set	90 set

Access Points

- Number of Access Points supported by current WLC device
- Max supported is defined in device configuration (WLC MIBs)

Alarm(s) are asserted on WLC

Mobile Nodes/Clients

- Number of Clients supported by current WLC device
- Max supported is defined in device configuration (WLC MIBs)

Alarm(s)are asserted on WLC

Access Point Users

- Number of Users supported by each AP (applicable for all APs under current WLC device)
- Max supported is configurable by the user (Default Value: 500)

Alarm(s) are asserted on AP

Download Volume Per Client

- Download data used per Client (applicable for all clients under current WLC device)
- Max supported is configurable by the user (Default value: 10240 MB)

Alarm(s) are asserted on AP, under which Client is latched

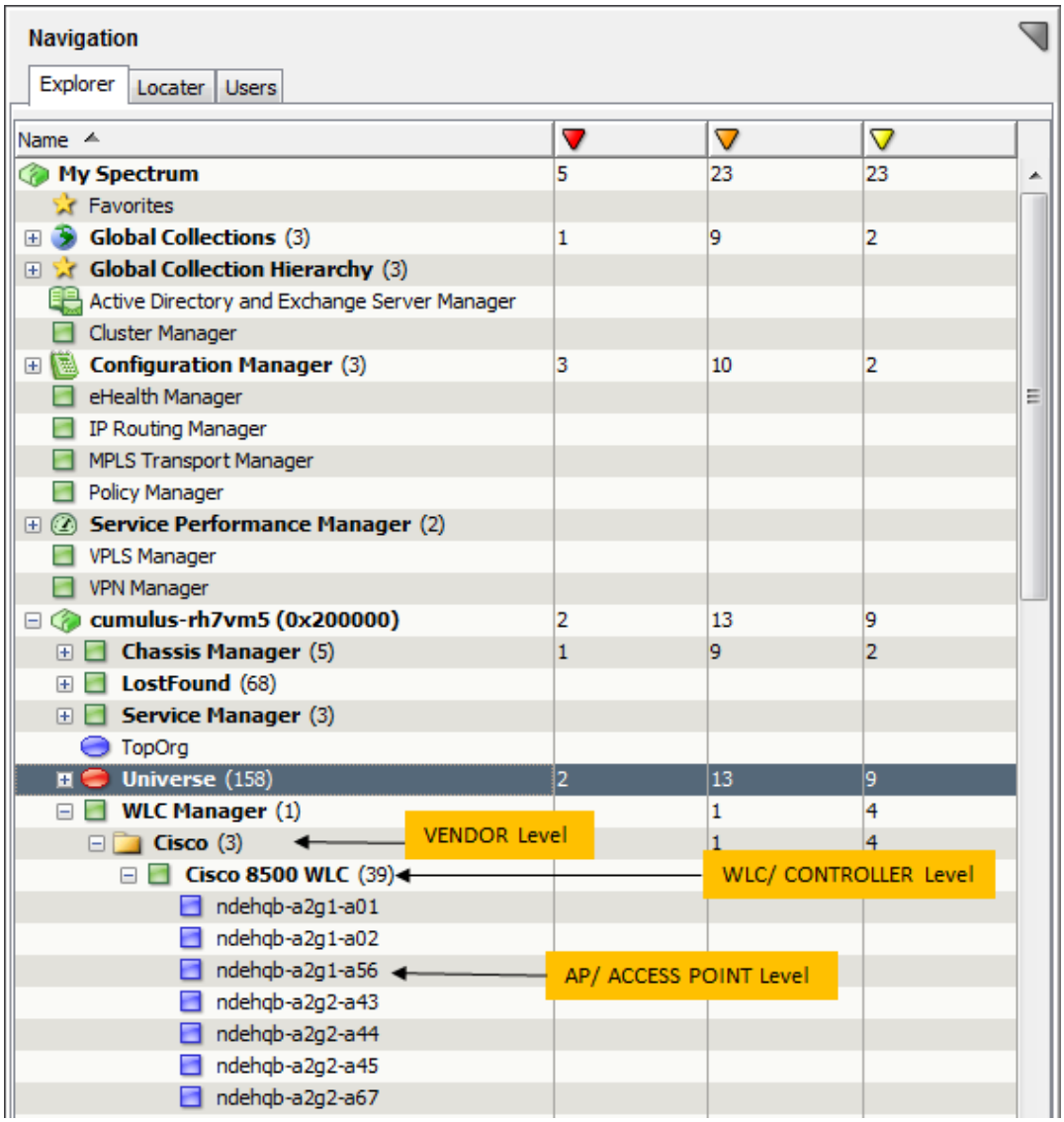
Upload Volume Per Client

- Upload data used per Client (applicable for all clients under current WLC device)
- Max supported is configurable by the user (Default value: 10240 MB)

Alarm(s) are asserted on AP, under which Client is latched

Explorer Hierarchy for WLC Manager

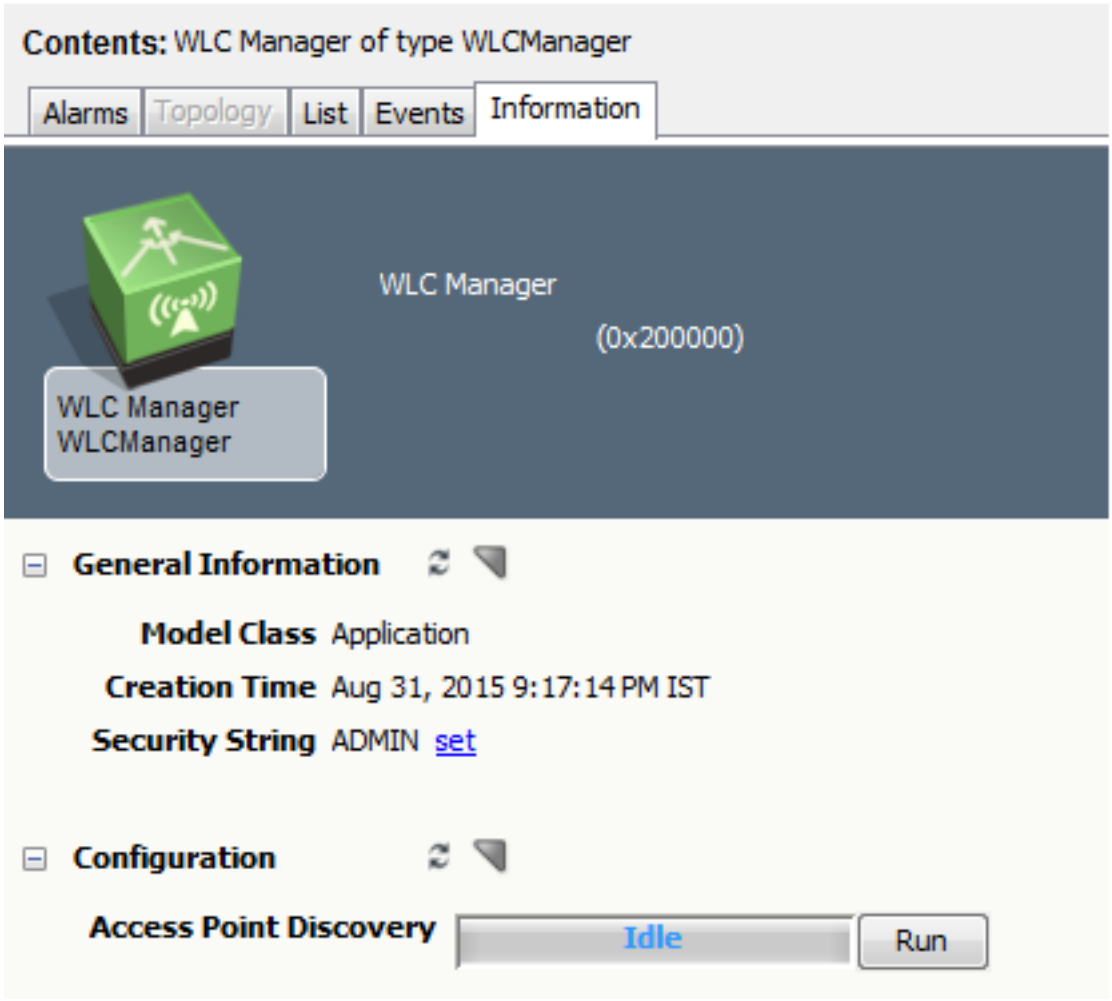
When the WLC devices are discovered, models will be created under **WLC Manager** under it's corresponding Vendor's folder, as indicated in the figure below. All access points will be created under their corresponding WLC Device(s) in the WLC Manager Hierarchy, as displayed in the **Explorer View> Landscape> WLC Manager > Navigation Hierarchy**.



In case you have not enabled the **"Create Wireless Access Points"** option in the **Discovery Console** > **Modeling Options** during initial discovery or used the **Create model by IP** option, Access Points **(APs)** will not be discovered.

To model APs from the WLC Manager (shown below) without performing the procedures mentioned above, do the following:

1. Navigate to **Explorer View > WLC Manager > Information Tab > Configuration** sub-view **> Access Point Discovery:** and then click **Run.**All access points **(APs)** will be created under their corresponding **WLC Device** (s).



WARNING

If you performs one of the following actions, affected WLC devices need to be remodeled (manually) in Spectrum:

- Delete/Remove operation at vendor level under **WLC Manager**
- Remove operation at WLC / Controller level under **WLC Manager** > **Vendor Level**

NOTE

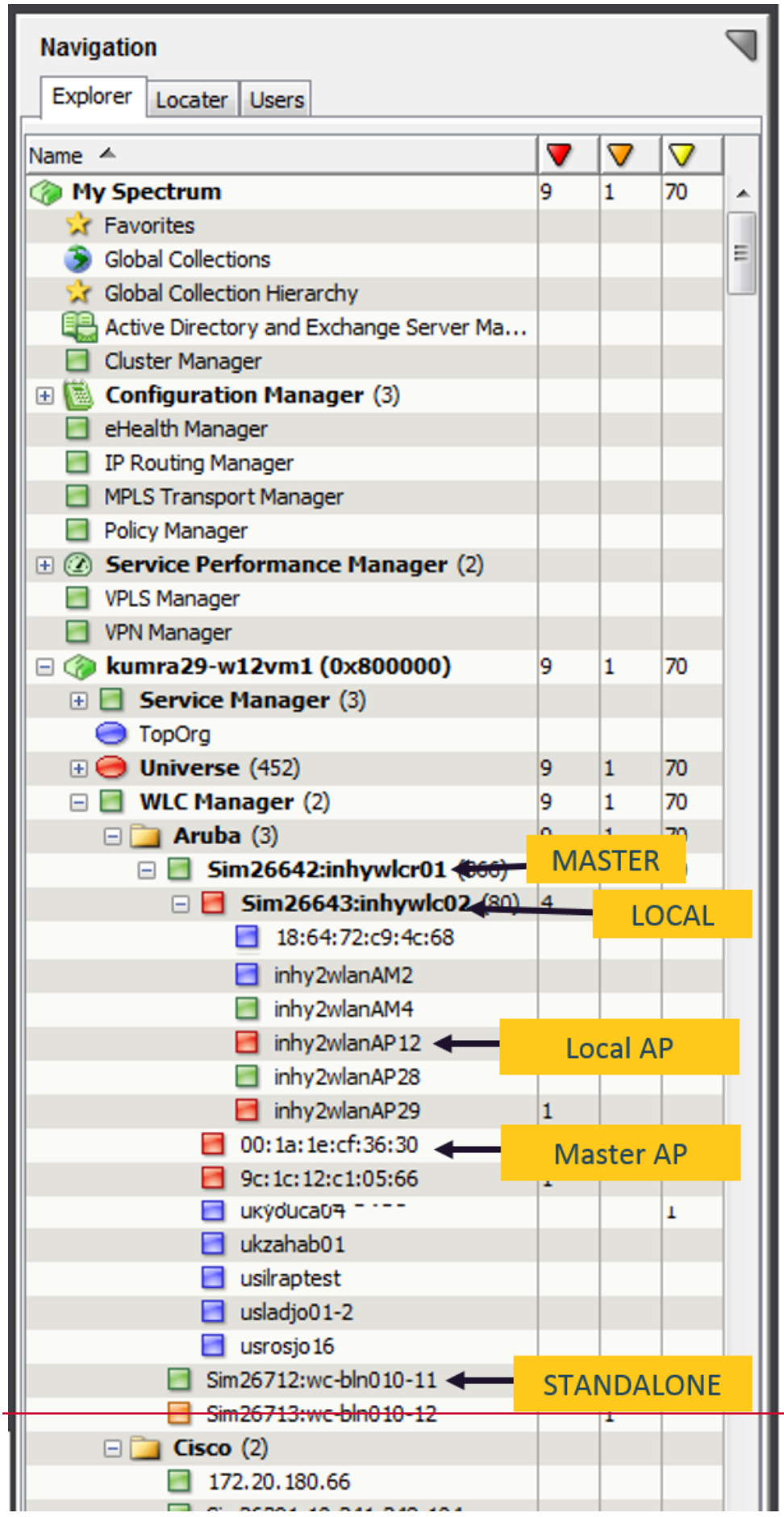
Cisco Wireless controller devices that were modeled prior to 10.1 will automatically associated under '**WLC Manager**' after upgrade.

If the modeled CISCO WLC device is down during the upgrade, Spectrum will not recognize device's vendor after upgrade. This device model will be kept under WLC Manager > **Other** vendor folder.

To discover & associate Access Points run "**Access Points Discovery**" from **Explorer View > WLC Manager > Information Tab > Configuration**. See image above for reference

Explorer Hierarchy for Aruba Controller

When the Aruba WLC devices are discovered, models will be created under **WLC Manager** under it's corresponding Vendor's folder, as indicated in the figure below. All the access points are created under their corresponding WLC Device(s) in the WLC Manager Hierarchy, as displayed in the **Explorer View> Landscape> WLC Manager > Navigation Hierarchy**. Since, Aruba WLC architecture follows a Master – Local design, the hierarchy is a little different from the Cisco WLC and AP hierarchy.


Navigation
Explorer
Locater
Users
Name
My Spectrum 9 1 70
Favorites
Global Collections
Global Collection Hierarchy
Active Directory and Exchange Server Ma...
Cluster Manager
Configuration Manager (3)
eHealth Manager
IP Routing Manager
MPLS Transport Manager
Policy Manager
Service Performance Manager (2)
VPLS Manager
VPN Manager
kumra29-w12vm1 (0x800000) 9 1 70
Service Manager (3)
TopOrg
Universe (452) 9 1 70
WLC Manager (2) 9 1 70
Aruba (3)
Sim26642:inhywlcr01
Sim26643:inhywlc02 (80) 4
18:64:72:c9:4c:68
inhy2wlanAM2
inhy2wlanAM4
inhy2wlanAP12
inhy2wlanAP28
inhy2wlanAP29 1
00:1a:1e:cf:36:30
9c:1c:12:c1:05:66
ukzahab01
usilraptest
usladjo01-2
usrosjo16
Sim26712:wc-bln010-11
Sim26713:wc-bln010-12
Cisco (2)
172.20.180.66
MASTER
LOCAL
Local AP
Master AP
STANDALONE

Access Point Switchover

Switchover driven by Access Points

In this scenario, there is no direct communication between primary or secondary WLCs. But switchover of traffic is driven by Access Points.
Access Points are manually configured (via Management Console/GUI), with Primary and Secondary WLC information. AP automatically switches to the secondary controller if the primary is not responding.

When Primary WLC goes down, AP will stop receiving a ‘keep-alive’ request from it.
By recognizing the loss of this request, AP starts sending a join request to secondary WLC IP (which is already configured), and establishes the connection. Once primary is up, AP again reverts its connection toward primary WLC IP.

This is the simplest failover configuration and the process may cause interruption of wireless service to the user.

NOTE

The WLCs are not HA aware and no heartbeat happens between WLC’s. However, an AP will always be up and running under any one of the WLCs.

AP-SSO: Access Point Stateful Switchover (Cisco WLC High Availability)

The Cisco SSO technology helps mitigate and improve user experience by reducing the downtime faced by the user by utilizing a secondary standby controller connected to the primary controller throughthe HA link over a dedicated redundancy port. In this case, when the primary controller goes down the AP can move over to the secondary controller instantly, this means the client devices will not experience any SSID downtime. The Cisco SSO technology also transfers the security keys of all the client devices connected through the AP, to the standby controller, so that the client can re-authenticate seamlessly.

NOTE

Cisco WLC High Availability.supports 1:1 (Active:Standby-Hot) stateful switchover of access points (AP SSO).

Before HA is configured, both the controllers will have their physical IPs and management IPs separately. Once HA is established, both the controllers will share common management IP (which is of primary) and primary acts like active, and secondary in Standby-Hot mode.

NOTE

There will be a heartbeat between WLCs and automatically if one WLC goes down the other will be active.

CA Spectrum 10.2 leverages the Cisco SSO technology where the two controllers are always in sync with a duplicate copy of the AP information.

As per Cisco's design, only active WLC is enabled with SNMP. In AP SSO, when the currently active controller (Primary) goes down, the standby controller (Secondary) becomes active and SNMP enabled.

When active WLC is modeled in DX NetOps Spectrum, standby WLC will be automatically modeled as Pingable (since SNMP is disabled on standby, as per Cisco design).
If secondary WLC is already down, at this time an alarm will be raised on the primary model, to indicate that the secondary WLC is down.

When the primary controller goes down the secondary controller becomes active, and its model will be upgraded from pingable to SNMP.
The SNMP community string, auto-placement and enable/disable of AP discovery, secure domain information, etc, for secondary WLC model, is referenced from the primary WLC model.

NOTE

Secondary WLC would be remodeled as an SNMP device model (that is, with a new model handle) replaces the earlier (pingable model) with an SNMP model.

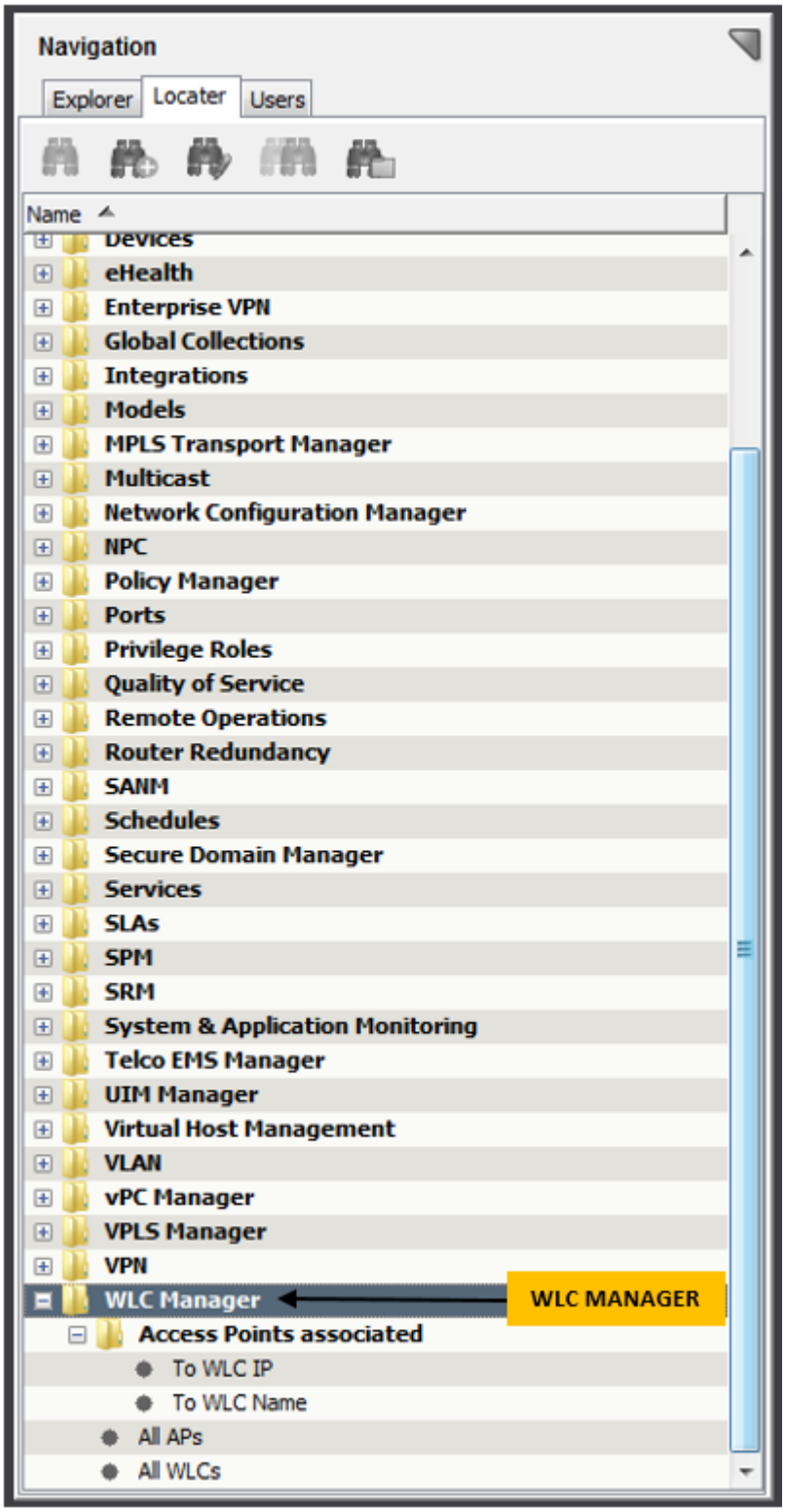
The Secondary (standby) controller will be upgraded to SNMP (active) based on two conditions:

- Primary controller contact status is down
- The secondary controller (pingable model) receives (Lost Peer, Moving to Active-No-Peer State) trap.

WARNING
Due to a limitation in the Cisco SSO technology, SNMP is enabled only on the Active WLC. Hence, Management Agent Lost alarm is generated only on the secondary WLC when the switchover happens from secondary to primary WLC.

Locater Search for Wireless Controller and Access Points

You can use pre-configured searches to locate entities in the DX NetOps Spectrum database that are related to Wireless technologies quickly. The searches are grouped under the **WLC Manager** folder in the **Locater** tab of the **Navigation** panel, as shown below:



The following searches are specific to Wireless models:

WLC Manager

Locates all devices that are modeled in the DX NetOps Spectrum database that have been identified as serving the specified role in one of the following searches:

- **Access Points associated**
  - **To WLC IP -** This search will return all Access Points associated with the specified WLC IP address.
  - **To WLC Name -** This search will return the Access Points associated with all WLCs whose model name matches the specified text.
- **All APs** - This search will return all access points modeled under the specified landscape(s).
- **All WLCs** - This search will return all the WLC devices under the specified landscape(s).

NOTE

Only those users with the appropriate privileges can access **WLC Manager** searches. For more information, see the OneClick Administration section.

Follow these steps, to view Access Points associated to WLC IP/ WLC Name:

1. Navigate to **Locater** tab, **WLC Manager**, Access Points associated to and select one of the following:
   - **TO WLC IP**
   - **TO WLC NAME**
2. Specify the WLC IP Address/ WLC Name in the relevant dialog box.
3. Do one of the following:
   - Click **Landscapes** button to filter the landscapes you wish to search against.
   - Click **List** button to import or enter values against which you wish to search.
4. Click **OK.**
   The results matching your query is displayed.

Schedule Discover Jobs for WLC Manager

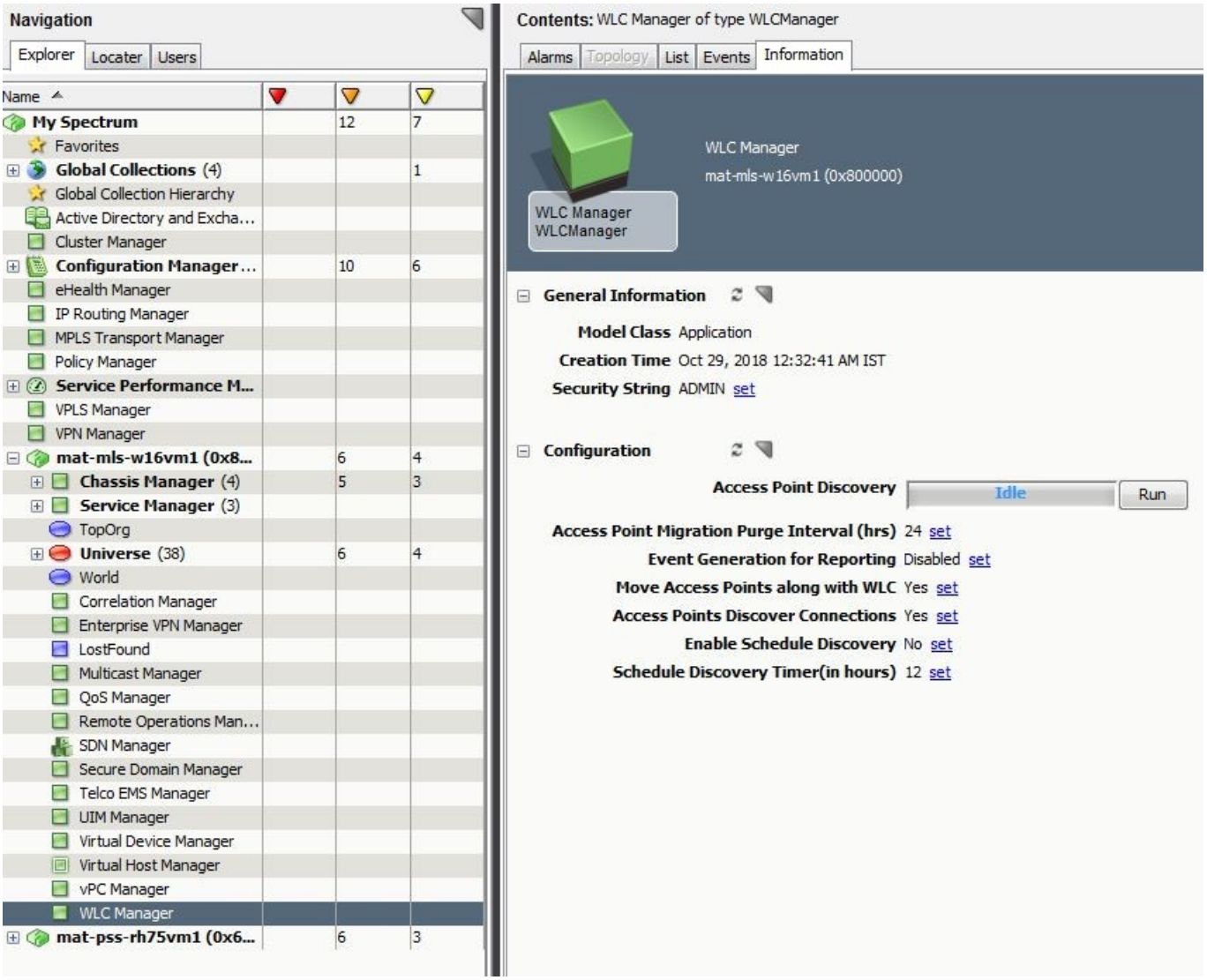
Scheduling discovery jobs for WLC AP addresses performance issues with the continuous discovery requests which are triggered from WLC APs. With an increase in the capacity that is the number of models, performance issues are prevalent, and scheduling discovery jobs can prove to be beneficial in managing large customer environments. This interim solution is introduced in Spectrum 10.3.1, to schedule time (default of 12 hours) and queue up all the discovery jobs to run them post schedule expiry time.

Users can now **set** the '**Enable Schedule Discovery**' feature to '**Yes**' to enable scheduling of discovery jobs, then **set** the '**Schedule Discovery Timer**' (in hours) to the time they want, for queuing up all the discovery jobs. Users can specify any given time (in hours) between 1 hour and 24 hours *only*. If a user does not specify a time, then by default the 'Timer' is set to 12 hours.

NOTE

If the schedule discovery option is enabled on an active server, that goes down and does not come up back until the scheduled discovery time, then the scheduled discovery does not run on the backup server.

Following is a screenshot of the WLC Manager Configuration Settings Tab:



Spotlighting Wireless Devices

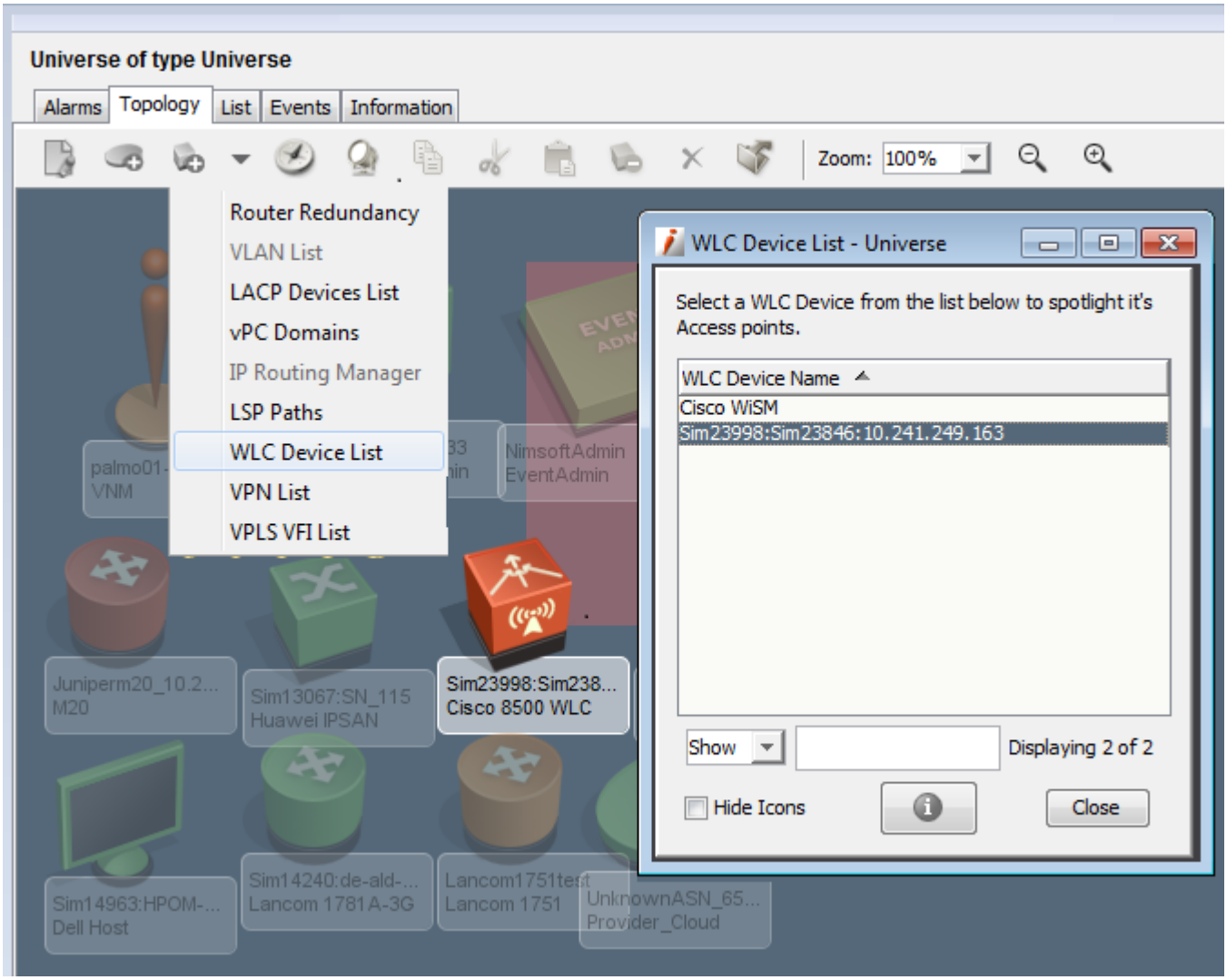
Spotlighting Wireless Devices in the Topology view helps you determine relationships between WLC Devices and Access Points on your network. Use the OneClick Spotlight feature to see all Access Points related to a WLC (Wireless Controller) in the Topology view.

NOTE
When spotlighting in the Topology views, you can select only single WLC device.

To spotlight Wireless Devices:

1. Open **OneClick.**
2. Expand the desired landscape on the **Explorer** tab and select **Universe**.
   Details about the selected **Universe** appear in the **Contents** panel.
3. Click the **Topology** tab.
   The topology of the Universe is displayed.

4. Click  **(S**
View) and select **WLC Device List** from the list of options.
The **WLC Device List** dialog opens.
5. Select a **WLC Device** from the list of options..

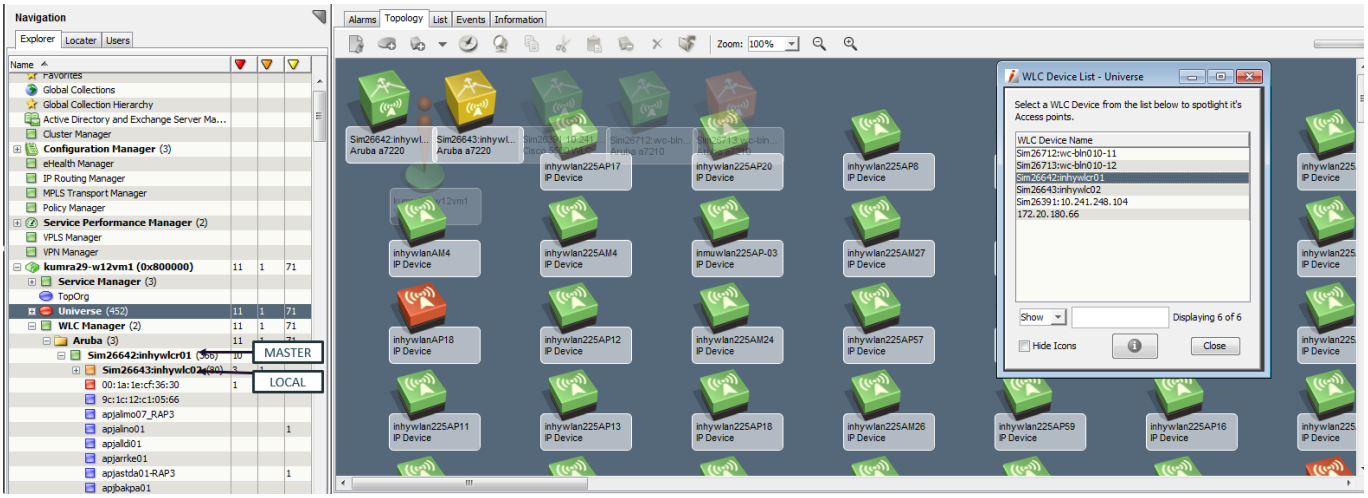


DX NetOps Spectrum spotlights the WLC device selected from the list and its corresponding APs (Access Points) by dimming all other models/devices in the topology.

Spotlighting Aruba Wireless Devices

Since an Aruba network uses a master/local architecture, if you select a Master controller from the WLC Device List, it's local controller(s) and the APs associated with the Master controller(s) are also highlighted.

As you can see in the following setup, the master controller is selected for spotlighting from the WLC Device List. The master controller and and it's associated AP along with it's local controller are also highlighted.



NOTE

You can use the **Hide Icons** option in the **WLC Device List** dialog to view only the selected WLC device and its APs.

NOTE

To view information about the selected WLC Device from the list, click the View the **Component Detail** icon on the **WLC Device List** dialog.

Support for Aruba WLC and Fault Tolerance

In a large campus WLAN that has separate network services and aggregation layers, APs and AMs should never terminate on the master controller. APs and AMs should terminate only on the local controller. Aruba Mobility Controllers are deployed in clusters that consist of a master and one or more local controllers to solve this management scalability issue. This design is the recommended model when two or more controllers exist in the same network.

Large scale deployments generally involve more than two controllers. When you have more than a single pair of controllers, change control and network consistency can become an issue. In an Aruba network that uses a master/local design, configuration is performed only on the master and it is pushed down to the locals. Local controllers reside at the aggregation layer of the Aruba overlay architecture. They handle AP termination, user authentication, and policy enforcement. When you configure any local controller, you must know the IP address of the master and the pre-shared key (PSK) that was used to encrypt communication between the controllers. The control channel between all Aruba controllers is protected by an IP Security (IPsec) connection.

NOTE

Note: The controllers have a preconfigured key at first boot. Change this key after the first boot so that the operation of the master/local cluster is secure.

For more details on the functions and responsibilities of master and local mobility controllers in Aruba architecture, see the Aruba Mobility Controllers and Deployment Models Validated Reference Design.

Fault Tolerance in Aruba WLC

Large network would have a master/backup master pair and a lot of local controllers that point to them. If the master controller has a problem, the backup master would step in, and you would still have an appliance that has read-write capability in your network without any interruption.

If you have a master/local, but no backup master, you would lose the ability to make global changes to your network. Your network would however, still be able to run.

With this configuration, if the master becomes unreachable or unavailable and no standby master has been configured, the network continues to operate as expected, except for certain operations. You cannot perform configuration, RF visualization, or location services until connection to the master controller is restored. The master controller is needed to perform configuration and reporting, but it is not a single point of failure in the network.

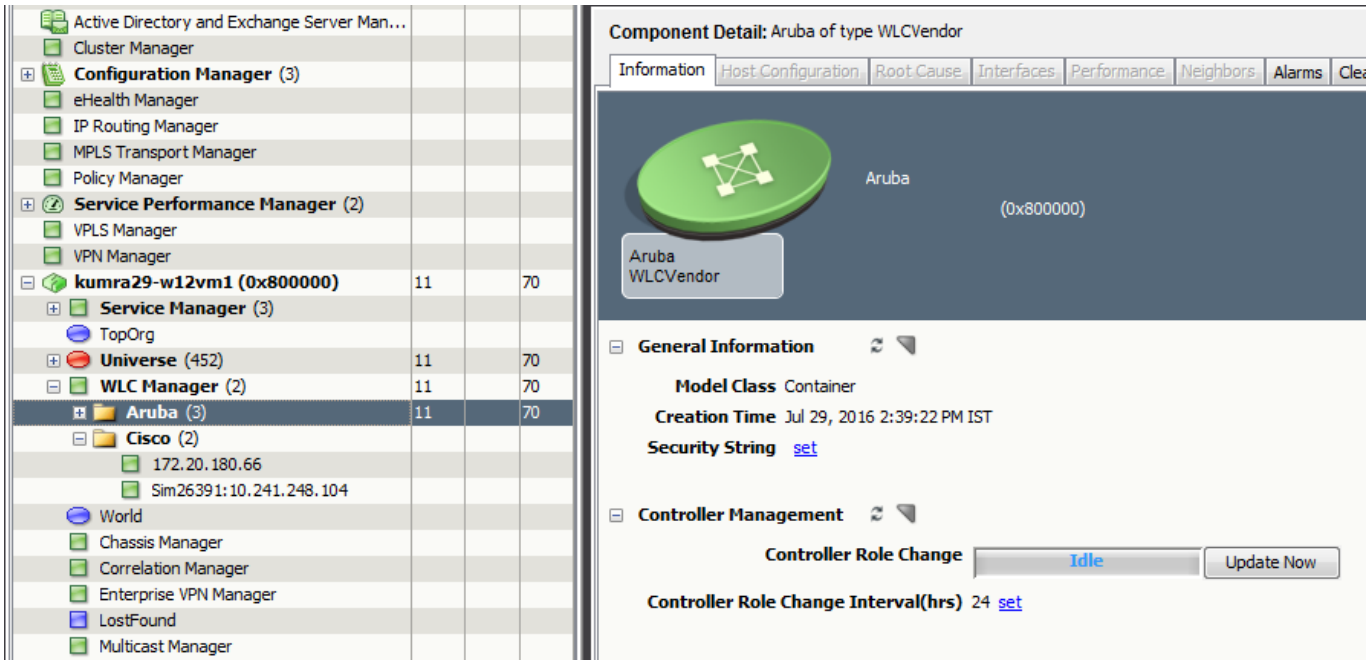
The master is where all configuration changes are made. The local connects to the master and gets the majority of its configuration from it. Selecting "Standalone" during the startup wizard just allows you to avoid configuring credentials for a local to connect to it. A "Standalone" controller is essentially a master. All controllers, whether Master or Local, can terminate access points.A backup master is a master controller that you cannot terminate access points on. It's sole responsibility is to back up the only read-write appliance you have in your network. A master does a lot of the database processing, so in larger environments, you don't want access points to terminate at all.

Controller Management

DX NetOps Spectrum allows you automatically detect the Role change (Master to Backup, Master to Local , Local to Master, etc all combinations).

This supports the ' ***wlsxNSwitchRoleChange'*** trap and poll based Association changes in WLC Manager Topology. AP's should automatically moved based on the role change(s) occurred between Master and Local controllers, after every device polling interval.You should be able to see the role changes for the configured **Role Change Interval Period**, which is 24 hours by default.However, the next scheduled polling interval is user configurable.

1. Select the **Aruba** vendor folder in the OneClick Explorer, navigate to the **Component detail: Aruba of type WLCVendor** and select the **Information** tab.
2. Expand **Controller Management**.



3. Use the **Update Now** button, for the following scenarios:
   - View changes made at Controller level to roles (master and local) and the resultant change in hierarchy
   - If Traps are disabled so switchover (role change) information and hierarchy mapping is not reflecting or happening
   - You have made changes to the hierarchy and wants the changes to reflect before the next scheduled polling interval.

4. Click **set** and configure your desired Controller Role Change Interval in hours.

Certifying and supporting virtual systems within Check Point Firewall

Overview

10.2 certifies and enables discovery and modeling of virtual systems present in the Checkpoint Point Firewall.

Each Checkpoint Firewall has a Primary context and multiple Virtual contexts, which can be treated as separate Firewalls. The primary and virtual contexts share the same IP address, but maintain their own set of interfaces and routing tables. With SNMPv2, you cannot discover and model virtual systems, only the root context information can be fetched.

WARNING

If you want to monitor virtual systems within a CheckPoint Firewall you need to have the Firewall configured with SNMPv3.

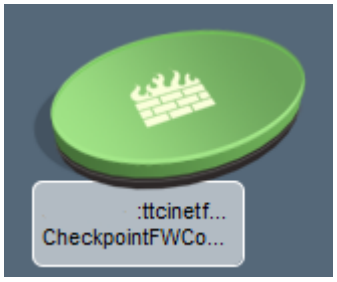
A separate container is created when DX NetOps Spectrum discovers a Check Point firewall device that has virtual systems. Using the context name to discover the virtual systems of the Checkpoint Firewall, DX NetOps Spectrum communicates with each virtual systems and fetches the corresponding interface information and other VPN, VSX, and connectivity related information.

NOTE

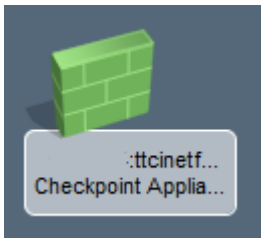
Please ensure that you enable the **802.3 Fanout** check box during discovery of upstream switches and Check Point Firewall.

Topology View

The following icons represent the virtual entities associated with Check Point virtual Firewall after they are modeled in DX NetOps Spectrum:



Represents the container for Check Point Firewall that has virtual systems.

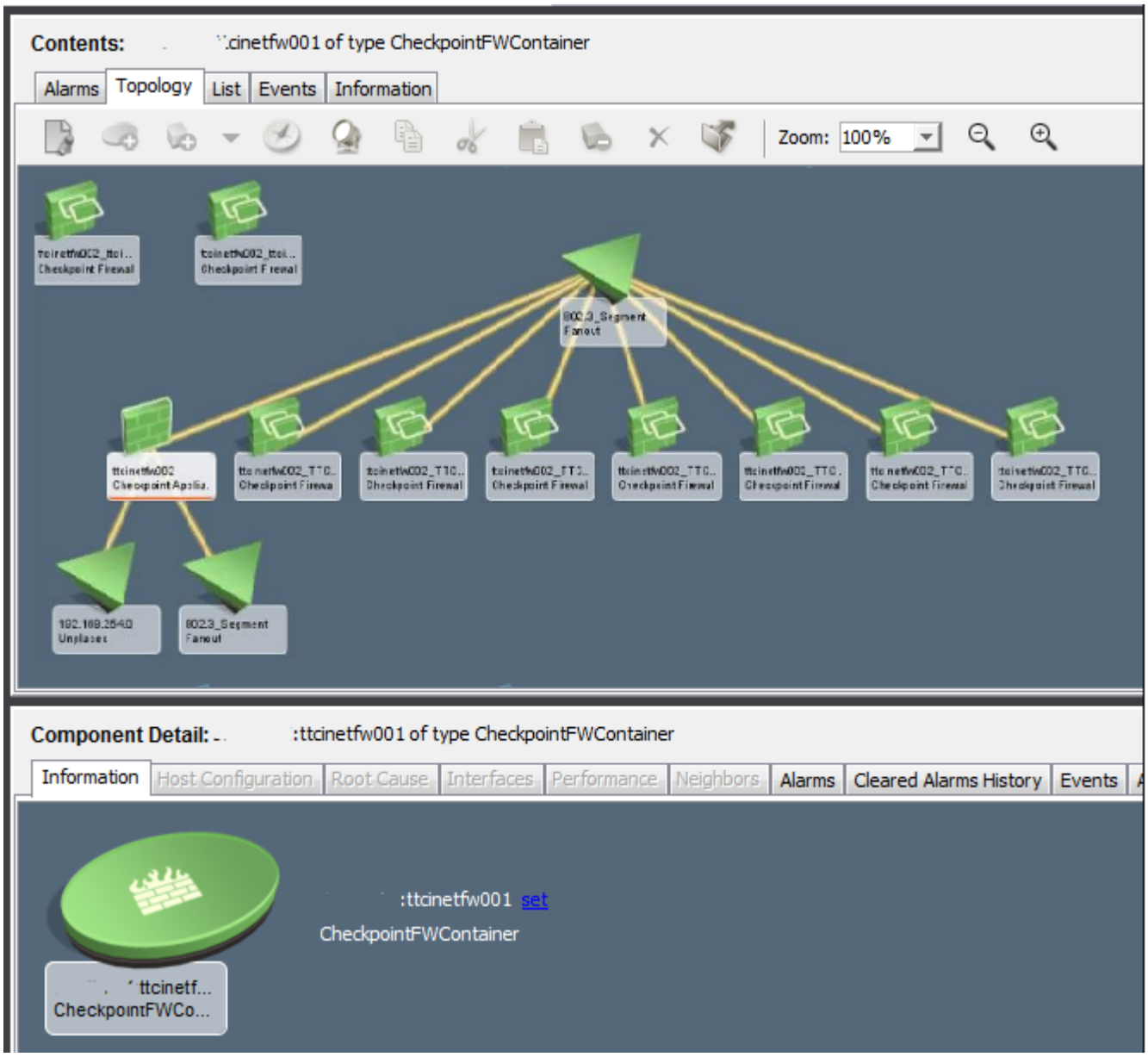


Represents the models with Root context

Represents the models with Virtual context

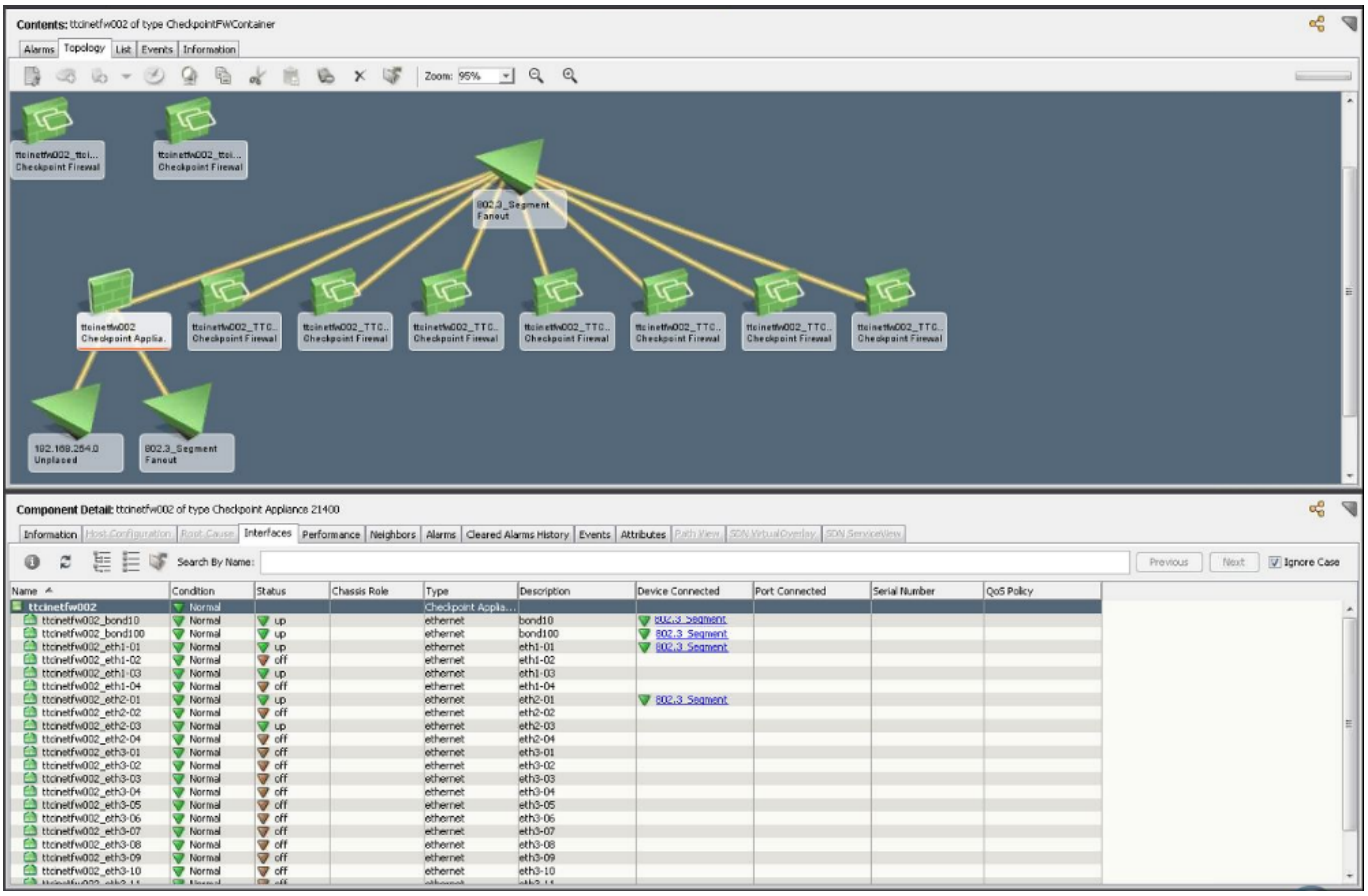
To view the models (related to Check Point Firewall) in its relevant context, follow these steps:

In the OneClick Console, **Explorer View**, navigate to the **Check Point Firewall container** > Check Point appliance/ root context, and select the **Topology** tab.
The Topology view displays all the systems in its root/ virtual context:



Interface Tab and connections between Neighbors and virtual systems

You can drill down and view the information corresponding to the interfaces and devices connected to the virtual systems within the Check Point firewall root context. This enables you to view neighboring virtual systems as individual devices and their connection details. You can view all the independent interfaces connected to a virtual system in a root context.

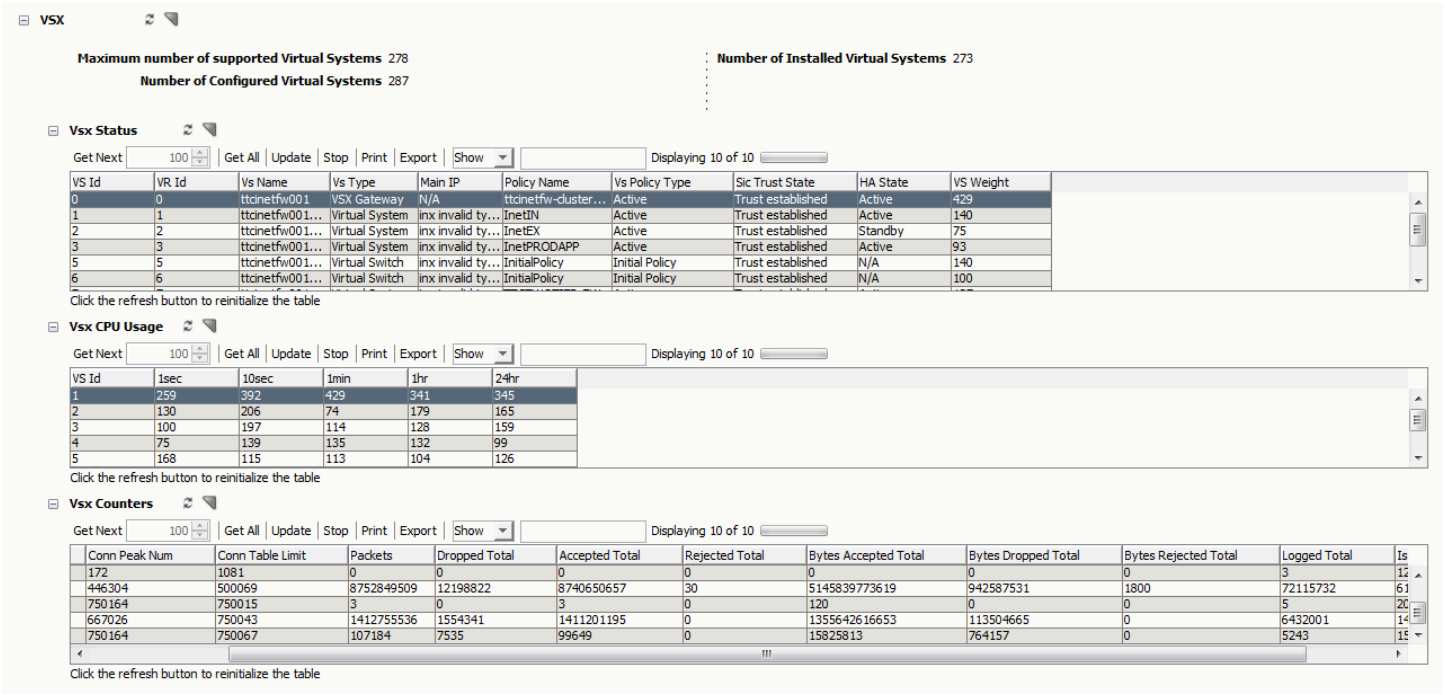


Information Tab details

The following fields of information are displayed under **Information Tab > CheckPoint Firewall** for Check Point Firewall root context models:

- **CheckPoint Firewall**
- **VPN**
- **Management**
- **VSX**

The following fields of information are displayed under the VSX node in the Information Tab:



- **ASG**

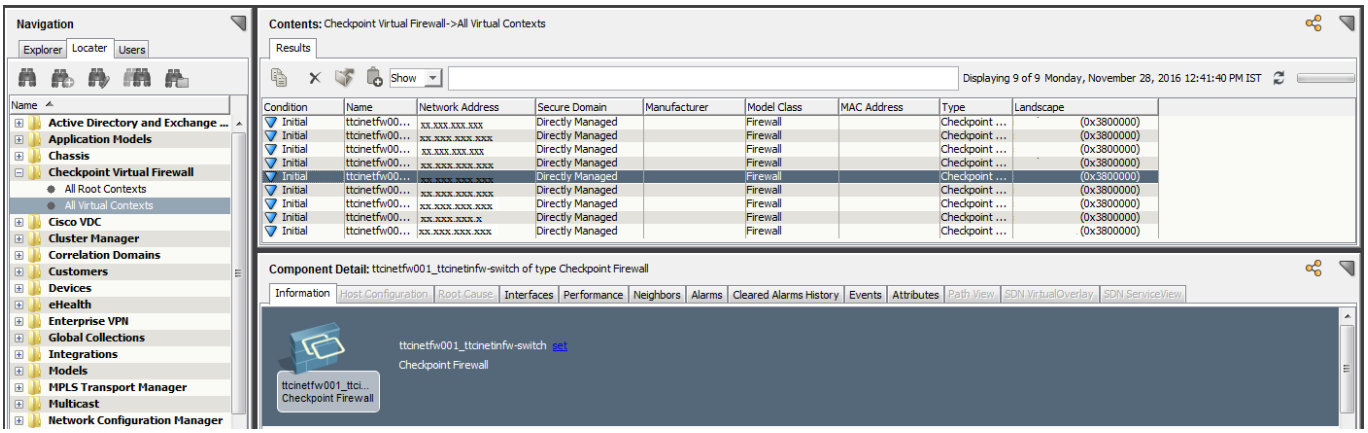
Alarm Correlation

After modeling of the virtual systems is complete, DX NetOps Spectrum monitors the virtual entities normally as it monitors independent switch / physical chassis devices. As a result, when any number of virtual systems within a Check Point firewall device go down, corresponding number of alarms are generated on DX NetOps Spectrum.

However, if the Check Point Firewall device is down and assuming that the virtual systems are functioning independently as per functionality, all virtual entities will go down and raise separate alarms. However, if all the virtual systems along with root context are down, using it's correlation domain capabilities, DX NetOps Spectrum suppresses alarms on all the virtual systems and generates a single alarm on the Checkpoint Firewall Container.

Locater Search for Check Point Virtual Firewall

You can use pre-configured searches to locate Check Point Virtual Firewall context entities related to Check Point Virtual Firewall devices, in the DX NetOps Spectrum database quickly. The searches are grouped under the **CheckPoint Virtual Firewall** folder in the **Locater** tab of the **Navigation** panel, as shown below:



The following searches are specific to Check Point Virtual Firewall models:

CheckPoint Virtual Firewall
Locates all devices that are modeled in the DX NetOps Spectrum database that have been identified as serving the specified role in one of the following searches:

- **All Root Contexts:** This search will return all models with Root context.
- **All Virtual Contexts**: This search will return all models with virtual context.

Follow these steps, to view appliances/ virtual devices associated to Check Point Virtual Firewall:

1. Navigate to **Locater** tab, **CheckPoint Virtual Firewall**, and select one of the following:
   – All Root Contexts:
   – All Virtual Contexts:
2. Select the landscapes you wish to search against, in the Select Landscapes to Search dialog box.
3. Click OK.
   The results matching your query is displayed in the Contents pane.

Troubleshooting for Checkpoint Virtual Context

Problem: Checkpoint virtual contexts stay blue initially.

Solution: Enable the virtual context to poll.

1. Create an SNMPv3 user.
2. Enable the VS mode.
3. Start the SNMP agent.

Use the following sample commands:

```
> add snmp usm user admin security-level authNoPriv auth-pass-phrase abcd1234
> set snmp mode vs
> set snmp agent on
```

The contexts are modeled and are green now.

Enhanced VRF support for Cisco Nexus devices

Starting from 10.2.1, VRF support for Cisco Nexus devices is enhanced. DX NetOps Spectrum now displays the list of VRF available on the devices along with the IP Routing and IP Address table for each VRF. You can now select the specific VRF you want from the VRFs available on the router/device and view the IP Routing/ IP Address table corresponding to the selected VRF.

Virtual Routing and Forwarding technology that allows multiple instances of a routing table to coexist on the same router at the same time. There can be multiple VRFs on the same router. Prior to this release the Interface tab would not populate the IP Addresses of non-default VRFs. Even if the device had multiple VRFs, only the instances belonging to the default VRF would display the IP Addresses. From this release we are polling the IP Address Table, per VRF, mapping the IP Address table and populating data in the **Interfaces** tab > **IP Address** column.

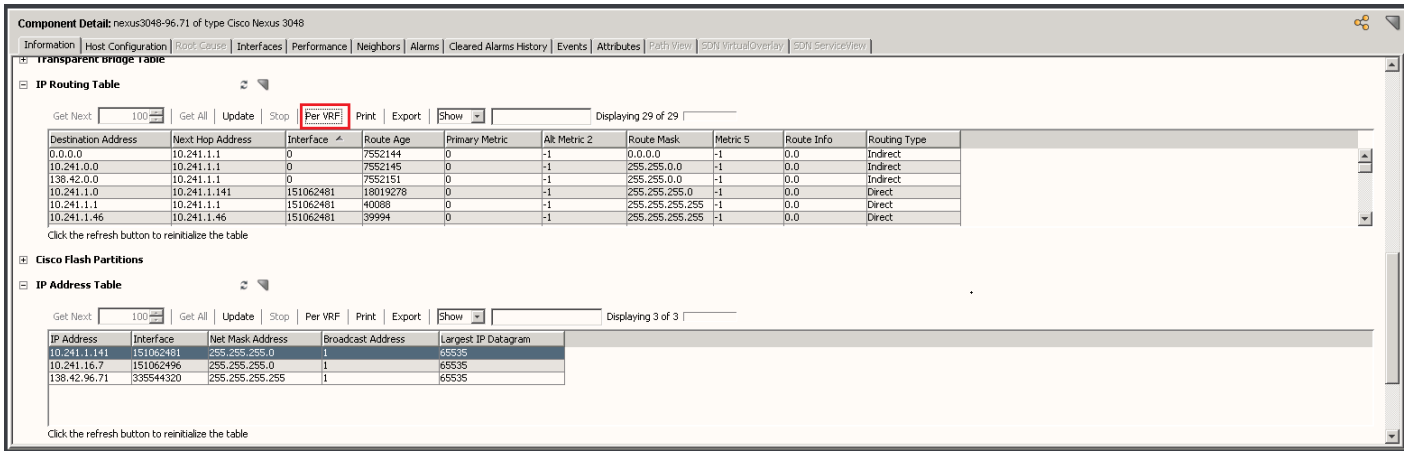
DX NetOps Spectrum now displays the list of VRFs available on the router/devices and the routing table for each VRF. You can select the VRF by clicking the Per VRF button and view the corresponding/associated routing table.

WARNING

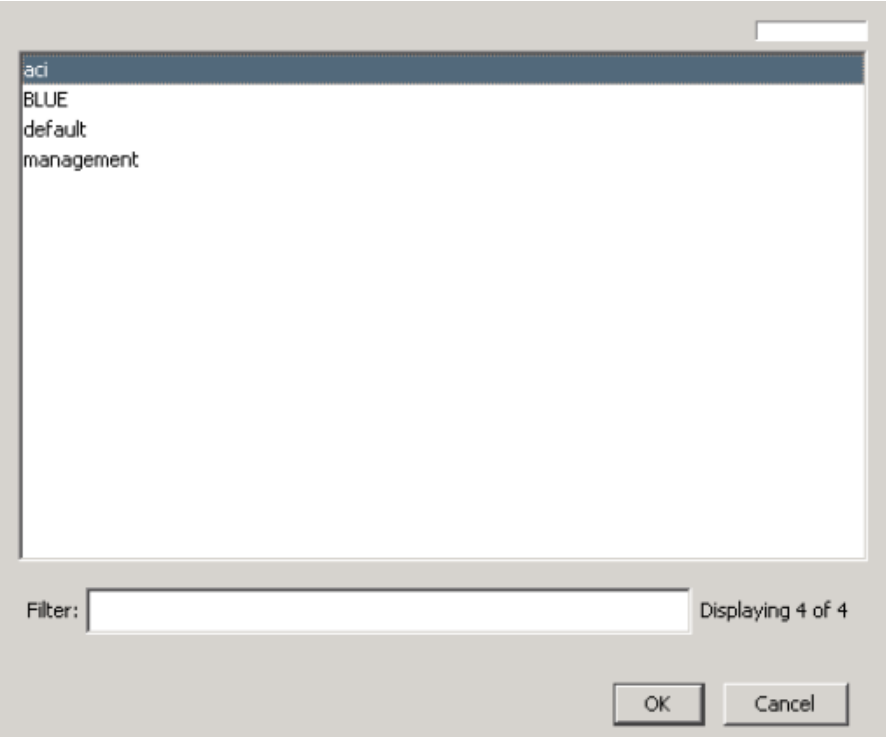
Prerequisite! You have to create/configure the VRF context in the device on the SNMP side, i.e. context mapping needs to be performed as a prerequisite to be able to get the VRF-IP Routing Table.

To view the IP Routing table for a specific VRF, follow these steps:

1. In the OneClick > **Explorer** View, navigate to **Virtual Device Manager** > Cisco Nexus device (which has VDC/VRF) > **Information** tab.
2. Select one of the following:
   - **IP Routing Table**
   - **IP Address Table**



3. Click **Per VRF**.
   The **Select VRF** dialog box displays the list of VRFs available on the router/ Cisco Nexus device.



4. Select the **VRF** you wish to view the IP Routing/ IP Address table data, and click **OK**.
   The IP Routing/ IP Address table now displays the data specific to the VRF you have selected.

   WARNING

   If there are multiple routes for the same destination in Cisco Nexus device, the IP Routing Table in the Information Tab, does not list all the routes, as the MIB used (IPRoutingTable) populates only one next hop address. This is applicable for default or any other VRF.

Alcatel Device Management

Alcatel PSS Series

The Photonic Service Switch (PSS) is a technology used for optical networking which converts a single wavelength of light into a virtual network. It is comprised of optmized platforms with various deployment environments for networking, interconnecting data centers positioned at network junction points to multi-layer, efficiently scaling large metro and multiservice optical networks. As part of service assurance and managing underlying IT infrastructure DX NetOps Spectrum leverages common software, hardware, management and control to offer seamless operations across the network through the Photonic Service Switch (PSS) technology.

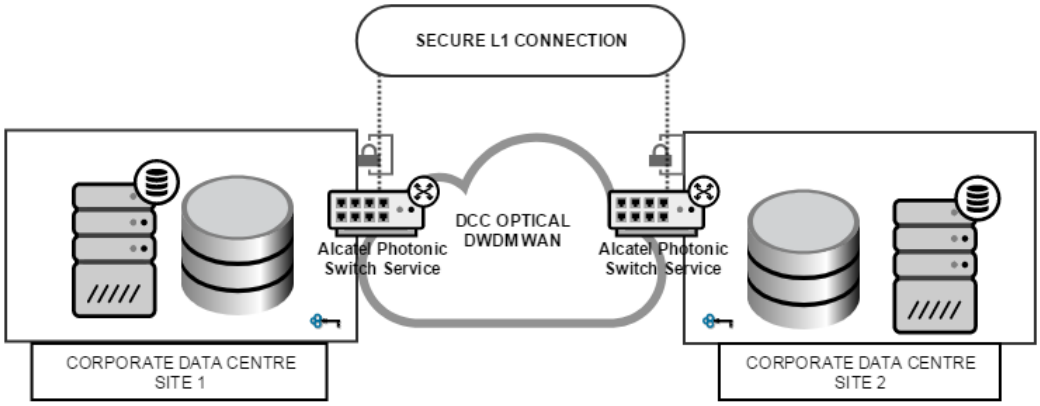
The various technologies involving in tracking, encoding and supporting, administrating and maintenance of an integrated optmized wavelength include:

- OTN switching
- Wavelength routing
- Embedded multi-layer capability
- Integrated packet transport
- Transport software-defined networking (T-SDN)

Alcatel Photonic Services Switch (PSS) Deployment Topology

Dense wavelength division multiplexing (DWDM) is a technology that offers data from various sources together on an optical fiber, with each signal carried at the same time on its own separate light wavelength. Using DWDM, up to 80 (and theoretically more) separate wavelengths or channels of data can be multiplexed into a light stream transmitted on a single optical fiber. Dense Wavelength Division Multiplexing (DWDM) is the only technology that allows full network flexibility and adaptability at speeds of 100G and beyond, quick service turnup to meet changing bandwidth needs, and a ultra-low latency connectivity. DWDM offers reliable transport-grade of protocol-independent data.

This is an illustration of the PSS Deployment Topology



Alcatel Photonic Service Switch (PSS) MIB Support

List of Photonic Service Switch MIBs:

TROPIC-SYSTEM-MIB
TROPIC-SYNCE-MIB
TROPIC-SOFTWARE-MIB
TROPIC-STATISTICS-MIB
TROPIC-SLOT-MIB
TROPIC-SHELF-MIB
TROPIC-PTP-MIB
TROPIC-POWERMGMT-MIB
TROPIC-PHMNOTIFICATION-MIB
TROPIC-OTUODU-MIB
TROPIC-OSPF-MIB
TROPIC-OTH-MIB
TROPIC-OPTICALCARD-MIB

TROPIC-OCH-MIB
TROPIC-NOTIFICATION-MIB
TROPIC-OPTICALPORT-MIB
TROPIC-LOG-MIB
TROPIC-L1SERVICE-MIB
TROPIC-GMPLS-NOTIFICATION-MIB
TROPIC-ABSNODE-NOTIFICATION-MIB
TROPIC-ABSNODE-MIB
TROPIC-EXPRSCALARS-MIB
TROPIC-VTSCONN-MIB
TROPIC-WAVEKEY-MIB
TROPIC-USERMGMT-MIB
TROPIC-TC
-TN-TC-MIB
TN-VRTR-MIB
TN-SAS-SERV-MIB
TN-SDP-MIB
TN-SAS-SDP-MIB
TN-SAS-PORT-MIB
TN-SAS-IEEE8021-PAE-MIB
TN-SAS-IEEE8021-CFM-MIB
TN-RMD-TSOP-MIB
TN-RMD-TC-MIB
TN-RMD-SYSTEM-MIB
TN-RMD-IF-MIB
TN-RMD-EFM-MIB
TN-RMD-CFM-MIB
TN-PORT-MIB
TN-PMON-MIB
TN-MIRROR-MIB
TN-MC-REDUNDANCY-MIB
TN-IGMP-SNOOPING-MIB
TN-IEEE8021-CFM-MIB
TN-ETH-RING-MIB
TN-FILTER-MIB
TN-DOT3-OAM-MIB
TN-CONN-PROF-MIB
TN-CLEAR-MIB
TN-BFD-MIB
TN-SAP-MIB
TN-SERV-MIB

TN-LOG-MIB
TN-LAG-MIB
TN-LLDP-MIB
TN-QOS-MIB
TN-SAS-QOS-MIB
TN-MPLS-TP-MIB
TN-OAM-TEST-MIB
TN-MPLS-MIB`

NOTE

The photonic service switch (PSS) device is not populating data related serial number and firmware version.

The device memory related OIDs are not found in MIBs and CPU related OIDs are not populating data, thereby disenabling the performance tab

Alcatel PSS Trap Support

VendorID	Model Type	OID	Trap Name	Varbind Name	Event ID	Alarm
Custom	Global Level	1.3.6.1.4.1.7483.2.1.2.2.2.2.1.0.149	tnPortTransmissionFailureRaisedNotif	tnTrapTime tnTrapObjectIDType tnTrapObjectID tnTrapCategory tnTrapDescr tnTrapData tnTrapServiceAffectin tnTrapEntityType tnTrapCondition	0x04ca061d	0x4ca061d

Managing Systems

This section contains information about how to use ADES Manager, Cluster Manager, and Virtual Host Manager.

Active Directory and Exchange Server Manager

The DX NetOps Spectrum Active Directory and Exchange Server Manager (ADES Manager) feature models and monitors your Microsoft Active Directory and Microsoft Exchange Server environments. ADES Manager provides an enterprise-wide view of your Active Directory and Exchange Server environments, showing topology as well as the logical relationships between servers. ADES Manager also provides visibility into key Active Directory and Exchange Server metrics. Finally, ADES Manager helps you pinpoint and effectively troubleshoot problems by applying unique fault isolation techniques to your Active Directory and Exchange Server environments.

ADES Manager is intended for DX NetOps Spectrum administrators who want to monitor Active Directory and Exchange Server hosts.

ADES Manager Features

DX NetOps Spectrum ADES Manager features include:

- Automated device discovery and modeling. ADES Manager automatically creates models and connections for all managed Active Directory and Exchange Server hosts.
- A distributed solution that can handle scaling. Domain management can be distributed across multiple SpectroSERVERs.
- Identification of Active Directory and Exchange Server hosts in the topology.
- Hierarchical representation of Active Directory and Exchange Server environments.
- Dedicated ADES Manager views that provide visibility into data specific to Active Directory and Exchange Server environments.
- Enhanced fault management. ADES Manager recognizes and suppresses symptomatic alarms and aids fault isolation with proxy management.
- Locater searches specific to Active Directory and Exchange Server.

System Requirements

ADES Manager works within DX NetOps Spectrum when all required components are configured properly. ADES Manager requires the following components:

- A dedicated host machine with:
  - CA SystemEDGE 5.x or later
  - Active Directory and Exchange Server AIM (ADES AIM) r12.7 with latest PTFs, or later

WARNING
The ADES AIM must be the only AIM installed on the CA SystemEDGE host. The CA SystemEDGE host itself cannot be a host in your Active Directory and Exchange Server environment.

NOTE
For CA SystemEDGE host and ADES AIM requirements, see the *CA Virtual Assurance for Infrastructure Managers* documentation.

Supported Technologies

DX NetOps Spectrum Active Directory and Exchange Server Manager supports the following product versions and technologies:

Active Directory for Windows Server 2012 and Windows 2016

- Active Directory Domain Services (AD DS) Server Role

Exchange Server 2007, 2010

- Hub Transport Server Rol
- Mailbox Server Role

Active Directory Overview

Active Directory is a directory service that provides administrators the ability to discover and manage network resources throughout the organization. Using Active Directory, you can efficiently manage directory-enabled objects (such as users, computers, groups, printers, and applications) from one secure, centralized location. DX NetOps Spectrum ADES Manager helps you to manage and monitor your Active Directory environment so that you can increase the availability of your network resources.

An Active Directory implementation can vary in size. You can have a few objects to millions of objects. Active Directory enables administrators to manage centrally enterprise-wide network information from a repository that is globally replicated. Once the information has been added to Active Directory, it is available throughout the entire enterprise.

Active Directory uses server roles to assign different functions to different servers, and a single server can fulfill several roles at once. The following server roles are available for Active Directory:

- **Active Directory Certificate Services (AD CS)**
  Allows you to create, distribute, and manage customized public key certificates.
- **Active Directory Domain Services (AD DS)**
  Stores directory data for all objects in your network and manages the communication between users and domains, including authentication requests and directory searches.
- **Active Directory Federation Services (AD FS)**
  Provides secure identity technologies that are used to authenticate users for access to resources.
- **Active Directory Lightweight Directory Services (AD LDS)**
  Provides support for directory-enabled applications without the restrictions of AD DS.
- **Active Directory Rights Management Services (AD RMS)**
  Protects your digital information from unauthorized use by identifying the rights that a user has to a file.

> **WARNING**
> DX NetOps Spectrum ADES Manager supports the AD DS server role only. The following section provides more information on this role.

Active Directory Domain Services (AD DS)

Active Directory Domain Services provide the central location of the directory. The directory stores configuration information, authentication requests, and other information about all the objects in your network. The basic internal structure of the Active Directory is a hierarchical arrangement of objects.

The following components of the Active Directory structure are used in DX NetOps Spectrum ADES Manager:

- **Forest**
  An Active Directory container structure that contains a collection of Active Directory objects, their attributes, and attribute syntax. A forest is at the highest level of the logical structure. A forest is a collection of domain trees sharing a common global catalog, directory configuration, directory, schema, and logical structure.
- **Domain**
  An Active Directory container structure that contains a collection of objects that share a common set of policies, name, and security database. A domain is at the lowest level of the logical structure of an entire network. The domain name identifies the domain.
- **Domain Controller**
  A host that is running AD DS. Typically multiple domain controllers host Active Directory within a domain. You can manage your network resources from any domain controller within your domain.

Exchange Server Overview

Exchange Server is a back-end product that provides messaging services (such as email, calendar, and contacts) to its end users. With email and messaging as business-critical tools, your Exchange Server implementation must be able to support a highly available messaging environment. DX NetOps Spectrum ADES Manager helps you to manage and monitor your Exchange Server environment so that you can achieve increased levels of reliability.

Exchange Server provides the underlying infrastructure to support a messaging system, which includes the following components:

- The database to store email data
- The transport infrastructure to move the data from one place to another
- Access points to access email data from a number of different clients

Exchange Server uses server roles to assign these different functions to servers throughout the enterprise, and you choose which roles each server supports. You can install only the roles you need, and you can split server functions across multiple servers. You can also install more than one role on a single machine.

The following server roles are available in Exchange Server:

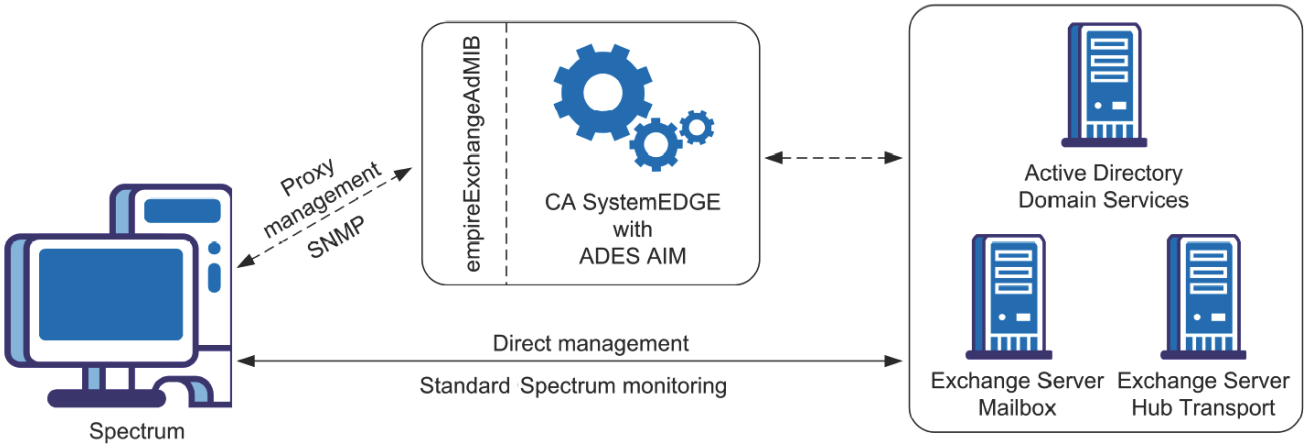
- **Mailbox**
  Provides email storage (including user mailboxes), advanced scheduling services, and supports public folders. Continuous replication technology provides a reliable failover mechanism in the event of failure. In Exchange 2007, continuous replication failover is at the server level. With Exchange 2010 and the introduction of database availability groups (DAGs), failover is at the database level.
- **Client Access**
  Handles how users connect to Exchange by supporting functions such as Outlook, POP3, and web services such as calendar sharing.
- **Hub Transport**
  Handles email flow and routing. All messages are delivered through this role, regardless of whether they are being delivered locally or remotely.
- **Unified Messaging**
  Integrates your phone system with your email, handling automated call routing and directing voicemails to the appropriate user mailbox.
- **Edge Transport**
  Supports antispam and antivirus functions for inbound and outbound messaging.

  WARNING
  DX NetOps Spectrum ADES Manager supports the Mailbox Server and Hub Transport Server roles only.

Solution Architecture

ADES Manager monitors your Active Directory and Exchange Server environments seamlessly within your network, while providing data that is specific to the supported server roles. DX NetOps Spectrum gathers information about your Active Directory and Exchange Server hosts using two different methods. As with other DX NetOps Spectrum-managed devices, ADES Manager uses standard DX NetOps Spectrum monitoring. In addition, ADES Manager also retrieves specialized information from an alternate (proxy) manager, a SystemEDGE Application Insight Module (AIM). Specifically, ADES Manager uses the Active Directory and Exchange Server AIM (ADES AIM).

An AIM is a specialized extension of the SystemEDGE agent and resides on its own host. This host is referred to as the Active Directory and Exchange Server Host Manager (ADES Host Manager). The ADES AIM obtains data from Active Directory and Exchange Server hosts that is specific to the Active Directory and Exchange Server role technologies. This data is then written to a CA-developed MIB (empireExchangeAdMIB). DX NetOps Spectrum then accesses the data from the MIB using SNMP requests. This solution allows other SNMP clients, such as CA eHealth, to utilize the ADES AIM. Each ADES AIM can support multiple domains, and ADES Manager can support multiple AIMs either within a single SpectroSERVER or distributed across multiple SpectroSERVERs.



NOTE
For more information about the ADES AIM, see the *CA Virtual Assurance for Infrastructure Managers* documentation.

Planning Your ADES Manager Implementation

The purpose of ADES Manager is to monitor your Active Directory and Exchange Server environments. ADES Manager is highly scalable and can monitor hundreds of servers using multiple AIMs and distributed SpectroSERVERs. You can configure your ADES Manager implementation differently for performance, geographical, and logical purposes. Understanding the various configuration and management options provides for a more efficient ADES Manager implementation.

Environment Management Considerations

During the setup of ADES Manager, you specify how to organize the management of your environments. With a small environment, you can have one ADES AIM managing all your hosts in one domain in one location on one SpectroSERVER. In a complex environment, you can have multiple ADES AIMs to manage different host subsets from multiple domains in different locations across multiple SpectroSERVER landscapes.

Although organizational specifications can be changed at any time, knowing the available configuration management options allows for a better initial set up.

Consider the following points when setting up your ADES Manager environment:

- A single ADES AIM can manage hosts from one or more domains.
- One or more ADES AIMs can manage a single domain.
- Domain management can be distributed across multiple SpectroSERVERs using multiple ADES AIMs. Multiple ADES AIMs can also be supported within a single landscape.
- Manage a host by one ADES AIM only. If a host fulfills multiple roles, manage all the roles by the same ADES AIM.

When deciding how to distribute host management, consider the number and location of hosts in your environment. The number of hosts an ADES AIM manages and the geographic proximity of the AIM to the monitored environment can affect performance. For the best performance, size and balance management of the environment appropriately.

NOTE
Domain-level management is controlled on the ADES AIM and not in DX NetOps Spectrum. For information on domain specification, load balancing, and sizing guidelines, see the CA Virtual Assurance for Infrastructure Managers Administration section.

Host Modeling

As with other network elements supported in DX NetOps Spectrum, you discover and model your Active Directory and Exchange Server hosts to monitor them. ADES Manager uses different types of discovery when modeling your ADES environment:

- Discovery by the ADES AIM (ADES AIM Discovery)
- Discovery within DX NetOps Spectrum (ADES Manager Discovery)

ADES Manager relies on the ADES AIM to discover and provide information about available Active Directory and Exchange Server hosts in the specified domains. ADES Manager then uses this information to model each host individually in DX NetOps Spectrum.

> **NOTE**
> Information that is used for the ADES Manager feature is gathered primarily from the ADES AIM. More information is also gathered directly from the hosts.

The following topics provide more details about the modeling process.

> **NOTE**
> Procedures for discovering and modeling your environment are provided in Discovery and Modeling ADES Environment.

What is Modeled

ADES AIM Discovery finds any host in the specified domain or domains that supports any of the supported roles. These hosts are made available for management in DX NetOps Spectrum. Supported server roles include Active Directory Domain Services, Exchange Server Hub Transport, and Exchange Server Mailbox. If a machine only has unsupported roles, the machine is not included in the list of available hosts and the machine is not modeled. Of the available hosts, not all are necessarily modeled in DX NetOps Spectrum. Only those hosts that are designated by the DX NetOps Spectrum administrator for DX NetOps Spectrum ADES Manager to manage are discovered and modeled in DX NetOps Spectrum.

> **NOTE**
> You can specify in the ADES AIM to monitor Active Directory only, Exchange Server only, or both technologies. When monitoring a single technology, only those hosts having a supported role in that technology are included as available hosts. You can also specify in the ADES AIM to manage automatically all newly available hosts. In this case, all newly available hosts having any of the supported roles are modeled automatically in DX NetOps Spectrum. Functionality that the AIM provides is described more in Understanding ADES AIM Technology.

Models that are created during ADES Manager Discovery are placed in the topology in an ADES Managed Hosts container specific to the ADES AIM. These models are also visible in the Active Directory and Exchange Server Manager hierarchy. If a host that has already been modeled in your DX NetOps Spectrum-managed network before ADES Manager Discovery, it is not modeled again. Also, the original model is not moved in the topology during ADES Manager Discovery. Although the model is not reflected in the ADES Managed Hosts container in the topology view automatically, it does appear in the ADES Manager hierarchy. In certain cases, you can move the host model into the ADES Managed Hosts container manually.

> **NOTE**
> Active Directory and Exchange Server host models can also be moved out of the ADES Managed Hosts container. This container is not limited to Active Directory and Exchange Server hosts. Other devices can be moved into the container when the administrator considers the inclusion to be logical. If the container is destroyed, all models in the container (except for models in a global collection) are moved to the Lost and Found (LostFound). Any models that are moved manually into the container and are not necessarily Active Directory or Exchange Server hosts are also moved.

Modeling Methods

Hosts in the Active Directory and Exchange Server environment are modeled as SNMP-managed elements when possible. SNMP-capable modeling supports enriched device monitoring that can provide added value to your ADES Manager solution. If an SNMP agent is not installed on the host, it is modeled as an ICMP (Pingable) device.

Model Naming

When modeling Active Directory and Exchange Server hosts, the model name that DX NetOps Spectrum assigns depends on the type of modeling used, as follows:

- For SNMP modeling, DX NetOps Spectrum automatically attempts to supply a name for the model using standard DX NetOps Spectrum naming conventions. Automatic naming is controlled at the SpectroSERVER level as indicated by the Model Naming Order field on the SpectroSERVER Control view for the VNM model.
- For ICMP (Pingable) modeling (when not a virtual device), DX NetOps Spectrum uses the hostname that the ADES AIM provides.

  WARNING
  For ICMP (Pingable) modeling, model names that Virtual Host Manager sets take precedence over ADES Manager.

The administrator can modify the name of an Active Directory or Exchange Server host model at any time. As with other managed network elements, DX NetOps Spectrum automatically updates the model name using established naming rules, which can replace the user-defined value. To retain a user-defined value, lock the model name.

NOTE
You can modify and lock the model name using the following model attributes: Model_Name (0x1006e) and Lock_Model_Name (0x12a52).

IP and MAC Address Determination

When modeling Active Directory and Exchange Server hosts, the IP and MAC addresses that DX NetOps Spectrum assigns depends on the type of modeling used:

- For SNMP modeling, DX NetOps Spectrum automatically attempts to determine the addresses by querying the resident SNMP agent.
- For ICMP (Pingable) modeling (when not a virtual device), DX NetOps Spectrum uses the addresses that the ADES AIM provides.

  WARNING
  For ICMP (Pingable) modeling, addresses that Virtual Host Manager sets take precedence over ADES Manager.

If neither SNMP modeling or Virtual Host Manager can supply a valid IP or MAC address, the ADES AIM value is used.

Host Management and Multiple ADES AIMs

Manage an Active Directory or Exchange Server host by a single ADES AIM. If, inadvertently, multiple ADES AIMs manage a host, expect the following behavior:

- If the managing ADES Host Managers are in a single landscape, duplicate models are not created for the Active Directory or Exchange Server host. ADES Manager recognizes when another ADES AIM is managing a host and issues an alarm on the Active Directory or Exchange Server host.
- If the managing ADES Host Managers exist across multiple landscapes, duplicate models exist.

  WARNING
  Managing a single host by multiple ADES AIMs can introduce performance issues. For more information, see the CA Virtual Assurance for Infrastructure Managers Administration section.

Hosts Managed by Multiple DX NetOps Spectrum AIM Solutions

When managing a host model by multiple DX NetOps Spectrum AIM solutions, defined ranked order of management applies, as follows:

1. Virtual Host Manager
2. Cluster Manager
3. Other technologies (such as ADES Manager)

When a host with a SystemEDGE agent is already modeled in DX NetOps Spectrum, ADES Manager recognizes the model and a duplicate model is not created. Instead, ADES Manager pulls the existing model into its own management, abiding by and applying the rules for each solution using the ranked order.

For example, when both the Virtual Host Manager and ADES Manager are managing a host, model parameters that Virtual Host Manager assigns are used. Examples of these parameters include the model name, IP address, and MAC address.

When a solution no longer manages a device, the rules of the remaining solutions are reapplied in the ranked order. Typically, any changes are made at the next polling cycle.

This defined order of management also affects how models appear in the Universe topology.

> **NOTE**
> For more information, see the Virtual Host Manager section and the Cluster Manager section.

Supporting Changes to Your Active Directory and Exchange Server Environments

After the initial modeling of your Active Directory and Exchange Server environments in DX NetOps Spectrum, changes to the environment are not automatically detected. ADES AIM discovery must be started manually to recognize and model any subsequent changes.

The ADES AIM performs its discovery of the Active Directory and Exchange Server environments when the AIM starts up. Then, DX NetOps Spectrum models the environment accordingly. After the initial ADES AIM Discovery, start an ADES AIM discovery manually for the ADES AIM to recognize any changes in the environment. DX NetOps Spectrum then automatically reflects these changes in its modeled environment.

Because of the potentially large size of the Active Directory and Exchange Server environments, dynamic updates would be expensive to maintain. The manual start process also allows the administrator to address many environment changes at the same time in a controlled time frame.

> **NOTE**
> For information about updating your modeled ADES environment, see Updating ADES Environment.

Understanding ADES AIM Technology

ADES Manager uses the ADES AIM for discovery, modeling, and monitoring of your Active Directory and Exchange Server environments. This section describes functionality that the ADES AIM controls and provides. Review these topics before setting up your ADES Manager implementation.

After the initial setup, configurations can be modified. Some settings can only be changed directly in the ADES AIM, while others can be changed from within DX NetOps Spectrum.

> **NOTE**
> This section describes features of the ADES AIM at a high level. For detailed information, see theVirtual Host Manager section.

Discovery Options for the ADES AIM

This section describes the ADES AIM Discovery options. These options are defined in the ADES AIM and control what hosts in your Active Directory and Exchange Server environments are discovered. These options also control how the discovered hosts are initially managed. The main settings on the ADES AIM are:

- **Domain name**
  Controls what domains to monitor.
- **Management entity**
  Controls what technologies to monitor: Active Directory, Exchange Server, or both. This setting is specified on a per-domain basis.
- **Management mode**
  Controls how discovered hosts are initially monitored: domain-based or host-based. This setting is specified on a per-domain basis.

The following explanation describes how these settings work together and how DX NetOps Spectrum ADES Manager uses them.

When you install and configure an ADES AIM, you provide the names of one or more domains to manage. You also specify by domain whether to manage Active Directory hosts, Exchange Server hosts, or both.

Using the specified domain names, the ADES AIM queries the Global Catalogs to retrieve all Active Directory and/or Exchange Server hosts (based on your setting). The ADES AIM then identifies the roles that are configured on each server. The ADES AIM uses the respective roles of each host to determine whether the host qualifies for inclusion as an available host.

For each domain, you also specify whether the ADES AIM initially manages (domain-based management) or does not manage (host-based management) newly discovered hosts. Domain-based management identifies and automatically manages all new hosts that serve the requested technology within the domain. Host-based management identifies but does not manage all new hosts by default. Domain-based management is typically used with domains that are small enough for a single AIM to manage. Host-based management is typically used with domains that are large enough to be managed by multiple ADES AIMs.

> **NOTE**
> Using host-based management, you specify manually within DX NetOps Spectrum which hosts to manage. See Specify Hosts to Manage by ADES Manager.

Polling by the ADES AIM

The ADES AIM polling interval indicates how often the ADES AIM queries the managed hosts for information. The default value is 300 seconds with a minimum value of 30 seconds. The interval must be a multiple of 30 seconds.

The ADES AIM polling interval can be modified from within DX NetOps Spectrum. For more information, see Updating ADES AIM Configuration Options.

> **WARNING**
> DX NetOps Spectrum uses its own polling interval to control how often to poll the ADES AIM. For more information on polling intervals that ADES Manager uses, see Controlling Polling Intervals to Update ADES Data.

Installing ADES Manager Components

ADES Manager is included in all DX NetOps Spectrum extraction keys. When you install DX NetOps Spectrum, the ADES Manager components are automatically installed and available for use. However, ADES Manager is operable only after you also install and configure the SystemEDGE agent and ADES AIM. To manage your environment properly, DX NetOps Spectrum must be able to contact the SystemEDGE agent, which has the ADES AIM loaded. The AIM must be able to communicate with your Active Directory and Exchange Server hosts.

To install ADES Manager properly, the administrator must complete these tasks:

- Install the SystemEDGE agent and load and configure the ADES AIM, specifying the domain or domains for monitoring. Note the following requirements:
  - The SystemEDGE agent and ADES AIM must be installed on a Windows host that meets the following conditions:
    - The Windows host Is a member server in one of the domains to be monitored, with a trust relationship to the remaining domains.
    - The Windows host does not have any Active Directory or Exchange Server roles.

      WARNING
      Do not install the SystemEDGE agent and ADES AIM on a host that ADES Manager is going to manage.
  - The ADES AIM must be the only AIM installed on the SystemEDGE host.
  - The ADES AIM can monitor the domain using various configurations. The ADES AIM can be configured to monitor Active Directory hosts, Exchange Server hosts, or both. The ADES can also be configured to use either domain-based or host-based management mode.

    NOTE
    For more information about ADES AIM functionality at a high level, see Understanding ADES AIM Technology. For details about installing the CA SystemEDGE agent and ADES AIM, see the *CA Virtual Assurance for Infrastructure Managers Installation section* and *CA Virtual Assurance for Infrastructure Managers Administration section*, respectively.
- Install DX NetOps Spectrum with ADES Manager included.

  WARNING
  Do not install the SpectroSERVER on a machine that ADES Manager is going to manage.

  NOTE
  For specific installation instructions, see the Fresh Install section.

You can now discover and model your Active Directory and Exchange Server environment in DX NetOps Spectrum. See Discovery and Modeling ADES Environment.

Discovery and Modeling ADES Environment

After you have installed the necessary components, discover and model any hosts that ADES Manager is going to manage. The following types of discovery are used:

- Discovery on the ADES AIM (ADES AIM Discovery)
- Standard DX NetOps Spectrum Discovery
- Discovery within DX NetOps Spectrum of Active Directory and Exchange Server hosts to be managed (ADES Manager Discovery)

ADES Manager relies on the ADES AIM discovery to collect information about available Active Directory and Exchange Server hosts in the specified domains. Then, through ADES Manager Discovery, ADES Manager uses this information to model each Active Directory or Exchange Server host individually. Models are placed in the hierarchy and in ADES Managed Hosts containers in the topology. Using standard DX NetOps Spectrum logic, connections are established between the hosts and the upstream devices that are modeled during standard DX NetOps Spectrum Discovery.

Discover and Model Your Active Directory and Exchange Server Environment

The following steps are necessary to discover and model your Active Directory and Exchange Server environments:

1. Run a standard Discovery to model the ADES Host Manager and connecting devices.
2. If necessary, upgrade the CA SystemEDGE model.

3. Specify hosts to manage by ADES Manager.
4. Let ADES Manager Discovery run.

ADES AIM Discovery

ADES AIM Discovery occurs automatically when you set up the ADES AIM on the ADES Host Manager. The following explanation describes the ADES AIM Discovery process and is provided for reference only. No action is required.

ADES AIM Discovery works as follows:

1. After successful installation, the ADES AIM queries the Global Catalogs to retrieve all Active Directory and/or Exchange Server hosts for the configured domains. The ADES AIM also identifies the roles that are configured on each host. Hosts that have one or more supported roles are made available for management by DX NetOps Spectrum ADES Manager. The management entity setting is used to include only those hosts having a supported role for the configured technology (Active Directory, Exchange Server, or both).
2. Using the specified management mode, the ADES AIM sets all hosts as managed (domain-based management) or not managed (host-based management) by default.
3. The ADES AIM immediately begins polling all hosts that are set as managed to gather information about their respective server role function.
4. After the initial discovery, ADES AIM Discovery must be started manually to detect any subsequent changes to the environment. For more information, see Updating Your Managed Environment.

NOTE
The ADES AIM stores a list of available hosts and their managed settings. On a restart of the ADES AIM, this host list is read in. Any host that still resides in the Active Directory and Exchange Server environment retains its previous managed setting.

Run Discovery to Model the ADES Host Manager and Connecting Devices

NOTE
After you have set up the ADES Host Manager and ADES AIM Discovery has started, model the ADES Host Manager and any connecting devices. You can use standard DX NetOps Spectrum Discovery to:

- Model the ADES Host Manager, which must be modeled with a read/write community string.
- Model the necessary upstream routers and switches of your Active Directory and Exchange Server environments. Modeling of connecting devices allows connections from the Active Directory and Exchange Server hosts to be established later.

WARNING
Do not specify Active Directory and Exchange Server hosts to be modeled. Active Directory and Exchange Server hosts are discovered and modeled automatically during ADES Manager Discovery.

NOTE
An administrator performs this procedure.

Follow these steps:

1. Open the Discovery console.

NOTE
Prepare by gathering the correct community strings, IP addresses, and port numbers for any SNMP agents that run on a nonstandard port.

2. In the Navigation panel, click the **Creates a new configuration** icon  .
   The Configuration dialog opens.
3. Specify a name and location for the new configuration, and then click **OK**.
   The Configuration dialog closes.
4. Enter individual IP addresses or the beginning and ending IP addresses in the IP/Host Name Boundary List fields, and then click **Add**.

   NOTE
   Verify that the range of IP addresses includes all ADES Host Managers and the interconnecting switches and routers.
5. Configure SNMP Information.

   WARNING
   You must model the ADES Host Manager with a read/write community string. If you are modeling the ADES Host Manager in this Discovery, verify that its community string is placed appropriately in the ordered list. Alternatively, you can change the community string for the ADES Host Manager to its read/write value after the discovery.
6. Configure your Modeling Options as follows:
   a. Select **Discover and automatically model to Spectrum**.
   b. Click **Modeling Options**.
      The Modeling Configuration dialog opens.
   c. Click **Protocol Options**.
      The Protocol Options dialog opens.
   d. Select **ARP Tables for Pingables**, and then click **OK**.
      The Protocol Options dialog closes.
   e. Click **OK** to close the Modeling Configuration dialog.
7. (Optional) Click **Advanced Options** in the Advanced Options group, add any nonstandard SNMP ports, and then click **OK**.
   The Advanced Options dialog closes.
8. Enter any additional values that are needed in the Discovery console for modeling your connecting devices and ADES Host Managers, and then click **Discover**.
   Models are created and added to your network topology in DX NetOps Spectrum for the following entities:
   – Active Directory and Exchange Server Host Manager (ADES Host Manager).

     NOTE
     If the Discovery process did not assign the read/write community string to this model, update this setting manually. Use the Modeling Information subview for the model.
   – The upstream switches and routers that connect the hosts in your Active Directory and Exchange Server environment to your network.

   When these models exist in DX NetOps Spectrum, ADES Manager Discovery can begin.

   NOTE
   Instead of using standard DX NetOps Spectrum Discovery, you can manually model your ADES Host Manager by IP address or hostname. If you do, model the upstream devices first (because modeling the ADES Host Manager automatically triggers an ADES Manager Discovery). Modeling in the proper order allows the correct creation of connections in the topology between your hosts and the remainder of your network. For more information, see the Modeling and Managing Your IT Infrastructure section.

Upgrade the SystemEDGE Host Model (If Necessary)

If the SystemEDGE host model was created before loading the ADES AIM on the agent, the existing model is not compatible with ADES Manager. Upgrade the SystemEDGE host (Host_systemEDGE) model so that ADES Manager can access the ADES AIM capabilities in SystemEDGE.

To upgrade the SystemEDGE host model, right-click the model and select Reconfiguration, Reconfigure Model.

The SystemEDGE host model is upgraded to support the ADES AIM.

> **NOTE**
> You can also send a reconfigure model action to the SystemEDGE agent using CLI. For instructions on how to send a reconfigure model action to the SystemEDGE agent, see the Modeling and Managing Your IT Infrastructure section.

Specify Hosts to Manage by ADES AIM

Hosts must be set as "managed" to be modeled in DX NetOps Spectrum. When using domain-based management, all hosts are set initially to be managed; with host-based management, all hosts are set initially not to be managed. Use the Universal Host Table (UHT) view in DX NetOps Spectrum for specifying hosts to manage, as described in this section.

Using host-based management, you can distribute the management of hosts in a domain across multiple ADES AIMs for load balancing. The UHT contains all available hosts in all domains for which the ADES AIM is configured and lets you specify hosts for management. Manage a host by a single ADES AIM only.

> **WARNING**
> Managing a single host by multiple ADES AIMs can introduce performance issues. For more information, see the *CA Virtual Assurance for Infrastructure Managers Administration Guide*.

> **NOTE**
> These settings can also be set using MIB Tools or another MIB browser. However, it is recommended that you use the UHT view that is provided in DX NetOps Spectrum.

When a host is set to be managed, it appears in the Managed Host Table (MHT) view in DX NetOps Spectrum. The MHT contains the subset of UHT hosts for which the ADES AIM is polling for Active Directory and Exchange Server metrics. The MHT resides in the ADES AIM. ADES Manager uses the MHT as its basis for creating, deleting, or updating Active Directory and Exchange Server host models.

> **NOTE**
> Only those users with the appropriate privileges can modify host management. For more information, see the OneClick Administration section.

Follow these steps:

1. Select the ADES Host Manager model. The model has a device type of 'Active Directory and Exchange Server Host Manager.'

   > **NOTE**
   > The host must have been modeled with a read/write community string.

   The Component Detail panel displays information for the selected ADES Host Manager.
2. In the Information tab in the Component Detail panel, expand the Active Directory and Exchange Server (ADES) Management, Host Management, Universal Host Table subview.
   The Universal Host Table lists all hosts that are available for management by this ADES AIM. The table also indicates the server roles that are defined for each host.
3. Select one or more hosts in the table. Click Manage or Do Not Manage to specify whether this ADES AIM is going to manage the hosts.

NOTE
When working with many hosts, undock the Universal Host Table subview to enable expansion of the table view. You can also use shift-click and control-click to control your host selection, as follows:

- To select a consecutive group of hosts, click the first host, press and hold down the Shift key, and then click the last host.
- To select nonconsecutive hosts, press and hold down the Ctrl key, and then click each host that you want to select.

A checkmark is displayed in the Managed column for any host that this ADES AIM manages. Each managed host exists in the MHT, as displayed by the Managed Host Table view.

NOTE
For information about moving a host from the management of one ADES AIM to another, see Modifying ADES Manager Host Management and Models.

NOTE

For SystemEDGE to populate CPU related OIDs with data, run the following command from the command prompt to reload the Windows performance counters and restart SystemEDGE.

```
lodctr /r
```

Specify Hosts to Manage by ADES AIM

Hosts must be set as "managed" to be modeled in DX NetOps Spectrum. When using domain-based management, all hosts are set initially to be managed; with host-based management, all hosts are set initially not to be managed. Use the Universal Host Table (UHT) view in DX NetOps Spectrum for specifying hosts to manage, as described in this section.

Using host-based management, you can distribute the management of hosts in a domain across multiple ADES AIMs for load balancing. The UHT contains all available hosts in all domains for which the ADES AIM is configured and lets you specify hosts for management. Manage a host by a single ADES AIM only.

NOTE

Managing a single host by multiple ADES AIMs can introduce performance issues. For more information, see the *CA Virtual Assurance for Infrastructure* Managers documentation.

NOTE

These settings can also be set using MIB Tools or another MIB browser. However, it is recommended that you use the UHT view that is provided in DX NetOps Spectrum.

When a host is set to be managed, it appears in the Managed Host Table (MHT) view in DX NetOps Spectrum. The MHT contains the subset of UHT hosts for which the ADES AIM is polling for Active Directory and Exchange Server metrics. The MHT resides in the ADES AIM. ADES Manager uses the MHT as its basis for creating, deleting, or updating Active Directory and Exchange Server host models.

NOTE

Only those users with the appropriate privileges can modify host management. For more information, see the OneClick Administration section.

1. Select the ADES Host Manager model. The model has a device type of 'Active Directory and Exchange Server Host Manag**er.'**

   NOTE
   The host must have been modeled with a read/write community string.

The Component Detail panel displays information for the selected ADES Host Manager.

2. In the Information tab in the Component Detail panel, expand the Active Directory and Exchange Server (ADES) Management, Host Management, Universal Host Table subview.
   The Universal Host Table lists all hosts that are available for management by this ADES AIM. The table also indicates the server roles that are defined for each host.
3. Select one or more hosts in the table. Click Manage or Do Not Manage to specify whether this ADES AIM is going to manage the hosts.

   NOTE
   When working with many hosts, undock the Universal Host Table subview to enable expansion of the table view. You can also use shift-click and control-click to control your host selection, as follows:

   - To select a consecutive group of hosts, click the first host, press and hold down the Shift key, and then click the last host.
   - To select nonconsecutive hosts, press and hold down the Ctrl key, and then click each host that you want to select.

   A checkmark is displayed in the Managed column for any host that this ADES AIM manages. Each managed host exists in the MHT, as displayed by the Managed Host Table view.

NOTE
For information about moving a host from the management of one ADES AIM to another, see Modifying ADES Manager Host Management and Models.

ADES Manager Discovery

ADES Manager Discovery is the modeling within DX NetOps Spectrum of any required ADES Manager components. These components include any managed Active Directory and Exchange Server hosts. Modeling the ADES Host Manager automatically triggers an ADES Manager Discovery.

After the ADES Host Manager is modeled initially, anytime you modify a host in the Universal Host Table subview, ADES Manager Discovery runs automatically. ADES Manager Discovery also runs automatically whenever an ADES AIM Discovery runs and effects changes in the Managed Host Table.

The following description explains the entire ADES Manager Discovery process and is provided for reference. No action is required.

ADES Manager Discovery works as follows:

1. After DX NetOps Spectrum models a CA SystemEDGE host and detects the presence of the ADES AIM, the following actions occur:
   a. When the ADES AIM is detected, the device type of the CA SystemEDGE host changes to 'Active Directory and Exchange Server Host Manager'.
   b. An application model that can run ADES AIM Discovery is created.
   c. An ADES Managed Hosts container is created to contain any new Active Directory or Exchange Server host models for this ADES AIM.
2. When communication between DX NetOps Spectrum and the ADES AIM is established, the host-modeling process begins automatically. ADES Manager uses information from the MHT in the ADES AIM to determine which hosts to model. For each host to be modeled, if a model does not exist, a new host model is created. If an SNMP agent exists on the host, an SNMP-capable host model is created. Otherwise, an ICMP (Pingable) model is created.
3. New Active Directory or Exchange Server host models appear in the Active Directory and Exchange Server Manager hierarchy in the Explorer view. New host models are also placed into the ADES Managed Hosts container for the ADES AIM in the topology view. Connections to any upstream devices are made.

WARNING
If a host is already modeled in your DX NetOps Spectrum-managed network before ADES Manager discovery, it is not modeled again. Also, the original model is not moved in the topology. However, it is still included in the Active Directory and Exchange Server Manager hierarchy.

NOTE
You can also manually control whether a host is managed using the Universal Host Table. Modifying the UHT adds or deletes the host in the Managed Host Table, which triggers the modeling process. An ADES AIM Discovery can also add or remove host models from the MHT, which also triggers the modeling process.

How to Model Your Environment When Using Multiple AIM Solutions

Depending on your environment, you can use ADES Manager along with other DX NetOps Spectrum AIM solutions simultaneously to manage your network entities. Some configurations, such as the following examples, require using multiple solutions to manage your environment completely:

- An Active Directory or Exchange Server host runs on a virtual machine.
- The ADES AIM runs on a virtual machine.
- An Active Directory or Exchange Server host is a cluster node.

Each of DX NetOps Spectrum AIM solutions provide information that is specific to the technology it supports. For example:

- Virtual Host Manager provides details that are specific to virtual technologies.
- Cluster Manager provides details that are specific to cluster technologies.
- ADES Manager provides details that are specific to the supported Active Directory and Exchange Server roles.

The combination of these features provides a complete monitoring solution. To set up your implementation of multiple AIM solutions effectively, the following approach is recommended.

WARNING
When using multiple AIMs, only a single AIM can be installed on a given SystemEDGE host.

Follow these steps:

1. Configure the AutoDiscovery settings on the VNM model.
2. Configure the Virtual Host Manager settings that are related to your virtual technology.
3. Set up Virtual Host Manager by modeling the virtual technology manager and all virtual technology components.
4. Set up Cluster Manager by modeling the cluster technology manager and all cluster components.
5. Set up ADES Manager by modeling the ADES Host Manager and all Active Directory and Exchange Server hosts.

NOTE
For more information, see the Virtual Host Manager section and Cluster Manager section.

How to Model Your ADES Environment When Using Multiple AIM Solutions

Depending on your environment, you can use ADES Manager along with other DX NetOps Spectrum AIM solutions simultaneously to manage your network entities. Some configurations, such as the following examples, require using multiple solutions to manage your environment completely:

- An Active Directory or Exchange Server host runs on a virtual machine.
- The ADES AIM runs on a virtual machine.
- An Active Directory or Exchange Server host is a cluster node.

Each of the DX NetOps Spectrum AIM solutions provide information that is specific to the technology it supports. For example:

- Virtual Host Manager provides details that are specific to virtual technologies.
- Cluster Manager provides details that are specific to cluster technologies.
- ADES Manager provides details that are specific to the supported Active Directory and Exchange Server roles.

The combination of these features provides a complete monitoring solution. To set up your implementation of multiple AIM solutions effectively, the following approach is recommended.

> **WARNING**
> When using multiple AIMs, only a single AIM can be installed on a given SystemEDGE host.

Follow these steps:

1. Configure the AutoDiscovery settings on the VNM model.
2. Configure the Virtual Host Manager settings that are related to your virtual technology.
3. Set up Virtual Host Manager by modeling the virtual technology manager and all virtual technology components.
4. Set up Cluster Manager by modeling the cluster technology manager and all cluster components.
5. Set up ADES Manager by modeling the ADES Host Manager and all Active Directory and Exchange Server hosts.

> **NOTE**
> For more information, see the Virtual Host Manager section and the Cluster Manager section.

Models Created for ADES Manager

ADES Manager provides several models to represent the components of your Active Directory and Exchange Server environments. Understanding the following basic models can help you to understand better Discovery and how the models relate to one another.

- Active Directory and Exchange Server Host Manager (ADES Host Manager)

Model Type: Host_systemEDGE
Represents the host that contains the ADES AIM. The ADES AIM monitors the Active Directory and Exchange Server hosts in your environment. Successful creation of this model indicates that all requisite intelligence to support ADES Manager has been installed on the ADES Host Manager. This model has a device type of Active Directory and Exchange Server Host Manager.

- <*ADES_Host_Manager_name*> Managed Hosts

Model Type: ADESHostContainer
Initially contains newly created host models that the named ADES AIM manages. You can add or remove models from the container, but you cannot destroy the container itself. When possible, this container model is created alongside the ADES Host Manager model. If a managed host has already been modeled elsewhere in your DX NetOps Spectrum-managed network, it is not modeled again. Also, the existing model is not moved into this container. This behavior applies to a host that another ADES AIM manages. When multiple ADES AIMs manage a host in one landscape, the host model does not appear in multiple ADES Managed Hosts containers. Instead, the host appears in the ADES Managed Hosts container for the first AIM to model it.

NOTE

- If the ADES Host Manager is a virtual machine, the container is placed in the same topology as the Virtual Host Manager physical host container.
- Active Directory and Exchange Server hosts that are virtual machines and managed by Virtual Host Manager are not placed into the ADES Managed Hosts container.

- Active Directory and Exchange Server Host

Represents an Active Directory or Exchange Server host.

NOTE
When multiple ADES AIMs resides in different SpectroSERVERs of a multi landscape environment manage a single host, duplicate models exist. For more information, see Duplicate Models Created After Discovery.

For information about changing your modeled environment, including modifying host management by ADES AIM or deleting ADES Manager models, see Maintaining Your ADES Environment.

Viewing Your Active Directory and Exchange Server Environments

The purpose of ADES Manager is to provide visibility into your Active Directory and Exchange Server environments. This visibility allows you to identify easily the function or role each host plays as well as the logical relationships between the players. Most importantly, when a problem occurs in your environment you can pinpoint its cause.

ADES Manager provides several methods for viewing your Active Directory and Exchange Server environments, as follows:

- The Active Directory and Exchange Server Manager hierarchy in the Explorer tab of the Navigation panel shows the logical relationships between entities. Examples of hierarchy nodes include forests, domains, and roles.
- A graphical topology view helps you group like-managed Active Directory and Exchange Server hosts together as well as visualize the connections between the hosts.
- Customized Information views in the Component Detail panel provide details that are specific to Active Directory and Exchange Server technologies.
- Customized searches provide a quick way to find hosts by using Active Directory and Exchange Server metrics.
- Customized icons for individual models provide status and model type information at a glance and are integrated throughout the ADES Manager feature.

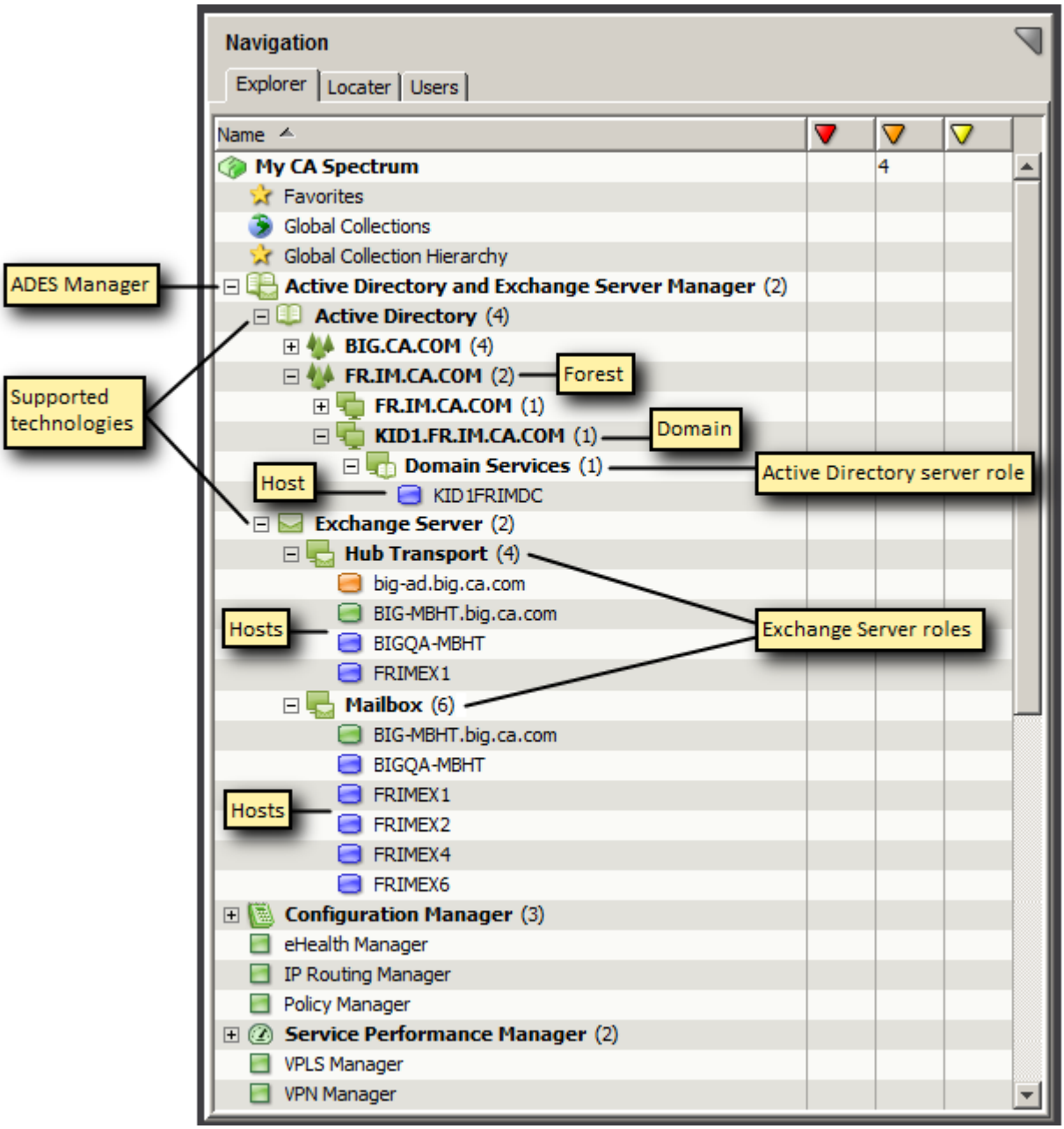
Understanding each of these methods helps you easily monitor your Active Directory and Exchange Server environments. Efficient monitoring then lets you troubleshoot issues and optimize performance more effectively.

Explorer View

On the Explorer tab of the Navigation panel, the Active Directory and Exchange Server Manager node provides a hierarchical tree structure. This format helps you visualize the logical organization of your managed Active Directory and Exchange Server environments. Distinctive icons distinguish the technologies, forests, domains, hosts, and roles that are used in your managed environment.

Using this information, you can see how the hosts are logically arranged as well as which functions or roles each host in your environment supports.

The following image is an example of the ADES Manager hierarchy:



The following nodes form the hierarchy:

- Active Directory and Exchange Server

  Manager
  Active Directory and Exchange Server Manager is the root node for the Active Directory and Exchange Server environments that are currently managed. ADES Manager is a distributed manager that handles multiple landscapes. For this reason, the node appears above the landscape level.
  Expanding the Active Directory and Exchange Server Manager node displays the technologies that are supported, as follows.

NOTE
Only technology, forest, domain, or role nodes that contain at least one managed host in their subhierarchy are displayed.

Topology View

The models for your managed Active Directory and Exchange Server environment are organized and integrated into the Universe topology view. These models include the ADES Host Manager (SystemEDGE host) and the Active Directory and Exchange Server host models. This graphical representation helps you visualize the structure of your managed environment, including connections between managed hosts and other elements of your network.

The host models that are created during ADES Manager Discovery are placed in an ADES Managed Hosts container specific to the ADES AIM. When possible, this container model is created alongside the corresponding ADES Host Manager model.

NOTE
The ADES Host Manager can be a virtual machine. If so, the ADES Managed Hosts container is created in the same topology as the Virtual Host Manager physical host container.

If a managed host has already been modeled in your DX NetOps Spectrum-managed network before ADES Manager Discovery, it is not modeled again. And, the original model is not moved into the ADES Managed Hosts container during ADES Manager Discovery. This behavior applies to a host that another ADES AIM within the same landscape is managing. When multiple ADES AIMs manage a host in one landscape, the host model does not appear in multiple ADES Managed Host containers. Instead, the host appears in the ADES Managed Hosts container for the first AIM to model it.

The following rules apply to the ADES Managed Hosts containers:

- You can move Active Directory and Exchange Server host models out of the ADES Managed Hosts container. Additionally, this container is not limited to Active Directory and Exchange Server hosts. In certain cases, you can move other host models into the ADES Managed Hosts container manually if the administrator considers the inclusion to be logical.
- You cannot destroy an ADES Managed Hosts container directly. The ADES Managed Hosts container is destroyed only when an ADES Host Manager model is deleted. When the container is destroyed, all models in the container are moved to the Lost and Found (LostFound). This action includes any models that are moved manually into the contains and are not necessarily Active Directory or Exchange Server hosts

  NOTE
  An exception is when a host model is in a global collection; in this case, the model remains in the global collection.

  WARNING
  The contents of an ADES Managed Hosts container are not necessarily a comprehensive reflection of all hosts that the corresponding ADES AIM manages. If a host is already modeled in the landscape, the host is not modeled again. And, the existing model is not moved into the ADES Managed Hosts container.

Placement of Models

The placement of host models in the topology during discovery occurs as follows:

- If ADES Manager discovery creates the model, the model is placed in an ADES Managed Hosts container.
- If the Active Directory and Exchange Server host is a virtual machine that Virtual Host Manager manages, the model remains in the physical host container. The host model does not appear in the ADES Managed Hosts container.

WARNING
A virtual machine cannot be moved from its physical host container. An Active Directory or Exchange Server host that is a virtual machine is not reflected in the ADES Managed Hosts container.

- If the model exists and is for a physical machine that another AIM solution manages, the model is not moved from its current container. The model is not represented in the ADES Managed Hosts container. An example of another AIM solution is Cluster Manager.
- If the model exists and no other AIM solution manages the model, the model is moved to the ADES Managed Hosts container.

Information Subviews

Customized views in the Component Detail panel provide detailed information about the components in your Active Directory and Exchange Server environments. You can view information specific to your managed Active Directory and Exchange Server environments by ADES Host Manager (ADES AIM) Subviews and Individual Host.

ADES Host Manager (ADES AIM) Subviews

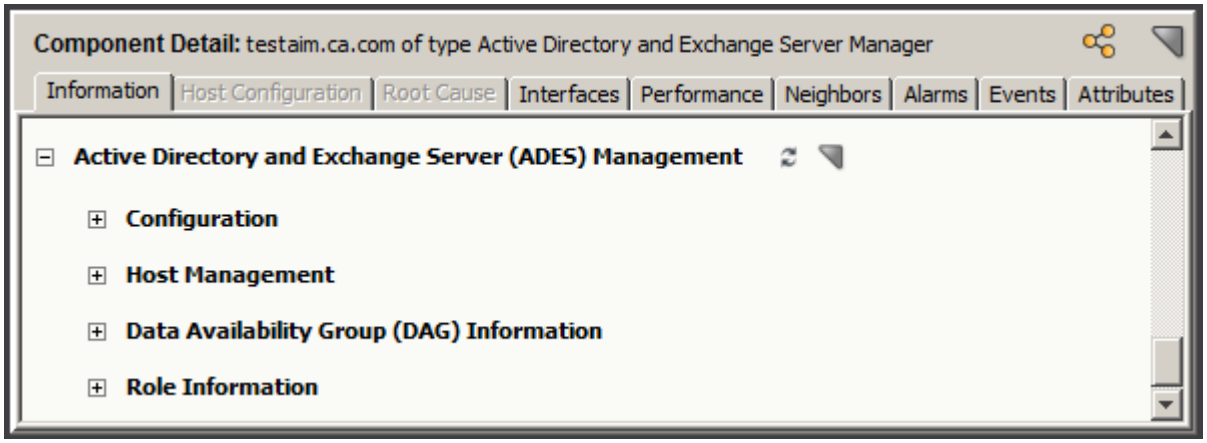
Using views that are provided at the ADES Host Manager (ADES AIM) level, you can view the following information:

- Information specific to the ADES Host Manager such as agent version, polling interval, and ADES AIM Discovery status. You can also control certain options in the ADES AIM from these views, including polling interval and initiation of an ADES AIM Discovery.
- Information about hosts that are available for management and that the ADES AIM is actually managing. You can also control which hosts to manage by the ADES AIM.
- Consolidated information about all hosts in your Active Directory and Exchange Server environments that this single ADES AIM is managing. Views at the ADES AIM level combine information from all hosts that the AIM is managing. For example, you can see collectively all hosts that serve a supported role or a list of domain controllers. From this perspective, you can also see a list of all mailbox databases that reside on hosts that an ADES AIM is managing.

The following procedure describes how to view information for an ADES Host Manager (ADES AIM).

Follow these steps:

1. Select the ADES Host Manager model. The model has a device type of 'Active Directory and Exchange Server Host Manager.'
   The Component Detail panel displays information for the selected ADES Host Manager.
2. In the Information tab in the Component Detail panel, expand the Active Directory and Exchange Server (ADES) Management subview.
   The expanded subview appears, as follows:



Active Directory and Exchange Server (ADES) Management Subviews - ADES Host Manager (ADES AIM)

The following subviews are available at the ADES Host Manager (ADES AIM) level.

> **NOTE**
> When viewing host information using the ADES Host Manager subviews, the Host Name that appears reflects the name as provided by the ADES AIM. This value can differ from the model name that DX NetOps Spectrum chose and which appears in the hierarchy and topology views. For more information on how DX NetOps Spectrum assigns model names for ADES Manager, see Model Naming.

- **Configuration**
  Provides information specific to the ADES Host Manager, including:
  - Agent identification, version, and status details.
  - ADES AIM polling interval and Windows Event settings. This view also lets you modify these settings, as described in Updating ADES AIM Configuration Options.
  - Date and time when the ADES AIM inventory was last updated.
  - Date and time when the last ADES AIM Discovery was performed. This view also lets you initiate a new ADES AIM Discovery, as described in Updating Your Managed Environment.
- **Host Management**
  Provides information about available hosts and whether this ADES AIM is managing them currently.
  - **Universal Host Table**
    Lists all hosts that are available for management by this ADES AIM and the server roles that are defined for each host. This table also lets you control which hosts are managed.
    For more information about the Universal Host Table, see Specify Hosts to Manage by ADES AIM.
  - **Managed Host Table**
    Lists all hosts that this ADES AIM is managing. The read-only Managed Host Table also provides AIM-specific polling and status information for each host.

    > **NOTE**
    > The Universal Host Table lists all hosts that this AIM *can* manage. The Managed Host Table lists those hosts that this AIM is *actually* managing.

    For more information about the Managed Host Table, see Specify Hosts to Manage by ADES AIM.
- **Data Availability Group (DAG) Information**
  Provides read-only information about the DAGs that any of the Mailbox hosts that this ADES AIM is managing are members of. Various subviews are provided, each presenting DAG-related information from a different perspective, including:
  - General DAG configuration details such as originating and witness server information, compression and encryption that are being used, and other replication details.
  - Database information including active or passive copy status, issued error messages, latest snapshots, and other backup details.
  - Organizational information for DAG hosts and networks.
- **Role Information**
  Provides read-only information specific to each of the supported server roles. For the Domain Services role, all domain controllers are included in the views whether managed or not. For other roles, only hosts that this ADES AIM is managing are included.

  > **NOTE**
  > A subview table contains data only when at least one host satisfying the stated criteria exists. Otherwise, the table is empty.

Individual Host Subviews

You can view information for individual hosts in your managed Active Directory and Exchange Server environments. Views for individual hosts provide information that is specific to all the server roles that the host provides. Examples include real-time LDAP activity metrics for Active Directory hosts or DAG information for Mailbox hosts.

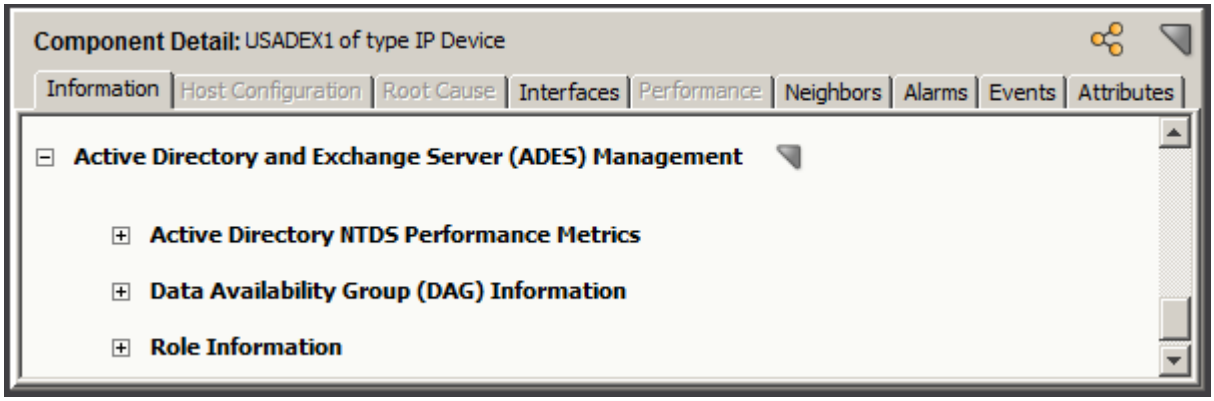
The following procedure describes how to view information for an individual host that ADES Manager is monitoring.

Follow these steps:

1. Select the model for a host that ADES Manager is managing. The model exists under the Active Directory and Exchange Server Manager hierarchy in the Explorer tab of the Navigation panel.

   NOTE
   You can also select the host using any of the standard DX NetOps Spectrum methods. These methods include from a list (for example, in the Contents panel, List tab) or in the topology.

   The Component Detail panel displays information for the selected Active Directory or Exchange Server host model.
2. In the Information tab in the Component Detail panel, expand the Active Directory and Exchange Server (ADES) Management subview.
   The expanded subview appears, as follows:



Active Directory and Exchange Server (ADES) Management Subviews - Individual Hosts

The following subviews are available for individual hosts.

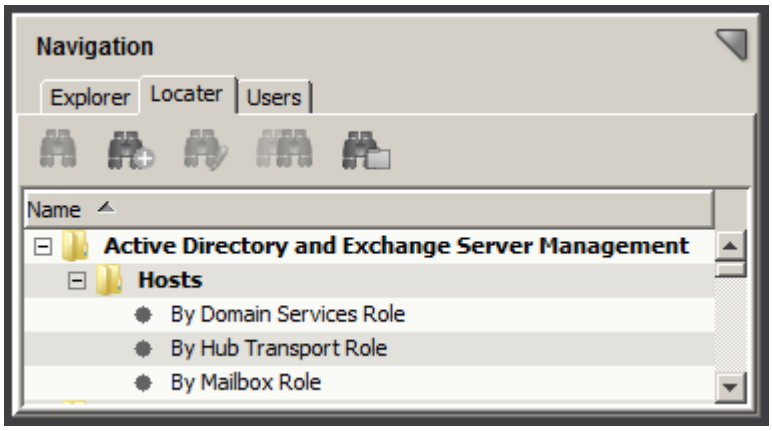
NOTE
When viewing host information using the individual host subviews, the referenced host names that appear reflect the names as provided by the ADES AIM. These values can differ from the model names DX NetOps Spectrum chose and which appear in the hierarchy and topology views. For more information on how DX NetOps Spectrum assigns model names for ADES Manager, see Model Naming.

- **Active Directory NTDS Performance Metrics**
  (For Domain Services hosts) Provides performance data on activities such as LDAP operations, object updates and replication, and authentication requests.
- **Data Availability Group (DAG) Information**
  (For Mailbox hosts) Provides read-only information about the DAG that the selected Mailbox host is a member of. Information includes active or passive status, data center activation mode, and other DAG-related details.
- **Role Information**
  Provides read-only information specific to each of the supported server roles, as follows.

  NOTE
  A subview table contains data only when the host supports the respective role.

Locater Searches

You can use preconfigured searches to locate entities in the DX NetOps Spectrum database that is related to the Active Directory and Exchange Server technologies quickly. The searches are grouped under the Active Directory and Exchange Server Management folder in the Locater tab of the Navigation panel, as shown:



The following searches are specific to Active Directory and Exchange Server technologies:

- **Hosts**
  Locates all hosts that are modeled in the DX NetOps Spectrum database that have been identified as serving the specified role in one of the following searches:
  – By Domain Services Role
  – By Hub Transport Role
  – By Mailbox Role

  NOTE
  Only those users with the appropriate privileges can access Active Directory and Exchange Server Management searches. For more information, see the OneClick Administration section.

Event Reports

To help you monitor your Active Directory and Exchange Server environments, you can create event reports. Event reports gather the information that helps you make informed decisions about your Active Directory and Exchange Server environment entities. Using the correct event filters, you can base these reports on any of the various events that are generated for your managed environment in DX NetOps Spectrum.

To report on ADES Manager events, the `ADES-events-filter.xml` event filter file is included with Report Manager:

NOTE
You can use the event codes of the .xml file, to generate event reports in Spectrum Report Manager. For more information, see the Report Manager section. You can also generate reports using the predefined event filter files. For more information, see the Install Report Manager section.Spectrum Report Manager

Maintaining Your ADES Environment

This section discusses the ways in which you can maintain your ADES Environments.

Controlling Polling Intervals to Update ADES Data

Polling intervals control how often information is obtained from managed devices. To keep data for your managed Active Directory and Exchange Server environments current, DX NetOps Spectrum ADES Manager uses polling intervals set on the following components:

- **ADES AIM**

The ADES AIM polling interval indicates how often the ADES AIM queries the managed hosts for information. The ADES AIM polling interval exists in the ADES AIM but can be modified from within DX NetOps Spectrum. The default value is 300 seconds with a minimum value of 30 seconds. The interval must be a multiple of 30 seconds. For information on how to update the ADES AIM polling interval from within DX NetOps Spectrum, see Updating ADES AIM Configuration Options.

- **ADES Host Manager model**
  The polling interval on the ADES Host Manager model determines how often DX NetOps Spectrum polls the ADES AIM. The default value is 300 seconds with a minimum value of 30 seconds. This setting is available on the DX NetOps Spectrum Modeling Information view for the Active Directory and Exchange Server Host Manager (Host_systemEDGE) model.

Updating ADES AIM Configuration Options

ADES Manager uses the ADES AIM for discovery, modeling, and monitoring of your Active Directory and Exchange Server environments. The ADES AIM provides and controls certain functionality exclusively. Some ADES AIM settings can only be changed directly in the AIM (such as what domains are managed). Other settings can be changed from within DX NetOps Spectrum (such as the ADES AIM polling interval). This section describes how to change certain ADES AIM settings from within DX NetOps Spectrum.

NOTE
For detailed information about all ADES AIM parameters, see the CA Virtual Assurance for Infrastructure Managers section.

Follow these steps:

1. Select the ADES Host Manager model that supports the ADES AIM. The model has a device type of 'Active Directory and Exchange Server Host Manager.'
   The Component Detail panel displays information for the selected ADES Host Manager.
2. In the Information tab in the Component Detail panel, expand the Active Directory and Exchange Server (ADES) Management, Configuration subview.
   The expanded Configuration subview appears.
3. Update ADES AIM polling and Windows Event parameters as appropriate.

Modifying ADES Manager Host Management and Models

When changing your modeled environment, consider the following points:

- When multiple ADES AIMs manage a host in one landscape, the host model does not appear in multiple ADES Managed Hosts containers. Instead, the host appears in the ADES Managed Hosts container for the first AIM that modeled it only. If the first ADES AIM no longer manages the host, the host remains in its original container even though another ADES AIM is managing it. To move the host to the ADES Managed Hosts container for another ADES AIM (or another location in the topology), manually move the model.

  NOTE
  Use cut-and-paste to move a model manually.
- When an ADES AIM no longer manages a host, the host is moved from the ADES Managed Hosts container to the Lost and Found (LostFound). The following cases are exceptions:
  - Another ADES AIM is managing the host.
  - The host is in a global collection.
- When the IP or MAC address for a host model that ADES Manager manages is modified, connections to any connecting devices are automatically updated.

Deleting ADES Manager Models

Consider the following regarding deleting models in your DX NetOps Spectrum modeled environment:

- When ADES Manager no longer manages a host, the host model is removed from the ADES Managed Hosts container.

  NOTE
  Host models that are removed from ADES Manager management are moved to the Lost and Found (LostFound). An exception is when a host is in a global collection, in which case, the model remains in the global collection.
- You can delete host models from your DX NetOps Spectrum modeled environment; however, you cannot manually delete forests, domains, and roles. ADES Manager Discovery automatically deletes these models when they are no longer needed.
- If you delete a host model that an ADES AIM is managing, a new host model is created on the next ADES Manager Discovery. Any customizations made to the original model are lost.
- Hosts that both Virtual Host Manager and ADES Manager manage adhere to all the standard modeling behaviors of virtual machines. These models cannot be deleted from the topology.
- When an ADES Host Manager model is deleted, the corresponding ADES Managed Hosts container is destroyed. All models in the container (except for models in a global collection) are moved to the Lost and Found (LostFound). Any models that are moved manually into the container and are not necessarily Active Directory or Exchange Server hosts are also moved.
- The ADES Manager hierarchy synchronizes after the Lost and Found (LostFound) is emptied.

Deleting Models When Using Multiple AIM Solutions

If you use ADES Manager along with other DX NetOps Spectrum AIM solutions, consider the following points when deleting models in your environment:

- If you plan to no longer manage the host models using Virtual Host Manager, configure Virtual Host Manager delete settings to retain models. Otherwise, Virtual Host Manager deletes the host model initially, losing any history or customization. ADES Manager then recreates the host model during the next ADES Manager Discovery.

  NOTE
  The Virtual Host Manager setting to retain models when the technology manager is deleted applies to SNMP-enabled device models only. For ICMP (Pingable) models, Virtual Host Manager deletes the model, and then ADES Manager recreates the model.
- When Virtual Host Manager unmanages a host and the model is retained, ADES Manager automatically pulls the model into its management.
- If another solution no longer manages a host, the rules of the remaining solutions are reapplied in the ranked order. Typically, any changes are made at the next polling cycle.
- When a higher ranking AIM solution no longer manages a host, the host model is removed from the respective solution containers. (Examples of higher ranking solutions are Virtual Host Manager or Cluster Manager.) If ADES Manager continues to manage the host, the model does not appear in the ADES Managed Hosts container automatically. To move the model into the ADES Managed Hosts container, cut and paste the model from the Lost and Found (or global collection, if applicable).

Updating Your Managed Environment

After the initial, automatic modeling of the Active Directory and Exchange Server environments, initiate ADES AIM Discovery manually to update DX NetOps Spectrum with any subsequent changes. Due to the potentially large size of the Active Directory and Exchange Server environments, dynamic updates would be expensive for the ADES AIM to

maintain. After ADES AIM Discovery completes, ADES Manager Discovery begins automatically, modeling changes in the environment as detected by the ADES AIM.

Discovery Process for Updating Your Modeled Environment

Discovery for ADES Manager consists of two phases:

1. Discovery by the ADES AIM identifies changes to your Active Directory and Exchange Server environment as reflected in the Global Catalogs. These hosts are made available for management in the ADES AIM.
2. Discovery by ADES Manager. This process uses the environment information that the ADES AIM gathers and the settings that the ADES AIM defines to update the DX NetOps Spectrum modeled environment. For a new host detected in a Global Catalog, the ADES AIM determines if the host is available to manage. This determination is based on the management entity for the domain. If the host can be managed, the ADES AIM then examines the management mode for its domain. Based on the management mode, the following behavior occurs:
   - In domain-based management, the host is set automatically to be managed. DX NetOps Spectrum models and begins monitoring the new host automatically.
   - In host-based management, the host is set automatically not to be managed. Specify explicitly to manage the host using the Universal Host Table so that DX NetOps Spectrum models and begins monitoring the new host.

     NOTE
     The determination of whether a host is managed or not managed applies to *newly detected* hosts only. A host that was previously detected in a Global Catalog and identified by the ADES AIM as available to manage retains its previous managed setting.

The next section describes how to update your DX NetOps Spectrum modeled environment.

How to Update Your Modeled Environment

The following steps describe how to update your modeled environment, which begins with initiating an ADES AIM Discovery.

NOTE
Only those users with the appropriate privileges can initiate ADES AIM Discovery. For more information, see the OneClick Administration section.

Follow these steps:

1. Select the ADES Host Manager model that supports the ADES AIM. The model has a device type of 'Active Directory and Exchange Server Host Manager.'
   The Component Detail panel displays information for the selected ADES Host Manager.
2. In the Information tab in the Component Detail panel, expand the Active Directory and Exchange Server (ADES) Management, Configuration subview.
   The expanded Configuration subview appears and displays information about when the last ADES AIM Discovery was run.
   - **Last Agent Inventory Update**
     Indicates the last time that an ADES AIM Discovery was performed or the last time a change was made to the ADES AIM inventory. The ADES AIM inventory is used for DX NetOps Spectrum modeling. Editing host management in the Universal Host Table is a direct way to alter the ADES AIM inventory.
   - **Last ADES AIM Discovery of Hosts**
     Indicates the last time that an ADES AIM Discovery was performed.
3. Click Run.

   WARNING
   Running an ADES AIM Discovery can require significant system resources on the ADES Host Manager.

The ADES AIM Discovery begins. Any change that is detected in your Active Directory and Exchange Server environment causes the following actions:

- Updates the forest, domain, and host information in the ADES AIM inventory.
- Updates the 'Last Agent Inventory Update' and 'Last ADES AIM Discovery of Hosts' timestamp values.
- Triggers an ADES Manager Discovery in DX NetOps Spectrum so that your modeled environment reflects any changes.

NOTE
If you are using host-based management and new hosts are introduced, manually specify to manage the hosts. See Specify Hosts to Manage by ADES AIM.

Alarms and Fault Management

Knowing about certain activities, such as a DAG failover, can minimize potential problems in your Active Directory and Exchange Server environment. To alert you, DX NetOps Spectrum generates alarms and uses advanced fault management techniques to isolate the root cause.

Problems with a single device can cause several other devices in your network to generate events. Deciding which devices are the root cause of an alarm can be challenging. For example, when you lose contact with the ADES Host Manager (the proxy manager), you also lose proxy communication with the hosts that it manages. As a result, alarms are generated for the ADES Host Manager and each of its managed hosts. Sifting through potentially hundreds of simultaneously produced alarms manually to pinpoint the problem could be a tedious and error-prone process. Using fault isolation techniques, ADES Manager significantly simplifies the troubleshooting process by automatically correlating these alarms to identify a single root cause. As a result, you can identify and correct the problem more quickly.

ADES Manager evaluates which devices are issuing alarms and the type of events that devices generate. DX NetOps Spectrum uses all available information to correlate the alarms to the appropriate root cause, only alarming on the isolated faulty device. ADES Manager relies on the combination of standard DX NetOps Spectrum monitoring, proxy management, and traps to create meaningful events and alarms.

ADES Manager Alarms

To alert you to problems within your monitored Active Directory and Exchange Server environments, ADES Manager generates alarms for the following conditions:

- Active Directory or Exchange Server proxy communication lost, which indicates that updated Active Directory or Exchange Server metrics can no longer be obtained.
- ADES Host Manager (ADES AIM) down or communication lost

Traps

DX NetOps Spectrum supports many of the traps that the ADES AIM generates. Trap activity generates an event in DX NetOps Spectrum and is reported initially on the ADES Host Manager model. Some events are then forwarded to a corresponding entity type (that is, the "destination" entity), depending on the type of trap.

The following table provides the traps and destination entity type and indicates whether the trap generates an alarm by default.

Trap Name	Trap OID	Alarm?	Destination Entity
exchAdForestAddedTrap	1.3.6.1.4.1.546.1.1.0.166100	No	ADES Host Manager

exchAdForestRemovedTrap	1.3.6.1.4.1.546.1.1.0.166101	No	ADES Host Manager
exchAdDomainAddedTrap	1.3.6.1.4.1.546.1.1.0.166102	No	ADES Host Manager
exchAdDomainRemovedTrap	1.3.6.1.4.1.546.1.1.0.166103	No	ADES Host Manager
exchAdManagedHostAddedTrap	1.3.6.1.4.1.546.1.1.0.166104	No	ADES Host Manager
exchAdManagedHostRemovedTrap	1.3.6.1.4.1.546.1.1.0.166105	No	ADES Host Manager
exchAdManagedHostPollStatusChangedTrap	1.3.6.1.4.1.546.1.1.0.166106	No*	ADES Host Manager
exchAdDomainControllerAddedTrap	1.3.6.1.4.1.546.1.1.0.166107	No	ADES Host Manager
exchAdDomainControllerRemovedTrap	1.3.6.1.4.1.546.1.1.0.166108	No	ADES Host Manager
exchAdDagAddedTrap	1.3.6.1.4.1.546.1.1.0.166109	No	ADES Host Manager
exchAdDagRemovedTrap	1.3.6.1.4.1.546.1.1.0.166110	No	ADES Host Manager
exchAdHostAddedToDagTrap	1.3.6.1.4.1.546.1.1.0.166111	No	ADES Host Manager
exchAdHostRemovedFromDagTrap	1.3.6.1.4.1.546.1.1.0.166112	No	ADES Host Manager
exchAdDagFailOverTrap	1.3.6.1.4.1.546.1.1.0.166117	Yes	ADES Host

* The polling intelligence handles the alarm generation.

NOTE
For more information on ADES AIM traps, use MIB Tools to view the empireExchangeAdMIB MIB. For more information, see the Modeling and Managing Your IT Infrastructure section.

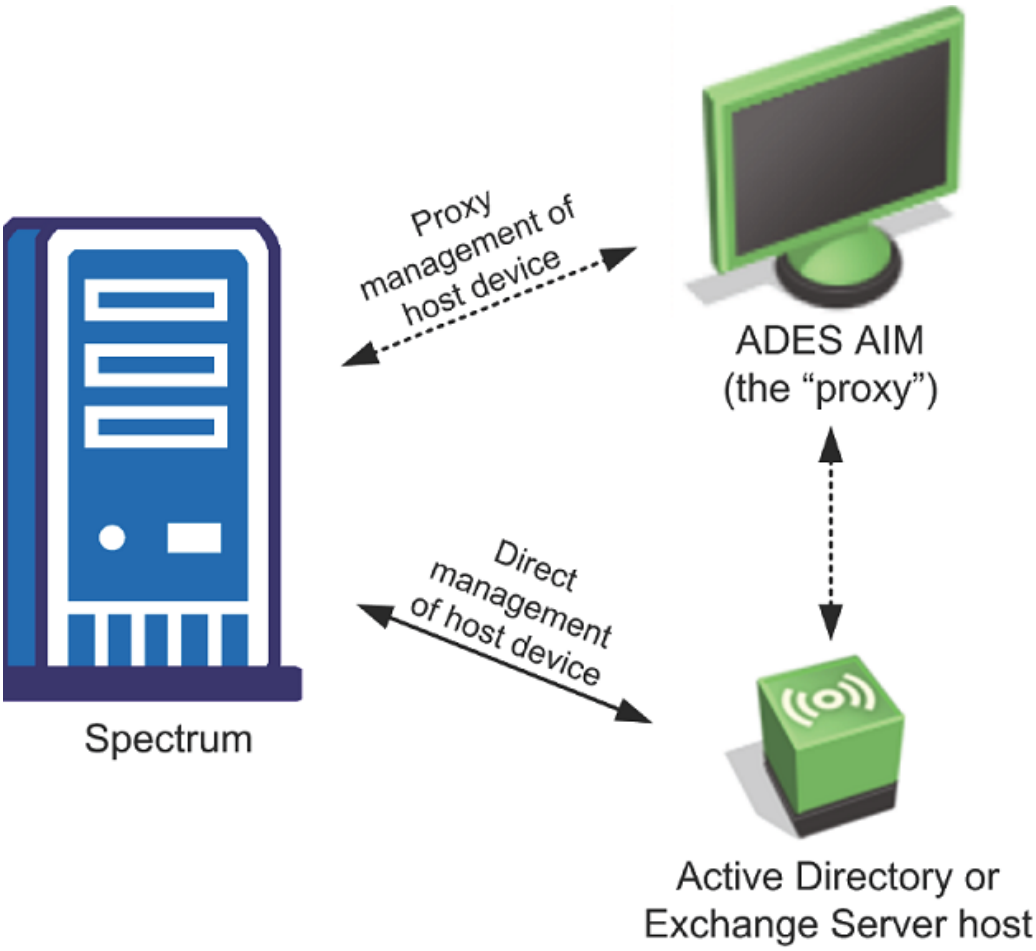
Proxy Management

Hosts that the ADES AIM manages provide a unique management opportunity. They provide DX NetOps Spectrum an alternate management perspective in addition to standard device monitoring methods.

Proxy management is the management of a device by way of an alternate source (such as the AIM) rather than from the device itself. The ADES AIM serves as a "proxy" from which DX NetOps Spectrum gathers information specific to the Active Directory and Exchange Server technologies.

Using standard monitoring, DX NetOps Spectrum gathers information directly from a device. Using proxy management, DX NetOps Spectrum also simultaneously gathers Active Directory and Exchange Server metrics for your managed hosts from the ADES AIM. These metrics include information that is specific to each server role, which cannot be obtained through standard monitoring.

NOTE
The ADES AIM is not a Proxy Model as defined by DX NetOps Spectrum Modeling Information. For information about models that are designated as a Proxy Model, see the Modeling and Managing Your IT Infrastructure section.



Standard DX NetOps Spectrum fault management handles this dual management in two ways:

- **Proxy management alarms.** By using the ADES AIM for management, proxy-related alarms can be generated. These alarms are unique because they alert you when the acquisition of Active Directory or Exchange Server metrics through the proxy is affected, not the state of the device or direct (SNMP) management. Knowing when contact through the proxy is lost is important because you could be missing important server role information about your environment. Proxy management alarms are of major severity and are not clearable by the user.
  - **Proxy unavailable** - When DX NetOps Spectrum cannot communicate with the ADES AIM, a proxy unavailable alarm is generated on the ADES Host Manager model.
    The following text is used for the proxy unavailable alarm:
    ```
    ACTIVE DIRECTORY AND EXCHANGE SERVER MANAGER UNAVAILABLE
    ```
  - **Proxy lost** - When DX NetOps Spectrum cannot obtain information about the managed host by way of the proxy, then a proxy lost alarm is generated. For example, when DX NetOps Spectrum cannot communicate with the ADES Host Manager model, DX NetOps Spectrum generates a proxy unavailable alarm. DX NetOps Spectrum also generates a proxy lost alarm for each of the hosts that the ADES Host Manager manages. As another example, DX NetOps Spectrum also generates a proxy lost alarm when the ADES AIM cannot successfully communicate with a managed host.
  - The following text is used for the proxy lost alarm:
    ```
    ACTIVE DIRECTORY AND EXCHANGE SERVER HOST PROXY LOST
    ```
    A proxy lost alarm is generated only for hosts that ADES Manager manages. If the host is removed from management by the proxy, the respective proxy management alarms are cleared. If more than one ADES AIM

manages a host, multiple proxy lost alarms are generated for the managed host, one per ADES Host Manager model.

NOTE
Manage a host by a single ADES AIM only.

- **Enhanced contact lost alarms.** DX NetOps Spectrum alarms that indicate loss of contact with the ADES Host Manager contain added correlation of ADES Manager proxy management alarms. These proxy management alarms indicate loss of Active Directory and Exchange Server metrics acquisition.
  The following text is used for the contact lost alarm:
  ```
  DEVICE HAS STOPPED RESPONDING TO POLLS
  ```

Alarm Correlation

Using standard DX NetOps Spectrum fault management and proxy information, ADES Manager automatically correlates the alarms to identify a single root cause. Proxy unavailable alarms on an ADES Host Manager model are correlated to any of the following alarms on the model:

- Contact lost
- Maintenance
- Hibernation
- Management lost

Also, the proxy lost alarms, which are generated upon creation of a proxy unavailable alarm, are correlated to the proxy unavailable alarm.

NOTE
You can find out more about alarms that are correlated or symptomatic by using the Impact tab for the alarm.

The scenarios in the next section provide details about generated alarms, what they indicate, and how they are correlated.

ADES Manager Fault Management Scenarios

The following scenarios describe how ADES Manager attempts to determine the root cause in different fault management situations.

Scenario: Contact between DX NetOps Spectrum and ADES AIM is lost

When DX NetOps Spectrum cannot communicate with ADES AIM, DX NetOps Spectrum loses information about the Active Directory and Exchange Server environment the AIM manages. To isolate the problem, ADES Manager determines the root cause as follows:

1. When DX NetOps Spectrum cannot communicate with the ADES AIM, a proxy unavailable alarm is generated on the ADES Host Manager model. For example, DX NetOps Spectrum cannot communicate with a device when the device is in maintenance or hibernation mode. The reason for the unavailability is identified through the event producing the alarm.

   NOTE
   Proxy management alarms are generated even when their models are in maintenance mode. These alarms are not visible in the Alarm view by default.
2. DX NetOps Spectrum cannot obtain information from the ADES AIM about any of the hosts it manages. As a result, proxy lost alarms are generated for each of the hosts that this AIM manages.
3. The proxy lost alarms are correlated to the proxy unavailable alarm as symptoms, which indicate that the proxy unavailable alarm is the root cause.
4. If DX NetOps Spectrum cannot communicate with the ADES Host Manager, one of the following standard DX NetOps Spectrum communication impairment alarms is also generated:

- Contact lost
- Maintenance
- Hibernation
- Management lost

5. Finally, the proxy unavailable alarm is correlated to the standard communication alarm forming a three-tiered hierarchy of alarm correlation: communication - proxy unavailable - proxy lost. Only the top-level communication alarm indicating the reason why DX NetOps Spectrum cannot communicate with the ADES AIM is visible to the user. This alarm indicates the root cause.

Scenario: Contact between ADES AIM and Active Directory or Exchange Server host is lost

If the ADES AIM loses contact with a managed host, it can no longer obtain Active Directory and Exchange Server metrics for that host. This situation creates outdated or "stale" values in the ADES AIM. As a result, a proxy lost alarm is generated for the host.

Scenario: Active Directory or Exchange Server host is down

When an Active Directory or Exchange Server host is down, DX NetOps Spectrum uses information from both the proxy and directly from the host. DX NetOps Spectrum handles this scenario as follows:

1. Because the management of the device by the ADES AIM is lost, a proxy lost alarm is generated.

   NOTE
   Manage a host by a single ADES AIM only. If, inadvertently, multiple ADES AIMs manage the host, multiple proxy lost alarms are generated. Proxy alarms are generated for each managing ADES Host Manager model.
2. Through direct or standard DX NetOps Spectrum fault management, a contact lost alarm is generated for the managed host.

Both the contact lost alarm and proxy lost alarm is visible to the user. The appearance of both alarms indicates that the proxy and direct methods of communication to the host are impacted.

Troubleshooting ADES Manager

This section describes some of the most common problems that occur with ADES Manager.

SystemEDGE Host Not Modeled as ADES Host Manager

Symptom:

After I model the SystemEDGE host with the ADES AIM installed, the created model is not configured as an ADES Host Manager. I am using the ADES AIM r12.7 with the latest PTFs installed (or later).

Solution:

The ADES AIM must complete its initialization and must have a ready status before DX NetOps Spectrum recognizes it as an ADES Host Manager. If you model the SystemEDGE host after the agent starts but before the ADES AIM finishes loading, the host is not modeled correctly.

To convert the existing SystemEDGE host model to an ADES Host Manager, perform the following procedure.

Follow these steps:

1. Wait for the ADES AIM to complete its initialization and reflect a ready status.
   You can use MIB Tools or another MIB browser to query the SystemEDGE host and view the exchAdAgentStatus variable in the empireExchangeAdMIB MIB. A value of "ready" indicates that initialization is complete.

NOTE
For more information on MIB Tools, see Certifications section.

2. Right-click the SystemEDGE host (Host_systemEDGE) model and select Reconfiguration, Reconfigure Model.
   The SystemEDGE host model is reconfigured as an ADES Host Manager.

Duplicate Models Created After Discovery

Symptom:

After ADES Manager Discovery runs, I see duplicate models in the ADES Manager hierarchy for some of the hosts in my modeled environment.

Solution:

Manage a host by a single ADES Host Manager. If, inadvertently, multiple ADES Host Managers residing in different landscapes are managing a host, duplicate models are created. One host model is created for each landscape where an ADES Host Manager is managing the host. To resolve duplicate models, use the following procedure.

Follow these steps:

1. Search for the host in each landscape. If a host is present:
   a. Unmanage the host in each ADES Host Manager in the landscape by using the Universal Host Table subview.
   b. (optional) After modifying the UHT subview, reconfigure the corresponding ADES Host Manager device to expedite its unmanagement process.
2. Repeat step 1 for every ADES Host Manager across all landscapes except one.

Multiply Managed Alarms

Symptom:

I do not see duplicate hosts in my ADES Manager hierarchy. But, I am getting an alarm on an Active Directory or Exchange Server host with the title "ACTIVE DIRECTORY AND EXCHANGE SERVER HOST MULTIPLY MANAGED".

Solution:

Manage a host by a single ADES Host Manager. If, inadvertently, multiple ADES Host Managers residing in the same landscape are managing a host, this alarm is generated. To avoid multiply managed alarms, use the following procedure.

Follow these steps:

1. Locate the landscape in which the host resides, and do the following steps:
   a. Unmanage the host in each ADES Host Manager in the landscape by using the Universal Host Table subview.
   b. (optional) After modifying the UHT subview, reconfigure the corresponding ADES Host Manager device to expedite its unmanagement process.
2. Repeat step 1 for every ADES Host Manager except for one.
   The alarm automatically clears when a single ADES Host Manager manages the Active Directory or Exchange Server host.

Connections Do Not Appear in Topology

Symptom:

My Active Directory and Exchange Server hosts are not showing connections to other devices in the OneClick topology view.

Solution:

To produce connections between your hosts and the rest of the network, models for any connecting devices must exist before the hosts are modeled.

When discovering and modeling your environment, run a standard DX NetOps Spectrum Discovery first to model upstream routers and switches. Then, ADES Manager Discovery can run, creating models and connections for the Active Directory and Exchange Server components.

To create connections for managed hosts, use the following procedure.

Follow these steps:

1. Verify that devices such as routers and switches that are upstream from your Active Directory and Exchange Server environment are modeled. If not, run a standard DX NetOps Spectrum Discovery to model these connecting devices.
2. If the connecting devices are modeled after your Active Directory and Exchange Server environment is modeled, run Discover Connections on each of the affected devices.

   NOTE
   For information on Discover Connections, see the Modeling and Managing Your IT Infrastructure section.

ADES Managed Hosts Container Not Created

Symptom:

I have modeled an ADES Host Manager, but I do not see a corresponding ADES Managed Hosts container being created for it.

Solution:

DX NetOps Spectrum ADES Manager requires ADES AIM r12.7 or later. Verify that ADES AIM r12.7 or later is running on the ADES Host Manager.

Host Not in ADES Managed Hosts Container

Symptom:

My ADES Managed Hosts container was created but, when I look inside, an expected Active Directory or Exchange Server host is not there.

Solution:

Hosts that were modeled before ADES Manager Discovery are not moved in the topology when brought into management by ADES Manager. To locate the expected host, look in other possible locations in the landscape for the model. In particular, if the Active Directory or Exchange Server host is a virtual machine, it is located in the physical host container.

Unable to Update Attribute

Symptom:

I get the following error when I try to manage or unmanage an Active Directory or Exchange Server host in the Universal Host Table subview:

```
Unable to update attribute exchAdUniversalHostManagedByAgent.  No response from the device.
```

Solution:

The ADES Manager Host must be modeled with a read/write community string. If not, you cannot update the model. To resolve this problem, change the community string of the ADES Host Manager to its read-write value. The SNMP Community String is on the DX NetOps Spectrum Modeling Information subview for the ADES Host Manager model.

Host Subview is Empty

Symptom:

A role-specific subview for my Active Directory or Exchange Server host does not contain any data.

Solution:

A role-specific subview contains data only when the host supports the respective role. Verify that the host supports the role whose data the subview is displaying. Subviews for unsupported roles are empty. For mailbox servers, also verify that the host supports the correct Exchange version, 2007 or 2010; the subview for the unsupported version is empty.

Changes in Environment Not Reflecting

Symptom:

When I modify my Active Directory or Exchange Server environment, the change is not reflected in my modeled environment. For example, after I add or remove a host, I do not see a difference in DX NetOps Spectrum.

Solution:

An ADES AIM Discovery must be run manually before the ADES AIM detects the environment change and models it in DX NetOps Spectrum. To update your modeled environment in DX NetOps Spectrum, initiate the ADES AIM Discovery process manually. ADES AIM Discovery then triggers ADES Manager Discovery, which reflects any changes in DX NetOps Spectrum.

ADES Manager Updates are Slow

Symptom:

The data specific to the Active Directory and Exchange Server technologies is not updating as frequently as I expect it to.

Solution:

Data update delays in DX NetOps Spectrum ADES Manager can be due to ADES AIM performance issues. The number of hosts that the ADES AIM manages can affect performance. The geographic proximity of the ADES AIM to the monitored environment can also affect performance. These factors can contribute to long poll times.

To improve performance by the ADES AIM, use the following procedure.

Follow these steps:

1. Verify that the polling interval is set appropriately.
2. If the polling interval is set as expected, the ADES AIM possibly can be overloaded or located too far from the managed hosts. Check that the recommended limits and configuration for the ADES AIM are adhered to.

   NOTE
   For information on load balancing and sizing guidelines for the ADES AIM, see the *CA Virtual Assurance for Infrastructure Managers* (CA VAIM) documentation.

Cluster Manager

This section explains how DX NetOps Spectrum monitors and manages IBM PowerHA and Microsoft Cluster Service environments.

Introduction to Cluster Manager

Cluster Manager is a DX NetOps Spectrum feature that models and monitors your cluster environment and is intended for administrators. Cluster Manager provides an enterprise-wide view of your cluster environments, showing both topology and the logical relationships among your cluster components. Cluster Manager also provides visibility into useful metrics and helps you pinpoint and effectively troubleshoot problems by applying unique fault isolation techniques.

A key challenge when monitoring your cluster environment is tracking where work is occurring, or identifying the active nodes. Clustering technology is designed to sustain high availability for your server-based applications by providing a fail-safe environment. Resource groups move from one cluster node to another when needed, resulting in the distribution of work and node status changing periodically. Cluster Manager keeps up with these activities by continuously monitoring your cluster components, and quickly notifies you of any changes to your environment.

Cluster Manager Features

Cluster Manager includes the following features:

- Automated device discovery and modeling. Cluster Manager automatically creates models and connections for all managed cluster components, as appropriate.
- A distributed solution that can handle scaling. Cluster management can be distributed across multiple SpectroSERVERs.
- Identification of cluster components in the topology.
- Hierarchical representation of cluster environments.
- Icons that distinguish devices in your cluster environment, including distinct identification of active and inactive nodes.
- Dedicated Cluster Manager views that provide visibility into data specific to cluster environments and respective technologies.
- Events and alarming on cluster entities and activities, provided out-of-box.
- Enhanced fault management. Cluster Manager recognizes and correlates symptomatic alarms and aids fault isolation with proxy management.
- Locater searches specific to cluster components.
- Consistent representation across all supported cluster technologies.

Supported Technologies

Cluster Manager supports the following cluster technologies when all required components are installed and configured properly per solution, as follows:

- **IBM PowerHA**
- 9.2.3 or later
- A dedicated host machine with:
  - SystemEDGE 5.x or later
  - High Availability Cluster Multiprocessing (HACMP) AIM r12.7 or later

    WARNING
    The HACMP AIM must be the only AIM installed on the SystemEDGE host. The SystemEDGE host itself cannot be a node in your managed cluster environment.
- **Microsoft Cluster Service (MSCS)**
- 9.2.3 or later
- A dedicated host machine with:
  - SystemEDGE 5.x or later
  - MSCS AIM r12.7 or later

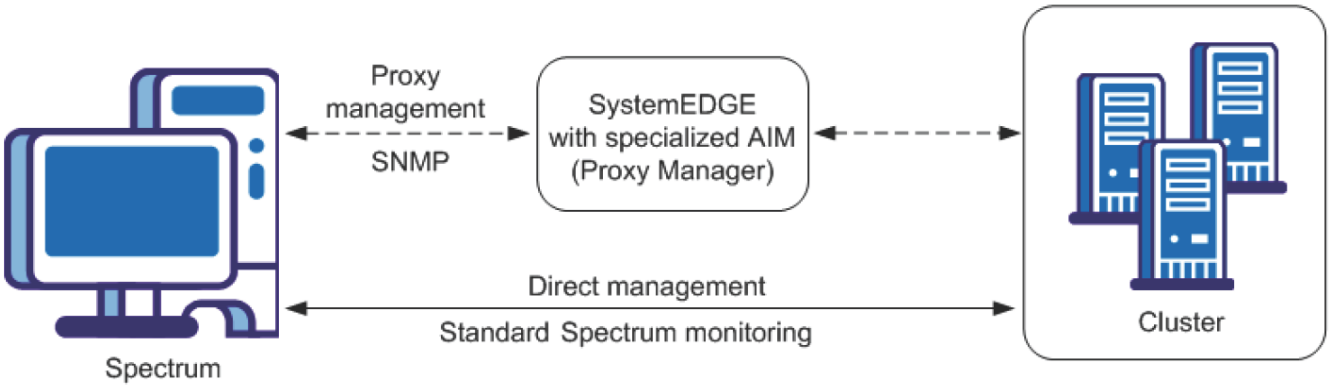
WARNING
The MSCS AIM must be the only AIM installed on the SystemEDGE host. The SystemEDGE host itself cannot be a node in your managed cluster environment.

NOTE
For more information about the SystemEDGE agent and AIM system requirements, see the CA Virtual Assurance for Infrastructure Managers section.

Solution Architecture for Managing Cluster Environments

Cluster Manager monitors your cluster components seamlessly within your network, while providing data that is specific to cluster technologies. DX NetOps Spectrum gathers information about your cluster components using two different methods. As with other DX NetOps Spectrum-managed devices, Cluster Manager uses standard DX NetOps Spectrum monitoring. In addition, Cluster Manager also retrieves specialized information from an alternate (proxy) manager, the SystemEDGE Application Insight Module (AIM).

An AIM is a specialized extension of the SystemEDGE agent and resides on its own host. The proxy manager communicates directly with entities in your cluster environment. DX NetOps Spectrum then uses SNMP to retrieve this information from the proxy manager and uses it to model and monitor your cluster components in OneClick.



The AIMs that work with Cluster Manager include:

- **High Availability Cluster Multiprocessing (HACMP) AIM**
  Provides capabilities to monitor your IBM PowerHA cluster environment.
- **Microsoft Cluster Service (MSCS) AIM**
  Provides capabilities to monitor your Microsoft Cluster Service environment.

NOTE
For more information about the SystemEDGE agent and AIMs, see the *CA Virtual Assurance for Infrastructure Managers* documentation.

Cluster Concepts

Administrators organize resources into functional units that are called resource groups, and assign these groups to individual nodes. If a node fails, the resource groups that were being hosted on a particular node move to other nodes in the cluster.

The following terms describe these components of a cluster environment and appear in the Cluster Manager solution:

- **Cluster**

A group of locally attached machines that provide distributed processing power and high availability. A cluster appears to clients as a single system image and IP address.

- **Node**
  An independent computer system that participates in a cluster.
  - **Active node**
    A system in a cluster environment where application processes (as part of a resource group) are currently running. Within Cluster Manager, an active node has resource groups as children.
  - **Inactive node**
    A system that is allocated to a cluster but not currently processing any resources. Within Cluster Manager, an inactive node does not have any resource groups as children.
- **Resource group**
  A collection of resources that forms a functional unit existing on a single node.
- **Resource**
  A logical component or entity that runs on only one node at a time. Resources encompass all elements needed for an application, such as network interfaces, disks, file systems, and application software.
- **Migration**
  Movement of a resource group from one node to another. Different terms are used regarding migration depending on the cluster technology; for example, failover, fall over, failback, and fallback.

Getting Started with Cluster Manager

This section provides the information to get started with understanding your cluster environment, the modeling process, installing the Cluster Manager. This section also provides information about how your cluster environment is organized in OneClick for visibility and management, and cluster fault management is also explained.

NOTE
Unless otherwise noted, the information in this section applies to all supported cluster technologies.

Planning Your Implementation

The purpose of Cluster Manager is to monitor your cluster components and notify you of various activities in your environment. Cluster Manager is highly scalable and can manage cluster nodes under different technologies using multiple AIMs across distributed SpectroSERVERs. Understanding how DX NetOps Spectrum manages the models for the components in your cluster environment provides for a more efficient Cluster Manager implementation.

Before you set up Cluster Manager, review the following topics:

- Environment Management Considerations
- Modeling

Environment Management Considerations

During setup of Cluster Manager, you specify how to organize the management of your environments. With a small environment, you can have a single AIM managing all of your cluster nodes (by vendor) in a single location on one SpectroSERVER. In a complex environment, you can have multiple AIMs across multiple SpectroSERVERs managing various cluster environments in different locations using different vendors.

Although organizational specifications can be changed at any time, knowing the available configuration options allows for a better initial setup.

Consider the following points when setting up your Cluster Manager environment:

- AIMs for Cluster Manager are vendor-specific. If you use cluster technologies from more than one vendor, you need multiple AIMs and dedicated SystemEDGE hosts.
- Each AIM can manage more than one cluster.
- Management of your cluster environment can be distributed across multiple AIMs, which can be supported in a single landscape or across multiple SpectroSERVERs.
- Management of each cluster can be by one cluster technology AIM only.

When deciding how to distribute management of your cluster environment, consider the number and location of nodes in your environment. The number of cluster nodes an AIM manages and the geographic proximity of the AIM to the monitored environment can affect performance. For best performance, size and balance management of the environment appropriately.

NOTE
The clusters that a particular AIM manages are controlled on the AIM and not in DX NetOps Spectrum. For information about defining nodes to manage by an AIM, see the CA Virtual Assurance for Infrastructure Manager section.

Modeling

As with other network elements supported in DX NetOps Spectrum, you discover and model the components of your cluster environment to monitor them. Cluster Manager obtains information about the clusters and nodes to manage from the AIM. DX NetOps Spectrum then uses this information to model each component using AutoDiscovery.

NOTE
Information that is used for the Cluster Manager feature is gathered primarily from the proxy manager (AIM). Additional information is also gathered directly from the nodes.

NOTE
The following topics provide more details about the modeling process:

- What Is Modeled
- Modeling Methods
- Node Management and Multiple Cluster AIMs
- Node Management and Multiple DX NetOps Spectrum AIM Solutions

What Is Modeled

Using information provided in the AIM MIB, DX NetOps Spectrum extracts and models any clusters, nodes, resource groups, and resources that the AIM manages.

Modeling Methods

Cluster nodes are modeled as SNMP-managed elements when possible. SNMP modeling supports enriched device monitoring that can provide added value to your Cluster Manager solution. If an SNMP agent is not installed on the host, it is modeled as an ICMP (Pingable) device.

Model Naming

When modeling cluster nodes, the model name assigned in DX NetOps Spectrum depends on the type of modeling used, as follows:

- For SNMP modeling, DX NetOps Spectrum automatically attempts to supply a name for the model using standard DX NetOps Spectrum naming conventions. Automatic naming is controlled at the SpectroSERVER level by the Model Naming Order value. This field is on the SpectroSERVER Control view for the VNM model.
- For ICMP (Pingable) modeling (when not a virtual device), DX NetOps Spectrum uses the host name provided in the AIM.

WARNING
For ICMP (Pingable) modeling, model names that DX NetOps Spectrum Virtual Host Manager sets take precedence over Cluster Manager.

The administrator can modify the name of a cluster node model at any time. As with other managed network elements, DX NetOps Spectrum automatically updates the model name using established naming rules, which can replace the user-defined value. To retain a user-defined value, lock the model name.

NOTE
You can modify and lock the cluster node model name using the following model attributes: Model_Name (0x1006e) and Lock_Model_Name (0x12a52).

IP Address and MAC Address Determination

When modeling cluster nodes, the IP address and MAC address assigned in DX NetOps Spectrum depend on the type of modeling used, as follows:

- For SNMP modeling, DX NetOps Spectrum automatically attempts to determine the addresses by querying the resident SNMP agent.
- For ICMP (Pingable) modeling (when not a virtual device), DX NetOps Spectrum uses the addresses that the AIM provides.

WARNING
For ICMP (Pingable) modeling, addresses that DX NetOps Spectrum Virtual Host Manager sets take precedence over Cluster Manager.

If the SNMP modeling or Virtual Host Manager cannot supply a valid IP address or MAC address, the AIM value is used.

Node Management and Multiple Cluster AIMs

Manage a cluster node by a single cluster technology AIM only. If you inadvertently manage a cluster node by multiple cluster technology AIMs, Cluster Manager issues the following alarm on the cluster model:

```
UNSUPPORTED CLUSTER AIM CONFIGURATION
```

Children are not created for the cluster model.

Node Management and Multiple DX NetOps Spectrum AIM Solutions

When managing a cluster node model by multiple DX NetOps Spectrum AIM solutions, a defined ranked order of management applies, as follows:

1. Virtual Host Manager
2. Cluster Manager
3. Other technologies (such as Active Directory and Exchange Server Manager)

When a node with a SystemEDGE agent is already modeled in DX NetOps Spectrum, Cluster Manager recognizes the model and a duplicate model is not created. Instead, Cluster Manager pulls the existing model into its own management, abiding by and applying the rules of each solution in the ranked order.

For example, when both Virtual Host Manager and Cluster Manager are managing a node, model parameters that Virtual Host Manager assigns are used. Examples of these parameters include the model name, IP address, and MAC address.

If a node is removed from management by a solution, the rules of the remaining solutions are reapplied in the ranked order. Typically, any changes are made at the next polling cycle.

This defined order of management also affects how models appear in the Universe topology.

How to Install Cluster Manager

When you install DX NetOps Spectrum, the Cluster Manager components are automatically installed and available for use. However, Cluster Manager is operable only after you also install and configure the appropriate proxy manager for your solution.

Refer to the respective section for your solution.

Cluster Manager Discovery and Modeling

After the necessary components have been installed, discover and model any entities that Cluster Manager is going to manage.

Cluster Manager uses the following types of discovery:

- Standard DX NetOps Spectrum Discovery to model the cluster technology AIM and connecting devices
- Cluster Manager discovery to model cluster components

After the cluster technology AIM is modeled successfully, Cluster Manager obtains information about the cluster components in your environment from the AIM. Using a list of machines that is obtained from the AIM, Cluster Manager uses AutoDiscovery to model each cluster node. All supporting cluster components (clusters, resource groups, and resources) are also modeled.

Cluster Manager models cluster nodes as SNMP-managed when possible and when AutoDiscovery parameters are set up correctly.

> **NOTE**
> For information about AutoDiscovery control settings, see the Modeling and Managing Your IT Infrastructure section.

The information provided in this topic applies to all cluster technologies. For more details, refer to the respective section for your solution.

How to Model Your Environment When Using Multiple AIM Solutions

Depending on your environment, you can use Cluster Manager in combination with other DX NetOps Spectrum AIM solutions simultaneously to manage your network entities. Any of the following configurations require the use of multiple solutions for complete management of your environment:

- A cluster node runs on a virtual machine.
- A cluster technology AIM runs on a virtual machine.
- A cluster node is an Active Directory or Exchange Server host.

Each of the DX NetOps Spectrum AIM solutions provides information that is specific to the technology it supports. For example:

- Virtual Host Manager provides details that are specific to virtual technologies.
- Cluster Manager provides details that are specific to cluster technologies.
- Active Directory and Exchange Server Manager (ADES) Manager provides details that are specific to the supported server roles in Active Directory and Exchange Server.

The combination of these features provides a complete monitoring solution. To set up your implementation of multiple AIM solutions effectively, the following approach is recommended.

> **WARNING**
> When using multiple AIMs, only a single AIM can be installed on a given SystemEDGE host.

Follow these steps:

1. Configure the AutoDiscovery settings on the VNM model.
2. Configure the Virtual Host Manager settings that are related to your virtual technology.
3. Set up Virtual Host Manager by modeling the virtual technology manager and all virtual technology components.
4. Set up Cluster Manager by modeling the cluster technology manager and all cluster components.
5. Set up ADES Manager by modeling the ADES Host Manager and all Active Directory and Exchange Server hosts.

NOTE
For more information, see the Virtual Host Manager section and the Active Directory and Exchange Server Manager section.

Viewing Your Cluster Environments

The purpose of Cluster Manager is to provide visibility into your cluster environments. This visibility lets you identify the organization of your environment, where resource groups are allocated, and the status of each node. Most importantly, when a problem occurs in your environment, you can pinpoint its cause.

Cluster Manager provides several methods for viewing your cluster environments, as follows:

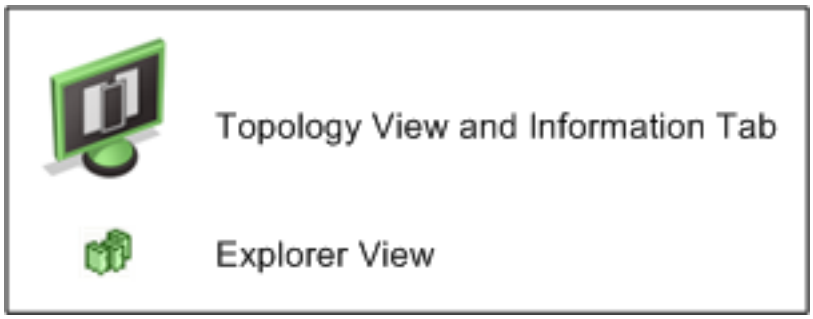
- The Cluster Manager hierarchy in the Navigation panel indicates the logical relationships between components. Examples of hierarchy nodes include clusters, cluster nodes, resource groups, and resources.
- A graphical topology view helps you to group cluster nodes and visualize the connections between them.
- Custom Information views in the Component Detail panel provide details that are specific to cluster technologies and specific vendors.
- Custom searches provide a quick way to find cluster elements.
- Custom icons for individual models provide status and model type information at a glance and are integrated throughout the Cluster Manager feature.

Understanding each of these methods can help you monitor your cluster environments more efficiently, letting you troubleshoot issues more effectively.

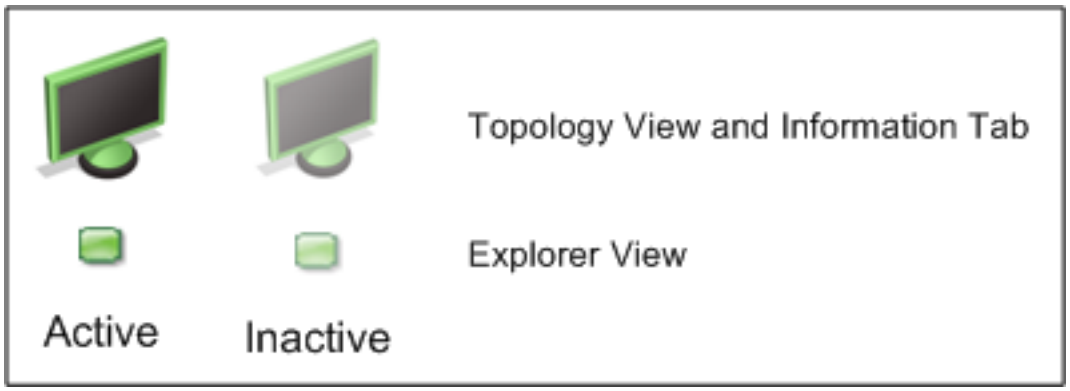
Icons for Cluster Manager

Cluster Manager provides icons that are designed specifically to distinguish devices in your cluster environment. The same icons are used across all cluster vendor technologies.

- **Cluster**
  Cluster icons have a distinctive cluster pattern, representing three workstations that are clustered together, as follows:

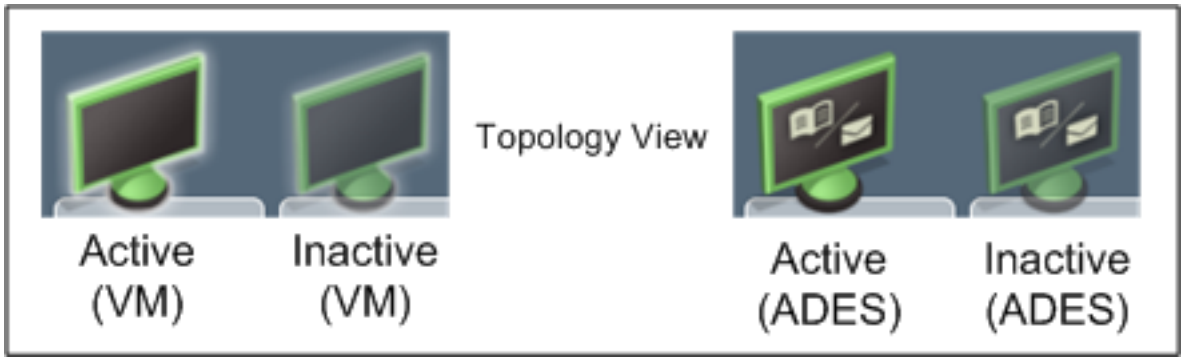


- **Cluster Node**
  Cluster nodes use standard workstation icons. An active node has a solid (nontransparent) representation whereas an inactive node is faded (transparent), as follows:



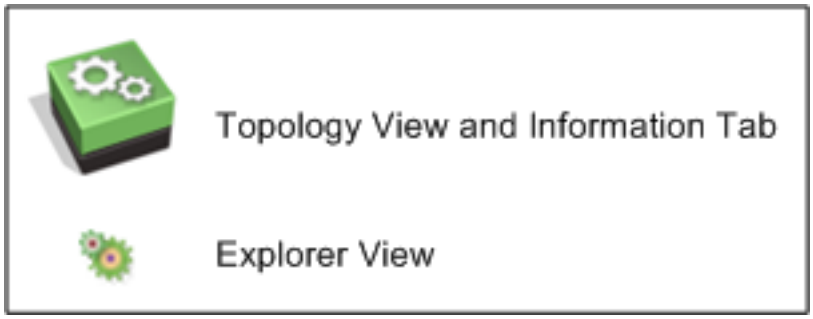
The icon also reflects when Cluster Manager is used in combination with other DX NetOps Spectrum AIM-based solutions. The following example shows:

– An active node and an inactive node in the topology where both nodes are virtual machines that Virtual Host Manager manages. Notice that the virtual halo is brighter for the active node.
– An active node and an inactive node in the topology where both nodes are Active Directory or Exchange Server hosts.



Note: When an inactive node is used with the spotlighting feature, the icon becomes even more transparent.

- **Resource Group**
  Resource groups have an icon that has multiple gears, as follows:



- **Resource**
  Resources have an icon that has a single gear, as follows:

NOTE
Resources are not displayed in the topology view.

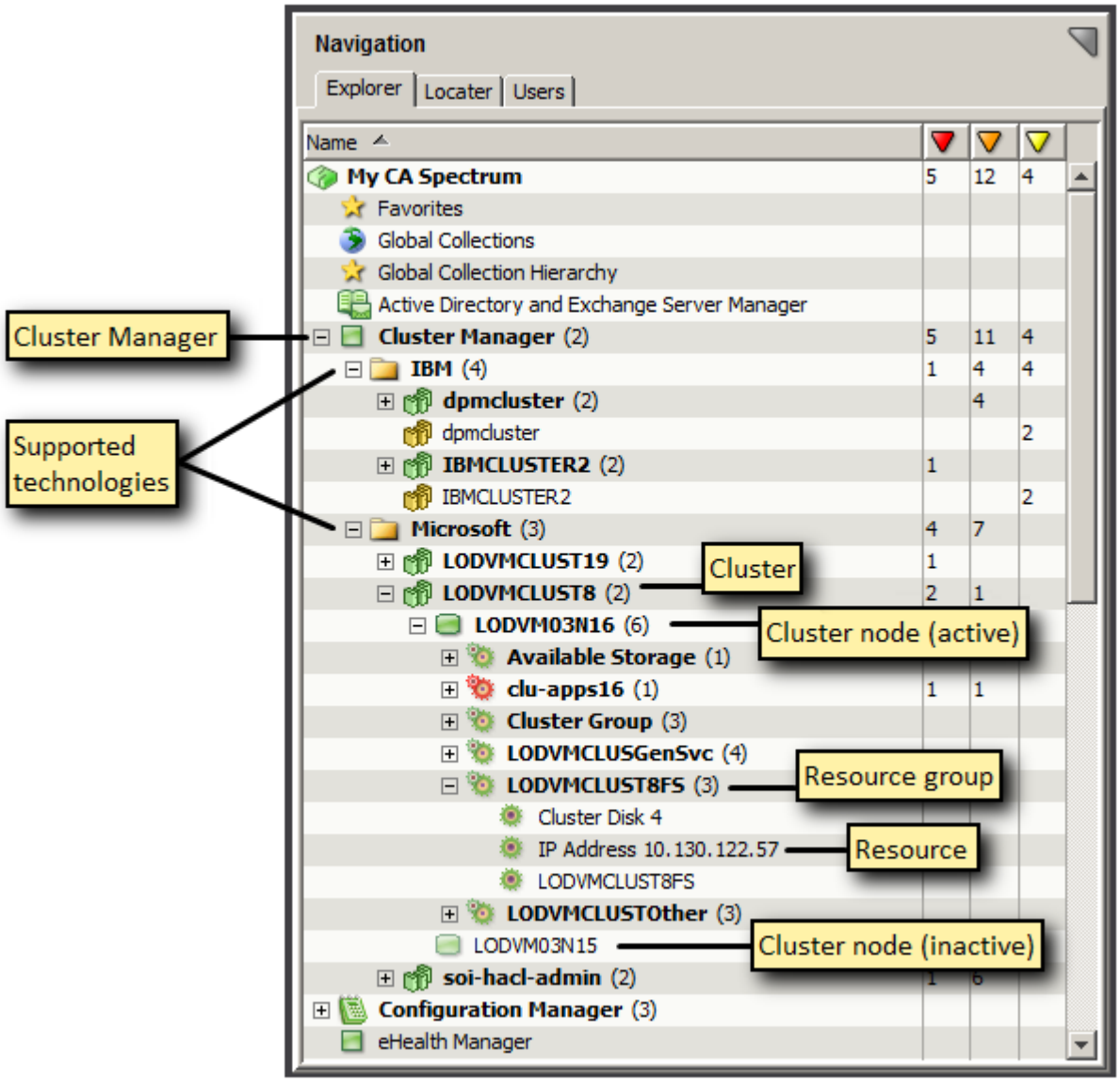
Explorer View

On the Explorer tab of the Navigation panel, Cluster Manager provides a hierarchical tree structure that illustrates the logical organization of your managed cluster environments. Custom icons provide status and model type information for your cluster components at a glance.

Using this information, you can see how the clusters and respective resources are arranged logically in your environment and where they are active.

NOTE
Only users with the appropriate privileges and model security access can view the Cluster Manager hierarchy and components. For more information, see the OneClick Administration section.

The following image is an example of the Cluster Manager hierarchy:



The following elements form the hierarchy. When an element in the hierarchy has children, the label is in bold.

- Cluster Manager
  Denotes the root for the cluster environments that are currently managed. Cluster Manager is a distributed solution that handles multiple landscapes and so appears above the landscape level. Expanding the Cluster Manager element displays the technologies that are supported. The clusters and participating cluster components for each technology are also presented, as depicted in the following diagram:

```
[-] Cluster_technology_1
    [-] Cluster_1
        [-] Cluster_node_1 (active)
            [-] Resource_group_1
                . Resource 1
                . Resource 2
                . Resource 3
            Cluster_node_2 (inactive)
```

```
            Cluster_node_3 (inactive)
      [+] Cluster 2
  [+] Cluster_technology_2
```

NOTE
Only those solutions for which a respective AIM has been installed appear in your implementation. The AIM itself does not appear in the Cluster Manager hierarchy. The Cluster Manager (AIM) model appears in the Universe topology and the Universe hierarchy.

- Cluster technology 
  Represents a vendor cluster technology. The cluster technology folder displays all managed clusters across all landscapes for the respective technology, such as IBM or Microsoft.
  The hierarchy within the vendor folder shows the logical relationships between the participating components. When all cluster components for a cluster technology are deleted, the empty cluster technology folder remains.
- Cluster 
  Represents a cluster. The cluster name that is used is obtained from the AIM and differs based on technology.
- Cluster node 
  Represents a cluster node. The transparency of the icon reflects whether a cluster node is active or inactive. A solid (nontransparent) icon represents an active node; a faded (transparent) icon represents an inactive node. An active node has resource groups as children; an inactive group does not have any resource groups.
- Resource group 
  Represents a resource group.
- Resource 
  Represents a resource.

  NOTE
  Resources are displayed in the hierarchy view only; resources are not displayed in the topology view.

Management by Multiple AIM Solutions

When Cluster Manager and another DX NetOps Spectrum AIM solution simultaneously and successfully manage a cluster node, the following details apply when viewing your environment:

- The Cluster Manager hierarchy provides a complete and accurate view of your cluster environment. The Universe hierarchy presents all models by defined order of solution management. For example, when a cluster node is a virtual machine, it does not appear within its cluster container in the Universe hierarchy. Instead, it appears in its physical host container in the Virtual Host Manager hierarchy.

  NOTE
  When multiple solutions are managing a node, you can quickly locate it in the correct hierarchy from the Contents panel. Locate the model in the List view or Topology view, then right-click the model and select Location.
- In the Explorer view, the icon for the highest ranking solution applies.
- If Virtual Host Manager manages the node, the node name that appears in the Cluster Manager hierarchy and List view is as follows:
  - For ICMP (Pingable) models, the model name that Virtual Host Manager sets is used.
  - For SNMP models, default Cluster Manager model naming is used.

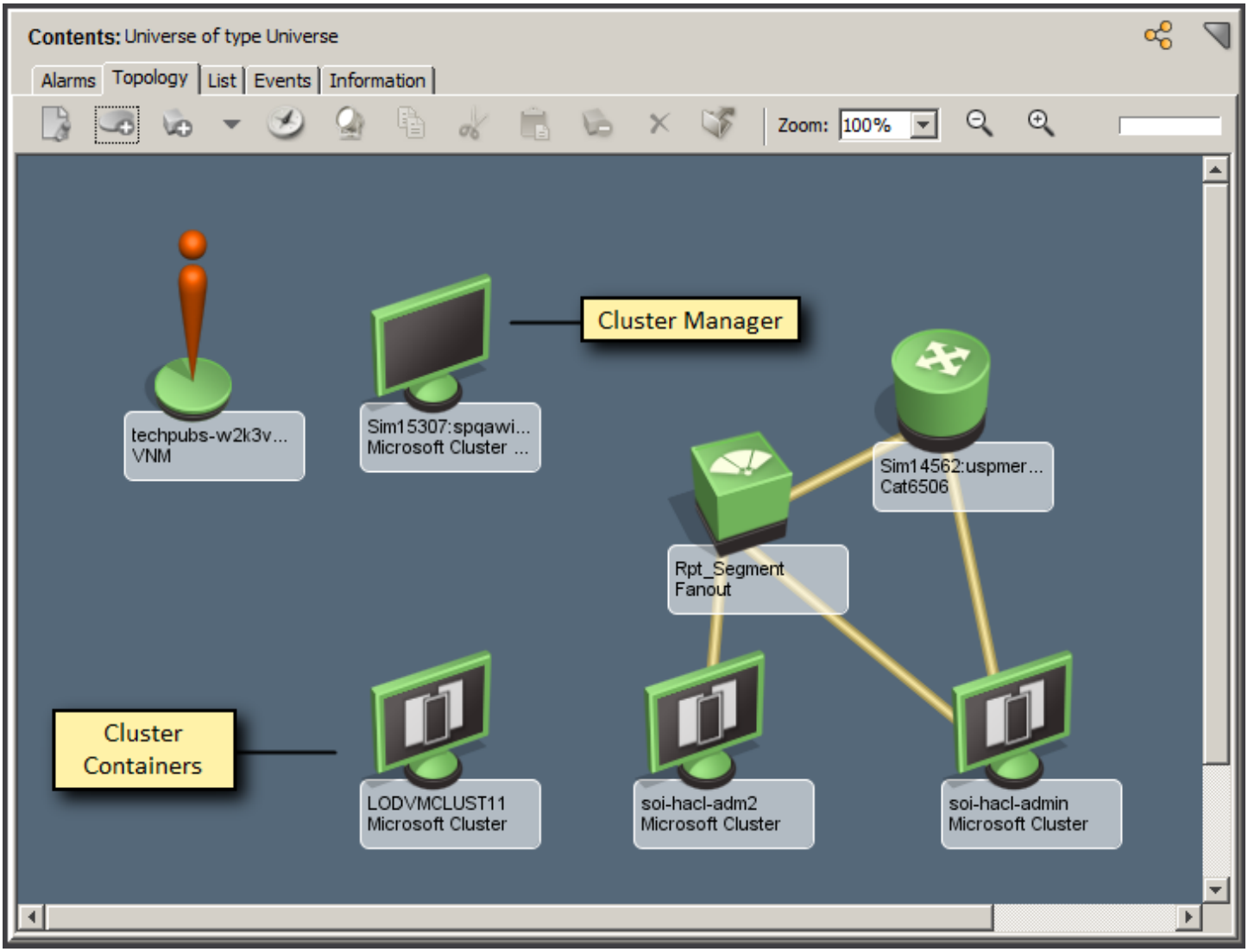
NOTE
Typically, most DX NetOps Spectrum AIM solutions use the Domain Naming System (DNS) name. Regardless, the highest ranking solution applies.

Topology View of a Cluster Environment

The models for your managed cluster environment are organized and integrated into the Universe topology view. These models include the Cluster Manager (SystemEDGE host), cluster, cluster node, and resource group models. This graphical representation helps you visualize the structure of your managed environment, including connections between cluster nodes and other elements of your network.

Cluster node and resource group models are organized in the topology in cluster containers. When possible, the cluster container model is created alongside the Cluster Manager model, as shown in the following Universe topology top view example.

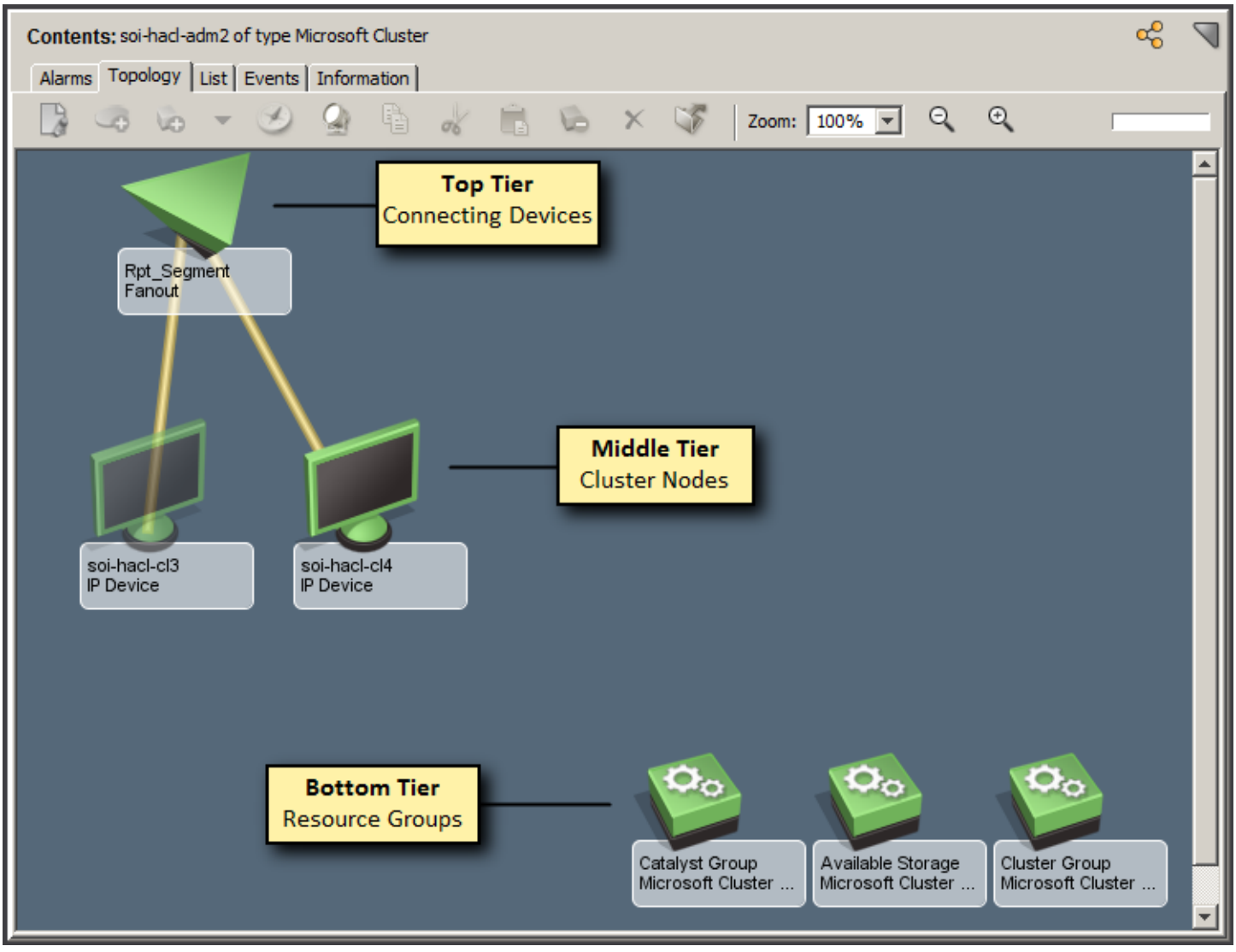
NOTE
If the Cluster Manager is a virtual machine, the cluster container is created in the same topology as the Virtual Host Manager physical host container.



Drilling down into a cluster container reveals its contents. These contents include off-page references to connected network devices, cluster nodes, and active resource groups participating in the cluster.

NOTE
Resources are displayed in the Cluster Manager hierarchy view only; resources are not displayed in the topology view.

The following illustration shows a drill-down view of a cluster container. Cluster components are arranged in three tiers, as shown:



WARNING
Although you can edit the cluster container topology view, DX NetOps Spectrum enforces the placement of the models within the appropriate tiers. If you rearrange the models in this view, their placement is not preserved. Other changes, such as text annotations and background changes, are retained.

- **Top tier**
  Displays any off-page references to connecting devices modeled in other views, such as upstream routers, repeaters, and switches. These elements connect your cluster nodes to your network.
- **Middle tier**

Displays the cluster nodes that participate in the cluster. If a cluster node has already been modeled in Virtual Host Manager before Cluster Manager discovery, it is not modeled again. However, it is included in the cluster container topology to provide a complete view of your cluster environment.

- **Bottom tier**
  Displays the resource groups that participate in the cluster.

  NOTE
  Associations between the resource groups and their respective nodes are not displayed in this view; this information is provided in the Cluster Manager hierarchy view.

The following rules apply to cluster containers:

- You cannot add or remove models from a cluster container. Changes in cluster container contents can occur when components are added or removed from management by an AIM or changes occur in the cluster environment. However, DX NetOps Spectrum controls the placement of models within a cluster container exclusively.
- You cannot destroy a cluster container directly. The cluster container is destroyed only when the respective Cluster Manager model is deleted or when the cluster is removed from management by the AIM. When the cluster container is destroyed, all cluster node models in the container are moved to the Lost and Found (LostFound). An exception is when a cluster node is in a global collection; in this case, the model remains in the global collection.

Placement of Models

The placement of cluster node and resource group models in the topology during Cluster Manager discovery occurs as follows:

- If Cluster Manager discovery creates the model, the model is placed in a cluster container.
- If the model exists and is for a virtual machine that Virtual Host Manager manages, the model remains in the physical host container. And, the model is also included in the cluster container. Within the cluster container, the cluster node icon retains the characteristics of a virtual machine.
- If the model exists and is for a physical machine that another AIM solution manages, the model is moved to the cluster container. An example of another AIM solution is ADES Manager.

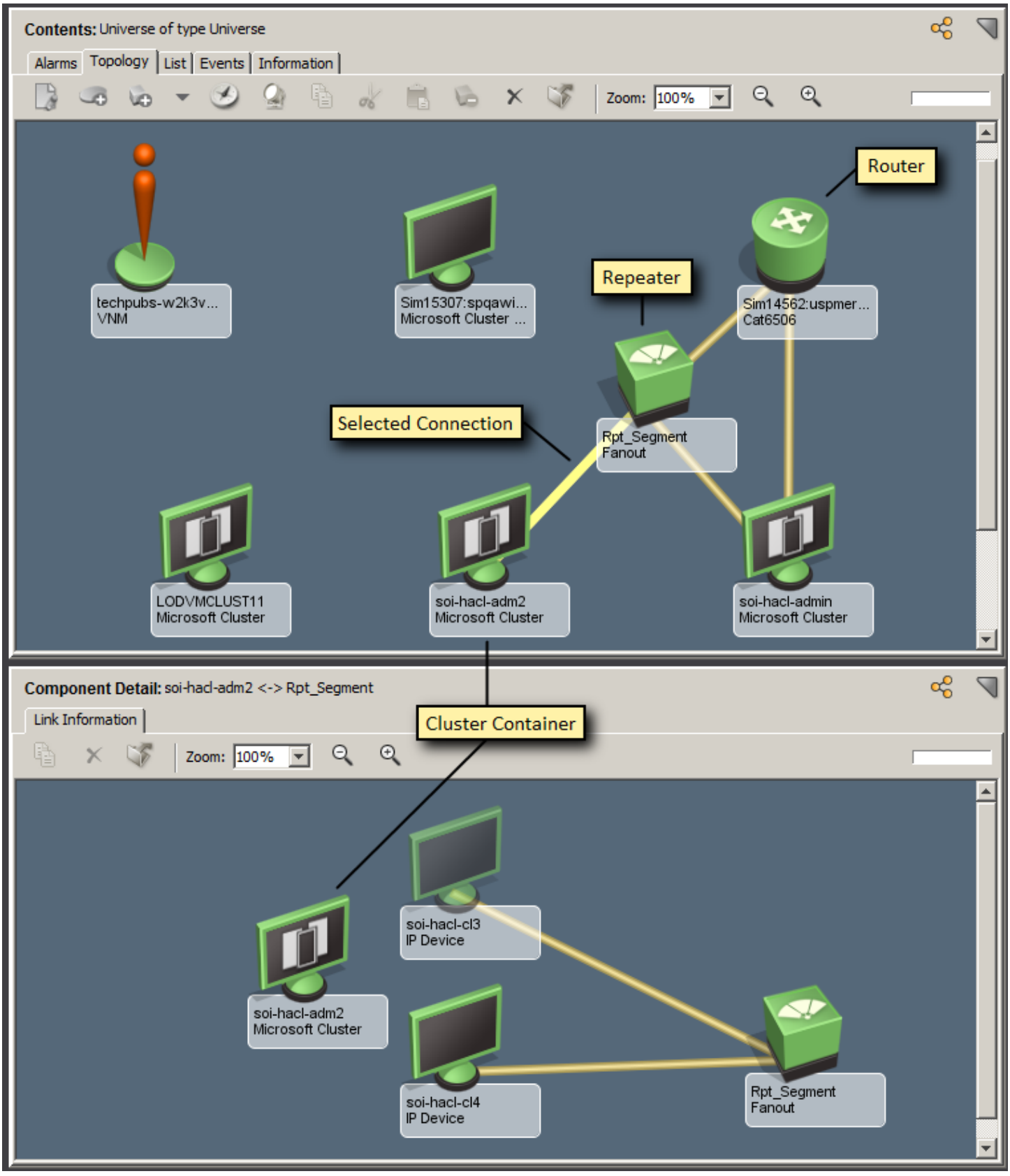
  NOTE
  When you remove a model from management by Cluster Manager, it is removed from the cluster container. If ADES Manager continues to manage the host, the model does not appear in the ADES Managed Hosts container automatically. To move the model, cut and paste the model from Lost and Found (LostFound) into the ADES Managed Hosts container.
- If the model exists and no other AIM solution manages the model, the model is moved to the cluster container.

Connectivity

The Topology view displays the connectivity for your cluster environment within your network. Cluster Manager provides the links between your cluster nodes and any connecting devices that have been modeled in your network.

WARNING
Cluster Manager provides the connection to the physical IP address of the cluster node and not to the virtual IP address of the cluster.

In the Universe view, the connections (pipes) represent connections from the connecting devices to the cluster nodes within the cluster containers. By selecting a connection to examine the link more closely, the connections to the specific nodes are displayed, as illustrated in the following example:


Contents: Universe of type Universe
Alarms
Topology
List
Events
Information
Zoom: 100%
techpubs-w2k3v...
VNM
Sim15307:spqawi...
Microsoft Cluster ...
Router
Repeater
Sim14562:uspmer...
Cat6506
Selected Connection
Rpt_Segment
Fanout
LODVMCLUST11
Microsoft Cluster
soi-hacl-adm2
Microsoft Cluster
soi-hacl-admin
Microsoft Cluster
Component Detail: soi-hacl-adm2 <-> Rpt_Segment
Link Information
Cluster Container
Zoom: 100%
soi-hacl-cl3
IP Device
soi-hacl-adm2
Microsoft Cluster
soi-hacl-cl4
IP Device
Rpt_Segment
Fanout

Discover Connections runs automatically when the following actions occur:

- A cluster node is initially discovered and modeled.
- The IP address or MAC address of the cluster node has changed.

Discover Connections runs when necessary on the poll cycle. If the Discover Connections process does not complete within a single polling interval, the next Cluster Manager discovery is postponed.

Information Subviews

DX NetOps Spectrum includes several tabs in the Contents and Component Detail panels to provide quick access to information you need for monitoring your cluster environment. The Information tab provides details about a single entity in your environment. These details are displayed in the expandable subviews and vary by solution.

Custom subviews provide detailed information that is specific to the cluster component type. Custom subviews are provided for the following cluster components:

- Cluster Manager
- Cluster
- Cluster Node
- Resource Group
- Resource

Locater Searches for Cluster Manager

DX NetOps Spectrum provides a collection of preconfigured searches on the Locater tab that are designed specifically for your cluster environment. You can use these searches to locate entities in the DX NetOps Spectrum database that are related to the supported cluster technologies. These searches identify specific models or groups of models and can help you obtain details that you can use when monitoring your cluster environment. The searches are grouped under the Cluster Manager folder in the Locater tab of the Navigation panel.

> **NOTE**
> Only users with the appropriate privileges can access Cluster Manager searches. For more information, see OneClick Administration .

Event and Availability Reports

To monitor the cluster environment, you can create event and availability reports. Event reports gather information that helps you make informed decisions about the components in the cluster environment. Using the event filters, you can base the event reports on any of the management events that are generated for the cluster environment in DX NetOps Spectrum.

To report on cluster events, the following event filter files are included with Spectrum Report Manager:

- **Cluster.xml**
  Contains all cluster events, including IBM and Microsoft.
- **IBM-Cluster-all.xml**
  Contains all of the IBM cluster events.
- **IBM-run-status.xml**
  Contains all of the IBM cluster events that are related to Status (such as up, down, offline).
- **MS-Cluster-all.xml**
  Contains all of the Microsoft cluster events.
- **MS-run-status.xml**
  Contains the Microsoft cluster events that are related to Status (such as up, down, offline).
- **ClusterTrap.xml**

Contains only the trap events from IBM and Microsoft clusters.

- **Cluster-spectrum-managing.xml**
  Contains the DX NetOps Spectrum management events, such as cluster proxy events, management events, and polling events.

Availability reports provide historical information about uptime and downtime for assets in the IT infrastructure. The calculation of uptime and downtime depends on the UP and DOWN events that correspond to the cluster model type.

> **NOTE**
> You can use the event codes of the .xml files to generate event reports in Spectrum Report Manager. For more information, see the Report Manager section. You can also generate reports using the predefined event filter files. For more information, see the Install Report Manager section.

Cluster Manager Alarms and Fault Management

Knowing about certain activities, such as a resource group migration or a cluster node failure, can minimize potential problems in your cluster environment. To alert you, DX NetOps Spectrum generates alarms and uses advanced fault management techniques to isolate the root cause.

Problems with a single device can cause several other devices in your network to generate events. Deciding which devices are the root cause of an alarm can be challenging. For example, when you lose contact with the Cluster Manager (the proxy manager), you also lose proxy communication with the cluster nodes that it manages. As a result, alarms are generated for the Cluster Manager and each of its managed components. Sifting through potentially hundreds of simultaneously produced alarms manually to pinpoint the problem could be a tedious and error-prone process. Using fault isolation techniques, Cluster Manager significantly simplifies the troubleshooting process by automatically correlating these alarms to identify a single root cause. As a result, you can identify and correct the problem more quickly.

Alarms and fault isolation vary by cluster technology. Cluster Manager evaluates which devices are issuing alarms and the type of events the devices generate. DX NetOps Spectrum uses all available information to correlate the alarms to the appropriate root cause, only alarming on the isolated faulty device. Cluster Manager relies on the combination of standard DX NetOps Spectrum monitoring, proxy management, state-polling, and traps to create meaningful events and alarms.

> **NOTE**
> In addition to what is provided out-of-box, you can also create your own custom watches to generate events and alarms on other specific metrics. For information about creating watches, see the Watches section.

Cluster Manager Alarms

Alarms are created from information that is obtained from technology-specific traps and polling. To alert you to important activities within your cluster environment, Cluster Manager generates (or clears) alarms for the following conditions:

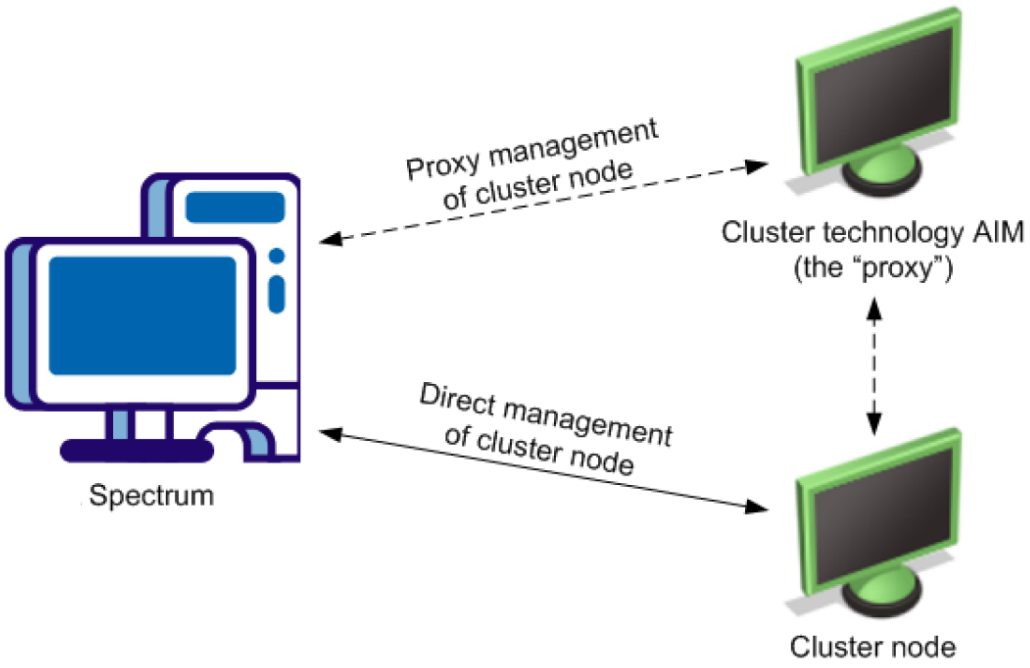
- A Cluster Manager (proxy) is down or communication lost
- Multiple cluster technology AIMs manage the same cluster
- A cluster is up, down, not configured, or in an unknown state
- A cluster node is up, down, joining, leaving, paused, or in an unknown state
- A resource group is online, offline, pending, unmanaged, in an unknown state, in various other states, or has produced an error
- A resource group migrates from one node to another
- A resource is online, online-pending, offline, offline-pending, initializing, pending, inherited, failed, or in an unknown state

> **NOTE**
> Alarms and conditions vary by cluster technology.

Proxy Management

Managing cluster components by a cluster technology AIM provides DX NetOps Spectrum a unique management opportunity. Using this approach, DX NetOps Spectrum has an alternate management perspective in addition to standard device monitoring methods.

Along with gathering information directly from a device, DX NetOps Spectrum also simultaneously gathers information specific to cluster components from the cluster technology AIM. The AIM serves as a "proxy" from which DX NetOps Spectrum gathers information specific to the cluster technologies. Management of a device using an alternate source (such as the AIM) rather than the device directly is called *proxy management*.



DX NetOps Spectrum fault isolation handles this dual management by producing the following alarms:

- **Proxy management alarms**
  By using the cluster technology AIM for management, proxy-related alarms can be generated. These alarms are unique because they alert you when acquisition of cluster-specific information through the proxy is affected, not the state of the device or direct (SNMP) management. When contact through the proxy is lost, you could be missing important cluster-specific information about your environment. Proxy management alarms are of major severity and are not clearable by the user.
  - **Proxy unavailable**
    When DX NetOps Spectrum cannot communicate with the cluster technology AIM, a proxy unavailable alarm is generated on the Cluster Manager model.
    The following text is used for the proxy unavailable alarm:
    ```
    CLUSTER MANAGER UNAVAILABLE
    ```
  - **Proxy lost**
    When DX NetOps Spectrum cannot obtain information about the managed device by way of the proxy, a proxy lost alarm is generated. A proxy lost alarm is generated for the following conditions:
    - When DX NetOps Spectrum cannot communicate with the vendor-specific Cluster Manager model. A proxy unavailable alarm on the Cluster Manager model is generated as well as a proxy lost alarm for each of its managed components.
    - When the cluster technology AIM cannot successfully communicate with the cluster node.

The following text is used for the proxy lost alarm:

```
CLUSTER MANAGER PROXY LOST FOR cluster_entity
```

Cluster entity values include the cluster, cluster node, resource group, and resource.
A proxy lost alarm is generated only for entities that Cluster Manager manages. If the host is removed from management by the proxy, the respective proxy management alarms are cleared.

- **Enhanced contact lost alarms**
  Standard DX NetOps Spectrum alarms that indicate loss of contact with the proxy contain added correlation of Cluster Manager proxy management alarms. These proxy management alarms indicate loss of cluster-specific data acquisition.
  The following text is used for the contact lost alarm:

```
DEVICE HAS STOPPED RESPONDING TO POLLS
```

Alarm Correlation

Using standard DX NetOps Spectrum fault management, state-monitoring data, and added information from the proxy, Cluster Manager automatically correlates the alarms to identify a single root cause. Various state-monitoring and proxy-related alarms are correlated to an alarm on the relevant model to pinpoint the true root cause, such as:

- Contact lost
- Management lost
- Entity down, offline, or in a problem state
- Maintenance
- Hibernation

Cluster Manager provides many default correlations. To view or modify correlations, use the Condition Correlation Editor in OneClick.

NOTE
After an alarm has been issued, use the Impact tab for the alarm to view any correlated or symptomatic alarms.

Maintaining Your Cluster Manager Implementation

This section provides the information that helps you maintain your Cluster Manager implementation after you discover your cluster environment.

Updating Cluster Data

When the DX NetOps Spectrum administrator runs the initial Discovery process, Cluster Manager populates the Explorer tab in the Navigation panel with your cluster environment models. After Cluster Manager builds this initial hierarchy, your cluster environment can change. Cluster Manager continually works to keep this information updated. The information is useful for troubleshooting issues and optimizing performance only when it accurately reflects your environment.

Understanding how and when the information is updated can help you better evaluate the data and how your cluster environment is operating. For example, the following events can change your cluster environment configuration:

- Creating or deleting clusters, cluster nodes, resource groups, and resources
- Migration of cluster components from one entity to another

To keep your information accurate, Cluster Manager detects these changes by polling the AIM. Your modeled cluster environment is updated in DX NetOps Spectrum at each polling cycle. When a change in your cluster environment is detected, DX NetOps Spectrum performs the following tasks:

- Updates the placement of your cluster component models in the Cluster Manager hierarchy in the Explorer view
- Automatically rediscovers connections to the affected components in the Universe topology

DX NetOps Spectrum also receives traps from the AIM and generates the corresponding events. By reviewing the event log, you can find out when changes occur, such as when a resource group has migrated.

WARNING
To reestablish connections to your cluster component models correctly, all interconnecting routers and switches must be modeled. If these models do not exist before connections to your cluster components are rediscovered, DX NetOps Spectrum cannot resolve those connections. As a result, DX NetOps Spectrum cannot display the information correctly in the Universe topology view.

Controlling Polling Intervals on Cluster AIM and Cluster Manager

Polling intervals control how often information is obtained from managed devices. To keep data for your managed cluster environments current, Cluster Manager uses the polling intervals set on the following components:

- **Cluster technology AIM**
  The AIM polling interval indicates how often the AIM queries the cluster components for information. The AIM polling interval exists in the AIM, but you can modify this value from within DX NetOps Spectrum. The default value is 300 seconds with a minimum value of 30 seconds.
- **Cluster Manager model**
  The polling interval on the Cluster Manager model determines how often DX NetOps Spectrum polls the cluster technology AIM. The default value is 300 seconds with a minimum value of 30 seconds. This setting is available on the DX NetOps Spectrum Modeling Information view for the Cluster Manager (Host_systemEDGE) model.

Modifying Cluster Manager Management and Models

When changing your modeled environment, consider the following behaviors:

- When Cluster Manager no longer manages a cluster node, the model moves to the Lost and Found (LostFound), except in the following cases:
  - Virtual Host Manager is managing the host.
  - The host is in a global collection.
- When cluster component names change, Cluster Manager reflects the new values automatically.
- When the IP address or MAC address for a cluster node model is modified, connections to any connecting devices are automatically updated.

How to Convert an ICMP (Pingable) Model to SNMP-Managed

You can model a cluster node as an ICMP (Pingable) model and then later install an SNMP agent on the host. To take advantage of the SNMP capabilities, the cluster node must be remodeled.

Perform a discovery for the node manually to replace the model. A new model is created and then is pulled into Cluster Manager management during the next Cluster Manager discovery.

Deleting Cluster Manager Models

Consider the following behaviors and restrictions regarding the deletion of cluster component models in your DX NetOps Spectrum modeled environment:

- Models typically can be deleted from OneClick at any time for various reasons. However, Cluster Manager restricts your ability to delete models from the Cluster Manager hierarchy in the Navigation panel. To delete models manually, you have the following options:

  - Delete the Cluster Manager model.
  - Remove a cluster component using the vendor-specific cluster management tool.
- In Cluster Manager, models are sometimes deleted automatically. The following circumstances cause DX NetOps Spectrum to delete Cluster Manager models automatically:
  - A Cluster Manager model is deleted. If you delete a Cluster Manager model, DX NetOps Spectrum deletes all associated models.
  - An entity is removed from a supported cluster environment. As you update your cluster environment by modifying cluster, cluster node, resource group, and resource allocation, DX NetOps Spectrum also modifies those models accordingly. This update includes deleting respective models and their children where appropriate.
  - Upgraded models exist. In some cases, a cluster node is first modeled for Cluster Manager without SNMP capabilities. If SNMP capabilities are later added to a node, the previous model is deleted and replaced with the new, manually discovered SNMP-managed model.
- Hosts that both Virtual Host Manager and Cluster Manager manage adhere to all the standard modeling behaviors of virtual machines. These models cannot be deleted from the topology.
- When a Cluster Manager (Host_SystemEDGE) model is deleted, the corresponding cluster containers are destroyed. All cluster node models in the containers are moved to the Lost and Found (LostFound). An exception is when a cluster node is in a global collection, in which case, the model remains in the global collection.
- When all cluster components for a particular cluster technology are deleted, the cluster technology folder remains in the Cluster Manager hierarchy. An unbolded label indicates an empty folder.

Deleting Models When Using Multiple AIM Solutions

If you use Cluster Manager in combination with other DX NetOps Spectrum AIM solutions, consider the following points when deleting models in your environment:

- If you plan to no longer manage models using Virtual Host Manager, configure Virtual Host Manager delete settings to retain models. Otherwise, Virtual Host Manager deletes the cluster node model initially, losing any history or customization on the models. Cluster Manager then recreates the cluster node model during the next Cluster Manager discovery, which occurs on the next poll cycle.
- When Virtual Host Manager unmanages a cluster node and the model is retained, it is pulled back into Cluster Manager management automatically.
- If a node is removed from management by a solution, the rules of the remaining solutions are reapplied in the ranked order. Typically, any changes are made at the next polling cycle.
- When you remove a cluster node model from management by Cluster Manager, it is removed from the cluster container. If ADES Manager continues to manage the host, the model does not appear in the ADES Managed Hosts container automatically. To move the model into the ADES Managed Hosts container, cut and paste the model from the Lost and Found (or global collection, if applicable).

  NOTE
  Cluster node models that are removed from Cluster Manager management are moved to the Lost and Found (LostFound). An exception is when a cluster node is in a global collection, in which case, the model remains in the global collection.
- The Cluster Manager hierarchy synchronizes after the Lost and Found (LostFound) is emptied.

IBM PowerHA

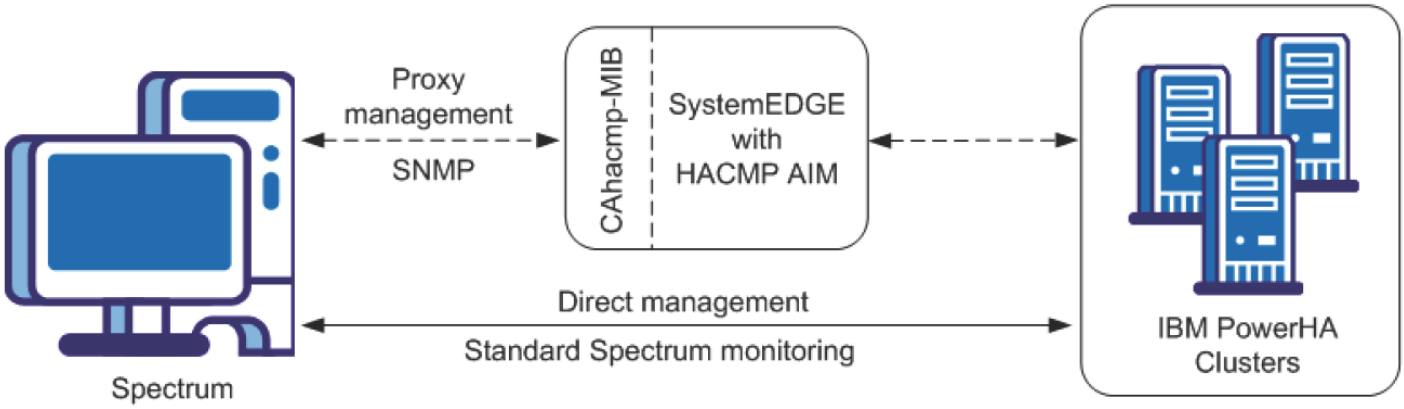
This section describes how DX NetOps Spectrum manages IBM PowerHA cluster environment including its models, the locater search, fault management, subviews in OneClick for visibility.

This section also provides the scenario that guides you in setting up Cluster Manager for IBM PowerHA.

Solution Architecture for IBM PowerHA

DX NetOps Spectrum gathers information about your IBM PowerHA cluster environment using two different methods. As with other DX NetOps Spectrum-managed devices, Cluster Manager uses standard DX NetOps Spectrum monitoring. In addition, Cluster Manager also retrieves specialized information for your IBM PowerHA environment from a proxy manager, the HACMP AIM.

The following diagram shows how DX NetOps Spectrum gathers information about your IBM PowerHA cluster environment:



The SystemEDGE agent with the HACMP AIM resides on its own host. This host is referred to as the IBM PowerHA Cluster Manager. The HACMP AIM obtains information from the IBM PowerHA cluster environment and writes this data to a CA-developed MIB (CAhacmp-MIB). DX NetOps Spectrum then uses SNMP to retrieve this information from the MIB and uses it to model and monitor your IBM PowerHA cluster components in OneClick.

Cluster Manager can support multiple HACMP AIMs either within a single DX NetOps Spectrum or distributed across multiple landscapes.

NOTE
For more information about the HACMP MIB, see the *CA Virtual Assurance for Infrastructure Manager* documentation.

How to Set Up Cluster Manager for IBM PowerHA

The following diagram shows the steps that are required for a DX NetOps Spectrum administrator to set up Cluster Manager to monitor IBM clusters:

Follow these steps:

1. Install DX NetOps Spectrum.
2. Install SystemEDGE agent with HACMP AIM.
3. Discover and Model Your IBM PowerHA Environment.

Install DX NetOps Spectrum

Cluster Manager is included in all DX NetOps Spectrum extraction keys. When you install DX NetOps Spectrum, the Cluster Manager components are automatically installed.

Follow this step:

- Install 9.2.3 or later.

  WARNING
  Do not install the SpectroSERVER on a host that Cluster Manager is going to manage.

  NOTE
  For specific installation instructions, see the Fresh Install section.

Install the SystemEDGE Agent and HACMP AIM

After DX NetOps Spectrum has been installed, install and configure the proxy manager; for IBM clusters, the proxy manager is the HACMP AIM.

The HACMP AIM is a specialized SystemEDGE AIM and resides on its own host. This host is referred to as the IBM PowerHA Cluster Manager.

When configuring the HACMP AIM, you manually specify the IBM PowerHA clusters to manage. Although your implementation can consist of multiple HACMP AIMs, manage each cluster with a single HACMP AIM only.

Follow this step:

- Install the SystemEDGE agent and load and configure the HACMP AIM on a host other than where DX NetOps Spectrum is installed. Note the following requirements:
  - Install only a single AIM on a particular SystemEDGE host.
  - Do not install the SystemEDGE agent and HACMP AIM on a node that Cluster Manager is going to manage.
  - Register each cluster and cluster node with a single HACMP AIM only.

  After the HACMP AIM has been successfully installed and configured, it begins gathering data for its managed components. This information is made available in the MIB.
  You can now discover and model your IBM cluster environment in DX NetOps Spectrum.

Discover and Model Your IBM PowerHA Environment

After you have installed the necessary components, discover and model any entities in your IBM PowerHA cluster environment that Cluster Manager is going to manage.

Follow these steps:

1. Run a Discovery for modeling the IBM PowerHA Cluster Manager and connecting devices.
2. (Optional) Upgrade the SystemEDGE model, if necessary.

   NOTE
   This step is required only if the SystemEDGE host has been modeled in DX NetOps Spectrum before installing the HACMP AIM on the agent.
3. Let Cluster Manager discovery run.

Run DX NetOps Spectrum Discovery to Model the IBM PowerHA Cluster Manager and Connecting Devices

After the SystemEDGE agent and HACMP AIM are set up, model the IBM PowerHA Cluster Manager and any connecting devices in DX NetOps Spectrum. You can use standard DX NetOps Spectrum Discovery to do the following actions:

- Model the IBM PowerHA Cluster Manager, which must be modeled with a read/write community string.
- Model the necessary upstream routers and switches of your IBM PowerHA cluster environment so that connections from the cluster models can later be established.

  WARNING
  Do not include cluster nodes. Clusters, cluster nodes, resource groups, and resources are discovered and modeled automatically using information from the AIM.

NOTE
For details about how to perform a Discovery, see the Modeling and Managing Your IT Infrastructure section.

Gather the correct community strings, IP addresses, and port numbers for any SNMP agents that run on a nonstandard port. Note the following guidelines when configuring your Discovery parameters:

- Include IP addresses for all IBM PowerHA Cluster Managers and interconnecting switches and routers.
- Model the IBM PowerHA Cluster Manager with a read/write community string. If you are modeling the IBM PowerHA Cluster Manager in this Discovery, place its community string appropriately in the SNMP Information ordered list. Alternatively, you can change the community string for the IBM PowerHA Cluster Manager to its read/write value after the discovery.
- Determine pingable MAC addresses during connectivity mapping by using the "ARP Tables for Pingables" option.

  NOTE
  Using this option can increase the time Discover Connections takes to run.
- Add any nonstandard SNMP ports using Advanced Options.

Discovery creates models for the following entities and adds them to your network topology in DX NetOps Spectrum:

- IBM PowerHA Cluster Manager.

  NOTE
  If the Discovery process did not assign the read/write community string to this model, update this setting manually. Use the DX NetOps Spectrum Modeling Information subview for the model.
- The upstream switches and routers that connect the IBM PowerHA cluster nodes to your network.

When Discovery has completed and these models exist in DX NetOps Spectrum, Cluster Manager discovery begins.

NOTE
Instead of using standard DX NetOps Spectrum Discovery, you can manually model your IBM PowerHA Cluster Manager by IP address or host name. If you do, model the upstream devices first (since modeling the IBM PowerHA Cluster Manager automatically triggers a Cluster Manager discovery). Modeling in the proper order allows the correct creation of connections in the topology between your cluster nodes and the remainder of your network. For more information, see the Modeling and Managing Your IT Infrastructure section.

Upgrade the SystemEDGE Host Model (If Necessary)

If the SystemEDGE host model was created before loading the HACMP AIM on the agent, the existing model is not compatible with Cluster Manager. Upgrade the SystemEDGE host (Host_systemEDGE) model so that Cluster Manager can access the HACMP AIM capabilities in SystemEDGE.

To upgrade the SystemEDGE host model, right-click the model and select Reconfiguration, Reconfigure Model.

The SystemEDGE host model is upgraded to support the HACMP AIM.

NOTE
You can also send a reconfigure model action to the SystemEDGE agent using CLI. For instructions on how to send a reconfigure model action to the SystemEDGE agent, see the Modeling and Managing Your IT Infrastructure section.

Cluster Manager Discovery

Cluster Manager discovery is the automatic discovery and modeling process within DX NetOps Spectrum of cluster components. The IBM PowerHA Cluster Manager initiates this process.

With communication between DX NetOps Spectrum and the HACMP AIM established, Cluster Manager gathers information about your IBM PowerHA environment from the HACMP AIM. A list of cluster nodes is passed to AutoDiscovery for modeling. For cluster node models, an SNMP-managed model is created if an SNMP agent exists on the host; otherwise, an ICMP (Pingable) model is created.

New cluster-related models appear in the Cluster Manager hierarchy in the Explorer view and are placed into new cluster containers in the topology view. Connections to any upstream devices are made.

> **NOTE**
> If a cluster node is already modeled in your DX NetOps Spectrum-managed network before Cluster Manager discovery, it is not modeled again. However, the model is included in the cluster container topology.

After the initial modeling, Cluster Manager discovery runs automatically at a frequency that is based on the IBM PowerHA Cluster Manager model poll cycle. During subsequent Cluster Manager discoveries, the modeling within DX NetOps Spectrum is updated with any changes in your cluster environment.

Models Created for IBM PowerHA

Cluster Manager provides several models to represent the components of your IBM PowerHA cluster environment, as follows:

- IBM PowerHA Cluster Manager

  Model Type: Host_systemEDGE
  Represents the host that contains the HACMP AIM. The HACMP AIM monitors the IBM PowerHA cluster elements (clusters, nodes, resource groups, and resources) in your environment.
- IBM PowerHA Cluster

  Model Type: ClusterIBMCluster
  Contains cluster node and resource group models that belong to the cluster. You cannot add or remove models from a cluster container, and you cannot destroy the container itself. When possible, this container model is created alongside the IBM PowerHA Cluster Manager model.

  > **NOTE**
  > If the IBM PowerHA Cluster Manager is a virtual machine, the cluster container is placed in the same topology as the physical host container.
- IBM PowerHA Cluster Node

Represents a cluster node in an IBM PowerHA cluster environment. Cluster nodes are modeled as SNMP-managed elements when possible. A cluster node can be active or inactive.
An active node has resource groups currently running on it and is represented with a solid (nontransparent) icon. An inactive node does not have any resource groups and is represented with a faded (transparent) icon. When resource groups fall over from one node to another, changing the state of the node, the icon transitions automatically.

NOTE
Unlike a model in maintenance mode or hibernation mode, the inactive node model is fully functional. Data is gathered for the node, and any alarm activity or events for the node are generated on the model.

- IBM PowerHA Cluster Resource Group

Model Type: ClusterIBMResourceGroup
Represents a resource group.

- IBM PowerHA Cluster Resource

Model Type: ClusterIBMResource
Represents a resource.

Custom Subviews for IBM PowerHA

Custom subviews in the Component Detail panel provide detailed information about the components in your cluster environment. You can view information specific to IBM PowerHA clusters by:

- IBM PowerHA Cluster Manager
- IBM PowerHA Cluster Component (Cluster, Cluster Node, Resource Group, Resource)

IBM PowerHA Cluster Manager

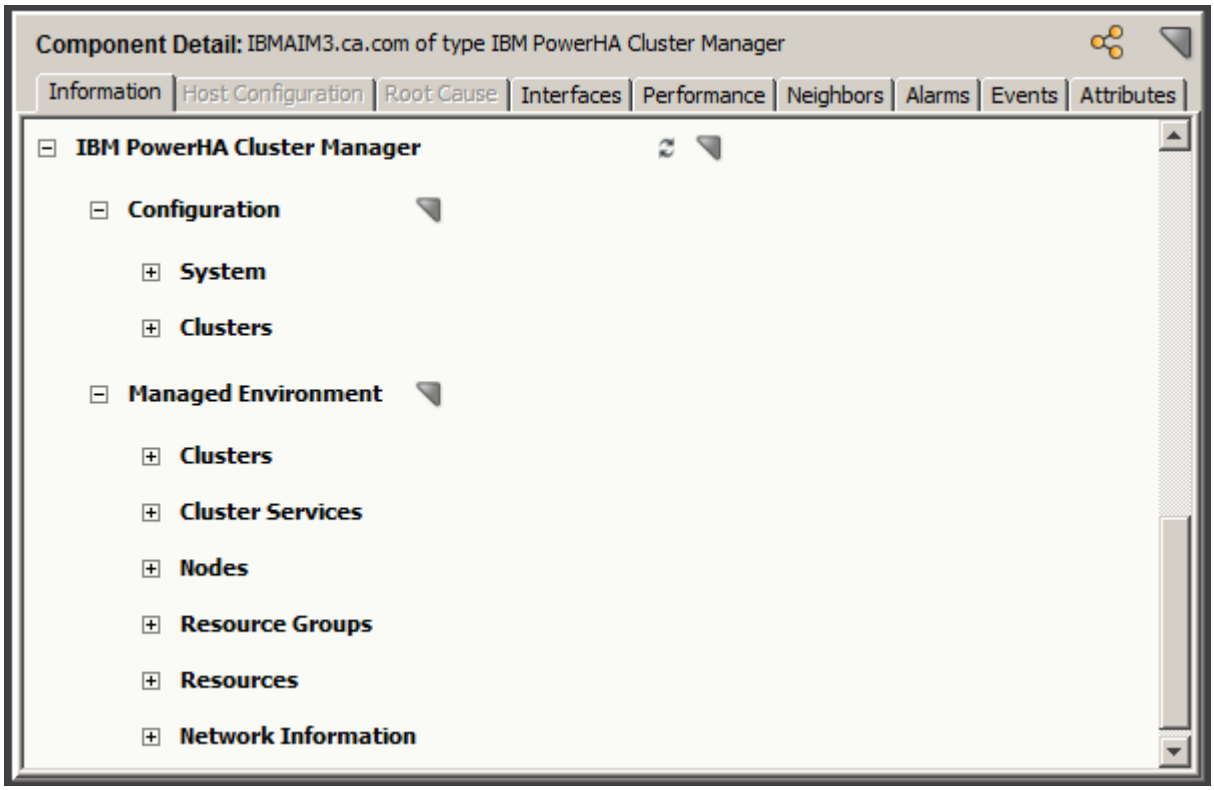
Using subviews that are provided for the IBM PowerHA Cluster Manager (HACMP AIM), you can view the following information:

- Information specific to the IBM PowerHA Cluster Manager host. Data includes the agent version, agent polling interval, and when the HACMP AIM MIB (CAhacmp-MIB) was last updated.
- List of clusters that have been registered to the AIM.
- Consolidated information about all cluster components that this HACMP AIM manages.

The following procedure describes how to view information for an IBM PowerHA Cluster Manager.

Follow these steps:

1. Select the IBM PowerHA Cluster Manager model in the Universe hierarchy or topology.
   The Component Detail panel displays information for the selected IBM PowerHA Cluster Manager.
2. In the Information tab in the Component Detail panel, expand the IBM PowerHA Cluster Manager subview.
   The expanded subview appears, as follows:



The following subviews are available for an IBM PowerHA Cluster Manager:

- **Configuration**
  Provides information specific to the IBM PowerHA Cluster Manager, including:
  – Information about the SystemEDGE agent including version, when the MIB was last updated, and polling interval. You can also modify the polling interval, as described in Controlling the HACMP AIM Polling Interval.
  – List of clusters that have been registered to this AIM and their respective readiness
- **Managed Environment**
  Provides consolidated information about all the entities that this AIM manages, including cluster components, services, resource groups, resources, and network information.

IBM PowerHA Cluster Component

You can view information for any of your clusters or cluster components (cluster node, resource group, resource) in your managed IBM PowerHA cluster environment. Views are tailored to the entity type, providing information that is specific to the component.

The following procedure describes how to view information for an IBM PowerHA cluster or cluster component.

Follow these steps:

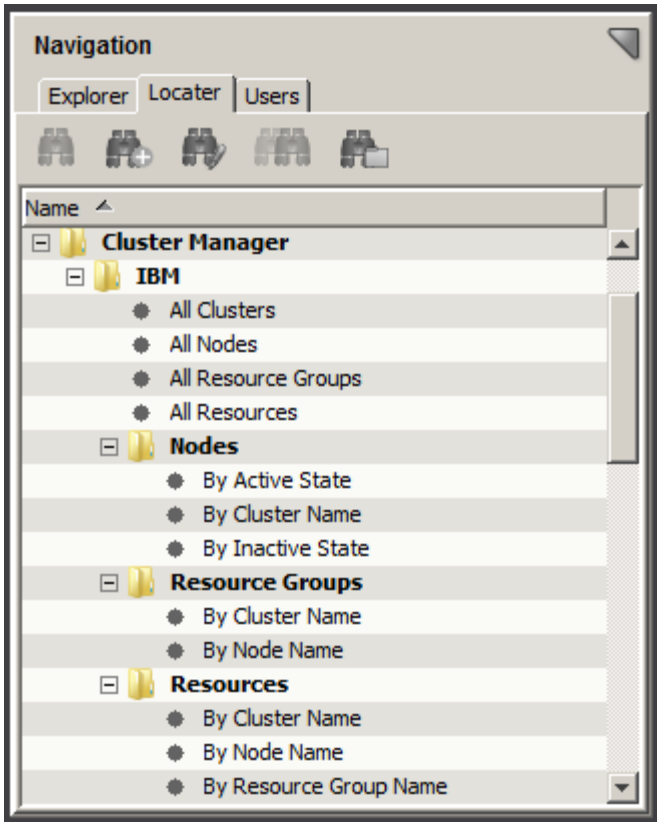
1. Select an IBM PowerHA Cluster, Cluster Node, Resource Group, or Resource model.
   The Component Detail panel displays information for the selected model.
2. In the Information tab in the Component Detail panel, expand the respective cluster-related subview for the model.
   The expanded subview appears, as follows, depending on the model type:
   – **Cluster Information**
   Provides general cluster information for the selected cluster model. Data includes cluster states, number of nodes, and instance name as registered in the AIM.
   – **Node Information**

Provides general node information such as the node state and the number of network interfaces it has. CPU usage and memory statistics are also provided.

- **Resource Group Information**
  Provides statistics such as the number of resources in the group, the node currently owning the group as well as the previous node. The number of startup, fallover, and fallback policies for the group are also provided.
- **Resource Information**
  Provides the resource type and index information.

Locater Searches for IBM PowerHA

You can use the Locater tab to run preconfigured searches. The search options are grouped under the Cluster Manager, IBM folder on the Locater tab, as shown:



These detailed searches can help you investigate information that is related to IBM PowerHA cluster entities that have been modeled in the DX NetOps Spectrum database.

NOTE
Only users with the appropriate privileges can access Cluster Manager searches. For more information, see the OneClick Administration section.

Alarms for IBM PowerHA

To alert you to problems within your IBM PowerHA cluster environment, DX NetOps Spectrum generates alarms. Quickly identifying any device faults helps you to maximize your system up-time and the reliability of your cluster environment and high availability applications. Alarms are created from information that is obtained from technology-specific traps

and polling. The following sections describe the Cluster Manager events and alarms for your IBM PowerHA cluster environment.

NOTE
To view specific event definitions that are related to Cluster Manager, use the Event Configuration application.

Traps for IBM PowerHA

DX NetOps Spectrum supports all traps that the HACMP AIM generates. An event is created for any trap activity and is reported initially on the IBM PowerHA Cluster Manager model. Some events are then forwarded to a corresponding cluster entity type (that is, the "destination" entity), depending on the type of trap.

The following table provides the traps and destination entity type and indicates whether the trap generates an alarm by default.

Trap Name	Trap OID	Alarm?	Destination Entity
hacmpAimInstanceAddedTrap	1.3.6.1.4.1.546.1.1.0.165800	No	Cluster Manager
hacmpAimInstanceRemovedTrap	1.3.6.1.4.1.546.1.1.0.165801	No	Cluster Manager
hacmpAimInstanceDataStatusChanged	1.3.6.1.4.1.546.1.1.0.165802	No	Cluster Manager
hacmpAimNodeAddedTrap	1.3.6.1.4.1.546.1.1.0.165803	No	Cluster Manager
hacmpAimNodeRemovedTrap	1.3.6.1.4.1.546.1.1.0.165804	No	Cluster Manager
hacmpAimResourceGroupAddedTrap	1.3.6.1.4.1.546.1.1.0.16580 5	No	Cluster Manager
hacmpAimResourceGroupRemovedTrap	1.3.6.1.4.1.546.1.1.0.165806	No	Cluster Manager
hacmpAimResourceGroupMigration	1.3.6.1.4.1.546.1.1.0.165807	No	Resource Group
hacmpAimResourceAddedTrap	1.3.6.1.4.1.546.1.1.0.165808	No	Cluster Manager
hacmpAimResourceRemovedTrap	1.3.6.1.4.1.546.1.1.0.165809	No	Cluster Manager
aggregateStateTrap*	1.3.6.1.4.1.546.1.1.0.20	Yes*	various*

* The aggregateStateTrap is a SystemEDGE trap. Alarms are generated for certain aggregateStateTrap conditions. For more information, see Self Monitors for IBM PowerHA.

NOTE
For more information about traps that the HACMP AIM generates, see the CA Virtual Assurance for Infrastructure Managers section. You can also use MIB Tools to view the traps in the "CAhacmp-MIB" MIB. For more information, see the Modeling and Managing Your IT Infrastructure section.

State Monitoring for IBM PowerHA

Cluster Manager monitors the state of various cluster components in your environment and obtains this information from the following sources:

- Polling the "CAhacmp-MIB" MIB. More than 100 objects are monitored regularly about the elements in your cluster environment. This information is updated in DX NetOps Spectrum according to the polling cycle. Cluster Manager derives pertinent information from these objects to create various events and alarms that provide insight into the health and status of your environment.
- Self-monitor traps. When installed, the HACMP AIM configures self monitors on the SystemEDGE agent which track various resources and activities of the managed cluster components. The monitors are threshold-based, and an aggregateState trap is sent when a threshold is violated. DX NetOps Spectrum then generates an event and, depending on the current severity state of the monitor, an applicable alarm. Data that is gathered from self monitors includes CPU or memory usage for a node.

Cluster Manager uses information from both sources to monitor the state of your cluster components. Alarms are generated and, when the condition has been corrected, cleared automatically. All state-based alarms are also user-clearable. When both trap and polling sources reveal the same activity, Cluster Manager identifies the overlap. A single alarm is created, with the alarm that polling generates taking precedence.

When a resource group moves from a primary node to a secondary node, an alarm occurs. When the resource group moves from the secondary node back to the primary, a new alarm is generated for the latest migration. The original alarm does not clear automatically, but it is user-clearable.

The following table lists the state-based alarm information by cluster component:

Entity	State	DX NetOps Spectrum Alarm Severity
Cluster	Up	Clear
Cluster	Down	Critical (Red)
Cluster	Unknown	Major (Orange)
Cluster	Not Configured	Critical (Red)
Cluster	Network state down*	Major (Orange)
Node	Up	Clear
Node	Down	Critical (Red)
Node	Joining	Event only
Node	Leaving	Event only
Node	Unknown	Major (Orange)
Node	High CPU Utilization*	Major (Orange)
Node	High Memory Utilization*	Major (Orange)
Node	Network interface state down*	Major (Orange)
Resource Group	Unknown	Major (Orange)
Resource Group	Online	Clear
Resource Group	Offline	Critical (Red)
Resource Group	Acquiring	Event only
Resource Group	Releasing	Event only
Resource Group	Error	Critical (Red)
Resource Group	Onlinesec	Clear
Resource Group	Acquiringsec	Event only
Resource Group	Releasingsec	Event only
Resource Group	Errorsec	Critical (Red)
Resource Group	Offline_due_to_failover	Minor (Yellow)
Resource Group	Off_line_due_to_parent_off	Critical (Red)
Resource Group	Unmanagedsec	Minor (Yellow)
Resource Group	Offline_due_to_lack_of_node	Critical (Red)
Resource Group	Unmanaged	Minor (Yellow)
Resource Group	Parent changes	Major (Orange)

* Alarms generated from self-monitor aggregateStateTrap.

Self Monitors for IBM PowerHA

Self monitors are threshold-based watches that are configured on the SystemEDGE agent. When installed, the HACMP AIM configures self monitors that are specific to the cluster environment. The HACMP AIM sets the initial severities and threshold values, but you can access and modify the values from within OneClick.

When a configured threshold is violated, the SystemEDGE agent sends the pertinent information to DX NetOps Spectrum using the aggregateStateTrap. DX NetOps Spectrum then generates an event and forwards the event to the respective entity.

For the following monitors only, DX NetOps Spectrum generates alarms by default:

- Node CPU Utilization
- Node Memory Utilization
- Network State
- Network Interface State

HACMP AIM State	DX NetOps Spectrum Alarm Severity
1: None/Unknown	Event only
2: OK	Clear
3: Warning	Clear
4: Minor	Minor (Yellow)
5: Major	Major (Orange)
6: Critical	Major (Orange)
7: Fatal	Major (Orange)

Controlling the HACMP AIM Polling Interval

Cluster Manager uses the HACMP AIM for discovery, modeling, and monitoring of your IBM PowerHA environment. The HACMP AIM has its own polling interval, which can be set from within DX NetOps Spectrum.

> **NOTE**
> For information about other HACMP AIM settings, see the *CA Virtual Assurance for Infrastructure Managers* documentation.

Follow these steps:

1. Select the IBM PowerHA Cluster Manager model that represents the HACMP AIM.
   The Component Detail panel displays information for the selected IBM PowerHA Cluster Manager.
2. In the Information tab in the Component Detail panel, expand the IBM PowerHA Cluster Manager, Configuration, System subview.
   The expanded System subview appears.
3. For the Agent Polling Interval, click set, modify the value, and press Enter.
   The polling interval for the HACMP AIM is updated.

Microsoft Cluster Service (MSCS)

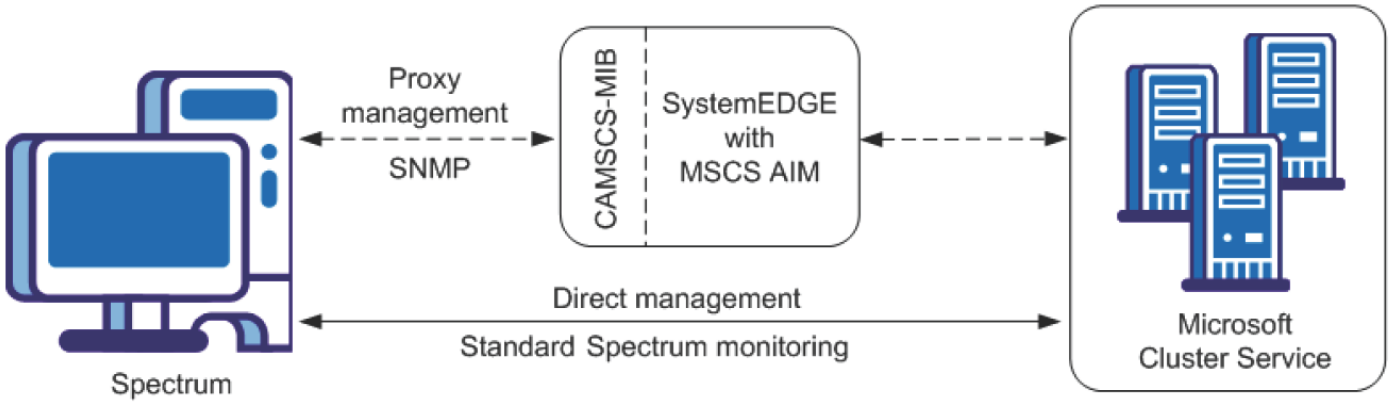
This section describes how DX NetOps Spectrum manages Microsoft Cluster Service (MSCS) including its models, the Locater search, fault management, subviews in OneClick for visibility.

This section also provides the scenario that guides you in setting up Cluster Manager for MSCS.

Solution Architecture for Managing MSCS

DX NetOps Spectrum gathers information about your Microsoft Cluster Service (MSCS) environment using two different methods. As with other DX NetOps Spectrum-managed devices, Cluster Manager uses standard DX NetOps Spectrum monitoring. In addition, Cluster Manager also retrieves specialized information for your MSCS environment from a proxy manager, the MSCS AIM.

The following diagram shows how DX NetOps Spectrum gathers information about your MSCS environment:



The SystemEDGE agent with the MSCS AIM resides on its own host. This host is referred to as the Microsoft Cluster Manager. The MSCS AIM obtains information from the MSCS environment and writes this data to a CA-developed MIB (CAMSCS-MIB). DX NetOps Spectrum then uses SNMP to retrieve this information from the MIB and uses it to model and monitor your MSCS cluster components in OneClick.

Cluster Manager can support multiple MSCS AIMs either within a single SpectroSERVER or distributed across multiple landscapes.

> **NOTE**
> For more information about the MSCS MIB, see the *CA Virtual Assurance for Infrastructure Managers* documentation.

How to Set Up Cluster Manager for MSCS

The following steps are required for a DX NetOps Spectrum administrator to set up Cluster Manager to monitor MSCS clusters:

1. Install DX NetOps Spectrum.
2. Install SystemEDGE agent with MSCS AIM
3. Discover and Model Your MSCS Environment.

Install DX NetOps Spectrum

Cluster Manager is included in all DX NetOps Spectrum extraction keys. When you install DX NetOps Spectrum, the Cluster Manager components are automatically installed.

Follow this step:

- Install 9.2.3 or later.

WARNING
Do not install the SpectroSERVER on a host that Cluster Manager is going to manage.

NOTE
For specific installation instructions, see the Fresh Install section.

Install the SystemEDGE Agent and MSCS AIM

After DX NetOps Spectrum has been installed, install and configure the proxy manager; for MSCS, the proxy manager is the MSCS AIM.

The MSCS AIM is a specialized SystemEDGE AIM and resides on its own host. This host is referred to as the Microsoft Cluster Manager.

When configuring the MSCS AIM, you manually specify the MSCS clusters to manage. Although your implementation can consist of multiple MSCS AIMs, manage each cluster with a single MSCS AIM only.

Follow this step:

- Install the SystemEDGE agent and load and configure the MSCS AIM on a host other than where DX NetOps Spectrum is installed. Note the following requirements:
  - Install only a single AIM on a particular SystemEDGE host.
  - Do not install the SystemEDGE agent and MSCS AIM on a node that Cluster Manager is going to manage.
  - Register each cluster and cluster node with a single MSCS AIM only.

  NOTE
  For more information, see the *CA Virtual Assurance for Infrastructure Managers Installation and Release Notes* section in C*A Virtual Assurance for Infrastructure Managers* documentation.

  After the MSCS AIM has been successfully installed and configured, it begins gathering data for its managed components. This information is made available in the MIB.
  You can now discover and model your MSCS environment in DX NetOps Spectrum.

Discover and Model Your MSCS Environment

After you have installed the necessary components, discover and model any entities in your MSCS environment that Cluster Manager is going to manage.

Follow these steps:

1. Run a DX NetOps Spectrum Discovery for modeling the Microsoft Cluster Manager and connecting devices.
2. (Optional) Upgrade the SystemEDGE model, if necessary.

   NOTE
   This step is required only if the SystemEDGE host has been modeled in DX NetOps Spectrum before installing the MSCS AIM on the agent.
3. Let the Cluster Manager discovery run, see Cluster Manager Discovery section on this page.

Run DX NetOps Spectrum Discovery to Model the Microsoft Cluster Manager and Connecting Devices

After the SystemEDGE agent and MSCS AIM are set up, model the Microsoft Cluster Manager and any connecting devices in DX NetOps Spectrum. You can use standard DX NetOps Spectrum Discovery to do the following actions:

- Model the Microsoft Cluster Manager, which must be modeled with a read/write community string.
- Model the necessary upstream routers and switches of your MSCS environment so that connections from the cluster models can later be established.

WARNING
Do not include cluster nodes. Clusters, cluster nodes, resource groups, and resources are discovered and modeled automatically using information from the AIM.

NOTE
For details about how to perform a Discovery, see the Modeling and Managing Your IT Infrastructure section.

Gather the correct community strings, IP addresses, and port numbers for any SNMP agents that run on a nonstandard port. Note the following guidelines when configuring your Discovery parameters:

- Include IP addresses for all Microsoft Cluster Managers and interconnecting switches and routers.
- Model the Microsoft Cluster Manager with a read/write community string. If you are modeling the Microsoft Cluster Manager in this Discovery, place its community string appropriately in the SNMP Information ordered list. Alternatively, you can change the community string for the Microsoft Cluster Manager to its read/write value after the discovery.
- Determine pingable MAC addresses during connectivity mapping by using the "ARP Tables for Pingables" option.

  NOTE
  Using this option can increase the time Discover Connections takes to run.
- Add any nonstandard SNMP ports using Advanced Options.

Discovery creates models for the following entities and adds them to your network topology in DX NetOps Spectrum:

- Microsoft Cluster Manager.

  NOTE
  If the Discovery process did not assign the read/write community string to this model, update this setting manually. Use the DX NetOps Spectrum Modeling Information subview for the model.
- The upstream switches and routers that connect the MSCS cluster nodes to your network.

When Discovery has completed and these models exist in DX NetOps Spectrum, Cluster Manager discovery begins.

NOTE
Instead of using standard DX NetOps Spectrum Discovery, you can manually model your Microsoft Cluster Manager by IP address or host name. If you do, model the upstream devices first (since modeling the Microsoft Cluster Manager automatically triggers a Cluster Manager discovery). Modeling in the proper order allows the correct creation of connections in the topology between your cluster nodes and the remainder of your network. For more information, see Modeling and Managing Your IT Infrastructure section.

Upgrade the SystemEDGE Host Model (If Necessary)

If the SystemEDGE host model was created before loading the MSCS AIM on the agent, the existing model is not compatible with Cluster Manager. Upgrade the SystemEDGE host (Host_systemEDGE) model so that Cluster Manager can access the MSCS AIM capabilities in SystemEDGE.

To upgrade the SystemEDGE host model, right-click the model and select Reconfiguration, Reconfigure Model.

The SystemEDGE host model is upgraded to support the MSCS AIM.

NOTE
You can also send a reconfigure model action to the SystemEDGE agent using CLI. For instructions on how to send a reconfigure model action to the SystemEDGE agent, see the Modeling and Managing Your IT Infrastructure section.

Cluster Manager Discovery

Cluster Manager discovery is the automatic discovery and modeling process within DX NetOps Spectrum of cluster components. The Microsoft Cluster Manager initiates this process.

With communication between DX NetOps Spectrum and the MSCS AIM established, Cluster Manager gathers information about your MSCS environment from the MSCS AIM. A list of cluster nodes is passed to AutoDiscovery for modeling. For cluster node models, an SNMP-managed model is created if an SNMP agent exists on the host; otherwise, an ICMP (Pingable) model is created.

New cluster-related models appear in the Cluster Manager hierarchy in the Explorer view and are placed into new cluster containers in the topology view. Connections to any upstream devices are made.

> **NOTE**
> If a cluster node is already modeled in your DX NetOps Spectrum-managed network before Cluster Manager discovery, it is not modeled again. However, the model is included in the cluster container topology.

After the initial modeling, Cluster Manager discovery runs automatically at a frequency that is based on the Microsoft Cluster Manager model poll cycle. During subsequent Cluster Manager discoveries, the modeling within DX NetOps Spectrum is updated with any changes in your cluster environment.

Models Created for MSCS

Cluster Manager provides several models to represent the components of your MSCS environment, as follows:

- Microsoft Cluster

  Manager
  Model Type: Host_systemEDGE
  Represents the host that contains the MSCS AIM. The MSCS AIM monitors the MSCS elements (clusters, nodes, resource groups, and resources) in your environment.
- Microsoft

  Cluster
  Model Type: ClusterMSCSCluster
  Contains cluster node and resource group models that belong to the cluster. You cannot add or remove models from a cluster container, and you cannot destroy the container itself. When possible, this container model is created alongside the Microsoft Cluster Manager model.

  > **NOTE**
  > If the Microsoft Cluster Manager is a virtual machine, the cluster container is placed in the same topology as the physical host container.
- Microsoft Cluster

  Node
  Represents a cluster node in an MSCS environment. Cluster nodes are modeled as SNMP-managed elements when possible. A cluster node can be active or inactive.

An active node has resource groups currently running on it and is represented with a solid (nontransparent) icon. An inactive node does not have any resource groups and is represented with a faded (transparent) icon. When resource groups fail over from one node to another, changing the state of the node, the icon transitions automatically.

NOTE
Unlike a model in maintenance mode or hibernation mode, the inactive node model is fully functional. Data is gathered for the node, and any alarm activity or events for the node are generated on the model.

- Microsoft Cluster Resource

Group
Model Type: ClusterMSCSResourceGroup
Represents a resource group.

- Microsoft Cluster

Resource
Model Type: ClusterMSCSResource
Represents a resource.

Custom Subviews for MSCS

Custom subviews in the Component Detail panel provide detailed information about the components in your cluster environment. You can view information specific to MSCS clusters by:

- Microsoft Cluster Manager
- MSCS Component (Cluster, Cluster Node, Resource Group, Resource)

Microsoft Cluster Manager

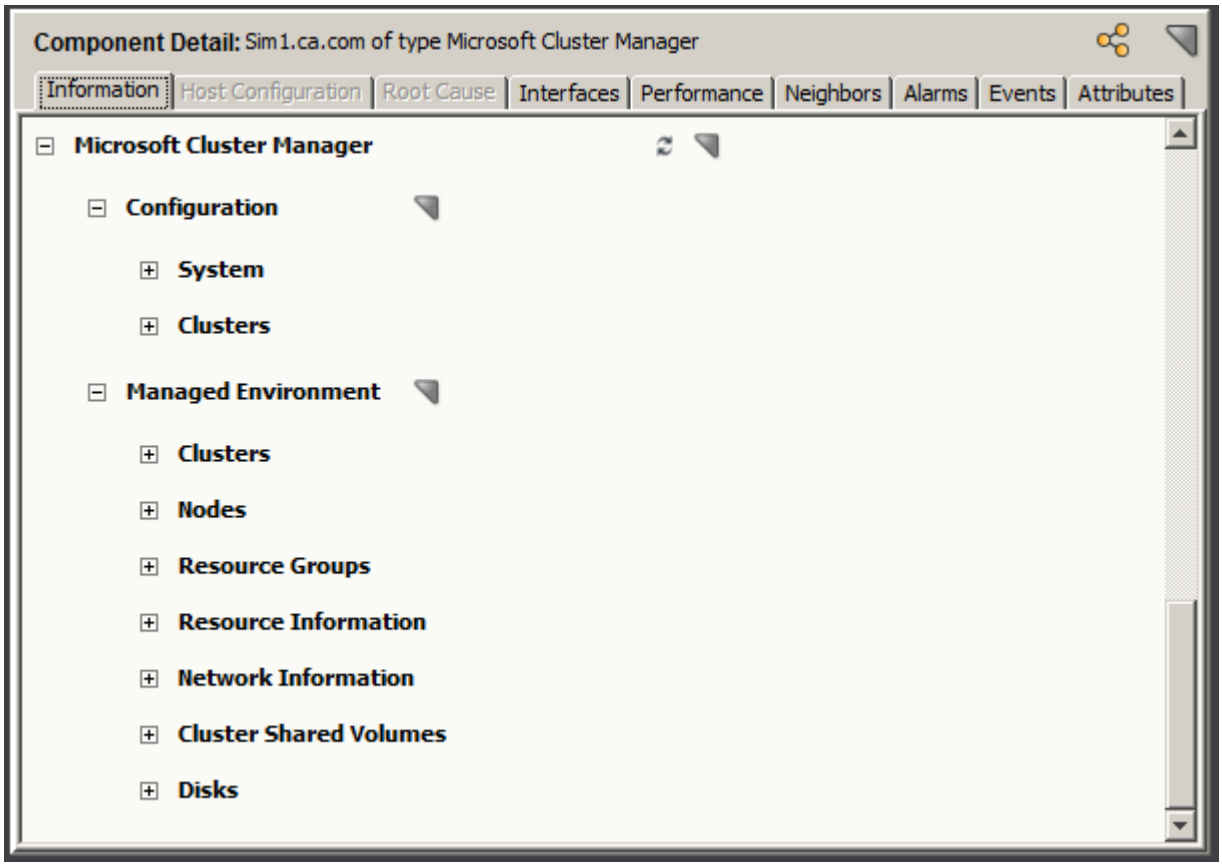
Using subviews that are provided for the Microsoft Cluster Manager (MSCS AIM), you can view the following information:

- Information specific to the Microsoft Cluster Manager host. Data includes the agent version, agent polling interval, and when the MSCS AIM MIB (CAMSCS-MIB) was last updated. You can also control the MSCS AIM polling interval from these views.
- List of clusters that have been registered to the AIM.
- Consolidated information about all cluster components that this MSCS AIM manages.

The following procedure describes how to view information for a Microsoft Cluster Manager.

Follow these steps:

1. Select the Microsoft Cluster Manager model in the Universe hierarchy or topology.
   The Component Detail panel displays information for the selected Microsoft Cluster Manager.
2. In the Information tab in the Component Detail panel, expand the Microsoft Cluster Manager subview.
   The expanded subview appears, as follows:



The following subviews are available for a Microsoft Cluster Manager:

- **Configuration**
  Provides information specific to the Microsoft Cluster Manager, including:
  - Information about the SystemEDGE agent including version, when the MIB was last updated, and polling interval. You can also modify the polling interval, as described in Controlling the MSCS AIM Polling Interval.
  - List of clusters that have been registered to this AIM and their respective readiness
- **Managed Environment**
  Provides consolidated information about all the entities that this AIM manages, including cluster components, resource groups, resources, network information, and storage devices.

MSCS Component

You can view information for any of your clusters or cluster components (cluster node, resource group, resource) in your managed MSCS environment. Views are tailored to the entity type, providing information that is specific to the component.

The following procedure describes how to view information for an MSCS cluster or cluster component.

Follow these steps:

1. Select a Microsoft Cluster, Cluster Node, Resource Group, or Resource model.
   The Component Detail panel displays information for the selected model.
2. In the Information tab in the Component Detail panel, expand the respective cluster-related subview for the model.
   The expanded subview appears, as follows, depending on the model type:
   - **Cluster Information**
     Provides cluster data including:

- The virtual IP address of the cluster
- The number of online and failed node resources
- Log level and the log file size
- Various timeout values
- Statistics on resources, crypto checkpoints, registry checkpoints, messages

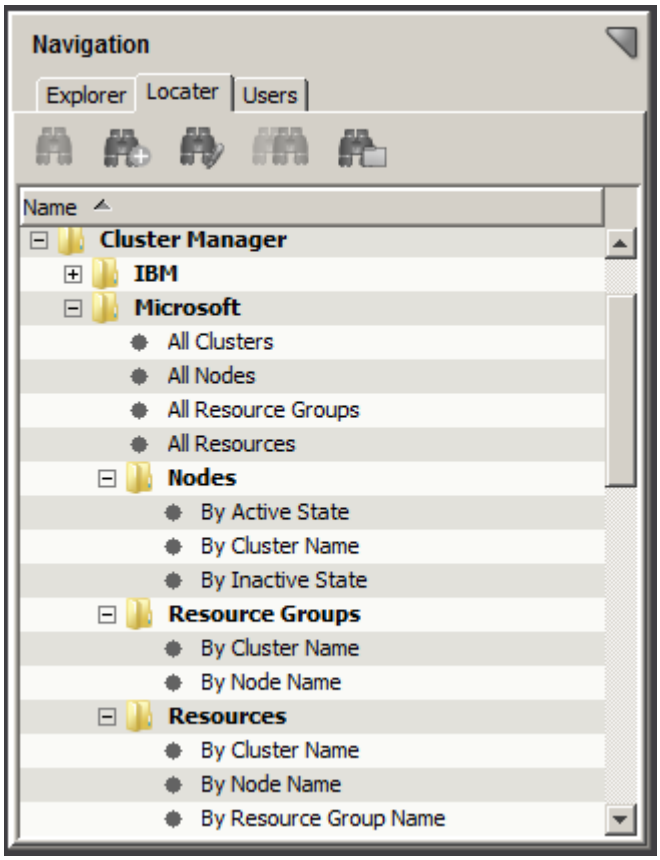
– **Node Information**
Provides cluster node data including:
- General node information including node state, installed Windows details, and the parent cluster
- Host CPU usage and memory statistics
- Data and message information

– **Resource Group Information**
Provides resource group data including:
- The state of the resource group
- The list of its preferred nodes
- Failback and failover threshold values

– **Resource Information**
Provides resource data including:
- The state of the resource
- Possible owners of the resource
- Various timeout, polling, and restart values

Locater Searches for MSCS

You can use the Locater tab to run preconfigured searches. The search options are grouped under the Cluster Manager, Microsoft folder on the Locater tab, as shown:



These detailed searches can help you investigate information that is related to MSCS cluster entities that have been modeled in the DX NetOps Spectrum database.

NOTE
Only users with the appropriate privileges can access Cluster Manager searches. For more information, see the OneClick Administration section.

Alarms for MSCS

To alert you to problems within your MSCS environment, DX NetOps Spectrum generates alarms. Quickly identifying any device faults helps you to maximize your system up-time and the reliability of your cluster environment and high availability applications. Alarms are created from information that is obtained from technology-specific traps and polling. The following sections describe the Cluster Manager events and alarms for your MSCS environment:

NOTE
To view specific event definitions that are related to Cluster Manager, use the Event Configuration application.

Traps for MSCS

DX NetOps Spectrum supports all traps that the MSCS AIM generates. An event is created for any trap activity and is reported initially on the Microsoft Cluster Manager model. Some events are then forwarded to a corresponding cluster entity type (that is, the "destination" entity), depending on the type of trap.

The following table provides the traps and destination entity type and indicates whether the trap generates an alarm by default.

Trap Name	Trap OID	Alarm?	Destination Entity
mscsAimInstanceAddedTrap	1.3.6.1.4.1.546.1.1.0.165100	No	Cluster Manager
mscsAimInstanceRemovedTrap	1.3.6.1.4.1.546.1.1.0.165101	No	Cluster Manager
mscsAimInstanceDataStatusChanged	1.3.6.1.4.1.546.1.1.0.165102	No	Cluster Manager
mscsAimResourceGroupMigration	1.3.6.1.4.1.546.1.1.0.165103	No	Resource Group
aggregateStateTrap*	1.3.6.1.4.1.546.1.1.0.20	Yes*	various*

* The aggregateStateTrap is a SystemEDGE trap. Alarms are generated for certain aggregateStateTrap conditions. For more information, see Self Monitors for MSCS.

NOTE
For more information on MSCS traps, use MIB Tools to view the "CAMSCS-MIB" MIB. For more information, see the Modeling and Managing Your IT Infrastructure section.

State Monitoring for MSCS

Cluster Manager monitors the state of various cluster components in your environment and obtains this information from the following sources:

- Polling the CAMSCS-MIB. Hundreds of objects are monitored regularly about the elements in your cluster environment. This information is updated in DX NetOps Spectrum according to the polling cycle. Cluster Manager derives pertinent information from these objects to create various events and alarms that provide insight into the health and status of your environment.
- Self-monitor traps. When installed, the MSCS AIM configures self monitors on the SystemEDGE agent which track various resources and activities of the managed cluster components. The monitors are threshold-based, and an aggregateState trap is sent when a threshold is violated. DX NetOps Spectrum then generates an event and, depending on the current severity state of the monitor, an applicable alarm. Data that is gathered from self monitors includes CPU or memory usage for a node.

Cluster Manager uses information from both sources to monitor the state of your cluster components. Alarms are generated and, when the condition has been corrected, cleared automatically. All state-based alarms are also user-clearable. When both trap and polling sources reveal the same activity, Cluster Manager identifies the overlap. A single alarm is created, with the alarm that polling generates taking precedence.

The following details apply:

- The MSCS AIM does not provide a state for a cluster. The state is determined by pinging the virtual IP address of the cluster.
- When a resource group moves from a primary node to a secondary node, an alarm occurs. When the resource group moves from the secondary node back to the primary, a new alarm is generated for the latest migration. The original alarm does not clear automatically, but it is user-clearable.

The following table lists the state-based alarm information by cluster component:

Entity	State	DX NetOps Spectrum Alarm Severity
Cluster	Up	Clear
Cluster	Down	Critical (Red)
Node	Up	Clear
Node	Down	Critical (Red)

Node	Joining	Event only
Node	Paused	Event only
Node	Unknown	Major (Orange)
Node	High CPU Utilization*	Major (Orange)
Node	High Memory Utilization*	Major (Orange)
Resource Group	Unknown	Major (Orange)
Resource Group	Online	Clear
Resource Group	Offline	Critical (Red)
Resource Group	Failed	Critical (Red)
Resource Group	Partial_online	Minor (Yellow)
Resource Group	Pending	Event only
Resource Group	Parent changes	Major (Orange)
Resource	Unknown	Major (Orange)
Resource	Inherited	Event only
Resource	Initializing	Event only
Resource	Online	Clear
Resource	Offline	Major (Orange)
Resource	Failed	Critical (Major)
Resource	Pending	Event only
Resource	Online_Pending	Event only
Resource	Offline_Pending	Event only

* Alarms generated from self-monitor aggregateStateTrap.

Self Monitors for MSCS

For the following monitors only, DX NetOps Spectrum generates alarms by default:

- Node CPU Utilization
- Node Memory Utilization

MSCS AIM State	DX NetOps Spectrum Alarm Severity
1: None/Unknown	Event only
2: OK	Clear
3: Warning	Clear
4: Minor	Minor (Yellow)
5: Major	Major (Orange)
6: Critical	Major (Orange)
7: Fatal	Major (Orange)

Controlling the MSCS AIM Polling Interval

Cluster Manager uses the MSCS AIM for discovery, modeling, and monitoring of your MSCS environment. The MSCS AIM has its own polling interval, which can be set from within DX NetOps Spectrum.

NOTE
For information about other MSCS AIM settings, see the *CA Virtual Assurance for Infrastructure Managers* documentation.

Follow these steps:

1. Select the Microsoft Cluster Manager model that represents the MSCS AIM.
   The Component Detail panel displays information for the selected Microsoft Cluster Manager.
2. In the Information tab in the Component Detail panel, expand the Microsoft Cluster Manager, Configuration, System subview.
   The expanded System subview appears.
3. For the Agent Polling Interval, click set, modify the value, and press Enter.
   The polling interval for the MSCS AIM is updated.

Viewing and Configuring Events and Alarms

This section explains how to view and modify cluster events, event definitions using the Event Configuration utility in OneClick. This section also explains how to view and modify alarm correlations using the Condition Correlation Editor in OneClick. And how to view and modify self-monitor threshold values in OneClick.

How to View and Modify Cluster Manager Event Definitions

To identify the events that Cluster Manager uses, you can use the Event Configuration application in OneClick. Using this application, you can also modify the generated alarm severity that is associated with the event.

NOTE
Using default settings, Cluster Manager identifies any overlap when multiple monitoring methods reveal the same activity, raising only a single alarm. If you use Event Configuration to add custom alarming, duplicate alarms for the same activity can occur.

Follow these steps:

1. Select Tools, Utilities, Event Configuration.
   The Event Configuration window opens. The Navigation panel displays all events that are defined in your DX NetOps Spectrum installation.
2. Filter for events that apply to Cluster Manager. Using the Show field, enter any of the following event codes one at a time:
   - **0x01169b32 - 0x01169b39, 0x01169c** - Related SystemEDGE events
   - **0x0621** - Cluster Manager events
3. Select an event.
   Event details appear in the Contents panel.
4. (Optional) Use the Details panel to modify any parameter for the event, including alarm severity, and click Save.

NOTE
For more information, see the Event Configuration section.

How to View and Modify Cluster Manager Correlations

To view the correlations that Cluster Manager uses, use the Condition Correlation Editor application in OneClick.

Follow these steps:

1. Select Tools, Utilities, Condition Correlation Editor.

The Condition Correlation Editor opens to the Conditions tab by default. The Conditions tab displays all conditions that are defined in your DX NetOps Spectrum installation.

2. On the Conditions tab, enter **0x0621** in the Show field to display conditions that apply to Cluster Manager.
   Only the conditions that apply to Cluster Manager are displayed. The Condition Name identifies the cluster component and its state. A condition is the basic building block of a correlation. Set Event and Clear Event codes for each condition are also displayed.

   NOTE
   For alarm information associated with a displayed event code, use the Event Configuration application.
3. (Optional) Edit a condition to modify any default settings.
4. Select the Rules tab.
   A list of all correlation rules that are defined for your installation are displayed. The rule defines the relationship between two or more conditions when specific criteria are met.
5. Enter **cluster** in the Show field to filter for rules that apply to Cluster Manager.
6. (Optional) Sort the results by Symptom Condition(s) or Root Cause Condition by selecting the respective column heading.
7. Select a rule.
   The conditions that define the rule are displayed in the Rule Criteria tab.
8. (Optional) Edit a rule to modify any default settings.

   NOTE
   For more information, see the Condition Correlation section.

How to Change Cluster Node Down Alarm Correlation

When a cluster node fails, Cluster Manager correlates the Cluster Node Down alarm to the Contact Lost alarm with Contact Lost as the root cause. You can modify the correlation behavior using the Condition Correlation Editor so that the Cluster Node Down alarm is the root cause. You can also have no correlation and get both alarms.

Modify the Correlation Rule

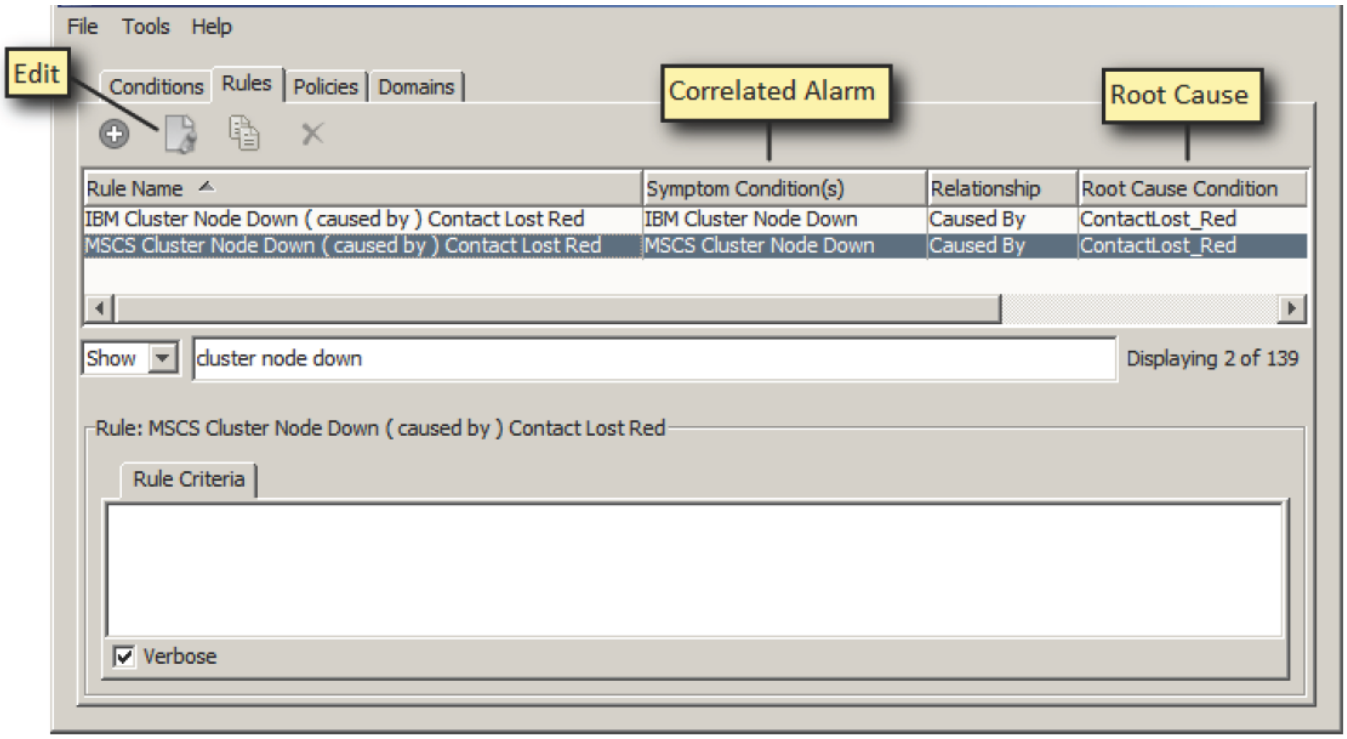
This procedure describes how to change the reported root cause when a cluster node fails by modifying the default correlation behavior.

Follow these steps:

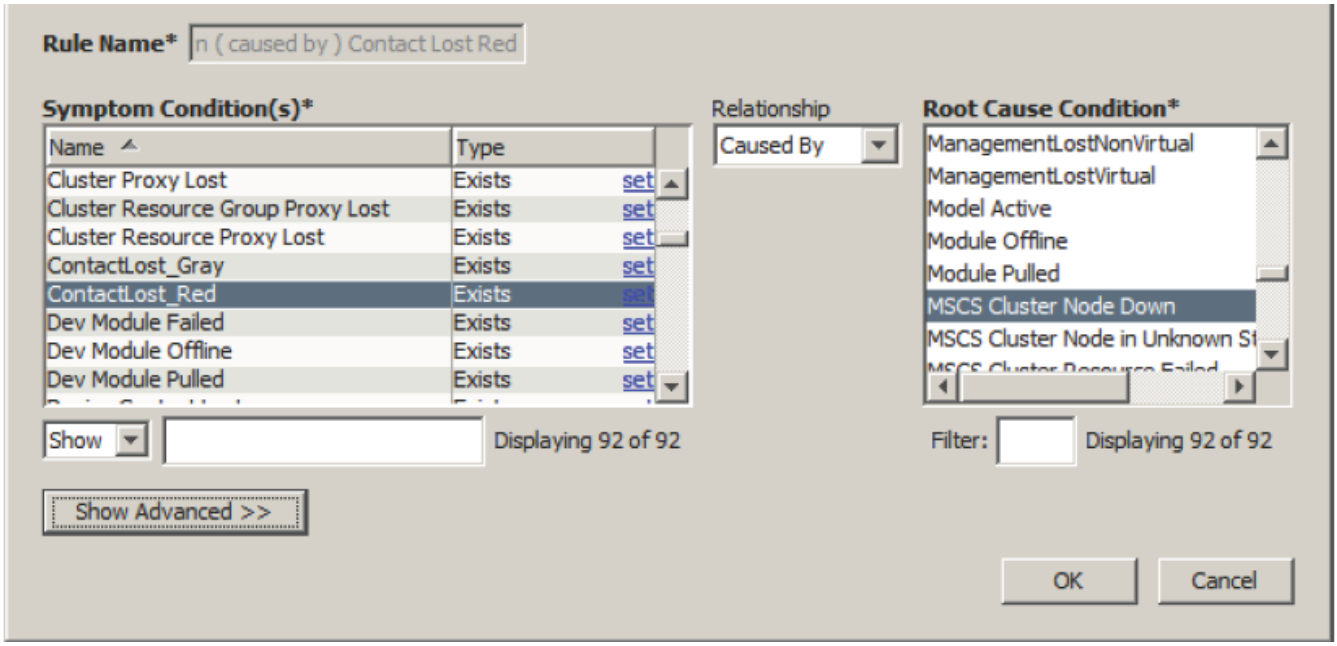
1. Select Tools, Utilities, Condition Correlation Editor.
   The Condition Correlation Editor opens.
2. Select the Rules Tab.
   A list of all correlation rules that are defined for your installation is displayed. The following rules apply to the Cluster Manager and the Cluster Node Down alarm:

   - IBM Cluster Node Down (caused by) Contact Lost Red
   - MSCS Cluster Node Down (caused by) Contact Lost Red

   As shown in the following example, the Symptom Condition of "Cluster Node Down" for each cluster solution correlates to the Root Cause Condition of ContactLost_Red.



3. Select the rule that you want to modify, and click the Edit button.
   The Edit Rule window opens.
4. Modify the values as follows:
   a. Select ContactLost_Red as the new Symptom Condition.
   b. Select the appropriate "Cluster Node Down" value as the Root Cause Condition. As defined in the original out-of-box rules, these values are:
      - IBM Cluster Node Down
      - MSCS Cluster Node Down



5. Click OK.
   Any new alarms for a cluster node failure use Cluster Node Down as the root cause. Existing alarms and symptoms do not change.

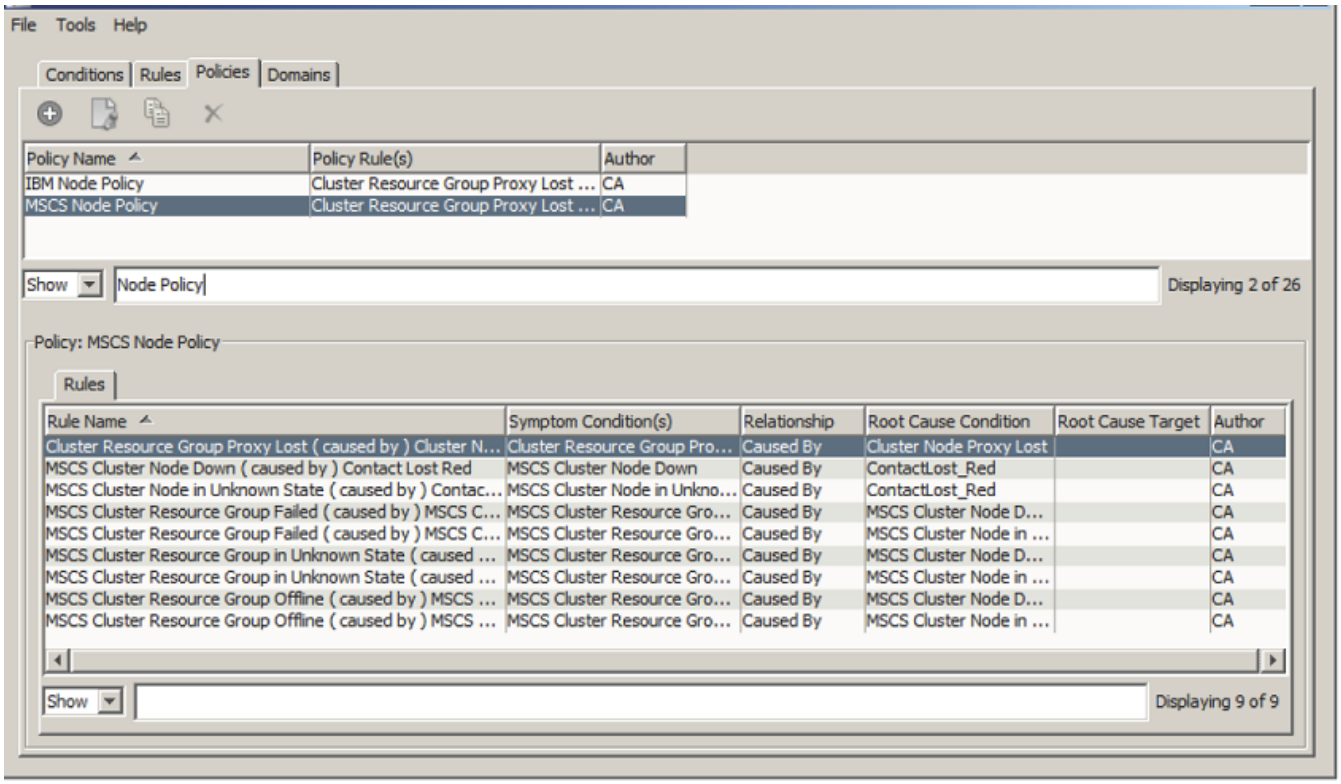
   NOTE
   For more information, see the Condition Correlation section.

Remove the Correlation Rule

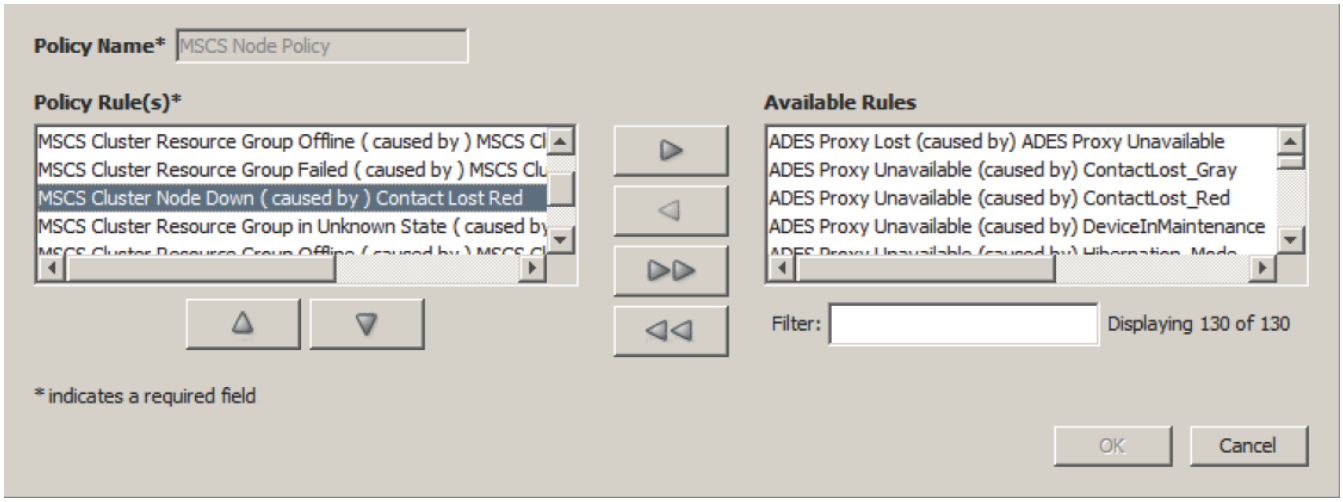
This procedure explains how to remove the correlation rule when a cluster node fails. As a result, both the Cluster Node Down and the Contact Lost alarms are reported.

Follow these steps:

1. Select Tools, Utilities, Condition Correlation Editor.
   The Condition Correlation Editor opens.
2. Select the Policies Tab.
   A list of all correlation policies that are defined for your installation is displayed. The following policies apply to Cluster Manager and the Cluster Node Down alarm:
   - IBM Node Policy
   - MSCS Node Policy



3. Select the policy that you want to modify, and click the Edit button.
   The Edit Policy window opens.
4. Move the appropriate "Cluster Node Down" rule to the right. As defined in the original out-of-box policies, these rules are:
   – IBM Cluster Node Down (caused by) Contact Lost Red
   – MSCS Cluster Node Down (caused by) Contact Lost Red



5. Click OK.
   The Cluster Node Down rule is no longer enabled in the policy. Any new cluster node failures result in both Cluster Node Down and Contact Lost alarms, and no correlation occurs. Existing alarms and symptoms do not change.

   NOTE
   For more information, see the Condition Correlation section.

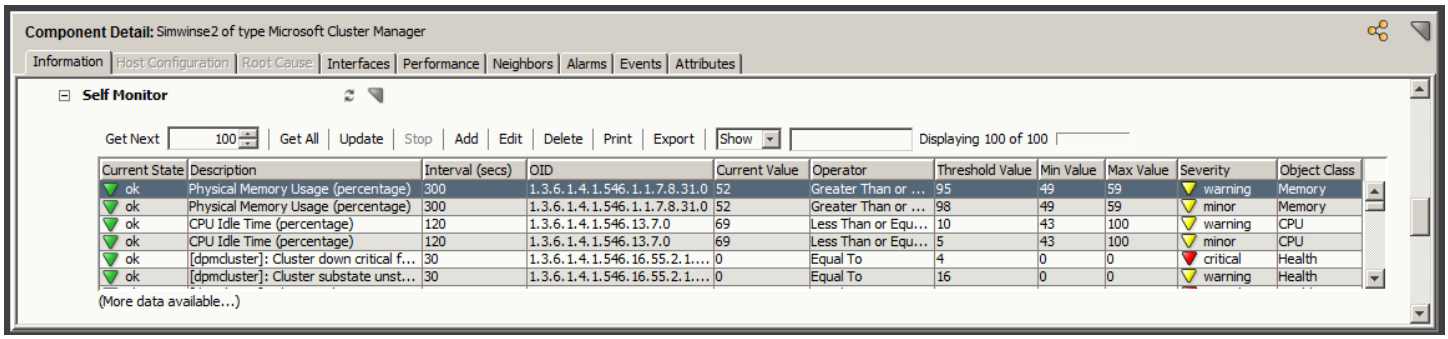
How to View and Modify Threshold Values

Cluster Manager uses self monitors that the cluster technology AIM configures on the SystemEDGE agent. The self monitors are threshold-based and track various resources and activities of the managed cluster components. When a threshold is violated, a DX NetOps Spectrum event and possibly an alarm is created. Configuration parameters for the self monitors are defined and stored on the SystemEDGE agent but can be modified from within DX NetOps Spectrum.

The following procedure describes how to modify self-monitor parameters for your cluster technology AIM from within DX NetOps Spectrum.

Follow these steps:

1. Select the Cluster Manager model in the Universe hierarchy or topology.
   The Component Detail panel displays information for the selected Cluster Manager.
2. In the Information tab in the Component Detail panel, expand the System Resources, Self Monitor subview.
   The expanded subview appears, as follows:



NOTE
Control the columns that appear by right-clicking the table column heading and using the Columns tab. You can also undock the subview.

- Select a row, and click Edit.
  The Edit Self Monitor Table Entry appears.
- Modify the Threshold Value and any other values of interest, and click OK.
  The new values are saved in the table and on the AIM.

Troubleshooting Cluster Manager

This section includes troubleshooting tips for some common problems that are encountered with Cluster Manager.

NetApp Cluster Certification Fixes

This release includes the following fixes:

- The NetApp Cluster Model type is modeled correctly as Modeltype_Name NetAppONTAPDev.
- The Model type Interface data reads from the property mib ie. 1.3.6.1.4.1.789.1.22.1.2
- On clicking the **Interface** tab, the Interface data is displayed.

Unsupported Cluster AIM Configuration

Symptom:

The following alarm message appears after attempting to model the cluster environment 'UNSUPPORTED CLUSTER AIM CONFIGURATION'.

Solution:

You can manage a cluster node by a single cluster technology AIM only. If you inadvertently attempt to manage a cluster node by multiple cluster technology AIMs, Cluster Manager issues this alarm on the cluster model. Children are not created for the cluster model.

Check your AIM configurations. Modify the configuration of your AIMs so that each cluster and cluster node is registered with a single AIM only.

NOTE
For more information, see the *CA Virtual Assurance for Infrastructure Managers* documentation.

Connections Do Not Appear in Topology

Symptom:

Cluster nodes do not show connections to other devices in the OneClick topology view.

Solution:

To produce connections between your cluster nodes and other elements in your network, any connecting devices must be modeled before the cluster nodes are modeled. When discovering and modeling your environment, run a standard DX NetOps Spectrum Discovery first to model upstream routers and switches. Then, Cluster Manager discovery can run, creating models and connections for the cluster components.

Follow these steps:

1. Verify that devices such as routers and switches that are upstream from your cluster nodes are modeled. If not, run a standard DX NetOps Spectrum Discovery to model these connecting devices.
2. If the connecting devices are modeled after your cluster environment is modeled, run Discover Connections on each of the affected devices.

NOTE
For information about Discover Connections, see the Modeling and Managing Your IT Infrastructure section.

Virtual Host Manager

Virtual Host Manager is an application that is provided with DX NetOps Spectrum that models and monitors the health of your virtual network environment. With this application, you can view details about your virtual networking components and the relationships between your physical and virtual components.

This broad view helps you better monitor the health of your network infrastructure, preventing service interruptions to your virtual components. Monitoring your virtual environment, such as monitoring resource utilization on hosts and virtual devices, can help you identify potential performance issues. Virtual Host Manager also helps you pinpoint and effectively troubleshoot problems within your entire network by applying DX NetOps Spectrum fault isolation techniques to virtual environments.

A key challenge when monitoring your virtual environment is keeping the data updated. Virtual environments are designed to optimize resource allocation as needed, so the relationship between the virtual and physical networks can change rapidly. Virtual Host Manager keeps up with these changes and continuously monitors the current state of your virtual network to detect any changes.

Who Should Use Virtual Host Manager

Multiple vendors provide virtual technology solutions. Virtual Host Manager is intended for DX NetOps Spectrum users who create and manage virtual environments. Virtual Host Manager allows the user to monitor the fault and performance of both their physical and virtual network entities.

Virtual Technologies Supported by Virtual Host Manager

Virtual Host Manager can model and manage virtual networks that are created with the following virtual network technologies:

- VMware vCenter Server (part of VMware Infrastructure and vSphere)
- Microsoft Hyper-V
- IBM logical partitions (LPARs)
- Huawei SingleCLOUD

System Requirements for Virtual Host Manager

Virtual Host Manager is an application that works within DX NetOps Spectrum when all required components are configured properly. Virtual Host Manager requires the following components by solution.

VMware

- Release 9.2.3 or later
- VMware vCenter Server
- Latest CA eHealth SystemEDGE agent with vCenter Server AIM

Hyper-V

- Release 9.2.3 or later
- CA eHealth SystemEDGE agent with Hyper-V AIM installed on each physical Microsoft Hyper-V server

IBM LPAR

- Release 9.2.3 or later
- CA eHealth SystemEDGE agent with IBM LPAR AIM installed on a Windows server separate from the HMC managing the IBM LPARs

Huawei SingleCLOUD

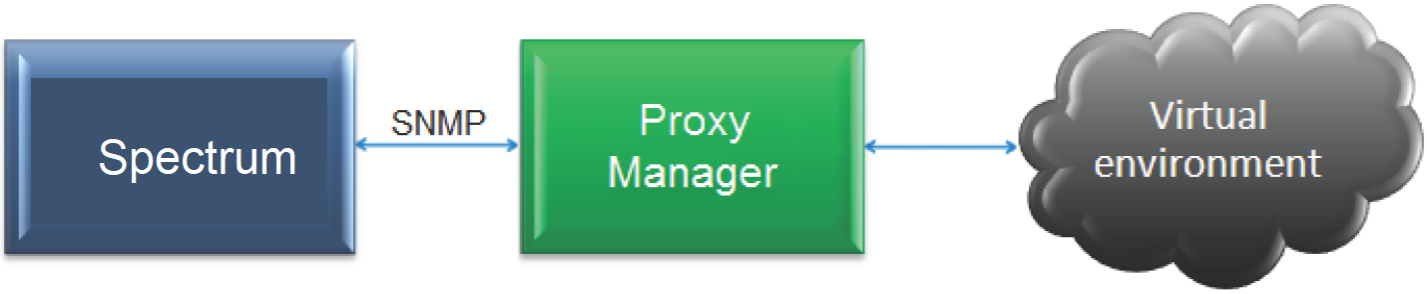
- Release 9.2.3 or later
- CA Mediation Manager with Huawei SingleCLOUD Device Pack

> **NOTE**
> For more information about the CA eHealth SystemEDGE agent and AIM system requirements, see *CA Virtual Assurance for Infrastructure Managers Implementation.* For more information about CA Mediation Manager, see the CA Mediation Manager documentation.

How Virtual Host Manager Works

Virtual Host Manager monitors your virtual network entities seamlessly beside your physical network entities within DX NetOps Spectrum. You get a full view of your network, which facilitates troubleshooting for both types of entities. Although your virtual network entities behave like physical components, the process for monitoring those entities differs from the general DX NetOps Spectrum monitoring process. Understanding how this process works can help you locate and resolve networking issues that are related to your virtual network.

DX NetOps Spectrum typically contacts an SNMP agent on your network devices to gather information. However, some network devices do not have an SNMP agent installed. Without an SNMP agent, it is difficult to gather information that is needed for monitoring status and pinpointing issues using fault isolation. Virtual Host Manager extends basic DX NetOps Spectrum functionality, using a proxy manager to gather the needed information, as shown in the following diagram:



The process to gather information about your virtual network environment is as follows:

1. The proxy manager communicates directly with entities in your virtual environment.

   NOTE
   The proxy manager resides on a server in your network. The location of the server depends on the virtual technology.
2. Using SNMP, DX NetOps Spectrum retrieves this information from the proxy manager and uses it to model and monitor your virtual entities.

Depending on the solution, Virtual Host Manager uses either of the following proxy managers, which are described in the following sections:

- CA eHealth SystemEDGE agent with a CA Virtual Assurance for Infrastructure Managers AIM module
- CA Mediation Manager with a solution-specific device pack

CA eHealth SystemEDGE Agent with CA Virtual Assurance for Infrastructure Managers AIMs

The following CA Virtual Assurance for Infrastructure Managers AIMs work with Virtual Host Manager:

- **vCenter Server AIM:** Provides the capabilities for managing and monitoring systems that are under VMware vCenter Server control. The AIM communicates directly with vCenter Server software to get an entire view of all ESX servers that the associated VMware vCenter Server manages.
- **Hyper-V AIM:** Provides the capabilities for monitoring VMs that are under Hyper-V Server control. The Microsoft Hyper-V AIM requires the CA eHealth SystemEDGE agent on the Microsoft Hyper-V server. The Microsoft Hyper-V AIM communicates with the Microsoft Hyper-V server through WMI. The Microsoft Hyper-V AIM must reside on the Microsoft Hyper-V server to monitor virtual machines.
- **IBM LPAR AIM:** Provides the capabilities for monitoring IBM LPARs managed by the HMC. The IBM LPAR AIM requires the CA eHealth SystemEDGE agent running on a Windows server separate from the HMC. The IBM LPAR AIM uses SSH to communicate with the HMCs, gathering information from the HMC to monitor the IBM LPAR instances.

CA Mediation Manager

The following CA Mediation Manager Device Pack is used with Virtual Host Manager:

- **Huawei SingleCLOUD**
  Provides the capabilities for monitoring the Huawei SingleCLOUD platform. CA Mediation Manager communicates directly with Huawei SingleCLOUD GalaX to obtain information about the Huawei HyperVisor Universal Virtualization Platform (UVP).

Overlapping Virtual Technologies

Your virtual environment has "overlapping" technologies when either of the following conditions exist:

- When two or more virtual technologies are used together in your environment
- When the same virtual technology is nested together

Virtual Host Manager does *not* support overlapping technologies that are modeled within a single SpectroSERVER. The following configuration represents overlapping virtual technologies:

DX NetOps Spectrum

- IBM LPAR AIM running on a VMware virtual machine or Hyper-V virtual machine

When DX NetOps Spectrum discovers an unsupported configuration between virtual technologies, the following behavior occurs:

- During initial modeling of a virtual technology manager, DX NetOps Spectrum prevents the creation of the technology folder. A minor alarm is generated, alerting you to the unsupported configuration.
- When a virtual technology manager monitors the same device that another manager is managing currently, DX NetOps Spectrum creates duplicate models for that device.

If you model the overlapping virtual technology manager on a separate SpectroSERVER, then Virtual Host Manager *can* support the overlapping technology managers.

Virtual Device Management and Multiple DX NetOps Spectrum AIM Solutions

When managing a device by multiple DX NetOps Spectrum AIM solutions, a defined ranked order of management applies, as follows:

1. Virtual Host Manager
2. Cluster Manager
3. Other technologies (such as Active Directory and Exchange Server Manager)

When a host with a CA eHealth SystemEDGE agent is already modeled in , Virtual Host Manager recognizes the model. A duplicate model is not created. Instead, Virtual Host Manager pulls the existing model into its own management, applying the rules for each solution using the ranked order.

For example, when both Virtual Host Manager and Cluster Manager are managing a device, model parameters that Virtual Host Manager assigns are used. Examples of these parameters include the model name, IP address, and MAC address.

When a solution no longer manages a device, the rules of the remaining solutions are reapplied in the ranked order. Typically, any changes are made at the next polling cycle.

The defined order of management also affects how models appear in the Universe topology. Because Virtual Host Manager is the highest in the management ranking, all virtual devices appear in the appropriate virtual host containers automatically.

For more information, see Cluster Manager and Active Directory and Exchange Server Manager.

Install Virtual Host Manager

This section describes the basic information that is required to install and begin using Virtual Host Manager. The information in this section applies to all virtual technologies supported by Virtual Host Manager.

How to Install Virtual Host Manager

When you install DX NetOps Spectrum, the Virtual Host Manager components are automatically installed and available for use. However, Virtual Host Manager is operable only after you also install and configure the appropriate proxy manager for your solution. For Huawei SingleCLOUD, use CA Mediation Manager. For all other supported technologies, use the CA Virtual Assurance for Infrastructure Managers AIM of the CA eHealth SystemEDGE agent.

To manage your virtual devices, DX NetOps Spectrum must be able to contact the proxy manager. And the proxy manager must be able to communicate with your network devices.

To install Virtual Host Manager, complete these tasks:

1. Install the appropriate proxy manager:

- For VMware, Hyper-V, and IBM LPAR solutions, install the CA eHealth SystemEDGE agent and load the appropriate CA Virtual Assurance for Infrastructure Managers AIM. Use the appropriate location for your virtual technology, as follows:
  - VMware: Install on a separate server that can contact vCenter remotely.
  - Hyper-V: Install on each Hyper-V Host.
  - IBM LPAR: Install on a Windows server separate from the HMC that manages the IBM LPARs.

    NOTE
    Monitor only one instance of the IBM LPAR Host with the IBM LPAR AIM. Do not manage a single IBM LPAR Host with multiple HMCs. Monitoring more than one instance can result in duplicate models in DX NetOps Spectrum.

  For installation instructions and more information about the AIM for your virtual technology, see *CA Virtual Assurance for Infrastructure Managers* documentation.
- For Huawei SingleCLOUD, install and configure CA Mediation Manager and the Huawei SingleCLOUD Device Pack. Do not install the CAMM components on the same server where DX NetOps Spectrum is installed.

  WARNING
  When configuring the Huawei SingleCLOUD Device Pack, you set the virtual IP addresses. The primary IP address of the device or virtual machine where the CAMM Presenter is installed cannot be used as a virtual IP address.

  For more information, see the CA Mediation Manager documentation.

2. Install DX NetOps Spectrum with Virtual Host Manager included.

   WARNING
   Do not install SpectroSERVER on a virtual machine (VM) that is managed by Virtual Host Manager. DX NetOps Spectrum identifies that SpectroSERVER machine as the VNM model. As a result, selecting that VM in the navigation hierarchy of OneClick displays the universe topology under that VM. This phenomenon continues in an indefinite loop.

For specific installation instructions, see Installation .

You can now model your virtual network in DX NetOps Spectrum.

How to Model Your Environment When Using Multiple AIM Solutions

Depending on your environment, you can use Virtual Host Manager with other DX NetOps Spectrum AIM solutions to manage your infrastructure. Some configurations, such as the following examples, require multiple solutions for comprehensive management:

- A cluster node is a virtual machine.
- An Active Directory or Exchange Server host is a virtual machine.

Each of the DX NetOps Spectrum AIM solutions provides information that is specific to the technology it supports. For example:

- Virtual Host Manager provides data that is specific to virtual technologies.
- Cluster Manager provides data that is specific to cluster technologies.
- Active Directory and Exchange Server (ADES) Manager data that is specific to the supported Active Directory and Exchange Server roles.

Combined, these features provide a complete monitoring solution. To set up your implementation of multiple AIM solutions, take the following recommended approach.

> **WARNING**
> When using multiple AIMs, only a single AIM can be installed on a CA eHealth SystemEDGE host.

Follow these steps:

1. Configure the AutoDiscovery settings on the VNM model.
2. Configure the Virtual Host Manager settings that are related to your virtual technology.
3. Set up Virtual Host Manager by modeling the virtual technology manager and all virtual technology components.
4. Set up Cluster Manager by modeling the cluster technology manager and all cluster components.
5. Set up ADES Manager by modeling the ADES Host Manager and all Active Directory and Exchange Server hosts.

For more information, see Cluster Manager Solution and Active Directory and Exchange Server Manager Solution.

Viewing the Virtual Environment

VHM Overview

The purpose of Virtual Host Manager is to provide visibility into your virtual environment. This visibility lets you view the logical relationships between devices, view performance data for individual entities, and report on the data you discover. Your virtual environment inevitably connects with your physical environment. Virtual Host Manager can help you visualize where these connections are and how they are performing.

Virtual Host Manager provides several methods for viewing your virtual environment, as follows:

- The Explorer tab hierarchy in the Navigation panel shows logical relationships.
- Icons for individual models provide status and model type information at a glance.
- A graphical topology view helps you visualize connections between virtual and physical entities.
- Information views in the Contents and Component Detail panels provide detailed information about individual entities in your virtual environment.

Understanding each of these methods can help you monitor your virtual environment, letting you troubleshoot issues and optimize performance.

For more information about using the OneClick interface, see Using OneClick .

Cut, Copy, Paste and Delete Functionality for VHM Models

The Cut, Copy, Paste and Delete functionality for VHM models is disabled by default to avoid duplication of models in ESX container. If you move or delete a virtual model from its original parent ESX container, at the next polling cycle the model gets recreated in the original ESX container again. To avoid this and other similar problems with virtual models,

the Cut, Copy, Paste and Delete functionality for VHM models is disabled by default. If required, this functionality can be enabled by changing the ‘editmodemask’ (attribute 0x130cc) value to zero in OneClick.

When you discover devices using SNMP Discovery in DX NetOps Spectrum, Virtual Host Manager creates SNMP models for all new SNMP-capable models and places them in the correct ESX container. It is not supported to move a virtual model from its original parent ESX container to another.

One operation that is supported for virtual models is to copy a model from VHM to the World topology view, as these are separate so do not lead to duplication.

You may need to delete a VM model and recreate it in Virtual Host Manager in the following scenarios:

- To update the DX NetOps Spectrum model type if an SNMP agent type changes from one type to another
- To change a model type from an SNMP-capable type to Pingable

NOTE

Please be aware that moving or deleting VHM models within the Universe can have unexpected consequences as they are automatically modeled in the correct ESX host container.

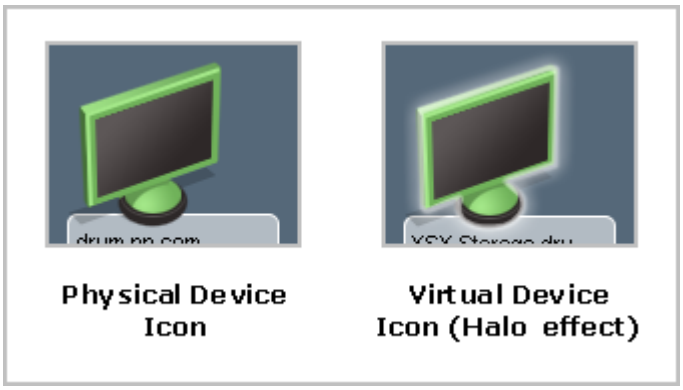
Changing the ‘editmodemask’ (attribute 0x130cc) value:

Follow these steps:

1. From OneClick, select Virtual Server, Component Details panel, Attributes tab.
2. Filter for the "EditModeMask" (0x130cc) attribute in the attribute list.
3. Move the "EditModeMask" attribute to the right side of the window and change the attribute value to '0'. You can cut, copy, paste or delete the device within 10-15 seconds.

Icons for Virtual Devices

Virtual Host Manager provides icons that are designed specifically to distinguish devices in your virtual environment. To distinguish physical and virtual entities, the virtual device icons have a halo-like appearance around the outer edge. For example, a virtual device model icon displays a halo around the perimeter, as follows:



For physical servers that host virtual devices, Virtual Host Manager uses a distinctive honeycomb pattern on the device icon, as follows:

Locating Virtual Models

The models that are created for your virtual environment are integrated into DX NetOps Spectrum in the following three places:

- **Universe group**
  Appears in the Navigation panel and provides a hierarchical tree structure that displays the logical relationships between devices, both physical and virtual.
- **Virtual Host Manager group**
  Appears in the Navigation panel and provides a hierarchical tree structure. This structure helps you to visualize the relationships between your virtual devices, physical devices, and the logical entities that are configured in your virtual technology.
- **Topology tab**
  Appears in the Contents panel, providing a graphical view of your physical network, virtual network, and virtual machines. The topology provides a layer 2 view of the network, showing how your virtual and physical networks are connected. You can use this view to resolve alarms involving these virtual network models.

  NOTE
  This tab is available for only items in the Universe group.

All these views are available from the Explorer tab in the Navigation panel. Understanding how your virtual environment information appears in DX NetOps Spectrum is the key to deciding which view is best for viewing your virtual entities.

For more information about using the OneClick interface, see Using OneClick .

Locate Models on the Explorer tab

As you work with models in the OneClick Console, you can quickly locate a selected model on the Explorer tab. The location feature is provided for items in the Contents and Component Detail panels that reference a single model. These items include rows from the alarms or events tables. The feature is also available from the search results table.

View the same model in different Explorer tab groups to gain insight into its relationships within your physical and virtual networks.

To locate models on the Explorer tab, use the following procedure.

Follow these steps:

1. Locate an item in the Contents panel or Component Detail panel that references a single model.
2. Right-click the item and select from the following options:

- **Location**

Changes the OneClick Console views to locate the selected model within the Explorer tab hierarchy in the Navigation panel. You can select from the following location options**:**

- – **Universe**
  Locates the model in the Universe group hierarchy on the Explorer tab.
- – **Virtual Host Manager**
  Locates the model in the Virtual Host Manager group hierarchy on the Explorer tab.

The OneClick Console locates the related model in the Explorer tab. The Contents and Component Detail panels display details about the selected model.

Topology View

The DX NetOps Spectrum topology views provide a graphical depiction of your physical network, virtual network, and virtual machines. The topology views are available on the Topology tab in the Contents panel. Use the views on the

Topology tab to resolve alarms involving these virtual network models. These views display the Layer 2 connectivity, showing how virtual and physical networks are connected.

DX NetOps Spectrum provides options for arranging the models in most topology views, such as the tree, radial, or manual layout. When selecting the tree layout, the Topology tab for the Universe group includes the following three *unlabeled* tiers of models:

- **Top tier**
  Displays the routers that are discovered with SNMP. These routers are the first level of routers within your virtual network environment that connect your virtual host devices to your physical network.
- **Middle tier**
  Contains any manageable switches that are discovered in your environment. These switches provide connectivity to the virtual host devices within the data center.
- **Bottom tier**
  Contains the virtual host device models and any unmanaged switches. The virtual host devices are the physical servers that run your virtualization technology.

When a server that hosts virtual machines is selected from the Explorer tab, only one layout option is available for the Topology tab. This automatic layout is organized into a tree structure and includes the following three *labeled* tiers:

- **Physical Network**
  Contains an off-page reference to any physical switches that detect traffic for a specific virtual machine. These entities are the components of your physical network that connect to your virtual network.
- **Virtual Network**
  Represents the internal or virtual switching that the virtual machine device provides. When a virtual switch has been configured with multiple virtual machines, DX NetOps Spectrum creates a model in the Virtual Network tier named a "repeater segment" or a "fanout." This fanout model represents the presence of a virtual switch.
- **Virtual Machines**
  Includes the virtual machines that are configured on the virtual host device that you selected in the Navigation panel.

Information Tab and Subviews

The tabs in the Contents and Component Details panels provide information that helps you monitor your virtual environment. The Information tab provides details about a single entity in your environment.

Expand the subviews to see detailed information. Most of the Information tabs include a General Information subview that lists general details about a selected model. Details include the IPv4 address, connection status, and other information.

Updating the Views

When you run the initial Discovery, Virtual Host Manager populates the Explorer tab with virtual device models. After Virtual Host Manager builds this initial hierarchy, your virtual network configuration can change frequently. Therefore, Virtual Host Manager continually updates this information. The information is useful for troubleshooting issues and optimizing performance only when it accurately reflects your virtual environment.

Understanding how and when the information is updated can help you evaluate the data and monitor your virtual environment.

Searching your Virtual Environment

Searching your virtual environment with DX NetOps Spectrum is a fundamental network management task. Virtual Host Manager does not provide a virtual-only topology view. Instead, DX NetOps Spectrum provides a collection of searches on the Locater tab that are designed specifically for your virtual network. These searches identify specific models or

groups of models on your virtual network. Using these searches can help you locate details that you can use to monitor the performance of your virtual environment.

Alarms and Fault Isolation

To alert you to problems within your virtual network, DX NetOps Spectrum generates alarms and uses advanced fault management techniques to isolate the root cause. Virtual networks provide a unique management opportunity because they provide an alternate management perspective in addition to standard device monitoring. While gathering information directly from a device, DX NetOps Spectrum also simultaneously gathers information from the proxy manager. With this extra monitoring capability, in addition to Contact Lost alarms, you can also incur Proxy Lost or Proxy Manager Unavailable alarms.

Alarms and fault isolation vary by virtual technology. The type of fault isolation that Virtual Host Manager uses depends on the devices that generate alarms and the type of events. DX NetOps Spectrum uses all available information to correlate the alarms to the appropriate root cause, avoiding multiple or false alarms.

Alarms on Initial Models

Until DX NetOps Spectrum contacts a model, the model remains in the Initial (blue) condition. Alarms are typically not visible on a model in the Initial condition; however, an exception applies when using Virtual Host Manager. If a virtual machine is brought into Virtual Host Manager management in the powered-down or suspended state, the critical Powered-Down or Suspended alarm overrides the Initial condition.

Creating Event Reports

Use event filters to create event reports in Spectrum Report Manager. You can base these reports on any of the traps and events that are generated for your virtual entities in DX NetOps Spectrum.

To report on Virtual Host Manager events, the following event filter files are included with Spectrum Report Manager:

- vhm.xml
- vhmtrap.xml

For more information about using Spectrum Report Manager to generate event reports from these codes, see Install Report Manager. For information about using the predefined event filter files to generate reports, see Install Report Manager section.

Deleting Models When Using Multiple AIM Solutions

If you use Virtual Host Manager with other DX NetOps Spectrum AIM solutions, consider the following points when deleting models in your environment:

- If you plan to stop managing the device models using Virtual Host Manager, configure Virtual Host Manager delete settings to retain models. Otherwise, Virtual Host Manager deletes the model initially, losing any history or customization. Another AIM solution then recreates the model.

  NOTE
  The Virtual Host Manager setting to retain models when the technology manager is deleted applies to SNMP-enabled device models only. For ICMP (Pingable) models, Virtual Host Manager deletes the model, and then another AIM solution recreates the model.
- When Virtual Host Manager unmanages a device and the model is retained, another AIM solution automatically pulls the model into its management.
- If a solution no longer manages a device, the rules of the remaining solutions are reapplied in the ranked order. Typically, any changes are made at the next polling cycle.
- The Explorer view hierarchy synchronizes after the Lost and Found (LostFound) is emptied.

VMware

This section is for VMware users and describes how to use Virtual Host Manager to manage your virtual entities that are created with VMware vCenter.

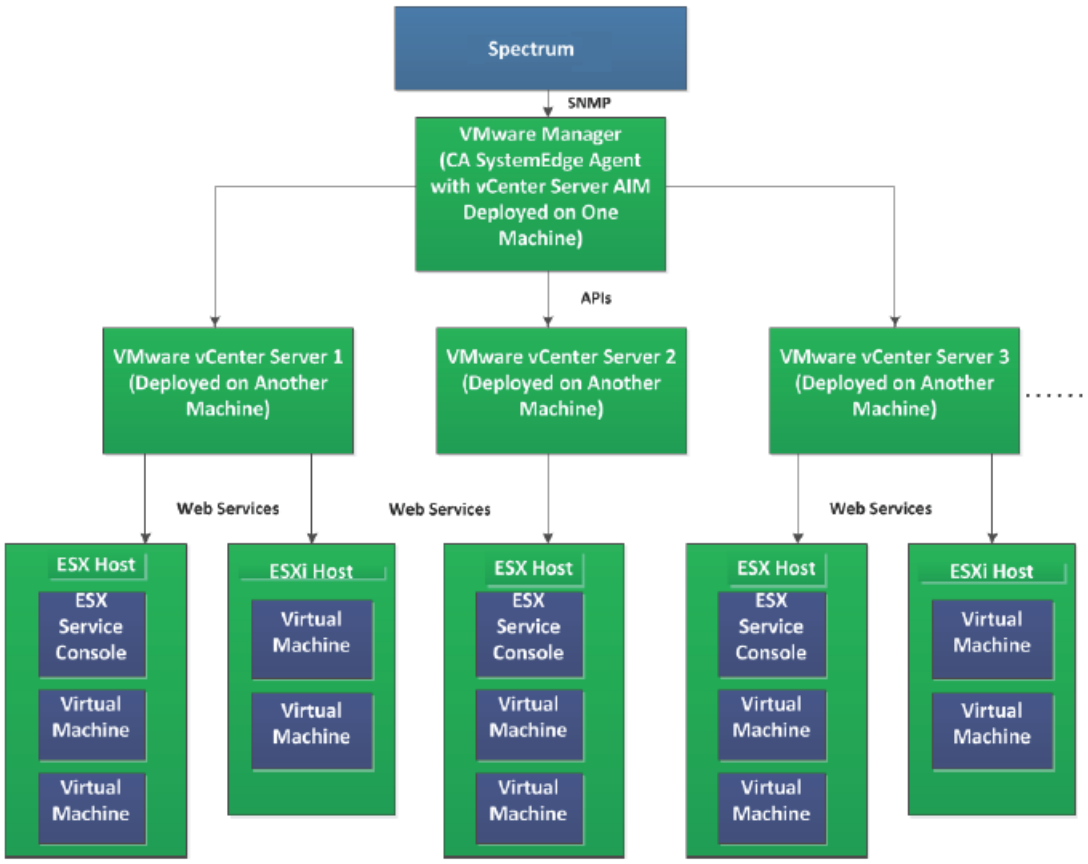
How Virtual Host Manager Works with VMware

Virtual Host Manager monitors your virtual network entities seamlessly with your physical network entities within DX NetOps Spectrum. You get a full view of your network where you can troubleshoot networking issues for both types of entities. Although your virtual network entities behave like physical components, the process for monitoring those entities differs from the general DX NetOps Spectrum monitoring process. Understanding how this process works can help you locate and resolve networking issues that are related to your virtual network.

DX NetOps Spectrum supports only the remote deployments of the latest CA eHealth SystemEDGE. The latest CA eHealth SystemEDGE comes with the latest vCenter Server AIM capable of managing multiple vCenter Server instances (multi-instance). As a result, you can have one or more remote CA eHealth SystemEDGE deployments for managing multiple VMware vCenter Servers. We recommend not to manage the same VMware vCenter Server using more than one remote CA eHealth SystemEDGE deployment.

NOTE
The latest version of CA eHealth SystemEDGE is 5.9. For more information about remotely deploying CA eHealth SystemEDGE, see *CA Virtual Assurance for Infrastructure Managers Implementation Guide*.

The following diagram shows how DX NetOps Spectrum gathers information about your VMware virtual environment using the latest remote CA eHealth SystemEDGE agent:



As shown in the diagram, the process to gather information about your VMware virtual environment is as follows:

1. The VMware vCenter application manages the ESX hosts in your virtual network. The VMware vCenter application stores detailed data about each ESX host and their virtual machines.
2. The CA eHealth SystemEDGE agent communicates with vCenter to gather the details about your virtual network. The CA eHealth SystemEDGE agent must have the vCenter Server AIM loaded.
3. Periodically, DX NetOps Spectrum retrieves information from CA eHealth SystemEDGE and uses it to model and monitor your virtual entities in OneClick.

Because Virtual Host Manager communicates with vCenter, DX NetOps Spectrum is aware of spontaneous network configuration changes. Examples are those changes that are due to VMware VMotion, HA technology, or a DRS scenario. Changes that are associated with these events are quickly reflected in OneClick and factored into the root cause analysis.

Models Created for VMware

Virtual Host Manager provides several models to represent the components of your VMware virtual technology network. Understanding the following basic models can help you better understand Discovery and how the virtual environment interfaces with your physical environment.

> **NOTE**
> Deployment of the CA eHealth SystemEDGE agent and vCenter Server AIM in your environment impacts the models that Virtual Host Manager displays.

Virtual Host Manager includes the following models and icons for VMware devices in the *remote* deployment scenario:

- **VMware Manager**

Represents a physical or virtual host that contains the CA eHealth SystemEDGE agent with vCenter Server AIM loaded. This CA eHealth SystemEDGE agent remotely monitors the vCenter application running on a separate host (represented by the VMware vCenter Server model).
Icon:

or

- **VMware vCenter Server**
  Represents a physical or virtual host that contains the vCenter application to manage your VMware virtual environment. The CA eHealth SystemEDGE agent with vCenter Server AIM monitors the vCenter application remotely. The CA eHealth SystemEDGE agent with vCenter Server AIM is on a separate host (represented by the VMware Manager model).
  Icon:

or

- **ESX Host**
  Represents an ESX host, as configured in your VMware virtualization technology. An *ESX host* is a physical computer that uses ESX Server virtualization software to run virtual machines. Hosts provide the CPU and memory resources that virtual machines use and give virtual machines access to storage and network connectivity. In the Universe topology, these models group your virtual entities into a separate view while showing how the virtual environment interacts with the physical network. The ESX host cannot be contacted directly for status information. Instead, the status of these models is inferred from the status of the items that it contains.
  Icon:

- **ESX Service Console**
  Represents the ESX service console component of your virtual environment. The *ESX service console* is a Linux kernel running on the ESX host that provides a management interface to the hosted virtual machines.

Icon:

- **Virtual Machines**
  Represents a virtual machine, as configured in your VMware virtualization technology. A *virtual machine (VM)* is a software computer that, like a physical computer, runs an operating system and applications. A virtual machine dynamically consumes resources on its physical host, depending on its workload. Because virtual machines are flexible computing units, their deployment comprises a wide range of environments. Examples include environments such as data centers, cloud computing, test environments, or desktops and laptops. In data center implementations, they are used for server consolidation, workload optimization, or higher energy efficiency.
  Icon:

Virtual Host Manager also creates models these additional VMware entities that organize the ESX hosts and their virtual machines:

- **Data Centers**
  Represents a data center, as configured in your VMware virtualization technology. A *data center* serves as a container for your hosts, virtual machines, resource pools, or clusters. Depending on their virtual configuration, data centers can represent organizational structures, such as geographical regions or separate business functions. You can also use data centers to create isolated virtual environments for testing or to organize your infrastructure. Components can interact within data centers, but interaction across data centers is limited. A data center can contain clusters or hosts.
  Icon:

- **Clusters**
  Represents a cluster, as configured in your VMware virtualization technology. A *cluster* is a group of ESX hosts and their associated virtual machines. When a host is added to a cluster, the host resources become part of the cluster resources. The cluster manages the resources of all hosts within it. A cluster can contain hosts, resource pools, or virtual machines.
  Icon:

- **Resource Pools**
  Represents a resource pool, as configured in your VMware virtualization technology. A *resource pool* defines partitions of physical computing and memory resources of a single host or a cluster. You can partition any resource pool into smaller resource pools to divide and assign resources to specific groups or for specific purposes. You can also hierarchically organize and nest resource pools. A resource pool can contain virtual machines or more resource pools.
  Icon:

WARNING
Resource pools that are named "Resources" are skipped during Discovery and modeling. This name is designated for internal use only. Therefore, Virtual Host Manager filters these resource pools out of the Discovery results. You can avoid missing models for resource pools and the devices that they contain by specifying a different name for VMware resource pools.

Discovering VMware Networks

This section describes the Discovery and modeling process for Virtual Host Manager. The Virtual Host Manager administrator typically performs these tasks.

How to Configure Discovery Options

After installation, configure Virtual Host Manager for vCenter Discovery. Selecting preferences helps Virtual Host Manager to model virtual devices correctly.

Select preferences for the following options:

- **Automatically Model New Data Centers**
  Determines whether new data centers that are discovered during vCenter Discovery are modeled automatically.
- **Maintenance Mode for New Virtual Machines**
  Lets you decide which newly discovered virtual machines are placed into maintenance mode until you are ready for DX NetOps Spectrum to manage them.
- **Allow Device Model Deletes During vCenter Discovery**
  Controls how DX NetOps Spectrum handles ESX host, ESX service console, and virtual machine models when vCenter no longer manages them. Controls how these models are handled when you configure DX NetOps Spectrum to disable management of their parent data center.
- **Search for Existing Models**
  Determines the secure domains that Virtual Host Manager searches during a vCenter Discovery.
- **Discover SNMP-Capable Devices**
  Controls how SNMP-capable devices are modeled during vCenter Discovery. By default, new models are initially created as VHM models only. But, this option lets you override the default and immediately create SNMP models for devices that meet the necessary criteria.
- **Retain SNMP-enabled Virtual Machines During VMware Manager Deletion**
  Controls how DX NetOps Spectrum handles SNMP-enabled virtual machine models when a VMware Manager model is deleted.

Configure Automatic Modeling for New Data Centers

For each SpectroSERVER in your networking environment, you can control whether DX NetOps Spectrum automatically models new data centers that are found during vCenter Discovery. Modeling data centers automatically means that DX NetOps Spectrum manages all data centers in your vCenter environment.

Follow these steps:

1. Open Virtual Host Manager in the Navigation panel.
   The main details page opens in the Contents panel for the selected Virtual Host Manager.
2. Click the Information tab.
3. Expand the Configuration, VMware, vCenter Discovery subview.
4. Click Set in the 'Automatically Model New Datacenters' field, and select one of the following options:

- **Yes**
  (Default) Models all data centers that are found during vCenter Discovery. Includes all of the contained clusters, resource pools, ESX hosts, ESX service consoles, and virtual machines.
- **No**
  Prevents the modeling of new data centers that are found during vCenter Discovery. DX NetOps Spectrum does not model the components that are contained within the data center.
  Use this option if your networking environment includes data centers that do not require monitoring. Then model your data centers manually.

Your setting is saved, and new data centers are modeled in Virtual Host Manager according to your selection.

Configure Maintenance Mode for New Virtual Machines

Virtual Host Manager automatically models the virtual machines that vCenter manages. DX NetOps Spectrum attempts to manage all discovered models. However, some virtual machines are not ready for DX NetOps Spectrum management when they are initially modeled. For example, DX NetOps Spectrum generates a Virtual Machine Powered Down alarm when it detects virtual machines that are powered down. To prevent undesired alarms on new models, you can select virtual machine models to be immediately placed into maintenance mode. Later, you can manually disable maintenance mode when you are ready to manage these devices.

Configure the maintenance mode for new virtual machines in OneClick.

Follow these steps:

1. Open Virtual Host Manager in the Navigation panel.
   The main details page opens in the Contents panel for the selected Virtual Host Manager.
2. Click the Information tab.
3. Expand the Configuration, VMware, vCenter Discovery subview.
4. Click Set in the "Maintenance Mode for New Virtual Machines" field, and select one of the following options:
   - **Place only Powered down VMs in Maintenance Mode**
     (Default) Applies maintenance mode only to powered-down or suspended virtual machine models at initial vCenter Discovery.
   - **Place all VMs in Maintenance Mode**
     Applies maintenance mode to all new virtual machine models upon initial vCenter Discovery.

   Your setting is saved, and new virtual machines that are modeled in Virtual Host Manager are placed into maintenance mode according to your selection.

Configure Model Searches Across Secure Domains

Rather than creating new models, vCenter Discovery attempts to locate models that exist in the SpectroSERVER. In an environment with Secure Domain Manager deployed, vCenter Discovery searches for models within the same secure domain as your VMware Manager. This domain is the "local" domain. However, some of your virtual environment devices can exist within a different secure domain. In this case, you can configure vCenter Discovery to search all secure domains for existing models.

You configure model searches across secure domains.

Follow these steps:

1. Open Virtual Host Manager in the Navigation panel.
   The main details page opens in the Contents panel for the selected Virtual Host Manager.
2. Click the Information tab.
3. Expand the Configuration, VMware, vCenter Discovery subview.
4. Click Set in the "Search for Existing Models" field and select from the following options:

- **In vCenter's Secure Domain**
  (Default) Searches for existing models within the same secure domain as the vCenter server.
- **In All Secure Domains**
  Searches for existing models within all secure domains managed by the SpectroSERVER. Select this option only in the following situations:
  - All devices have unique IP addresses.
  - When secure domains are used for security purposes or to isolate network traffic.

  NOTE
  Do not select this option for a NAT environment.

Your setting is saved. vCenter Discovery searches for the specified models in DX NetOps Spectrum. When duplicate models (models with the same IP address) exist in multiple secure domains, Virtual Host Manager handles the situation as follows:

- Virtual Host Manager selects the model in the local secure domain, if available.
- If a duplicate model does not exist in the local domain, Virtual Host Manager randomly selects a model from another secure domain.
- In both cases, Virtual Host Manager generates a minor alarm for the duplicate IP addresses on the VMware Manager model.

Configure SNMP Modeling Preferences

SNMP-capable virtual machines support enriched device monitoring, such as process and file system monitoring capabilities. However, SNMP agents can be costly and time-consuming to deploy. By default, vCenter Discovery creates ESX service consoles and virtual machines as VHM models. You can later upgrade them to SNMP models. However, you can also configure vCenter Discovery to model all new SNMP-capable devices as SNMP models. Although vCenter Discovery can take longer to complete, initially modeling as SNMP models avoids manually upgrading these models later.

WARNING
Enable SNMP modeling *before* you model your vCenter servers. If you model the vCenter servers first, all child models are created as VHM models, which must be manually upgraded to SNMP models.

Follow these steps:

1. Open Virtual Host Manager in the Navigation panel.
   The main details page opens in the Contents panel for the selected Virtual Host Manager.
2. Click the Information tab.
3. Expand the Configuration, VMware, vCenter Discovery, SNMP Discovery subview.

   WARNING
   To prepare your devices and DX NetOps Spectrum for SNMP Discovery, follow the steps in the subview. If devices are not properly prepared before vCenter Discovery, Virtual Host Manager cannot create SNMP models.
4. Click Set in the 'Discover SNMP-Capable Devices' field and select from the following options:
   - **Yes**
     Enables SNMP modeling during vCenter Discovery. Only those devices that meet the specified criteria in the SNMP Discovery subview text are modeled as SNMP devices. Applies to *new* models only.
   - **No**
     (Default) Models all new devices that are found during vCenter Discovery as VHM models. You can manually upgrade these models to SNMP models later.

   Your setting is saved.

Manage Device Models for Devices Deleted from vCenter

The devices and the relationships between them change frequently in virtual networks. DX NetOps Spectrum attempts to reflect these changes accurately. When an ESX host is removed or a virtual machine is deleted in vCenter, DX NetOps Spectrum removes the corresponding device model from the Virtual Host Manager hierarchy. The option to "Allow Device Model Deletes During vCenter Discovery" controls whether DX NetOps Spectrum deletes the model. This option also controls the handling of device models that are contained in a data center when you disable management of the data center in Virtual Host Manager.

WARNING
When models are deleted, all notes or other customizations on those models are lost. You can disable this option if your models are likely to be recreated in vCenter later.

You can manage device models for devices that are deleted from vCenter.

Follow these steps:

1. Open Virtual Host Manager in the Navigation panel.
   The main details page opens in the Contents panel for the selected Virtual Host Manager.
2. Click the Information tab.
3. Expand the Configuration, VMware, vCenter Discovery subview.
4. Click Set in the "Allow Device Model Deletes During vCenter Discovery" field and select one of the following options:
   – **Yes**
   (Default) Deletes the Virtual Host Manager models that correspond to entities that are no longer managed in vCenter. Also deletes data center models for which you disable modeling in Virtual Host Manager.
   – **No**
   Places Virtual Host Manager models in the LostFound container if their corresponding entity is no longer managed in vCenter. Also places data center models in the LostFound container when you disable modeling for the data center in Virtual Host Manager.

   NOTE
   Models with more associations, such as a model that is included in a Global Collection, are handled differently. These models are removed from the Universe, but they are not moved to the LostFound container.

Your setting is saved, and device models are handled as such after the device is deleted from vCenter.

Manage SNMP-Enabled Virtual Machine Models After VMware Manager Deletion

By default, SNMP-enabled devices are deleted from DX NetOps Spectrum when the following items are deleted:

- VMware Manager model for the device
- VMware folder in the Navigation panel

SNMP-enabled device models can include significant customizations that you want to retain. You can adjust your settings to avoid deleting these models. They are placed into the LostFound container for later use.

You can retain SNMP-enabled device models after VMware Manager or VMware folder deletion.

Follow these steps:

1. Open Virtual Host Manager in the Navigation panel.
   The main details page opens in the Contents panel for the selected Virtual Host Manager.
2. Click the Information tab.
3. Expand the Configuration, VMware, vCenter Discovery subview.
4. Click Set in the "Retain SNMP-enabled Virtual Machines During VMware Manager Deletion" field and select one of the following options:
   – **Yes**
   Retains SNMP-enabled virtual machine models in the LostFound container when their VMware Manager or the VMware folder is deleted.

NOTE
Models with more associations, such as a model that is included in a Global Collection, are handled differently. These models are removed from the Universe, but they are not moved to the LostFound container.

- – **No**
  (Default) Deletes all virtual machine models when their VMware Manager or the VMware folder is deleted.

Your setting is saved, and SNMP-enabled device models are handled appropriately when VMware Manager models or the VMware folder is deleted.

How to Discover and Model Your Virtual Environment (VMware)

To monitor your virtual environment, discover and model the virtual entities such as datacenters, resource pools, clusters, ESX hosts, ESX service consoles, and virtual machines. Modeling these entities in Virtual Host Manager lets you view your complete network topology in one tool. You can see the relationships between your physical and virtual components.

The main steps for modeling your virtual environment are as follows:

- **Run a standard Discovery**
  This Discovery ensures that the upstream routers and switches are modeled before vCenter Discovery runs. Or, if the SNMP Modeling option is disabled, this step can also model the SNMP-capable ESX service consoles and virtual machines. When modeling these entities, be sure that your modeling options are set correctly to support Virtual Host Manager.
- Upgrade the CA eHealth SystemEDGE Model
  This step is required only when your CA eHealth SystemEDGE agent on the vCenter server was modeled in a release earlier than r9.1.
- **Let vCenter Discovery run**
  When you model the CA eHealth SystemEDGE agent (with the vCenter Server AIM), vCenter Discovery begins automatically. Each of these vCenter Server models has its own vCenter Discovery process. vCenter Discovery finds the virtual entities that vCenter manages and models the ones that do not exist. vCenter Discovery then places the models in the CA eHealth SystemEDGE view of the Navigation panel.

Run a Standard Discovery

To discover your VMware environment, run the standard DX NetOps Spectrum Discovery. This Discovery ensures that the upstream routers and switches are modeled so that later connections from the virtual entities can be established. You can also model the SNMP-capable ESX service consoles and virtual machines during DX NetOps Spectrum Discovery.

NOTE
Modeling SNMP-capable ESX service consoles and virtual machines is necessary during DX NetOps Spectrum Discovery only when the SNMP Modeling option is disabled during vCenter Discovery.

NOTE
Only an administrator performs this task.

Follow these steps:

1. Open the Discovery console.

   IMPORTANT
   Ensure that you know the correct community strings, IP addresses, and port numbers for any SNMP agents that run on a nonstandard port.

2. In the Navigation panel, click the **Creates a new configuration** icon  .
3. Configure your options for supporting virtual network modeling:
   a. Click **Modeling Options** in the Modeling Options group.
      The Modeling Configuration dialog opens.

b. Click **Protocol Options**.
The Protocol Options dialog opens.
c. Select the **ARP Tables for Pingables** option, and then click **OK**.
The Modeling Configuration dialog opens.
d. (Optional) Click **Advanced Options** in the Advanced Options group. Add your nonstandard SNMP ports (such as the CA eHealth SystemEDGE agent port), and then click **OK**.

4. Enter individual IP addresses or the beginning and ending IP addresses in the IP Boundary List fields, and then click **Add**.

IMPORTANT
Be sure that the range of IP addresses includes all servers with CA eHealth SystemEDGE and vCenter Server AIM installed and the interconnecting switches and routers. Or you can include the SNMP-capable ESX service consoles and virtual machines for which you want to create SNMP models.

5. Enter any additional values in the Discovery console, and then click **Discover**.
The following models are created and are added to your network topology in DX NetOps Spectrum:
– vCenter servers and the switches and routers that connect them to your network -- Information about your virtual environment comes from the vCenter server. When these vCenter Server models exist in DX NetOps Spectrum, vCenter Discovery can begin.
– ESX service consoles and virtual machines -- If you decide not to model these entities with DX NetOps Spectrum Discovery, vCenter Discovery creates them as VHM models.

NOTE
You can also manually model your virtual network by IP address. In this case, we recommend modeling the upstream devices first. Modeling in the correct order ensures that the relationships between these entities are built correctly in the topology. For more information about Discovery, see Modeling and Managing Your IT Infrastructure section.

Upgrade the CA eHealth SystemEDGE Model

The CA eHealth SystemEDGE agent could have been modeled in DX NetOps Spectrum before installing Virtual Host Manager or before the vCenter Server AIM was loaded on the agent. In this case, the existing CA eHealth SystemEDGE model is not compatible with Virtual Host Manager. Upgrade the model so that Virtual Host Manager can access the vCenter Server AIM capabilities in CA eHealth SystemEDGE. *This procedure is not required if the CA eHealth SystemEDGE agent with vCenter Server AIM is modeled after installing DX NetOps Spectrum.*

NOTE
When you are already running the latest remote CA eHealth SystemEDGE with the latest vCenter server AIM, the CA eHealth SystemEDGE model is upgraded automatically.

How vCenter Discovery Works

vCenter Discovery is a specialized discovery process that gathers detailed information about your virtual environment entities. vCenter Discovery finds the virtual entities that vCenter manages and models the ones that do not exist. vCenter Discovery then places the models in the Virtual Host Manager view of the Navigation panel.

A key benefit of vCenter Discovery is that it runs automatically in the background, continually keeping your virtual environment data updated in DX NetOps Spectrum. Understanding how vCenter Discovery works reinforces the importance of properly installing and modeling the various components of Virtual Host Manager.

The vCenter Discovery process works as follows:

1. When the CA eHealth SystemEDGE agent and vCenter Server AIM are operational, the AIM communicates with vCenter servers to gather information about the virtual entities it manages. The vCenter Server AIM stores this information.

WARNING
The CA eHealth SystemEDGE agent and vCenter Server AIM must be installed and configured so that CA eHealth SystemEDGE, vCenter, and DX NetOps Spectrum can communicate. If they cannot, vCenter Discovery cannot run.

2. During DX NetOps Spectrum Discovery, DX NetOps Spectrum creates a vCenter Server model for each server that is referenced in step 1. DX NetOps Spectrum intelligence is enabled to handle communication between DX NetOps Spectrum and the CA eHealth SystemEDGE agent.
3. DX NetOps Spectrum polls the vCenter Server AIM to gather the vCenter information that was stored in Step 1.
4. DX NetOps Spectrum begins vCenter Discovery. The information from the AIM is used to update modeling in the DX NetOps Spectrum Topology tab and the Virtual Host Manager hierarchy in the Navigation panel, as follows:
   a. If you enable SNMP Discovery before Step 2, Virtual Host Manager Discovery creates SNMP models for all new SNMP-capable models meeting the SNMP Discovery criteria.

      NOTE
      By default, SNMP Discovery is disabled during vCenter Discovery.

   b. VHM models are created for data centers, clusters, and resource pools.

      WARNING
      Resource pools that are named "Resources" are skipped during Discovery and modeling. This name is designated for internal use only. Therefore, Virtual Host Manager filters these resource pools out of the Discovery results. You can avoid missing models for resource pools and the devices that they contain by specifying a different name for VMware resource pools.

   c. Previously existing ESX service console and virtual machine models are changed to VHM models.
   d. VHM models are created for the ESX service consoles and virtual machines that are not yet modeled in DX NetOps Spectrum.
   e. VHM models are created for the ESX host models. These models display their associated ESX service console and virtual machine models in the Navigation panel, under Virtual Host Manager and the Universe topology.
   f. All models for your virtual network are added to the Virtual Host Manager portion of the Navigation panel.

      NOTE
      In a virtual environment, devices on separate ESX hosts can have the same IP or MAC address. In this case, DX NetOps Spectrum creates duplicate models for each occurrence of an IP or MAC address.

5. vCenter Discovery automatically repeats this process at each regularly scheduled vCenter polling interval.

NOTE
By default, the vCenter polling interval is controlled by a setting on the VMware Manager model. Or you can control vCenter polling independent of the vCenter Server device model using the vCenter server application model.

Adding SNMP Capabilities to VHM Models (VMware)

SNMP-capable virtual machines support enriched device monitoring, such as process and file system monitoring capabilities, that can provide added value to your solution. However, SNMP agents can be costly and time-consuming to deploy throughout an enterprise. When an SNMP agent is not available or SNMP Discovery is disabled, Virtual Host Manager creates ESX service consoles and virtual machines as VHM models.

Later, you can install an SNMP agent on any virtual machine. You can then upgrade its modeling in DX NetOps Spectrum. Depending on your needs, you can upgrade to SNMP models as follows:

- **Upgrade only selected devices**
  This method works quickly when you have a small selection of models that require an upgrade. This method first deletes the VHM models and child models. After DX NetOps Spectrum deletes the models, the new SNMP models are

created during the next vCenter Discovery and placed in Virtual Host Manager. This method requires you to know the IP addresses for the models to upgrade.

- **Upgrade all SNMP-capable VHM models**
  This method upgrades models in batch, and this method is preferred when upgrading Virtual Host Manager to a new release. For this method, you are not required to know the IP addresses of individual models. Another advantage is that after DX NetOps Spectrum deletes the VHM models, the upgraded SNMP models are immediately placed in the Virtual Host Manager hierarchy. You do not have to wait for the next polling cycle. Therefore, child models are not left unmanaged. The drawback to this method is that it can take a long time to complete. The time that is required to complete this upgrade depends on how many community strings and SNMP ports Virtual Host Manager must search when locating SNMP-capable devices.

  NOTE

  Virtual Host Manager attempts to identify SNMP agents on powered-up pingable virtual machines only.

  WARNING

  When models are deleted, all notes or other customizations on those models are lost.

Upgrade Selected VHM Models to SNMP Models

When an SNMP agent is not available or SNMP Discovery is disabled during vCenter Discovery, Virtual Host Manager creates VHM models. This modeling applies to ESX service consoles and virtual machines. Later, you can install an SNMP agent on any virtual machine. You can then upgrade its modeling in DX NetOps Spectrum. You are required to know the IP addresses for the device models you want to upgrade. Manually selecting models to upgrade works quickly, but all notes or customizations on these models are lost during the upgrade.

Follow these steps:

1. Deploy or enable an SNMP agent on the device, if required.
2. Model the device again using one of the following methods:
   - DX NetOps Spectrum Discovery
   - Model individual devices by IP address

   When the new SNMP-capable model is created, DX NetOps Spectrum removes the previous model from Virtual Host Manager and deletes the model. At the next vCenter Server AIM polling cycle, DX NetOps Spectrum adds the SNMP-capable model to Virtual Host Manager in the Navigation panel.

   WARNING
   When models are deleted, all notes or other customizations on those models are lost.

Upgrade All VHM Models to SNMP Models

Virtual Host Manager creates ESX service consoles and virtual machines as VHM models in the following cases:

- When an SNMP agent is not available.
- When SNMP Discovery is disabled during vCenter Discovery.

Later, you can install an SNMP agent on any virtual machine and then upgrade its modeling in DX NetOps Spectrum. When upgrading in batch, DX NetOps Spectrum searches your VHM models, locating those models that are now SNMP-capable devices. Then DX NetOps Spectrum converts them to SNMP models. This method can take a long time, depending on how many community strings and ports Virtual Host Manager must search. However, this method ensures that child models are not unmanaged while parent models are upgrading.

You can upgrade all VHM models for VMware to SNMP models.

Follow these steps:

1. Deploy or enable an SNMP agent on your devices, as required.
2. Open Virtual Host Manager in the Navigation panel.
   The main details page opens in the Contents panel for the selected Virtual Host Manager.
3. Select the VMware Manager model in the Navigation panel that manages the models to upgrade.
4. Click the Information tab.
5. Expand the VMware Manager Modeling Control, ICMP-Only Device Upgrades subview.
6. Click the Upgrade ICMP-Only Devices button.

   WARNING
   When models are deleted, all notes or other customizations on those models are lost.

Virtual Host Manager searches the devices that the vCenter Server AIM manages on the selected VMware Manager device. Virtual Host Manager upgrades all ICMP-only devices that meet the criteria for SNMP devices and places them within the Virtual Host Manager hierarchy.

Move an ESX Host to a Different vCenter

Moving an ESX host from one DX NetOps Spectrum managed vCenter to another can cause modeling problems when both vCenter hosts are modeled on the same SpectroSERVER. Some possible symptoms of these modeling problems are as follows:

- DX NetOps Spectrum deletes the models that are associated with the ESX host and does not recreate them after the move.
- False Proxy Lost alarms are created and remain, even though the new vCenter can contact the ESX host and all hosted virtual machines.

If you move your ESX host in the correct order, you can avoid these problems.

To move an ESX host to a different vCenter server, use the following procedure.

Follow these steps:

1. (Optional) Change the "Allow Device Model Deletes During vCenter Discovery" option to No.

   NOTE
   Perform this step only if both the originating and destination vCenters are modeled in the same SpectroSERVER. This setting keeps the existing ESX host, ESX service console, and virtual machine models when they become unmanaged by the first vCenter server. Therefore, customizations or historical details for the models are preserved and available after the move.
2. Open VMware and remove the ESX host from the management of the first vCenter server.
3. Wait for Virtual Host Manager in the Navigation panel to reflect the changes.
4. Open VMware and add the ESX host into the destination vCenter server.

   NOTE
   Virtual Host Manager is not DSS aware. Therefore, when moving the ESX host to a vCenter server modeled on a different SpectroSERVER, a new set of models are created. These models represent the ESX host, ESX service console, and the hosted virtual machines.
5. (Optional) Change the "Allow Device Model Deletes During vCenter Discovery" option back to Yes on the originating vCenter server model.
   The ESX host is successfully moved from one vCenter server to a second vCenter server.

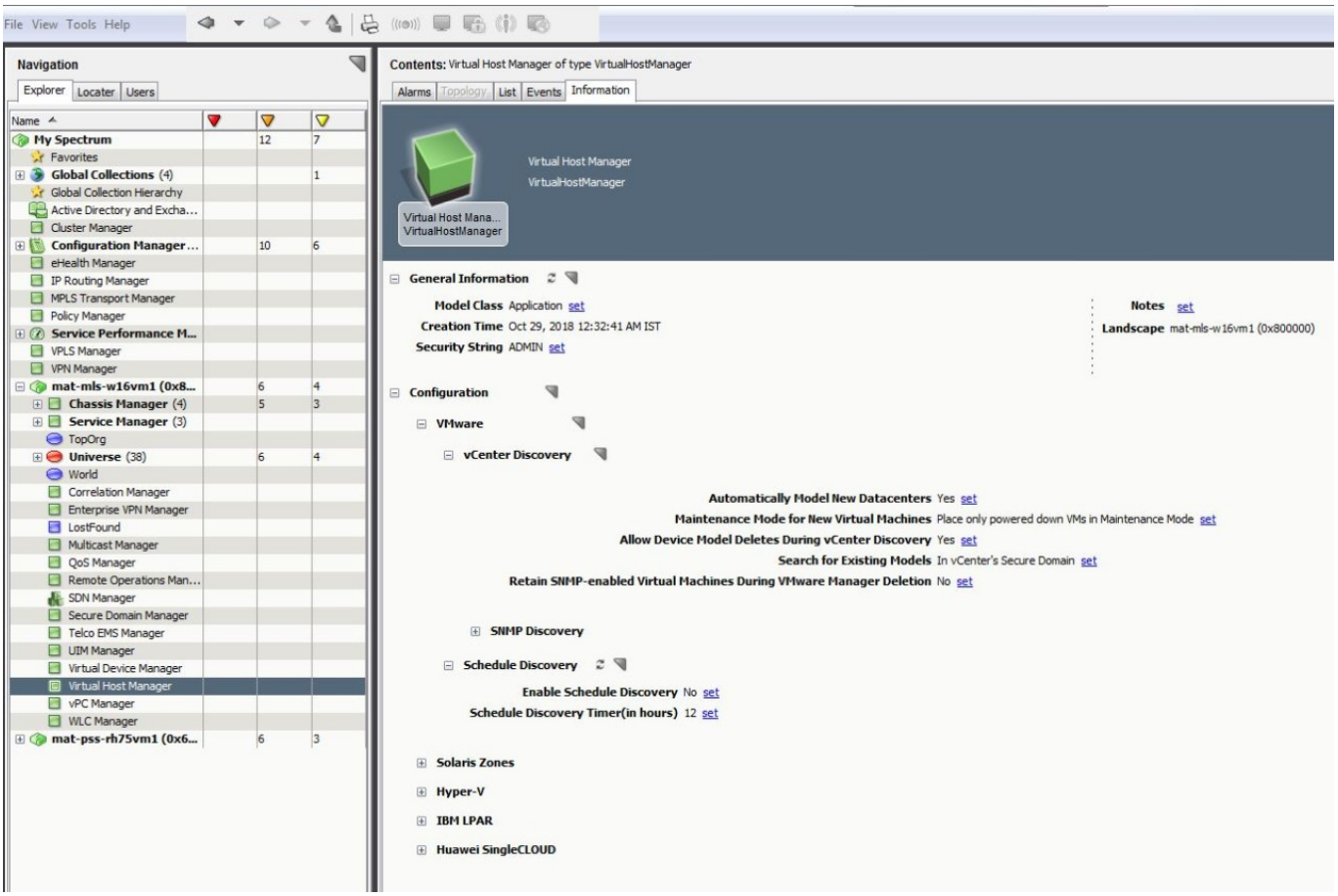
Schedule Discovery

Scheduling discovery jobs for VHM VMs addresses performance issues with the continuous discovery requests which are triggered from VMware. With an increase in the capacity that is the number of models, performance issues are prevalent, and scheduling discovery jobs can prove to be beneficial in managing large customer environments. This interim solution is introduced to schedule time (default of 12 hours) and queue up all the discovery jobs to run them post schedule expiry time.

Users can now **set** the '**Enable Schedule Discovery**' feature to '**Yes**' to enable scheduling of discovery jobs, then **set** the '**Schedule Discovery Timer**' (in hours) to the time they want, for queuing up all the discovery jobs. Users can also specify any given time (in hours) between 1 hour and 24 hours *only*. If a user does not specify a time, then by default the 'Timer' is set to 12 hours.

> **NOTE**
>
> If the schedule discovery option is enabled on an active server, that goes down and does not come up back until the scheduled discovery time, then the scheduled discovery does not run on the backup server.

Following is a screenshot of the VHM Vmware Configuration Settings Tab:

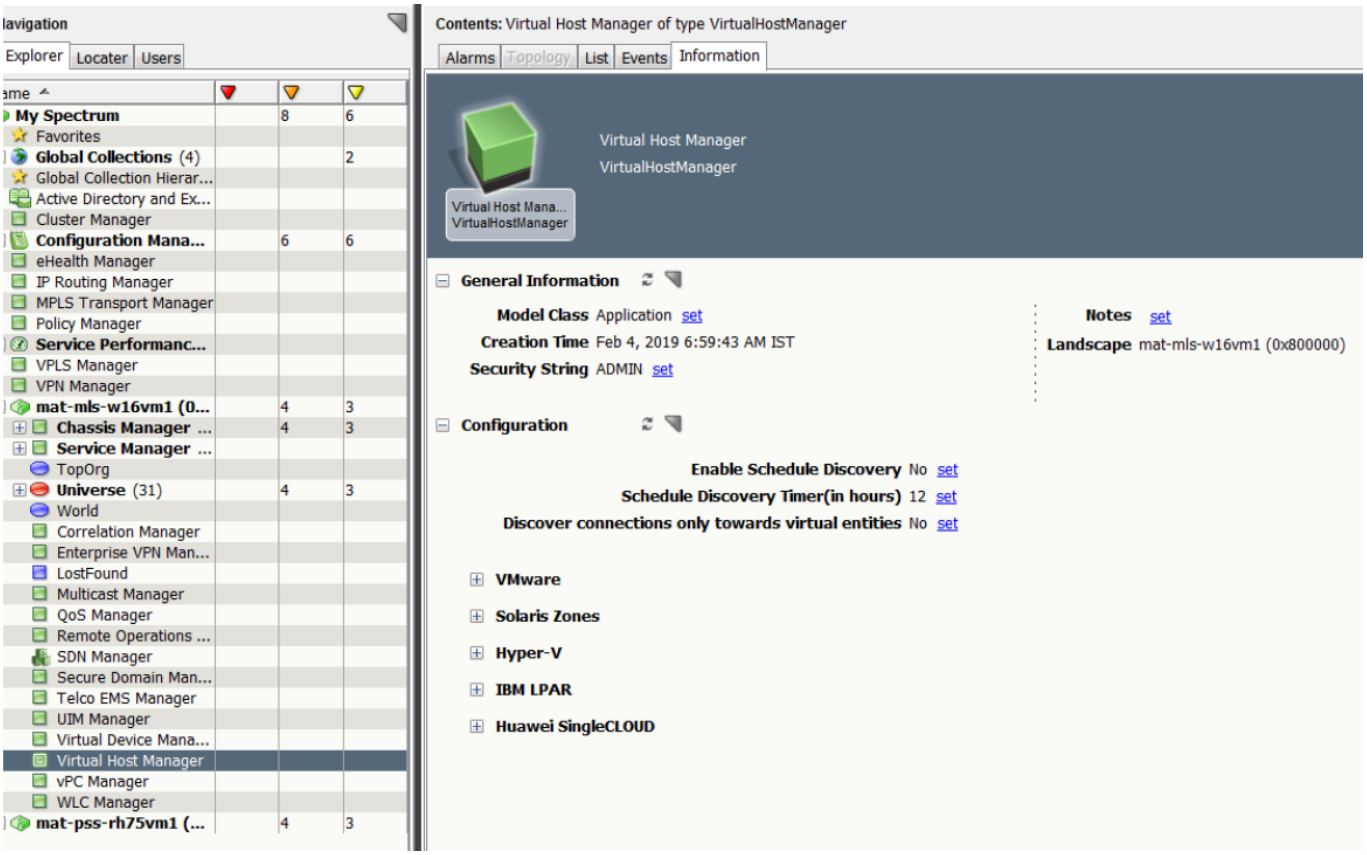


Discover Connections Only toward Acess Points

Previously VHM discovered connections on all the subnet switches corresponding to the VM. From 10.3.1 onwards there is an option under the VHM Manager to discover connections with the right upstream switches of the VM. This

feature skips creating other neighboring connections for the subnet switches, and uses the mac address (SAT Table) or IP address (CDP Table) of the VM to discover connections toward its upstream switch.

Following is a screenshot of the new 'Discover connections only towards virtual entities' option:



Viewing Your VMware Virtual Environment

This section describes concepts for viewing your VMware virtual environment and the associated alarms. The basic steps are no different from the standard DX NetOps Spectrum procedures. However, this section describes conceptual differences and details that only apply to the VMware virtual technology.

Viewing Your VMware Virtual Network

On the Explorer tab, the Virtual Host Manager node provides a hierarchical tree structure. This layout helps you visualize the logical relationships between your virtual environment resources.

Using this information, you can see how resources are shared among your virtual hosts. This information can help you identify opportunities to reorganize and optimize your virtual environment. The hierarchy also provides a quick way to monitor the performance of your resources and troubleshoot alarms.

Because Virtual Host Manager is not aware of a DSS environment, it is located within a landscape hierarchy. The following example, showing where Virtual Host Manager appears on the Explorer tab in the Navigation panel, illustrates the virtual environment hierarchy:

```
[-] <ss> host
    [+] Universe
    [-] Virtual Host Manager
        [-] VMware
            [-] VMware Manager 1
                [-] vCenter server 1
                    [-] Datacenter 1
                        [-] ESX host 1
                         .  ESX service console 1
                         .  Virtual machine 1
                         .  Virtual machine 2
                [-] vCenter server 2
                    [-] Datacenter 2
                        [-] ESX host 2
                             .  ESX service console 2
                             .  Virtual machine 3
                             .  Virtual machine 4
                                [+] Resource pool 1
                                 .  Virtual machine A
                                 .  Virtual machine B
                        [+] Cluster 1
                        [-] Cluster 2
                            [-] ESX host A
                                 .  ESX service console A
                             .  Virtual machine 3
                             .  Virtual machine 4
                            [-] Resource Pool 2
                                 .  Virtual machine C
                                [+] Resource Pool A
                                [+] Resource Pool B
                    [+] Datacenter 3
            [-] VMware Manager 2
                [-] vCenter server 3
                    [-] Datacenter 4
                        [-] ESX host 1
                         .  ESX service console 1
                         .  Virtual machine 1
                         .  Virtual machine 2
                [-] vCenter server 4
                    [-] Datacenter 5
                        [-] ESX host 2
                             .  ESX service console 2
                             .  Virtual machine 3
                             .  Virtual machine 4
```

Virtual Host Manager is the root node for the entire virtual environment that this SpectroSERVER manages. Selecting this node in the Navigation panel displays Virtual Host Manager details in the Contents panel. You can view details such as events and alarms that are related to your virtual environment.

Directly under Virtual Host Manager, virtual environments are organized within folders that represent the associated technology. In the example hierarchy above, the VMware folder contains the portion of the virtual environment that was created using VMware virtualization technology. In this folder, Virtual Host Manager lists all CA eHealth SystemEDGE servers with the vCenter Server AIM and the vCenter servers that this SpectroSERVER manages. These entities are represented separately as a VMware Manager model with a vCenter Server model directly beneath it in the hierarchy.

Selecting a VMware Manager in the Navigation panel displays details in the Contents panel, such as the Configuration, Managed Environment, and Events.

Each vCenter server contains only the portion of the entire virtual environment that it manages. Selecting a vCenter server in the Navigation panel displays details in the Contents panel, such as Configuration and Utilization of the selected vCenter servers.

Under each vCenter server, the hierarchy represents the logical relationships among the following virtual entities:

- **Data Centers**
  A data center can contain clusters or hosts. Selecting a data center in the Navigation panel displays details in the Contents panel. These details include events and alarms that are related to the data center or a list of clusters. Components can interact within data centers, but interaction across data centers is limited.
- **Clusters**
  Clusters can contain ESX hosts, resource pools, or virtual machines. Selecting a cluster in the Navigation panel displays details in the Contents panel, which include:
  – Events and alarms that are related to the cluster.
  – A list of ESX hosts and virtual machines that are contained in the cluster.
  – The DRS and HA settings.
- **Resource pools**
  A resource pool can contain virtual machines or other resource pools. Selecting a resource pool in the Navigation panel displays details in the Contents panel, which include:
  – Overall CPU usage.
  – Events and alarms that are related to the resource pool.
  – A list of other virtual network objects that are contained in the resource pool.

    WARNING
    Resource pools that are named "Resources" are skipped during Discovery and modeling. This name is designated for internal use only. Therefore, Virtual Host Manager filters these resource pools out of the Discovery results. You can avoid missing models for resource pools and the devices that they contain by specifying a different name for VMware resource pools.

- **ESX hosts**
  An ESX host can contain an ESX service console, resource pools, or virtual machines. Selecting an ESX host in the Navigation panel displays details in the Contents panel, which include:
  – Total virtual machine memory.
  – CPU state.
  – A list of virtual machines that the ESX host manages.

    NOTE
    When a cluster contains an ESX host, the virtual machines that are associated with the host are not grouped under the host. Instead, they appear under the cluster beside the ESX host on the Explorer tab.
- **ESX service consoles**
  The ESX service console model appears as a child to its corresponding ESX host model. The ESX service console model is always a leaf node on the Virtual Host Manager hierarchy tree. This model has the same name as its parent. The model icon in the Contents and Component Detail panels distinguishes the ESX service console models from their parent ESX host model. The DeviceType attribute also distinguishes these models. Selecting an ESX service console in the Navigation panel displays details in the Contents panel.

NOTE
The ESX service console model is the only VMware model type within Virtual Host Manager that does not provide a Virtual Host Manager-specific subview on the Information tab.

- **Virtual machines**
  A virtual machine is always a leaf node on the Virtual Host Manager hierarchy tree. Selecting a virtual machine in the Navigation panel displays details in the Contents panel, including power status, memory usage, and related events and alarms.

Understanding the VMware Virtual Topology

The vCenter server, ESX host, ESX service console, and virtual machine models that are created for your virtual environment are integrated into the topology views. ESX host models automatically group their associated ESX service console and virtual machines. The topology shows how these ESX service consoles and virtual machines are connected to your physical network entities.

The following example shows how these models can appear on the Explorer tab in the Navigation panel under the Universe group:

```
[-] Universe
     .  Physical switch 1
     .  Physical switch 2
    [-] ESX host
         .  ESX service console
         .  Fanout 1
         .  Fanout 2
         .  Virtual machine 1
         .  Virtual machine 2
         .  Virtual machine 3
```

Selecting one of these models displays these relationships graphically on the Topology tab in the Contents panel.

How the VMware Data is Updated in Virtual Host Manager

During your initial vCenter Discovery, DX NetOps Spectrum populates the Virtual Host Manager hierarchy in the Navigation panel with your virtual device models. After DX NetOps Spectrum builds this initial hierarchy, your virtual network configuration can change frequently. Virtual Host Manager continually works to keep this information accurate in DX NetOps Spectrum. For example, the following events can change your virtual network configuration:

- Adding a new vCenter server to be managed by an existing CA eHealth SystemEDGE.
- Creating or deleting datacenters, clusters, resource pools, ESX hosts, or virtual machines in the vCenter application
- HA or DRS settings in VMware, which can cause virtual machines to move spontaneously to a new ESX host
- Manually migrating a virtual machine from one ESX host to another

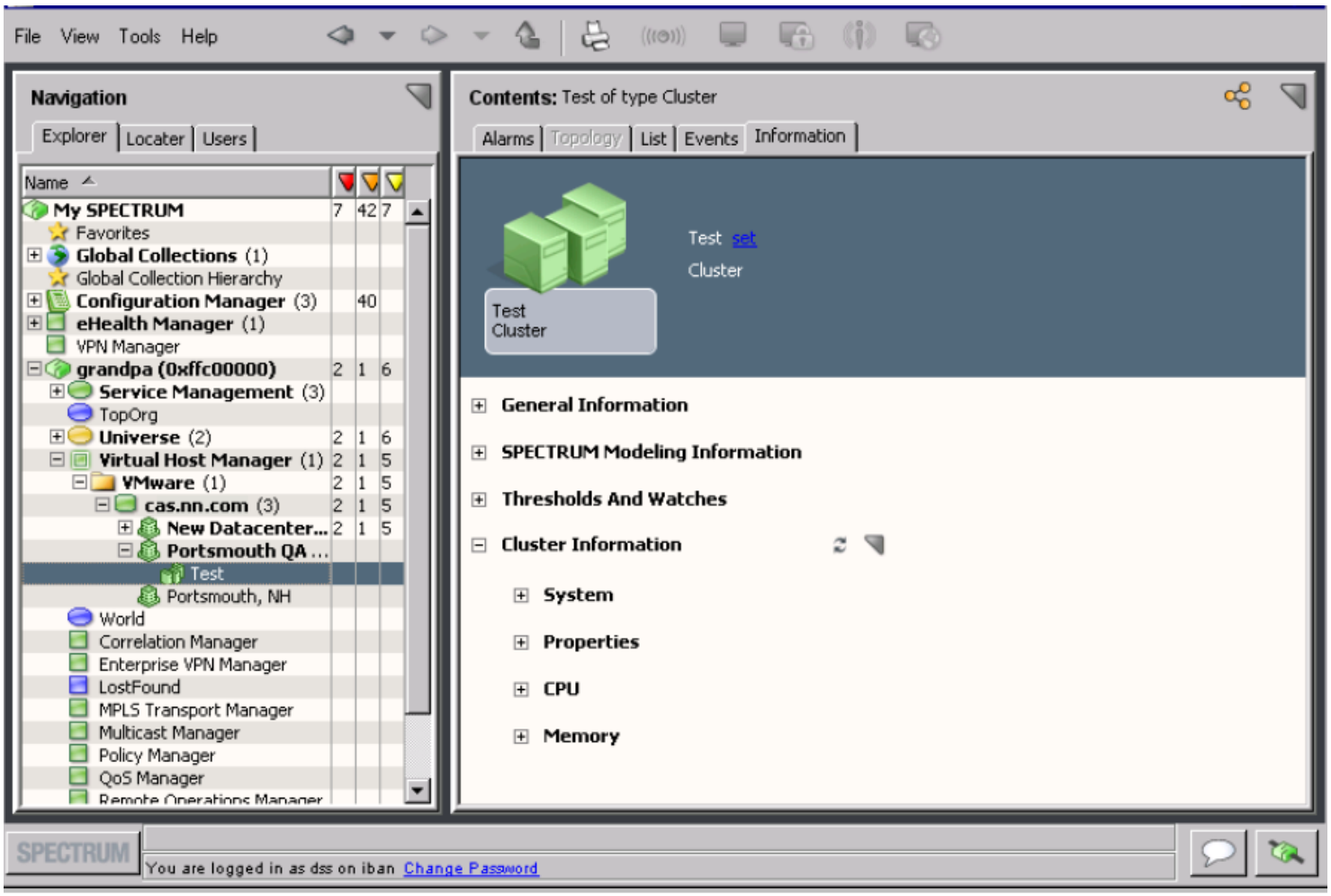
To keep your information accurate, Virtual Host Manager detects these changes by polling the vCenter Server AIM. Therefore, your virtual network configuration changes, if any, are reflected in DX NetOps Spectrum at each polling cycle. DX NetOps Spectrum also receives traps from the AIM and generates the corresponding events. By reviewing the event log, you can find out when configuration changes occur. Example configuration changes include when a virtual device is migrated because of HA or DRS. When it detects a change in your virtual network configuration, DX NetOps Spectrum performs the following tasks:

- Updates the placement of your virtual entity models in the Virtual Host Manager hierarchy of the Explorer tab
- *Automatically* rediscovers connections to the affected ESX service console and virtual machine models and associates them with the correct ESX host in the Universe topology

WARNING
To reestablish connections to your virtual models correctly, all interconnecting routers and switches in your physical network must be modeled. If these models do not exist before connections to your virtual devices are rediscovered, DX NetOps Spectrum cannot resolve those connections in the Universe topology view. The ESX hosts are placed in the same LAN container as the CA eHealth SystemEDGE model.

Custom Subviews for Virtual Entity Types

Your Virtual Host Manager models collectively provide information about your virtual environment. Individually, each model provides unique information or configuration settings, depending on the virtual entity type it represents. This custom subview appears on the Information tab in the Contents panel. These subviews can contain real-time data, such as disk space available or memory utilization. Also, these subviews provide access to threshold settings. For example, the custom subview for a cluster is the "Cluster Information" subview, as shown:



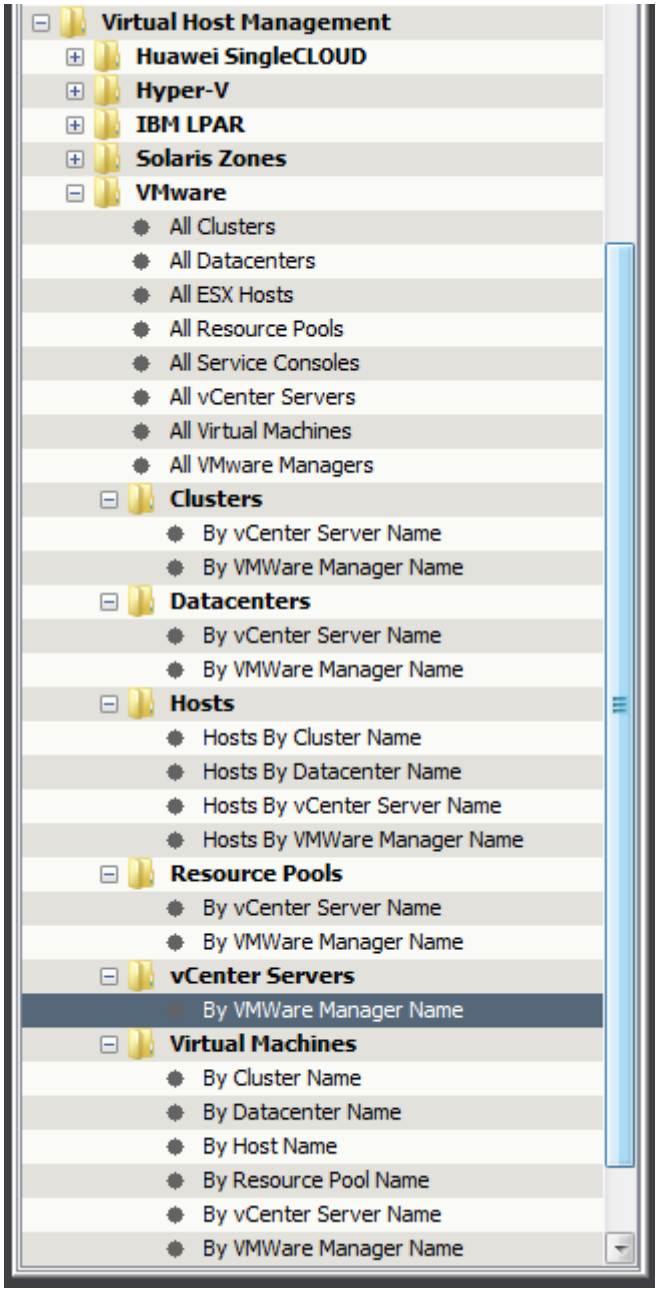
The ESX service console model is the only VMware model type within Virtual Host Manager that does not provide a Virtual Host Manager-specific subview.

NOTE
The vCenter model provides combined information for all virtual devices that the vCenter server manages. Select the VMware Manager model in the Navigation panel to see information about the selected manager and combined information about all of its entities. These entities include vCenter server, ESX hosts, ESX

service consoles, virtual machines, virtual switches, NICs, and datastores. This information is the same data that is displayed on the Information tab for each individual entity model. The combined subview in the VMware Manager model can provide a good overview about all of the virtual entities that it manages.

Locater Tab for VMware Searches

In addition to viewing details about your virtual environment on the Explorer tab, you can also use the Locater tab to run preconfigured Virtual Host Manager searches. The search options are grouped under the Virtual Host Management, VMware folder on the Locater tab, as shown:



These detailed searches can help you investigate information that is related to virtual entities only, such as a specific resource pool or ESX host. For example, if you know the name of a specific ESX host, you can search for all virtual

machines that it manages. Creating this list of virtual machines can be useful when checking the status of a group of virtual machines. Or, you can use the list to determine which machines require management changes in VMware. Example management changes include moving the virtual machines to a different ESX or placing them in maintenance mode.

NOTE
Although Virtual Host Manager is not DSS aware, these preconfigured searches let you select multiple landscapes to search in the search parameters.

The Locater tab in the Navigation panel includes the following searches for Virtual Host Manager information:

- **All Clusters**
  Locates all clusters that have been modeled in the DX NetOps Spectrum database for the virtual network.
- **All Datacenters**
  Locates all datacenters that have been modeled in the DX NetOps Spectrum database for your virtual network.
- **All ESX Hosts**
  Locates all ESX host servers that have been modeled in the DX NetOps Spectrum database for your virtual network.
- **All Resource Pools**
  Locates all resource pools that have been modeled in the DX NetOps Spectrum database for your virtual network.
- **All Service Consoles**
  Locates all ESX service consoles that have been modeled in the DX NetOps Spectrum database for your virtual network.
- **All vCenter Servers**
  Locates all VMware vCenter host servers that have been modeled in the DX NetOps Spectrum database for your virtual network.
- **All Virtual Machines**
  Locates all virtual machines that have been modeled in the DX NetOps Spectrum database for your virtual network.
- **All VMware Managers**
  Locates all servers hosting the CA eHealth SystemEDGE agent with vCenter Server AIM enabled and that have been modeled in the DX NetOps Spectrum database for your virtual network.
- **Clusters**
  Locates Clusters in the DX NetOps Spectrum database. Results are limited to only those entities that the containers that are specified in one of the following searches manage:
  – By Vmware Manager Name
  – By vCenter Server Name
- **Datacenters**

Locates Datacenters in the DX NetOps Spectrum database. Results are limited to only those entities that the containers that are specified in one of the following searches manage:

By Vmware Manager Name

- By vCenter Server Name

Hosts

Locates ESX host servers or ESX service consoles in the DX NetOps Spectrum database. Results are limited to only those entities that the containers that are specified in one of the following searches manage:

- ESX Hosts By Cluster Name
- ESX Hosts By Datacenter Name
- By VMware Manager Name
- By vCenter Server Name

Resource Pools

Locates Resource Pools in the DX NetOps Spectrum database. Results are limited to only those entities that the containers that are specified in one of the following searches manage:

By Vmware Manager Name

- By vCenter Server Name

vCenter Servers

Locates vCenter Servers in the DX NetOps Spectrum database. Results are limited to only those entities that the containers that are specified in one of the following searche manage:

By Vmware Manager Name

Virtual Machines

Locates virtual machines in the DX NetOps Spectrum database. Results are limited to only the virtual machines that the containers that are specified in one of the following searches manage:

- By Cluster Name
- By Datacenter Name
- By Host Name
- By Resource Pool Name
- By VMware Manager Name
- By vCenter Server Name

Status Monitoring Options

DX NetOps Spectrum provides a wide range of options for monitoring the state of your virtual network resources. The status information available for a resource varies, depending on the type of virtual entity you are monitoring. Also, your ability to configure a status option depends on its type. For example, some status options are read-only, but others let you configure threshold values, enable behaviors, or select an alarm severity. Providing this range of options and levels of customization, DX NetOps Spectrum lets you decide how best to monitor the performance of your virtual network.

Status fields are located in the OneClick subviews. All status information for a given virtual environment is available on the VMware Manager model in a tabular format. Also, each virtual entity type that has a unique model in DX NetOps Spectrum provides a subset of the same status information for easy viewing. Status-related settings, including the alert type, monitor, and thresholds, can be set from either subview location.

The following tables outline the type of status information available for each virtual entity type. The Subview Location column describes where the corresponding status fields are located in OneClick. For example, DX NetOps Spectrum lets you monitor "memory" information for your resource pool models. Thus, the corresponding status fields are available from the Resource Pool and VMware Manager subviews on the Information tab in OneClick. To explore the exact status options available for each status information type, locate the subview in OneClick.

Datacenter

Status Information Type	Subview Location
Overall	Datacenter and VMware Manager

Resource Pool

Status Information Type	Subview Location
Overall	Resource Pool and VMware Manager
CPU	Resource Pool and VMware Manager

Memory	Resource Pool and VMware Manager

Virtual Machine

Status Information Type	Subview Location
Percent ready	Virtual Machine and VMware Manager
CPU	Virtual Machine and VMware Manager
Memory	Virtual Machine and VMware Manager
Heartbeat	Virtual Machine and VMware Manager
Power	Virtual Machine and VMware Manager
OS state	Virtual Machine and VMware Manager
Connected	Virtual Machine and VMware Manager
VMware tools	Virtual Machine and VMware Manager
Virtual NICs	VMware Manager only

ESX Host

Status Information Type	Subview Location
CPU	ESX Host and VMware Manager
Sensor CPU Memory Fan Temperature Voltage Power	VMware Manager only
Physical NICs	VMware Manager only

ESX Service Console

Status Information Type	Subview Location
Memory	ESX Host and VMware Manager

Datastores

Status Information Type	Subview Location
Free space	vCenter only
Capacity	vCenter only

vCenter

Status Information Type	Subview Location
Overall	vCenter
CPU	vCenter
Memory	vCenter

How to Configure Management Options (VMware)

After your virtual network is modeled, you can configure Virtual Host Manager options for viewing and managing your device models. Configuring your preferences helps ensure that Virtual Host Manager handles your virtual device models correctly and monitors only the information that is important to you.

To configure your installation of Virtual Host Manager, perform the following procedures after you discover and model your virtual network:

- Configure the vCenter Server AIM
  These options let you select settings for the CA eHealth SystemEDGE vCenter Server AIM, such as the vCenter Server AIM polling interval and various traps.
- Configure and Monitor Resource Status
  These options let you determine the information that you want to monitor and how DX NetOps Spectrum manages the various events that occur in your virtual network.

Configure the vCenter Server AIM

The vCenter Server AIM communicates with vCenter to manage and collect information about your virtual network. In Virtual Host Manager, you can configure the AIM to determine how it handles traps, and events. The AIM settings let you balance the information to gather against the amount of required resources.

To configure the vCenter Server AIM in Virtual Host Manager, use the following procedure.

Follow these steps:

1. Open Virtual Host Manager in the Navigation panel.
   The main details page opens in the Contents panel for the selected Virtual Host Manager.
2. Locate and click your VMware Manager on the Explorer tab in the Navigation panel.
   The tabs in the Contents panel are populated with details about all vCenter servers.
3. Click the Information tab.
4. Expand the SystemEDGE Application Insight Modules (AIMs), VMware vCenter, Configuration subview.
5. Click Set to change the settings for the following fields, as needed:
   – **Trap Enable Mask**
   Determines which class of traps the vCenter Server AIM sends. The value that is entered in this field determines the class. The values are as follows:
   - **0**
     Sends no traps.
   - **1**
     Sends detected vCenter change traps.
   - **2**
     Sends detected AIM state change traps.
   - **3**
     Sends detected vCenter change traps and detected AIM state change traps.
   - **4**
     Sends AIM configuration change traps only.
   - **5**
     Sends AIM configuration change and detected vCenter change traps.
   - **6**
     Sends AIM configuration change traps and detected AIM state change traps.
   - **7**
     (Default) Sends all traps.

NOTE
Default: 7**Limits:** 0-7

– **Log Level**
Specifies the level of information that is written to the vCenter Server AIM log file. The levels are cumulative (for example, log level 4 writes all messages at levels 0 through 4). The following log levels are available:
- 0: Fatal
- 1: Critical
- 2: Warning
- 3: Info
- 4: Debug
- 5: Debug Low
- 6: Debug Lower
- 7: Debug Lowest

NOTE
Default: 2 Specifying a debug level greater than 4 is not recommended.

6. Expand Configuration, Instances subview.
A table containing all vCenter server instances and their corresponding parameters are displayed.
7. In the Instances table, set any of the following parameters on the required vCenter server instance.
– **Poll Interval (Seconds)**
Specifies the time interval (in seconds) when the vCenter Server AIM polls and caches status and modeling information from the vCenter server. This polling retrieves the following status and modeling updates and more:
- Virtual machine powered down status
- ESX host disconnected
- New data center available
- New ESX host
- New virtual machine

Default: 120
Limits: Numbers greater than or equal to 30
Note: For best results, we recommend that you set this interval no larger than the DX NetOps Spectrum poll cycle interval.

– **VC Event Poll (Seconds)**
Specifies the time interval (in seconds) when the vCenter Server AIM polls and caches event information from the vCenter server. This polling interval affects the polling of the vCenter event queue.
Default: 120
Limits: Numbers greater than or equal to 120

– **VC Event Enable**
Determines how Virtual Host Manager handles events that are collected from the vCenter server and from the vCenter Server AIM. The following options are available:
- **Disable**
Specifies that no events are collected.
- **Collect**
Specifies that events are gathered but no traps are sent for those events having traps.
- **Collect and trap**
Specifies that the events are gathered and traps are sent.

Default: Disable

– **VC Event Monitor Info**
Determines whether vCenter information events are collected. The options are Enable and Disable.
Default: Disable

– **VC Event Monitor User**

Determines whether vCenter user events are collected. The options are Enable and Disable.
Default: Disable

- **VC Event Monitor Error**
  Determines whether vCenter error events are collected. The options are Enable and Disable.
  Default: Disable
- **VC Event Monitor Warning**
  Determines whether vCenter warning events are collected. The options are Enable and Disable.
  Default: Disable

Your vCenter Server AIM is configured with your selections.

Configure and Monitor Resource Status

You can monitor the status of virtual resources in OneClick. For example, you can view the total physical memory, used physical memory, percent of free space on a datastore, and more. Also, you can set monitoring options, such as enabling alerts and setting threshold values for traps. This information can help you optimize your virtual network performance and troubleshoot alarms.

> **NOTE**
> The vCenter Server AIM sets and manages the traps, but you can configure these threshold values from the OneClick subviews. A read/write community string is required to change any threshold values or settings.

You can view or configure resource status options and information for virtual devices on the Information tab.

Follow these steps:

1. Open Virtual Host Manager in the Navigation panel.
   The main details page opens in the Contents panel for the selected Virtual Host Manager.
2. Locate and click the virtual device on the Explorer tab in the Navigation panel.
   The device details display in the Contents panel.
3. Click the Information tab.
   Multiple subviews are available for viewing. Generally, the subview at the bottom of the tab includes the resource allocation and utilization information for the selected model. For example, a datacenter model displays a subview that is named "Datacenter Information". This subview includes details for the specific datacenter model that you selected in the Navigation panel.
4. Expand the appropriate subview.
   All available resource status details and monitoring options for the selected device model are displayed.

> **NOTE**
> The VMware Manager model provides combined information for all virtual devices that the VMware Manager manages. Select the VMware Manager model in the Navigation panel to see information about all the vCenter servers and combined information about all of its entities. These entities include ESX hosts, ESX service consoles, virtual machines, virtual switches, NICs, and datastores. This information is the same data that is displayed on the Information tab for each individual entity model. The combined subview in the VMware Manager model can provide a good overview about all of the virtual entities that it manages.

Controlling vCenter Server AIM Polling (VMware)

When tuning Virtual Host Manager performance, you can change the vCenter server polling rate or you can disable vCenter polling. By default, the polling attributes on the vCenter Server device model control the VMware-related polling behavior. Or you can change this VMware-related polling behavior independently. The vCenter application model, VMWareVCAIMApp, controls your VMware-related polling.

The following two attribute values on the application specifically control the VMware polling logic:

- PollingStatus
- Polling_Interval

Both the vCenter server and the VMWareVCAIMApp application model contain these attributes. PollingStatus enables and disables polling, while Polling_Interval controls the polling frequency. If their values are different, the VMWareVCAIMApp application model attribute values take precedence.

As stated, DX NetOps Spectrum allows you to set the values for the device model and application model separately. This capability lets you fine-tune your VMware-related polling independently of the vCenter server device polling. For both attributes, modifying the attribute on the vCenter Server device model also changes the corresponding application model attribute if their values are the same.

Configure the vCenter Server Polling Interval

You can change the vCenter server polling rate to increase or decrease the frequency. Configure the polling interval by setting the Polling_Interval attribute on the vCenter application model.

Follow these steps:

1. Open OneClick and click the Locater tab in the Navigation panel.
2. Expand the Application Models folder and double-click 'By Device IP Address.'
   A search dialog opens.
3. Enter the IP address for your vCenter server in the Device IP Address field and click OK.
   A list of application models for the vCenter server appears in the Contents panel.
4. Select the VMWareVCAIMApp application model.
   The application model details appear in the Component Details panel.
5. Click the Information tab in the Component Details panel.
6. Double-click the DX NetOps Spectrum Modeling Information subview.
7. Click Set in the Polling Interval (sec) field, enter a new value.
   NOTE
   A value of 0 disables vCenter server polling.
   The vCenter server polling interval setting is configured.

Disable vCenter Server Polling

You can disable vCenter polling. Disabling vCenter polling is the same as disabling Virtual Host Manager. You can disable polling by setting the PollingStatus attribute on the vCenter application model.

To disable vCenter server polling on the application model, use the following procedure.

Follow these steps:

1. Open OneClick and click the Locater tab in the Navigation panel.
2. Expand the Application Models folder and double-click 'By Device IP Address.'
   A search dialog opens.
3. Enter the IP address for your vCenter server in the Device IP Address field, and click OK.
   A list of application models for the vCenter server appears in the Contents panel.
4. Select the VMWareVCAIMApp application model.
   The application model details appear in the Component Details panel.
5. Click the Information tab in the Component Details panel.
6. Click the DX NetOps Spectrum Modeling Information subview.
7. Click 'set' in the Polling field and select Off.
   Polling is disabled for the selected vCenter server.

Disabling DNS Lookup for Virtual Machines

Starting with 9.4 release, you can disable the DNS lookup for a virtual machine that has a blank IP address. You can have scenarios when you do not want DX NetOps Spectrum the DNS lookup for a virtual machine with a blank IP address. In such scenarios, you can set the attribute "VMWare_vmDNSLookuponBlankIPAddr" to "No" at the Virtual Host Manager level in OneClick. When this attribute is set to "No", DX NetOps Spectrum skips the DNS lookup for a virtual machine with a blank IP address. As a result, the IP address of such a virtual machine is not populated in OneClick.

If this attribute is set to "Yes", DX NetOps Spectrum performs a DNS lookup to find the IP address of a virtual machine without an IP address. If DX NetOps Spectrum finds the IP address of that virtual machine, that IP address is populated in the OneClick.

WARNING

If you do not want to perform the DNS lookup after a SpectroSERVER restart, set the **'VMWare_vmDNSLookonVNMRestart'** attribute value to **'No'.**

Deleting Virtual Host Manager Models

Generally, models can be deleted from OneClick at any time. However, Virtual Host Manager restricts your ability to delete models from the Virtual Host Manager hierarchy in the Navigation panel. To delete models manually, you have the following two options:

- Delete the VMware folder or a vCenter server model in Virtual Host Manager
- Remove a virtual entity from your VMware virtual environment using vCenter

In Virtual Host Manager, models are sometimes deleted automatically. The following circumstances cause DX NetOps Spectrum to delete Virtual Host Manager models automatically:

- **VMware folder or VMware Manager model is deleted**
  If you delete a VMware Manager model or the VMware folder, DX NetOps Spectrum deletes all related child models. The set of deleted models includes the vCenter Server model that is related to the remote VMware Manager model.
- **An entity is removed from VMware**
  As you delete data centers, resource pools, clusters, ESX hosts, and virtual machines in VMware, DX NetOps Spectrum also deletes those models and their child models from Virtual Host Manager.
- **Disabled data center in Virtual Host Manager**
  If you disable management of a data center, DX NetOps Spectrum deletes the child models that are related to that data center.
- **Upgraded models exist** -- In some cases, an ESX service console or virtual machine is first modeled for Virtual Host Manager without SNMP capabilities. If SNMP capabilities are later added to a VHM model, the previous model is deleted and replaced with the new SNMP-capable model.

NOTE

Although the default setting is to delete models, you can configure Virtual Host Manager to retain the models instead. In this case, Virtual Host Manager places the ESX host, ESX service console, and virtual machine models in the LostFound container when they are removed from Virtual Host Manager. This setting is respected when you remove an entity from VMware or you disable a data center in Virtual Host Manager. This setting does not apply when you delete the VMware folder or you delete a VMware Manager model. The setting also does not apply when you remove a VMware Manager and vCenter Server model or upgrade a VHM model.

Distributed and Selective Management

This section describes the concepts and procedures for selectively managing individual data centers on your vCenter servers. This section also describes how to distribute the management of data centers across multiple SpectroSERVERs.

Selective Data Center Modeling

By default, each vCenter Server model monitors all data centers that it manages within your virtual environment. Virtual Host Manager lets you selectively monitor only a subset of these data centers. To perform selective modeling, you can configure each vCenter Server model to enable or disable modeling for the individual data centers that it manages.

This feature offers the following benefits:

- Organizations can disable management for data centers that do not require monitoring, such as a lab environment.
- Virtual Host Manager can distribute management of your virtual environment.

How to Selectively Manage Your Datacenters

By default, each vCenter server model monitors all data centers that it manages within your virtual environment. However, you can configure your vCenter server models to monitor a subset of data centers. This feature is useful when you have data centers that do not require monitoring, such as a lab environment.

The following process describes how to configure your vCenter server to monitor only selected data centers:

1. Select your preference for automatically modeling data centers in Virtual Host Manager. This setting is used as the default for your data center models in Step 3.
2. Model your vCenter server.
3. Enable on each vCenter server the selected data centers for monitoring. The vCenter server models only the data centers and its contained components for which you enable modeling.

Distributed Management of Your Virtual Environment

Using the selective data center modeling feature, you can distribute the management of your data centers across multiple SpectroSERVERs. For large organizations with geographically dispersed networks or large virtual environments, the potential benefits include the following:

- Improved Virtual Host Manager performance -- Resources that are required to model each data center can be spread across multiple SpectroSERVERs. Ideally, we recommend modeling using a single SpectroSERVER to reduce resources that are required to poll from multiple servers. However, if a single SpectroSERVER cannot effectively manage your virtual environment, a distributed environment can improve your Virtual Host Manager performance despite the additional polling resources required.
- Organizational flexibility -- Because of organizational or geographical boundaries, you may prefer to distribute the management of data centers across multiple SpectroSERVERs.

You can accomplish distributed management by first modeling your vCenter servers on separate SpectroSERVERs. In each SpectroSERVER environment, you can then selectively enable or disable the data centers managed by each vCenter server.

For example, you model the 'cas' vCenter server on two SpectroSERVERs: SS_1 and SS_2. After vCenter Discovery, Virtual Host Manager discovers that 'cas' manages the following three data centers:

- DCenter-A
- DCenter-B
- DCenter-C

On each SpectroSERVER, you can configure data center modeling for 'cas' as follows:

Data Center	cas on SS_1	cas on SS_2
DCenter-A	enabled	disabled
DCenter-B	enabled	disabled

DCenter-C	disabled	enabled

In this scenario, management of the data centers for 'cas' is distributed across the two SpectroSERVERs.

WARNING
Distributed data center management is not a scalable solution. For every vCenter server that is modeled on a SpectroSERVER, Virtual Host Manager must poll *all* data center data during each polling interval. Therefore, even if you disable modeling for a data center, Virtual Host Manager polls the data center. To minimize duplication of effort when polling, be mindful of how many SpectroSERVERs model the same vCenter server.

How to Distribute Management of Your Virtual Environment

To help improve Virtual Host Manager performance or organization, you can use the selective data center modeling feature. Distributed management spreads your data center modeling across multiple SpectroSERVERs.

The process for distributing management is similar to the selective data center modeling process, with a few additional steps, as follows:

1. Select your preference for automatically modeling data centers in Virtual Host Manager. In a distributed data center management environment, you must decide how to handle new data centers added to VMware. When configuring Virtual Host Manager for data center management, you have the following two options:
   - Disable automatic data center modeling for all SpectroSERVERs -- In this case, you must manually model all new data centers that you want Virtual Host Manager to monitor. Although it requires manual modeling, this option ensures that Virtual Host Manager is monitoring only the data centers that require management.
   - Enable automatic data center modeling on *one* SpectroSERVER, and disable for all others -- This option ensures that all new data centers are modeled on one SpectroSERVER. We recommend this option so that important network components are not forgotten. After your data center models appear in Virtual Host Manager, you can manually move their management to a different SpectroSERVER, if needed.

     WARNING
     Do not enable automatic data center modeling on multiple SpectroSERVERs. Virtual Host Manager models all new data centers on multiple SpectroSERVERs, resulting in duplicate effort, which can affect Virtual Host Manager performance.
2. Model your vCenter server. Be sure to model this vCenter server on each SpectroSERVER where you manage one or more of its data centers.
3. On each SpectroSERVER, enable on each vCenter server the selected data centers to monitor. The vCenter server models only the data centers and its contained components for which you enable modeling.
4. Configure the CA eHealth SystemEDGE agent to send traps to each SpectroSERVER. To ensure that data centers are properly monitored, traps must be sent to all SpectroSERVERs where your vCenter servers are modeled. For more information, see Trap Management in a Distributed Data Center Environment.

Trap Management in a Distributed Data Center Environment

To monitor data centers and their components, the related traps must reach the SpectroSERVER where each data center is managed. In a distributed data center management scenario, configure the CA eHealth SystemEDGE agent to send traps to each SpectroSERVER where your vCenter servers are modeled.

When properly configured, DX NetOps Spectrum sends all traps that are generated by the vCenter Server AIM to these SpectroSERVERs. Each SpectroSERVER filters the traps and drops the traps that are generated for data centers and their components that are *not* modeled on that SpectroSERVER. Only the traps that are related to modeled data center components generate events and alarms.

NOTE
For more information about configuring traps on the vCenter Server AIM, see *CA Virtual Assurance for Infrastructure Managers Implementation*.

Alarms and Fault Isolation for VMWare

This section describes the traps used by Virtual Host Manager and the resulting alarms. The following topics explains how Virtual Host Manager fault isolation differs from basic DX NetOps Spectrum fault isolation:

Virtual Host Manager Alarms for VMware

To alert you to problems within your virtual network, DX NetOps Spectrum generates alarms. Alarms are created in two ways:

- Traps sent from the CA eHealth SystemEDGE agent
- Polling

Two alarms are generated from polling: Powered Down/Suspended and Proxy Lost/Unavailable. However, several traps can generate alarms on your virtual devices. DX NetOps Spectrum supports all traps that are sent by the vCenter Server AIM from the CA eHealth SystemEDGE agent. To optimize these traps, you can configure the threshold values for each virtual device individually.

If a trap breaches your threshold value and generates an alarm, DX NetOps Spectrum uses the value of the “state” varbind passed with the trap to determine the alarm severity. All state varbinds have the following possible values, which receive the same DX NetOps Spectrum alarms:

- 0: Unknown
- 1: OK
- 2: Warning
- 3: Critical

DX NetOps Spectrum maps these vCenter states to a DX NetOps Spectrum alarm severity, as shown:

vCenter State	DX NetOps Spectrum Alarm Severity
0: Unknown	Clear
1: OK	Clear
2: Warning	Minor (Yellow)
3: Critical	Major (Orange)

Forwarding Traps from CA SystemEDGE

DX NetOps Spectrum supports all traps that the vCenter Server AIM sends. These traps are initially sent to the vCenter CA eHealth SystemEDGE model. If the destination for a trap is not the vCenter model, DX NetOps Spectrum forwards the trap to the correct virtual model.

NOTE
For specific event codes related to the traps, use the Event Configuration application and filter on “0x056e.” Or you can launch MIB tools to view the traps in the Trap Support table for the “EMPIRE-CAVMVCA-MIB” MIB. For more information about using the Event Configuration application, see Event Configuration. For more information about using MIB tools, see Modeling and Managing Your IT Infrastructure .

DX NetOps Spectrum determines where to forward the trap by using the following process:

1. When DX NetOps Spectrum receives a trap, it maps the UID of the entity type to a well-known varbind location.

   NOTE
   For the Host Sensor traps, DX NetOps Spectrum uses the virtual entity name, not the UID. If multiple hosts have the same vCenter name, DX NetOps Spectrum maps to the first entry.

2. DX NetOps Spectrum uses this UID to look up and locate the DX NetOps Spectrum model that is tied to a given UID. The entity type of all traps is predetermined. Depending on the results of the look-up, DX NetOps Spectrum forwards the trap as follows:
   – If it finds a DX NetOps Spectrum model of a specific type with a given UID, DX NetOps Spectrum forwards the event and corresponding alarm to the destination model.
   – If it cannot find a DX NetOps Spectrum model for a given UID, DX NetOps Spectrum generates a new generic event (0x56e109f) on the vCenter model. This new event includes the following details:
     • Trap details
     • Entity type searched for
     • Additional information from attempts to find the information about a model

NOTE
DX NetOps Spectrum often cannot find a related model when a trap is sent immediately after changing your virtual network entities in vCenter. vCenter Discovery has not yet identified and created the corresponding model in DX NetOps Spectrum.

Traps Supported in Virtual Host Manager

All traps that the vCenter Server AIM generates are supported in DX NetOps Spectrum. The traps are initially sent to the vCenter model. Then they are forwarded to a corresponding virtual entity type (that is, the "destination" entity), depending on the type of trap. Using these traps, you can monitor the performance of your virtual network, resolve any resulting alarms, or trigger events.

NOTE
For more information about traps that are generated by the vCenter Server AIM, see *CA Virtual Assurance for Infrastructure Managers Administration*.

The following tables list the traps for a specific destination entity type and specify whether the trap generates an alarm.

NOTE

Information!

*The vCenter ESX host name is used to locate the ESX host model in DX NetOps Spectrum. If two ESX host models exist with the same name, DX NetOps Spectrum alarms on the first model matching the name.

**These traps do not generate alarms because DX NetOps Spectrum vCenter polling intelligence detects and generates these alarms on the next vCenter polling cycle.

***These events are generated on the vCenter server because vCenter Discovery generates similar events on each entity that is discovered or removed from DX NetOps Spectrum management.

Trap Name	Trap OID	Alarm?
vmvcAimClusterHADRSChangeTrap	1.3.6.1.4.1.546.1.1.0.165253	No
vmvcAimClusterRenamedTrap	1.3.6.1.4.1.546.1.1.0.165254	No
vmvcAimClusterDRSConfigChangeTrap	1.3.6.1.4.1.546.1.1.0.165255	No

Trap Name	Trap OID	Alarm?
vmvcAimDCRenamedTrap	1.3.6.1.4.1.546.1.1.0.165248	No
vmvcAimDCConfigChangeTrap	1.3.6.1.4.1.546.1.1.0.165249	No
vmvcAimDCOverallStateChangeTrap	1.3.6.1.4.1.546.1.1.0.165240	Yes
vmvcAimDCTotalCPUStateChangeTrap	1.3.6.1.4.1.546.1.1.0.165245	Yes

vmvcAimDCTotalMEMStateChangeTrap	1.3.6.1.4.1.546.1.1.0.165250	Yes

Trap Name	Trap OID	Alarm?
vmvcAimHostCpuStateChangeTrap	1.3.6.1.4.1.546.1.1.0.165208	Yes
vmvcAimHostTotalCpuStateChangeTrap	1.3.6.1.4.1.546.1.1.0.165209	Yes
vmvcAimHostTotalMemStateChangeTrap	1.3.6.1.4.1.546.1.1.0.165210	Yes
vmvcAimHostConfigChangeTrap	1.3.6.1.4.1.546.1.1.0.165212	No
vmvcAimHostTotalVMCpuStateChangeTrap	1.3.6.1.4.1.546.1.1.0.165213	Yes
vmvcAimHostThresholdChangeTrap	1.3.6.1.4.1.546.1.1.0.165215	No
vmvcAimHostVMotionTrap	1.3.6.1.4.1.546.1.1.0.165218	No
vmvcAimHostConnectionStateTrap	1.3.6.1.4.1.546.1.1.0.165219	No**
vmvcAimHostTotalVMMemStateChangeTrap	1.3.6.1.4.1.546.1.1.0.165220	Yes
vmvcAimPNICStateChangeTrap	1.3.6.1.4.1.546.1.1.0.165241	Yes
vmvcAimPNICAddedTrap	1.3.6.1.4.1.546.1.1.0.165242	No
vmvcAimPNICRemovedTrap	1.3.6.1.4.1.546.1.1.0.165243	No
vmvcAimPNICConfigChangeTrap	1.3.6.1.4.1.546.1.1.0.165244	No
vmvcAimHostDiskAddedTrap	1.3.6.1.4.1.546.1.1.0.165291	No
vmvcAimHostDiskRemovedTrap	1.3.6.1.4.1.546.1.1.0.165292	No
vmvcAimCPUSensorStateChangeTrap*	1.3.6.1.4.1.546.1.1.0.165281	Yes
vmvcAimMemSensorStateChangeTrap*	1.3.6.1.4.1.546.1.1.0.165282	Yes
vmvcAimFanSensorStateChangeTrap*	1.3.6.1.4.1.546.1.1.0.165283	Yes
vmvcAimVoltageSensorStateChangeTrap*	1.3.6.1.4.1.546.1.1.0.165284	Yes
vmvcAimTempSensorStateChangeTrap*	1.3.6.1.4.1.546.1.1.0.165285	Yes
vmvcAimPowerSensorStateChangeTrap*	1.3.6.1.4.1.546.1.1.0.165286	Yes
vmvcAimHostFTConfigChangeTrap	1.3.6.1.4.1.546.1.1.0.165955	No
vmvcAimHostPowerStateTrap	1.3.6.1.4.1.546.1.1.0.165910	No
vmvcAimStatVMSRMStatusChangeTrap	1.3.6.1.4.1.546.1.1.0.165969	No

Trap Name	Trap OID	Alarm?
vmvcAimHostMemStateChangeTrap	1.3.6.1.4.1.546.1.1.0.165211	Yes
vmvcAimHostMemOtherStateChangeTrap	1.3.6.1.4.1.546.1.1.0.165214	Yes

Trap Name	Trap OID	Alarm?
vmvcAimResourcePoolCpuStateChangeTrap	1.3.6.1.4.1.546.1.1.0.165258	Yes
vmvcAimResourcePoolConfigChangeTrap	1.3.6.1.4.1.546.1.1.0.165259	No
vmvcAimResourcePoolRenamedTrap	1.3.6.1.4.1.546.1.1.0.165260	No
vmvcAimResourcePoolMemStateChangeTrap	1.3.6.1.4.1.546.1.1.0.165264	Yes
vmvcAimResourcePoolHealthStateChangeTrap	1.3.6.1.4.1.546.1.1.0.165265	Yes
vmvcAimResourcePoolVCConfigChangeTrap	1.3.6.1.4.1.546.1.1.0.165962	No

Trap Name	Trap OID	Alarm?
vmvcAimServerStateChangeTrap	1.3.6.1.4.1.546.1.1.0.165201	Yes

vmvcAimVCCPUStateChangeTrap	1.3.6.1.4.1.546.1.1.0.165203	Yes
vmvcAimVCMemStateChangeTrap	1.3.6.1.4.1.546.1.1.0.165206	Yes
vmvcAimHostAddedTrap***	1.3.6.1.4.1.546.1.1.0.165216	No
vmvcAimHostRemovedTrap***	1.3.6.1.4.1.546.1.1.0.165217	No
vmvcAimVMAddedTrap***	1.3.6.1.4.1.546.1.1.0.165222	No
vmvcAimVMRemovedTrap***	1.3.6.1.4.1.546.1.1.0.165223	No
vmvcAimVMMigratedTrap***	1.3.6.1.4.1.546.1.1.0.165230	No
vmvcAimDCAddedTrap***	1.3.6.1.4.1.546.1.1.0.165246	No
vmvcAimDCRemovedTrap***	1.3.6.1.4.1.546.1.1.0.165247	No
vmvcAimClusterAddedTrap***	1.3.6.1.4.1.546.1.1.0.165251	No
vmvcAimClusterRemovedTrap***	1.3.6.1.4.1.546.1.1.0.165252	No
vmvcAimResourcePoolAddedTrap***	1.3.6.1.4.1.546.1.1.0.165256	No
vmvcAimResourcePoolRemovedTrap***	1.3.6.1.4.1.546.1.1.0.165257	No
vmvcAimTemplateAddedTrap	1.3.6.1.4.1.546.1.1.0.165261	No
vmvcAimTemplateRemovedTrap	1.3.6.1.4.1.546.1.1.0.165262	No
vmvcAimTemplateRenamedTrap	1.3.6.1.4.1.546.1.1.0.165263	No
vmvcAimCustomizationSpecAddedTrap	1.3.6.1.4.1.546.1.1.0.165266	No
vmvcAimCustomizationSpecRemovedTrap	1.3.6.1.4.1.546.1.1.0.165267	No
vmvcAimDatastoreAddedTrap	1.3.6.1.4.1.546.1.1.0.165271	No
vmvcAimDatastoreRemovedTrap	1.3.6.1.4.1.546.1.1.0.165272	No
vmvcAimDatastoreAccessibleStateChangeTrap	1.3.6.1.4.1.546.1.1.0.165273	Yes
vmvcAimDatastoreConfigChangeTrap	1.3.6.1.4.1.546.1.1.0.165274	No
vmvcAimDatastoreRenamedTrap	1.3.6.1.4.1.546.1.1.0.165275	No
vmvcAimDatastoreFreeSpaceStateChangeTrap	1.3.6.1.4.1.546.1.1.0.165276	Yes
vmvcAimDCFolderAddedTrap	1.3.6.1.4.1.546.1.1.0.165277	No
vmvcAimDCFolderRemovedTrap	1.3.6.1.4.1.546.1.1.0.165278	No
vmvcAimDCFolderConfigChangeTrap	1.3.6.1.4.1.546.1.1.0.165279	No
vmvcAimSnapshotAddedTrap	1.3.6.1.4.1.546.1.1.0.165287	No
vmvcAimSnapshotRemovedTrap	1.3.6.1.4.1.546.1.1.0.165288	No
vmvcAimSnapshotCurrentUpdateTrap	1.3.6.1.4.1.546.1.1.0.165289	No
vmvcAimSCSIControllerAddedTrap	1.3.6.1.4.1.546.1.1.0.165296	No
vmvcAimSCSIControllerRemovedTrap	1.3.6.1.4.1.546.1.1.0.165297	No
vmvcAimServerTotalCPUStateChangeTrap	1.3.6.1.4.1.546.1.1.0.165293	Yes
vmvcAimServerTotalMemStateChangeTrap	1.3.6.1.4.1.546.1.1.0.165294	Yes
vmvcAimServerTotalDSFreeSpaceStateChangeTrap	1.3.6.1.4.1.546.1.1.0.165295	Yes
vmvcAimVSwitchStateChangeTrap	1.3.6.1.4.1.546.1.1.0.165235	Yes
vmvcAimVMGuestDiskAddedTrap	1.3.6.1.4.1.546.1.1.0.165920	No
vmvcAimVMGuestDiskRemovedTrap	1.3.6.1.4.1.546.1.1.0.165921	No
vmvcAimVMGuestDiskStateChangeTrap	1.3.6.1.4.1.546.1.1.0.165922	No
vmvcAimVMGuestDiskConfigChangeTrap	1.3.6.1.4.1.546.1.1.0.165923	No

vmvcAimStorageSensorStateChangeTrap	1.3.6.1.4.1.546.1.1.0.165905	Yes

Trap Name	Trap OID	Alarm?
vmvcAimServerReadyTrap	1.3.6.1.4.1.546.1.1.0.165200	No
vmvcAimVCConfigChangeTrap	1.3.6.1.4.1.546.1.1.0.165202	No
vmvcAimVCThresholdChangeTrap	1.3.6.1.4.1.546.1.1.0.165204	No
vmvcAimVCEventReceivedTrap	1.3.6.1.4.1.546.1.1.0.165205	No
vmvcAimVSwitchAddedTrap	1.3.6.1.4.1.546.1.1.0.165915	No
vmvcAimVSwitchRemovedTrap	1.3.6.1.4.1.546.1.1.0.165916	No
vmvcAimVSwitchConfigChangeTrap	1.3.6.1.4.1.546.1.1.0.165917	No
vmvcAimHostVNICAddedTrap	1.3.6.1.4.1.546.1.1.0.165925	No
vmvcAimHostVNICRemovedTrap	1.3.6.1.4.1.546.1.1.0.165926	No
vmvcAimPortGroupAddedTrap	1.3.6.1.4.1.546.1.1.0.165930	No
vmvcAimPortGroupRemovedTrap	1.3.6.1.4.1.546.1.1.0.165931	No
vmvcAimPortGroupVCConfigChangeTrap	1.3.6.1.4.1.546.1.1.0.165932	No
vmvcAimDistribVSwitchAddedTrap	1.3.6.1.4.1.546.1.1.0.165935	No
vmvcAimDistribVSwitchRemovedTrap	1.3.6.1.4.1.546.1.1.0.165936	No
vmvcAimDistribVSwitchStateChangeTrap	1.3.6.1.4.1.546.1.1.0.165937	Yes
vmvcAimDistribVSwitchConfigChangeTrap	1.3.6.1.4.1.546.1.1.0.165938	No
vmvcAimDistribVSwitchVCConfigChangeTrap	1.3.6.1.4.1.546.1.1.0.165939	No
vmvcAimDVPortGroupAddedTrap	1.3.6.1.4.1.546.1.1.0.165940	No
vmvcAimDVPortGroupRemovedTrap	1.3.6.1.4.1.546.1.1.0.165941	No
vmvcAimDVPortGroupVCConfigChangeTrap	1.3.6.1.4.1.546.1.1.0.165942	No
vmvcAimDVUplinkPortGroupAddedTrap	1.3.6.1.4.1.546.1.1.0.165943	No
vmvcAimDVUplinkPortGroupRemovedTrap	1.3.6.1.4.1.546.1.1.0.165944	No
vmvcAimDVUplinkPortGroupVCConfigChangeTrap	1.3.6.1.4.1.546.1.1.0.165945	No
vmvcAimDistribVSwitchPortPolicyChangeTrap	1.3.6.1.4.1.546.1.1.0.165950	No
vmvcAimDVPortGroupPortPolicyChangeTrap	1.3.6.1.4.1.546.1.1.0.165951	No
vmvcAimCustSpecNICAddedTrap	1.3.6.1.4.1.546.1.1.0.165960	No
vmvcAimCustSpecNICRemovedTrap	1.3.6.1.4.1.546.1.1.0.165961	No
vmvcAimVAppAddedTrap	1.3.6.1.4.1.546.1.1.0.165963	No
vmvcAimVAppRemovedTrap	1.3.6.1.4.1.546.1.1.0.165964	No
vmvcAimVAppVCConfigChangeTrap	1.3.6.1.4.1.546.1.1.0.165965	No
vmvcAimVMAddedTovAppTrap	1.3.6.1.4.1.546.1.1.0.165966	No
vmvcAimVMRemovedFromvAppTrap	1.3.6.1.4.1.546.1.1.0.165967	No
vmvcAimVMvAppVCConfigChangeTrap	1.3.6.1.4.1.546.1.1.0.165968	No
vmvcAimNetFolderAddedTrap	1.3.6.1.4.1.546.1.1.0.165970	No
vmvcAimNetFolderRemovedTrap	1.3.6.1.4.1.546.1.1.0.165971	No
vmvcAimNetFolderConfigChangeTrap	1.3.6.1.4.1.546.1.1.0.165972	No
vmvcAimCustomizationSpecChangeTrap	1.3.6.1.4.1.546.1.1.0.165280	No

vmvcAimCustSpecNICChangeTrap	1.3.6.1.4.1.546.1.1.0.165973	No

Trap Name	Trap OID	Alarm?
vmvcAimVMCpuStateChangeTrap	1.3.6.1.4.1.546.1.1.0.165221	Yes
vmvcAimVMConfigChangeTrap	1.3.6.1.4.1.546.1.1.0.165224	No
vmvcAimVMThresholdChangeTrap	1.3.6.1.4.1.546.1.1.0.165225	No
vmvcAimVMPercentReadyTrap	1.3.6.1.4.1.546.1.1.0.165226	Yes
vmvcAimVMRenamedTrap	1.3.6.1.4.1.546.1.1.0.165227	No
vmvcAimVMBehaviourChangeTrap	1.3.6.1.4.1.546.1.1.0.165228	No
vmvcAimVMConnectionStateChangeTrap	1.3.6.1.4.1.546.1.1.0.165229	No**
vmvcAimVMNICStateChangeTrap	1.3.6.1.4.1.546.1.1.0.165231	Yes
vmvcAimVMNICAddedTrap	1.3.6.1.4.1.546.1.1.0.165232	No
vmvcAimVMNICRemovedTrap	1.3.6.1.4.1.546.1.1.0.165233	No
vmvcAimVMNICConfigChangeTrap	1.3.6.1.4.1.546.1.1.0.165234	No
vmvcAimVMVDiskStateChangeTrap	1.3.6.1.4.1.546.1.1.0.165236	Yes
vmvcAimVMVDiskAddedTrap	1.3.6.1.4.1.546.1.1.0.165237	No
vmvcAimVMVDiskRemovedTrap	1.3.6.1.4.1.546.1.1.0.165238	No
vmvcAimVMVDiskConfigChangeTrap	1.3.6.1.4.1.546.1.1.0.165239	No
vmvcAimVMMemStateChangeTrap	1.3.6.1.4.1.546.1.1.0.165268	Yes
vmvcAimVMPowerStateChangeTrap	1.3.6.1.4.1.546.1.1.0.165269	No**
vmvcAimVMHBStateChangeTrap	1.3.6.1.4.1.546.1.1.0.165270	No
vmvcAimVMFTConfigChangeTrap	1.3.6.1.4.1.546.1.1.0.165956	No
vmvcAimVMFTFailoverTrap	1.3.6.1.4.1.546.1.1.0.165957	Yes
vmvcAimVMVCConfigChangeTrap	1.3.6.1.4.1.546.1.1.0.165958	No
vmvcAimVMVDiskSizeChangeTrap	1.3.6.1.4.1.546.1.1.0.165902	No

Fault Management for Virtual Networks

The goal of fault isolation is to narrow down the root cause of a networking problem. Finding the root cause can help you to troubleshoot and quickly correct the problem or to correct the problem programmatically with automated scripts. Deciding which devices are the root cause of an alarm can be difficult, because problems with a single device can cause several devices in your network to generate events.

For example, losing contact with an ESX host often means that you have also lost contact with the virtual machines that the ESX manages. Therefore, the ESX device model and all affected virtual machines generate alarms. Using fault isolation techniques, Virtual Host Manager correlates these alarms to identify a single root cause.

Virtual networks provide a unique management opportunity because they provide DX NetOps Spectrum an alternate management perspective. DX NetOps Spectrum can gather information through direct contact with your virtual devices or through the virtual network management application, VMware vCenter. This alternate management perspective enhances standard DX NetOps Spectrum fault management in two ways:

- **Enhanced contact lost alarms** -- Two sources of information about a device mean Virtual Host Manager can pinpoint the cause and more easily correlate events to a single root cause.
- **Proxy failure alarms** -- *Proxy management* is the act of managing network devices using an alternate management source in place of or in addition to the primary manager. For example, DX NetOps Spectrum can manage virtual network devices by contacting them directly or through the virtual technology application's contact with the devices. When vCenter loses contact with a virtual network device, Virtual Host Manager generates a Proxy Management

Lost alarm for each device. These alarms are unique,alerting you that *management* of the device through the *proxy* is affected, not the state of the device or direct (SNMP) management.

How Fault Isolation Works when Device Contact is Lost

To help you troubleshoot networking problems with your devices, DX NetOps Spectrum uses fault isolation to narrow down the root cause of an alarm. For virtual networks, Virtual Host Manager uses information from direct contact with the device plus information provided by vCenter through the vCenter Server AIM. In many cases, standard DX NetOps Spectrum fault management can pinpoint the root cause. But in special circumstances, the method for isolating problems in a virtual network goes beyond the standard methods.

The type of fault isolation Virtual Host Manager uses to discover the root cause depends on which devices are alarming and the type of events the devices generate. The following scenarios describe two unique fault management situations and how DX NetOps Spectrum isolates the networking error in your virtual network.

Scenario 1: Virtual machine is powered down or suspended

In a virtual environment, the virtual management application can provide more details than DX NetOps Spectrum can discover through standard device monitoring. For example, the management application is aware when a virtual machine is placed into one of the following modes:

- Powered down
- Suspended

If a virtual machine is in one of these modes and DX NetOps Spectrum loses contact with it, but proxy management of the ESX host is uninterrupted, DX NetOps Spectrum determines the root cause as follows:

1. When DX NetOps Spectrum loses contact with the virtual machine, it generates a Contact Lost alarm.
2. During its next polling cycle, the vCenter server model polls the vCenter Server AIM to gather information about the virtual machines. Because vCenter manages the virtual machines, it can provide a unique view into the possible cause of alarms generated by a virtual machine.
3. If vCenter finds that a virtual machine is powered down or suspended, it generates the appropriate alarm.

   NOTE
   The Powered Down and Suspend alarms are cleared upon the first vCenter polling cycle after the virtual machine is powered on.
4. Virtual Host Manager correlates these Powered Down and Suspend alarms to the corresponding Contact Lost alarm created by DX NetOps Spectrum. Virtual Host Manager makes the Contact Lost alarm appear as a symptom of the Powered Down or Suspend alarms.

Scenario 2: ESX host is down

If DX NetOps Spectrum loses contact with a modeled ESX service console and all virtual machines running on that host, DX NetOps Spectrum checks the status of the upstream routers and switches. Depending on their status, DX NetOps Spectrum determines the root cause as follows:

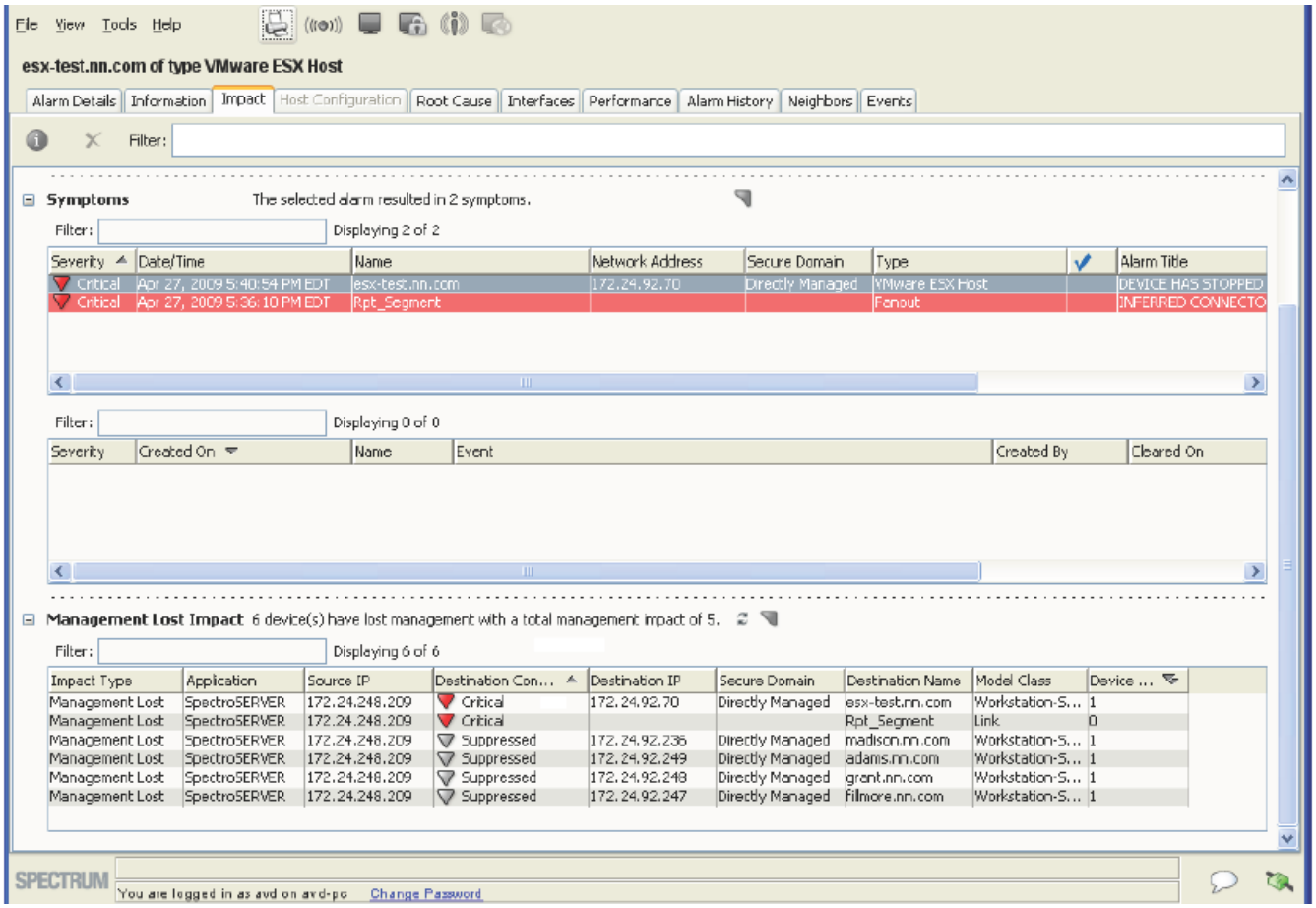
- All upstream devices for one or more virtual machines or the ESX service console are unavailable -- Standard DX NetOps Spectrum fault isolation techniques determine the root cause, as follows:
  - Device Stopped Responding to Polls alarm -- Generated on the ESX host when at least one upstream connected device for any virtual machine or ESX service console is up.
  - Gateway Unreachable alarm -- Generated on the ESX host when *all* upstream connected devices are down.
- At least one upstream device is available for every virtual machine and ESX service console model connected to the ESX host -- DX NetOps Spectrum infers that the ESX host is the root cause and responds as follows:
  a. The ESX service console model and all virtual machines, ports, and fanouts that are directly connected to the ESX service console model or virtual machine models generate the standard fault isolation alarms.
  b. Virtual Host Manager creates a Physical Host Down alarm for the ESX host model.

c. All fault isolation-related alarms that are created for the impacted devices (such as virtual machines, ESX service consoles, ports, and fanouts) are correlated to the Physical Host Down alarm, making them symptoms of the Physical Host Down alarm. These symptom alarms appear in the Symptoms table on the Impact tab for the Physical Host Down alarm.

NOTE
For each ESX host model, Virtual Host Manager creates a "virtual fault domain." This domain includes the ESX host, ESX service console, and virtual machines, plus all ports and fanouts directly connected to the ESX service console model or virtual machines. When the ESX host generates the Physical Host Down alarm, all standard fault isolation alarms within the domain are correlated to it. Correlating these alarms as symptoms indicates that the Physical Host Down alarm on the ESX host is the root cause.

d. All impacted devices are listed in the Management Lost Impact table on the Impact tab for the Physical Host Down alarm.

NOTE
Devices that are suppressed do not have a corresponding alarm in the Symptoms table, which is why the following example shows only two alarms but six impacted devices:



If all upstream devices for one or more virtual machines or the ESX service console go down, DX NetOps Spectrum can no longer reliably state that the fault lies with the ESX host. Therefore, DX NetOps Spectrum clears the Physical Host Down alarm and applies the standard DX NetOps Spectrum fault isolation techniques.

How Fault Isolation Works when Proxy Management is Lost

The VMware vCenter application used to create your virtual network provides DX NetOps Spectrum a unique management opportunity. DX NetOps Spectrum can use the standard methods to contact your virtual devices directly,

plus DX NetOps Spectrum can simultaneously gather virtual device information from vCenter. In this sense, vCenter is a "proxy" from which DX NetOps Spectrum gathers virtual device information. If DX NetOps Spectrum loses direct contact with a device, it generates alarms. Likewise, if vCenter loses contact with a virtual device or if Virtual Host Manager loses contact with the vCenter application, Virtual Host Manager generates alarms -- Proxy Management Lost alarms.

In response, DX NetOps Spectrum attempts to isolate the cause of the proxy management failure. Proxy fault isolation is similar to the standard DX NetOps Spectrum fault isolation, except that these alarms alert you to the fact that *proxy* management of a virtual device is affected. Proxy management fault isolation cannot tell you whether a virtual device is up or down. However, it is important to know when contact through the proxy is lost, because you could be missing important virtual information about a device.

The type of proxy fault isolation Virtual Host Manager uses to discover the root cause depends on which devices are alarming and the type of events the devices generate. The following scenarios describe two unique proxy fault management situations and how Virtual Host Manager isolates the networking error in your virtual network.

Scenario 1: Contact between vCenter and ESX is lost

If vCenter loses contact with one of the ESX hosts it is managing, the vCenter data about that ESX and all hosted virtual devices is lost. To isolate the problem, Virtual Host Manager determines the root cause as follows:

1. A Proxy Management Lost alarm is generated on the ESX host, ESX service console, all hosted virtual machines, and any resource pools defined in that ESX.
2. The virtual machine alarms are correlated to the ESX Proxy Management Lost alarm, making them symptoms of the ESX alarm. Correlating these alarms as symptoms indicates that the ESX alarm is the root cause.
3. If DX NetOps Spectrum also loses contact with the ESX host and generates a Physical Host Down alarm, the Proxy Management Lost alarm generated for the ESX is correlated to the Physical Host Down alarm. In this case, the Proxy Management Lost alarm becomes a symptom of the Physical Host Down alarm. Correlating this alarm as a symptom indicates that the Physical Host Down alarm on the ESX is the root cause.

Scenario 2: Contact between DX NetOps Spectrum and vCenter is lost

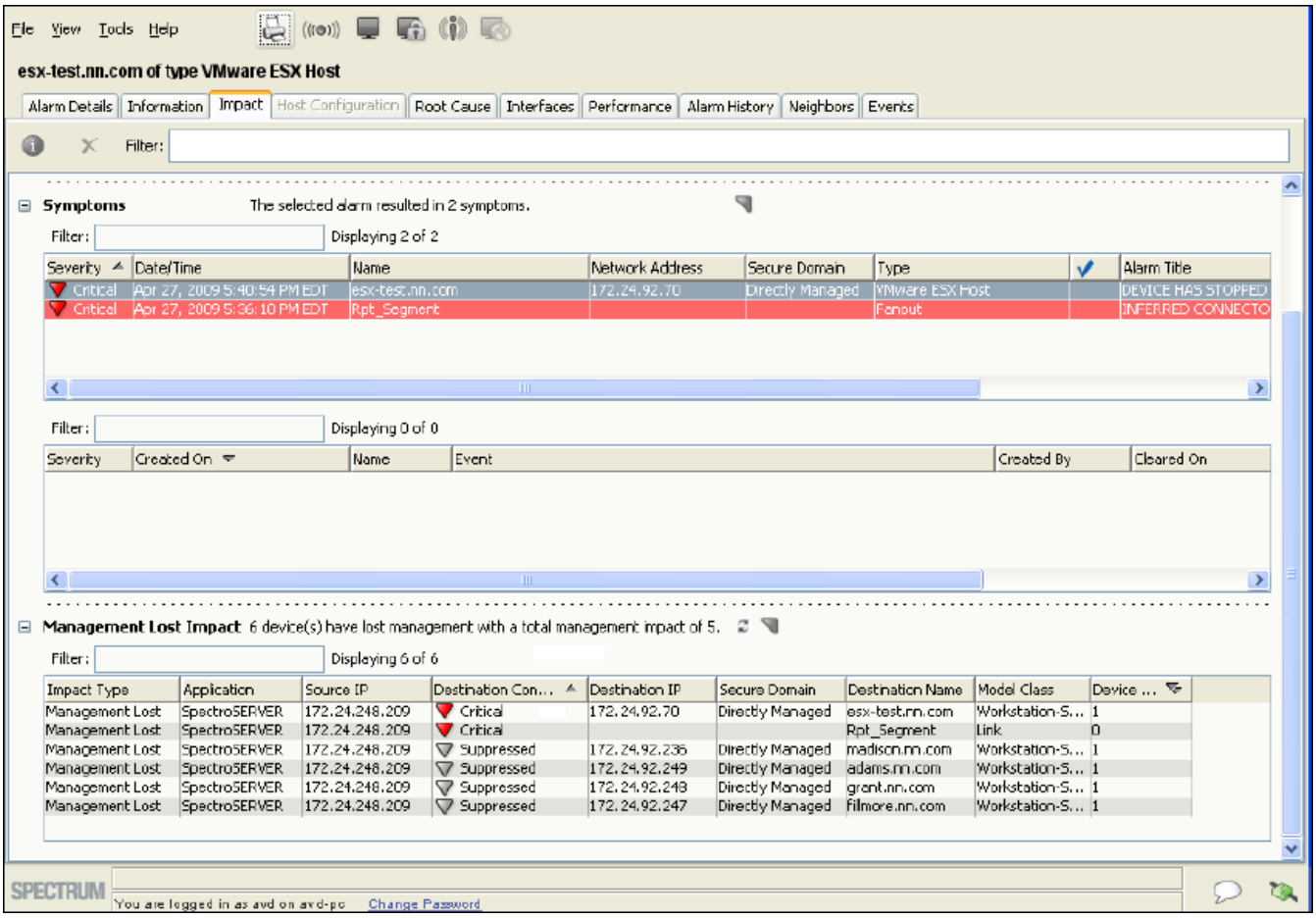
If DX NetOps Spectrum loses contact with a vCenter model, DX NetOps Spectrum loses vCenter data about all virtual models managed by that vCenter server. To isolate the problem, Virtual Host Manager determines the root cause as follows:

1. DX NetOps Spectrum generates Proxy Management Lost alarms for all virtual models managed by that vCenter server, including virtual machines, ESX hosts, ESX service consoles, datacenters, resource pools, and clusters. DX NetOps Spectrum also generates a separate Proxy Unavailable alarm that vCenter server model.
2. The virtual machine alarms are correlated to their corresponding ESX model alarm.
3. The ESX, datacenter, resource pool, and cluster alarms are correlated to the vCenter model Proxy Unavailable alarm.
4. The vCenter alarm is correlated to another alarm generated by standard DX NetOps Spectrum fault management, such as the alarms created for the following situations:
   - Lost management of vCenter (that is, a problem occurred with the remote SystemEDGE agent)
   - Machine contact is lost
   - vCenter is in maintenance mode

Determining Virtual Machines Affected by ESX Outages

When contact with an ESX is interrupted or the ESX goes down, all virtual machines hosted by the ESX are affected. Because vCenter cannot communicate with the ESX to get usage information, you might not receive alarms for a critical virtual machine that is hosted on that ESX. To find out whether a critical virtual machine is affected, access a list of affected virtual machines on the Impact tab of the alarm. The following views are available:

- Symptoms view -- displays all symptom alarms that the affected virtual machines generate
- Management Lost Impact view -- lists the virtual machines that are affected by the alarm



Microsoft Hyper-V

This section is for Microsoft Hyper-V virtualization technology users and describes how to use Virtual Host Manager to manage your virtual entities created with Hyper-V.

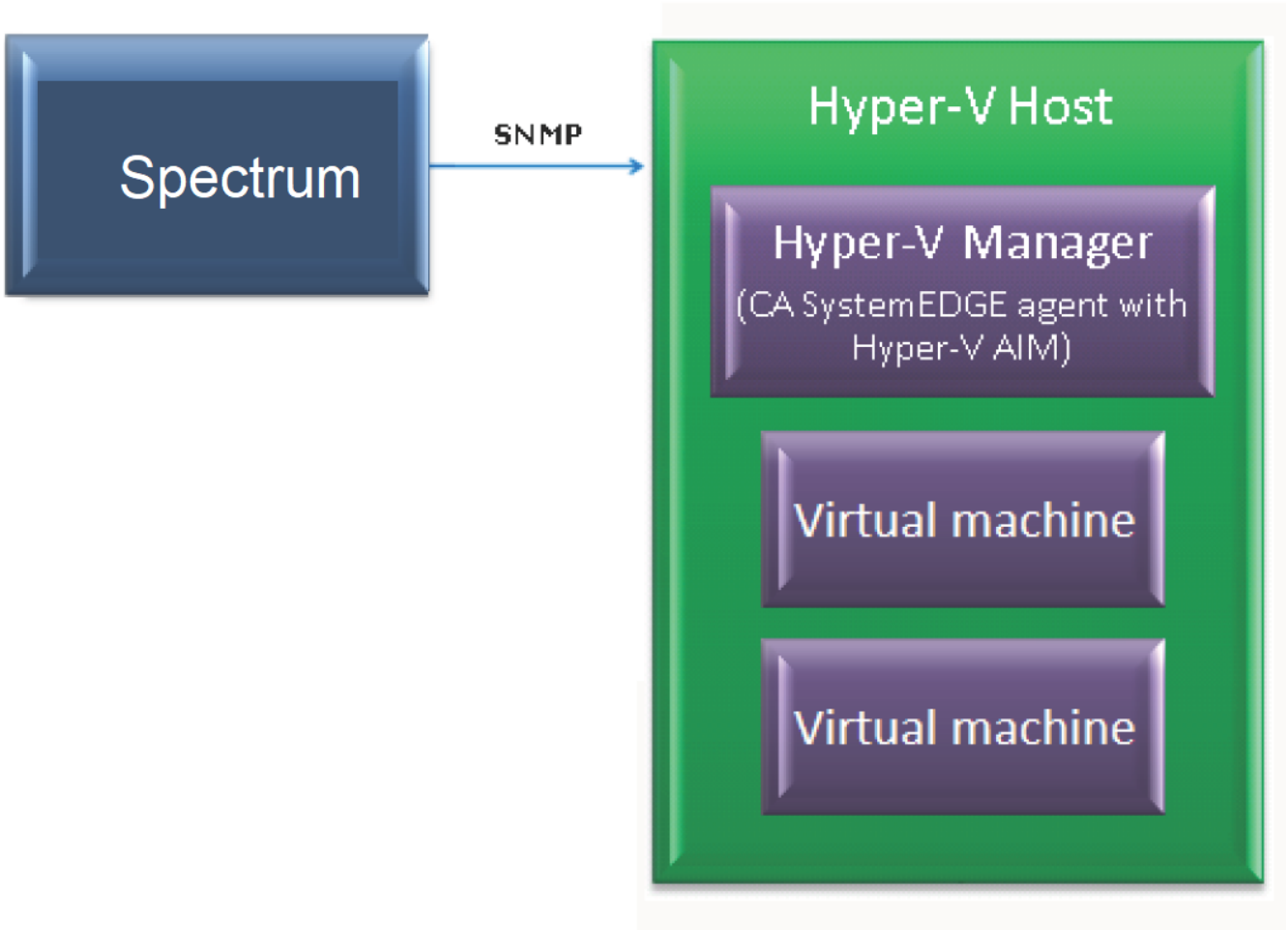
How Virtual Host Manager Works with Hyper-V

The Microsoft Hyper-V AIM collects the data of the monitored Hyper-V resources through server-internal queries without accessing the network. Therefore, the CA eHealth SystemEDGE agent and the Hyper-V AIM must run on each Microsoft Hyper-V Server that you want to monitor through DX NetOps Spectrum.

If your Microsoft Hyper-V virtual machine is a Windows platform virtual machine, we recommend installing the Microsoft Hyper-V integration services on each virtual machine in your Microsoft Hyper-V environment. The Hyper-V integration services optimize the virtualization of virtual machines. Without these tools, many features are not available.

The Microsoft Hyper-V Server provides the functionality to create, run, and manage virtual machines. The Hyper-V AIM and CA eHealth SystemEDGE agent integrates with the Microsoft Hyper-V Server and collects data for Hyper-V monitoring through DX NetOps Spectrum.

The following diagram shows how DX NetOps Spectrum gathers information about your Microsoft Hyper-V virtual environment using the CA eHealth SystemEDGE agent with the Hyper-V AIM loaded:



As shown in the diagram, the process to gather information about your Microsoft Hyper-V virtual environment is as follows:

1. The Microsoft Hyper-V management operating system resides on the Microsoft Hyper-V Host in your virtual environment, storing detailed data about each host and their virtual machines.
2. The Microsoft Hyper-V Manager, which contains the CA eHealth SystemEDGE agent with the Microsoft Hyper-V AIM loaded, resides on the Hyper-V host server. With that AIM loaded, the CA eHealth SystemEDGE agent communicates with the Microsoft Hyper-V management operating system to gather the details about your virtual environment.
3. Periodically, DX NetOps Spectrum retrieves the information from the Hyper-V Manager and uses it to model and monitor your virtual entities.

Models Created for Hyper-V

Virtual Host Manager provides several models to represent the components of your Microsoft Hyper-V virtual technology network. Understanding the following basic models can help you better understand Discovery and how the virtual environment interfaces with your physical environment.

Virtual Host Manager includes the following models and icons for Hyper-V devices:

- **Hyper-V Manager**
  Represents a server that contains the CA eHealth SystemEDGE agent with the Hyper-V AIM loaded. There can be only one Hyper-V Manager per Hyper-V Host.

Icon:

NOTE
The Hyper-V management operating system is represented in the virtual topology as part of the Hyper-V Manager model. The *Hyper-V management operating system* is the original operating system running on the Hyper-V Host. Microsoft Hyper-V uses this operating system to configure the hosted Hyper-V virtual machines. As appropriate, this Hyper-V Manager model is repeated in the hierarchy and topology views to represent the Hyper-V management operating system.

- **Hyper-V Host**
  Represents a Hyper-V Host, as configured in your Hyper-V virtualization technology. A *Hyper-V Host* is a physical computer that uses Microsoft Hyper-V virtualization software to run virtual machines. Hosts provide the CPUs and memory resources that Hyper-V virtual machines use. They also give these virtual machines access to storage and network connectivity. These models serve as container models within the topology views, helping to group your virtual entities into a separate view while showing where the virtual environment interfaces with your physical network. The Hyper-V Host cannot be contacted directly for status information. Instead, the status of these models is inferred from the status of its contained items.
  Icon:

- **Hyper-V virtual machine**
  Represents a Hyper-V virtual machine, as configured in your Hyper-V virtualization technology. A Hyper-V *virtual machine* is a software computer that, like a physical computer, runs an operating system and applications. A virtual machine dynamically consumes resources on its physical host, depending on its workload.
  Icon:

Discovering Hyper-V Networks

This section describes the Discovery and modeling process for Virtual Host Manager. These tasks are typically performed by the Virtual Host Manager administrator.

How to Configure Discovery Options

After Virtual Host Manager is installed, you can configure Virtual Host Manager for Hyper-V Discovery. Configuring your preferences helps ensure that Virtual Host Manager models your virtual devices correctly.

To configure your installation of Virtual Host Manager for Hyper-V Discovery, select your preferences for the following options:

- Maintenance Mode for New Virtual Machines -- Lets you decide which newly discovered virtual machines to place into maintenance mode until you are ready for DX NetOps Spectrum to manage them.
- Allow Device Model Deletes During Hyper-V Discovery -- Controls how DX NetOps Spectrum handles Hyper-V host and Hyper-V virtual machine models when Microsoft Hyper-V no longer manages them.
- Search for Existing Models -- Determines which secure domains Virtual Host Manager searches during a Hyper-V Discovery.
- Discover SNMP-Capable Devices -- Controls how SNMP-capable devices are modeled during Hyper-V Discovery. By default, new models are initially created as VHM models only. But, this option lets you override the default and immediately create SNMP models for devices that meet the necessary criteria.
- Retain SNMP-enabled Virtual Machines During Hyper-V Manager Deletion -- Controls how DX NetOps Spectrum handles SNMP-enabled virtual machine models when a Hyper-V Manager model is deleted.

Configure Maintenance Mode for New Hyper-V Virtual Machines

Virtual Host Manager automatically models the virtual machines that are managed by Microsoft Hyper-V. DX NetOps Spectrum attempts to manage all models discovered. However, some newly discovered Hyper-V virtual machines are not ready for DX NetOps Spectrum management when they are initially modeled. To prevent undesired alarms on new Hyper-V virtual machine models, you can decide which new models are immediately placed into maintenance mode. Later, you can manually disable maintenance mode on individual models when you are ready for DX NetOps Spectrum to manage these devices.

Follow these steps:

1. Open Virtual Host Manager in the Navigation panel.
   A details page opens in the Contents panel for the selected Virtual Host Manager.
2. Click the Information tab.
3. Expand the Configuration, Hyper-V, Hyper-V Discovery subview.
4. Click Set in the 'Maintenance Mode for New Hyper-V Virtual Machines' field and select one of the following options:
   - **Place non-enabled VMs in Maintenance Mode**
     (Default) Applies maintenance mode to only non-enabled Hyper-V virtual machine models on initial Hyper-V Discovery.
   - **Place all VMs in Maintenance Mode**
     Applies maintenance mode to all newly discovered Hyper-V virtual machine models upon initial Hyper-V Discovery.

   Your setting is saved and newly discovered Hyper-V virtual machine models created by Virtual Host Manager are placed into maintenance mode per your selection.

Manage Device Models for Devices Deleted from Microsoft Hyper-V

The devices and the relationships among them change frequently in virtual environments. Maintaining accurate and timely data about your virtual environment in DX NetOps Spectrum is challenging. For example, when a Hyper-V virtual machine is removed, DX NetOps Spectrum removes the corresponding device models from Virtual Host Manager in the Navigation panel. However, should DX NetOps Spectrum keep or delete the model? You can select settings to control model deletion.

> **WARNING**
> When models are deleted, all notes or other customizations on those models are lost. Disable this option if models are likely to be recreated in your Hyper-V environment later.

Follow these steps:

1. Open Virtual Host Manager in the Navigation panel.
   The main details page opens in the Contents panel for the selected Virtual Host Manager.
2. Click the Information tab.

3. Expand the Configuration, Hyper-V, Hyper-V Discovery subview.
4. Click Set in the 'Allow Device Model Deletes During Hyper-V Discovery' field and select one of the following options:
   – **Yes**
   (Default) Deletes the models that correspond to entities no longer managed by your Microsoft Hyper-V environment.
   – **No**
   Places Virtual Host Manager models in the LostFound container when their corresponding entity is no longer managed by your Hyper-V environment, but the models are not deleted from DX NetOps Spectrum.

   NOTE
   Models with more associations, such as a model that is included in a Global Collection, are handled differently. These models are removed from the Universe, but they are not moved to the LostFound container.

Your setting is saved, and device models are handled accordingly after the device is deleted from your Hyper-V environment.

Configure Model Searches Across Secure Domains (Hyper-V)

Before creating new models, Hyper-V Discovery attempts to locate models in the SpectroSERVER. In an environment with Secure Domain Manager deployed, Hyper-V Discovery searches for models within the same secure domain as your Hyper-V Manager. This domain is the "local" domain. However, some of your virtual environment devices can exist within a different secure domain. In this case, you can configure Hyper-V Discovery to search all secure domains for existing models.

Follow these steps:

1. Open Virtual Host Manager in the Navigation panel.
   A details page opens in the Contents panel for the selected Virtual Host Manager.
2. Click the Information tab.
3. Expand the Configuration, Hyper-V, Hyper-V Discovery subview.
4. Click Set in the 'Search for Existing Models' field and select from the following options:
   – **In Hyper-V Manager's Secure Domain**
   (Default) Searches for existing models within the same secure domain as the Hyper-V Manager server.
   – **In All Secure Domains**
   Searches for existing models within all secure domains managed by the SpectroSERVER. Select this option only in the following situations:
   • All devices have unique IP addresses
   • When secure domains are used for security purposes or to isolate network traffic

   NOTE
   Do not select this option for a NAT environment.

   Your setting is saved and Hyper-V Discovery searches for existing models in DX NetOps Spectrum according to your selection. If duplicate models (that is, models that share the same IP address) exist in multiple secure domains, Virtual Host Manager does the following:
   – Selects the model in the local secure domain, if available.
   – If a duplicate model does not exist in the local domain, Virtual Host Manager randomly selects a model from another secure domain.
   – In both cases, Virtual Host Manager generates a minor alarm for the duplicate IP addresses on the Hyper-V Manager model.

Configure SNMP Modeling Preferences (Hyper-V)

SNMP-capable virtual machines support enriched device monitoring, such as process and file system monitoring capabilities. However, SNMP agents can be costly and time-consuming to deploy. By default, Hyper-V Discovery creates

virtual machines as VHM models, which you can later upgrade to SNMP models. However, you can configure Hyper-V Discovery to model all new SNMP-capable devices as SNMP models. Although Hyper-V Discovery may take longer to complete, initially modeling these as SNMP models avoids manually upgrading these models later.

> **WARNING**
> Enable SNMP modeling *before* you model your Hyper-V Manager servers. If you model the Hyper-V Manager servers first, all child models are created as VHM models, which must be manually upgraded to SNMP models.

Follow these steps:

1. Open Virtual Host Manager in the Navigation panel.
   The main details page opens in the Contents panel for the selected Virtual Host Manager.
2. Click the Information tab.
3. Expand the Configuration, Hyper-V, Hyper-V Discovery, SNMP Discovery subview.

   > **WARNING**
   > Follow the steps in the subview to prepare your devices and DX NetOps Spectrum for SNMP Discovery. If devices are not properly prepared before Hyper-V Discovery, Virtual Host Manager cannot create SNMP models.
4. Click Set in the 'Discover SNMP-Capable Devices' field and select from the following options:
   - **Yes**
     Enables SNMP modeling during Hyper-V Discovery. Only devices that meet the criteria specified in the SNMP Discovery subview text are modeled as SNMP devices. Applies to *new* models only.
   - **No**
     (Default) Models all new devices found during Hyper-V Discovery as VHM models. You can manually upgrade these models to SNMP models later.

   Your setting is saved and new devices are modeled in Virtual Host Manager according to your selection.

Manage SNMP-Enabled Virtual Machine Models After Hyper-V Manager Deletion

By default, SNMP-enabled devices are deleted from DX NetOps Spectrum when the following items are deleted:

- Hyper-V Manager model for the device
- Hyper-V folder in the Navigation panel

SNMP-enabled device models can include significant customizations that you want to retain. You can adjust your settings to avoid deleting these models. They are placed into the LostFound container for later use.

Follow these steps:

1. Open Virtual Host Manager in the Navigation panel.
   The main details page opens in the Contents panel for the selected Virtual Host Manager.
2. Click the Information tab.
3. Expand the Configuration, Hyper-V, Hyper-V Discovery subview.
4. Click Set in the ‘Retain SNMP-enabled Virtual Machines During Hyper-V Manager Deletion' field and select one of the following options:
   - **Yes**
     Retains SNMP-enabled virtual machine models in the LostFound container when their Hyper-V Manager or the Hyper-V folder is deleted.

     > **NOTE**
     > Models with more associations, such as a model that is included in a Global Collection, are handled differently. These models are removed from the Universe, but they are not moved to the LostFound container.
   - **NO**
     (Default) Deletes all virtual machine models when their Hyper-V Manager or the Hyper-V folder is deleted.

Your setting is saved. SNMP-enabled device models are handled according to your selection when Hyper-V Manager models or the Hyper-V folder is deleted.

How to Discover and Model Your Virtual Environment (Hyper-V)

To monitor your virtual environment, discover and model your virtual entities -- Hyper-V Managers, Hyper-V Hosts, and Hyper-V virtual machines. Modeling these entities in Virtual Host Manager lets you view your complete network topology in one tool, showing the relationships between your physical and virtual components.

The main steps for modeling your virtual environment are as follows:

1. Run a standard Discovery.
   The purpose of this Discovery is to ensure that the upstream routers and switches are modeled before Hyper-V Discovery runs. Or, if the SNMP Modeling option is disabled, this step can also model the SNMP-capable virtual machines and Hyper-V servers. When modeling these entities, be sure that your modeling options are set correctly to support Virtual Host Manager.
2. Upgrade the CA eHealth SystemEDGE model.
   This step is required only when your CA eHealth SystemEDGE agent on the Hyper-V server was modeled in a release before 9.2.1.
3. Let Hyper-V Discovery run.
   When you model the CA eHealth SystemEDGE agent (with the Hyper-V AIM) on the Hyper-V server, Hyper-V Discovery begins automatically. Each of these Hyper-V Server models has its own Hyper-V Discovery process. The purpose of Hyper-V Discovery is to find the virtual entities managed by Hyper-V, model the ones that do not exist, and place them in the Virtual Host Manager view of the Navigation panel.

Run Discovery (Hyper-V)

To discover your Hyper-V environment, run the standard DX NetOps Spectrum Discovery. This Discovery ensures that the upstream routers and switches are modeled so that later connections from the virtual entities can be established. You can also model the SNMP-capable Hyper-V virtual machines during DX NetOps Spectrum Discovery.

NOTE
Modeling SNMP-capable Hyper-V virtual machines is necessary during DX NetOps Spectrum Discovery only when the SNMP Modeling option is disabled during Hyper-V Discovery.

NOTE
Only an administrator performs this task.

Follow these steps:

1. Open the Discovery console.

   NOTE
   Before modeling, be sure that you know the correct community strings, IP addresses, and port numbers for any SNMP agents that run on a nonstandard port.
2. In the Navigation panel, click the Creates a new configuration icon  .
3. Configure your options to support virtual network modeling, as follows:
   a. Click the Modeling Options button in the Modeling Options group.
      The Modeling Configuration dialog opens.
   b. Click the Protocol Options button.
      The Protocol Options dialog opens.
   c. Select the ARP Tables for Pingables option, and click OK.

The Modeling Configuration dialog opens.

d. (Optional) Click the Advanced Options button in the Advanced Options group. Add your nonstandard SNMP ports (such as, the CA eHealth SystemEDGE agent port), and click OK.

4. Enter individual IP addresses or the beginning and ending IP addresses in the IP Boundary List fields and click Add.

NOTE
Be sure that the range of IP addresses includes all servers with CA eHealth SystemEDGE and Hyper-V AIM installed and the interconnecting switches and routers. Or you can include the SNMP-capable Hyper-V virtual machines that require SNMP models.

5. Enter any additional values in the Discovery console, and click Discover.
The following models are created and added to your network topology in DX NetOps Spectrum:
   - Hyper-V Manager servers and the switches and routers that connect them to your network -- Information about your virtual environment comes from the Hyper-V Manager. When these Hyper-V Manager models exist in DX NetOps Spectrum, Hyper-V Discovery can begin.
   - Hyper-V Hosts and Hyper-V virtual machines -- If you decide not to model these entities with DX NetOps Spectrum Discovery, Hyper-V Discovery creates them as VHM models.

NOTE
You can also manually model your virtual network by IP address. In this case, we recommend modeling the upstream devices first. Modeling in the correct order ensures that the relationships among these entities are built correctly in the topology. For more information about how to perform a Discovery, see Modeling and Managing Your IT Infrastructure section.

Upgrade the CA SystemEDGE Model (Hyper-V)

The CA eHealth SystemEDGE agent could have been modeled in DX NetOps Spectrum before installing Virtual Host Manager or before the Hyper-V AIM was loaded on the agent. In this case, the existing CA eHealth SystemEDGE model is not compatible with Virtual Host Manager. Upgrade the model so that Virtual Host Manager can access the Hyper-V AIM capabilities in CA eHealth SystemEDGE. *This procedure is not required if the CA eHealth SystemEDGE agent with Hyper-V AIM is modeled after installing DX NetOps Spectrum.*

To upgrade the CA eHealth SystemEDGE model, right-click the model and select Reconfiguration, Reconfigure Model.

The CA eHealth SystemEDGE model is upgraded to support the Hyper-V AIM.

NOTE
You can also send a reconfigure model action to CA eHealth SystemEDGE using the CLI. For more information, see Modeling and Managing Your IT Infrastructure section.

How Hyper-V Discovery Works

Hyper-V Discovery is a specialized discovery process that gathers detailed information about your virtual environment. The purpose of Hyper-V Discovery is to obtain the virtual entities managed by Microsoft Hyper-V, model the ones that do not exist in DX NetOps Spectrum, and place them under Virtual Host Manager in the Navigation panel.

A key benefit of Hyper-V Discovery is that it runs automatically in the background, continually keeping your virtual environment data updated in DX NetOps Spectrum. Understanding how Hyper-V Discovery works reinforces the importance of properly installing and modeling the various components of Virtual Host Manager.

The Hyper-V Discovery process works as follows:

1. Immediately after the CA eHealth SystemEDGE agent and Hyper-V AIM are installed, the Hyper-V AIM communicates with the Hyper-V Host to gather information about the virtual entities it manages. The Hyper-V AIM stores this information.

WARNING
The CA eHealth SystemEDGE agent and Hyper-V AIM must be installed so that CA eHealth SystemEDGE, Hyper-V virtualization technology, and DX NetOps Spectrum can communicate. If they cannot, Hyper-V Discovery cannot run.

2. During DX NetOps Spectrum Discovery, DX NetOps Spectrum creates a Hyper-V Manager model for each server in Step 1 and enables DX NetOps Spectrum to handle communication between DX NetOps Spectrum and the CA eHealth SystemEDGE agent.
3. DX NetOps Spectrum polls the Hyper-V AIM to gather the Hyper-V information that is stored in Step 1.
4. DX NetOps Spectrum begins Hyper-V Discovery and uses this information from the AIM to update modeling in the DX NetOps Spectrum Topology tab and the Virtual Host Manager hierarchy in the Navigation panel, as follows:
   a. If you enable SNMP Discovery before Step 2, Virtual Host Manager Discovery creates SNMP models for all new SNMP-capable models that meet the SNMP Discovery criteria.

      NOTE
      By default, SNMP Discovery is disabled during Hyper-V Discovery.

   b. VHM models are created for the Hyper-V Managers.
   c. Previously existing Hyper-V virtual machine models are changed to VHM models.
   d. VHM models are created for the Hyper-V virtual machines that do not exist in DX NetOps Spectrum.
   e. VHM models are created for the Hyper-V Host models, and these models group their associated Hyper-V virtual machine models in the Navigation panel, under Virtual Host Manager and the Universe topology.
   f. All models for your virtual network are added to the Virtual Host Manager portion of the Navigation panel.

      NOTE
      In a virtual environment, devices on separate ESX hosts can have the same IP address or MAC address. In this case, DX NetOps Spectrum creates duplicate models for each occurrence of an IP address or MAC address.

5. Hyper-V Discovery automatically repeats this process at each regularly scheduled Hyper-V polling interval.

   NOTE
   By default, the Hyper-V polling interval is controlled by setting the polling interval on the Hyper-V Manager model. Or you can control Hyper-V polling by using the Hyper-V server application model.

Adding SNMP Capabilities to VHM Models

SNMP-capable virtual machines support enriched device monitoring, such as process and file system monitoring capabilities. However, SNMP agents can be costly and time-consuming to deploy. When an SNMP agent is not available or SNMP Discovery is disabled, Virtual Host Manager creates Hyper-V virtual machines as VHM models.

Later, you can install an SNMP agent on any virtual machine and upgrade its modeling in DX NetOps Spectrum. Options for upgrading to SNMP models are as follows:

- **Upgrade only selected devices** -- This method works quickly when you have a small selection of models to upgrade. The VHM models and child models are deleted first. A drawback of this method is that after DX NetOps Spectrum deletes the models, you must wait for the next Hyper-V Discovery to create the new SNMP models and place them in Virtual Host Manager. Knowledge of the IP addresses for the models to upgrade is required.
- **Upgrade all SNMP-capable VHM models** -- This method upgrades models in batch. It is preferred when upgrading Virtual Host Manager to a new release. For this method, knowledge of the IP addresses of individual models is not required. Another advantage is that after DX NetOps Spectrum deletes the VHM models, the upgraded SNMP models are immediately placed in the Virtual Host Manager hierarchy without waiting for the next polling cycle. Therefore, the child models are not left unmanaged.
  One drawback of this method is that it can take a long time to complete. The time required to complete this upgrade depends on how many community strings and SNMP ports Virtual Host Manager must search when locating SNMP-capable devices.

NOTE
Virtual Host Manager attempts to identify SNMP agents on powered-up pingable virtual machines only.

WARNING
When models are deleted, all notes or other customizations on those models are lost.

Upgrade Selected VHM Models to SNMP Models

When an SNMP agent is not available or SNMP Discovery is disabled during Hyper-V Discovery, Virtual Host Manager creates Hyper-V virtual machines as VHM models. Later, you can install an SNMP agent on any virtual machine and upgrade its modeling in DX NetOps Spectrum. You must know the IP addresses for the device models to upgrade. Manually selecting models to upgrade works quickly, but all notes or customizations on these models are lost during the upgrade.

Follow these steps:

1. Deploy or enable an SNMP agent on the device, if required.
2. Model the device again using one of the following methods:
   – DX NetOps Spectrum Discovery
   – Model individual devices by IP address

   When the new SNMP-capable model is created, DX NetOps Spectrum removes the previous model from Virtual Host Manager and deletes it. At the next Hyper-V AIM polling cycle, DX NetOps Spectrum adds the SNMP-capable model to Virtual Host Manager in the Navigation panel.

   WARNING
   When models are deleted, all notes or other customizations on those models are lost.

Upgrade All VHM Models to SNMP Models (Hyper-V)

When an SNMP agent is not available or SNMP Discovery is disabled during Hyper-V Discovery, Virtual Host Manager creates Hyper-V virtual machines as VHM models. Later, you can install an SNMP agent on any virtual machine and upgrade its modeling in DX NetOps Spectrum. When upgrading in batch, DX NetOps Spectrum searches all VHM models to find models that are now SNMP-capable devices. DX NetOps Spectrum converts them to SNMP models. This method can take a long time to complete, depending on how many community strings and ports Virtual Host Manager must search.

Follow these steps:

1. Deploy or enable an SNMP agent on your devices, as required.
2. Open Virtual Host Manager in the Navigation panel.
   The main details page opens in the Contents panel for the selected Virtual Host Manager.
3. Select the Hyper-V Manager model in the Navigation panel that manages the models that you want to upgrade.
4. Click the Information tab.
5. Expand the Hyper-V Manager, DX NetOps Spectrum Modeling Control subview.
6. Click Upgrade ICMP-Only Devices.

   WARNING
   When models are deleted, all notes or other customizations on those models are lost.

Virtual Host Manager searches for VHM models managed by the Hyper-V AIM on the selected Hyper-V Manager device. Virtual Host Manager upgrades the ICMP-only devices that meet the criteria for SNMP devices and places them within the Virtual Host Manager hierarchy.

Move a Hyper-V Virtual Machine to a Different Hyper-V Host

Moving a Hyper-V virtual machine from one Hyper-V Host to another can result in lost data. The risk depends on your Virtual Host Manager configuration. The Hyper-V AIM does not support virtual machine migration. To Virtual Host Manager, a move is two events -- the virtual machine is deleted from the original Hyper-V Host, and a new virtual machine is added to the new Hyper-V Host. In this case, Virtual Host Manager deletes the original virtual machine model and creates a new one. If you customized the original model, deleting it can result in lost data. You can avoid this data loss when you configure your Virtual Host Manager settings correctly before moving the virtual machine.

Follow these steps:

1. Change the 'Allow Device Model Deletes During Hyper-V Discovery' option to No.

   NOTE
   Disabling this option means that DX NetOps Spectrum does not delete the virtual machine model from DX NetOps Spectrum when the model is removed from Virtual Host Manager management.
2. Use the Microsoft Hyper-V virtualization technology to remove the virtual machine from the original Hyper-V Host.
3. Wait for Virtual Host Manager to reflect the changes in the Navigation panel.
4. Use the Microsoft Hyper-V virtualization technology to add the virtual machine to the other Hyper-V Host.
   When Hyper-V Discovery finds the new virtual machine, Virtual Host Manager reconciles it with the existing model. Virtual Host Manager places that model into Virtual Host Manager management.
5. (Optional) Change the 'Allow Device Model Deletes During Hyper-V Discovery' option back to Yes on the originating Hyper-V Manager model.
   The virtual machine is successfully moved.

Viewing Your Hyper-V Virtual Environment

This section describes concepts for viewing your Hyper-V virtual environment and the associated alarms. The basic steps are no different from the standard DX NetOps Spectrum procedures. However, this section describes conceptual differences and details that only apply to the Hyper-V virtual technology.

Viewing Your Hyper-V Virtual Network

On the Explorer tab, the Virtual Host Manager node displays a hierarchical tree structure that helps you visualize the logical relationships among your virtual environment resources.

Using this information, you can see how resources are shared among your Hyper-V Managers. This information can help you identify opportunities to reorganize and optimize your virtual environment. The hierarchy also provides a quick way to monitor the performance of resources and troubleshoot alarms.

Because Virtual Host Manager is not aware of a DSS environment, it is located within a landscape hierarchy. The following example shows where Virtual Host Manager appears on the Explorer tab in the Navigation panel and illustrates the virtual environment hierarchy:

```
[-] <ss> host
    [+] Universe
    [-] Virtual Host Manager
        [-] Hyper-V
            [+] Hyper-V Manager 1
            [-] Hyper-V Manager 2
                [-] Hyper-V Host
                     .  Hyper-V Manager 2 (management operating system)
                     .  Hyper-V virtual machine 1
                     .  Hyper-V virtual machine 2
```

NOTE
The Hyper-V management operating system is represented in the virtual topology as part of the Hyper-V Manager model.

Virtual Host Manager is the root node for the entire virtual environment that is managed by this SpectroSERVER. Selecting this node in the Navigation panel displays Virtual Host Manager details in the Contents panel. You can view details such as events and alarms related to your virtual environment.

Directly under Virtual Host Manager, virtual environments are organized within folders that represent the associated technology. In the example hierarchy above, the Hyper-V folder contains the portion of the virtual environment that was created using Microsoft Hyper-V virtualization technology. In this folder, Virtual Host Manager lists all Hyper-V Manager hosts managed by this SpectroSERVER.

Each Hyper-V Manager contains only the portion of the entire virtual environment that it manages. Selecting a Hyper-V Manager in the Navigation panel displays details in the Contents panel, such as the Hyper-V Hosts or Hyper-V virtual machines managed by the selected Hyper-V Manager.

Under each Hyper-V Manager, the hierarchy represents the logical relationships between the following entities:

- **Hyper-V Hosts**
  A Hyper-V Host contains the Hyper-V virtual machines that it manages. Selecting a Hyper-V Host in the Navigation panel displays details in the Contents panel such as events and alarms related to the Hyper-V Host, memory usage, status, and more.
- **Hyper-V Management Operating System**
  The Hyper-V Management Operating System model appears as a child to its corresponding Hyper-V Host model and is always a leaf node on the Virtual Host Manager hierarchy tree. This model shares the name and model type of its parent. Although this model appears to be the same as the Hyper-V Manager model, the instance that appears under the Hyper-V Host model represents the management operating system running on the Hyper-V Host. Hyper-V uses this operating system to configure the hosted Hyper-V virtual machines. Selecting a Hyper-V Management Operating System model in the Navigation panel displays details in the Contents panel, including system status and CPU and memory usage.
- **Hyper-V Virtual Machines**
  A Hyper-V virtual machine is always a leaf node on the Virtual Host Manager hierarchy tree. Selecting a Hyper-V virtual machine in the Navigation panel displays details in the Contents panel, such as events and alarms that are related to the virtual machine, memory usage, and status.

Understanding the Hyper-V Virtual Topology

The Hyper-V Manager/Management Operating System, Hyper-V Host, and Hyper-V virtual machine models created for your virtual environment are integrated into the topology view. Hyper-V Host models automatically group their associated Hyper-V virtual machines. The topology shows how these Hyper-V virtual machines are connected to your physical network entities.

NOTE
The Hyper-V management operating system is represented in the virtual topology as part of the Hyper-V Manager model.

The following example shows how these models can appear on the Explorer tab in the Navigation panel under the Universe group:

```
[-] Universe
     .  Physical switch 1
     .  Physical switch 2
    [-] Hyper-V Host
         .  Fanout 1
```

```
. Fanout 2
. Hyper-V Manager (management operating system)
. Hyper-V virtual machine 1
. Hyper-V virtual machine 2
. Hyper-V virtual machine 3
```

Selecting one of these models displays these relationships graphically on the Topology tab in the Contents panel.

How the Hyper-V Data is Updated in Virtual Host Manager

After DX NetOps Spectrum builds your initial Hyper-V hierarchy, your virtual network configuration can change. Virtual Host Manager continually works to keep this information accurate in DX NetOps Spectrum. For example, the following events can change your virtual network configuration:

- Creating or deleting a Hyper-V virtual machine on a Hyper-V Host
- Manually moving a Hyper-V virtual machine from one Hyper-V Host to another

To keep your information accurate, Virtual Host Manager detects these changes by polling the Hyper-V AIM. Therefore, your virtual network configuration changes, if any, are reflected in DX NetOps Spectrum at each polling cycle. DX NetOps Spectrum also receives traps from the AIM and generates the corresponding events. By reviewing the event log, you can find out when configuration changes occur, such as when a new virtual machine is created.

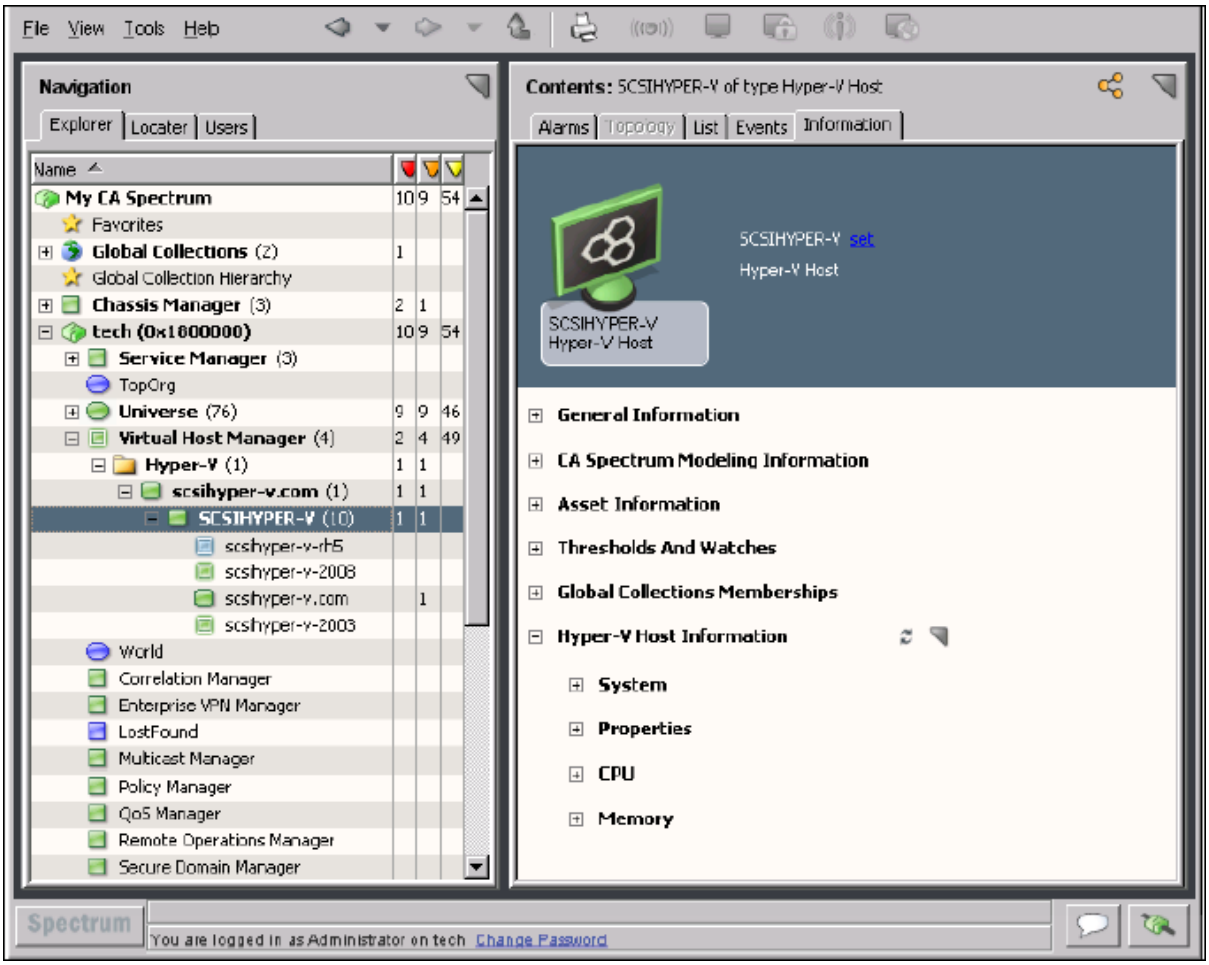
When a virtual machine is deleted, DX NetOps Spectrum removes the models from the Virtual Host Manager hierarchy on the Explorer tab. When the AIM detects an addition to your virtual network configuration, such as creating a new virtual machine or placing one into management, DX NetOps Spectrum performs the following tasks:

- Updates the placement of your virtual device models in the hierarchy of the Explorer tab
- *Automatically* rediscovers connections to the affected Hyper-V Manager and virtual machine models and associates them with the correct Hyper-V Host in the topology

> **NOTE**
> Although most components of your virtual environment are discovered automatically, the DX NetOps Spectrum administrator should initiate a new SNMP discovery to model *new* switches or routers. This discovery is necessary only when a new virtual host is configured that does not share connections with the existing virtual network models.

Custom Subviews for Virtual Entity Types (Hyper-V)

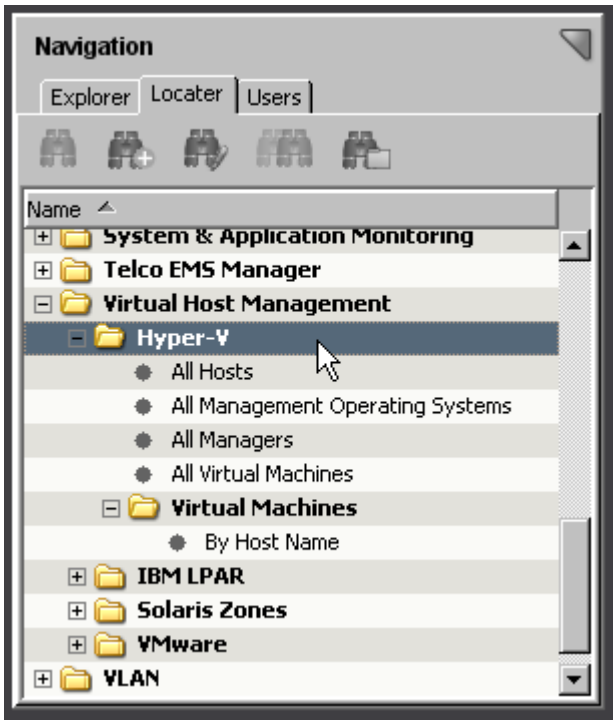
Your Virtual Host Manager models collectively provide information about your virtual environment. Individually, each model provides unique information or configuration settings, depending on the virtual entity type it represents. This custom subview appears on the Information tab in the Contents panel. These subviews can contain real-time data, such as CPU status or memory utilization. For example, the custom subview for a Hyper-V Manager is the "Hyper-V Manager" subview, as shown:



NOTE
The Hyper-V Manager model provides combined information for all virtual devices managed by the Hyper-V Manager. That is, selecting the Hyper-V Manager model in the Navigation panel displays information about the selected Hyper-V Manager host *and* combined information about all Hyper-V Hosts and Hyper-V virtual machines. This information is the same data displayed on the Information tab for each individual entity model. The combined view in the Hyper-V Manager model can provide a good overview about all of the virtual entities it manages.

Locater Tab for Hyper-V Searches

In addition to viewing details about your virtual environment on the Explorer tab, you can also use the Locater tab to run preconfigured Virtual Host Manager searches. The search options are grouped under the Virtual Host Management, Hyper-V folder on the Locater tab, as shown:



These detailed searches can help you investigate information related to virtual entities only, such as locating all Hyper-V virtual machines within a landscape.

NOTE
Although Virtual Host Manager is not DSS aware, these preconfigured searches let you select multiple landscapes to search in the search parameters.

The Locater tab in the Navigation panel includes the following searches for Virtual Host Manager information:

- **All Hosts**
  Locates all Hyper-V Host servers that have been modeled in the DX NetOps Spectrum database for your virtual network.
- **All Management Operating Systems**
  Locates all Hyper-V management operating systems that have been modeled in the DX NetOps Spectrum database for your virtual network.

  NOTE
  The Hyper-V management operating system is represented in the virtual topology as part of the Hyper-V Manager model.
- **All Managers**
  Locates servers hosting the CA eHealth SystemEDGE agent with Hyper-V AIM enabled that have been modeled in the DX NetOps Spectrum database for your virtual network.
- **All Virtual Machines**
  Locates all Hyper-V virtual machines that have been modeled in the DX NetOps Spectrum database for your virtual network.
- **Virtual Machines, By Host Name**
  Locates virtual machines in the DX NetOps Spectrum database managed by only one or a select group of Hyper-V Hosts.

Status Monitoring Options (Hyper-V)

DX NetOps Spectrum provides a wide range of options for monitoring the state of your virtual network resources. The status information available for a resource varies, depending on the type of virtual entity you are monitoring. Also, your ability to configure a status option depends on its type. For example, some status options are read-only, but others let you enable behaviors or select an alarm severity. Providing this range of options and levels of customization, DX NetOps Spectrum lets you decide how to best monitor the performance of your virtual network.

Status fields are located in the OneClick subviews. All status information for a given virtual environment is available on the Hyper-V Manager model in a tabular format. Also, each virtual entity type that has a unique model in DX NetOps Spectrum provides a subset of the same status information for easy viewing. Status-related settings, including the alert type and monitor settings, can be set from either view location.

The following tables outline the type of status information available for each virtual entity type. The Subview Locations column describes where the corresponding status fields are located in OneClick. For example, "memory" information for a Hyper-V virtual machine model is available on the Information tab in the following two locations:

- Virtual Machine Information subview for the Hyper-V virtual machine model
- Hyper-V Manager, Managed Environment, Virtual Machines subview for the Hyper-V Manager model

To explore the exact status options available for each status information type, locate the subview in OneClick.

Hyper-V Manager

Status Information Type	Subview Locations
Overall	Hyper-V Manager

Hyper-V Host

Status Information Type	Subview Locations
Overall	Hyper-V Host
CPU	Hyper-V Host, Hyper-V Manager
Memory	Hyper-V Host, Hyper-V Manager

Hyper-V Virtual Machine

Status Information Type	Subview Locations
Overall	Hyper-V virtual machine, Hyper-V Manager
Memory	Hyper-V virtual machine, Hyper-V Manager
CPU	Hyper-V virtual machine, Hyper-V Manager

How to Configure Management Options (Hyper-V)

After your virtual network is modeled, you can configure Virtual Host Manager options for viewing and managing your device models. Configuring your preferences helps ensure that Virtual Host Manager handles your virtual device models correctly and monitors only the information that is important to you.

To configure your installation of Virtual Host Manager, perform the following procedure after you discover and model your virtual network:

- Configure threshold values and other status monitoring options -- These options let you determine which information you want to monitor and how DX NetOps Spectrum manages the various events that occur in your virtual network.

Configure and Monitor Resource Status (Hyper-V)

You can monitor the status of virtual resources in OneClick. For example, you can view the total physical memory or used physical memory, and more. You can also set monitoring options, such as enabling alerts. This information can help you optimize your virtual network performance and troubleshoot alarms.

NOTE
Traps are set on and managed by the Hyper-V AIM.

You can view or configure resource status options and information for virtual devices on the Information tab.

Follow these steps:

1. Open Virtual Host Manager in the Navigation panel.
   The main details page opens in the Contents panel for the selected Virtual Host Manager.
2. Locate and click the virtual device on the Explorer tab in the Navigation panel.
   The device details display in the Contents panel.
3. Click the Information tab.
   Multiple subviews are available for viewing. Generally, the subview at the bottom of the tab includes the resource allocation and utilization information for the selected model. For example, a Hyper-V Host model displays a subview named "Hyper-V Host Information" that includes details for the specific Hyper-V Host model you selected in the Navigation panel.
4. Expand the appropriate subview.
   All available resource status details and monitoring options for the selected device model are displayed.

   NOTE
   The Hyper-V Manager model provides combined information for all virtual devices managed by the Hyper-V Manager. That is, selecting the Hyper-V Manager model in the Navigation panel displays information about the selected Hyper-V Manager host *and* combined information about all Hyper-V Hosts and Hyper-V virtual machines. This information is the same data displayed on the Information tab for each individual entity model. The combined view in the Hyper-V Manager model can provide a good overview about all of the virtual entities it manages.

Controlling Hyper-V AIM Polling

When you are tuning Virtual Host Manager performance, you can change the Hyper-V Manager polling rate or disable Hyper-V technology polling. By default, the polling attributes on the Hyper-V Manager model control the Hyper-V polling behavior. Or you can change this Hyper-V polling behavior independently. The Hyper-V technology application model, HyperVAimApp, controls your Hyper-V polling.

The following two attribute values on the application specifically control the Hyper-V polling logic:

- PollingStatus
- Polling_Interval

Both the Hyper-V Manager model and the HyperVAimApp application model contain these attributes. PollingStatus disables and enables polling while Polling_Interval controls the polling frequency. If their values are different, the HyperVAimApp application model attribute values take precedence when determining Hyper-V technology polling behavior.

This ability to set the value for the device model and application model lets you fine-tune your Hyper-V technology polling. For both PollingStatus and Polling_Interval, modifying the attribute on the Hyper-V Manager device model also changes the corresponding application model attribute when their values are the same.

Configure the Hyper-V AIM Polling Interval

You can change the Hyper-V AIM polling rate. Configure the polling interval by setting the Polling_Interval attribute on the Hyper-V technology application model.

Follow these steps:

1. Open OneClick and click the Locater tab in the Navigation panel.
2. Expand the Application Models folder and double-click 'By Device IP Address.'
   A search dialog opens.
3. Enter the IP address for your Hyper-V Manager device in the Device IP Address field, and click OK.
   A list of application models for the Hyper-V Manager appears in the Contents panel.
4. Select the HyperVAimApp application model.
   The application model details appear in the Component Details panel.
5. Click the Information tab in the Component Details panel.
6. Open the Modeling Information subview.
7. Click set in the Polling Interval (sec) field, enter a new value.

   NOTE
   Changing the Polling Interval value from any number to 0 also sets the Polling field to Off, disabling Hyper-V AIM polling. However, if you set the Polling Interval to 0 and set the Polling field to On, Hyper-V AIM polling continues, using the polling interval for the Hyper-V Manager device.

   The Hyper-V AIM polling interval setting is modified.

Disable Hyper-V AIM Polling

You can disable Hyper-V AIM polling. Disabling Hyper-V polling is the same as disabling Virtual Host Manager. Disable polling by setting the PollingStatus attribute on the Hyper-V virtual technology application model.

Follow these steps:

1. Open OneClick and click the Locater tab in the Navigation panel.
2. Expand the Application Models folder and double-click 'By Device IP Address.'
   A search dialog opens.
3. Enter the IP address for your Hyper-V Manager device in the Device IP Address field and click OK.
   A list of application models for the Hyper-V Manager appears in the Contents panel.
4. Select the HyperVAimApp application model.
   The application model details appear in the Component Details panel.
5. Click the Information tab in the Component Details panel.
6. Open the DX NetOps Spectrum Modeling Information subview.
7. Click set in the Polling field and select Off.
   Polling is disabled for the Hyper-V AIM on the selected Hyper-V Manager.

Deleting Virtual Host Manager Models (Hyper-V)

Models can be deleted from OneClick at any time for various reasons. However, Virtual Host Manager restricts your ability to delete models from the Virtual Host Manager hierarchy in the Navigation panel. To delete models manually, you have the following two options:

- Delete the Hyper-V folder or a Hyper-V Manager model in Virtual Host Manager
- Remove a virtual entity using your Microsoft Hyper-V virtualization technology

In Virtual Host Manager, models are sometimes deleted automatically. The following circumstances cause DX NetOps Spectrum to delete Virtual Host Manager models automatically:

- **Hyper-V folder deleted or Hyper-V Manager model removed from Virtual Host Manager**
  If you remove a Hyper-V Manager model or delete the Hyper-V folder from the Navigation panel, DX NetOps Spectrum deletes all related child models.
- **An entity removed from Hyper-V virtual environment**

As you delete Hyper-V Hosts and the Hyper-V Manager using your Microsoft Hyper-V virtualization technology, DX NetOps Spectrum may also delete those models and their child models from Virtual Host Manager, according to your configuration settings.

- **Upgraded models exist** -- In some cases, a Hyper-V Host is first modeled for Virtual Host Manager without SNMP capabilities. If SNMP capabilities are later added to a VHM model, the previous model is deleted and replaced with the new SNMP-capable model.

  NOTE
  Although the default setting is to delete the models, you can configure Virtual Host Manager to place the Hyper-V Host and Hyper-V virtual machine models in the LostFound container when they are removed from Virtual Host Manager. This setting is respected only when you remove an entity using your Microsoft Hyper-V virtual environment. However, this setting does not apply when you delete the Hyper-V folder, remove a Hyper-V Manager model, or upgrade a VHM model.

Alarms and Fault Isolation for Hyper-V

This section describes the traps used by Virtual Host Manager and the resulting alarms. This section also explains how Virtual Host Manager fault isolation differs from basic DX NetOps Spectrum fault isolation.

Virtual Host Manager Alarms for Hyper-V

To alert you to problems within your virtual network, DX NetOps Spectrum generates alarms during polling. Polling generates four alarms: Hyper-V Proxy Lost, Hyper-V Host Proxy Lost, Hyper-V Manager Unavailable, and Hyper-V Virtual Machine Not Running.

Forwarding Traps from CA SystemEDGE (Hyper-V)

DX NetOps Spectrum supports all traps that the Hyper-V AIM sends. These traps are initially sent to Hyper-V CA eHealth SystemEDGE model. If the destination for a trap is not the Hyper-V model, DX NetOps Spectrum forwards the trap to the correct virtual model.

NOTE
For specific event codes related to the traps, use the Event Configuration application and filter on “0x056e.” Alternately, you can launch MIB tools to view the traps in the Trap Support table for the “CAHYPERV-AIM-MIB” MIB.

DX NetOps Spectrum determines where to forward the trap by using the following process:

1. When DX NetOps Spectrum receives a trap, it uses varbind information in the trap to identify the UID for the target device.
2. DX NetOps Spectrum uses this UID to look up and locate the DX NetOps Spectrum model that is tied to a given UID. The entity type of all traps is predetermined. Depending on the results of the look-up, DX NetOps Spectrum forwards the trap as follows:
   – If it finds a DX NetOps Spectrum model of a specific type with a given UID, DX NetOps Spectrum forwards the event and corresponding alarm to the destination model.
   – If it cannot find a DX NetOps Spectrum model for a given UID, DX NetOps Spectrum generates a new generic event on the Hyper-V Manager model. This new event includes details about the trap.

     NOTE
     DX NetOps Spectrum often cannot find a related model when a trap is sent immediately after changing your virtual network entities in the Hyper-V virtualization technology. Hyper-V Discovery has not yet identified and created the corresponding model in DX NetOps Spectrum.

Traps Supported in Virtual Host Manager (Hyper-V)

All traps that are generated by the Hyper-V AIM are supported in DX NetOps Spectrum. The traps are initially sent to the Hyper-V Manager model. Then, the traps are forwarded to a corresponding virtual entity type (that is, the "destination" entity), depending on the type of trap. Using these traps, you can monitor the performance of your virtual network, resolve any resulting alarms, or trigger events.

NOTE
For more information about traps that are generated by the Hyper-V AIM, see *CA Virtual Assurance for Infrastructure Managers Administration*.

The following tables list the traps for a specific destination entity type and specify whether the trap generates an alarm.

Trap Name	Trap OID	Alarm?
hypervAimStatVMAddTrap	1.3.6.1.4.1.546.1.1.6.16501	No
hypervAimStatVMRemoveTrap	1.3.6.1.4.1.546.1.1.6.16502	No
hypervAimStatVMMigrateTrap	1.3.6.1.4.1.546.1.1.6.16505	No

Trap Name	Trap OID	Alarm?
hypervAimStatVMEnabledTrap	1.3.6.1.4.1.546.1.1.6.16504	No

Fault Management for Virtual Networks (Hyper-V)

The goal of fault isolation is to narrow down the root cause of a networking problem. Finding the root cause can help you to troubleshoot and quickly correct the problem or to correct the problem programmatically with automated scripts. Deciding which devices are the root cause of an alarm can be difficult, because problems with a single device can cause several devices in your network to generate events.

For example, losing contact with a Hyper-V Host often means that you have also lost contact with the Hyper-V virtual machines it manages. Therefore, the Hyper-V Host device model and all affected virtual machines generate alarms. Using fault isolation techniques, Virtual Host Manager correlates these alarms in an attempt to identify a single root cause.

Virtual networks provide a unique management opportunity, because they provide DX NetOps Spectrum an alternate management perspective. That is, DX NetOps Spectrum can gather information through direct contact with your virtual devices or through the virtual network management technology, Microsoft Hyper-V. This alternate management perspective enhances standard DX NetOps Spectrum fault management in two ways:

- **Enhanced Contact Lost alarms** -- Two sources of information about a device means Virtual Host Manager can pinpoint the cause and more easily correlate events to a single root cause.
- **Proxy Failure alarms** -- *Proxy management* is the act of managing network devices using an alternate management source in place of or in addition to the primary manager. For example, DX NetOps Spectrum can manage virtual network devices by contacting them directly or through the virtual technology application's contact with the devices. When Hyper-V virtualization technology loses contact with a virtual network device, Virtual Host Manager generates one of the Proxy Management Lost alarms for each device. These alarms are unique, because they are alerting you to the fact that *management* of the device through the *proxy* is affected, not the state of the device or direct (SNMP) management.

How Fault Isolation Works when Device Contact is Lost

To help you troubleshoot networking problems with your devices, DX NetOps Spectrum uses fault isolation to narrow down the root cause of an alarm. For virtual networks, Virtual Host Manager uses information from direct contact with the device plus information provided by Hyper-V virtualization technology through the Hyper-V AIM. In many cases, standard DX NetOps Spectrum fault management can pinpoint the root cause. But in special circumstances, the method for isolating problems in a virtual network go beyond the standard methods.

The type of fault isolation Virtual Host Manager uses to discover the root cause depends on which devices are alarming and the type of events the devices generate. The following scenarios describe two unique fault management situations and how DX NetOps Spectrum isolates the networking error in your virtual network.

Scenario 1: Hyper-V virtual machine is not running

In a virtual environment, the virtual management application can provide more details than DX NetOps Spectrum can discover through standard device monitoring. For example, the Hyper-V virtualization technology is aware when a Hyper-V virtual machine changes from the "running" state to something else, such as the "not running" state.

If a Hyper-V virtual machine is no longer running and DX NetOps Spectrum loses contact with it, but proxy management of the Hyper-V Manager is uninterrupted, DX NetOps Spectrum determines the root cause as follows:

1. When DX NetOps Spectrum loses contact with a Hyper-V virtual machine, it generates a Contact Lost alarm.
2. During its next polling cycle, the Hyper-V Manager model polls the Hyper-V AIM to gather information about the virtual machine. Because Hyper-V technology manages the virtual machine, it can provide a unique view into the possible cause of alarms generated by a Hyper-V virtual machine.
3. If the Hyper-V virtualization technology finds that the virtual machine is in the not-running mode, it generates a Virtual Machine Not Running alarm.

   NOTE
   This alarm is cleared upon the first Hyper-V AIM polling cycle after the virtual machine is running again.
4. Virtual Host Manager correlates the Contact Lost alarm to the corresponding Virtual Machine Not Running alarm created by DX NetOps Spectrum. Virtual Host Manager makes the Contact Lost alarm appear as a symptom of the Virtual Machine Not Running alarm.

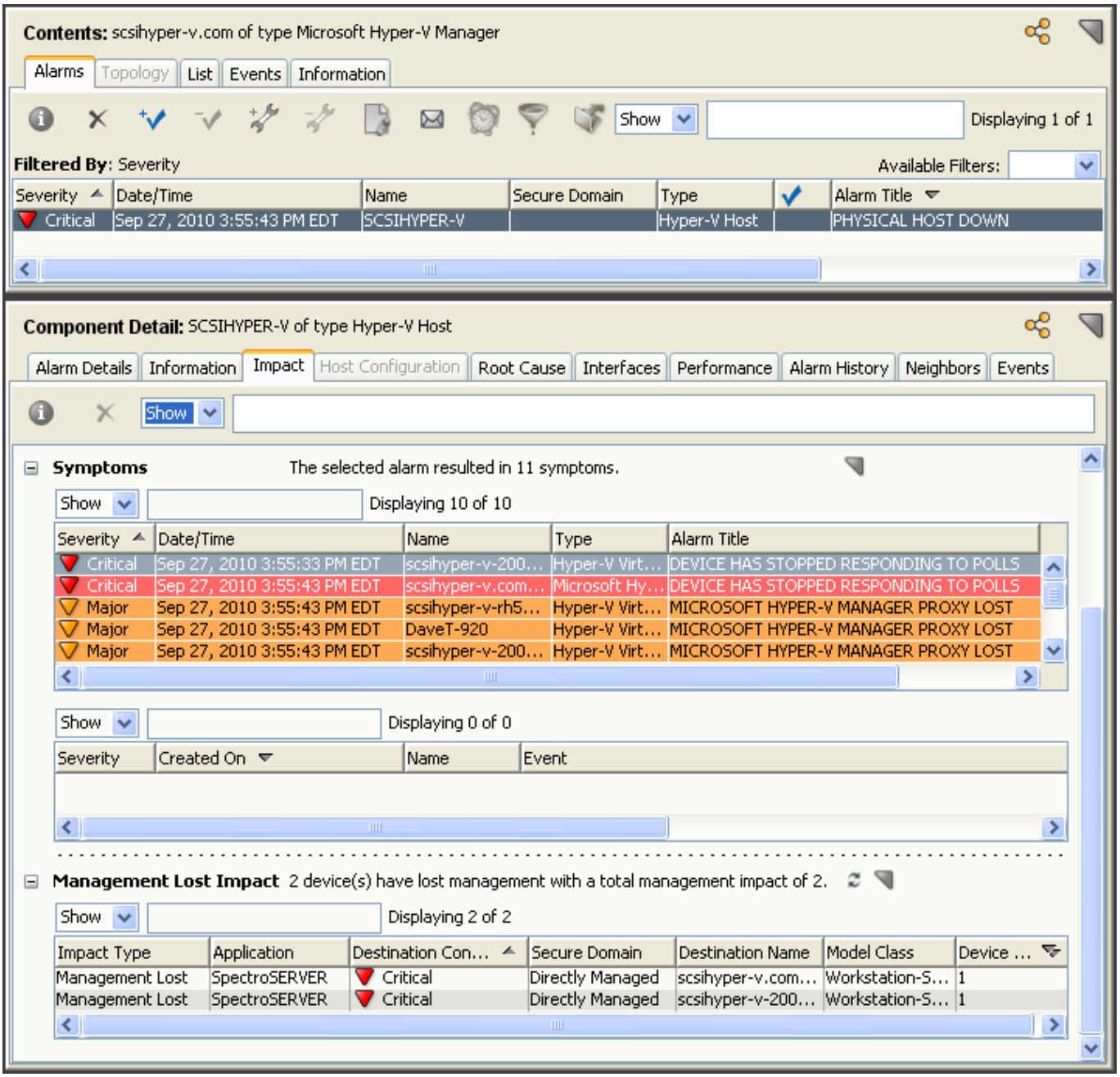
Scenario 2: Hyper-V Host is down

If DX NetOps Spectrum loses contact with a modeled Hyper-V Manager and all Hyper-V virtual machines running on that host, DX NetOps Spectrum checks the status of the upstream routers and switches. Depending on their status, DX NetOps Spectrum determines the root cause as follows:

- All upstream devices for one or more virtual machines or the Hyper-V Manager are unavailable -- Standard DX NetOps Spectrum fault isolation techniques determine the root cause, as follows:
  - Device Stopped Responding to Polls alarm -- Generated on the Hyper-V Host when at least one upstream connected device for any virtual machine or Hyper-V Manager is up.
  - Gateway Unreachable alarm -- Generated on the Hyper-V Host when *all* upstream connected devices are down.
- At least one upstream device is available for every virtual machine and Hyper-V Manager model connected to the Hyper-V Host -- DX NetOps Spectrum infers that the Hyper-V Host is the root cause and responds as follows:
  a. The Hyper-V Manager model and all Hyper-V virtual machines, ports, and fanouts that are directly connected to the Hyper-V Manager model or virtual machine models generate the standard fault isolation alarms.
  b. Virtual Host Manager creates a Physical Host Down alarm for the Hyper-V Host model.
  c. All fault isolation-related alarms that are created for the impacted devices (such as virtual machines, ports, and fanouts) are correlated to the Physical Host Down alarm, making them symptoms of the Physical Host Down alarm. These symptom alarms appear in the Symptoms table on the Impact tab for the Physical Host Down alarm.

     NOTE
     For each Hyper-V Host model, Virtual Host Manager creates a "virtual fault domain." This domain includes the Hyper-V Host, Hyper-V Manager, and virtual machines, plus all ports and fanouts directly connected to the Hyper-V Manager model or virtual machines. When the Hyper-V Host generates the Physical Host Down alarm, all standard fault isolation alarms within the domain are correlated to it. Correlating these alarms as symptoms indicates that the Physical Host Down alarm on the Hyper-V Host is the root cause.
  d. All impacted devices are listed in the Management Lost Impact table on the Impact tab for the Physical Host Down alarm.

NOTE
Devices that are suppressed do not have a corresponding alarm in the Symptoms table.



If all upstream devices for one or more virtual machines or the Hyper-V Manager go down, DX NetOps Spectrum can no longer reliably state that the fault lies with the Hyper-V Host. Therefore, DX NetOps Spectrum clears the Physical Host Down alarm and applies the standard DX NetOps Spectrum fault isolation techniques.

How Fault Isolation Works when Proxy Management is Lost (Hyper-V)

The Microsoft Hyper-V virtualization technology used to create your virtual network provides DX NetOps Spectrum a unique management opportunity. DX NetOps Spectrum can use the standard methods to contact your virtual devices directly, plus DX NetOps Spectrum can simultaneously gather virtual device information from Hyper-V technology. In this sense, the Hyper-V technology is a "proxy" from which DX NetOps Spectrum gathers virtual device information. If DX NetOps Spectrum loses direct contact with a device, it generates alarms. Likewise, if Hyper-V technology loses contact with a virtual device or if Virtual Host Manager loses contact with the Hyper-V Manager, Virtual Host Manager generates alarms -- Proxy Management Lost alarms.

In response, DX NetOps Spectrum attempts to isolate the cause of the proxy management failure. Proxy fault isolation is similar to the standard DX NetOps Spectrum fault isolation, except that these alarms alert you to the fact that *proxy* management of a virtual device is affected. Proxy management fault isolation cannot tell you whether a virtual device is up or down. However, it is important to know when contact through the proxy is lost, because you could be missing important virtual information about a device.

The type of proxy fault isolation Virtual Host Manager uses to discover the root cause depends on which devices are alarming and the type of events the devices generate. The following scenario describes a unique proxy fault management situation and how Virtual Host Manager isolates the networking error in your virtual network.

Scenario: Contact between DX NetOps Spectrum and Hyper-V Manager is lost

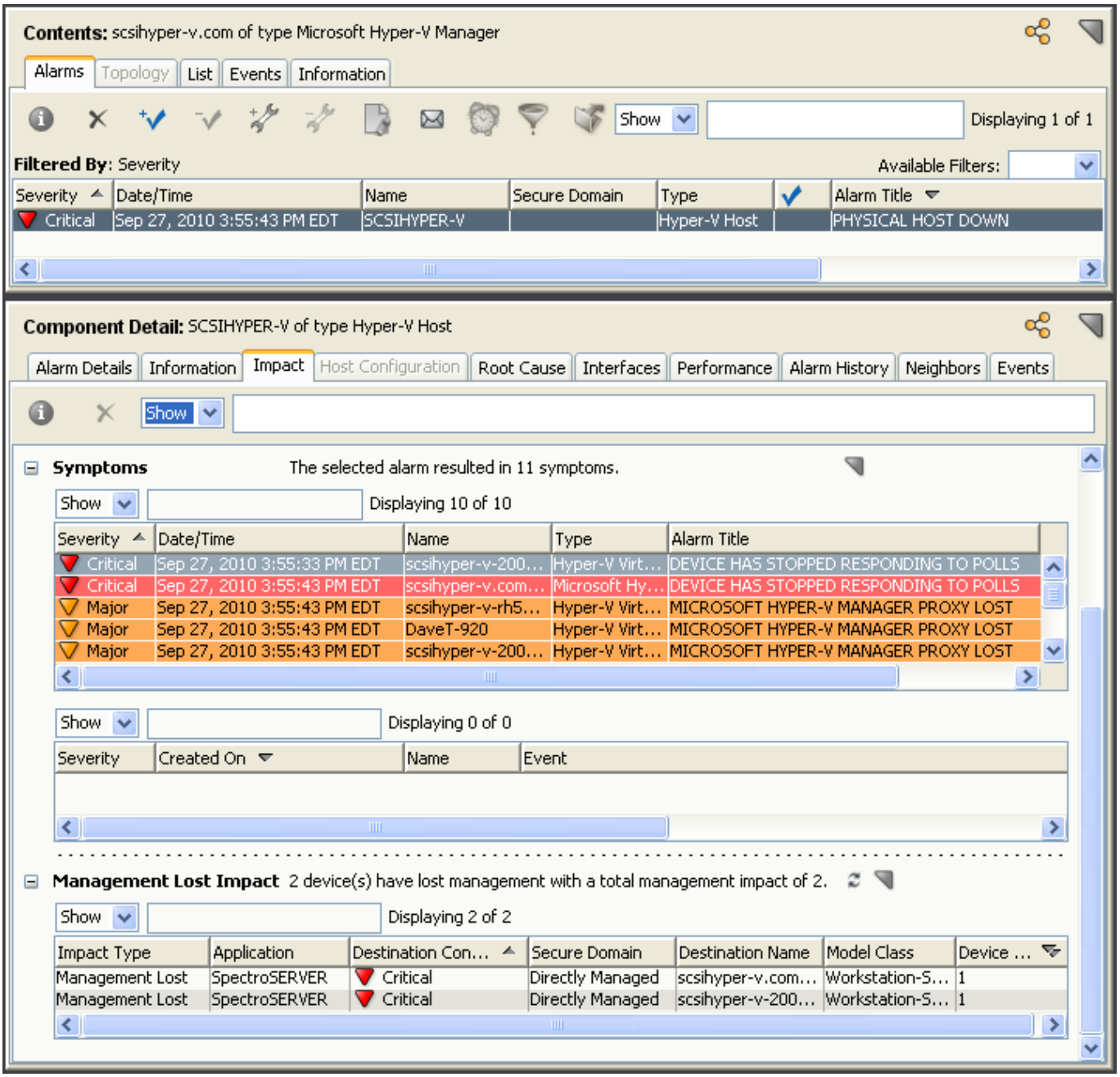
If DX NetOps Spectrum loses contact with or stops polling the Hyper-V Manager model, DX NetOps Spectrum loses the Hyper-V virtualization technology data about all virtual models managed by that Hyper-V Manager. To isolate the problem, Virtual Host Manager determines the root cause as follows:

1. DX NetOps Spectrum generates Proxy Lost alarms for all virtual models managed by that Hyper-V Manager, including virtual machines and Hyper-V Hosts. DX NetOps Spectrum also generates a separate Proxy Unavailable alarm on the Hyper-V Manager model.
2. The virtual machine alarms are correlated to their corresponding Hyper-V Host model alarm.
3. The Hyper-V Host model alarms are correlated to a Proxy Unavailable alarm for the Hyper-V Manager model.
4. This Proxy Unavailable alarm is then correlated to the root cause of the Hyper-V Manager being down. The root cause is typically an alarm generated by standard DX NetOps Spectrum fault management, such as the alarms created for the following situations:
   – Lost management of Hyper-V Manager (that is, a problem occurred with the CA eHealth SystemEDGE agent on the Hyper-V Manager host)
   – Machine contact is lost
   – Hyper-V Manager model is in maintenance mode

Determining Hyper-V Virtual Machines Affected by Hyper-V Host Outages

When contact with a Hyper-V Host is interrupted or the Hyper-V Host goes down, all Hyper-V virtual machines hosted by the Hyper-V Host are affected. Because Hyper-V technology cannot communicate with the Hyper-V Host to get usage information, you might not receive alarms for a critical virtual machine on that Hyper-V Host. To find out if a critical virtual machine is impacted, you can view a list of affected virtual machines on the Impact tab of the alarm, as follows:

- Symptoms subview -- displays all symptom alarms generated by the affected Hyper-V virtual machines
- Management Lost Impact subview -- lists the Hyper-V virtual machines impacted by the alarm



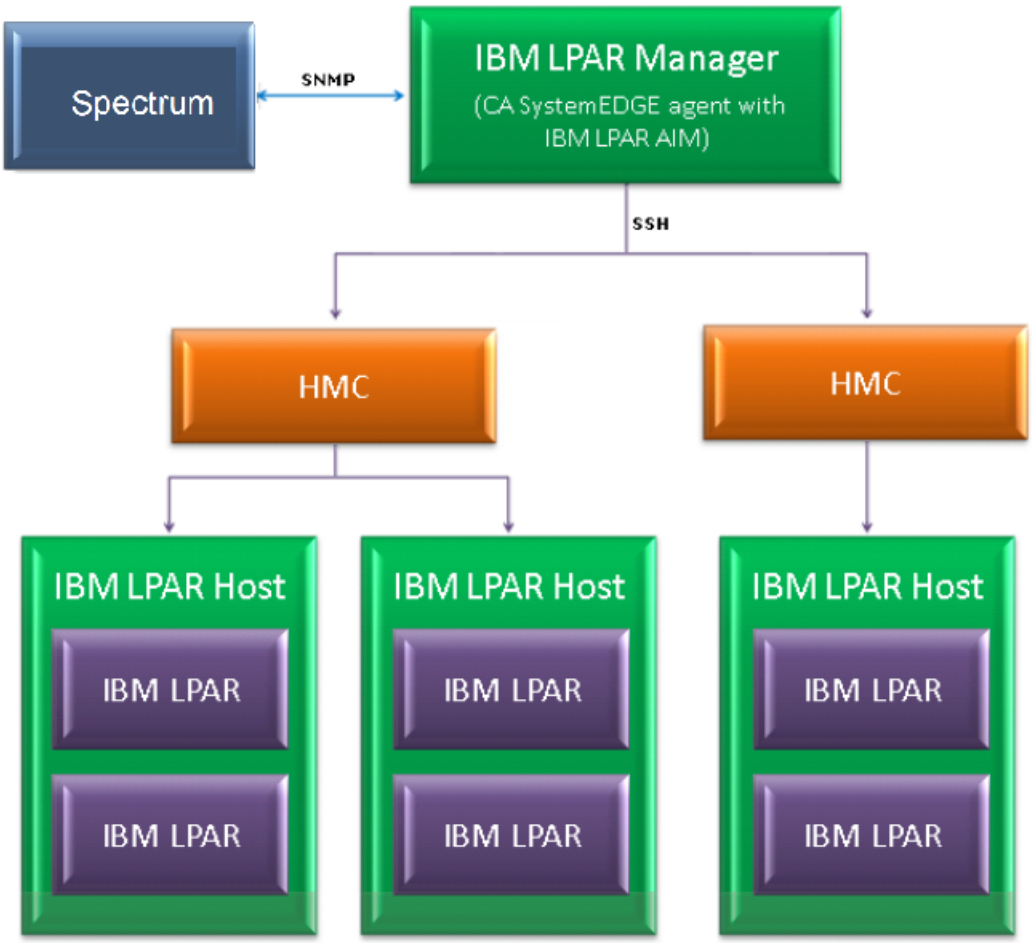
IBM LPAR

This section is for IBM LPAR virtualization technology users and describes how to use Virtual Host Manager to manage your virtual entities created with IBM LPAR technology.

How Virtual Host Manager Works with IBM LPARs

Virtual Host Manager monitors your virtual network entities seamlessly with your physical network entities. You get a full view of your network where you can troubleshoot networking issues for both types of entity. Although your virtual network entities behave like physical components, the process for monitoring those entities differs from the general DX NetOps Spectrum monitoring process. Understanding how this process works can help you locate and resolve networking issues related to your virtual network.

The *IBM LPAR Manager* in Virtual Host Manager is the CA eHealth SystemEDGE agent with the IBM LPAR AIM enabled. The IBM LPAR Managers are responsible for reporting on all of the configured IBM LPARs. Virtual Host Manager communicates with the IBM LPAR Managers to gather details about your IBM LPAR virtual environment. The following diagram shows how DX NetOps Spectrum gathers information about your IBM LPAR virtual environment using the IBM LPAR Manager:



As shown in the diagram, the process to gather information about your IBM LPAR virtual environment is as follows:

1. The HMC communicates with each IBM LPAR Host that it manages.
2. The IBM LPAR Manager uses SSH to communicate with each of its managed HMCs to gather details about your virtual environment.

NOTE
Monitor only one instance of the IBM LPAR Host with the IBM LPAR AIM. Do not manage a single IBM LPAR Host with multiple HMCs. Monitoring more than one instance can result in duplicate models in DX NetOps Spectrum.

3. Periodically, DX NetOps Spectrum communicates with the IBM LPAR Manager to retrieve these details. The IBM LPAR Manager has the CA eHealth SystemEDGE agent installed with the IBM LPAR AIM enabled. DX NetOps Spectrum uses SNMP to communicate with the CA eHealth SystemEDGE agent and uses the information to model and monitor your virtual environment in DX NetOps Spectrum.

Models Created for IBM LPARs

Virtual Host Manager provides several models to represent the components of your IBM LPAR virtual technology network. Understanding the following basic models can help you better understand Discovery and how the virtual environment interfaces with your physical environment.

Virtual Host Manager includes the following models and icons for IBM LPAR devices:

- **IBM LPAR Manager**
  Represents a server that contains the CA eHealth SystemEDGE agent with the IBM LPAR AIM loaded.
  Icon:
  
- **IBM LPAR Host**
  Represents an IBM LPAR Host, as configured in the HMC. An *IBM LPAR Host* is a physical computer that uses IBM LPAR virtualization software to host IBM LPAR instances. IBM LPAR Hosts provide the CPU and memory resources that IBM LPARs use. They also give these IBM LPARs access to storage and network connectivity. These models serve as container models within the Universe topology, helping to group your virtual entities into a separate view while showing where the virtual environment interfaces with your physical network. The IBM LPAR Host cannot be contacted directly for status information. Instead, the status of these models is inferred from the status of its contained items.
  Icon:
  
- **IBM LPAR**
  Represents an IBM LPAR, as configured in the HMC. An *IBM LPAR* is a logical partition instance configured on the IBM LPAR Host that, like a physical computer, runs an operating system and applications. An IBM LPAR dynamically consumes resources on its physical host, depending on its workload and configuration.
  Icon:
  

Discovering IBM LPAR Networks

This section describes the Discovery and modeling process for Virtual Host Manager. These tasks are typically performed by the Virtual Host Manager administrator.

How to Configure Discovery Options

After Virtual Host Manager is installed, you can configure Virtual Host Manager for IBM LPAR Discovery. Configuring your preferences helps ensure that Virtual Host Manager models your virtual devices correctly.

To configure your installation of Virtual Host Manager for IBM LPAR Discovery, select your preferences from the following options:

- Maintenance Mode for New IBM LPARs -- Lets you decide which newly discovered IBM LPAR instances to place into maintenance mode until you are ready for DX NetOps Spectrum to manage them.
- Allow Device Model Deletes During IBM LPAR Discovery -- Controls how DX NetOps Spectrum handles IBM LPAR virtualization technology models when Virtual Host Manager no longer manages them.
- Search for Existing Models -- Determines which secure domains Virtual Host Manager searches during an IBM LPAR Discovery.
- Discover SNMP-Capable Devices -- Controls how SNMP-capable devices are modeled during IBM LPAR Discovery. By default, new models are initially created as VHM models only. But, this option lets you override the default and immediately create SNMP models for devices that meet the necessary criteria.
- Retain SNMP-enabled LPARs During IBM LPAR Manager Deletion -- Controls how DX NetOps Spectrum handles SNMP-enabled LPAR models when an IBM LPAR Manager model is deleted.

Configure Maintenance Mode for New IBM LPARs

Virtual Host Manager automatically models the IBM LPAR instances in your IBM LPAR virtual environment. DX NetOps Spectrum attempts to manage all models that are discovered. However, some new IBM LPARs are not ready for DX NetOps Spectrum management when they are initially modeled. For example, a non-running IBM LPAR causes DX NetOps Spectrum to generate a Contact Lost alarm. To prevent undesired alarms on new IBM LPAR models, you can decide which new models are immediately placed into maintenance mode. Later, you can manually disable maintenance mode when you are ready for DX NetOps Spectrum to manage these devices.

Follow these steps:

1. Open Virtual Host Manager in the Navigation panel.
   A details page opens in the Contents panel for the selected Virtual Host Manager.
2. Click the Information tab.
3. Expand the Configuration, IBM LPAR, IBM LPAR Discovery subview.
4. Click Set in the 'Maintenance Mode for New IBM LPARs' field and select one of the following options:
   - **Place non-enabled LPARs in Maintenance Mode**
     (Default) Applies maintenance mode to only non-enabled IBM LPAR models upon initial IBM LPAR Discovery.
   - **Place all LPARs in Maintenance Mode**
     Applies maintenance mode to all new IBM LPAR models upon initial IBM LPAR Discovery.

   Your setting is saved and new IBM LPAR instances created by Virtual Host Manager are placed into maintenance mode per your selection.

Manage Device Models for Devices Deleted from IBM LPAR Manager

The devices and the relationships among them change frequently in virtual environments. Maintaining accurate and timely data about your virtual environment in DX NetOps Spectrum is challenging. For example, when an IBM LPAR Host or IBM LPAR instance is removed, DX NetOps Spectrum knows to remove the corresponding device models from Virtual

Host Manager in the Navigation panel. However, should DX NetOps Spectrum keep or delete the model? You can select settings to control model deletion.

> **WARNING**
> When models are deleted, all notes or other customizations on those models are lost. You can disable this option if your models are likely to be recreated in your IBM LPAR environment later.

Follow these steps:

1. Open Virtual Host Manager in the Navigation panel.
   The main details page opens in the Contents panel for the selected Virtual Host Manager.
2. Click the Information tab.
3. Expand the Configuration, IBM LPAR, IBM LPAR Discovery subview.
4. Click Set in the 'Allow Device Model Deletes During IBM LPAR Discovery' field and select one of the following options:
   - **Yes**
     (Default) Deletes the Virtual Host Manager models that correspond to entities no longer managed by your IBM LPAR environment.
   - **No**
     Places Virtual Host Manager models in the LostFound container if their corresponding entity is no longer managed by your IBM LPAR environment.

     > **NOTE**
     > Models with more associations, such as a model that is included in a Global Collection, are handled differently. These models are removed from the Universe, but they are not moved to the LostFound container.

Your setting is saved, and device models are handled accordingly after the device is deleted from your IBM LPAR environment.

Configure Model Searches Across Secure Domains (IBM LPAR)

Rather than creating new models, IBM LPAR Discovery attempts to locate models in the SpectroSERVER. In an environment with Secure Domain Manager deployed, IBM LPAR Discovery searches for models within the same secure domain as your IBM LPAR Manager. This domain is the "local" domain. However, some of your virtual environment devices can exist within a different secure domain. In this case, you can configure IBM LPAR Discovery to search all secure domains for existing models.

Follow these steps:

1. Open Virtual Host Manager in the Navigation panel.
   The main details page opens in the Contents panel for the selected Virtual Host Manager.
2. Click the Information tab.
3. Expand the Configuration, IBM LPAR, IBM LPAR Discovery subview.
4. Click Set in the 'Search for Existing Models' field and select from the following options:
   - **In IBM LPAR Manager's Secure Domain**
     (Default) Searches for existing models within the same secure domain as the IBM LPAR Manager server.
   - **In All Secure Domains**
     Searches for existing models within all secure domains managed by the SpectroSERVER. Select this option only in the following situations:
     - All devices have unique IP addresses
     - When secure domains are used for security purposes or to isolate network traffic

       > **NOTE**
       > Do not select this option for a NAT environment.

Your setting is saved and IBM LPAR Discovery searches for existing models in DX NetOps Spectrum according to your selection. If duplicate models (that is, models that share the same IP address) exist in multiple secure domains, Virtual Host Manager does the following:

- Selects the model in the local secure domain, if available.
- If a duplicate model does not exist in the local domain, Virtual Host Manager randomly selects a model from another secure domain.
- In both cases, Virtual Host Manager generates a minor alarm for the duplicate IP addresses on the IBM LPAR Manager model.

Configure SNMP Modeling Preferences (IBM LPAR)

SNMP-capable devices support enriched device monitoring, such as process and file system monitoring capabilities. However, SNMP agents can be costly and time-consuming to deploy. By default, IBM LPAR Discovery creates IBM LPAR instances as VHM models. You can later upgrade them to SNMP models. However, you can also configure IBM LPAR Discovery to model all new SNMP-capable devices as SNMP models. Although IBM LPAR Discovery can take longer to complete, initially modeling these as SNMP models avoids manually upgrading these models later.

WARNING
Enable SNMP modeling *before* you model your IBM LPAR Hosts. If you model the IBM LPAR Hosts first, all child models are created as VHM models, which must be manually upgraded to SNMP models.

Follow these steps:

1. Open Virtual Host Manager in the Navigation panel.
   The main details page opens in the Contents panel for the selected Virtual Host Manager.
2. Click the Information tab.
3. Expand the Configuration, IBM LPAR, IBM LPAR Discovery, SNMP Discovery subview.

   WARNING
   Follow the steps in the subview to prepare your devices and DX NetOps Spectrum for SNMP Discovery. If devices are not properly prepared prior to IBM LPAR Discovery, Virtual Host Manager cannot create SNMP models.
4. Click Set in the 'Discover SNMP-Capable Devices' field and select from the following options:
   - **Yes**
     Enables SNMP modeling during IBM LPAR Discovery. Only devices that meet the criteria specified in the SNMP Discovery subview text are modeled as SNMP devices. Applies to *new* models only.
   - **No**
     (Default) Models all new devices found during IBM LPAR Discovery as VHM models. You can manually upgrade these models to SNMP models later.

   Your setting is saved and new devices are modeled in Virtual Host Manager according to your selection.

Manage SNMP-Enabled LPAR Models After IBM LPAR Manager Deletion

By default, SNMP-enabled devices are deleted from DX NetOps Spectrum when the following items are deleted:

- IBM LPAR Manager model for the device
- IBM LPAR folder in the Navigation panel

SNMP-enabled device models can include significant customizations that you want to retain. You can adjust your settings to avoid deleting these models. They are placed into the LostFound container for later use.

Follow these steps:

1. Open Virtual Host Manager in the Navigation panel.
   The main details page opens in the Contents panel for the selected Virtual Host Manager.

2. Click the Information tab.
3. Expand the Configuration, IBM LPAR, IBM LPAR Discovery subview.
4. Click Set in the ‘Retain SNMP-enabled LPARs During IBM LPAR Manager Deletion' field and select one of the following options:
   – **Yes**
   Retains SNMP-enabled LPAR models in the LostFound container when their IBM LPAR Manager or the IBM LPAR folder is deleted.

   NOTE
   Models with more associations, such as a model that is included in a Global Collection, are handled differently. These models are removed from the Universe, but they are not moved to the LostFound container.

No

(Default) Deletes all LPAR models when their IBM LPAR Manager or the IBM LPAR folder is deleted.

Your setting is saved, and SNMP-enabled device models are handled accordingly when IBM LPAR Manager models or the IBM LPAR folder is deleted.

How to Discover and Model Your Virtual Environment (IBM LPAR)

To monitor your virtual environment, you must discover and model your virtual entities -- IBM LPAR Hosts and IBM LPAR instances. Modeling these entities in Virtual Host Manager lets you view your complete network topology in one tool, showing the relationships between your physical and virtual components.

The main steps for modeling your virtual environment are as follows:

1. Run a standard Discovery.
   The purpose of this Discovery is to help ensure the upstream routers and switches are modeled before IBM LPAR Discovery runs. Optionally, if the SNMP Modeling option is disabled, this step can also model the SNMP-capable IBM LPAR Managers. When modeling these entities, be sure that your modeling options are set correctly to support Virtual Host Manager.
2. Upgrade the CA eHealth SystemEDGE model.
   This step is required only when your CA eHealth SystemEDGE agent on the IBM LPAR Manager host was modeled in a release before 9.2.1.
3. Let IBM LPAR Discovery run.
   When you model the CA eHealth SystemEDGE agent with IBM LPAR AIM on the IBM LPAR Manager host, IBM LPAR Discovery begins automatically. Each of these IBM LPAR Manager models has its own IBM LPAR Discovery process. The purpose of IBM LPAR Discovery is to find the virtual entities in your IBM LPAR environment, model the ones that do not exist, and place them in the Virtual Host Manager view of the Navigation panel.

Run Discovery (IBM LPAR)

To discover your IBM LPAR environment, run the standard DX NetOps Spectrum Discovery. This Discovery ensures that the upstream routers and switches are modeled so that later connections from the virtual entities can be established. You can also model the SNMP-capable IBM LPAR Hosts and IBM LPAR instances during DX NetOps Spectrum Discovery.

NOTE
Modeling SNMP-capable IBM LPAR Hosts and IBM LPAR instances is necessary during DX NetOps Spectrum Discovery only when the SNMP Modeling option is disabled during IBM LPAR Discovery.

NOTE
Only an administrator performs this task.

Follow these steps:

1. Open the Discovery console.

   NOTE
   Before modeling, be sure that you know the correct community strings, IP addresses, and port numbers for any SNMP agents that run on a nonstandard port.
2. In the Navigation panel, click the Creates a new configuration icon  .
3. Configure your options to support virtual network modeling, as follows:
   a. Click the Modeling Options button in the Modeling Options group.
      The Modeling Configuration dialog opens.
   b. Click the Protocol Options button.
      The Protocol Options dialog opens.
   c. Select the ARP Tables for Pingables option, and click OK.
      The Modeling Configuration dialog opens.
   d. (Optional) Click the Advanced Options button in the Advanced Options group, add your nonstandard SNMP ports (such as, the CA eHealth SystemEDGE agent port), and click OK.
4. Enter individual IP addresses or the beginning and ending IP addresses in the IP Boundary List fields and click Add.

   NOTE
   Be sure that the range of IP addresses includes all servers with CA eHealth SystemEDGE and the IBM LPAR AIM installed and the interconnecting switches and routers. Or you can include the SNMP-capable IBM LPAR Hosts and IBM LPAR instances that require SNMP models.
5. Enter any additional values in the Discovery console, and click the Discover button.
   The following models are created and are added to your network topology in DX NetOps Spectrum:
   – IBM LPAR Managers and the switches and routers that connect them to your network -- Information about your virtual environment comes from the IBM LPAR Manager. When these IBM LPAR Manager models exist in DX NetOps Spectrum, IBM LPAR Discovery can begin.
   – IBM LPAR instances -- If you decide not to model these entities with DX NetOps Spectrum Discovery, IBM LPAR Discovery creates them as VHM models.

   NOTE
   You can also manually model your virtual network by IP address. In this case, we recommend modeling the upstream devices first. Modeling in the correct order ensures that the relationships among these entities are built correctly in the topology. For more information about how to perform a Discovery, see Modeling and Managing Your IT Infrastructure section.

Upgrade the CA SystemEDGE Model (IBM LPAR)

The CA eHealth SystemEDGE agent could have been modeled in DX NetOps Spectrum before installing Virtual Host Manager or before the IBM LPAR AIM was loaded on the agent. In this case, the existing CA eHealth SystemEDGE model is not compatible with Virtual Host Manager. Upgrade the model so Virtual Host Manager can access the IBM LPAR AIM capabilities in CA eHealth SystemEDGE. *This procedure is not required if the CA eHealth SystemEDGE agent with IBM LPAR AIM is loaded and is modeled after installing DX NetOps Spectrum.*

To upgrade the CA eHealth SystemEDGE model, right-click the model and select Reconfiguration, Reconfigure Model.

The CA eHealth SystemEDGE model is upgraded to support the IBM LPAR AIM.

NOTE
You can also send a reconfigure model action to CA eHealth SystemEDGE using the CLI. For more information, see Modeling and Managing Your IT Infrastructure Administrator *section*.

.

How IBM LPAR Discovery Works

IBM LPAR Discovery is a specialized discovery process that gathers detailed information about your virtual environment. The purpose of IBM LPAR Discovery is to obtain the virtual entities managed by the HMC, model the ones that do not exist in DX NetOps Spectrum, and place them under Virtual Host Manager in the Navigation panel.

A key benefit of IBM LPAR Discovery is that it runs automatically in the background, keeping your virtual environment data updated in DX NetOps Spectrum. Understanding how IBM LPAR Discovery works reinforces the importance of properly installing and modeling the various components of Virtual Host Manager.

The IBM LPAR Discovery process works as follows:

1. Immediately after the IBM LPAR Manager is configured (the CA eHealth SystemEDGE agent is installed with the IBM LPAR AIM enabled), the IBM LPAR Manager uses SSH to contact each HMC that it monitors. The IBM LPAR Manager gathers and stores information from the HMC about your virtual environment.

   NOTE
   Monitor only one instance of the IBM LPAR Host with the IBM LPAR AIM. Do not manage a single IBM LPAR Host with multiple HMCs. Monitoring more than one instance can result in duplicate models in DX NetOps Spectrum.

   WARNING
   The CA eHealth SystemEDGE agent and IBM LPAR AIM must be installed so that CA eHealth SystemEDGE, the HMCs, and DX NetOps Spectrum can communicate. If they cannot, IBM LPAR Discovery cannot run.

- During DX NetOps Spectrum Discovery, DX NetOps Spectrum creates a model for each IBM LPAR Manager in Step 1 and enables DX NetOps Spectrum to handle communication between DX NetOps Spectrum and the CA eHealth SystemEDGE agent.
- DX NetOps Spectrum polls the IBM LPAR AIM to gather the IBM LPAR Manager information that is stored in Step 1.
- DX NetOps Spectrum begins IBM LPAR Discovery and uses this information from the AIM to update modeling in the DX NetOps Spectrum Topology tab and the Virtual Host Manager hierarchy in the Navigation panel, as follows:
  a. If you enable SNMP Discovery before Step 2, Virtual Host Manager Discovery creates SNMP models for all new SNMP-capable models that meet the SNMP Discovery criteria.

     NOTE
     By default, SNMP Discovery is disabled during IBM LPAR Discovery.

  b. VHM models are created for the remaining non-SNMP IBM LPAR Hosts and IBM LPAR instances, as follows:
     - Previously existing IBM LPAR models are changed to VHM models.
     - VHM models are created for the IBM LPAR instances that *do not* previously exist in DX NetOps Spectrum.
     - VHM models are created for the IBM LPAR Host models, and these models group their associated IBM LPAR instance models in the Navigation panel, under Virtual Host Manager and the Universe topology.
  c. All models for your virtual network are added to the Virtual Host Manager portion of the Navigation panel.

     NOTE
     In a virtual environment, devices on separate IBM LPAR Hosts can have the same IP address or MAC address. In this case, DX NetOps Spectrum creates duplicate models for each occurrence of an IP address or MAC address.
- IBM LPAR Discovery automatically repeats this process at each regularly scheduled IBM LPAR technology polling interval.

NOTE
By default, the IBM LPAR polling interval is controlled by setting the polling interval on the IBM LPAR Manager model. Or you can control IBM LPAR polling independently using the IBM LPAR virtualization technology application model.

Adding SNMP Capabalities to VHM Models (IBM LPAR)

SNMP-capable devices support enriched device monitoring, such as process and file system monitoring capabilities. However, SNMP agents can be costly and time-consuming to deploy. When an SNMP agent is not available or SNMP Discovery is disabled, Virtual Host Manager creates IBM LPARs as VHM models.

Later, you can install an SNMP agent on any IBM LPAR Host or IBM LPAR and upgrade its modeling in DX NetOps Spectrum. Options for upgrading to SNMP models are as follows:

- **Upgrade only selected devices** -- This method works quickly when you have a small selection of models to upgrade. The VHM models are deleted first. One drawback of this method is that after DX NetOps Spectrum deletes the models, you must wait for the next IBM LPAR Discovery to create the SNMP models and place them in Virtual Host Manager. Knowledge of the IP addresses for the models to upgrade is required.
- **Upgrade all SNMP-capable VHM models** -- This method upgrades models in batch. It is preferred when upgrading Virtual Host Manager to a new release. Knowledge of the IP addresses of individual models is not required. Another advantage is that after DX NetOps Spectrum deletes the VHM models, the upgraded SNMP models are immediately placed in the Virtual Host Manager hierarchy without waiting for the next polling cycle. Therefore, Virtual Host Manager manages the models more quickly. The drawback to this method is that it can take a long time to complete. The time required to complete this upgrade depends on how many community strings and SNMP ports Virtual Host Manager must search when locating SNMP-capable devices.

  NOTE
  Virtual Host Manager attempts to identify SNMP agents on powered-up pingable devices only.

  WARNING
  When models are deleted, all notes or other customizations on those models are lost.

Upgrade Selected VHM Models to SNMP Models (IBM LPAR)

When an SNMP agent is not available or SNMP Discovery is disabled during IBM LPAR Discovery, Virtual Host Manager creates IBM LPAR instances as VHM models. Later, you can install an SNMP agent on these devices and upgrade their modeling in DX NetOps Spectrum. You must know the IP addresses for the device models to upgrade. Manually selecting models to upgrade works quickly, but all notes or customizations on these models are lost during the upgrade.

Follow these steps:

1. Deploy or enable an SNMP agent on the device, if required.
2. Model the device again using one of the following methods:
   - DX NetOps Spectrum Discovery
   - Model individual devices by IP address

   When the new SNMP-capable model is created, DX NetOps Spectrum removes the previous model from Virtual Host Manager and deletes it. At the next IBM LPAR AIM polling cycle, DX NetOps Spectrum adds the SNMP-capable model to Virtual Host Manager in the Navigation panel.

   WARNING
   When models are deleted, all notes or other customizations on those models are lost.

Upgrade All VHM Models to SNMP Models (IBM LPAR)

When an SNMP agent is not available or SNMP Discovery is disabled during IBM LPAR Discovery, Virtual Host Manager creates IBM LPAR instances as VHM models. Later, you can install an SNMP agent on any IBM LPAR and upgrade its

modeling in DX NetOps Spectrum. When upgrading in batch, DX NetOps Spectrum searches your VHM models and locates models that are now SNMP-capable devices. Then, DX NetOps Spectrum converts these to SNMP models. This method can take a long time to complete, depending on how many community strings and ports Virtual Host Manager must search.

Follow these steps:

1. Deploy or enable an SNMP agent on your devices, as required.
2. Open Virtual Host Manager in the Navigation panel.
   The main details page opens in the Contents panel for the selected Virtual Host Manager.
3. Select the IBM LPAR Manager model in the Navigation panel that manages the models you want to upgrade.
4. Click the Information tab.
5. Expand the IBM LPAR Manager, DX NetOps Spectrum Modeling Control subview.
6. Click the Upgrade ICMP-Only Devices button.

   WARNING
   When models are deleted, all notes or other customizations on those models are lost.

Virtual Host Manager searches for VHM models managed by the IBM LPAR AIM on the selected IBM LPAR Manager device. Virtual Host Manager upgrades the ICMP-only devices that meet the criteria for SNMP devices and places them within the Virtual Host Manager hierarchy.

Move IBM LPAR to a Different Host

Moving an IBM LPAR from one IBM LPAR Host to another can potentially result in lost data, depending on your configuration settings in Virtual Host Manager and the HMC. The IBM LPAR AIM does not support IBM LPAR migration. To Virtual Host Manager, a move is treated as two events -- an IBM LPAR is deleted in the HMC and a new IBM LPAR is created. Based on your Virtual Host Manager configuration, DX NetOps Spectrum can delete the original IBM LPAR model and create a new one. If you customized the original model, deleting it can result in lost data. You can avoid this data loss when you configure your Virtual Host Manager settings correctly before moving the IBM LPAR in the HMC.

Follow these steps:

1. Change the 'Allow Device Model Deletes During IBM LPAR Discovery' option to No.

   NOTE
   With this option disabled, DX NetOps Spectrum does not delete the IBM LPAR model from DX NetOps Spectrum, even though the model is removed from Virtual Host Manager management.
2. Using the HMC, remove the IBM LPAR from the original IBM LPAR Host.
3. Wait for Virtual Host Manager in the Navigation panel to reflect the changes.
   DX NetOps Spectrum places the IBM LPAR model into the LostFound container.

   WARNING
   For Virtual Host Manager to reconcile the new IBM LPAR with the existing model in the LostFound container, the IBM LPAR name, MAC address, *and* IP address must remain the same after migrating the IBM LPAR in the HMC. If any of these values change, Virtual Host Manager cannot use the existing model.
4. Using the HMC, add the IBM LPAR to the other IBM LPAR Host.
   When IBM LPAR Discovery finds the new IBM LPAR, Virtual Host Manager reconciles it with the existing model, removes it from the LostFound container, and places that model into Virtual Host Manager management.
5. (Optional) Change the 'Allow Device Model Deletes During IBM LPAR Discovery' option back to Yes on the originating IBM LPAR Manager model.
   The IBM LPAR is moved from one IBM LPAR Host to another.

Viewing Your IBM LPAR Virtual Environment

This section describes concepts for viewing your IBM LPAR virtual environment and the associated alarms. The basic steps are no different from the standard DX NetOps Spectrum procedures. However, this section describes conceptual differences and details that only apply to the IBM LPAR virtual technology.

Viewing Your IBM LPAR Virtual Network

On the Explorer tab, the Virtual Host Manager node displays a hierarchical tree structure that helps you visualize the logical relationships among your virtual environment resources.

Using this information, you can see how resources are shared among your IBM LPAR Managers, which can help you identify opportunities to reorganize and optimize your virtual environment. This hierarchy also provides a quick way to monitor the performance of your resources and troubleshoot their alarms.

Because Virtual Host Manager is not aware of a DSS environment, it is located within a landscape hierarchy. The following example shows where Virtual Host Manager appears on the Explorer tab in the Navigation panel and illustrates the virtual environment hierarchy:

```
[-] <ss> host
    [+] Universe
    [-] Virtual Host Manager
        [-] IBM LPAR
            [+] IBM LPAR Manager 1
            [-] IBM LPAR Manager 2
                [-] IBM LPAR Host 1
                     .  IBM LPAR 1
                     .  IBM LPAR 2
                [+] IBM LPAR Host 2
                [+] IBM LPAR Host 3
```

Virtual Host Manager is the root node for the entire virtual environment managed by this SpectroSERVER. Selecting this node in the Navigation panel displays Virtual Host Manager details in the Contents panel. You can view details such as events and alarms that are related to your virtual environment.

Directly under Virtual Host Manager, virtual environments are organized within folders that represent the associated technology. In the example hierarchy above, the IBM LPAR folder contains the portion of the virtual environment that was created using IBM LPAR virtualization technology. In this folder, Virtual Host Manager lists all IBM LPAR Manager hosts that are managed by this SpectroSERVER.

Each IBM LPAR Manager contains only the portion of the entire virtual environment that it manages. Selecting an IBM LPAR Manager in the Navigation panel displays details in the Contents panel, such as the IBM LPAR Hosts or IBM LPAR instances that are managed by the selected IBM LPAR Manager. You can also view general statistics and view details about other components that are not modeled in DX NetOps Spectrum, such as the following:

- System profiles
- Profiles
- Slots
- Virtual Ethernet devices
- Virtual SCSI devices
- Virtual serial devices
- Physical disks

Under each IBM LPAR Manager, the hierarchy represents the logical relationships between the following entities:

- **IBM LPAR Hosts**

An IBM LPAR Host contains the IBM LPAR instances that it manages. Selecting an IBM LPAR Host in the Navigation panel displays details in the Contents panel, such as events and alarms that are related to the IBM LPAR Host and CPU usage.

- **IBM LPAR instances**
  An IBM LPAR instance is always a leaf node on the Virtual Host Manager hierarchy tree. Selecting an IBM LPAR in the Navigation panel displays details in the Contents panel, including events and alarms, memory usage, and status.

Understanding the IBM LPAR Virtual Topology

The IBM LPAR Manager, IBM LPAR Host, and IBM LPAR instance models created for your virtual environment are integrated into the topology view. IBM LPAR Host models automatically group their associated IBM LPAR instances. The topology shows how these IBM LPARs are connected to your physical network entities.

The following example shows how these models can appear on the Explorer tab in the Navigation panel under the Universe group:

```
[-] Universe
     .  Physical switch 1
     .  Physical switch 2
     .  IBM LPAR Manager
    [-] IBM LPAR Host
         .  Fanout A
         .  Fanout B
         .  IBM LPAR A
         .  IBM LPAR B
         .  IBM LPAR C
```

Selecting one of these models displays these relationships graphically on the Topology tab in the Contents panel.

How the IBM LPAR Data is Updated in Virtual Host Manager

During your initial IBM LPAR Discovery, DX NetOps Spectrum populates Virtual Host Manager hierarchy in the Navigation panel with your virtual device models. After DX NetOps Spectrum builds this initial hierarchy, your virtual network configuration can change, and Virtual Host Manager must continually work to keep this information accurate in DX NetOps Spectrum. For example, the following events can change your virtual network configuration:

- Creating or deleting an IBM LPAR on an IBM LPAR Host
- Moving an IBM LPAR from one IBM LPAR Host to another

To keep your information accurate, Virtual Host Manager detects these changes by polling the IBM LPAR AIM. Therefore, your virtual network configuration is updated in DX NetOps Spectrum at each polling cycle. DX NetOps Spectrum also receives traps from the AIM and generates the corresponding events. By reviewing the event log, you can find out when configuration changes occur, such as when a new IBM LPAR is created.

When an IBM LPAR is deleted, DX NetOps Spectrum removes the models from the Virtual Host Manager hierarchy in the Navigation panel. When the AIM detects an addition to your virtual network configuration, such as provisioning a new IBM LPAR or placing one into management, DX NetOps Spectrum performs the following tasks:

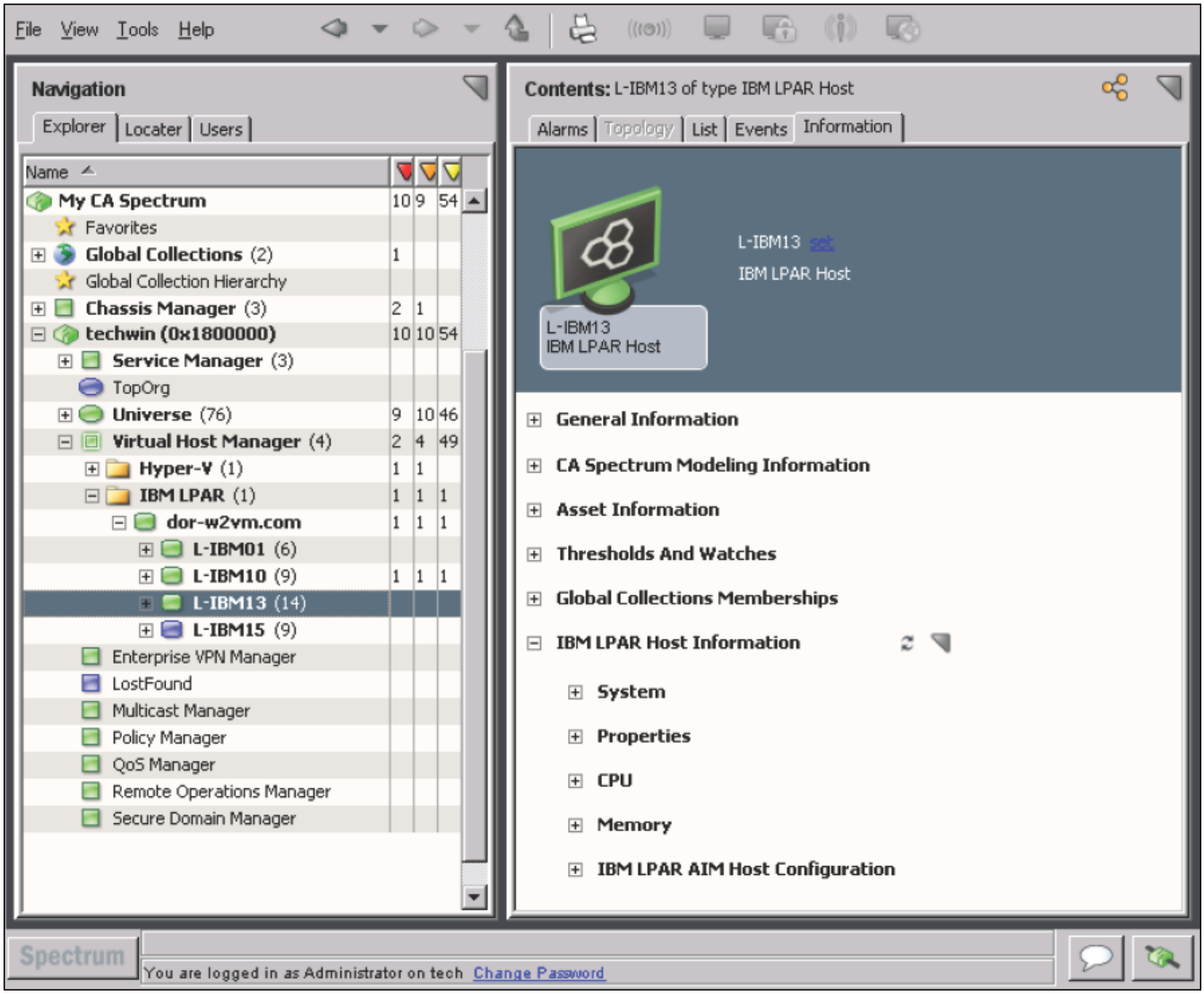
- Updates the placement of your virtual device models in the Virtual Host Manager hierarchy of the Navigation panel
- *Automatically* rediscovers connections to the affected IBM LPAR models and associates them with the correct IBM LPAR Host in the Universe topology.

  WARNING
  To reestablish connections to your virtual models correctly, all interconnecting routers and switches in your physical network must be modeled. If these models do not exist before connections to your virtual devices are rediscovered, DX NetOps Spectrum cannot resolve those connections and display the information

correctly in the Universe topology view. The IBM LPAR Hosts are placed in the same LAN container as the CA eHealth SystemEDGE model.

Custom Subviews for Virtual Entity Types (IBM LPAR)

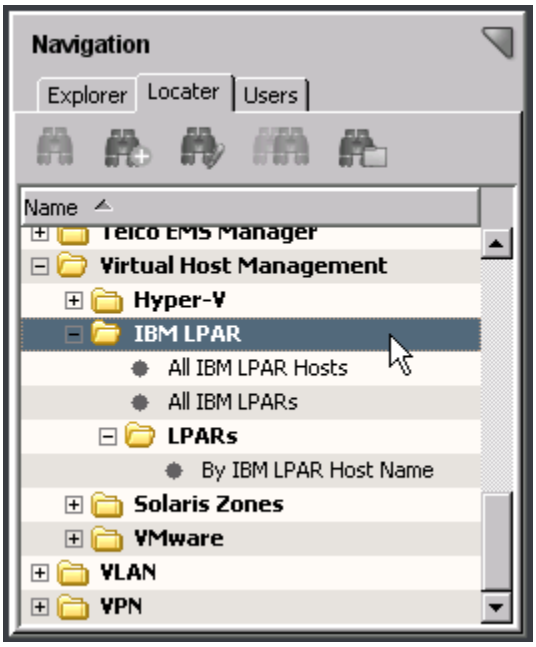
Your Virtual Host Manager models collectively provide information about your virtual environment. Individually, each model provides unique information or configuration settings, depending on the virtual entity type it represents. This custom subview appears on the Information tab in the Contents panel. These subviews can contain real-time data, such as CPU status or memory utilization, and provide access to threshold settings. For example, the custom subview for IBM LPAR Host is the "IBM LPAR Host Information" subview, as shown:



NOTE
The IBM LPAR Manager model provides combined information for all virtual devices managed by the IBM LPAR Manager. That is, selecting the IBM LPAR Manager model in the Navigation panel displays information about the selected IBM LPAR Manager host *and* combined information about all IBM LPAR Hosts, IBM LPAR instances, system profiles, virtual Ethernet devices, and more. This information contains the same data displayed on the Information tab for each individual entity model. The combined view in the IBM LPAR Manager model can provide a good overview about all of the virtual entities it manages.

Locater Tab for IBM LPAR Searches

In addition to viewing details about your virtual environment on the Explorer tab, you can also use the Locater tab to run preconfigured Virtual Host Manager searches. The search options are grouped under the Virtual Host Management, IBM LPAR folder on the Locater tab, as shown:



These detailed searches can help you investigate information related to virtual entities only, such as locating all IBM LPAR instances within a landscape.

> **NOTE**
> Although Virtual Host Manager is not DSS aware, these preconfigured searches let you select multiple landscapes to search in the search parameters.

The Locater tab in the Navigation panel includes the following searches for Virtual Host Manager information:

- **All IBM LPAR Hosts**
  Locates all IBM LPAR Hosts that have been modeled in the DX NetOps Spectrum database for your virtual network.
- **All IBM LPARs**
  Locates all IBM LPAR instances that have been modeled in the DX NetOps Spectrum database for your virtual network.
- **LPARs, By IBM LPAR Host Name**
  Locates all IBM LPAR instances that have been modeled in the DX NetOps Spectrum database for your virtual network, limited to only those IBM LPARs managed by a selected IBM LPAR Host.

Status Monitoring Options (IBM LPAR)

DX NetOps Spectrum provides a wide range of options for monitoring the state of your virtual network resources. The status information available for a resource varies, depending on the type of virtual entity you are monitoring. Also, your ability to configure a status option depends on its type. For example, some status options are read-only, but others let you configure threshold values, enable behaviors, or select an alarm severity. Providing this range of options and levels of customization, DX NetOps Spectrum lets you decide how to best monitor the performance of your virtual network.

Status fields are located in the OneClick subviews. All status information for a given virtual environment is available on the IBM LPAR Manager model in a tabular format. Also, each virtual entity type that has a unique model in DX NetOps Spectrum provides a subset of the same status information for easy viewing. Status-related settings, including the alert type, monitor, and thresholds, can be set from either view location.

The following tables outline the type of status information available for each virtual entity type. The Subview Locations column describes where the corresponding status fields are located in OneClick. For example, "memory" information for your IBM LPAR models is available on the Information tab in the following two locations:

- IBM LPAR Information, Memory subview for the IBM LPAR model
- IBM LPAR Manager, Managed Environment, LPARs subview for the IBM LPAR Manager model

To explore the exact status options available for each status information type, locate the subview in OneClick.

IBM LPAR Host

Status Information Type	Subview Locations
Overall	IBM LPAR Host, IBM LPAR Manager
CPU	IBM LPAR Host, IBM LPAR Manager

IBM LPAR

Status Information Type	Subview Locations
System	IBM LPAR, IBM LPAR Manager
CPU	IBM LPAR, IBM LPAR Manager
Memory	IBM LPAR, IBM LPAR Manager

How to Configure Management Options (IBM LPAR)

After your virtual network is modeled, you can configure Virtual Host Manager options for viewing and managing your device models. Configuring your preferences helps ensure that Virtual Host Manager handles your virtual device models correctly and monitors only the information that is important to you.

To configure your installation of Virtual Host Manager, perform the following procedures after you discover and model your virtual network:

- Configure the IBM LPAR AIM options -- These options let you select settings for the CA eHealth SystemEDGE IBM LPAR AIM, such as the AIM polling interval and various traps.
- Configure threshold values and other status monitoring options -- These options let you determine which information you want to monitor and how DX NetOps Spectrum manages the various events that occur in your virtual environment.

Configure the IBM LPAR AIM

The IBM LPAR AIM communicates with the IBM LPAR Manager to manage and collect information about your virtual environment. In Virtual Host Manager, you can configure the AIM to determine how it handles polling, traps, and events. The IBM LPAR AIM configuration settings let you determine the right balance of information to gather against the amount of required resources.

Follow these steps:

1. Open Virtual Host Manager in the Navigation panel.
   The main details page opens in the Contents panel for the selected Virtual Host Manager.
2. Locate and click your IBM LPAR Manager on the Explorer tab in the Navigation panel.
   The tabs in the Contents panel are populated with details about your IBM LPAR Manager.

3. Click the Information tab.
4. Expand the IBM LPAR Manager, IBM LPAR AIM, Configuration subview.
5. Click set to change the settings for the following fields, as needed:
   - **Polling Interval (Seconds)**
     Specifies the time interval (in seconds) when the IBM LPAR AIM polls and caches status and modeling information from the configured IBM LPAR Hosts. This polling retrieves status and modeling updates, such as an IBM LPAR not-running status, IBM LPAR Host disconnected, new IBM LPAR available, new IBM LPAR Host, and more.
     Default: 300
     Limits: Values greater than or equal to 300.
     Note: For best results, we recommend setting this interval lower than the DX NetOps Spectrum poll cycle interval.
   - **Log Level**
     Specifies the level of information written to the IBM LPAR AIM log file. The levels are cumulative (for example, log level 4 writes all messages at levels 0 through 4). The log levels are as follows:
     - 0: Fatal
     - 1: Critical
     - 2: Warning
     - 3: Info
     - 4: Debug
     - 5: Debug Low
     - 6: Debug Lower
     - 7: Debug Lowest

     NOTE
     Default: 2 Specifying a debug level greater than 4 is discouraged.
   - **Events Max**
     Specifies the maximum number of events to store in the Events table. When the maximum rows are reached, DX NetOps Spectrum begins overwriting event rows, beginning with the oldest recorded events.
     Default: 500
     Limits: 1 - 2147483647
   - **History (days)**
     Specifies the amount of history information that is available in the Events table, in days. Events older than the specified number of days are purged from the Events table.
     Note: The value in the Events Max field also affects this setting. When the max is reached, the Events table cannot always store events that span the number of days that you specify in the History (days) field. For example, 800 events occur in the past 30 days. The most recent 500 events occurred within the past 10 days. If the Events Max field specifies 500, only 10 days of history are available in the Events table.
     Default: 30
     Limits: 1 - 365
   - **Clear Events**
     Determines whether to clear events from the Events table. Select from the following options:
     - **do-not-clear**
       (Default) Retains all events in the Event table until the Events Max or History (days) values are reached.
     - **clear**
       Clears all events from the Event table when you start the IBM LPAR AIM.

   Your IBM LPAR AIM is configured with your selections.

Configure and Monitor Resource Status (IBM LPAR)

Yu can monitor the status of virtual resources in OneClick. For example, you can view the total memory, used memory, percent of CPU usage, and more. Also, you can set monitoring options, such as enabling alerts and setting threshold values for traps. This information can help you optimize your virtual network performance and troubleshoot alarms.

NOTE
Traps are set on and managed by the IBM LPAR AIM, but you can configure these threshold values from the OneClick subviews. A read/write community string is required to change any threshold values or settings.

You can view or configure resource status options and information for virtual devices on the Information tab.

Follow these steps:

1. Open Virtual Host Manager in the Navigation panel.
   The main details page opens in the Contents panel for the selected Virtual Host Manager.
2. Locate and click the virtual device on the Explorer tab in the Navigation panel.
   The device details display in the Contents panel.
3. Click the Information tab.
   Multiple subviews are available for viewing. Generally, the subview at the bottom of the tab includes the resource allocation and utilization information for the selected model. For example, an IBM LPAR Host model displays a subview named "IBM LPAR Host Information" that includes details for the specific model you selected in the Navigation panel.
4. Expand the appropriate subview.
   All available resource status details and monitoring options for the selected device model are displayed.

   NOTE
   The IBM LPAR Manager model provides combined information for all virtual devices managed by the IBM LPAR Manager. That is, selecting the IBM LPAR Manager model in the Navigation panel displays information about the selected IBM LPAR Manager host *and* combined information about all IBM LPAR Hosts, IBM LPAR instances, system profiles, virtual Ethernet devices, and more. This information contains the same data displayed on the Information tab for each individual entity model. The combined view in the IBM LPAR Manager model can provide a good overview about all of the virtual entities it manages.

Controlling IBM LPAR AIM Polling

When tuning Virtual Host Manager performance, you can change the IBM LPAR Manager polling rate or disable IBM LPAR technology polling. By default, the polling attributes on the IBM LPAR Manager model control the IBM LPAR-related polling behavior. Or you can change this IBM LPAR-related polling behavior independently. The IBM LPAR virtual technology application model, IBMLPARAIMApp, controls your IBM LPAR-related polling.

The following two attribute values on the application specifically control the IBM LPAR technology polling logic:

- PollingStatus
- Polling_Interval

Both the IBM LPAR Manager device model and the IBMLPARAIMApp application model contain these attributes. PollingStatus disables and enables polling, while Polling_Interval controls the polling frequency. If the values are different, the IBMLPARAIMApp application model attribute values take precedence.

This ability to set the value for the device model and application model lets you fine-tune your IBM LPAR-related polling. For both PollingStatus and Polling_Interval, modifying the attribute on the IBM LPAR Manager device model also changes the corresponding application model attribute if their values are the same.

Configure the IBM LPAR Polling Interval

You can change the IBM LPAR AIM polling rate. Configure the polling interval by setting the Polling_Interval attribute on the IBM LPAR virtual technology application model.

Follow these steps:

1. Open OneClick and click the Locater tab in the Navigation panel.
2. Expand the Application Models folder and double-click 'By Device IP Address.'

A search dialog opens.
3. Enter the IP address for your IBM LPAR Manager in the Device IP Address field and click OK.
   A list of application models for the IBM LPAR Manager appears in the Contents panel.
4. Select the IBMLPARAIMApp application model.
   The application model details appear in the Component Details panel.
5. Click the Information tab in the Component Details panel.
6. Click the Modeling Information subview.
7. Click set in the Poll Interval (sec) field, enter a new value.

   NOTE

   Changing the Poll Interval (sec) value from any number to 0 also sets the Polling field to Off, disabling IBM LPAR AIM polling. However, if you set the Poll Interval (sec) to 0 and set the Polling field to On, IBM LPAR AIM polling continues, using the polling interval set for the IBM LPAR Manager device.

   The IBM LPAR AIM polling interval setting is configured.

Disable IBM LPAR Polling

You can disable IBM LPAR AIM polling. Disabling IBM LPAR polling is the same as disabling Virtual Host Manager. Disable polling by setting the PollingStatus attribute on the IBM LPAR virtual technology application model.

Follow these steps:

1. Open OneClick and click the Locater tab in the Navigation panel.
2. Expand the Application Models folder and double-click 'By Device IP Address.'
   A search dialog opens.
3. Enter the IP address for your IBM LPAR Manager in the Device IP Address field and click OK.
   A list of application models for the IBM LPAR Manager appears in the Contents panel.
4. Select the IBMLPARAIMApp application model.
   The application model details appear in the Component Details panel.
5. Click the Information tab in the Component Details panel.
6. Click the DX NetOps Spectrum Modeling Information subview.
7. Click set in the Polling field and select Off.
   Polling is disabled for the IBM LPAR AIM on the selected IBM LPAR Manager.

Deleting Virtual Host Manager Models (IBM LPAR)

Models can be deleted from OneClick at any time for various reasons. However, Virtual Host Manager restricts your ability to delete models from the Virtual Host Manager hierarchy in the Navigation panel. To delete models manually, you have the following two options:

- Delete the IBM LPAR folder or an IBM LPAR Manager model in Virtual Host Manager
- Remove a virtual entity using the HMC

In Virtual Host Manager, models are sometimes deleted automatically. The following circumstances cause DX NetOps Spectrum to automatically delete Virtual Host Manager models:

- **IBM LPAR folder deleted or IBM LPAR Manager model removed from Virtual Host Manager**
  If you remove an IBM LPAR Manager model or delete the IBM LPAR folder from the Navigation panel, DX NetOps Spectrum deletes all related child models.
- **An entity removed from IBM LPAR virtual environment**

As you delete IBM LPAR Hosts and IBM LPAR instances using the HMC, DX NetOps Spectrum also deletes those models and their child models from Virtual Host Manager.

- **Upgraded models exist** -- In some cases, an IBM LPAR instance is first modeled for Virtual Host Manager without SNMP capabilities. If SNMP capabilities are later added to a VHM model, the previous model is deleted and replaced with the new SNMP-capable model.

  NOTE
  Although the default setting is to delete the models, you can configure Virtual Host Manager to place the IBM LPAR Host and IBM LPAR instances in the LostFound container when they are removed from Virtual Host Manager. This configuration setting applies only when you remove devices using your HMC. However, this setting does not apply when you delete the IBM LPAR folder, remove an IBM LPAR Manager model, or upgrade a VHM model.

Alarms and Fault Isolation for IBM LPAR

This section describes the traps used by Virtual Host Manager and the resulting alarms. This section also explains how Virtual Host Manager fault isolation differs from basic DX NetOps Spectrum fault isolation.

Virtual Host Manager Alarms for IBM LPAR

To alert you to problems within your virtual network, DX NetOps Spectrum generates alarms. Alarms are created in two ways:

- Traps sent from the CA eHealth SystemEDGE agent
- Polling

Polling generates four alarms: IBM LPAR Proxy Lost, IBM LPAR Host Proxy Lost, IBM LPAR Manager Unavailable, and IBM LPAR Not Running. However, several traps can generate alarms on your virtual devices. DX NetOps Spectrum supports all traps sent by the IBM LPAR AIM from the CA eHealth SystemEDGE agent. To get the greatest value from these traps when monitoring your devices, you can configure the threshold values for each virtual device individually.

If a trap breaches your threshold value and generates an alarm, DX NetOps Spectrum uses the value of the “state” varbind passed with the trap to determine the alarm severity. All state varbinds have the following possible values, which DX NetOps Spectrum alarms on the same way:

- 0: Unknown
- 1: OK
- 2: Warning
- 3: Critical

The "Unknown" state does not have an associated alarm severity and does not change the alarm severity of a device. DX NetOps Spectrum maps the other IBM LPAR technology states to a DX NetOps Spectrum alarm severity:

IBM LPAR State	DX NetOps Spectrum Alarm Severity
1: OK	Normal (Green)
2: Warning	Minor (Yellow)
3: Critical	Major (Orange)

Forwarding Traps from CA SystemEDGE (IBM LPAR)

DX NetOps Spectrum supports all traps sent by the IBM LPAR AIM. These traps are initially sent to the IBM LPAR CA eHealth SystemEDGE model. If the destination for a trap is not this model, DX NetOps Spectrum forwards the trap to the correct virtual model.

NOTE
For specific event codes related to the traps, use the Event Configuration application and filter on “0x056e.” Or you can launch MIB tools to view the traps in the Trap Support table for the “EMPIRE-CALPARA-MIB” MIB.

DX NetOps Spectrum determines where to forward the trap by using the following process:

1. When DX NetOps Spectrum receives a trap, it uses varbind information in the trap to locate the correct virtual entity, as follows:
   – For traps that are forwarded to an IBM LPAR Host, DX NetOps Spectrum uses the UID to locate the correct host.
   – For traps that are forwarded to an IBM LPAR instance, DX NetOps Spectrum uses the UID to determine first the correct IBM LPAR Host. Based on the UID or the IBM LPAR name, DX NetOps Spectrum locates the correct IBM LPAR instance within the list of IBM LPARs managed by this IBM LPAR Host.
2. DX NetOps Spectrum uses this UID to look up and locate the DX NetOps Spectrum model that is tied to a given UID. The entity type of all traps is predetermined. Depending on the results of the look-up, DX NetOps Spectrum forwards the trap as follows:
   – If it finds a DX NetOps Spectrum model of a specific type with a given UID and in some cases an IBM LPAR name, DX NetOps Spectrum forwards the event and corresponding alarm to the destination model.
   – If it cannot find a DX NetOps Spectrum model for a given UID and in some cases an IBM LPAR name, DX NetOps Spectrum generates a new generic event on the IBM LPAR Manager model. This new event includes details about the trap.

NOTE
DX NetOps Spectrum often cannot find a related model when a trap is sent immediately after changing your virtual network entities in the HMC. IBM LPAR Discovery has not yet identified and created the corresponding model in DX NetOps Spectrum.

Traps Supported in Virtual Host Manager (IBM LPAR)

All traps that are generated by the IBM LPAR AIM are supported in DX NetOps Spectrum. The traps are initially sent to the IBM LPAR Manager model. Then, the traps are forwarded to a corresponding virtual entity type (that is, the "destination" entity), depending on the type of trap. Using these traps, you can monitor the performance of your virtual network, resolve any resulting alarms, or trigger events.

NOTE
For more information about traps that are generated by the IBM LPAR AIM, see *CA Virtual Assurance for Infrastructure Managers Implementation*.

The following tables list the traps for a specific destination entity type and specify whether the trap generates an alarm.

Trap Name	Trap OID	Alarm?
lparAimSysAdded	1.3.6.1.4.1.546.1.1.0.165317	No
lparAimSysRemove	1.3.6.1.4.1.546.1.1.0.165316	No

Trap Name	Trap OID	Alarm?
lparAimLPAdded	1.3.6.1.4.1.546.1.1.0.165321	No
lparAimLPDeleted	1.3.6.1.4.1.546.1.1.0.165322	No
lparAimSlotAdd	1.3.6.1.4.1.546.1.1.0.165340	No
lparAimSlotDelete	1.3.6.1.4.1.546.1.1.0.165341	No
lparAimSlotLPChange	1.3.6.1.4.1.546.1.1.0.165342	No
lparAimSlotMonitorChange	1.3.6.1.4.1.546.1.1.0.165343	No

lparAimSysCfgAlertChange	1.3.6.1.4.1.546.1.1.0.165312	No
lparAimSysCfgMonitorChange	1.3.6.1.4.1.546.1.1.0.165311	No
lparAimSysCPUThresholdChange	1.3.6.1.4.1.546.1.1.0.165313	No
lparAimSysDown	1.3.6.1.4.1.546.1.1.0.165315	No
lparAimSysMEMThresholdChange	1.3.6.1.4.1.546.1.1.0.165314	No
lparAimSysProfAdd	1.3.6.1.4.1.546.1.1.0.165360	No
lparAimSysProfChange	1.3.6.1.4.1.546.1.1.0.165362	No
lparAimSysProfDelete	1.3.6.1.4.1.546.1.1.0.165361	No
lparAimSysStateChangeTrap	1.3.6.1.4.1.546.1.1.0.165310	Yes
lparAimSysCpuStateChange	1.3.6.1.4.1.546.1.1.0.165318	Yes

Trap Name	Trap OID	Alarm?
lparAimLPAlert	1.3.6.1.4.1.546.1.1.0.165324	No
lparAimLPCPUCritThreshold	1.3.6.1.4.1.546.1.1.0.165329	No
lparAimLPCPULagSetting	1.3.6.1.4.1.546.1.1.0.165327	No
lparAimLPCPUMonitor	1.3.6.1.4.1.546.1.1.0.165325	No
lparAimLPCPUState	1.3.6.1.4.1.546.1.1.0.165333	Yes
lparAimLPCPUWarnThreshold	1.3.6.1.4.1.546.1.1.0.165328	No
lparAimLPMemoryCritThreshold	1.3.6.1.4.1.546.1.1.0.165331	No
lparAimLPMemoryMonitor	1.3.6.1.4.1.546.1.1.0.165326	No
lparAimLPMemoryState	1.3.6.1.4.1.546.1.1.0.165332	Yes
lparAimLPMemoryWarnThreshold	1.3.6.1.4.1.546.1.1.0.165330	No
lparAimLPMonitor	1.3.6.1.4.1.546.1.1.0.165323	No
lparAimLPStateChange	1.3.6.1.4.1.546.1.1.0.165320	Yes
lparAimProfAdd	1.3.6.1.4.1.546.1.1.0.165350	No
lparAimProfDelete	1.3.6.1.4.1.546.1.1.0.165351	No
lparAimVIOvEthernetAdd	1.3.6.1.4.1.546.1.1.0.165373	No
lparAimVIOvEthernetRemoved	1.3.6.1.4.1.546.1.1.0.165374	No
lparAimVIOvSCSIAdd	1.3.6.1.4.1.546.1.1.0.165370	No
lparAimVIOvSCSIRemoved	1.3.6.1.4.1.546.1.1.0.165371	No
lparAimVIOvSerialAdd	1.3.6.1.4.1.546.1.1.0.165375	No
lparAimVIOvSerialRemoved	1.3.6.1.4.1.546.1.1.0.165376	No

Fault Management for Virtual Networks (IBM LPAR)

The goal of fault isolation is to narrow down the root cause of a networking problem. Finding the root cause can help you to troubleshoot and quickly correct the problem or to correct the problem programmatically with automated scripts. Deciding which devices are the root cause of an alarm can be difficult, because problems with a single device can cause several devices in your network to generate events.

For example, losing contact with an IBM LPAR Host often means that you have also lost contact with the IBM LPAR instances it manages. Therefore, the IBM LPAR Host device model and all affected IBM LPAR instances generate alarms. Using fault isolation techniques, Virtual Host Manager correlates these alarms in an attempt to identify a single root cause.

Virtual networks provide a unique management opportunity, because they provide DX NetOps Spectrum an alternate management perspective. That is, DX NetOps Spectrum can gather information through direct contact with your virtual devices or through the virtual network management technology, IBM LPAR. This alternate management perspective enhances standard DX NetOps Spectrum fault management in two ways:

- **Enhanced Contact Lost alarms** -- Two sources of information about a device means Virtual Host Manager can pinpoint the cause and more easily correlate events to a single root cause.
- **Proxy Failure alarms** -- *Proxy management* is the act of managing network devices using an alternate management source in place of or in addition to the primary manager. For example, DX NetOps Spectrum can manage virtual network devices by contacting them directly or through the virtual technology application's contact with the devices. When IBM LPAR virtualization technology loses contact with a virtual network device, Virtual Host Manager generates one of the Proxy Management Lost alarms for each device. These alarms are unique, because they are alerting you to the fact that *management* of the device through the *proxy* is affected, not the state of the device or direct (SNMP) management.

How Fault Isolation Works when Device Contact is Lost (IBM LPAR)

To help you troubleshoot networking problems with your devices, DX NetOps Spectrum uses fault isolation to narrow down the root cause of an alarm. For virtual networks, Virtual Host Manager uses information from direct contact with the device plus information provided by IBM LPAR technology through the IBM LPAR AIM. In many cases, standard DX NetOps Spectrum fault management can pinpoint the root cause. But in special circumstances, the method for isolating problems in a virtual network go beyond the standard methods.

The type of fault isolation Virtual Host Manager uses to discover the root cause depends on which devices are alarming and the type of events the devices generate. The following scenarios describe two unique fault management situations and how DX NetOps Spectrum isolates the networking error in your virtual network.

Scenario 1: IBM LPAR instance is not running

In a virtual environment, the virtual management application can provide more details than DX NetOps Spectrum can discover through standard device monitoring. For example, the IBM LPAR virtualization technology is aware when an IBM LPAR changes from the "running" state to something else, such as the "open-firmware" state.

If an IBM LPAR is no longer running and DX NetOps Spectrum loses contact with it, but proxy management of the IBM LPAR Manager is uninterrupted, DX NetOps Spectrum determines the root cause as follows:

1. When DX NetOps Spectrum loses contact with an IBM LPAR, it generates a Contact Lost alarm.
2. During its next polling cycle, the IBM LPAR Manager model polls the IBM LPAR AIM to gather information about the IBM LPAR instance. Because IBM LPAR technology manages the IBM LPAR instances, it can provide a unique view into the possible cause of alarms generated by an IBM LPAR.
3. If the IBM LPAR technology finds that the IBM LPAR is in the not-running mode, it generates an IBM LPAR Not Running alarm.

   NOTE
   This alarm is cleared upon the first IBM LPAR AIM polling cycle after the IBM LPAR is running again.
4. Virtual Host Manager correlates the Contact Lost alarm to the corresponding IBM LPAR Not Running alarm created by DX NetOps Spectrum. Virtual Host Manager makes the Contact Lost alarm appear as a symptom of the IBM LPAR Not Running alarm.

Scenario 2: IBM LPAR Host is down

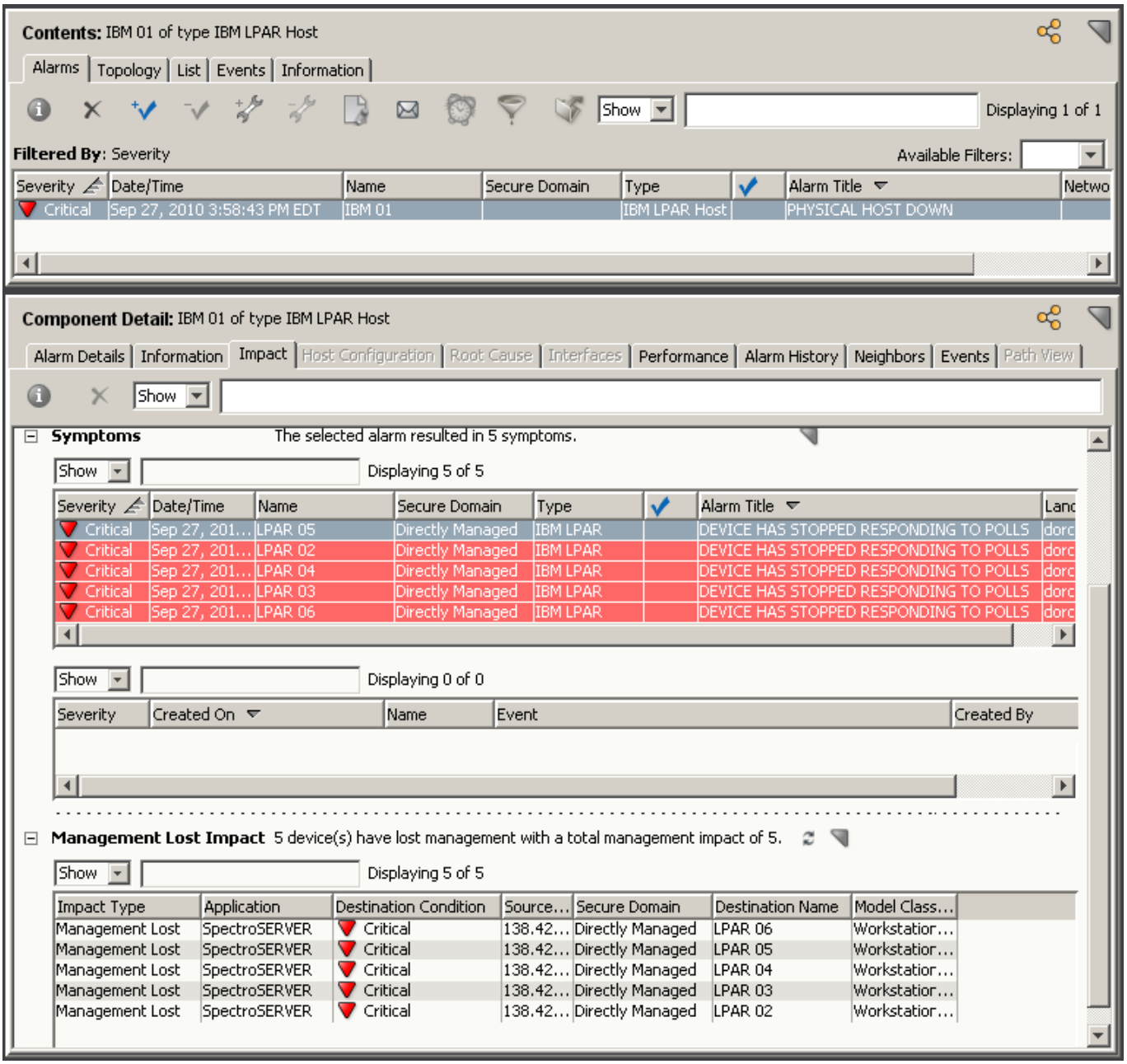
If DX NetOps Spectrum loses contact with all IBM LPARs running on an IBM LPAR Host, DX NetOps Spectrum checks the status of the upstream routers and switches. Depending on their status, DX NetOps Spectrum determines the root cause as follows:

- All upstream devices for one or more IBM LPAR instances are unavailable -- Standard DX NetOps Spectrum fault isolation techniques determine the root cause, as follows:

  - Device Stopped Responding to Polls alarm -- Generated on the IBM LPAR Host when at least ne upstream connected device for any IBM LPAR is up.
  - Gateway Unreachable alarm -- Generated on the IBM LPAR Host when *all* upstream connected devices are down.
- At least one upstream device is available for every IBM LPAR instance connected to the IBM LPAR Host -- DX NetOps Spectrum infers that the IBM LPAR Host is the root cause and responds as follows:
  a. All IBM LPARs, ports, and fanouts that are directly connected to the IBM LPAR models generate the standard fault isolation alarms.
  b. Virtual Host Manager creates a Physical Host Down alarm for the IBM LPAR Host model.
  c. All fault isolation-related alarms that are created for the impacted devices (such as IBM LPARs, ports, and fanouts) are correlated to the Physical Host Down alarm, making them symptoms of the Physical Host Down alarm. These symptom alarms appear in the Symptoms table on the Impact tab for the Physical Host Down alarm.

     NOTE
     For each IBM LPAR Host model, Virtual Host Manager creates a "virtual fault domain." This domain includes the IBM LPAR Host and IBM LPAR instances, plus all ports and fanouts directly connected to the IBM LPARs. When the IBM LPAR Host generates the Physical Host Down alarm, all standard fault isolation alarms within the domain are correlated to it. Correlating these alarms as symptoms indicates that the Physical Host Down alarm on the IBM LPAR Host is the root cause.
  d. All impacted devices are listed in the Management Lost Impact table on the Impact tab for the Physical Host Down alarm.

     NOTE
     Devices that are suppressed do not have a corresponding alarm in the Symptoms table.



If all upstream devices for one or more IBM LPAR instances go down, DX NetOps Spectrum can no longer reliably state that the fault lies with the IBM LPAR Host. Therefore, DX NetOps Spectrum clears the Physical Host Down alarm and applies the standard DX NetOps Spectrum fault isolation techniques.

How Fault Isolation Works when Proxy Management is Lost (IBM LPAR)

The IBM LPAR virtualization technology used to create your virtual network provides DX NetOps Spectrum a unique management opportunity. DX NetOps Spectrum can use the standard methods to contact your virtual devices directly, plus DX NetOps Spectrum can simultaneously gather virtual device information from IBM LPAR technology. In this sense, the IBM LPAR technology is a "proxy" from which DX NetOps Spectrum gathers virtual device information. If DX NetOps Spectrum loses direct contact with a device, it generates alarms. Likewise, if IBM LPAR technology loses contact with

a virtual device or if Virtual Host Manager loses contact with the IBM LPAR Manager, Virtual Host Manager generates alarms -- Proxy Management Lost alarms.

In response, DX NetOps Spectrum attempts to isolate the cause of the proxy management failure. Proxy fault isolation is similar to the standard DX NetOps Spectrum fault isolation, except that these alarms alert you to the fact that *proxy* management of a virtual device is affected. Proxy management fault isolation cannot tell you whether a virtual device is up or down. However, it is important to know when contact through the proxy is lost, because you could be missing important virtual information about a device.

The type of proxy fault isolation Virtual Host Manager uses to discover the root cause depends on which devices are alarming and the type of events the devices generate. The following scenarios describe two unique proxy fault management situations and how Virtual Host Manager isolates the networking error in your virtual network.

Scenario 1: Contact between IBM LPAR Manager and HMC is lost

If the IBM LPAR Manager loses contact with an HMC and all IBM LPAR Hosts and IBM LPARs the HMC is managing, the IBM LPAR Manager data about the IBM LPAR Hosts and all hosted IBM LPAR instances is lost. To isolate the problem, Virtual Host Manager determines the root cause as follows:

1. A Proxy Lost alarm is generated on the IBM LPAR Hosts and all hosted IBM LPARs.
2. The IBM LPAR alarms are correlated to the Proxy Lost alarm for the IBM LPAR Host, making these IBM LPAR alarms symptoms of the IBM LPAR Host alarm. Correlating these alarms as symptoms indicates that the IBM LPAR Host alarm is the root cause.
3. If DX NetOps Spectrum also loses contact with the IBM LPAR Host and generates a Physical Host Down alarm, the Proxy Lost alarm generated for the IBM LPAR Host is correlated to the Physical Host Down alarm. In this case, the Proxy Lost alarm becomes a symptom of the Physical Host Down alarm. Correlating this alarm as a symptom indicates that the Physical Host Down alarm on the IBM LPAR Host is the root cause.

Scenario 2: Contact between DX NetOps Spectrum and IBM LPAR Manager is lost

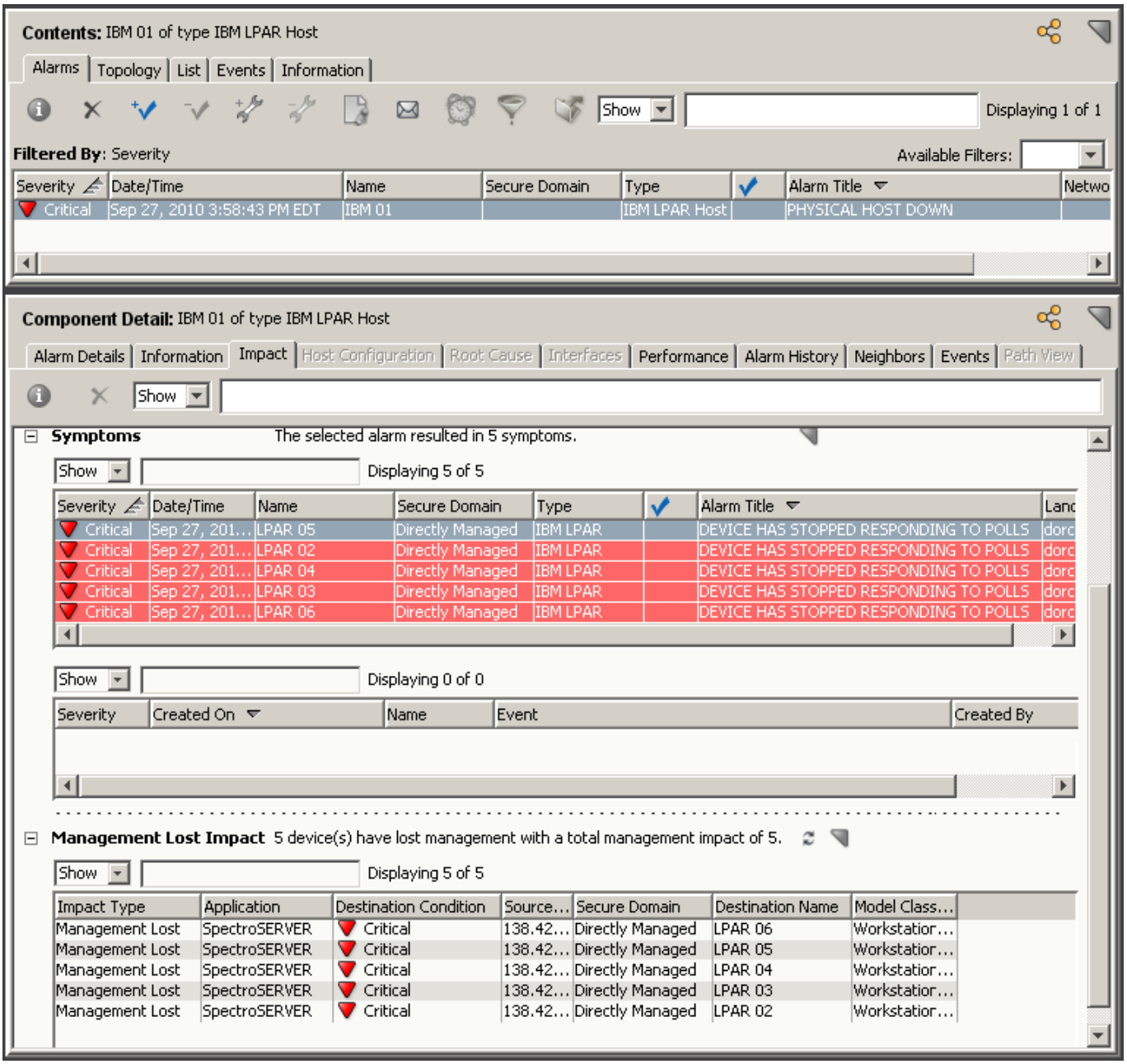
If DX NetOps Spectrum loses contact with or stops polling the IBM LPAR Manager model, DX NetOps Spectrum loses the IBM LPAR technology data about all virtual models managed by that IBM LPAR Manager. To isolate the problem, Virtual Host Manager determines the root cause as follows:

1. DX NetOps Spectrum generates Proxy Lost alarms for all virtual models managed by that IBM LPAR Manager, including IBM LPAR instances and IBM LPAR Hosts. DX NetOps Spectrum also generates a separate Proxy Unavailable alarm on the IBM LPAR Manager model.
2. The IBM LPAR alarms are correlated to their corresponding IBM LPAR Host model alarm.
3. The IBM LPAR Host model alarms are correlated to a Proxy Unavailable alarm for the IBM LPAR Manager model.
4. This Proxy Unavailable alarm is then correlated to the root cause of the IBM LPAR Manager being down. The root cause is typically an alarm generated by standard DX NetOps Spectrum fault management, such as the alarms created for the following situations:
   - Lost management of IBM LPAR Manager (that is, a problem occurred with the CA eHealth SystemEDGE agent on the IBM LPAR Manager host)
   - Machine contact is lost
   - IBM LPAR Manager model is in maintenance mode

Determining IBM LPARs Affected by Host Outages

When contact with an IBM LPAR Host is interrupted or the IBM LPAR Host goes down, all IBM LPAR instances hosted by the IBM LPAR Host are affected. Because IBM LPAR technology cannot communicate with the IBM LPAR Host to get usage information, you might not receive alarms for a critical IBM LPAR hosted on that IBM LPAR Host. To find out if a critical IBM LPAR is impacted, you can view a list of affected IBM LPAR instances on the Impact tab of the alarm, as follows:

- Symptoms subview -- displays all symptom alarms generated by the affected IBM LPAR instances
- Management Lost Impact subview -- lists the IBM LPAR instances impacted by the alarm



Huawei SingleCLOUD

This section is for Huawei SingleCLOUD virtualization technology users and describes how to use Virtual Host Manager to manage the virtual entities in your Huawei SingleCLOUD platform.

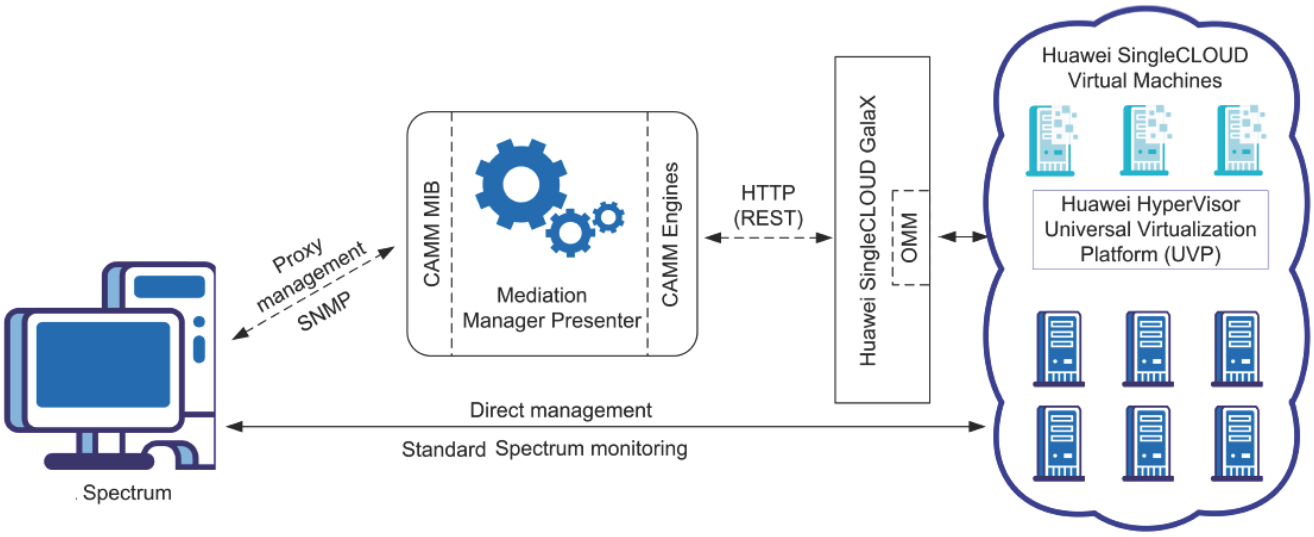
How Virtual Host Manager Works with Huawei SingleCLOUD

Virtual Host Manager monitors your virtual network entities seamlessly with your physical network entities. You get a full view of your network where you can troubleshoot networking issues for both types of entities. Although your virtual network entities behave like physical components, the process for monitoring those entities differs from the general DX NetOps Spectrum monitoring process. Understanding how this process works can help you locate and resolve networking issues related to your virtual network.

The Huawei SingleCLOUD platform consists of a complete system of network, storage, servers, and software for creating private or public clouds. Virtual Host Manager helps you to manage and monitor your Huawei SingleCLOUD virtual environment.

DX NetOps Spectrum gathers information about your Huawei SingleCLOUD virtual environment by two different methods. As with other DX NetOps Spectrum managed devices, Virtual Host Manager uses standard DX NetOps Spectrum monitoring methods. In addition, Virtual Host Manager for Huawei SingleCLOUD also retrieves specialized information from an alternate (proxy) manager, DX Mediation Manager (DX MM).

The following diagram shows how DX NetOps Spectrum gathers information about your Huawei SingleCLOUD environment:



DX Mediation Manager resides on its own host and obtains information from the Huawei SingleCLOUD environment by using HTTP (REST) services to communicate with Huawei SingleCLOUD GalaX, a software suite that collectively manages Huawei SingleCLOUD.

DX MM uses the following components:

- The Engine. The Engine is DX MM's polling engine which gathers information from the Huawei SingleCLOUD platform. The CAMM Engine communicates directly with the Huawei SingleCLOUD GalaX Operation and Management Module (OMM) to obtain information about the Huawei HyperVisor Universal Virtualization Platform (UVP).
- The Presenter. The Presenter receives the information from the Engine and uses it to populate a CA-developed MIB (CAMEDIATIONMANAGER-ENTERPRISES-HUAWEI-SINGLECLOUD-MIB).

DX NetOps Spectrum uses SNMP to retrieve data from the CAMM MIB and uses this information to model and monitor your Huawei SingleCLOUD environment in OneClick.

Models Created for Huawei SingleCLOUD

Virtual Host Manager provides several models to represent the components of your Huawei SingleCLOUD virtual network. Understanding the following basic models can help you better understand Discovery and how the virtual environment interfaces with your physical environment.

Virtual Host Manager includes the following models and icons for Huawei SingleCLOUD entities:

- **Huawei SingleCLOUD CAMM Presenter**
  Represents a CA Mediation Manager (CAMM) Presenter. The CAMM Presenter model allows configuration of the virtual IP addresses used by the CAMM Engines to communicate with Huawei SingleCLOUD GalaX. Each CAMM Presenter can support multiple virtual IP addresses.
  Icon:
  
- **Huawei SingleCLOUD Manager**
  Represents a virtual IP address on the CAMM Presenter. CAMM communicates with Huawei SingleCLOUD GalaX, which is responsible for managing the Huawei SingleCLOUD virtual platform. Information for each Huawei SingleCLOUD GalaX being monitored by CAMM is provided through a virtual IP address on the CAMM Presenter and is represented by a Huawei SingleCLOUD Manager model.
  Icon:
  
- **Huawei SingleCLOUD Cloud**
  Represents a collection of physical and virtual hosts that make up a private or public cloud network.
  Icon:
  
- **Huawei SingleCLOUD Host**
  Represents the Computing Node Agent (CNA) that hosts the virtual machines. In the Universe topology, these models group your virtual entities into a separate view while showing where the virtual environment interfaces with your physical network. The Huawei SingleCLOUD Host cannot be contacted directly for status information. Instead, the status of this model is inferred from the status of its contained items.
  Icon:
  
- **Huawei SingleCLOUD CNA FIP**
  Represents the management interface of the CNA that is hosting the virtual machines. This model is assigned the IP address of the CNA FIP and lives within the host container.

Icon:

- **Huawei SingleCLOUD Virtual Machine**
  Represents a virtual machine, as configured in your Huawei SingleCLOUD platform.
  Icon:

Discovering Huawei SingleCLOUD Networks

Before you can use Virtual Host Manager to monitor your Huawei SingleCLOUD virtual environment, you have to discover and model any network elements that are to be managed. This section describes the discovery and modeling process for Virtual Host Manager for Huawei SingleCLOUD. These tasks are typically performed by the Virtual Host Manager administrator.

Follow these steps:

1. Perform the following pre-Discovery steps:
   a. Define CA Mediation Manager Presenters.
   b. Configure Discovery Options.
2. Discover and model your Huawei SingleCLOUD network.

Define CA Mediation Manager Presenters

After Virtual Host Manager is installed, you define the CA Mediation Manager Presenters to DX NetOps Spectrum. The Huawei SingleCLOUD CAMM Presenter model allows configuration of the virtual IP addresses associated with Huawei SingleCLOUD GalaX. Defining a CAMM Presenter creates a container model that will later be used to organize and contain Huawei SingleCLOUD Manager models.

NOTE
Creating a Huawei SingleCLOUD CAMM Presenter model does not trigger Discovery.

Follow these steps:

1. Select the Virtual Host Manager node in the Explorer tab in the Navigation panel.
   The Contents panel displays information for the Virtual Host Manager feature.
2. Select the Information tab.
3. Expand the Configuration, Huawei SingleCLOUD, CA Mediation Manager Presenters subview.
   The CA Mediation Manager Presenters table appears.
4. Click Add.
   The 'Create Model Of Type HuaweiSCCAMMPresenter' dialog appears.
5. Enter a Name and Description, and click OK.
   The Huawei SingleCLOUD CAMM Presenter model is created and appears in the table.

Configure Discovery Options

Before you perform Discovery to create models for your Huawei SingleCLOUD entities, specify options to control aspects of the Discovery process. Configuring your preferences helps Virtual Host Manager model your virtual devices as expected.

To configure your installation of Virtual Host Manager for Huawei SingleCLOUD Discovery, select your preferences from the following options:

- Maintenance Mode for New Huawei SingleCLOUD Virtual Machines -- Lets you decide whether to place newly discovered virtual machines into maintenance mode until you are ready for DX NetOps Spectrum to manage them.
- Allow Device Model Deletes During Huawei SingleCLOUD Discovery -- Controls how DX NetOps Spectrum handles Huawei SingleCLOUD models when Virtual Host Manager no longer manages them.
- Search for Existing Models -- Determines which secure domains Virtual Host Manager searches during a Huawei SingleCLOUD Discovery.
- Retain SNMP-enabled Virtual Machines During Huawei SingleCLOUD Manager Deletion -- Controls how DX NetOps Spectrum handles SNMP-enabled Huawei SingleCLOUD models when a Huawei SingleCLOUD Manager model is deleted.
- Discover SNMP-Capable Devices -- Controls how SNMP-capable devices are modeled during Huawei SingleCLOUD Discovery. By default, new models are initially created as VHM models only. But this option lets you override the default and immediately create SNMP models for devices that meet the criteria.

Configure Maintenance Mode for New Huawei SingleCLOUD Devices

areVirtual Host Manager automatically models the virtual machines that make up the Huawei SingleCLOUD virtual environment. DX NetOps Spectrum attempts to manage all models that are discovered. However, some newly discovered Huawei SingleCLOUD virtual machines are not ready for DX NetOps Spectrum management when they are initially modeled. To prevent undesired alarms on new Huawei SingleCLOUD Virtual Machine models, you can decide which new models are immediately placed into maintenance mode. Later, you can manually disable maintenance mode when you are ready for DX NetOps Spectrum to manage these devices.

Follow these steps:

1. Click the Virtual Host Manager node in the Explorer tab in the Navigation panel.
   The Contents panel displays information for the Virtual Host Manager feature.
2. Click the Information tab.
3. Expand the Configuration, Huawei SingleCLOUD, Huawei SingleCLOUD Discovery subview.
   Configurable Discovery options appear.
4. Click Set in the 'Maintenance Mode for New Huawei SingleCLOUD Virtual Machines' field and select one of the following options:
   - **Place non-enabled VMs in Maintenance Mode**
     (Default) Applies maintenance mode to only non-enabled Huawei SingleCLOUD Virtual Machine models upon initial Huawei SingleCLOUD Discovery.
   - **Place all VMs in Maintenance Mode**
     Applies maintenance mode to all newly discovered Huawei SingleCLOUD Virtual Machine models upon initial Huawei SingleCLOUD Discovery.

   Your setting is saved and newly discovered Huawei SingleCLOUD Virtual Machine models created by Virtual Host Manager are placed into maintenance mode per your selection.

Manage Device Models for Deleted Huawei SingleCLOUD Devices

The devices and the relationships among them change frequently in virtual environments. Maintaining accurate and timely data about your virtual environment in DX NetOps Spectrum is challenging. For example, when a Huawei SingleCLOUD virtual machine is removed, DX NetOps Spectrum knows to remove the corresponding device models from Virtual Host Manager in the Navigation panel. However, should DX NetOps Spectrum keep or delete the model? You can select settings to control model deletion.

> **WARNING**
> When models are deleted, all notes or other customizations on those models are lost. You can disable this option if your models are likely to be recreated in your Huawei SingleCLOUD environment later.

Follow these steps:

1. Click the Virtual Host Manager node in the Explorer tab in the Navigation panel.
   The Contents panel displays information for the Virtual Host Manager feature.
2. Click the Information tab.
3. Expand the Configuration, Huawei SingleCLOUD, Huawei SingleCLOUD Discovery subview.
   Configurable Discovery options appear.
4. Click Set in the 'Allow Device Model Deletes During Huawei SingleCLOUD Discovery' field and select one of the following options:
   - **Yes**
     (Default) Deletes the Virtual Host Manager models that correspond to entities no longer managed in your Huawei SingleCLOUD environment.
   - **No**
     Places Virtual Host Manager models in the LostFound container if their corresponding entity is no longer managed by your Huawei SingleCLOUD environment, but the models are not deleted from DX NetOps Spectrum.

     > **NOTE**
     > Models with more associations, such as a model that is included in a Global Collection, are handled differently. These models are removed from the Universe, but they are not moved to the LostFound container.

Your setting is saved, and device models are handled accordingly after the device is deleted from your Huawei SingleCLOUD environment.

Configure Model Searches Across Secure Domains (Huawei SingleCLOUD)

Rather than creating new models, Huawei SingleCLOUD Discovery attempts to locate models in the SpectroSERVER. In an environment with Secure Domain Manager deployed, Huawei SingleCLOUD Discovery searches for models within the same secure domain as your Huawei SingleCLOUD Manager. This domain is the "local" domain. However, some of your devices can exist within a different secure domain. In this case, you can configure Huawei SingleCLOUD Discovery to search all secure domains for existing models.

Follow these steps:

1. Click the Virtual Host Manager node in the Explorer tab in the Navigation panel.
   The Contents panel displays information for the Virtual Host Manager feature.
2. Click the Information tab.
3. Expand the Configuration, Huawei SingleCLOUD, Huawei SingleCLOUD Discovery subview.
   Configurable Discovery options appear.
4. Click Set in the 'Search for Existing Models' field and select from the following options:
   - **In Huawei SingleCLOUD Manager's Secure Domain**
     (Default) Searches for models within the same secure domain as the Huawei SingleCLOUD Manager server.
   - **In All Secure Domains**
     Searches for models within all secure domains managed by the SpectroSERVER. Select this option only in the following situations:

- All devices have unique IP addresses
- When secure domains are used for security purposes or to isolate network traffic

  NOTE
  Do not select this option for a NAT environment.

Your setting is saved, and Huawei SingleCLOUD Discovery searches for existing models in DX NetOps Spectrum according to your selection. If duplicate models (that is, models that share the same IP address) exist in multiple secure domains, Virtual Host Manager does the following:

– Selects the model in the local secure domain, if available.
– If a duplicate model does not exist in the local domain, Virtual Host Manager randomly selects a model from another secure domain.

In both cases, Virtual Host Manager generates a minor alarm for the duplicate IP addresses on the Huawei SingleCLOUD Manager model.

Manage Deletion of SNMP-Enabled Huawei SingleCLOUD Models

By default, SNMP-enabled devices are deleted from DX NetOps Spectrum when the following items are deleted:

- Huawei SingleCLOUD folder in the Explorer tab
- Huawei SingleCLOUD CAMM Presenter model
- Huawei SingleCLOUD Manager model

SNMP-enabled device models can include significant customizations that you want to retain. You can adjust your settings to avoid deleting these models. They can be placed into the LostFound container for later use.

Follow these steps:

1. Click the Virtual Host Manager node in the Explorer tab in the Navigation panel.
   The Contents panel displays information for the Virtual Host Manager feature.
2. Click the Information tab.
3. Expand the Configuration, Huawei SingleCLOUD, Huawei SingleCLOUD Discovery subview.
   Configurable Discovery options appear.
4. Click Set in the ‘Retain SNMP-enabled Virtual Machines During Huawei SingleCLOUD Manager Deletion' field and select one of the following options:
   – **Yes**
     Retains SNMP-enabled virtual machine models in the LostFound container when their Huawei SingleCLOUD folder, Huawei SingleCLOUD CAMM Presenter model, or Huawei SingleCLOUD Manager model is deleted.

     NOTE
     Models with more associations, such as a model that is included in a Global Collection, are handled differently. These models are removed from the Universe, but they are not moved to the LostFound container.

   – **No**
     (Default) Deletes all Huawei SingleCLOUD models when their Huawei SingleCLOUD folder, Huawei SingleCLOUD CAMM Presenter model, or Huawei SingleCLOUD Manager model is deleted.

Your setting is saved, and SNMP-enabled device models are handled accordingly when the Huawei SingleCLOUD folder, Huawei SingleCLOUD CAMM Presenter model, or Huawei SingleCLOUD Manager model is deleted.

Configure SNMP Modeling Preferences (Huawei SingleCLOUD)

SNMP-capable devices support enriched device monitoring, such as process and file system monitoring capabilities. However, SNMP agents can be costly and time-consuming to deploy. By default, Huawei SingleCLOUD Discovery creates Huawei SingleCLOUD virtual machines as VHM models. You can later upgrade them to SNMP models. However, you can also configure Huawei SingleCLOUD Discovery to model all new SNMP-capable devices as SNMP models. Although

Huawei SingleCLOUD Discovery may take longer to complete, initially modeling these as SNMP models spares you from manually upgrading these models later.

> **WARNING**
> Enable SNMP modeling *before* modeling Huawei SingleCLOUD components. If you model the Huawei SingleCLOUD components first, any child models are created as VHM models, which must be manually upgraded to SNMP models.

Follow these steps:

1. Click the Virtual Host Manager node in the Explorer tab in the Navigation panel.
   The Contents panel displays information for the Virtual Host Manager feature.
2. Click the Information tab.
3. Expand the Configuration, Huawei SingleCLOUD, Huawei SingleCLOUD Discovery, SNMP Discovery subview.
4. Click Set in the 'Discover SNMP-Capable Devices' field and select from the following options:
   - **Yes**
     Enables SNMP modeling during Huawei SingleCLOUD Discovery. Only devices that meet the criteria in the SNMP Discovery subview text are modeled as SNMP devices. Applies to *new* models only.
   - **No**
     (Default) Models all new devices found during Huawei SingleCLOUD Discovery as VHM models. You can manually upgrade these models to SNMP models later.

   Your setting is saved and new devices are modeled in Virtual Host Manager per your selection.

Discover and Model Your Huawei SingleCLOUD Environment

To monitor your virtual environment, you must discover and model your virtual entities -- Huawei SingleCLOUD Managers, Clouds, Hosts, and Virtual Machines. Modeling these entities in Virtual Host Manager lets you view your complete network topology in one tool, showing the relationships between your physical and virtual components.

The main steps for modeling your virtual environment are as follows:

1. Run a standard Discovery.
   The purpose of this Discovery is to model the upstream routers and switches before Huawei SingleCLOUD Discovery runs. When modeling these entities, be sure that your modeling options are set correctly to support Virtual Host Manager.
2. Define Huawei SingleCLOUD Managers.
   This step discovers and models the virtual IP addresses that CAMM uses to communicate with Huawei SingleCLOUD GalaX, the management application for the Huawei SingleCLOUD virtual platform. DX NetOps Spectrum uses these models to retrieve information about the Huawei SingleCLOUD architecture and its virtual machines.
3. Let Huawei SingleCLOUD Discovery run.
   When you model the Huawei SingleCLOUD Manager, Huawei SingleCLOUD Discovery begins automatically, discovering and modeling the virtual entities in your Huawei SingleCLOUD environment.
4. (Optional) Add SNMP Capabilities to VHM Models.
   If you modeled Huawei SingleCLOUD entities as VHM models. you can upgrade them to SNMP models.
5. (Optional) Move a Huawei SingleCLOUD Host to a Different Huawei SingleCLOUD GalaX.
   When moving a Huawei SingleCLOUD Host from management by one Huawei SingleCLOUD GalaX to another, steps should be performed in a certain order to accurately reflect the changes in the modeled DX NetOps Spectrum environment.

Each of these steps is described in detail in the following sections.

Run Discovery (Huawei SingleCLOUD)

To accurately reflect your complete Huawei SingleCLOUD environment, run standard DX NetOps Spectrum Discovery to locate any connecting devices. Upstream routers and switches are modeled so that later connections from the virtual entities can be established.

NOTE
Only an administrator performs this task.

Follow these steps:

1. Open the Discovery console.

   NOTE
   Before modeling, be sure that you know the correct community strings, IP addresses, and port numbers for any SNMP agents that run on a nonstandard port.
2. In the Navigation panel. click Creates a new configuration icon  .
   The Configuration dialog opens.
3. Specify a name and location for the new configuration, and click OK.
   The Configuration dialog closes.
4. Enter individual IP addresses or the beginning and ending IP addresses in the IP/Host Name Boundary List fields and click Add.

   NOTE
   Be sure that the range of IP addresses includes all the interconnecting switches and routers.
5. Configure your Modeling Options as follows:
   a. Select the 'Discover and automatically model to CA Spectrum' option.
   b. Click the Modeling Options button.
      The Modeling Configuration dialog opens.
   c. Click the Protocol Options button.
      The Protocol Options dialog opens.
   d. Select the ARP Tables for Pingables option, and click OK.
      The Protocol Options dialog closes.
   e. Click OK to close the Modeling Configuration dialog.
6. (Optional) Click the Advanced Options button in the Advanced Options group, add your nonstandard SNMP ports (such as the CAMM port), and click OK.
7. Enter any additional values in the Discovery console, and click Discover.
   Models are created and added to your network topology in DX NetOps Spectrum for the switches and routers that connect your Huawei SingleCLOUD entities to your network.

Define Huawei SingleCLOUD Managers

After the Huawei SingleCLOUD CAMM Presenter has been defined and you have modeled your connecting devices, you can model and discover your Huawei SingleCLOUD Managers. The successful creation of the Huawei SingleCLOUD Manager model automatically triggers Huawei SingleCLOUD Discovery.

Follow these steps:

1. Select the Huawei SingleCLOUD CAMM Presenter in the Huawei SingleCLOUD folder in the Virtual Host Manager hierarchy in the Explorer tab in the Navigation panel.
   The Component Detail panel displays information for the CAMM Presenter.
2. In the Information tab in the Component Detail panel, expand the Huawei SingleCLOUD Managers subview.
   The Huawei SingleCLOUD Managers table appears.
3. Click Add.

The 'Create Model Of Type HuaweiSCManager' dialog appears.

4. Enter the information for your Huawei SingleCLOUD Manager, and click OK. Notice the following field:
   – **Network Address**
     Enter the virtual IP address used by the CAMM Presenter to communicate with Huawei SingleCLOUD GalaX.

     WARNING
     This value should not be the same as the primary IP address of the device or virtual machine where the CAMM Presenter is installed.

   A Huawei SingleCLOUD Manager model is created and appears in the table. Information about your Huawei SingleCLOUD environment comes from the Huawei SingleCLOUD Manager. When this model is created, Huawei SingleCLOUD Discovery begins.

Huawei SingleCLOUD Discovery

Huawei SingleCLOUD Discovery is a specialized discovery process that gathers detailed information about your virtual environment. The purpose of Discovery is to discover and model the virtual entities in the Huawei SingleCLOUD platform. Understanding how Huawei SingleCLOUD Discovery works reinforces the importance of installing and modeling the various components of Virtual Host Manager properly.

A key benefit of Huawei SingleCLOUD Discovery is that it runs automatically in the background, continually updating virtual environment data in DX NetOps Spectrum. Because of this automated capability, portions of Huawei SingleCLOUD Discovery have already occurred in previous steps. The following description explains the Huawei SingleCLOUD Discovery in its entirety and is provided for reference. No action is required.

The Huawei SingleCLOUD Discovery process works as follows:

1. Immediately after CAMM and the Huawei SingleCLOUD Device Pack are installed correctly, the CAMM Engine starts communicating with Huawei SingleCLOUD GalaX. Information is processed by the CAMM Presenter and is made available to DX NetOps Spectrum.

   WARNING
   CA Mediation Manager and the Huawei SingleCLOUD Device Pack must be installed and configured so that DX NetOps Spectrum, CAMM, and Huawei SingleCLOUD GalaX can communicate. If they cannot, Huawei SingleCLOUD Discovery cannot run.
2. During DX NetOps Spectrum Discovery, DX NetOps Spectrum creates models for connecting devices so that later connections from the virtual entities can be established.
3. Huawei SingleCLOUD CAMM Presenter and Huawei SingleCLOUD Manager models are created. Creation of the Huawei SingleCLOUD Manager enables DX NetOps Spectrum to handle communication between DX NetOps Spectrum and CAMM.
4. The Huawei SingleCLOUD Manager polls CAMM to gather information about the Huawei SingleCLOUD environment, which was gathered in Step 1.
5. DX NetOps Spectrum begins Huawei SingleCLOUD Discovery. DX NetOps Spectrum uses this information to update modeling in the DX NetOps Spectrum Topology tab and the Virtual Host Manager hierarchy in the Navigation panel, as follows:
   a. If you enable SNMP Discovery before Step 2, Virtual Host Manager Discovery creates SNMP models for all new SNMP-capable models that meet the SNMP Discovery criteria.

      NOTE
      By default, SNMP Discovery is disabled during Huawei SingleCLOUD Discovery.
   b. VHM models are created for the remaining non-SNMP Huawei SingleCLOUD entities.

NOTE
In a virtual environment, devices on separate Huawei SingleCLOUD Hosts can have the same IP or MAC address. In this case, DX NetOps Spectrum creates duplicate models for each occurrence of an IP address or MAC address.

6. Huawei SingleCLOUD Discovery repeats this process at each regularly scheduled Huawei SingleCLOUD Manager polling interval.

Add SNMP Capabilities to VHM Models (Huawei SingleCLOUD)

SNMP-capable devices support enriched device monitoring, such as process and file system monitoring capabilities. However, SNMP agents can be costly and time-consuming to deploy. When an SNMP agent is not available or SNMP Discovery is disabled, Virtual Host Manager creates Huawei SingleCLOUD entities as VHM models.

Later, you can install an SNMP agent on any Huawei SingleCLOUD Host or virtual machine and upgrade its modeling in DX NetOps Spectrum. Options for upgrading to SNMP models are as follows:

- Upgrade only selected devices -- This method works quickly when you have a small selection of models to upgrade. The VHM models are deleted first. One drawback of this method is that after DX NetOps Spectrum deletes the models, you must wait for the next Huawei SingleCLOUD Discovery to create the new SNMP models and place them in Virtual Host Manager. Knowledge of the IP addresses for the models to upgrade is required.
- Upgrade all SNMP-capable VHM models -- This method upgrades models in batch, and this method is preferred when upgrading Virtual Host Manager to a new release. Knowledge of the IP addresses of individual models is not required. Another advantage is that after DX NetOps Spectrum deletes the VHM models, the upgraded SNMP models are immediately placed in the Virtual Host Manager hierarchy without waiting for the next polling cycle. Therefore, Virtual Host Manager manages the models more quickly.
  One drawback of this method is that it can take a long time to complete. The time required to complete this upgrade depends on how many community strings and SNMP ports Virtual Host Manager must search when locating SNMP-capable devices.

  NOTE
  Virtual Host Manager attempts to identify SNMP agents on powered-up pingable devices only.

  WARNING
  When models are deleted, all notes or other customizations on those models are lost.

Upgrade Selected VHM Models to SNMP Models (Huawei SingleCLOUD)

When an SNMP agent is not available or SNMP Discovery is disabled during Huawei SingleCLOUD Discovery, Virtual Host Manager creates Huawei SingleCLOUD Hosts and Virtual Machines as VHM models. Later, you can install an SNMP agent on these devices and upgrade their modeling in DX NetOps Spectrum. You must know the IP addresses for the device models to upgrade. Manually selecting models to upgrade works quickly, but all notes or customizations on these models are lost during the upgrade.

Follow these steps:

1. Deploy or enable an SNMP agent on the device, if required.
2. Model the device again using one of the following methods:
   – DX NetOps Spectrum Discovery
   – Model individual devices by IP address

   When the new SNMP-capable model is created, DX NetOps Spectrum removes the previous model from Virtual Host Manager and deletes it. At the next Huawei SingleCLOUD Manager polling cycle, DX NetOps Spectrum adds the SNMP-capable model to Virtual Host Manager in the Navigation panel.

   WARNING
   When models are deleted, all notes or other customizations on those models are lost.

Upgrade All VHM Models to SNMP Models (Huawei SingleCLOUD)

When an SNMP agent is not available or SNMP Discovery is disabled during Huawei SingleCLOUD Discovery, Virtual Host Manager creates Huawei SingleCLOUD Hosts and Virtual Machines as VHM models. Later, you can install an SNMP agent on these devices and upgrade their modeling in DX NetOps Spectrum. When upgrading in batch, DX NetOps Spectrum searches your VHM models, and locates SNMP-capable devices. Then DX NetOps Spectrum converts these to SNMP models. This method can take a long time to complete, depending on how many community strings and ports Virtual Host Manager must search.

Follow these steps:

1. Deploy or enable an SNMP agent on your devices, as required.
2. Open Virtual Host Manager in the Navigation panel.
   The main details page opens in the Contents panel for the selected Virtual Host Manager.
3. Select the Huawei SingleCLOUD Manager model in the Navigation panel that manages the models to upgrade.
4. Click the Information tab.
5. Expand the Huawei SingleCLOUD Manager, DX NetOps Spectrum Modeling Control subview.
6. Click the Upgrade ICMP-Only Devices button.

   WARNING
   When models are deleted, all notes or other customizations on those models are lost.

Virtual Host Manager searches for VHM models that are managed by the Huawei SingleCLOUD Manager. Virtual Host Manager upgrades the ICMP-only devices that meet the criteria for SNMP devices and places them within the Virtual Host Manager hierarchy.

Move a Huawei SingleCLOUD Host to a Different Huawei SingleCLOUD GalaX

Moving a Huawei SingleCLOUD Host from management by one Huawei SingleCLOUD GalaX to another can cause problems with DX NetOps Spectrum modeling when both Huawei SingleCLOUD Managers are modeled on the same SpectroSERVER.

Some possible symptoms of these modeling problems are as follows:

- DX NetOps Spectrum deletes the models associated with the host but does not recreate them after the move.
- False Proxy Lost alarms are created and remain, even though the new managing Huawei SingleCLOUD GalaX can contact the host and all hosted virtual machines.

To avoid these problems, perform the steps to move your host and reflect the changes in your modeled DX NetOps Spectrum environment in the correct order, as follows.

Follow these steps:

1. (Optional) Change the 'Allow Device Model Deletes During Huawei SingleCLOUD Discovery' option to No.

   NOTE
   Perform this step only if both the originating and destination Huawei SingleCLOUD Managers are modeled in the same SpectroSERVER. Setting this option to No keeps the existing Huawei SingleCLOUD Host, CNA FIP, and Virtual Machine models from being deleted when they become unmanaged by the first Huawei SingleCLOUD GalaX. Therefore, customizations or historical details for the models are preserved and available after the move.
2. Use Huawei SingleCLOUD GalaX to remove the host from management.
3. Wait for Virtual Host Manager to reflect the changes in the Navigation panel.
4. Use the destination Huawei SingleCLOUD GalaX to add management of the host.

NOTE
Virtual Host Manager is not DSS aware. Therefore, when moving the host to a Huawei SingleCLOUD GalaX managed on a different SpectroSERVER, a new set of models are created to represent the host, CNA FIP, and virtual machines.

5. (Optional) Change the 'Allow Device Model Deletes During Huawei SingleCLOUD Discovery' option back to Yes. The host is successfully moved from management by one Huawei SingleCLOUD GalaX to another and accurately reflected in the DX NetOps Spectrum modeled environment.

Viewing Your Huawei SingleCLOUD Virtual Environment

This section describes concepts for viewing your Huawei SingleCLOUD virtual environment. The basic steps are no different from the standard DX NetOps Spectrum procedures. However, this section describes conceptual differences and details that only apply to the Huawei SingleCLOUD platform.

Viewing Your Huawei SingleCLOUD Virtual Network

On the Explorer tab under the Virtual Host Manager node, the expanded Huawei SingleCLOUD folder displays a hierarchical tree structure that helps you visualize the logical organization of your managed Huawei SingleCLOUD environment.

Using this information, you can see how resources are shared across your Huawei SingleCLOUD Managers, which can help you identify opportunities to reorganize and optimize your virtual environment. This hierarchy also provides a quick way to validate the appropriate status of allocated resources in your cloud architecture, monitor performance, and troubleshoot alarms.

Because Virtual Host Manager is not aware of a DSS environment, it is located within a landscape hierarchy. The following diagram shows where Virtual Host Manager appears on the Explorer tab in the Navigation panel and illustrates the Huawei SingleCLOUD hierarchy:

```
[-] <ss> host
    [+] Universe
    [-] Virtual Host Manager
        [-] Huawei SingleCLOUD
            [+] Huawei SingleCLOUD CAMM Presenter 1
            [-] Huawei SingleCLOUD CAMM Presenter 2
                [-] Huawei SingleCLOUD Manager 1
                     [+] Huawei SingleCLOUD Cloud 1
                     [-] Huawei SingleCLOUD Cloud 2
                         [+] Huawei SingleCLOUD Host 1
                         [-] Huawei SingleCLOUD Host 2
                             . Huawei SingleCLOUD CNA FIP
                             . Huawei SingleCLOUD Virtual Machine 1
                             . Huawei SingleCLOUD Virtual Machine 2
                [+] Huawei SingleCLOUD Manager 2
                [+] Huawei SingleCLOUD Manager 3
```

Virtual Host Manager is the root node for the entire virtual environment managed by this SpectroSERVER. Selecting this node in the Navigation panel displays Virtual Host Manager details in the Contents panel. You can view details such as events and alarms related to your virtual environment as a whole.

Directly under Virtual Host Manager, virtual environments are organized within folders that represent the technology with which they are created. In the example hierarchy above, the Huawei SingleCLOUD folder contains the portion of the virtual environment that was created and managed by Huawei SingleCLOUD. In this folder, Virtual Host Manager lists

all CAMM Presenters, Huawei SingleCLOUD Managers, and clouds managed by this SpectroSERVER. Each Huawei SingleCLOUD Manager contains only the portion of the virtual environment that it manages.

The hierarchy represents the logical relationships between the following virtual entities:

- **Huawei SingleCLOUD CAMM Presenter**
  A Huawei SingleCLOUD CAMM Presenter node groups together all of the Huawei SingleCLOUD Managers that it manages. Selecting a CAMM Presenter provides access to the Huawei SingleCLOUD Managers subview, where you define the Huawei SingleCLOUD Managers managed by the Presenter by specifying the virtual IP addresses used to communicate with Huawei SingleCLOUD GalaX. Selecting the CAMM Presenter node also displays events and alarms related to the entities in its managed environment. A Physical Host Down alarm generated on the Huawei SingleCLOUD CAMM Presenter model indicates that all the Huawei SingleCLOUD Managers (virtual IP addresses) it manages have gone down.
- **Huawei SingleCLOUD Manager**
  A Huawei SingleCLOUD Manager represents the virtual IP address used by CAMM to communicate with the Huawei SingleCLOUD GalaX. DX NetOps Spectrum uses the virtual IP address to obtain information from the Huawei SingleCLOUD GalaX about the Huawei SingleCLOUD environment it manages. Selecting a Huawei SingleCLOUD Manager displays information about its managed environment, such as details about the GalaX OMMs, managed clouds, hosts, and virtual machines, including when data in the MIB was updated last. Traps received from the Huawei SingleCLOUD trap service are generated on the Huawei SingleCLOUD Manager model.
- **Huawei SingleCLOUD Cloud**
  A Huawei SingleCLOUD Cloud is the name of the managed cloud as defined in the Huawei SingleCLOUD platform. Selecting a Huawei SingleCLOUD Cloud displays details about the cloud, including:
  – Cloud type (public or private).
  – MIB-related status information, including when data for the cloud was updated last and how much time is expected between updates.

  In the hierarchy, beneath each cloud are the Huawei SingleCLOUD Hosts associated with the cloud.
- **Huawei SingleCLOUD Host**
  A Huawei SingleCLOUD Host is a CNA that hosts virtual machines. In the hierarchy, beneath each host are the CNA FIP and the virtual machines it manages. A Huawei SingleCLOUD Host can be part of a public or private cloud. Selecting the Huawei SingleCLOUD Host displays detailed host information including:
  – Cloud type (public or private) that the host is a part of.
  – Resources consumed by the CNA process, such as storage, CPU, and memory utilization.
  – Geographical information of the host for quick physical location during fault resolution.
  – MIB-related status information, including when data for the host was updated last and how much time is expected between updates.

  NOTE
  Available host information differs depending on if the host is part of a public or private cloud. The cloud type is identified in the Host Information, CNA, Properties subview.
- **Huawei SingleCLOUD CNA FIP**
  The Huawei SingleCLOUD CNA FIP is the management interface to the hosted virtual machines. The model appears as a child to its corresponding Huawei SingleCLOUD Host model and is always a leaf node on the Virtual Host Manager hierarchy tree.

  NOTE
  The Huawei SingleCLOUD CNA FIP model is the only Huawei SingleCLOUD model type that does not provide a Virtual Host Manager-specific subview on the Information tab.
- **Huawei SingleCLOUD Virtual Machine**
  A virtual machine is always a leaf node on the Virtual Host Manager hierarchy tree. Selecting a virtual machine displays details including:

- Identifying information, such as IP address and MAC address.
- Resource information, including storage, CPU, and memory utilization.
- MIB-related status information, including when data for the virtual machine was updated last and how much time is expected between updates.

Understanding the Huawei SingleCLOUD Virtual Topology

The Huawei SingleCLOUD Host, CNA FIP, and Virtual Machine models created for your virtual environment are integrated into the topology view. Huawei SingleCLOUD Host models automatically group their associated CNA FIP and Virtual Machine models. The topology shows how these elements are connected to your physical network entities.

The following example shows how these models can appear on the Explorer tab in the Navigation panel under the Universe group:

```
[-] Universe
    [-] Huawei SingleCLOUD Host
          .  Huawei SingleCLOUD CNA FIP
          .  Huawei SingleCLOUD Virtual Machine 1
          .  Huawei SingleCLOUD Virtual Machine 2
          .  Huawei SingleCLOUD Virtual Machine 3
```

Selecting one of these models displays these relationships graphically on the Topology tab in the Contents panel.

How the Huawei SingleCLOUD Data is Updated in Virtual Host Manager

During your initial Huawei SingleCLOUD Discovery, DX NetOps Spectrum populates the Virtual Host Manager hierarchy in the Navigation panel with your virtual device models. After DX NetOps Spectrum builds this initial hierarchy, your virtual network configuration can change, and Virtual Host Manager must continually work to keep this information accurate in DX NetOps Spectrum. For example, the following events can change your virtual network configuration:

- Creating or deleting clouds, hosts, or virtual machines
- Moving a virtual machine from one Huawei SingleCLOUD host to another

To keep your information accurate, Virtual Host Manager detects these changes by polling the CAMM Presenter to retrieve information about the virtual environment from the CAMM MIB. Accordingly, your virtual network configuration is updated in DX NetOps Spectrum at each polling cycle. Because of the communication with Huawei SingleCLOUD GalaX, DX NetOps Spectrum is aware of spontaneous network configuration changes (such as migrations and host outages), which are quickly reflected in OneClick and factored into the root cause analysis.

When DX NetOps Spectrum detects a change in your virtual network configuration, DX NetOps Spectrum performs the following tasks:

- Updates the placement of your virtual device models in the Virtual Host Manager hierarchy of the Navigation panel
- *Automatically* rediscovers connections to the affected models and associates them with the correct Huawei SingleCLOUD Host in the Universe topology

  WARNING
  To reestablish connections to your virtual models correctly, all interconnecting routers and switches in your physical network must be modeled. If these models do not exist before connections to your virtual devices are rediscovered, DX NetOps Spectrum cannot resolve those connections and display the information correctly in the Universe topology view.

In addition to polling-based events, DX NetOps Spectrum also supports traps from the Huawei SingleCLOUD trap service and generates corresponding events. By reviewing the event log, you can find out when configuration changes occur, such as when a virtual machine is created or migrated.

CAMM MIB Updates

DX NetOps Spectrum retrieves information about the Huawei SingleCLOUD environment from the CAMM MIB. To keep the modeling in DX NetOps Spectrum accurate, the data in the MIB must be up-to-date.

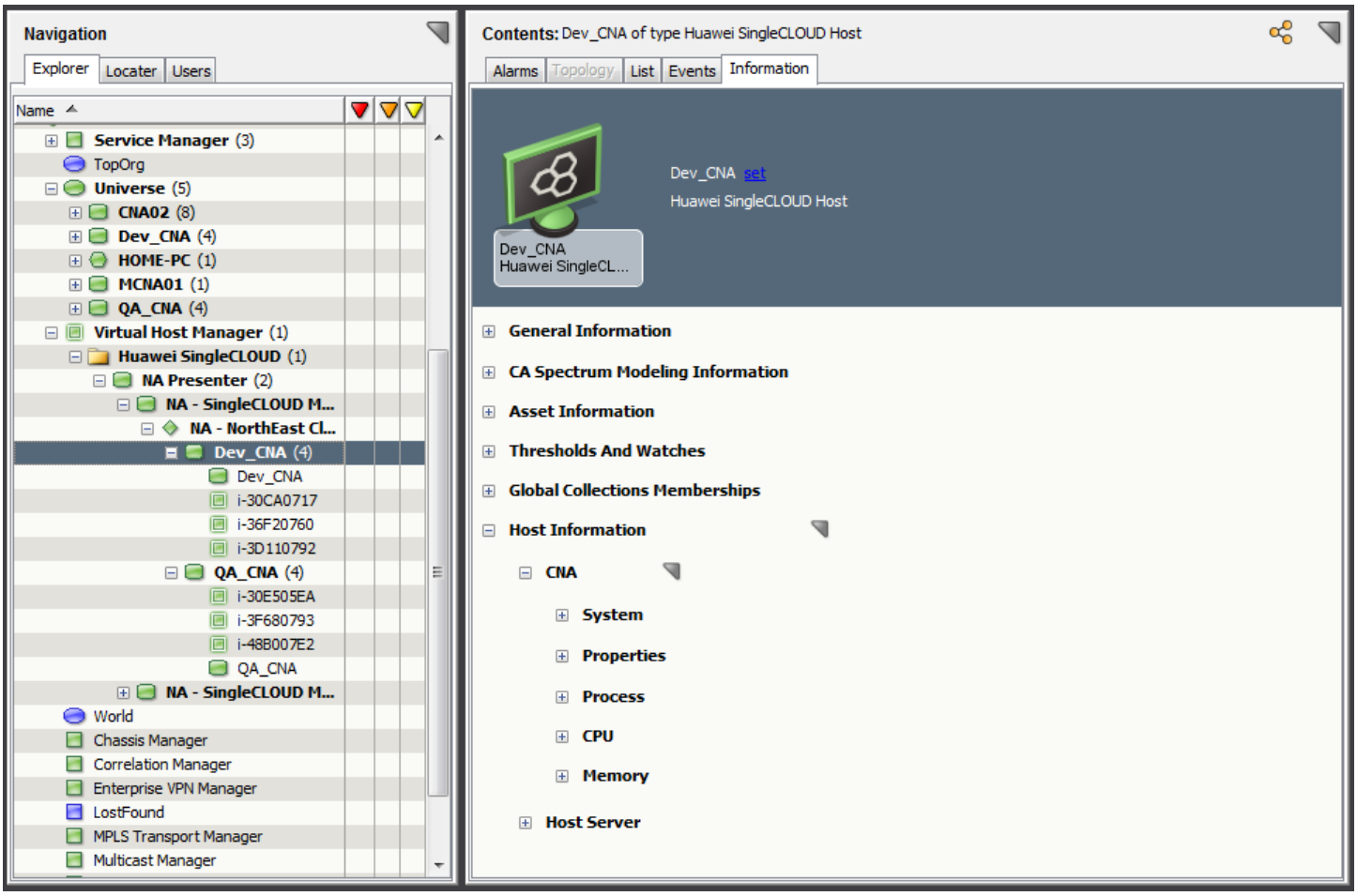
CAMM obtains data from Huawei SingleCLOUD GalaX and updates the MIB according to a configured poll rate in the CAMM Engine. DX NetOps Spectrum uses SNMP to retrieve data from the MIB according to the poll interval specified on the Huawei SingleCLOUD Manager model.

You can determine when CAMM last updated the MIB and when it is expected to be updated again by using the custom subviews for various Huawei SingleCLOUD entity models, where the following MIB-related status information fields are provided:

- **Last Updated**
  Displays when the MIB was last updated. This timestamp value is updated by CAMM when information about the respective entity is obtained from Huawei SingleCLOUD GalaX. You can use this value to determine how current the MIB information for the entity is.
- **Expected Update Delta (s)**
  Displays the expected amount of time between MIB updates, in seconds. If a host or virtual machine has not been updated within the expected amount of time, a Proxy Lost alarm is generated to indicate that updated information cannot be obtained for the entity.

Custom Subviews

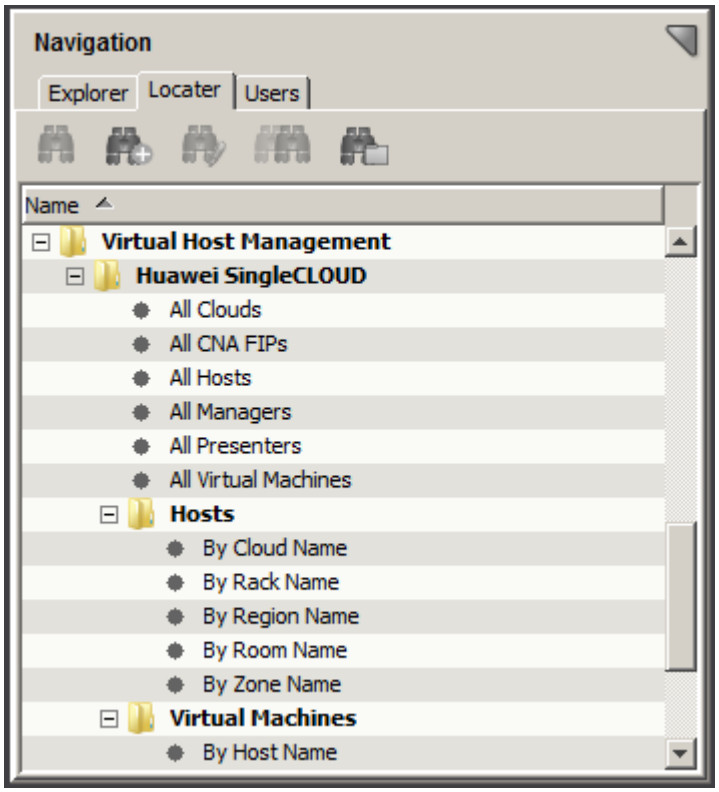
Your Virtual Host Manager models collectively provide information about your virtual environment. Individually, each model provides unique information or configuration settings, depending on the virtual entity type it represents. Custom subviews appear on the Information tab in the Contents panel. These subviews can contain real-time data, such as CPU status or memory utilization. For example, the custom subview for a Huawei SingleCLOUD Host model is the Host Information subview, which provides details specific to a host, as shown:



NOTE
The Huawei SingleCLOUD Manager model provides combined information for all virtual devices managed by the Huawei SingleCLOUD Manager. Selecting the Huawei SingleCLOUD Manager model in the Navigation panel displays unique information, such as about the Huawei SingleCLOUD GalaX OMMs, as well as combined information about all Huawei SingleCLOUD clouds, hosts, and virtual machines it manages. This information contains some of the same data that is displayed on the Information tab for each individual entity model. The combined view in the Huawei SingleCLOUD Manager model can provide a good overview about all of the virtual entities it manages.

Locater Tab for Huawei SingleCLOUD Searches

In addition to viewing details about your virtual environment on the Explorer tab, you can also use the Locater tab to run preconfigured searches. The search options are grouped under the Virtual Host Management, Huawei SingleCLOUD folder on the Locater tab, as shown:



These detailed searches can help you investigate information related to Huawei SingleCLOUD entities that have been modeled in the DX NetOps Spectrum database.

NOTE
Although Virtual Host Manager is not DSS aware, these preconfigured searches let you select multiple landscapes to search in the search parameters.

NOTE
The following types of searches are provided for Huawei SingleCLOUD:

- **Huawei SingleCLOUD**
  Locates Huawei SingleCLOUD entities that have been modeled in the DX NetOps Spectrum database by model type. These include:
  - All Clouds
  - All CNA FIPs
  - All Hosts
  - All Managers
  - All Presenters
  - All Virtual Machines
- **Hosts**
  Locates Huawei SingleCLOUD Hosts by cloud name or geographic location (rack, region, room, and zone).
- **Virtual Machines**
  Locates all virtual machines that reside on a particular Huawei SingleCLOUD Host.

Deleting Virtual Host Manager Models (Huawei SingleCLOUD)

Models can be deleted from OneClick at any time for various reasons. However, Virtual Host Manager restricts your ability to delete models from the Virtual Host Manager hierarchy in the Navigation panel. To delete models manually, you have the following options:

- Delete the Huawei SingleCLOUD folder, a Huawei SingleCLOUD Presenter model, or a Huawei SingleCLOUD Manager model in Virtual Host Manager
- Remove a virtual entity using Huawei SingleCLOUD GalaX

In Virtual Host Manager, models are sometimes deleted automatically. The following circumstances cause DX NetOps Spectrum to automatically delete Virtual Host Manager models:

- **The Huawei SingleCLOUD folder, a Huawei SingleCLOUD CAMM Presenter model, or a Huawei SingleCLOUD Manager model is deleted**
  If you delete the Huawei SingleCLOUD folder, a Huawei SingleCLOUD CAMM Presenter model, or a Huawei SingleCLOUD Manager model from the Navigation panel, DX NetOps Spectrum deletes all related child models. For the CAMM Presenter model, this includes all virtual IP addresses.
- **An entity is removed from a Huawei SingleCLOUD virtual environment**
  As you delete Huawei SingleCLOUD hosts and virtual machines using Huawei SingleCLOUD GalaX, DX NetOps Spectrum also deletes those models and their child models from Virtual Host Manager.
- **Upgraded models exist.**
  In some cases, a virtual machine is first modeled for Virtual Host Manager without SNMP capabilities. If SNMP capabilities are later added to a VHM model, the previous model is deleted and replaced with the new SNMP-capable model.

  NOTE
  Although the default setting is to delete the models, you can configure Virtual Host Manager to place the Huawei SingleCLOUD Host and Huawei SingleCLOUD Virtual Machine models in the LostFound container when they are removed from Virtual Host Manager. This configuration setting applies only when you remove an entity using Huawei SingleCLOUD GalaX. However, this setting does not apply when you delete the Huawei SingleCLOUD folder, delete a Huawei SingleCLOUD Manager model, or upgrade a VHM model.

Alarms and Fault Isolation for Huawei SingleCLOUD

To alert you to problems within your virtual network, DX NetOps Spectrum generates alarms. Quickly identifying any device faults helps you to maximize system up-time and the reliability of your cloud architecture. Alarms are created by:

- Traps sent from Huawei SingleCLOUD.
- Polling. Alarms are generated for the following conditions:
  - A Huawei SingleCLOUD Manager (proxy) is down or communication is lost.
  - A Huawei SingleCLOUD virtual machine is not running.
  - CAMM has not been updated within the defined poll rate.
  - A virtual machine has moved to a new Huawei SingleCLOUD Host.
  - An unsupported Virtual Host Manager configuration has been encountered.

  Alarms that are generated from polling are described in Fault Management for Huawei SingleCLOUD.

Traps for Huawei SingleCLOUD

Traps are generated by the Huawei SingleCLOUD trap service and identify events related to configuration changes, process status, disk or memory usage, and power supply or fan status, as well as others. Traps are generated on the Huawei SingleCLOUD Manager model and can generate alarms in DX NetOps Spectrum.

This section includes the following topics:

Traps and Alarm Severity for Huawei SingleCLOUD

If a trap is received and generates an alarm, DX NetOps Spectrum uses the value of the “state” varbind passed with the trap to determine the alarm severity. DX NetOps Spectrum maps these Huawei SingleCLOUD states to DX NetOps Spectrum alarm severity, as shown:

Huawei SingleCLOUD State	DX NetOps Spectrum Alarm Severity
0: Warning	Minor (Yellow)
1: Minor	Minor (Yellow)
2: Major	Major (Orange)
3: Critical	Critical (Red)

Supported Traps for Huawei SingleCLOUD

The following tables provide the supported Huawei SingleCLOUD traps and their respective trap types. The value of the OID suffix (the lowest node in the trap OID) indicates the trap type.

OID Suffix	Trap Type
.1	Set
.2	Update
.3	Clear

Trap Name	Trap OID
Config Management Agent Process Abnormal	1.3.6.1.4.1.60001.10.1.10.1000001.6.1 1.3.6.1.4.1.60001.10.1.10.1000001.6.2 1.3.6.1.4.1.60001.10.1.10.1000001.6.3
Occupied Space in the Directory Too High	1.3.6.1.4.1.60001.10.1.15.1000203.6.1 1.3.6.1.4.1.60001.10.1.15.1000203.6.2 1.3.6.1.4.1.60001.10.1.15.1000203.6.3
CNA Node Disk Usage Above the Threshold	1.3.6.1.4.1.60001.10.1.15.1000036.6.1 1.3.6.1.4.1.60001.10.1.15.1000036.6.2 1.3.6.1.4.1.60001.10.1.15.1000036.6.3
Hard Disk Lost	1.3.6.1.4.1.60001.10.1.15.1000202.6.1 1.3.6.1.4.1.60001.10.1.15.1000202.6.2 1.3.6.1.4.1.60001.10.1.15.1000202.6.3
VM MEM Usage Over the Threshold	1.3.6.1.4.1.60001.10.1.15.1000102.6.1 1.3.6.1.4.1.60001.10.1.15.1000102.6.2 1.3.6.1.4.1.60001.10.1.15.1000102.6.3

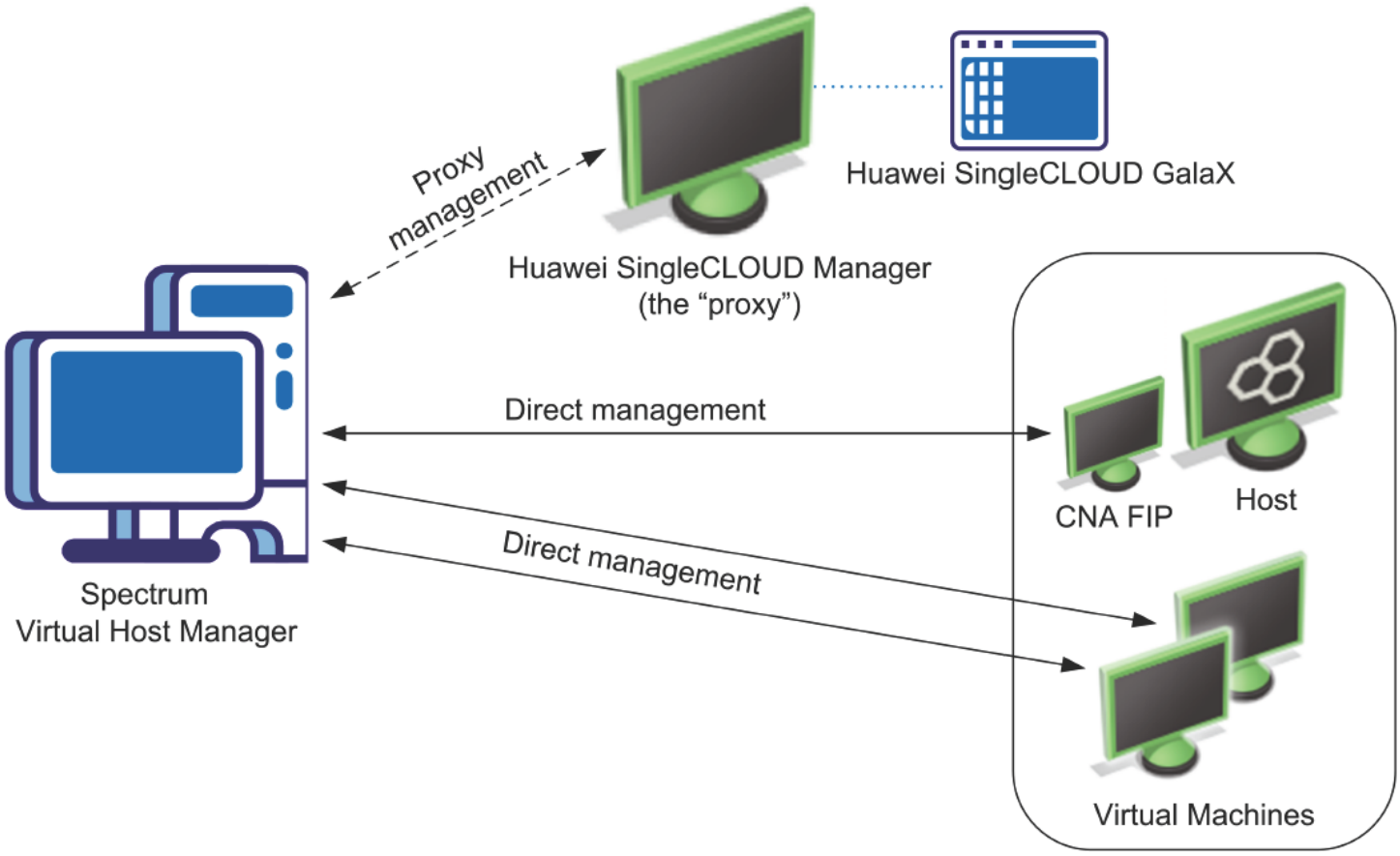
Fan Status Abnormal	1.3.6.1.4.1.60001.10.1.15.1000017.6.1 1.3.6.1.4.1.60001.10.1.15.1000017.6.2 1.3.6.1.4.1.60001.10.1.15.1000017.6.3
Controller Node Hard Disk Usage Above the Threshold	1.3.6.1.4.1.60001.10.1.15.1000015.6.1 1.3.6.1.4.1.60001.10.1.15.1000015.6.2 1.3.6.1.4.1.60001.10.1.15.1000015.6.3
Power Supply Status Abnormal	1.3.6.1.4.1.60001.10.1.15.1000016.6.1 1.3.6.1.4.1.60001.10.1.15.1000016.6.2 1.3.6.1.4.1.60001.10.1.15.1000016.6.3

Fault Management for Huawei SingleCLOUD

The goal of fault isolation is to narrow down the root cause of a networking problem. Finding the root cause can help you to troubleshoot and quickly correct the problem or to correct the problem programmatically with automated scripts. Deciding which device is the root cause of an alarm can be difficult, because problems with a single device can cause several devices in your network to generate events.

For example, losing contact with a Huawei SingleCLOUD Host often means that you have also lost contact with the virtual machines it manages. Therefore, the Huawei SingleCLOUD Host model and all affected virtual machines generate alarms. Using fault isolation techniques, Virtual Host Manager correlates these alarms in an attempt to identify a single root cause.

Virtual networks provide a unique management opportunity because they provide DX NetOps Spectrum an alternate management perspective. That is, DX NetOps Spectrum can gather information through direct contact with your virtual devices or through CAMM, which communicates with Huawei SingleCLOUD GalaX.



This alternate management perspective enhances standard DX NetOps Spectrum fault management in two ways:

- **Enhanced Contact Lost alarms** -- Two sources of information about a device means Virtual Host Manager can pinpoint the cause and more easily correlate events to a single root cause.
- **Proxy Failure alarms** -- Proxy management is the act of managing network devices using an alternate management source in place of or in addition to the primary manager. For example, DX NetOps Spectrum can manage Huawei SingleCLOUD virtual machines by contacting them directly or through CAMM, which obtains data from Huawei SingleCLOUD GalaX. When the Huawei SingleCLOUD Manager loses contact with Huawei SingleCLOUD GalaX or when Huawei SingleCLOUD GalaX loses contact with a virtual device, Virtual Host Manager generates one of the proxy management alarms for each device. These alarms alert you to the fact that *management* of the device through the *proxy* is affected, not the state of the device or direct (SNMP) management.

How Fault Isolation Works when Device Contact is Lost (Huawei SingleCLOUD)

To help you troubleshoot networking problems with your devices, DX NetOps Spectrum uses fault isolation to narrow down the root cause of an alarm. For virtual networks, Virtual Host Manager uses information from direct contact with the device plus information provided by Huawei SingleCLOUD GalaX through CAMM. In many cases, standard DX NetOps Spectrum fault management can pinpoint the root cause. But in special circumstances, the approach for isolating problems in a virtual network goes beyond the standard methods.

The type of fault isolation Virtual Host Manager uses to discover the root cause depends on which devices are alarming and the type of events the devices generate. The following scenarios describe fault management situations and how DX NetOps Spectrum isolates the networking error in your virtual network.

Scenario 1: Huawei SingleCLOUD Virtual Machine is not running

In a virtual environment, the virtual management application can provide more details than DX NetOps Spectrum can discover through standard device monitoring. For example, Huawei SingleCLOUD GalaX is aware when a virtual machine changes from the running state to another state.

If a virtual machine is no longer running and DX NetOps Spectrum loses contact with it, but proxy management of the virtual machine is uninterrupted, DX NetOps Spectrum determines the root cause as follows:

1. When DX NetOps Spectrum loses contact with a virtual machine, it generates a Contact Lost alarm.
2. During its next polling cycle, the Huawei SingleCLOUD Manager model polls CAMM, which communicates with Huawei SingleCLOUD GalaX, to gather information about the virtual machine. Because Huawei SingleCLOUD GalaX manages the virtual machines, it can provide a unique view into the possible cause of alarms generated by the virtual machine.
3. If Huawei SingleCLOUD GalaX indicates that the virtual machine is in the not-running mode, it generates a Huawei SingleCLOUD Not Running alarm.

   NOTE
   This alarm is cleared during the poll cycle after which CAMM determines the virtual machine is running again.
4. Virtual Host Manager correlates the Contact Lost alarm to the corresponding Huawei SingleCLOUD Not Running alarm created by DX NetOps Spectrum. Virtual Host Manager makes the Contact Lost alarm appear as a symptom of the Huawei SingleCLOUD Not Running alarm.

Scenario 2: Huawei SingleCLOUD Host is down

If DX NetOps Spectrum loses contact with all virtual machines running on a Huawei SingleCLOUD Host, DX NetOps Spectrum checks the status of the upstream routers and switches. Depending on their status, DX NetOps Spectrum determines the root cause as follows:

- **When all upstream devices for one or more Huawei SingleCLOUD Virtual Machines are unavailable**
  Standard DX NetOps Spectrum fault isolation techniques are used to determine the root cause:
  – A Gateway Unreachable alarm is generated on the Huawei SingleCLOUD Host when *all* upstream connected devices are down.
- **When at least one upstream device is available for every virtual machine connected to the Huawei SingleCLOUD Host**
  DX NetOps Spectrum infers that the Huawei SingleCLOUD Host is the root cause and responds as follows:
  a. All Huawei SingleCLOUD virtual machines, ports, and fanouts that are directly connected to the Huawei SingleCLOUD Host models generate the standard fault isolation alarms.
  b. Virtual Host Manager creates a Physical Host Down alarm for the Huawei SingleCLOUD Host model.
  c. All fault isolation-related alarms that are created for the impacted devices (such as virtual machines, ports, and fanouts) are correlated to the Physical Host Down alarm, making them symptoms of the Physical Host Down alarm. These symptom alarms appear in the Symptoms table on the Impact tab for the Physical Host Down alarm.

     NOTE
     For each Huawei SingleCLOUD Host model, Virtual Host Manager creates a "virtual fault domain." This domain includes the Huawei SingleCLOUD Host, CNA FIP, and Virtual Machines, plus all ports and fanouts directly connected to the Huawei SingleCLOUD Hosts. When the Huawei SingleCLOUD Host generates the Physical Host Down alarm, all standard fault isolation alarms within the domain are correlated to it. Correlating these alarms as symptoms indicates that the Physical Host Down alarm on the Huawei SingleCLOUD Host is the root cause.
  d. All impacted devices are listed in the Management Lost Impact table on the Impact tab for the Physical Host Down alarm.

     NOTE
     Devices that are suppressed do not have a corresponding alarm in the Symptoms table.

For more information about using the Impact tab for alarm information, see Determining Virtual Machines Affected by Host Outages.

e. If all upstream devices for one or more virtual machines go down, DX NetOps Spectrum can no longer reliably state that the fault lies with the Huawei SingleCLOUD Host. DX NetOps Spectrum clears the Physical Host Down alarm and applies the standard DX NetOps Spectrum fault isolation techniques.

How Fault Isolation Works when Proxy Management is Lost (Huawei SingleCLOUD)

Huawei SingleCLOUD GalaX, which is used to create your virtual network, provides DX NetOps Spectrum a unique management opportunity. DX NetOps Spectrum can use the standard methods to contact your virtual devices directly, plus DX NetOps Spectrum can simultaneously gather virtual device information from CAMM with the Huawei SingleCLOUD Device Pack, which communicates with Huawei SingleCLOUD GalaX. In this sense, CAMM is a "proxy" from which DX NetOps Spectrum gathers virtual device information. If DX NetOps Spectrum loses direct contact with a device, it generates alarms. Likewise, if CAMM loses contact with a virtual device (by way of Huawei SingleCLOUD GalaX) or if Virtual Host Manager loses contact with CAMM, Virtual Host Manager generates alarms -- proxy management alarms.

In response, DX NetOps Spectrum attempts to isolate the cause of the proxy management failure. Proxy fault isolation is similar to the standard DX NetOps Spectrum fault isolation, except that these alarms alert you to the fact that *proxy* management of a virtual device is affected. Proxy management fault isolation cannot tell you whether a virtual device is up or down. However, it is important to know when contact through the proxy is lost, because you could be missing important virtual information about a device.

The type of proxy fault isolation Virtual Host Manager uses to discover the root cause depends on which devices are alarming and the type of events the devices generate. The following scenarios describe proxy fault management situations and how Virtual Host Manager isolates the networking error in your virtual network.

Scenario 1: Contact between DX NetOps Spectrum and CAMM (Huawei SingleCLOUD Manager) is lost

If DX NetOps Spectrum loses contact with or stops polling the Huawei SingleCLOUD Manager model, DX NetOps Spectrum cannot obtain updated Huawei SingleCLOUD GalaX data about all virtual models managed by that Huawei SingleCLOUD Manager. To isolate the problem, Virtual Host Manager determines the root cause as follows:

1. DX NetOps Spectrum generates Proxy Lost alarms for all virtual models managed by that Huawei SingleCLOUD Manager, including Huawei SingleCLOUD Clouds, Hosts, CNA FIPs, and Virtual Machines. DX NetOps Spectrum also generates a separate Manager Unavailable alarm on the Huawei SingleCLOUD Manager model.
2. The Huawei SingleCLOUD alarms are correlated to their corresponding Huawei SingleCLOUD Host model alarm.
3. The Huawei SingleCLOUD Cloud and Host model alarms are correlated to a Proxy Unavailable alarm for the Huawei SingleCLOUD Manager model.
4. This Proxy Unavailable alarm is then correlated to the root cause of the Huawei SingleCLOUD Manager being down. The root cause is typically an alarm generated by standard DX NetOps Spectrum fault management, such as the alarms created for the following situations:
   - Lost management of Huawei SingleCLOUD Manager (that is, a problem occurred with CAMM)
   - Machine contact is lost
   - Huawei SingleCLOUD Manager model is in maintenance mode

Scenario 2: Contact between CAMM and Huawei SingleCLOUD GalaX is lost

If CAMM is not updated within a certain amount of time, the Huawei SingleCLOUD platform data reported by CAMM may not be current. Using a heartbeat indicator and the configured poll rate, DX NetOps Spectrum can identify when a managed Huawei SingleCLOUD entity was last updated.

When a Huawei SingleCLOUD Host or Virtual Machine has not been updated within the configured amount of time, Virtual Host Manager determines that the CAMM is unable to contact Huawei SingleCLOUD GalaX. Proxy Lost alarms are generated on the Huawei SingleCLOUD Host, CNA FIP and Virtual Machine models managed by this Huawei SingleCLOUD Manager. When multiple elements have not been updated, DX NetOps Spectrum correlates these alarms

to the appropriate root cause (for example, multiple virtual machine alarms correlated to a host). In addition, when none of the virtual IPs associated with the CAMM Presenter can be contacted, a Physical Host Down alarm is generated on the Huawei SingleCLOUD CAMM Presenter model. Identifying information for the faulting CAMM Engines and the time since the last successful communication is provided in the alarm text.

Scenario 3: Contact between Huawei SingleCLOUD GalaX and Huawei SingleCLOUD Host is lost

If Huawei SingleCLOUD GalaX loses contact with one of the Huawei SingleCLOUD Hosts it is managing, the proxy data about the host and all hosted virtual devices is lost. To isolate the problem, Virtual Host Manager determines the root cause as follows:

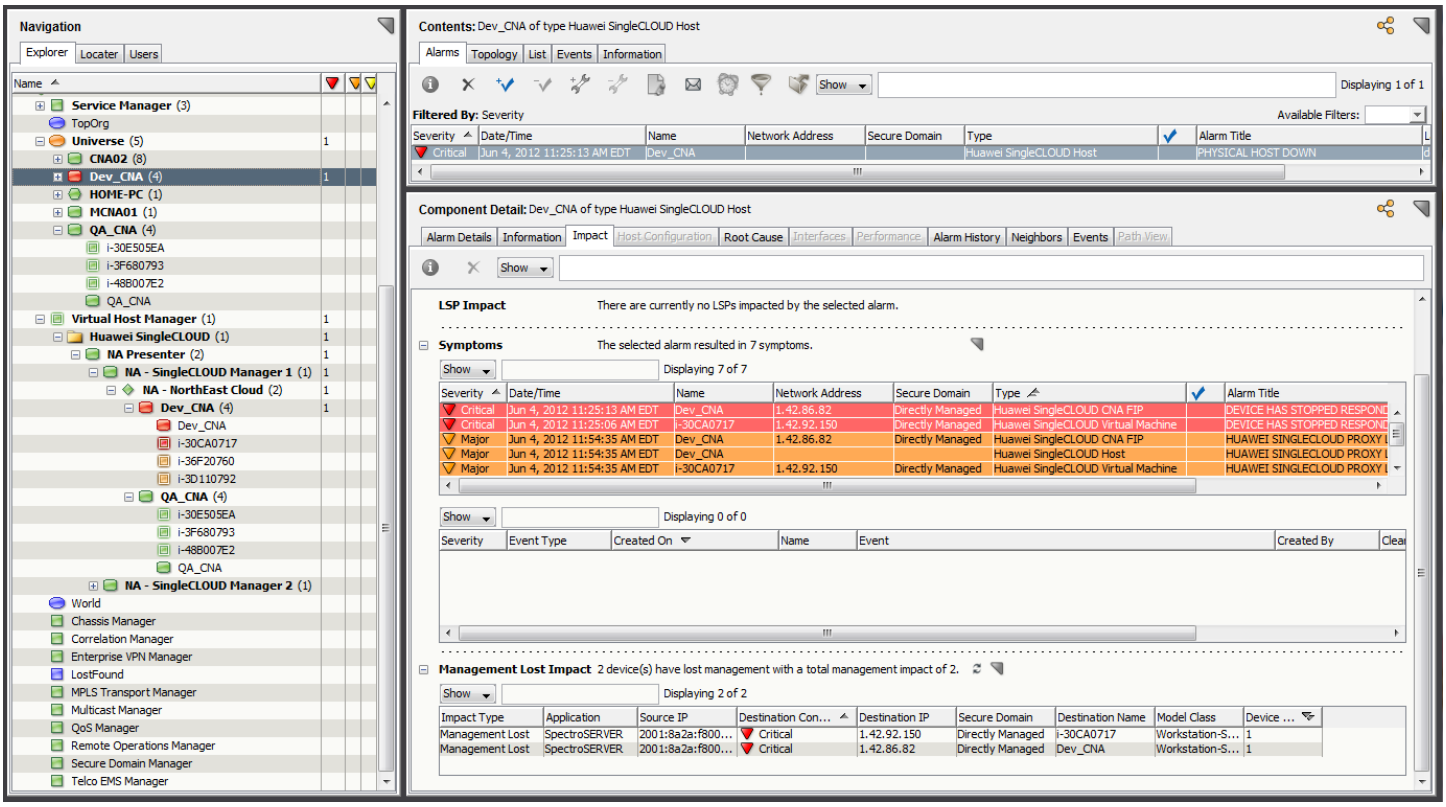
1. A Proxy Lost alarm is generated on the Huawei SingleCLOUD Host, CNA FIP, and all hosted virtual machines.
2. The CNA FIP and virtual machine alarms are correlated to the Proxy Lost alarm for the Huawei SingleCLOUD Host, making these alarms symptoms of the Huawei SingleCLOUD Host alarm. Correlating these alarms as symptoms indicates that the Huawei SingleCLOUD Host alarm is the root cause.
3. If DX NetOps Spectrum also loses contact with the Huawei SingleCLOUD Host and generates a Physical Host Down alarm, the Proxy Lost alarm generated for the Huawei SingleCLOUD Host is correlated to the Physical Host Down alarm. In this case, the Proxy Lost alarm becomes a symptom of the Physical Host Down alarm. Correlating this alarm as a symptom indicates that the Physical Host Down alarm on the Huawei SingleCLOUD Host is the root cause.

Determining Virtual Machines Affected by Host Outages

When contact with a Huawei SingleCLOUD Host is interrupted or the Huawei SingleCLOUD Host goes down, all virtual machines hosted by the Huawei SingleCLOUD Host are affected. Because the Huawei SingleCLOUD Manager cannot communicate with the Huawei SingleCLOUD Host to get usage information, you might not receive alarms for a critical virtual machine hosted on that Huawei SingleCLOUD Host.

To find out if a critical virtual machine is impacted, you can view a list of affected virtual machines on the Impact tab of the alarm, as follows:

- Symptoms subview -- displays all symptom alarms generated by the affected virtual machines
- Management Lost Impact subview -- lists the virtual machines impacted by the alarm



Troubleshooting

This section describes common symptoms or issues that can occur when using Virtual Host Manager and our recommended solution.

Duplicate Models Created After SNMP and vCenter Discovery

Symptom:

After I run the standard DX NetOps Spectrum Discovery on my virtual network and then let Virtual Host Manager run a vCenter Discovery, I get a Duplicate Models alarm for some of my virtual machines. Which model should I delete, and how do I prevent it from happening again?

Solution:

When modeling a virtual environment, a duplicate model can be created when a virtual machine does not have either VMware tools or an SNMP agent installed. The duplicate model is created as follows:

1. DX NetOps Spectrum Discovery models the virtual machine using a pingable model type, because the virtual machine does not have an SNMP agent installed. This model contains an IP address, but it does not contain a MAC address. The upstream router that usually looks up the MAC address for a device is not yet modeled, so the MAC address for the virtual machine cannot be resolved.
2. Virtual Host Manager runs a vCenter Discovery and finds the same virtual machine. Because the virtual machine does not have VMware tools installed, Discovery can identify a MAC address but cannot determine the IP address. Therefore, vCenter Discovery does not recognize it as the existing model created in step 1, so it creates a second model for the virtual machine. This model contains a MAC address but no IP address.

3. When DX NetOps Spectrum finds a model with no IP address, it performs an OS call using the model name to get the IP address. If the virtual machine name in vCenter matches the name returned from the OS, the OS passes the IP address of the virtual machine device to DX NetOps Spectrum. DX NetOps Spectrum sets the IP address in the model created by the vCenter Discovery -- this model now contains both the MAC address and IP address.
4. The Duplicate Model alarm is triggered for each model, because they both have the same IP address.

To correct the problem, delete the virtual machine device model created by the DX NetOps Spectrum Discovery (that is, the model that contains the IP address only) -- *keep the model that has both an IP and MAC address*. Otherwise, the same problem repeats the next vCenter poll cycle. If the delete affects the Virtual Host Manager hierarchy, wait one polling cycle of the vCenter server host, and the modeling is restored.

To avoid this problem when modeling your virtual environment using DX NetOps Spectrum Discovery, verify that the upstream routers for all virtual machines without VMware tools meet one of the following criteria:

- Routers have already been modeled with an SNMP-capable model type
- Routers are included in your Discovery range along with the proper SNMP credentials

By including the upstream routers, DX NetOps Spectrum attempts to resolve the physical address belonging to each host that does not have an SNMP agent.

If you model your virtual environment by modeling your VMware vCenter server by IP address and Discovery creates models without IP addresses, you must manually specify the IP for those devices before running DX NetOps Spectrum Discovery.

Customizing

This section contains information about customizing the product.

Modeling Gateway Toolkit

The DX NetOps Spectrum Modeling Gateway toolkit lets integrators import and export network topology data into and out of DX NetOps Spectrum. The toolkit includes a Document Type Definition (DTD) that defines XML elements and attributes. The toolkit also includes a resource file that defines DX NetOps Spectrum syntax and what information to import or export.

For a topology import, using the DTD elements, you can create an XML file that describes devices, ports, and connections on your network. This XML file can create new topology data in DX NetOps Spectrum, update existing data, or can destroy data that is no longer correct. Additionally, the elements and attributes that are used in the XML syntax can be expanded and customized to suit the needs of most integrations.

The toolkit also lets you use comma-delimited ASCII text files to import Frame Relay or ATM connections. You can also import this connection information using the XML functionality that was mentioned previously.

Once the network topology data exists in DX NetOps Spectrum, you can manage these devices like any other models that are created manually or by Discovery. You can view the results of the import, as well as any diagnostic information about each import.

The Modeling Gateway toolkit also lets you export topology information and configuration settings from DX NetOps Spectrum using an XML file. The information can then be imported into a specified SpectroSERVER through the Modeling Gateway.

Populating DX NetOps Spectrum with dynamic network topology information on an ongoing basis was previously a difficult task. Discovery and manual modeling are not suited to the constant updates necessary in a changing environment. Modeling connectivity using Discovery can also be a challenge with various physical infrastructures, such as those found in these environments:

- Cable MSO (Multi-Service Operator)
- ATM (Asynchronous Transfer Mode)
- Frame Relay
- Wireless Devices

NOTE

When you use the Modeling Gateway with devices located in a Secure Domain, you need to add the attribute details of the secure domain to the import to allow the Modeling Gateway to find models in question and discover the secure domain. Normally, devices can be found with just the device IP.

Example: (Device ip_dnsname="x.x.x.x" secdomain_ipname="y.y.y.y")"

WARNING
When 10.2.1 Modeling Gateway files (containing SNMPv3 profiles) get imported into 10.4.1, then SpectroSERVER crashes and stops responding. As a workaround, users must first upgrade to the 10.2.3 (or later) release to support this scenario.

Modeling Gateway Export and Import Support for World and TopOrg Views

10.3 supports an out of the box approach to export and import xml files via Modeling Gateway for **World** and **Top_Org** models. This support is beneficial for customers moving from 10.3 to any higher version.

NOTE

By default the 'RootContainderToExport' value is '**Universe**'.

To export models under **World** View:

1. In the $SPECROOT/SS-Tools\modelinggatewayresource.xml file, change the RootContainerToExport value to '**World**'.

To export models under **TopOrg** View:

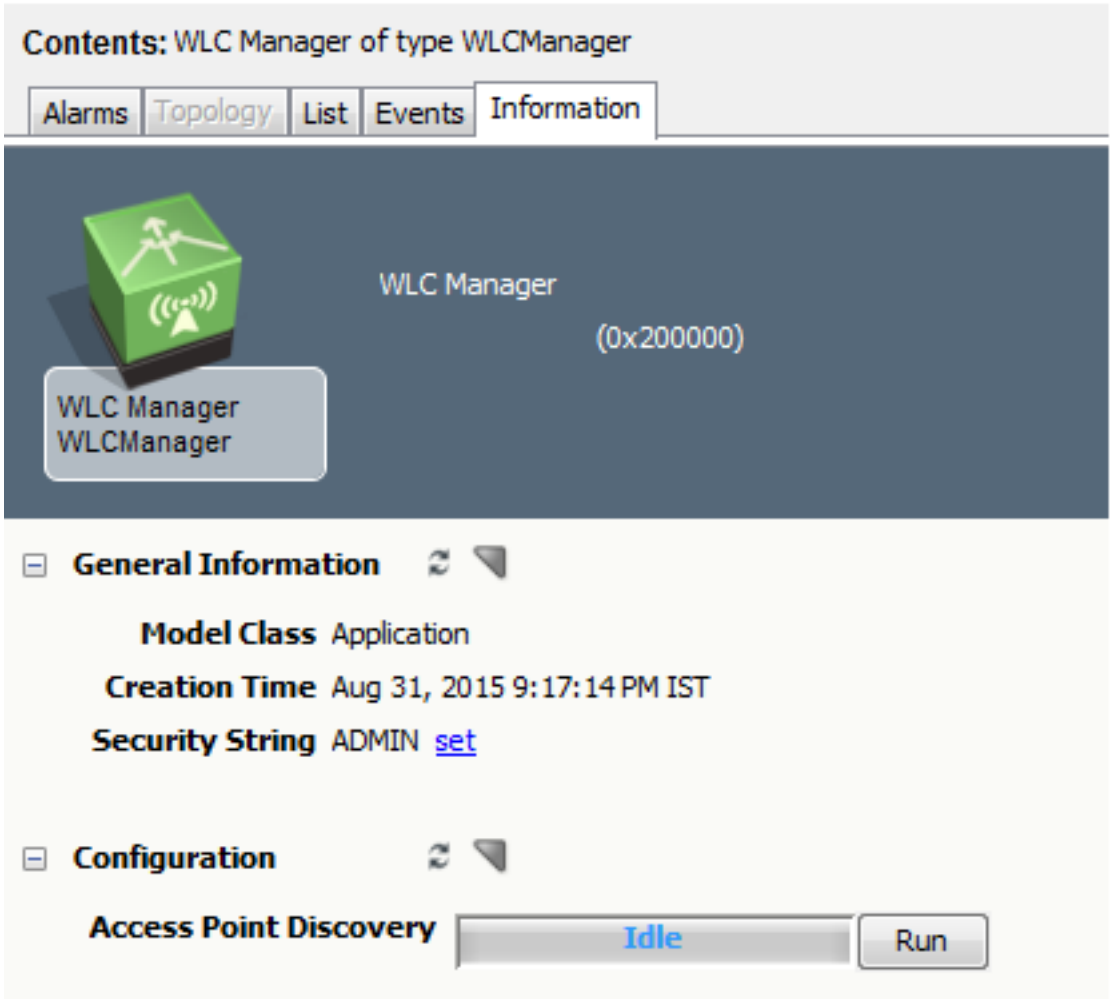
1. In the $SPECROOT/SS-Tools\modelinggatewayresource.xml file, change the RootContainerToExport value to '**Top_Org**'.

Modeling Gateway Support for Wireless Devices

From 10.1, you can import and export topology data and configuration settings for wireless devices from and to DX NetOps Spectrum. However, AccessPoint devices are not supported by the Modeling Gateway Toolkit.

To discover and model AccessPoint devices, follow these steps:

1. Import the relevant comma-delimited ASCII text/ XML file, using the **Modeling Gateway Toolkit**.
2. Navigate to **Explorer View > WLC Manager > Information Tab > Configuration** sub-view **> Access Point Discovery:** and then click **Run.**



All existing access points **(APs)** will be created under their corresponding **WLC Device**(s).

DX NetOps Spectrum Modeling Gateway is an effective solution for these problems.

Modeling Gateway Prerequisites

Before you use the DX NetOps Spectrum Modeling Gateway toolkit, make sure that you:

- Have had significant exposure to DX NetOps Spectrum.
- Read SpectroSERVER and DX NetOps Spectrum Databases Overview .
- Have a working knowledge of XML.
- Understand the concept of a Document Type Definition (DTD).
- Have a detailed understanding of the network topology you are importing.
- Can use UNIX or Windows to navigate through the file system, copy and delete files, as well as, create and edit text files.

Import and Export Architecture

Import Architecture

For an import, during the import integration process you take data from the third-party database and create an input file. Depending on the content, this input file can be an XML file or a comma-delimited ASCII file. The XML input file gives you the widest range of import options and is the main focus of this section. The comma-delimited file lets you create connections for Frame Relay and ATM circuits.

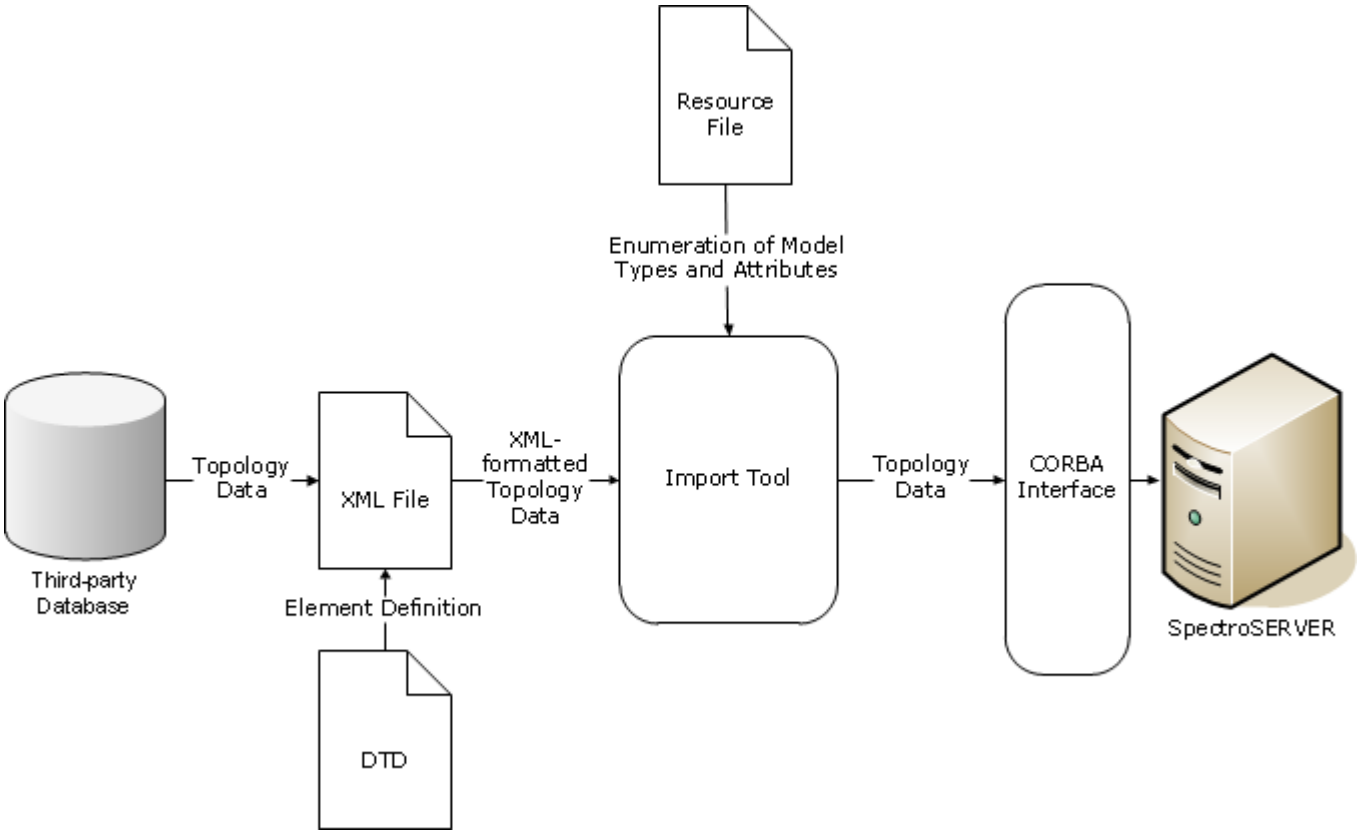
When creating an XML input file, work with the provided Document Type Definition (DTD) file and the .modelinggatewayresource.xml file. The DTD defines the XML elements, attributes, and their associated syntax rules. The .modelinggatewayresource.xml file shows which DX NetOps Spectrum model types and attributes are available for use. This file relates the DX NetOps Spectrum model type names and attribute names with the unique hexadecimal identifier that DX NetOps Spectrum uses for that model type or attribute. The .modelinggatewayresource.xml file can be customized to suit the needs of your specific integration.

Once you create the first input file, it can act as a template for multiple data sets representing the same type of input. For example, you can create an XML file for importing devices and can use this file repeatedly by substituting the device-specific topology data.

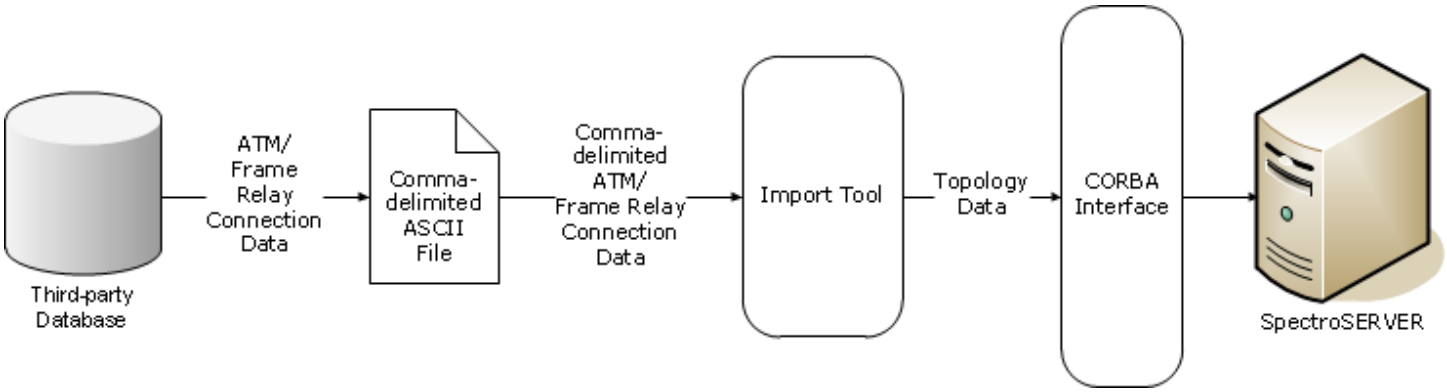
The modelinggateway tool is a command-line utility that reads the network topology information from the input file and sends the data to the SpectroSERVER database. With the data from the XML file, the import tool can create, destroy, and update connections, devices, and container models. The tool can also be used to export data from DX NetOps Spectrum.

The DX NetOps Spectrum Modeling Gateway also provides mechanisms to verify the safety and accuracy of each database import. For example, it can maintain an audit trail that includes a record of each creation, deletion, association, and update made. You can view information about the import within OneClick. DX NetOps Spectrum reports the error by generating an event when any type of critical failure occurs during the import process. All errors and their possible causes are logged in an error log file. You can also turn on a debug log which can help you locate the source of problems or inaccuracies.

The following illustration shows how the Modeling Gateway uses an XML file for importing data. The data flows from the third-party database and is formatted in the XML file using the DTD and .modelinggatewayresource.xml for syntax purposes. The import tool then interprets the XML file and sends data into DX NetOps Spectrum through the DX NetOps Spectrum CORBA interface.



The following illustration shows how the Modeling Gateway uses a comma-delimited ASCII text file to import Frame Relay and ATM connection information.



Export Architecture

For export, using the SpectroSERVER as a resource, Modeling Gateway can export topology and configuration data into an XML file. This XML file can then be integrated with a third-party tool or reimported into another SpectroSERVER. For example, the file can be reimported for a DX NetOps Spectrum partition or landscape handle change.

Spectrum modelling gateway can export Global Collections and Services without any errors even if they have any duplicate names, as the modelling gateway uses model handle to uniquely identify them.

Import Topology Data into the Product

You can import the third-party topology data into DX NetOps Spectrum using Modeling Gateway. Initially, extract the topology data from the third-party database and format the data in an input file.

To import third-party topology data into DX NetOps Spectrum using Modeling Gateway, perform these tasks:

1. Extract Topology Data.
   Extract the network topology data from the third-party database. Since each database system is different, see the documentation for the database you are working with to complete this step.
2. Format the Data in an Input File.
   To format the data for the import tool, create an XML or a comma-delimited input file.
3. (Optional) Move the modelinggateway Tool.

   NOTE
   To run the modelinggateway tool on another server, move the modelinggateway tool and all of its support files to that server. For more information, see Distributed SpectroSERVER Administration .
4. Run the modelinggateway Tool.
   Once the input file is created, use the import tool to send the data into DX NetOps Spectrum.
5. (Optional) Import Comma-delimited Files.
   You can import Frame Relay and ATM connection data from the OneClick interface without the import tool.
6. View Import Information
   Verify the progress and results of the import in OneClick.

   NOTE
   Do not use Modeling Gateway to migrate models from one version of DX NetOps Spectrum to another. The methodology that is used to identify and model an entity can differ between DX NetOps Spectrum versions. Therefore, do not use Modeling Gateway to import any XML files that are exported from a different version of DX NetOps Spectrum.

Format the Data in an Input File

Two types of input files are used to import data using the Modeling Gateway: XML files and comma-delimited files. XML input files can be used to create or destroy models and connections, and update attribute values and connectivity information. The syntax in the DTD provided with the Modeling Gateway defines the elements to be used in the XML input file. Comma-delimited files can only be used to create ATM and Frame Relay connections. The following sections outline how to create each of the types of input files.

XML Input Files

To understand the elements that create an XML input file, be familiar with the process DX NetOps Spectrum uses to model a network infrastructure. The following section provides an overview of this process and discusses how it applies to the XML elements used in an XML input file. If you are comfortable with these concepts, you can skip this section and can go to XML Input File Syntax:

Hierarchical Views

A view in DX NetOps Spectrum is a way to organize data so it can be displayed or manipulated. The hierarchical views represent ways to structure your network data. When structuring your network data in the XML file, you choose from elements that represent each of the hierarchical views. The two types of hierarchical views are Topology and Location.

Topology View

The Topology view is really an abstraction of networking components. When working with this view, you represent the physical or logical components of your network and group these components with logical connectivity in mind. You can also choose to represent connections graphically using pipes that show how devices are connected at the port or device level. In OneClick, this view appears as the Universe topology.

Location View

The Location view organizes your network data by physical location. Using this view, you can depict your network in terms of geography. You can start with your global offices. Then, go right to the wiring closet on each floor of each building in each region where your offices are located. In OneClick, this view appears as the World topology.

NOTE
For more information about the topology views available in OneClick, see the Modeling and Managing Your IT Infrastructure .

Models and Model Types

Numerous model types are predefined in DX NetOps Spectrum. When model types are instantiated in the DX NetOps Spectrum interface to represent a specific network entity, they are referred to as models. The two major categories of model types are:

- Intelligent model types
- Container model types

Intelligent model types can be instantiated to represent actual devices that operate on the network. They have IP and MAC addresses, and DX NetOps Spectrum can communicate directly with these devices using SNMP. Container model types are instantiated into models that are primarily used to group models together.

Models can be grouped based on the type of hierarchical view being used. For example, you can use the Topology view to create a LAN model that groups certain devices on a segment of your network. Or, you can use the Location view to create a Room model that groups the devices together in one room of your building.

A container model can contain other container models, intelligent models, or both, depending on the specific model type. For example, a network container model could contain an intelligent model to represent a router. The network container model could also contain a LAN container model to represent a range of IP addresses. On the other hand, a Building model can only contain container models; for example, a Floor, a Section, or a Room.

The elements in the DTD let you depict your network topology using any of the hierarchical views and their respective container model types. You can also use any of the instantiable intelligent model types. Intelligent model types are not dependent on the type of hierarchical view used.

NOTE

You can specify the model handle rather than the model type to identify a model, if desired. When specifying a model handle, Modeling Gateway ignores any other model identifiers, and it uses only the model handle to identify the model.

Not all model types that are defined in the DX NetOps Spectrum knowledge base can actually be used to create a model in OneClick. Some are used as base model types from which other model types are derived.

For more information, see SpectroSERVER and DX NetOps Spectrum Databases Overview .

Modeling Methods

Two methods are used to model devices in DX NetOps Spectrum. The first method is to use the IP address or the DNS name of the device. With this information, DX NetOps Spectrum contacts the device and creates a model using the model type that best represents the functionality of the device.

The second method is to provide a model type for the device model creation. You still must provide an IP address or a DNS name so DX NetOps Spectrum can communicate with the device. However, your chosen model type or model handle is instantiated regardless of the DX NetOps Spectrum assessment of the device functionality.

NOTE
To create a nondevice model, like a container model, a model type and model name must be provided for the import.

DX NetOps Spectrum Attributes

Each model type has a set of associated attributes. Each attribute describes the model type in some way. The attributes in an instantiated model take on values that reflect the device that the model represents and describe the current state of the model. For example, the model type Host_Sun has the attribute IPAddress. If a model of the type Host_Sun is instantiated, the value of this attribute reflects the IP address of the device that the model represents.

XML syntax also uses the term attribute. The XML attributes describe more information about an element. In DX NetOps Spectrum Modeling Gateway XML syntax, some XML attributes are used to give value to DX NetOps Spectrum attributes.

NOTE
Attributes that DX NetOps Spectrum defines are referred to as DX NetOps Spectrum attributes; the generic XML attributes are referred to simply as attributes.

XML Input File Syntax

Use the syntax rules that are defined in the DTD file when you generate the XML file. The following section provides an overview of the functionality of each of the DTD elements.

NOTE
The following explanations and examples do not cover all the attributes of each element. For a complete reference on each element and its attributes, see Document Type Definition Elements.

Root Element

The elements that are defined in the DTD exist in a hierarchical structure that parallels the network representation within DX NetOps Spectrum. The root element that must be used with each XML import file is the Import element. XML syntax rules specify that the root element is the outermost element and denotes the beginning and end of the XML file. Therefore, the Import element surrounds the rest of the XML elements that are used in your document.

Model-Oriented Elements

The model-oriented elements define physical or logical components of your network. They are container-type elements that are used to create models which define logical ways of grouping network elements. Grouping is based on the type of DX NetOps Spectrum hierarchical view they are in. Each of these container-type elements can exist in one of the specific hierarchical views.

- Topology_Container
- Location_Container
- Device
- Schedule
- Port
- Connection
- GenericView_Container
- **Topology_Container**

The Topology_Container element creates a model that groups other models according to physical or logical topology. The Topology_Container element creates container models, so use the model_type attribute or a model handle to identify the specific container you want to use. An enumeration of possible model_type values is in the DTD. A LAN is an example of a Topology_Container model_type value. You specify the name of the Topology_Container using the name attribute. The name and model_type attribute uniquely identify the created model. If you specify a model handle, the name and model_type attributes are ignored. Topology_Containers can contain other Topology_Container elements, devices, or connections. The Topology_Container models are always placed in the OneClick Topology view.

NOTE
By default, the name and model_type attributes give values to the DX NetOps Spectrum attributes Model_Name and Modeltype_Name. However, you can change which DX NetOps Spectrum attribute the name attribute gives value to by editing the .modelinggatewayresource.xml file. Whatever new DX NetOps Spectrum attribute is chosen (along with the model type) is used to identify uniquely the container. This change lets two containers have the same model name.

- **Location_Container**
  The Location_Container element groups other models according to physical or geographical location. A Building and a Room are both examples of Location_Container element model type values. The Location_Container element creates container models, so use the model_type attribute or a model handle to identify the specific container you want to use. An enumeration of possible model_type values is in the DTD. You specify the name of the Location_Container using the name attribute. The name and model_type attributes uniquely identify the created model. If you specify a model handle, the name and model_type attributes are ignored.

  NOTE
  By default, the name and model_type attributes give values to the DX NetOps Spectrum attributes Model_Name and Modeltype_Name. However, you can change which DX NetOps Spectrum attribute the name attribute gives value to by editing the .modelinggatewayresource.xml file. Whatever new DX NetOps Spectrum attribute is chosen (along with the model type) is used to identify uniquely the container. This change lets two containers have the same model name.

- **Device**
  The Device element defines a device on the network. This element is used with other elements to create, update, or destroy an instance of a device model in DX NetOps Spectrum. When working with an SNMP device, provide a valid and unique IP address or DNS name to identify uniquely the device using the ip_dnsname attribute. This unique identification lets DX NetOps Spectrum communicate with the device and select the most appropriate model type, which is based on the device functionality. Set the ip_dnsname to a valid string. If ip_dnsname is invalid or not contactable, the device model creation can fail. If model_type is provided with an invalid or noncontactable ip_dnsname, a device model is still created with the specified model type. However, the device model is not activated to provide any valid network information or status. Possible model_type values for devices are enumerated in the .modelinggatewayresource.xml file.
- **Schedule**
  The Schedule element defines when a device model is put into maintenance mode. When a device model is in maintenance mode, management traffic to the device and its components is suspended. Suspending traffic prevents DX NetOps Spectrum from generating any events or alarms on the device model while you are performing maintenance on the device.
- **Port**
  Ports are automatically created for a device when you create the device model. The Port element lets you modify some DX NetOps Spectrum port attribute values, or specify a port-level connection. You can specify different kinds of ports, including a Frame Relay or ATM circuit. To identify the port on the device, provide the values for the identifier_name and identifier_value attributes. The possible values for identifier_name are enumerated in the DTD. The identifier_value is the value of the identifier that the identifier_name attribute chooses. A Port element must always be specified as a child of a Device element.
- **Connection**

The Connection element defines a physical or logical connection between two devices, including WAN link connections, and therefore must contain two Device child elements. If a Port element is specified in the Device element, the connection is resolved on the specified port for that device. If the Device element does not specify a Port element, DX NetOps Spectrum Discovery tries to determine the ports in the connection to be resolved.

- **GenericView_Container**
  To create a container model in a Generic view, use the GenericView_Container element. Both the GenericView and GenericView_Container elements are used to create a customized view. Therefore, as the integrator, you decide when or how to use this container.

Task-Oriented Elements

The rest of the elements that are defined in the DTD are task-oriented elements. These elements and their attributes help define the type of action the input file generates. Using them, you can create new topology information, update, overwrite, or delete existing topology information. An individual input file can use zero or one of each of these elements, except for the Connection element. You can use as many Connection elements as necessary.

- Topology
- Location
- GenericView
- Connection
- Update
- Destroy

New Topology Data

The task-oriented elements define what action you would like to take with your XML file. Use the Topology and Location elements when you would like to create new network topology data in DX NetOps Spectrum. These elements define the hierarchical view where you would like to create the data. You can then use the corresponding model elements as child elements to create models for the network entities. To customize a view for your specific integration needs, use the GenericView element.

Create a Location View

You can create your topology information in the Location view for viewing in the World topology in OneClick. To create this information in the Location view, construct your XML file using the Location element inside the Import root element. The Location element can contain Location_Container elements to create a specific container model, or Device elements to create device models.

Example: Site Container in Location View

The following example creates a Site container in the Location view and a device within that container.

```
<?xml version = "1.0" standalone = "no" ?>
<!DOCTYPE Import SYSTEM ".modelinggateway.dtd">
<Import>
    <Location>
        <Location_Container  model_type = "Site" name = "My_Town" >
            <Device ip_dnsname= "10.253.9.18"
                community_string="public"/>
        </Location_Container>
    </Location>
</Import>
```

The Import element is the root element and is always contained in the input file.

The Location element indicates that you are creating models in the Location view.

The Location_Container element creates a container model. This model is a logical component rather than a physical component of the network. Therefore, DX NetOps Spectrum cannot contact it and cannot define the model type using an IP address or a DNS name. To indicate the type of container model to create, provide a value for the model_type attribute and name attribute. Possible model_type attribute values are listed in the DTD and in the following section: Location_Container. The name attribute is required and must specify a unique name for the model.

> **NOTE**
> When you specify a model handle, the model handle is used to identify the container model. Therefore the name and model_type attributes are ignored if provided.

The Device element creates a model inside the Site Location_Container model. The ip_dnsname attribute is a required attribute for the Device element. If a device can be contacted, DX NetOps Spectrum uses the IP Address or the DNS name to find the device.

For a complete reference of these elements and their possible attributes, see Document Type Definition Elements.

Create a Topology View

You can import your network data into the Topology view for viewing in the Universe topology in OneClick. To import into the Topology view, construct your XML file using the Topology element inside the Import root element. The Topology element can contain:

- Topology_Container elements to create a specific type of container model.
- Device elements to create a specific type of device model.
- Connection elements to create connections between two devices.

Using the hierarchy and syntax rules that are outlined in the DTD, you can accurately express the physical and logical connectivity of your network.

Example: LAN Container in Topology View

This example creates a LAN container in the Topology view and a device within that container.

```
<?xml version = "1.0" standalone = "no" ?>
<!DOCTYPE Import SYSTEM ".modelinggateway.dtd">
<Import>
    <Topology>
        <Topology_Container  model_type = "Lan"
            name = "Sample_LAN" Security_String = "public"
            subnet_address= "10.253.9.0" subnet_mask = "255.255.255.0">
            <Device ip_dnsname= "10.253.9.18"
                community_string="public"/>
        </Topology_Container>
    </Topology>
</Import>
```

The Import element is the root element and is always contained in the input file.

The Topology element indicates that you are going to create models in the Topology view.

The Topology_Container element creates a container model. This model is a logical component rather than a physical component of the network. Therefore, DX NetOps Spectrum cannot contact it and cannot define the model type using an IP address or a DNS name. To indicate the type of container model to create, provide a value for the model_type attribute and name attribute. The possible model_type attribute values are listed in the DTD and in the following section: Topology_Container. The name attribute is required and must specify a unique name for the model. The other attributes that are specified are optional.

NOTE

When you specify a model handle, the model handle is used to identify the container model. Therefore the name and model_type attributes is ignored, if provided.

The Device element creates a model inside the LAN Topology_Container model. The ip_dnsname attribute is a required attribute for the Device element. If DX NetOps Spectrum can contact the device, the IP Address or the DNS name is used to find the device. When DX NetOps Spectrum locates the device, it determines the appropriate model type to use to create the model.

Represent the Same Device in Multiple Views

You can create an XML file that represents a device in multiple views. To create this file, we recommended that each Device element you use to create a model of this device has identical attributes and attribute values. If they are not identical, the import tool attempts to merge the attributes and values of these Device elements to create a set of consistent attributes and values. Sometimes an attribute is specified in each of these Device elements, but different values are used. In this case, the value for the last Device element that is listed in the XML file overrides all previous values for that attribute in the other Device elements that are used to create a model of that device.

Remember that some attributes have default values. For example, the default value of the attribute community_string is public. Therefore, when you specify a Device element attribute and value to represent device A, we recommended that you specify it in any other Device element that represents device A. Doing so _helps ensure that a default value for that attribute is not used to override the previously specified value.

Example: Create a Device in Both the Topology and Location Views

In the following example, device 10.253.9.16 is created in the Topology and Location views. You can see that the attributes and values that are used to describe the device is the same for both views.

```
<?xml version = "1.0" standalone = "no" ?>
<!DOCTYPE Import SYSTEM ".modelinggateway.dtd">
<Import>
<!-- Topology View import -->
 <Topology discover_connections="false" complete_topology="false">
  <Device ip_dnsname="10.253.9.16" community_string="zippo" />
 </Topology>
<!-- Location View import -->
 <Location complete_topology="true">
  <Location_Container model_type="Site" name="Durham">
   <Device ip_dnsname="10.253.9.16" community_string="zippo"/>
  </Location_Container>
 </Location>
</Import>
```

Create Connections Using Discovery

A DX NetOps Spectrum connection represents a physical or logical link between two devices. You can create connections two different ways in the XML input file. The first method employs DX NetOps Spectrum Discovery, using the discover_connections attribute of the Topology element. Discovery runs on the newly created device models when the discover_connections attribute is set to true. Then, Discovery establishes connectivity for these devices.

NOTE

For more information about Discovery, see the Modeling and Managing Your IT Infrastructure .

Example: Using Discovery to Create Connections

```
<?xml version = "1.0" standalone = "no" ?>
<!DOCTYPE Import SYSTEM ".modelinggateway.dtd">
<Import>
    <Topology >
        <Topology_Container  model_type = "Lan" name = "Sample_LAN" discover_connections= "true"
            Security_String = "public" subnet_address= "10.253.9.0"
            subnet_mask = "255.255.255.0" >
            <Device ip_dnsname= "10.253.9.18"
                community_string="public"/>
            <Device ip_dnsname= "10.253.9.20"
                community string="public"/>
        </Topology_Container>
    </Topology>
</Import>
```

Create Connections Using the Connection Element

The second way to create connections is to use the Connection element, which connects devices that are already created. Connections can be specified between two ports, between a device and a port, or between two devices.

Specifying the connectivity between devices lets DX NetOps Spectrum isolate faults to the device, but specifying port-level connections is preferred. The port-level connections are a finer grade of connectivity, allowing DX NetOps Spectrum to resolve the connections and analyze faults at the port level. DX NetOps Spectrum automatically attempts to determine the ports when you specify a connection between two devices but you do not specify one or both ports that are used in the connection. If this process is successful, DX NetOps Spectrum resolves the connection to the port level.

DX NetOps Spectrum generates an error indicating a connection failure when both of these conditions apply:

- DX NetOps Spectrum is unable to determine both ports that are used in the connection.
- At least one of these devices is a manageable device.

The error is written to the error log file.

If DX NetOps Spectrum can determine only one of the ports, then the connection is resolved only to the port level on one side of the connection. The other side remains resolved to the device level.

If both devices are unmanageable devices, DX NetOps Spectrum establishes the connection at the device level.

NOTE

When you use the Modeling Gateway with devices located in a Secure Domain, you need to add the attribute details of the secure domain to the import to allow the Modeling Gateway to find models in question and discover the secure domain. Normally, devices can be found with just the device IP.

Example: (Device ip_dnsname="x.x.x.x" secdomain_ipname="y.y.y.y")"

Example: Create a Connection between Existing Ports

The following example creates a connection between two existing ports, each port belonging to a different device. The Connection element identifies both the Port and Device elements to be linked. The connection is resolved at the port level for both devices because a Port element is specified within each Device element.

```
<?xml version = "1.0" standalone = "no" ?>
<!DOCTYPE Import SYSTEM ".modelinggateway.dtd">
<Import>
    <Connection>
        <Device ip_dnsname= "172.19.57.93">
            <Port identifier_name = "frCircuitTableInstance"
                identifier_value="4.161"/>
```

```
        </Device>
        <Device ip_dnsname = "192.168.125.161">
            <Port identifier_name= "frCircuitTableInstance"
                identifier_value= "2.861"/>
        </Device>
    </Connection>
</Import>
```

The previous example specifies a connection between the DLCI ports. Because the value of the identifier_name attribute is frCircuitTableInstance, the port is identified using the OID instance value from the frCircuitTable object in the MIB. The OID instance value is specified using the identifier_value attribute.

The Connection element is contained within a Topology element or a Topology_Container element to indicate the hierarchical placement of the devices. This case does not change the results of the input file.

WARNING
Modeling Gateway does not report an error when you attempt to import an XML file that contains multiple Connection elements using the same Port element. A single port cannot have multiple connections. If the same port is specified in multiple Connection elements, the last Connection element in the XML file overrides all the previous Connection elements specifying that port.

Create WA_Link Connections

To create WA_Link connections, use the following syntax:

```
<Connection>
    <Device ip_dnsname=10.253.9.18/>
    <Device ip_dnsmame=10.253.9.100 model_type="WA_Link">
</Connection>
```

Modeling Gateway automatically creates a WA_Segment. The link is created between the segment and the device or devices. To specify a connection between a second device and the link, add a second connection tag to the import file.

Synchronize Information Between the Product and the Third-Party Database

The Topology, Location, Topology_Container, and Location_Container elements have an attribute named complete_topology. Setting the value of this attribute to true indicates that the XML file defines all the models and connections that DX NetOps Spectrum must know about. When the XML file is imported into DX NetOps Spectrum, any models in that DX NetOps Spectrum view that are not represented in the XML file are sent to the Lost and Found. If there are subcontainers in the view, DX NetOps Spectrum refers to the value of the complete_topology attribute that is set in the element specifying the subcontainer. If the complete_topology attribute value is not specified in the subcontainer element, the value is inherited from the parent element. Thus, if the parent element has a complete_topology setting of true and the subcontainer element does not specify a setting for complete_topology, the complete_topology value for the subcontainer is also true.

When DX NetOps Spectrum imports the XML file, models are sent to the Lost and Found under these conditions:

- The models exist either directly in the view you are importing into or in the subcontainer of that view.
- The models *do not* exist in the XML file.

This behavior is useful when synchronizing the data in your third-party database with the data in DX NetOps Spectrum.

Example: Complete_Topology Set to True

In the following example, complete_topology is set to true within the Topology element. Except for models that are specified in this input file, all existing models in the Topology view would be sent to the Lost and Found. With this sample input file, only two models are specified:

- The LAN Topology_Container
- The device at IP address 10.253.9.18

If these models did not exist, they would be created. If they existed, DX NetOps Spectrum would update their DX NetOps Spectrum attribute values using the attribute values in the input file. Any other model existing in the Topology view (except for the VNM) would be sent to the Lost and Found.

```
<?xml version = "1.0" standalone = "no" ?>
<!DOCTYPE Import SYSTEM ".modelinggateway.dtd">
<Import>
    <Topology complete_topology="true">
        <Topology_Container  model_type = "Lan" name ="Sample_LAN"
            Security_String = "public" subnet_address= "10.253.9.0"
            subnet_mask = "255.255.255.0">
            <Device ip_dnsname= "10.253.9.18"
                community_string="public"/>
        </Topology_Container>
    </Topology>
</Import>
```

If the complete_topology attribute was used in the Topology_Container element instead of the Topology element, DX NetOps Spectrum would remove only unspecified models from that Topology_Container down through the hierarchy.

Update Information

To update DX NetOps Spectrum attribute and association information for existing models, use the Update element. The Update element can enclose Container elements, Device elements, and Association elements. The value of Port attributes is updated using the appropriate Device element.

Example: Update Attributes for Two Different Models and Create an Association

The following example shows an Update input file. In this case, two attributes for two separate models are updated and an association is created between the two models.

```
<?xml version = "1.0" standalone = "no" ?>
<!DOCTYPE Import SYSTEM ".modelinggateway.dtd">
<Import>
  <Update>
    <Topology_Container model_type="Lan" name="Sample"
                        model_name = "newLAN"/>
    <Device ip_dnsname= "Test1" poll_interval= "1108"/>
    <Association relation="0x10002">
       <Left_Model> <Topology_Container name="Net"
                                        model_type="Network" /></Left_Model>
       <Right_Model> <Device ip_dnsname="172.24.94.94" /></Right_Model>
    </Association>
  </Update>
</Import>
```

The first updated attribute is the model_name attribute of the LAN container model. The model name is changed from Sample to newLAN. Note the use of the following attributes: name and model_name. Both of these attributes exist to change the DX NetOps Spectrum attribute Model_Name. To identify the container model and then specify the new name of the container model using the model_name attribute, use the name attribute with the current name as the value.

Next, the example changes the value of the poll interval for the device from Test1 to 1108. Assigning a new value to the poll_interval attribute overwrites the old value.

This example also creates an association of relation 0x10002 between container model "Net" of the Network model type and device model 172.24.94.94, as long as both models exist on the SpectroSERVER.

Destroy Information

Use the Destroy element to delete container models, device models, connections, and associations. When you destroy a device, all port and application models that are associated with the device are also destroyed.

Example: Destroy a LAN Container, a Connection, and an Association

In the following example, the LAN topology container named newLAN is destroyed. All models within this container are sent to the Lost and Found, unless they are specified to be destroyed. This example also destroys a connection between two devices, Test1 and Test2, which is specified at the port level. If a Collects association exists between the container model "Net" of model type Network and device 172.24.94.94, it is destroyed.

```
<?xml version = "1.0" standalone = "no" ?>
<!DOCTYPE Import SYSTEM ".modelinggateway.dtd">
<Import>
   <Destroy>
    <Topology_Container model_type="Lan" name="newLAN"/>
    <Connection>
      <Device ip_dnsname= "Test1">
       <Port identifier_name= "ifIndex" identifier_value= "1"/>
      </Device>
      <Device ip_dnsname= "Test2">
       <Port identifier_name="ipAddress"
              identifier_value = "10.253.8.18"/>
      </Device>
    </Connection>
    <Association relation="Collects">
       <Left_Model> <Topology_Container name="Net"
                                    model_type="Network" /></Left_Model>
       <Right_Model> <Device ip_dnsname="172.24.94.94" /></Right_Model>
    </Association>
   </Destroy>
</Import>
```

> **WARNING**
> To destroy a model representing a device that has already been removed from the network, use the IP address of the device rather than the DNS name when specifying the ip_dnsname attribute of the Device in the Destroy element of an XML file. Once the device has been removed from the network, the DNS entry for that device no longer exists. And, the Modeling Gateway cannot identify the appropriate model to delete.

The .modelinggatewayresource.xml File

The Topology_Container, Location_Container, and Device elements have a model_type attribute that must have a value equal to a valid DX NetOps Spectrum model type. DX NetOps Spectrum uniquely identifies model types using a hexadecimal number. These hexadecimal values have been enumerated in the resource file .modelinggatewayresource.xml. This file pairs a text value for the model type with the unique hexadecimal identifier. The text values are then displayed in the DTD.

Many of the attributes that are defined in the DTD correspond to DX NetOps Spectrum attributes. DX NetOps Spectrum attributes are uniquely identified in DX NetOps Spectrum using a hexadecimal number. The .modelinggatewayresource.xml file does not use these hexadecimal values in the DTD or in the XML file. Instead, the file pairs the hexadecimal identifiers of the DX NetOps Spectrum attributes with more intuitive text-based names.

Both the ModelType element and the Attribute element of the .modelinggatewayresource.xml file can be customized.

NOTE
The .modelinggatewayresource.xml file is also used for exporting topology data from DX NetOps Spectrum. For more information, see Export Topology Data from DX NetOps Spectrum.

Define Character Set Encoding

The DX NetOps Spectrum XML input file is encoded with UTF-8 by default. To import special characters or foreign languages, specify the appropriate character set encoding in the XML file header, as shown in the following example.

Example: Set the Input File Character Encoding to Greek

The following example shows how you would modify the XML file to set the character encoding to Greek:

```
<?xml version="1.0" encoding="ISO-8859-7" standalone="no"?>
<!DOCTYPE Import SYSTEM ".modelinggateway.dtd">
```

View Character Set Encoding Information

Determine the character set encoding that DX NetOps Spectrum uses from the OneClick Administration Pages.

Follow these steps:

1. Click Administration in the OneClick home page.
   The Administration Pages open.
2. Click Character Set in the panel on the left.
   The Character Set Encoding page opens, displaying a list of encodings and their applicable languages.

Comma-Delimited Input Files

The Modeling Gateway can specify ATM and Frame Relay connectivity in an XML input file. The toolkit can also accept ATM and Frame Relay connectivity information from a comma-delimited ASCII text file. This file can be used to import information about connections between:

- Two ATM circuits
- Two Frame Relay circuits
- An ATM and a Frame Relay circuit

You can specify that a live pipe is created in OneClick to represent the connection. Multiple connections can be specified in the same input file.

NOTE
The device models that are involved in these connections must previously exist in DX NetOps Spectrum.

Comma-Delimited Input File Syntax

The following example shows the format that is used in the comma-delimited input file:

```
<Device_IP>, <OID>, <Device_IP>, <OID>, <CircuitName>, <CircuitID>, <Pipe>
```

- **Device_IP**
  Specifies the IP address of each device that is involved in the connection. Required.
- **OID**
  Specifies the OID instance of frCircuitTable, atmVclTable, or atmVplTable to specify the circuit link on the device.
- **CircuitName**
  (Optional) Specifies the name of the circuit involved.
- **CircuitID**

(Optional) Specifies the ID of the circuit involved.

- **Pipe**
  (Optional) Has two possible values: CREATE_PIPE or NO_CREATE_PIPE. If the value is set to CREATE_PIPE, live pipes are created between the connections specified. If the value is set to NO_CREATE_PIPE, live pipes are not created between the connections specified. If no value is specified for this parameter, a default value of CREATE_PIPE is assumed.

Example: Specified Connection between Frame Relay Circuits

The following example shows an input file that specifies the connection between two Frame Relay circuits. A live pipe is created between these two ports.

```
172.19.57.93, 4.161, 192.168.125.161, 2.161, FR_Circuit_Name, Circuit_Id_123, CREATE_PIPE
```

Run the modelinggateway Tool for Import

Import a Single File

To run the modelinggateway tool for importing a single file, use the following syntax.

Windows

```
modelinggateway.bat -vnm vnm_name -i import_file [-o outputfile] [-debug debugfile]
```

Linux

```
modelinggateway -vnm vnm_name -i import_file [-o outputfile] [-debug debugfile]
```

- **-vnm *vnm_name***
  Specifies the SpectroSERVER host name.
- **- i *import_file***

Specifies the XML file name which contains the necessary input information (that is compiled with .modelinggateway.dtd.)

> **NOTE**
>
> If you are importing from multiple files, specify the files names that are enclosed in comma-separated {} brackets. For example, refer to the syntax sample for importing multiple files.

- **-o *outputfile***

(Optional) Logs the error information to the file named in the *outputfile* parameter.

> **NOTE**
>
> If you don't specify the debug/output file names, the error information is logged to a file named *import_file*.log;where, *Import_file* is the name of the XML file.In case of multiple imports, the debug/output files are appended and you can see consolidated logs.

- **-debug *debugfile***

(Optional) Indicates that you would like to create a debugging output file during the import process. When using the -debug option, you can provide your own debug file name for output. If you do not supply a value for *debugfile*, the debug file name defaults to the *import_file* name suffixed with ".debug."

NOTE
The -debug option requires disk space on the machine where Modeling Gateway is run. For example, a large debugging output file can result when the number of models in the *import_file* is large or when the device models have large interface densities.

Import Multiple Files

The modelinggateway tool now supports import from multiple XML files. You can now specify any number of import files.

To run the modelinggateway tool for importing multiple files, use the following syntax:

```
modelinggateway -vnm vnm_name [-user SS user][-i importfile1,importfile2,..]|[-o outputfile] [-debug
 debugfile]
```

For Example :
./modelinggateway.bat -vnm <vnm name> -i test1.xml, test2.xml

When you execute the above syntax, you see the following result message:
Multiple import XML files specified
Import will be done sequentially one after the other

ImportConfiguration Element

To control certain aspects of how Modeling Gateway imports data, use the ImportConfiguration element.

The ImportConfiguration element has the following syntax:

```
<ImportConfiguration
   do_not_process_pre_existing_devices_under_container_node = "false"
   import_to_primary_ss_only = "false"
   max_device_creation_threads = "50"
/>
```

- **do_not_process_pre_existing_devices_under_container_node**
  Specifies whether DX NetOps Spectrum processes devices that are found under a container element which previously exist in DX NetOps Spectrum.
  Default: false
- **import_to_primary_ss_only**
  Specifies whether Modeling Gateway connects to the secondary SpectroSERVER when the primary SpectroSERVER is down.
  Default: false
- **max_device_creation_threads**
  Specifies how many device models can be created and activated simultaneously.

  NOTE
  Setting this value to an amount higher than 50 can result in too much SNMP traffic.

  Default: 50

Import Comma-Delimited Files

The previous section described how to import input files using the modelinggateway import tool. You can also import Frame Relay and ATM connection data from the OneClick interface.

Follow these steps:

1. Click the applicable VNM model in the OneClick Console.

2. Click the Information tab in the Component Detail panel.
3. Click Logical Connection Import to expand the section.
4. Click Import, locate the comma-delimited file containing the data that you want to import into DX NetOps Spectrum, and then click Open.
   OneClick imports the data. The Import Results dialog appears, providing you with information about the success of the import.

View Import Information

The DX NetOps Spectrum Modeling Gateway provides mechanisms to help ensure the safety and accuracy of each database import. DX NetOps Spectrum Modeling Gateway maintains an audit trail that includes a record of each creation, deletion, association, and update made. You can view data about the import within OneClick and you can also track information about import problems in the error and debug logs.

View Modeling Gateway Results in OneClick

You can verify the results of the Modeling Gateway import from the VNM model, Information tab in the OneClick Console.

Follow these steps:

1. Click the applicable VNM model in the OneClick Console.
2. Click the Information tab in the Component Detail panel.
3. Click Modeling Gateway to expand the section.
4. Review the table in the Modeling Gateway section for information about recent imports. The table contains the following information:
   - **Import File**
     Displays the name of the import file.
   - **Log File**
     Displays the name of the log file.
   - **Start Time**
     Indicates when the topology import process began.
   - **End Time**
     Indicates when the topology import process completed.
   - **Progress**
     The progress field shows the status of a topology import that has not yet finished. The possible values for this field are:
     - Initializing
     - Identifying Models
     - Creating Models
     - Activating Models
     - Mapping Connectivity
     - Placing Models
     - Creating Connections
     - Updating Models
     - Destroying Models
     - Complete
     - Disconnected
   - **Errors**

Displays the number of errors that are generated during the import process.
- **Models Created**
  Displays the number of models that are created during the import process.
- **Models Destroyed**
  Displays the number of models that are eliminated during the import process.
- **Models Updated**
  Displays the number of models that are updated during the import process.
- **Connections Created**
  Displays the number of connections that are created during the import process.
- **Connections Removed**
  Displays the number of connections that are removed during the import process.

5. Click the Max Records set link to modify the number of import files that are listed in the table.
6. Click any of the table column headers to sort the data as needed.
7. Enter text in the Filter field to restrict the import data to specific criteria.
8. Click Update to check the status of an import as it is processing.
   The screen refreshes and displays the most recent import information available.

Error Log

All errors and their possible causes are logged in an error log file. By default, the import tool creates an error log named <*nameofimportfile*>.log where <*nameofimportfile*> is the name of your import file. You can also specify a particular name for your log file using the syntax that is specified in the section on the import tool. When the import is complete, the log file appears in the SS-Tools directory. The log file records the number of successful creations, deletions, and updates of models and connections. The log file also records each single failure that occurred during the importing process.

Export Topology Data

Modeling Gateway supports exporting topology information and configuration settings from a SpectroSERVER. The information is exported into an XML-formatted file, which can then be imported into a specified SpectroSERVER using the Modeling Gateway.

The following types of information are exported by default:

- Device elements, with configuration attributes.
- Port elements, with configuration attributes.
- Container elements, with configuration attributes.
- Connections (resolved and unresolved, WA_Link connections).
- Universe topology hierarchy.
- Layout for each view in the Universe topology, including annotations and zoom information but excluding background images.
- User models and the entire user scheme such as user-related relations, attributes, and models like LicenseRole, AccessGroup, PrivilegeRole, and UserGroup.
- Discovery configurations.
- Service Management schemes and attributes.
- Static and dynamic global collections including all the models in each global collection, all dynamic collection criteria, zoomed list, grouped list, and topology layout.

WARNING
Modeling Gateway partitions a SpectroSERVER database across two or more SpectroSERVERs, or to change the landscape handle of a single SpectroSERVER database. To configure behaviors and set up modeling for data that is not currently supported in the export capabilities, some manual work is required after the exported

data is imported. Examples of types of data which are not exported includes, but are not limited to, Service Performance Manager (SPM) tests, NCM configurations, and Events.

Configure Export Settings

You can control what is exported by modifying the ExportConfiguration element in the .modelinggatewayresource.xml file. By default, all topology and modeling information under the Universe container is exported. You can modify the RootContainerToExport element to specify a different root container from which to export. All the contents of the root container and each of its subcontainers are exported.

NOTE

You do not need to use the DTD for exporting data; the DTD is used only for importing data.

The attributes that are exported for devices, containers, and ports are defined in the following elements respectively: DeviceExportAttributes, ContainerExportAttributes, and PortExportAttributes. Add and subtract attributes from these elements as needed.

The SpectrumConfigurationExport element controls what types of DX NetOps Spectrum configuration data are exported when the export_spectrum_settings flag in the ExportConfiguration element is set to true.

For example, the following element controls the attributes that are exported for the LostFound model:

```
<SpectrumConfigurationExport model_type="LostFound" >
    <Automatic_Model_Destruction attribute_id="0x11de1" />
    <Model_Destruction_Interval_Hours attribute_id="0x11de3" />
    <Model_Destruction_Interval_Minutes attribute_id="0x11de4" />
</SpectrumConfigurationExport>
```

NOTE

The modeling information of the devices which are in the "LostFound" folder is not exported.

ExportConfiguration Element

To control export behavior, use the ExportConfiguration element in the .modelinggatewayresource.xml file.

The ExportConfiguration element has the following syntax:

```
<ExportConfiguration
   export_devices           = "true"
   export_containers        = "true"
   export_port_attributes   = "true"
   export_links             = "true"
   export_topology_layout   = "true"
   export_annotation        = "true"
   export_WA_Link_models    = "true"
   export_spectrum_settings = "true"
   export_user_models       = "true"
   export_service_modeling  = "true"
   export_schedules         = "true"
   export_global_collections = "true"
   export_discovery_configs = "true"
```

```
    export_from_primary_ss_only = "false"
    export_policy_manager = "true"
 />
```

- **export_devices**
  Exports device models.
- **export_containers**
  Exports container models.
- **export_port_attributes**
  Exports port attributes.
- **export_links**
  Exports device links.
- **export_topology_layout**
  Exports device and container models' x,y coordinates in the topology.
- **export_annotation**
  Exports the annotations and model group information.
- **export_WA_Link_models**
  Exports WA_Link models. If you decide not to export WA_Link models, they are treated as transparent. Wide area links between two device models are exported as a direct link.
- **export_spectrum_settings**
  Exports DX NetOps Spectrum settings such as the settings for fault isolation, Discovery, and VNM control.
- **export_user_models**
  Exports user models, user licenses, user privileges, user preferences, and all other user-related relations attributes and models.
- **export_servicemodeling**
  Exports the service management schemes and attributes.

  NOTE
  For more information about Service Manager, see Service Manager .
- **export_schedules**
  Exports the schedules.
- **export_global_collections**
  Exports static and dynamic global collections including all the models in each global collection, all dynamic collection criteria, zoomed list, grouped list, and topology layout.
- **export_discovery_configs**
  Exports the Discovery configurations.
- **export_from_primary_ss_only**
  Specifies whether Modeling Gateway connects to the secondary SpectroSERVER when the primary SpectroSERVER is down.
- **export_policy_manager**
  Exports the Policy Manager policies. All related models, policies, rules, permissions, and templates are included.

Example:

The following example exports everything except for port attribute information. This example also tells Modeling Gateway *not* to connect to the secondary SpectroSERVER when the primary is down.

```
<ExportConfiguration
   export_devices           = "true"
   export_containers        = "true"
   export_port_attributes   = "false"
   export_links             = "true"
   export_topology_layout   = "true"
```

```
    export_annotation        = "true"
    export_WA_Link_models    = "true"
    export_spectrum_settings = "true"
    export_user_models       = "true"
    export_service_modeling  = "true"
    export_schedules         = "true"
    export_global_collections = "true"
    export_discovery_configs = "true"
    export_from_primary_ss_only = "true"
    export_policy_manager = "true"

/>
```

The modelinggateway Tool for Export

The Modeling Gateway command-line tool, 'modelinggateway,' (modelinggateway.bat on Windows) is located in the SS-Tools directory. Its syntax for export is:

Windows

```
modelinggateway.bat -vnm vnm_name [-cmdb] -e export_file [-o outputfile] [-debug debugfile]
```

Linux

```
modelinggateway -vnm vnm_name [-cmdb] -e export_file [-o outputfile] [-debug debugfile]
```

- **-vnm *vnm_name***
  Specifies the name of the SpectroSERVER host.
- **-cmdb**
  (Optional) Exports the contents of your SpectroSERVER in a format that can be used when integrating DX NetOps Spectrum with CA CMDB. For more information on implementing this integration, contact CA Support.
- **-e *export_file***
  Exports DX NetOps Spectrum topology data.
- **-o *outputfile***
  (Optional) Logs the error information to the file named in the *outputfile* parameter. If this option is not used, the error information is logged to a file named *export_file*.log. *Export_file* is the name of the XML file.
- **-debug *debugfile***
  (Optional) Indicates that you would like to create a debugging output file during the export process. When using the -debug option, you can provide your own debug file name for output. If you do not supply a value for *debugfile*, the debug file name defaults to the *export_file* name suffixed with .debug.
  Note: The -debug option requires disk space on the machine where Modeling Gateway is run. The number of models in the *export_file* affects the size of the debugging output file: the greater the number of models in the database, the larger the debug file produced.

  NOTE
  To run the modelinggateway tool on another server, move the modelinggateway tool and all of its support files to that server. For more information, see the Distributed SpectroSERVER Administration .

Export DX NetOps Spectrum Topology Data

Export DX NetOps Spectrum topology data using the modelinggateway tool.

Follow these steps:

To export DX NetOps Spectrum topology data, use the -e flag. For example, running the following command exports the data from the SpectroSERVER on NOC1_Spectrum into a Modeling Gateway formatted xml file named NOC1_data.xml:

```
modelinggateway -vnm NOC1_Spectrum -e NOC1_data.xml
```

Import Modeling Gateway XML file

You can import the data from a Modeling Gateway formatted XML file into DX NetOps Spectrum.

Follow these steps:

To import from a Modeling Gateway formatted XML file, use the -i flag. For example, running the following command imports the data from NOC1_data.xml into the SpectroSERVER at NOC2_Spectrum.

```
modelinggateway -vnm -user SS user NOC2_Spectrum -i NOC1_data.xml
```

Appendix A. Document Type Definition Elements

This section describes the functionality of each element that is defined in the Document Type Definition (DTD). This section also provides context for each one. This section *does not* describe the XML syntax that is used in the DTD. For syntax information, see an XML reference.

Association

Syntax

Parent Elements:

- Update
- Destroy

Child Elements:

- Left_Model
- Right_Model

Rules: The Association element must contain one Left_Model element and one Right_Model element.

Usage

The Association element creates or destroys associations between models.

If the Association element is used as a child of the Destroy element, the association that is specified is destroyed. If the Association element is used as a child of the Update element, the association that is specified is created.

Attributes

- **relation**
  Specifies the name or handle of the DX NetOps Spectrum relation between the Left_Model and Right_Model in this association.

Connection

Syntax

Parent Elements:

- Topology
- Topology_Container
- Update
- Destroy

Child Elements: Device

Rules: The Connection element must contain two device elements.

Usage

The Connection element specifies a connection between two devices. The Connection element must always contain two device elements and each of these Device elements can contain zero or one Port element. If a port or ports are specified, the connection is resolved at the port level. If a port or ports are not specified, Discovery is triggered to find the port or ports for the connection.

If the Connection element is used as a child of the Destroy element, the connection that is specified is destroyed. If the connection is used in any other context, the connection is created.

Attributes

- **create_pipe**
  Indicates whether a graphical representation of the specified connection is shown in OneClick.
  Default: True

Correlation

Syntax

Parent Element: Import

Child Element: Correlation_Domain

Rule: The Correlation element can contain any number of child elements.

Usage

The Correlation element represents a Correlation Manager model and is used with the DX NetOps Spectrum Service Manager. See Service Manager for usage details.

Attributes

None.

Correlation_Domain

Syntax

Parent Element: Correlation

Child Elements:

- Device
- Port
- Model_Attr
- GenericView_Container

Rule: The Correlation_Domain element can contain any number of child elements.

Usage

The Correlation_Domain element is used with the DX NetOps Spectrum Service Manager. See Service Manager for usage details.

Attributes

See Service Manager for attribute definitions.

Attribute	Data Type	Default Value	Possible Values
name *(required)*	Character	N/A	N/A

CustomerManager

Syntax

Parent Element: SM_Service_Mgt

Child Elements:

- SM_Customer
- SM_CustomerGroup

Rule: The CustomerManager element can contain any number of these child elements.

Usage

The CustomerManager element is used with the DX NetOps Spectrum Service Manager. See Service Manager for usage details.

Attributes

NOTE
See Service Manager for attribute definitions.

Attribute	Data Type	Default Value	Possible Values
name *(required)*	Character	N/A	N/A
containment_relation	Character	Groups_Customers	N/A
model_type	Character	CustomerManager	N/A

Destroy

Syntax

Parent Element: Import

Child Elements:

- Topology_Container
- Location_Container
- GenericView_Container
- Device
- Model
- Connection
- EventModel
- SM_Service
- SM_AttrMonitor
- SM_LatencyMon
- SM_ConnectMon
- SM_SLA
- SM_Guarantee
- SM_Customer
- Association

Rule: The Destroy element can contain as many of each of these child elements as necessary.

Usage

Use the Destroy element to remove container models, device models, connections, and associations. You cannot nest elements in the Destroy element to express hierarchies or destroy hierarchies. The only hierarchy that is allowed in the Destroy element is the Connection-Device-Port hierarchy, which specifies at the port level the connection to destroy. You could destroy a container model without destroying the device models or other container models that are contained inside it. In this case, the remaining models are placed in the DX NetOps Spectrum Lost and Found.

Attributes

The Destroy element does not have any attributes.

Device

Syntax

Parent Elements:

- Topology
- Location
- Topology_Container
- Left_Model
- Right_Model
- Location_Container
- GenericView
- GenericView_Container
- Connection
- Update
- Destroy
- SM_Service
- SM_AttrMonitor

Child Elements:

- Port
- Schedule

Rules:

- **Port:** If a Device element is contained within a Connection element, only one port element is allowed. If a Device element in contained within an Update element to update ports, more than one port element is allowed. The ports are ignored when a Device element is contained in a View, a Container, or a Destroy element.
- **Schedule:** The Device element can contain one Schedule element.

Usage

To create, destroy, or update a device model, use the Device element. The Device element lets you define the device model using either an IP address or a DNS name.

> **NOTE**
> If community_string and agent_port attributes are not provided during the device creation, DX NetOps Spectrum creates the device using the predefined SNMP credentials. These credentials are configured in the Modeling and Protocol Options section of the AutoDiscovery Control subview in the VNM model Information tab in OneClick.

Attributes

- **ip_dnsname**
  Specifies the IP address or DNS name of the device. If the device does not support SNMP communication, you can use a unique string here with model_type specified.
- **secdomain_ipname**
  (Optional) Specifies the IP address of a host running an SDConnector in the secure domain where the device is located.
  Default: 0.0.0.0
- **model_handle**
  (Optional) Specifies the model_handle to identify an existing device model.

  > **NOTE**
  > If you provide a model_handle, the value of ip_dnsname is ignored.
- **model_type**
  (Optional) The DX NetOps Spectrum model type that is used to model your device. This device model can be any intelligent model type that is defined in the .modelinggatewayresource.xml file.

  > **NOTE**
  > If you have provided a valid IP address or DNS name, you do not need to specify a value here.
- **community_string**
  (Optional) The community string of the device.

  > **NOTE**
  > If community_string is not included, DX NetOps Spectrum uses the first SNMP Community Strings value to create the device. These values are specified in the VNM model Information tab, AutoDiscovery Control subview, Modeling and Protocol Options subview in OneClick.
- **agent_port**
  (Optional) Controls the port number that is used when communicating with the SNMP agent of a device.

  > **NOTE**
  > If agent_port is not included, DX NetOps Spectrum uses the first SNMP Ports value to create the device. These values are specified in the VNM model Information tab, AutoDiscovery Control subview, Modeling and Protocol Options subview in OneClick.
- **is_managed**
  (Optional) Puts the device model in maintenance mode, when set to true.

Default: True

- **poll_interval**
  (Optional) Specifies the time interval, in seconds, that the SpectroSERVER reads all attributes of the device model that are flagged as POLLED.
- **log_ration**
  (Optional) Specifies the number of SpectroSERVER polls of a device that occur before logging the poll results in the database.
- **poll_status**
  (Optional) Lets you disable the SpectroSERVER polls of a device by setting the polling status to false.
- **model_name**
  (Optional) Specifies the name of the model.
- **DeviceType**
  (Optional) Specifies the device type of the model.

  NOTE
  See the Certifications for more information about device types.
- **reconfig**

  NOTE
  (Optional) Specifies whether Modeling Gateway sends an action to the SpectroSERVER to reconfigure the device model.
- **discover_connections**
  (Optional) Runs Discovery on any newly created device models to map automatically the model connectivity, when set to true.

EventModel

Syntax

Parent Element:

- Topology_Container
- Left_Model
- Right_Model

Child Element: None

Rule: N/A

Usage

To import the EventModel models for use with a Southbound Gateway integration, use the EventModel element. For more information about Southbound Gateway, see the Southbound Gateway Toolkit .

Attributes

- **model_name**
  Specifies the unique name of the model that is instantiated or identified.
- **unique_id**
  Specifies the identifier that is used to define uniquely the event source that this EventModel model represents. For more information, see the Southbound Gateway Toolkit .
- **model_handle**
  (Optional) Specifies the model_handle to identify an existing device model.
- **Security_String**
  (Optional) Specifies the security string for the EventModel.

Default: public

- **manager_name**
  (Optional) Specifies the name of the third-party application that is using the Southbound Gateway.
  Note: Use the default value for any application that is not listed here.
  Default: 0
  - **1**
    NetMentor
  - **2**
    SSM
  - **3**
    Omni2000

GenericView

Syntax

Parent Element: Import

Child Elements:

- GenericView_Container
- Device

Rule: The GenericView element can contain any number of these child elements.

Usage

To create a customized hierarchical view other than the Topology view and Location view, use the GenericView element. You can modify this element to meet the needs of your integration.

Attributes

- **containment_relation**
  Specifies a relation handle that defines which DX NetOps Spectrum relation defines the containment relationship within this view.
  Limits: Must be a DX NetOps Spectrum containment relation.
- **model_type**
  Specifies a model type that represents the top container model that is defined for this view. This model_type must be specified with its model handle in the .modelinggatewayresource.xml file.
- **name**
  Specifies the unique name of the instantiated container model highest in the GenericView hierarchy.
- **complete_topology**
  (Optional) Destroys any unspecified, existing container and device model in the GenericView view when set to true. Also, destroys these models in the sub-containers of that view.

GenericView_Container

Syntax

Parent Element: GenericView

Child Elements:

- GenericView_Container
- Device

Rule: The GenericView_Container element can contain any number of child elements.

Usage

To create a container model in the Generic view, use the GenericView_Container element. Both the GenericView and GenericView_Container elements are used to create a customized view. Therefore, as the integrator, you decide when or how to use this container.

Attributes

- **name**
  Specifies the name of the model that is instantiated or identified. The model_type and the name attribute are required to identify uniquely the GenericView_Container. By default, this attribute is used to set the value of the DX NetOps Spectrum model name attribute (attr id 0x1006e). However, this attribute can be changed to any other attribute in the .modelinggatewayresource.xml file. You can change the .modelinggatewayresource.xml so that the name maps to a different attribute. In this case, that new attribute (along with the model type) is used to identify the container. This behavior lets two containers have the same model name.
- **model_type**
  Specifies the DX NetOps Spectrum model type that is used to create the model. This model_type must be specified with its model handle in the .modelinggatewayresource.xml file. The model_type and the name attribute are required to identify uniquely the GenericView_Container.
- **containment_relation**
  (Optional) The name of the DX NetOps Spectrum relation that exists between the Generic_Container and the models within the container. If no value for this attribute is specified, the containment relationship of the parent model is inherited.

GlobalCollection

Syntax

Parent Element: Import

Child Elements:

- Device
- Topology_Container
- Location_Container

Usage

Represents a GlobalCollection model.

Attributes

- **name**
  Specifies a name for this global collection.
- **containment_relation**
  Specifies a relation handle that defines which DX NetOps Spectrum relation defines the containment relationship within this view.
  Default: "GlobalCollect"
- **collectionDescription**
  (Optional) Describes the global collection.
- **Security_String**
  (Optional) Specifies the security string for the global collection.

Import

Syntax

Parent Element: None

Child Elements:

- Topology
- Location
- GenericView
- Update
- Destroy
- SM_Service_Mgt
- Correlation
- GlobalCollection

Rule: The Import element can contain one of each of these child elements.

Usage

The Import element is the root element, and it must be included in each input file.

Attributes

- **model_activation_time**
  Specifies the maximum number of minutes that are allowed for each device model activation.
  Data Type: Character
  Default: 5 minutes

Left_Model

Syntax

Parent Element: Association

Child Elements:

- Device
- Port
- Topology_Container
- Location_Container
- EventModel
- Model

Rules: The Left_Model element can contain only one child element.

Usage

The Left_Model element defines the left side model in an association.

Attributes

None.

List_Value

Syntax

Parent Element: Model_Attr

Child Element: None

Rule: N/A

Usage

To specify a DX NetOps Spectrum list attribute value, use the List_Value element.

Attributes

None.

Location

Syntax

Parent Element: Import

Child Elements:

- Location_Container
- Device

Rule: The Location element can contain any number of child elements.

Usage

To specify that you would like to create models in the Location view (World topology) of OneClick, use the Location element.

Attributes

- **complete_topology**
  (Optional) When set to true, any unspecified, existing containers and device models in the Location view, or any subcontainers, are destroyed during the import.
  Default: False
  Data Type: Boolean

Location_Container

Syntax

Parent Elements:

- Location
- Location_Container
- Left_Model
- Right_Model
- GlobalCollection

Child Elements:

- Location_Container
- Device

Rule: The Location_Container element can contain any number of child elements.

Usage

To create or specify a Location_Container model that is used to group models and other location containers in the Location view, use the Location_Container element.

Attributes

- **name**
  The name of the model that is instantiated or identified. The model_type and the name attribute are required to identify uniquely the Location_Container.
  By default, this attribute is used to set the value of the DX NetOps Spectrum model name attribute (attr id 0x1006e). However, this attribute can be changed to any other attribute in the .modelinggatewayresource.xml file. You can change the .modelinggatewayresource.xml so that the name maps to a different attribute. In this case, that new attribute (along with the model type) is used to identify the container. This behavior lets two containers have the same model name.
- **model_type**
  Indicates the type of model you want to create. The model_type and the name attribute are required to identify uniquely the Location_Container. Possible values include:
  – Country
  – Region
  – Site
  – Building
  – Floor
  – Section
  – Room
- **model_handle**
  (Optional) Can be used to identify models. If you provide a model_handle, the values of name and model_type are ignored.
- **Security_String**
  (Optional) Defines the requirements for access to a model by DX NetOps Spectrum users. Each security string consists of one or more Security Community entries and is assigned to models.
- **model_name**
  (Optional) To change the name of the model, use the name and the model_name attributes. The name attribute specifies the old name and the model_name attribute specifies the new name.
- **model_modify_author**
  (Optional) Writes data to the DX NetOps Spectrum attribute mdl_modfy_athr.
- **complete_topology**
  (Optional) When set to true, any unspecified, existing containers and device models in the Location view, or any subcontainers, are destroyed during the import.
  Default: False

Model_Attr

Syntax

Parent Element: Correlation_Domain

Child Element: List_Value

Rule: The Model_Attr element can contain any number of child elements.

Usage

To specify DX NetOps Spectrum attributes whose values contain multiple lines of text or a list of values, use the Model_Attr element.

Attributes

- **attr_id**
  Indicates the DX NetOps Spectrum attribute ID of the attribute you are specifying.
  Data Type: Character

Model element

Syntax

Parent Elements:

- Left_Model
- Right_Model

Child Elements: None

Usage

To represent any DX NetOps Spectrum model, use the Model element.

Attributes

- **name**
  Specifies a name for this model.
- **model_type**
  Specifies the model type of this model.
- **model_handle**
  (Optional) Specifies a model_handle for this model. If a model_handle value is specified, the name and model_type values are ignored.

MonitorPolicy_Attr

Syntax

Parent Element:

- SM_Service
- SM_AttrMonitor

Child Element: None

Rule: N/A

Usage

The MonitorPolicy_Attr element is used with the DX NetOps Spectrum Service Manager. See Service Manager for usage details.

Attributes

None.

Port

Syntax

Parent Elements:

- Device
- Left_Model
- Right_Model
- Correlation_Domain
- SM_Service
- SM_AttrMonitor

Child Element: None

Rule: N/A

Usage

The Port element is used to specify connections at the port level or to update port attributes. When updating, the parent Device element is contained within an Update element. When specifying a connection, the parent device element is contained within a Connection element.

Attributes

- **identifier_name**
  Works with the identifier_value attribute to identify uniquely the port. The identifier_name can be any one of the MIB OID names that are listed in the Possible Values column. The portID value can be used to identify a port by its Component_OID attribute (0x1006a). If the Port element represents a Frame Relay virtual circuit, use frCircuitTableInstance. If the Port element represents an ATM virtual channel or path link, use atmVclTableInstance. The portID value can be used to identify a port by its Component_OID attribute (0x1006a). If the Port element represents a Frame Relay virtual circuit, use frCircuitTableInstance.If the Port element represents an ATM virtual channel or path link, use atmVclTableInstance. Possible values include:
  - ifIndex
  - ipAddress
  - ifPhysAddress
  - ifName
  - ifAlias
  - model_name
  - portDescription
  - portID
  - frCircuitTableInstance
  - atmVclTableInstance
  - atmVplTableInstance
- **identifier_value**
  Specifies the value of the identifier_name selection.
- **model_handle**
  (Optional) Specifies the model_handle to identify an existing model.
  Note: If you provide a model_handle, the values for identifier_name and identifier_value are ignored.
- **ip_dnsname**
  (Optional) Specifies the IP address or DNS name of the port model. If the port model does not support SNMP communication, you can use a unique string here with model_type specified.
- **model_name**
  (Optional) Specifies the name of the model you are updating.
- **circuit_id**
  (Optional) Identifies the circuit that is involved in an ATM or Frame Relay connection by ID.
- **circuit_name**
  (Optional) Identifies the circuit that is involved in an ATM or Frame Relay connection by name.
- **log_ratio**
  (Optional) Specifies the number of port model polls that occur before logging the poll results in the database.
- **poll_interval**
  (Optional) Specifies the time interval, in seconds, that the SpectroSERVER reads all attributes of the port model that is flagged as POLLED.
- **poll_status**
  (Optional) Lets an administrator disable port model polls by setting polling status to False.

Right_Model

Syntax

Parent Element: Association

Child Elements:

- Device
- Port
- Topology_Container
- Location_Container
- EventModel
- Model

Rules: The Right_Model element can contain only one child element.

Usage

The Right_Model element defines the right side model in an association.

Attributes

None.

RTM_Test

Syntax

Parent Elements:

- SM_Service
- SM_AttrMonitor

Child Element: None

Rule: N/A

Usage

The RTM_Test element is used with the DX NetOps Spectrum Service Manager. See the Service Manager section for usage details.

Attributes

See the Service Manager section for attribute definitions.

Attribute	Data Type	Default Value	Possible Values
name *(required)*	Character	N/A	N/A
model_type	Character	RTM_Test	N/A

Schedule

Syntax

Parent Elements: Device

Child Element: None

Usage

To create a maintenance mode schedule for a particular device model, use the Schedule element.

Attributes

- **name**
  Specifies the name of the schedule.
- **SCHED_Recurrence**
  Specifies how often the device model is put into maintenance mode.
  - **1**
    Always (24x7)
  - **2**
    Daily
  - **3**
    Weekly
  - **4**
    Monthly
  - **5**
    Yearly

  Default: 1
- **SCHED_Start_Hour**
  (Optional) Specifies the hour that the device goes into maintenance mode.
  Limits: 0-23
- **SCHED_Start_Minute**
  (Optional) Specifies the minute the device goes into maintenance mode.
  Limits: 0-59
- **SCHED_Start_DoW**
  (Optional) Specifies the day of the week the device goes into maintenance mode.
  - **0**
    Sunday
  - **1**
    Monday
  - **2**
    Tuesday
  - **3**
    Wednesday
  - **4**
    Thursday
  - **5**
    Friday
  - **6**
    Saturday
- **SCHED_Start_DoM**
  (Optional) Specifies the day of the month the device goes into maintenance mode.
  Limits: 1-31
- **SCHED_Start_Month**
  (Optional) Specifies the month that the device goes into maintenance mode.
  - **0**
    January
  - **1**

February
  - **2**
    March
  - **3**
    April
  - **4**
    May
  - **5**
    June
  - **6**
    July
  - **7**
    August
  - **8**
    September
  - **9**
    October
  - **10**
    November
  - **11**
    December
- **SCHED_Duration**
  (Optional) Specifies the length of time the device is in maintenance mode, which is defined in seconds.
  Default: 0
- **SCHED_Recurrence_Multiplier**
  (Optional) Specifies the number of recurrence units (days, weeks, months, years) that determine the length of time between the start of each scheduled maintenance mode.
  Default: 1
- **SCHED_Daily_Repeat_Limit**
  (Optional) Specifies the number of consecutive days to repeat the scheduled maintenance (specified by SCHED_Start_Hour and SCHED_Start_Minute) during each recurrence period. This attribute is only applicable to a weekly, monthly, or yearly recurrence.

SM_AttrMonitor

Syntax

Parent Element: SM_Service

Child Element: None

Rules: N/A

Usage

The SM_AttrMonitor element is used with the DX NetOps Spectrum Service Manager. See the Service Manager section for usage details.

Attributes

See the Service Manager section for attribute definitions.

Attribute	Data Type	Default Value	Possible Values
name *(required)*	Character	N/A	N/A
containment_relation	Character	N/A	SLMMonitors SLMWatchesContainer
AttrToWatch	Character	N/A	N/A
MonitorPolicy_ID	Character	N/A	N/A
is_managed	Boolean	N/A	True False
Generate_Service_Alarms	Boolean	N/A	True False
model_type	Character	SM_AttrMonitor	N/A

SM_Customer

Syntax

Parent Elements:

- SM_CustomerGroup
- CustomerManager

Child Element: None

Rule: N/A

Usage

The SM_Customer element is used with the DX NetOps Spectrum Service Manager. See the Service Manager section for usage details.

Attributes

See the Service Manager section for attribute definitions.

Attribute	Data Type	Default Value	Possible Values
name *(required)*	Character	N/A	N/A
containment_relation	Character	N/A	SlmAgreesTo SlmUses
Security_String	Character	N/A	N/A
CustomerID	Character	N/A	N/A
Criticality	Character	N/A	N/A
CustomerField4	Character	N/A	N/A
CustomerField5	Character	N/A	N/A
CustomerField6	Character	N/A	N/A
CustomerField7	Character	N/A	N/A
Contact_Name	Character	N/A	N/A

Contact_Title	Character	N/A	N/A
Contact_Location	Character	N/A	N/A
Email_Address	Character	N/A	N/A
Phone_Number	Character	N/A	N/A
Mobile_Phone_Number	Character	N/A	N/A
Pager_Number	Character	N/A	N/A
Fax_Number	Character	N/A	N/A
User_Defined_1	Character	N/A	N/A
User_Defined_2	Character	N/A	N/A
User_Defined_3	Character	N/A	N/A
User_Defined_4	Character	N/A	N/A
Secondary_Contact_Name	Character	N/A	N/A
Secondary_Contact_Location	Character	N/A	N/A
Secondary_Email_Address	Character	N/A	N/A
Secondary_Phone_Number	Character	N/A	N/A
Secondary_Mobile_Phone_ Number	Character	N/A	N/A
Secondary_Pager_Number	Character	N/A	N/A
Secondary_Fax_Number	Character	N/A	N/A
Secondary_User_Defined_1	Character	N/A	N/A
Secondary_User_Defined_2	Character	N/A	N/A
Secondary_User_Defined_3	Character	N/A	N/A
Secondary_User_Defined_4	Character	N/A	N/A
model_type	Character	SM_Customer	N/A

SM_CustomerGroup

Syntax

Parent Elements:

- CustomerManager
- SM_CustomerGroup

Child Elements:

- SM_CustomerGroup
- SM_Customer

Rule: The SM_CustomerGroup element can contain any number of these child elements.

Usage

The SM_CustomerGroup element is used with the DX NetOps Spectrum Service Manager. See the Service Manager section for usage details.

Attributes

See the Service Manager section for attribute definitions.

Attribute	Data Type	Default Value	Possible Values
name *(required)*	Character	NA	N/A
containment_relation	Character	Groups_Customer	N/A
model_type	Character	SM_CustomerGroup	N/A

SM_Guarantee

Syntax

Parent Element: SM_SLA

Child Element: None

Rule: N/A

Usage

The SM_Guarantee element is used with the DX NetOps Spectrum Service Manager. See the Service Manager section for usage details.

Attributes

See the Service Manager section for attribute definitions.

Attribute	Data Type	Default Value	Possible Values
name *(required)*	Character	N/A	N/A
containment_relation	Character	SlmIsMeasuredBy	N/A
is_managed	Boolean	N/A	True False
DegradedTimeViolationLevel	Character	N/A	N/A
DegradedTimeWarningLevel	Character	N/A	N/A
DownTimeViolationLevel	Character	N/A	N/A
DownTimeWarningLevel	Character	N/A	N/A
LorTimeViolationLevel	Character	N/A	N/A
LorTimeWarningLevel	Character	N/A	N/A
model_type	Character	SM_Guarantee	N/A

SM_LatencyMon

Syntax

Parent Elements:

- SM_Guarantee
- SM_AttrMonitor
- SM_Service

Child Elements:

- Topology_Container
- MonitorPolicy_Attr

Rule: The SM_LatencyMon element can contain any number of these child elements.

Usage

The SM_LatencyMon element is used with the DX NetOps Spectrum Service Manager. See the Service Manager section for usage details.

Attributes

See the Service Managersection for attribute definitions.

Attribute	Data Type	Default Value	Possible Values
name *(required)*	Character	N/A	N/A
containment_relation	Character	N/A	SlmMonitors SlmWatchesContainer
is_managed	Boolean	N/A	True False
DefaultMaxRTT	Character	N/A	N/A
DefaultMeasureInterval	Character	N/A	N/A
mode_type	Character	SM_LatencyMon	N/A

SM_Service

Syntax

Parent Elements:

- SM_Service
- SM_ServiceMgr

Child Elements:

- SM_Service
- SM_AttrMonitor

Rule: The SM_Service element can contain any number of these child elements.

Usage

The SM_Service element is used with the DX NetOps Spectrum Service Manager. See the Service Managersection for usage details.

Attributes

See the Service Manager section for attribute definitions.

Attribute	Data Type	Default Value	Possible Values
name *(required)*	Character	N/A	N/A
Criticality	Character	N/A	N/A
containment_relation	Character	N/A	SlmMonitors SlmWatchesContainer

AttrToWatch	Character	N/A	N/A
MonitorPolicy_ID	Character	N/A	N/A
is_managed	Boolean	N/A	True False
Generate_Service_Alarms	Boolean	N/A	True False
Security_String	Character	N/A	N/A
model_type	Character	SM_Service	N/A

SM_Service_Mgt

Syntax

Parent Element: Import

Child Elements:

- SM_ServiceMgr
- CustomerManager
- SM_SLA_Mgr

Rule: Only a single instance of each of these child elements can exist in SM_Service_Mgt.

Usage

The SM_Service_Mgt element is used with the DX NetOps Spectrum Service Manager. See the Service Manager section for usage details.

Attributes

See the Service Manager section for attribute definitions.

Attribute	Data Type	Default Value	Possible Values
name *(required)*	Character	Service Management	N/A
containment_relation	Character		N/A
model_type	Character		N/A

SM_ServiceMgr

Syntax

Parent Element: SM_Service_Mgt

Child Element: SM_Service

Rule: The SM_ServiceMgr element can contain any number of this child element.

Usage

The SM_ServiceMgr element is used with the DX NetOps Spectrum Service Manager. See the Service Manager section for usage details.

Attributes

See the Service Manager section for attribute definitions.

Attribute	Data Type	Default Value	Possible Values
name	Character	N/A	N/A
containment_relation	Character	SImContains	N/A
model_type	Character	SM_ServiceMgr	N/A

SM_SLA

Syntax

Parent Element: SM_SLA_Mgr

Child Element: SM_Guarantee

Rule: The SM_SLA element can contain any number of this child element.

Usage

The SM_SLA element is used with the DX NetOps Spectrum Service Manager. See the Service Manager section for usage details.

Attributes

See the Service Manager section for attribute definitions.

Attribute	Data Type	Default Value	Possible Values
name *(required)*	Character	N/A	N/A
containment_relation	Character	SImContains	N/A
is_managed	Boolean	N/A	True False
Security_String	Character	N/A	N/A
model_type	Character	SM_SLA	N/A

SM_SLA_Mgr

Syntax

Parent Element: SM_Service_Mgt

Child Element: SM_SLA

Rule: The SM_SLA_Mgr element can contain any number of this child element.

Usage

The SM_SLA_Mgr element is used with the DX NetOps Spectrum Service Manager. See the Service Manager section for usage details.

Attributes

See the Service Manager section for attribute definitions.

Attribute	Data Type	Default Value	Possible Values
name	Character		N/A
containment_relation	Character	SlmContains	N/A
model_type	Character	SM_ServiceMgr	N/A

Topology

Syntax

Parent Element: Import

Child Elements:

- Topology_Container
- Device
- Connection

Rule: The Topology element can contain any number of these child elements.

Usage

To create models in the OneClick Topology view (Universe topology), use the Topology element.

Attributes

- **complete_topology**
  Destroys any unspecified, existing container and device model in the Topology view during the import when set to true. Also, destroys any subcontainers of that view.
  Default: False
- **discover_connections**
  When set to true, Discovery runs on any newly created device models to map automatically the connectivity of the model.
  Default: False

Topology_Container

Syntax

Parent Elements:

- Topology
- Topology_Container
- SM_Service
- SM_AttrMonitor

Child Elements:

- Topology_Container
- Device
- EventModel
- Connection

Rule: The Topology_Container element can contain any number of these child elements.

Usage

To create or specify a Topology_Container model that is used to group models and other topology containers in the Topology view, use the Topology_Container element. Possible model types are listed in the following table in the model_type section.

Attributes

- **model_type**
  Indicates the type of model you want to create. The model_type and the name attribute are required to identify uniquely the Topology_Container.
  - Network
  - LAN
  - IPClassA
  - IPClassB
  - IPClassC
  - LAN_802_3
  - LAN_803_5
  - EventAdmin
  - ATM_Network
- **model_handle**

  NOTE
  (Optional) Can be used to identify existing models. If you provide a model_handle, the values of model_type and model_name are ignored.
- **Security_String**
  (Optional) Specifies the assigned DX NetOps Spectrum security level for the model.
- **subnet_address**
  (Optional) Specifies the subnet address of the device.
- **subnet_mask**
  (Optional) Specifies the mask that determines what subnet the device IP address belongs to.
- **model_name**
  (Optional) Specifies a model name for the container model.
- **trapIPAddress**
  (Optional) For the EventAdmin models only.
- **x_coordinate**
  (Optional) Specifies the x coordinate of the model in the topology.
- **y_coordinate**
  (Optional) Specifies the y coordinate of the model in the topology.
- **complete_topology**
  (Optional) When set to True, any unspecified, existing container and device model in this Topology_Container and, any subcontainers, are destroyed during the import.
  Default: False
- **discover_connections**
  (Optional) Specifies whether DX NetOps Spectrum discovers and models devices that are connected to this model.

Update

Syntax

Parent Element: Import

Child Elements:

- Topology_Container
- Location_Container
- GenericView_Container
- Connection
- Device
- Model
- EventModel
- SM_Service
- SM_AttrMonitor
- SM_LatencyMon
- SM_ConnectMon
- SM_SLA
- SM_Guarantee
- SM_Customer
- Association

Rule: The Update element can contain any number of these child elements.

Usage

To update attributes for any device, container, or port subelements, use the Update element.

> **NOTE**
> The hierarchical specifications are not allowed when using this element, except for using the Port element within the Device element.

Attributes

None.

Appendix B. Document Type Definition File

This section contains the Document Type Definition (DTD), which defines XML elements and attributes for import.

> **WARNING**
>
> Note that the following DTD file is not meant for editing, but to be used as a reference, when editing the XML file. For example, there is an attribute for model activation timeout, referenced in the DTD, that can be used to extend the model activation time, for large imports.
> This should be added to the Import tag of the XML import file, at root node as below, in order to increase the timeout to 50 mins:
> landscape_handle="0xXXX00000" model_activation_time="50">
>
> **NOTE**
> The following code might not be the latest version of this file. For the latest version of the DTD, use the actual file that shipped with your Modeling Gateway toolkit. The file is located in the SS-Tools directory and is named .modelinggateway.dtd.

Sample Document Type Definition File

```
<!-- ********************************************************** -->
<!-- The root Import element contains 0 or 1 of the Topology,     -->
<!-- Location, GenericView, Update, Destroy elements,             -->
<!-- SM_Service_Mgt, Correlation, and Global Collection.          -->
<!--                                                              -->
<!-- This element has one attribute model_activation_time which   -->
<!-- is the maximum waiting time in minutes, for each device model-->
```

```
<!-- activation. It defaults to 5 minutes.                        -->
<!-- ********************************************************** -->
<!ELEMENT Import ( ( Topology       |
            Location       |
            GenericView    |
            Update         |
            Destroy        |
            SM_Service_Mgt |
            Correlation    |
            GlobalCollection )*) >
<!ATTLIST Import model_activation_time  CDATA     "5">
<!-- ************************************************** -->
<!-- The Topology element, which is used for the Topology -->
<!-- view, contains any number of Topology_Container,     -->
<!-- Device and Connection elements.                      -->
<!--                                                      -->
<!-- This element has two attributes, complete_topology   -->
<!-- and discover_connection.                             -->
<!--                                                      -->
<!-- ************************************************** -->

<!ELEMENT Topology ((Topology_Container | Device | Connection)*) >
<!ATTLIST Topology
    complete_topology    (false | true)     #IMPLIED
    discover_connections (false | true)     #IMPLIED>
<!-- ************************************************** -->
<!-- The Topology_Container element, which is used for a  -->
<!-- topology container model, contains any number of     -->
<!-- Topology_Container, Device and Connection elements.  -->
<!--                                                      -->
<!-- "model_type" and "name" are the required attributes  -->
<!-- to uniquely identify Topology_Container models.      -->
<!--                                                      -->
<!-- "model_handle" can also be used to identify models.  -->
<!-- If "model_handle" is provided, the values of "name"  -->
<!-- and "model_type" will be ignored.                    -->
<!--                                                      -->
<!-- "trapIPAddress" attribute should be only used for    -->
<!-- EventAdmin models.                                   -->
<!--                                                      -->
<!-- ************************************************** -->
<!ELEMENT Topology_Container ((Topology_Container |
                   Device |
                   EventModel |
                   Connection )*) >
<!ATTLIST Topology_Container
    name          CDATA                  #REQUIRED
    model_type          ( Network      |
                Lan          |
                IPClassA     |
                IPClassB     |
                IPClassC     |
                LAN_802_3    |
```

```
                LAN_803_5    |
                EventAdmin   |
                ATM_Network  ) #REQUIRED
    model_handle          CDATA          #IMPLIED
        Security_String       CDATA          #IMPLIED
    subnet_address        CDATA          #IMPLIED
    subnet_mask           CDATA          #IMPLIED
    model_name            CDATA          #IMPLIED
    trapIPAddress         CDATA          #IMPLIED
    x_coordinate          CDATA          #IMPLIED
    y_coordinate          CDATA          #IMPLIED
    complete_topology    (false | true)  #IMPLIED
    discover_connections (false | true)  #IMPLIED >
<!-- ************************************************** -->
<!-- Location element, which is for Location view,         -->
<!-- contains any number of Location_Container and         -->
<!-- Device elements                                       -->
<!--                                                       -->
<!-- This element has attribute complete_topology.         -->
<!-- ************************************************** -->
<!ELEMENT Location ( (Location_Container  | Device )* ) >
<!ATTLIST Location
    complete_topology    (false | true)     #IMPLIED>
<!-- ************************************************** -->
<!-- The Location_Container element, which is used for    -->
<!-- Location container, may contain any number of        -->
<!-- Location_Container and Device elements.              -->
<!--                                                       -->
<!-- "model_type" and "name" are the required attributes  -->
<!-- to uniquely identify Location_Container models.      -->
<!--                                                       -->
<!-- "model_handle" can also be used to identify models.  -->
<!-- If "model_handle" is provided, the values of "name"  -->
<!-- and "model_type" will be ignored.                    -->
<!-- ************************************************** -->
<!ELEMENT Location_Container ( ( Location_Container | Device )* )>
<!ATTLIST Location_Container
    name        CDATA                   #REQUIRED
    model_type        ( Country    |
            Region     |
            Site       |
            Building   |
            Floor      |
            Section    |
            Room       )     #REQUIRED
          model_handle        CDATA            #IMPLIED
    Security_String     CDATA            #IMPLIED
    model_name          CDATA            #IMPLIED
    model_modify_author CDATA            #IMPLIED
    complete_topology   (false | true)   #IMPLIED >
<!-- ********************************************************** -->
<!-- Device element is used for device model.                    -->
<!--                                                             -->
```

```
<!-- The "ip_dnsname" is the required attribute to uniquely       -->
<!-- identify device models.                                      -->
<!--                                                              -->
<!-- model_handle" can also be used to identify device models.    -->
<!-- If "model_handle" is provided, the value of "ip_dnsname"     -->
<!-- will be ignored.                                             -->
<!--                                                              -->
<!-- CAUTION:                                                     -->
<!--                                                              -->
<!-- 1. If the attribute is_managed is set to false, the device   -->
<!--    is not contactable.  You must therefore set the model_type-->
<!--    attribute.                                                -->
<!--                                                              -->
<!-- 2. A Device can contain 0, 1 or more than 1 Port in the      -->
<!--    following situations respectively:                        -->
<!--                                                              -->
<!-- (a) If a Device is in a Container or a Destroy element,      -->
<!--     the Port element is not needed.  If there are Port       -->
<!--     elements provided, they will be ignored.                 -->
<!--                                                              -->
<!-- (b) If a Device is in a Connection element, only one Port    -->
<!--     element is allowed.                                      -->
<!--                                                              -->
<!-- (c) If a Device is contained in an Update element to update  -->
<!--     ports, than one Port elements are allowed.               -->
<!--                                                              -->
<!-- (d) If you specify discover_connections="true" on a device   -->
<!--     tag, do not also specify it on that device's parent      -->
<!--     container.  This has performance and efficiency related  -->
<!--     issues as specifying that attribute on a container       -->
<!--     causes Spectrum to discover connections on each model    -->
<!--     in that container anyway.                                -->
<!--                                                              -->
<!-- ************************************************************ -->
<!ELEMENT Device ( Port* | Schedule ) >
<!ATTLIST Device
    ip_dnsname        CDATA              #REQUIRED
    secdomain_ipname  CDATA              #IMPLIED
    model_handle      CDATA              #IMPLIED
    model_type        CDATA              #IMPLIED
    community_string  CDATA              #IMPLIED
       agent_port        CDATA              #IMPLIED
    poll_interval     CDATA              #IMPLIED
    log_ratio         CDATA              #IMPLIED
    model_name        CDATA              #IMPLIED
    DeviceType        CDATA              #IMPLIED
       x_coordinate      CDATA              #IMPLIED
       y_coordinate      CDATA              #IMPLIED
    is_managed        (true  | false)    #IMPLIED
    reconfig          (true  | false)    #IMPLIED
    poll_status       (true  | false)    #IMPLIED
    discover_connections (false | true)  #IMPLIED >
<!-- ************************************************************ -->
```

```
<!-- Port element is used for device port model.                  -->
<!--                                                              -->
<!-- "identifier_name" and "identifier_value" are the required    -->
<!-- attributes to uniquely identify port models.                 -->
<!--                                                              -->
<!-- "model_handle" can also be used to identify port models.     -->
<!-- If "model_handle" is provided, the value of identifier_name  -->
<!-- and identifier_value will be ignored.                        -->
<!-- ************************************************************ -->
<!ELEMENT Port ( Port* ) >
<!ATTLIST Port
    identifier_name ( portDescription        |
            model_name             |
            ifIndex                |
            ipAddress              |
            ifPhysAddress          |
            ifName                 |
            ifAlias                |
            portID                 |
            frCircuitTableInstance |
            atmVclTableInstance    |
            atmVplTableInstance    ) #REQUIRED
    identifier_value  CDATA                    #REQUIRED
    model_handle      CDATA                    #IMPLIED
    ip_dnsname        CDATA                    #IMPLIED
    model_type        CDATA                    #IMPLIED
    model_name        CDATA                    #IMPLIED
    circuit_id        CDATA                    #IMPLIED
    circuit_name      CDATA                    #IMPLIED
    log_ratio         CDATA                    #IMPLIED
    poll_interval     CDATA                    #IMPLIED
    poll_status       (false | true)           #IMPLIED >
<!-- ********************************************************************** -->
<!-- Represents a Schedule model                                            -->
<!--                                                                        -->
<!-- SCHED_Recurrence can have the following values                         -->
<!--                                                                        -->
<!--  1 = Always (24 x 7)                                                   -->
<!--  2 = Daily                                                             -->
<!--  3 = Weekly                                                            -->
<!--  4 = Monthly                                                           -->
<!--  5 = Yearly                                                            -->
<!--  6 = Once                                                              -->
<!--                                                                        -->
<!--  SCHED_Start_Hour: value range 0-23                                    -->
<!--  SCHED_Start_Minute: value range 0-59                                  -->
<!--  SCHED_Start_DoW: Day of the week (range 0-6 where Sunday is 0) for    -->
<!--                   Weekly recurrence                                    -->
<!--  SCHED_Start_DoM: Day of the month (range 1-31) for Monthly and Yearly -->
<!--                   recurrence                                           -->
<!--  SCHED_Start_Month: range 0-11 where January is 0 for Yearly           -->
<!--                     recurrence                                         -->
<!--  SCHED_Duration: active period duration in seconds. May be 0 (default) -->
```

```
<!--  SCHED_Recurrence_Multiplier: number of recurrence units  that         -->
<!--                               determine the length of time between the -->
<!--                               start of each active period.             -->
<!--                               Default is 1.                            -->
<!--  SCHED_Daily_Repeat_Limit: number of consecutive days to repeat a      -->
<!--                            daily schedule (specified by               -->
<!--                            SCHED_Start_Hour and SCHED_Start_Minute)   -->
<!--                            at the start of each recurrence period.    -->
<!--                            Only applicable tp Weekly, Monthly or      -->
<!--                            Yearly recurrence.                         -->
<!-- SCHED_DayBitMask: The days of the week on which a WEEKLY Schedule     -->
<!--                   should be ACTIVE.  Values are:                      -->
<!--                   Sunday = 1,                                         -->
<!--                   Monday = 2,                                         -->
<!--                   Tuesday = 4,                                        -->
<!--                   Wednesday = 8,                                      -->
<!--                   Thursday = 16,                                      -->
<!--                   Friday = 32,                                        -->
<!--                   Saturday = 64                                       -->
<!--                   e.g if we want Mon, Wed and Fri the value would be  -->
<!--                   2+8+32=42                                           -->
<!--  SCHED_Start_MoY,                                                     -->
<!--  SCHED_START_YEAR,                                                    -->
<!--  SCHED_START_DAY: Used in conjunction with SCHED_START_MONTH to       -->
<!--                   indicate that a schedule should be active on some   -->
<!--                   day on the future.  Both SCHED_START_YEAR and       -->
<!--                   SCHED_START_DAY must be non-zero.                   -->
<!--                   Otherwise the schedule behaves normally and goes    -->
<!--                   active as early as today.                           -->
<!--                   Note that SCHED_START_YEAR should be specified as   -->
<!--                   the number of years since 1900.                     -->
<!-- SCHED_Description: Description for this Schedule.                     -->
<!--                                                                       -->
<!-- ********************************************************************* -->
<!ELEMENT Schedule ( #PCDATA) >
<!ATTLIST Schedule
    name                        CDATA               #REQUIRED
    SCHED_Recurrence            ( 1 | 2 | 3 |
                      4 | 5 | 6 )        #REQUIRED
    SCHED_Daily_Repeat_Limit    CDATA               #REQUIRED
    SCHED_Duration              CDATA               #REQUIRED
    SCHED_Recurrence_Multiplier CDATA               #REQUIRED
    SCHED_Start_DoM             CDATA               #REQUIRED
    SCHED_Start_DoW             CDATA               #REQUIRED
    SCHED_Start_Hour            CDATA               #REQUIRED
    SCHED_Start_Minute          CDATA               #REQUIRED
    SCHED_Start_Month           CDATA               #REQUIRED
    SCHED_Start_Day             CDATA               #REQUIRED
    SCHED_DayBitMask            CDATA               #REQUIRED
    SCHED_Start_Year            CDATA               #REQUIRED
    SCHED_Start_MoY             CDATA               #REQUIRED
    SCHED_Description           CDATA               #REQUIRED
    >
```

```
<!-- ************************************************************ -->
<!-- Element used to represent EventModels.                       -->
<!--                                                              -->
<!-- The unique id must be specified for performance reasons.     -->
<!-- ************************************************************ -->
<!ELEMENT EventModel ( #PCDATA ) >
<!ATTLIST EventModel
    model_name        CDATA        #REQUIRED
    unique_id         CDATA        #REQUIRED
    model_handle      CDATA        #IMPLIED
    Security_String   CDATA        "public"
    manager_name      CDATA        "0">
<!-- ************************************************************ -->
<!-- Connection element represents device connectivity.           -->
<!-- It contains two Device elements involved in the connection.  -->
<!-- Each Device may have 0 or 1 Port element.                    -->
<!--                                                              -->
<!-- Connection has one attribute, create_pipe. Normally for ATM  -->
<!-- circuit links, create_pipe is set to false to avoid the      -->
<!-- creation of numerous pipes within the view. In this case,    -->
<!-- one can use ATM Manager to view the connections. It's up     -->
<!-- to users to decide the setting for create_pipe. By default,  -->
<!-- a pipe will be created for each connection.                  -->
<!--                                                              -->
<!-- ************************************************************ -->
<!ELEMENT Connection (Device, Device)>
<!ATTLIST Connection  create_pipe  (true | false)  "true" >
<!-- ************************************************************ -->
<!-- Update element is to update SPECTRUM model attributes and    -->
<!-- associations.                                                -->
<!--                                                              -->
<!-- The Update element is allowed to contain any number of       -->
<!-- Container, Device and Association elements to be updated.    -->
<!-- To update a port, the Port element needs to placed inside a  -->
<!-- Device element and then place the Device element into the    -->
<!-- Update element.                                              -->
<!-- ************************************************************ -->
<!ELEMENT Update  ( ( Topology_Container           |
                     Location_Container           |
                     GenericView_Container        |
                     Connection                   |
                     Device                       |
                     EventModel                   |
                     SM_Service                   |
                     SM_AttrMonitor               |
                     SM_LatencyMon                |
                     SM_ConnectMon                |
                     SM_SLA                       |
                     SM_Guarantee                 |
                     SM_Customer                  |
                     Association
                   )* ) >
<!-- ************************************************************ -->
```

```
<!-- Destroy elemen: destroy SPECTRUM models and associations.    -->
<!--                                                              -->
<!-- The Destroy element is allowed to contain any number of      -->
<!-- Container, Device and Connection and Association elements    -->
<!-- to be destroyed.                                             -->
<!-- ************************************************************ -->
<!ELEMENT Destroy ( ( Topology_Container            |
                      Location_Container            |
                      GenericView_Container         |
                      Device                        |
                      Connection                    |
                      EventModel                    |
                      SM_Service                    |
                      SM_AttrMonitor                |
                      SM_LatencyMon                 |
                      SM_ConnectMon                 |
                      SM_SLA                        |
                      SM_Guarantee                  |
                      SM_Customer                   |
                      Association
                    )* ) >

<!-- ************************************************************ -->
<!-- Association element defines SPECTRUM association between two -->
<!-- models for creation or destroy.                              -->
<!--                                                              -->
<!-- Association element contains one Left_Model and one          -->
<!-- Right_Model element.                                         -->
<!-- ************************************************************ -->
<!ELEMENT Association ((Left_Model | Right_Model)*) >
<!ATTLIST Association relation CDATA #REQUIRED >

<!-- ************************************************************ -->
<!-- Left_Model element defines left side model in a Spectrum     -->
<!-- association.                                                 -->
<!--                                                              -->
<!-- Left_Model element is only allowed to contain one child      -->
<!-- element.                                                     -->
<!-- ************************************************************ -->
<!ELEMENT Left_Model (Device                        |
                      Port                          |
                      Topology_Container            |
                      Location_Container            |
                      EventModel                    |
                      Model
                     ) >


<!-- ************************************************************ -->
<!-- Right_Model element defines right side model in a Spectrum   -->
<!-- association.                                                 -->
<!--                                                              -->
<!-- Right_Model element is only allowed to contain one child     -->
```

```
<!-- element.                                                    -->
<!-- ********************************************************** -->
<!ELEMENT Right_Model (Device                        |
                       Port                          |
                       Topology_Container            |
                       Location_Container            |
                       EventModel                    |
                       Model
                      ) >

<!-- ********************************************************** -->
<!-- Model element can be used to represent any SPECTRUM models.  -->
<!--                                                            -->
<!-- model_type and name have to be provided to define a model.   -->
<!-- "model_type" and "name" should be used to uniquely identify  -->
<!-- models.   However, if "model_handle" is provided, the values -->
<!-- of "model_type" and "name" will not be used.                 -->
<!-- ********************************************************** -->
<!ELEMENT Model ( #PCDATA ) >
<!ATTLIST Model
          name           CDATA    #REQUIRED
          model_type     CDATA    #REQUIRED
          model_handle   CDATA    #IMPLIED >

<!-- ********************************************************** -->
<!-- GenericView element, which is used for customized view,      -->
<!-- may contain any number of GenericView_Container and Device   -->
<!-- elements.                                                    -->
<!--                                                            -->
<!-- This element has 3 required attributes containment_relation, -->
<!-- model_type and name.                                         -->
<!-- ********************************************************** -->
<!ELEMENT GenericView ((GenericView_Container | Device )*) >
<!ATTLIST GenericView
    containment_relation  CDATA          #REQUIRED
    model_type            CDATA          #REQUIRED
    name                  CDATA          #REQUIRED
    complete_topology     (false | true) #IMPLIED >
<!-- ********************************************************** -->
<!-- GenericView_Container element, which is used for the        -->
<!-- GenericView container, may contain any number of            -->
<!-- GenericView_Container, and Device elements.                 -->
<!--                                                            -->
<!-- This element requires model_type and name attribute.        -->
<!-- Attribute containment_relation is not required.  If it's not-->
<!-- specified, the parent's containment_relation will be        -->
<!-- inheritted.  The Model_Attr can be used for multi-lines     -->
<!-- text string SPECTRUM attrs, or list attributes.             -->
<!-- ********************************************************** -->
<!ELEMENT GenericView_Container ( GenericView_Container |
    Device
    )*>
<!ATTLIST GenericView_Container
```

```
    name                CDATA          #REQUIRED
    model_type          CDATA          #REQUIRED
    containment_relation CDATA         #IMPLIED >
<!-- ********************************************************** -->
<!-- Model_Attr is used for multi-lines text string or list      -->
<!-- SPECTRUM attributes.                                        -->
<!--                                                             -->
<!-- attr_id is required for this element to specify the         -->
<!-- SPECTRUM attribute.  This element can contain multi-lines   -->
<!-- of text strings for SPECTRUM Text String attr type, or      -->
<!-- multiple List_Value elements for a SPECTRUM list attribute. -->
<!-- ********************************************************** -->
<!ELEMENT Model_Attr ( #PCDATA | List_Value )* >
<!ATTLIST Model_Attr
           attr_id   CDATA          #REQUIRED >
<!-- ********************************************************** -->
<!-- List_Value is used for SPECTRUM list attribute values.      -->
<!--                                                             -->
<!-- Each List_Value conatins a PCDATA and serves for one        -->
<!-- instance value for the list attribute.                      -->
<!-- ********************************************************** -->
<!ELEMENT List_Value ( #PCDATA ) >
<!-- ********************************************************** -->
<!-- Service Level Management topology elements.                -->
<!-- ********************************************************** -->
<!ELEMENT CustomerManager ( SM_Customer |
               SM_CustomerGroup
               )*>
<!ATTLIST CustomerManager
    name                  CDATA                 #IMPLIED
    containment_relation  ( Groups_Customers )  #IMPLIED
    model_type            ( CustomerManager )   #IMPLIED
    >
<!ELEMENT SM_ServiceMgr ( SM_Service )*>
<!ATTLIST SM_ServiceMgr
    name            CDATA                          #IMPLIED
    containment_relation          ( SlmContains )   #IMPLIED
    model_type                    ( SM_ServiceMgr ) #IMPLIED
    >
<!ELEMENT SM_SLA_Mgr ( SM_SLA )*>
<!ATTLIST SM_SLA_Mgr
    name            CDATA                        #IMPLIED
    containment_relation  ( SlmContainsSLAs )    #IMPLIED
    model_type            ( SM_SLA_Mgr )         #IMPLIED
    >
<!ELEMENT SM_Service_Mgt ( CustomerManager |
           SM_ServiceMgr   |
           SM_SLA_Mgr
           )*>
<!ATTLIST SM_Service_Mgt
    name            CDATA                               #IMPLIED
    containment_relation  ( SlmHasServiceComponent )    #IMPLIED
    model_type            ( SM_Service_Mgt )            #IMPLIED
```

```
    >
<!-- The Correlation element represents the root model Correlation_Manager -->
<!-- (0x10469).  It can only contain Correlation_Domain elements, and has  -->
<!-- no attributes.  All Correlation_Domains will be related to the        -->
<!-- Correlation Manager by the CORRELATES relation.                       -->
<!-- Since Correlation_Manager is a unique model, this element does not    -->
<!-- actually cause the SpectroSERVER to create a model.  It represents    -->
<!-- a pre-existing model.                                                 -->
<!ELEMENT Correlation ( Correlation_Domain )*>
<!-- The Correlation_Domain can contain any number of Devices, Ports,      -->
<!-- Model_Attrs, or GenericView_Containers.  They will all be related     -->
<!-- to the Correlation_Domain by the CORRELATES relation.  Correlation_   -->
<!-- Domains also have no attributes.                                      -->
<!ELEMENT Correlation_Domain (  Device               |
                Port                |
                Model_Attr          |
                GenericView_Container
                )*>
<!ATTLIST Correlation_Domain
    name            CDATA                #REQUIRED
    >
<!ELEMENT RTM_Test ( #PCDATA) >
<!ATTLIST RTM_Test
    name            CDATA             #REQUIRED
    model_type      ( RTM_Test )      #IMPLIED
    >
<!ELEMENT SM_Service ( SM_Service         |
            SM_AttrMonitor     |
            SM_LatencyMon      |
            SM_ConnectMon      |
            Device             |
            Port               |
            Topology_Container |
            RTM_Test           |
            MonitorPolicy_Attr |
            Schedule
            )*>
<!-- ********************************************************************* -->
<!-- Represents a SM_Service model                                         -->
<!--                                                                       -->
<!-- Criticality can be one of the following                               -->
<!--                                                                       -->
<!--  10 - Low                                                             -->
<!--  15 - Medium Low                                                      -->
<!--  20 - Medium                                                          -->
<!--  25 - Medium High                                                     -->
<!--  10 - High                                                            -->
<!--                                                                       -->
<!-- AttrToWatch can be one of the following                               -->
<!--                                                                       -->
<!--  Condition - can be used for most models, polices 1-5                 -->
<!--  Contact_Status - typically used for device models, policies 10-13    -->
<!--  Port_Status - used for interface models, policies 14-17              -->
```

```
<!--  LatestErrorStatus - used for RTM_Test models, policies 18-21          -->
<!--  or Response_Time                                                      -->
<!--  RM_Condition - SM_AttrMonitor or SM_Service models, policies 6-9      -->
<!--  or Service_Health                                                     -->
<!--                                                                        -->
<!-- MonitorPolicy_ID - 1-21 Index of SLM_DefaultPolicies of GlobalConfig   -->
<!--                                                                        -->
<!--  1  - Condition Rollup                                                 -->
<!--  2  - Condition Redundancy                                             -->
<!--  3  - Condition High Sensitivity                                       -->
<!--  4  - Condition Low Sensitivity                                        -->
<!--  5  - Condition Percentage                                             -->
<!--  6  - Service Health Redundancy                                        -->
<!--  7  - Service Health High Sensitivity                                  -->
<!--  8  - Service Health Low Sensitivity                                   -->
<!--  9  - Service Health Percentage                                        -->
<!--  10 - Contact Status Redundancy                                        -->
<!--  11 - Contact Status High Sensitivity                                  -->
<!--  12 - Contact Status Low Sensitivity                                   -->
<!--  13 - Contact Status Percentage                                        -->
<!--  14 - Port Status Redundancy                                           -->
<!--  15 - Port Status High Sensitivity                                     -->
<!--  16 - Port Status Low Sensitivity                                      -->
<!--  17 - Port Status Percentage                                           -->
<!--  18 - Response Time Redundancy                                         -->
<!--  19 - Response Time High Sensitivity                                   -->
<!--  20 - Response Time Low Sensitivity                                    -->
<!--  21 - Response Time Low Percentage                                     -->
<!--                                                                        -->
<!-- ********************************************************************** -->
<!ATTLIST SM_Service
    name              CDATA                          #REQUIRED
    containment_relation  ( SlmMonitors |
                SlmWatchesContainer |
                MaintenenceScheduledBy ) #IMPLIED
    Criticality  ( 10 | 15 | 20 | 25 | 30 )          #IMPLIED

    AttrToWatch             CDATA                    #IMPLIED
    MonitorPolicy_ID        CDATA                    #IMPLIED
    is_managed              (true  | false)          #IMPLIED
    Generate_Service_Alarms (true  | false)          #IMPLIED
    Security_String         CDATA                    #IMPLIED
    model_type        ( SM_Service )                 #IMPLIED
    >
<!ELEMENT SM_AttrMonitor ( SM_Service         |
            SM_AttrMonitor     |
            SM_LatencyMon      |
            SM_ConnectMon      |
            Device             |
            Port               |
            Topology_Container |
            RTM_Test           |
            MonitorPolicy_Attr |
```

```
            )*>

<!-- ********************************************************************** -->
<!-- Represents a SM_AttrMonitor model                                      -->
<!--                                                                        -->
<!-- AttrToWatch can be one of the following                                -->
<!--                                                                        -->
<!--  Condition - can be used for most models, polices 1-5                  -->
<!--  Contact_Status - typically used for device models, policies 10-13     -->
<!--  Port_Status - used for interface models, policies 14-17               -->
<!--  LatestErrorStatus - used for RTM_Test models, policies 18-21          -->
<!--  or Response_Time                                                      -->
<!--  RM_Condition - SM_AttrMonitor or SM_Service models, policies 6-9      -->
<!--  or Service_Health                                                     -->
<!--                                                                        -->
<!-- MonitorPolicy_ID - 1-21 Index of SLM_DefaultPolicies of GlobalConfig   -->
<!--                                                                        -->
<!--  1  - Condition Rollup                                                 -->
<!--  2  - Condition Redundancy                                             -->
<!--  3  - Condition High Sensitivity                                       -->
<!--  4  - Condition Low Sensitivity                                        -->
<!--  5  - Condition Percentage                                             -->
<!--  6  - Service Health Redundancy                                        -->
<!--  7  - Service Health High Sensitivity                                  -->
<!--  8  - Service Health Low Sensitivity                                   -->
<!--  9  - Service Health Percentage
<!--  10 - Contact Status Redundancy                                        -->
<!--  11 - Contact Status High Sensitivity                                  -->
<!--  12 - Contact Status Low Sensitivity                                   -->
<!--  13 - Contact Status Percentage                                        -->
<!--  14 - Port Status Redundancy                                           -->
<!--  15 - Port Status High Sensitivity                                     -->
<!--  16 - Port Status Low Sensitivity                                      -->
<!--  17 - Port Status Percentage                                           -->
<!--  18 - Response Time Redundancy                                         -->
<!--  19 - Response Time High Sensitivity                                   -->
<!--  20 - Response Time Low Sensitivity                                    -->
<!--  21 - Response Time Low Percentage                                     -->
<!--                                                                        -->
<!--  Special_Cause_List - List or range of alarm causes which can be       -->
<!--                       used to specify included or excluded from        -->
<!--                       impacting the service health of the Service      -->
<!--                       or Resource Monitor model.  Available only       -->
<!--                       when AttrToWatch is Condition.                   -->
<!--                                                                        -->
<!--  Cause_List_Control - Specifies how Special_Cause_List is used.        -->
<!--                       0 - Unused                                       -->
<!--                       1 - Inclusive                                    -->
<!--                       2 - Exclusive                                    -->
<!--                                                                        -->
<!-- ********************************************************************** -->

<!ATTLIST SM_AttrMonitor
```

```
    name                    CDATA                       #REQUIRED
    containment_relation    ( SlmMonitors |
                SlmWatchesContainer )  #IMPLIED
    is_managed              (true  | false)          #IMPLIED
    AttrToWatch             CDATA                       #IMPLIED
    MonitorPolicy_ID        CDATA                       #IMPLIED
    Generate_Service_Alarms (true  | false)          #IMPLIED
    model_type      ( SM_AttrMonitor )              #IMPLIED
    >
<!ELEMENT MonitorPolicy_Attr ( #PCDATA ) >
<!ELEMENT SM_SLA ( SM_Service   |
        SM_Guarantee |
        Schedule
        )*>
<!-- ********************************************************************** -->
<!-- Represents a SM_Guarantee model                                        -->
<!--                                                                        -->
<!-- SLAControl can have the following values                               -->
<!--                                                                        -->
<!--  0 = Inactive                                                          -->
<!--  1 = Active                                                            -->
<!--                                                                        -->
<!-- ********************************************************************** -->
<!ATTLIST SM_SLA
    name              CDATA                   #REQUIRED
    containment_relation  ( SlmHasGuarantee |
                SlmGuarantees |
                SlaPeriod )       #IMPLIED
    is_managed          (true  | false)       #IMPLIED
    SLA_Control         ( 0 | 1 )             #IMPLIED
    SLA_ExpirationDate    CDATA               #IMPLIED
    SLA_Notes             CDATA               #IMPLIED
    SLA_Description       CDATA               #IMPLIED
    Security_String       CDATA               #IMPLIED
    model_type          ( SM_SLA )            #IMPLIED
    >
<!ELEMENT SM_Guarantee ( SM_Service      |
            SM_AttrMonitor  |
            SM_LatencyMon   |
            SM_ConnectMon   |
            Schedule
            )*>
<!-- ********************************************************************** -->
<!-- Represents a SM_Guarantee model                                        -->
<!--                                                                        -->
<!-- GuaranteeControl can have the following values                         -->
<!--                                                                        -->
<!--  0 = Inactive                                                          -->
<!--  1 = Active                                                            -->
<!--                                                                        -->
<!-- GuranteeType can have the following values                             -->
<!--                                                                        -->
<!--  0 = Availability                                                      -->
```

```
<!--  1 = Performance                                                    -->
<!--  2 = Mean Time To Repair                                            -->
<!--  3 = Maximum Outage Time                                            -->
<!--                                                                     -->
<!-- ServiceHealthType can have the following values                     -->
<!--                                                                     -->
<!--  1 - Down                                                           -->
<!--  2 - Degraded                                                       -->
<!--                                                                     -->
<!-- ******************************************************************* -->
<!ATTLIST SM_Guarantee
    name                        CDATA          #REQUIRED
    containment_relation  ( SlmIsMeasuredBy |
                   SlmSchedulesGuarantee )  #IMPLIED
    is_managed            (true  | false)      #IMPLIED
    GuaranteeControl      ( 0 | 1 )            #IMPLIED
    GuaranteeType         ( 0 | 1 | 2 | 3 )    #REQUIRED
    ServiceHealthType     ( 1 | 2 )            #IMPLIED
    WarningThreshold            CDATA          #IMPLIED
    WarningThresholdPercent     CDATA          #IMPLIED
    ViolationThreshold          CDATA          #IMPLIED
    ViolationThresholdPercent   CDATA          #IMPLIED
    GuaranteeNotes              CDATA          #IMPLIED
    GuaranteeDescription        CDATA          #IMPLIED
    model_type             ( SM_Guarantee )    #IMPLIED
    MOT_Threshold               CDATA          #IMPLIED
    MTTR_Threshold              CDATA          #IMPLIED
    MTBF_Threshold              CDATA          #IMPLIED
    >
<!ELEMENT SM_LatencyMon ( Topology_Container |
                MonitorPolicy_Attr )*>
<!ATTLIST SM_LatencyMon
    name                    CDATA                #REQUIRED
    containment_relation  ( SlmMonitors |
                   SlmWatchesContainer ) #IMPLIED
    is_managed            ( true  | false )       #IMPLIED
    DefaultMaxRTT           CDATA                #IMPLIED
    DefaultMeasureInterval  CDATA                #IMPLIED
    model_type            ( SM_LatencyMon )      #IMPLIED
    >
<!ELEMENT SM_CustomerGroup ( SM_CustomerGroup |
            SM_Customer
            )*>
<!ATTLIST SM_CustomerGroup
    name                           CDATA          #REQUIRED
    containment_relation ( Groups_Customers )     #IMPLIED
    model_type           ( SM_CustomerGroup )     #IMPLIED
    >
<!ELEMENT SM_Customer ( SM_Service |
                     SM_SLA     |
             )*>
<!-- ******************************************************************* -->
<!-- Represents a SM_Customer model                                      -->
```

```
<!--                                                                        -->
<!-- Criticality can be one of the following                                -->
<!--                                                                        -->
<!--  10 - Low                                                              -->
<!--  15 - Medium Low                                                       -->
<!--  20 - Medium                                                           -->
<!--  25 - Medium High                                                      -->
<!--  10 - High                                                             -->
<!--                                                                        -->
<!-- ********************************************************************** -->
<!ATTLIST SM_Customer
    name                          CDATA  #REQUIRED
    containment_relation  ( SlmAgreesTo |
                SlmUses )     #IMPLIED
    Security_String               CDATA  #IMPLIED
    CustomerID                    CDATA  #IMPLIED
    Criticality                   CDATA  #IMPLIED
    CustomerField4                CDATA  #IMPLIED
    CustomerField5                CDATA  #IMPLIED
    CustomerField6                CDATA  #IMPLIED
    CustomerField7                CDATA  #IMPLIED
    Contact_Name                  CDATA  #IMPLIED
    Contact_Title                 CDATA  #IMPLIED
    Contact_Location              CDATA  #IMPLIED
    Email_Address                 CDATA  #IMPLIED
    Phone_Number                  CDATA  #IMPLIED
    Mobile_Phone_Number           CDATA  #IMPLIED
    Pager_Number                  CDATA  #IMPLIED
    Fax_Number                    CDATA  #IMPLIED
    User_Defined_1                CDATA  #IMPLIED
    User_Defined_2                CDATA  #IMPLIED
    User_Defined_3                CDATA  #IMPLIED
    User_Defined_4                CDATA  #IMPLIED
    Secondary_Contact_Name        CDATA  #IMPLIED
    Secondary_Contact_Title       CDATA  #IMPLIED
    Secondary_Contact_Location    CDATA  #IMPLIED
    Secondary_Email_Address       CDATA  #IMPLIED
    Secondary_Phone_Number        CDATA  #IMPLIED
    Secondary_Mobile_Phone_Number CDATA  #IMPLIED
    Secondary_Pager_Number        CDATA  #IMPLIED
    Secondary_Fax_Number          CDATA  #IMPLIED
    Secondary_User_Defined_1      CDATA  #IMPLIED
    Secondary_User_Defined_2      CDATA  #IMPLIED
    Secondary_User_Defined_3      CDATA  #IMPLIED
    Secondary_User_Defined_4      CDATA  #IMPLIED
    model_type           ( SM_Customer )  #IMPLIED
    >
<!-- ********************************************************************** -->
<!-- Represents a GlobalCollection model                                    -->
<!--                                                                        -->
<!-- ********************************************************************** -->
<!ELEMENT GlobalCollection (( Device             |
            Topology_Container |
```

```
            Location_Container )*) >

<!ATTLIST GlobalCollection
    name                  CDATA              #REQUIRED
    containment_relation  CDATA              "GlobalCollect"
    collectionDescription CDATA              #IMPLIED
    Security_String       CDATA              #IMPLIED >
```

Appendix C. XML Examples

This section contains the following XML examples to help you work with the DTD:

> **NOTE**
> The element names in each example are highlighted in bold. Making the names bold makes the examples easier to read; it is not intended to imply formatting necessary for the XML input file.

Example 1 Importing into the Topology View

This example shows a basic input file that imports information into the DX NetOps Spectrum Topology view. This file creates a Network container model in the Topology view. In the Network container, a LAN container model is created. Within the LAN container two devices are created. The DNS name deadlock identifies one device, and the IP address identifies the other device.

The complete_topology attribute of the Topology element is set to False. In this case, DX NetOps Spectrum respects other models that previously exist in the Topology view. Consequently, this file only looks to create models for entries that are listed in the XML file that are not already modeled. The models that are created are placed in the Topology hierarchy as specified in the file. Models that previously exist in the Topology hierarchy are not rediscovered but are moved to the container specified in the file.

> **NOTE**
> When complete_topology is set to false, existing models that are in the container but not listed in the import file are *not* sent to the Lost and Found. These models *are* sent to Lost and Found when complete_topology is set to true.

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE Import SYSTEM ".modelinggateway.dtd">
<Import>
<!-- ********************************** -->
<!-- This part is for Topology view import -->
<!-- ********************************** -->
<Topology complete_topology="false">
    <Device ip_dnsname="10.253.9.17" model_type="GnSNMPDev"
        community_string="public"/>
    <Device ip_dnsname="nmcss52-5" />
    <Topology_Container model_type="Network" name="My Network"
        Security_String="public" subnet_address="10.253.0.0"
        subnet_mask="255.255.0.0">
        <Topology_Container model_type="Lan" name="Lan1"
            Security_String="public" subnet_address="10.253.9.0"
            subnet_mask="255.255.255.0">
            <Device ip_dnsname="deadlock" />
            <Device ip_dnsname="10.253.9.18" poll_interval="333" />
```

```
        </Topology_Container>
    </Topology_Container>
</Topology>
</Import>
```

Example 2 Creating Connections

The following example shows the creation of a connection between two ATM circuits and the creation of a connection between two Frame Relay circuits. The example also shows a connection between a FrameRelay DLCI port and an ATM VCL port.

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE Import SYSTEM ".modelinggateway.dtd">
<Import>
<Topology complete_topology="false">
 <Connection create_pipe="false">
  <Device ip_dnsname="10.253.32.225">
   <Port identifier_name="atmVclTableInstance"
    identifier_value="5.0.5"
    circuit_name="ATM Link1"
    circuit_id = "ATM 5017" />
  </Device>
  <Device ip_dnsname="192.168.52.25">
   <Port identifier_name="atmVclTableInstance"
    identifier_value="3.0.12"
    circuit_name="ATM Link1"
    circuit_id = "ATM 5017" />
  </Device>
 </Connection>
 <Connection>
  <Device ip_dnsname="10.253.9.18">
   <Port identifier_name="frCircuitTableInstance"
    identifier_value="2.27"/>
  </Device>
  <Device ip_dnsname="nmcss52-5">
   <Port identifier_name="frCircuitTableInstance
    identifier_value="4.161"/>
  </Device>
 </Connection>
 <!-- ****************************************** -->
 <!-- Connection between a FrameRelay DLCI port and -->
 <!-- an ATM VCL port.                           -->
 <!-- ****************************************** -->
 <Connection>
  <Device ip_dnsname="10.253.9.18">
   <Port identifier_name="frCircuitTableInstance"
    identifier_value="2.27"/>
  </Device>
  <Device ip_dnsname="10.253.32.225">
   <Port identifier_name="atmVclTableInstance"
    identifier_value="5.0.17"/>
  </Device>
 </Connection>
```

```
 <Connection>
  <Device ip_dnsname="nmcss52-5">
   <Port identifier_name="ifIndex" identifier_value="3"/>
  </Device>
  <Device ip_dnsname="10.253.9.17">
   <Port identifier_name="ifPhysAddress"
    identifier_value="0:4:27:C:91:C0"/>
  </Device>
 </Connection>
<Topology>
</Import>
```

Example 3 Updating and Destroying

This example illustrates the use of the Update and the Destroy elements.

The Update element contains a Location_Container element. This example updates the model name by using the name and model_name attributes of the Location_Container element. The name attribute is set equal to the current name and identifies the model to be updated. The model_name attribute updates the value of the name attribute to Peace2.

The Update element also contains a Device element and Port element. The attributes identifer_name and identifier_value are used to identify the port to be updated. The other attributes that are specified are those attributes whose values are updated. The port model name is changed to port 2 and the poll_status is changed to False.

The Destroy element eliminates the device model deadlock. Any connections or ports that are associated with deadlock are automatically destroyed. The Building container model Durham is also eliminated. Any models that are contained within the Durham building container are sent to the Lost and Found.

The Destroy element also removes the connection between the specified port on device nmcss52-5 and the specified port on nmcss52-3.

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE Import SYSTEM ".modelinggateway.dtd">
<Import>
<!-- ********************************************** -->
<!-- Model update.....................................-->
<!-- ********************************************** -->
 <Update>
<!-- ********************************************** -->
<!-- Change container Peace's model name from Peace to-->
<!-- Peace2 ..................................... ....-->
<!-- ********************************************** -->
  <Location_Container model_type="Building" name="Peace"
   model_name="Peace2"/>
<!-- ************************************** -->
<!-- Update port ifIndex=2 on device nmcss52-5  -->
<!-- ************************************** -->
  <Device ip_dnsname="nmcss52-5">
   <Port identifier_name="ifIndex" identifier_value="2"
    model_name="port 2" poll_status="false" />
  </Device>
 </Update>
  <!-- ***************************** -->
  <!-- Destroy models and connections. -->
  <!-- ***************************** -->
```

```
 <Destroy>
  <Device ip_dnsname="deadlock"/>
  <Location_Container  model_type="Building" name="Durham" />
  <Connection>
   <Device ip_dnsname="nmcss52-5">
    <Port identifier_name="ifIndex"
     identifier_value="1"/>
   </Device>
   <Device ip_dnsname="10.253.9.17">
    <Port identifier_name="ipAddress"
     identifier_value="10.253.8.18"/>
   </Device>
  </Connection>
 </Destroy>
</Import>
```

Example 4 Creating, Updating, and Destroying

The following XML file illustrates most of the functionality of the elements that are contained within the DTD. This file creates data in both the Topology and Location views, creates connections, updates attributes, and destroys models and connections.

The first section of the XML file creates models and connections between these models in the Topology view. The section begins with the Topology element, <Topology...>. The container and device models are created first and then connections are established. This section ends when the Topology element closes, </Topology>.

After the Topology element is closed, there is a section where another connection is created. This section illustrates that you can create connections without nesting the Connection element within the Topology element.

The next section of the file begins with the Location element, <Location...>. This section demonstrates creating a container and device models in the Location view. This section ends when the Location element closes, </Location>.

The next section begins with the Update element, <Update>. In this section, attribute values of a container, a device, and a port are modified. You cannot know the current attribute values of each of the elements that are represented by looking at the XML file. Therefore, it is not easy to discern which elements are being updated. In general, each element contains an attribute that uniquely identifies the model or port to be updated. The rest of the attributes are specified to update their values. For example, the first element is the Location_Container element. The name attribute uniquely identifies the model. The model_type attribute is specified to update its value, perhaps from Region to Building. The only hierarchy that can be defined in the Update element is the Device/Port hierarchy that specifies the port you would like to update. This section ends when the Update element closes, </Update>.

The last section of this file uses the Destroy element. The Destroy element eliminates:

- The device at 10.253.9.19
- The container model Durham
- The connection between the specified ports on device nmcss52-5 and the device at 10.253.9.17

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE Import SYSTEM ".modelinggateway.dtd">
<Import>
<!-- ********************************** -->
<!-- This part is for Topology view import -->
<!-- ********************************** -->
 <Topology discover_connections="false" complete_topology="false">
  <Device ip_dnsname="10.253.9.109" model_type="GnSNMPDev"
```

```
 community_string="public" is_managed="false"/>
<Device ip_dnsname="10.253.9.17"
 poll_interval="333" log_ratio="11"/>
<Device ip_dnsname="10.253.9.19" community_string="public"/>
<Device ip_dnsname="nmcss52-5" />
<Topology_Container model_type="Network" name="My Network"
 Security_String="public" subnet_address="10.253.0.0"
 subnet_mask="255.255.0.0" complete_topology="true">
 <Topology_Container model_type="Lan"
  name="MyLan" Security_String="public"
  subnet_address="10.253.9.0"
  subnet_mask="255.255.255.0">
  <Device ip_dnsname="10.253.9.18"
   community_string="public"
   poll_interval="333"
   log_ratio="5"/>
 </Topology_Container>
</Topology_Container>
<Topology_Container model_type="IPClassC" name="my_net"
 subnet_address="172.19.57.0">
 <Device  model_type="Pingable"
  ip_dnsname="172.19.57.91"/>
 <Device  model_type="Fanout" ip_dnsname="1.2.3.4"/>
 <Device  ip_dnsname="10.253.9.16"
  community_string="public"/>
</Topology_Container>
<Topology_Container model_type="Lan" name="lan2"
 Security_String="public" subnet_address="10.253.7.0"
 subnet_mask="255.255.255.0" complete_topology="true">
 <Device ip_dnsname="10.253.7.17"
  community_string="public"
  poll_interval="333" log_ratio="11"/>
 <Device ip_dnsname="10.253.32.101"/>
 <Device ip_dnsname="192.168.125.161"
  model_type="GnSNMPDev"/>
</Topology_Container>
<Device ip_dnsname="172.19.57.92" />
<Device ip_dnsname="172.19.57.93" />
<Device ip_dnsname="10.253.32.225" model_type="M46_04"/>
<Connection>
 <Device ip_dnsname="172.19.57.93">
  <Port identifier_name= "frCircuitTableInstance"
   identifier_value="4.161"/>
 </Device>
 <Device ip_dnsname="192.168.125.161">
  <Port identifier_name ="frCircuitTableInstance"
   identifier_value="2.161"/>
 </Device>
</Connection>
<Connection create_pipe="false">
 <Device ip_dnsname="10.253.32.101">
  <Port identifier_name ="atmVclTableInstance"
   identifier_value="3.1.52"/>
```

```
    </Device>
    <Device ip_dnsname="10.253.32.225">
     <Port identifier_name ="atmVclTableInstance"
      identifier_value="5.0.68"
      circuit_name="ATM 68"
      circuit_id ="ATM ID 68"/>
    </Device>
   </Connection>
   <Connection>
    <Device ip_dnsname="nmcss52-5">
     <Port identifier_name="ifIndex"
      identifier_value="1"/>
    </Device>
    <Device ip_dnsname="10.253.9.17">
     <Port identifier_name="ipAddress"
      identifier_value="10.253.8.18"/>
    </Device>
   </Connection>
  </Topology>
  <Connection>
   <Device ip_dnsname="172.19.57.92">
    <Port identifier_name="ifPhysAddress"
     identifier_value="0:E0:63:7C:19:61"/>
   </Device>
   <Device ip_dnsname="10.253.9.17">
    <Port identifier_name="ipAddress"
     identifier_value="10.253.8.65"/>
   </Device>
  </Connection>
<!-- ********************************** -->
<!-- This part is for Location view import -->
<!-- ********************************** -->
  <Location complete_topology="true">
   <Location_Container model_type="Country" name="USA"
    Security_String="whatever">
    <Location_Container model_type="Region"
     name="New Hampshire"
     complete_topology="false">
     <Location_Container model_type="Site"
     name="Durham"
      <Device ip_dnsname = "10.253.32.10"/>
      <Device ip_dnsname = "172.19.57.93" />
     </Location_Container>
    </Location_Container>
   </Location_Container>
   <Location_Container model_type="Building" name="Durham"
    Security_String="public">
    <Location_Container model_type="Room" name="my_room"
     Security_String="hahaha">
     <Device ip_dnsname="10.253.9.16"
      community_string="public"/>
     <Device ip_dnsname="10.253.9.17" />
     <Device ip_dnsname = "10.253.9.18"/>
```

```
    </Location_Container>
   </Location_Container>
   <Location_Container model_type="Building" name="Peace"
    Security_String="aprisma">
    <Location_Container model_type="Room" name= "Lab 1">
     <Device ip_dnsname="10.253.7.17"
      community_string="public"/>
     <Device ip_dnsname="192.168.125.161"/>
    </Location_Container>
   </Location_Container>
 </Location>
<!-- ************************************* -->
<!-- This part is for model update             -->
<!-- ************************************* -->
 <Update>
  <Location_Container model_type="Building" name="Peace"
   model_modify_author="ltang"/>
  <Device ip_dnsname="172.19.57.93" poll_interval="101"
   model_name="haha" />
    <!-- ************************************* -->
    <!-- This part is to update the port ifIndex=2 -->
    <!-- on device nmcss52-5                       -->
    <!-- ************************************* -->
  <Device ip_dnsname="nmcss52-5">
   <Port identifier_name="ifIndex" identifier_value="2"
    model_name="port 2" poll_interval="1103"
    poll_status="false" log_ratio="12"/>
  </Device>
  <Topology_Container model_type="Lan" name="lan2"
   Security_String="top secret"/>
 </Update>
<!-- ******************************************** -->
<!-- This part is for model and connection deletion -->
<!-- ******************************************** -->
 <Destroy>
  <Device ip_dnsname="10.253.9.19"/>
  <Location_Container model_type="Building" name="Durham"/>
  <Connection>
   <Device ip_dnsname="nmcss52-5">
    <Port identifier_name="ifIndex"
     identifier_value="1"/>
   </Device>
   <Device ip_dnsname="10.253.9.17">
    <Port identifier_name="ipAddress"
     identifier_value="10.253.8.18"/>
   </Device>
  </Connection>
 </Destroy>
</Import>
```

Appendix D. modelinggatewayresource.xml

This section contains a copy of the .modelinggatewayresource.xml file. However, this copy might not be the latest version of this file. For the latest version, use the actual file that shipped with your Modeling Gateway Toolkit.

```
<?xml version="1.0" standalone="no"?>
<TopologyImportExportResourceFile>
<!-- ************************************* -->
<!-- SPECTRUM attribute names and IDs that    -->
<!-- are used for topology export and import. -->
<!-- ************************************* -->
<Attributes
   circuit_id                    = "0xc4042f"
   circuit_name                  = "0xc40430"
   community_string              = "0x10024"
   agent_port                    = "0x10023"
   DeviceType                    = "0x23000e"
   is_managed                    = "0x1295d"
   log_ratio                     = "0x10072"
   manager_name                  = "0x3dc0009"
   model_modify_author           = "0x11025"
   model_name                    = "0x1006e"
   name                          = "0x1006e"
   poll_interval                 = "0x10071"
   poll_status                   = "0x1154f"
   TryCount                      = "0x110c5"
   Security_String               = "0x10009"
   subnet_address                = "0x1027f"
   subnet_mask                   = "0x110b8"
   subnet_list                   = "0x11953"
   TimeOut                       = "0x110c4"
   trapIPAddress                 = "0x3dc0007"
   unique_id                     = "0x3dc0004"
   Value_When_Orange             = "0x1000d"
   Value_When_Red                = "0x1000e"
   Value_When_Yellow             = "0x1000c"
   LatestErrorStatus             = "456008c"
   Response_Time                 = "456008c"
   AttrToWatch                   = "0x12a43"
   MonitorPolicy                 = "0x12a3e"
   MonitorPolicy_ID              = "0x12a51"
   Generate_Service_Alarms       = "0x12a66"
   Special_Cause_List            = "0x12b47"
   Cause_List_Control            = "0x12d50"
Contact_Status                = "0x10004"
Port_Status                   = "0x10f1b"
RM_Condition                  = "0x12a40"
Service_Health                = "0x12a40"
Criticality                   = "0x1290c"
Condition                     = "0x1000a"
Condition_Value               = "0x1000b"
AccumulationMethod            = "0x4500007"
GuaranteeControl              = "0x4500022"
```

```
GuaranteeNotes                = "0x4500021"
GuaranteeDescription          = "0x12a4b"
GuaranteeType                 = "0x4500018"
ServiceHealthType             = "0x4500019"
ViolationThreshold            = "0x450001e"
ViolationThresholdPercent     = "0x4500024"
WarningThreshold              = "0x450001d"
WarningThresholdPercent       = "0x4500023"
SCHED_Daily_Repeat_Limit      = "0x1299a"
SCHED_Duration                = "0x12993"
SCHED_Recurrence_Multiplier   = "0x1299b"
SCHED_Recurrence              = "0x12994"
SCHED_Start_DoM               = "0x12991"
SCHED_Start_DoW               = "0x12990"
SCHED_Start_Hour              = "0x1298f"
SCHED_Start_Minute            = "0x1298e"
SCHED_Start_Month             = "0x12992"
SCHED_Start_Day               = "0x129e4"
SCHED_DayBitMask              = "0x129da"
SCHED_Start_Year              = "0x129e3"
SCHED_Start_MoY               = "0x12b48"
SCHED_Description             = "0x12bbc"
SLA_Control                   = "0x4500015"
SLA_Notes                     = "0x4500017"
SLA_ExpirationDate            = "0x4500025"
SLA_Description               = "0x12a4b"
DefaultMaxRTT                 = "0x4500001"
DefaultMeasureInterval        = "0x4500002"
CustomerID                    = "0x12a44"
CustomerField4                = "0x12a39"
CustomerField5                = "0x12a3a"
CustomerField6                = "0x12a3b"
CustomerField7                = "0x12a3c"
Contact_Name                  = "0x12a20"
Contact_Title                 = "0x12a21"
Contact_Location              = "0x12a22"
Email_Address                 = "0x12a27"
Phone_Number                  = "0x12a23"
Mobile_Phone_Number           = "0x12a24"
Pager_Number                  = "0x12a25"
Fax_Number                    = "0x12a26"
User_Defined_1                = "0x12a28"
User_Defined_2                = "0x12a29"
User_Defined_3                = "0x12a2a"
User_Defined_4                = "0x12a2b"
Secondary_Contact_Name        = "0x12a2c"
Secondary_Contact_Title       = "0x12a2d"
Secondary_Contact_Location    = "0x12a2e"
Secondary_Email_Address       = "0x12a33"
Secondary_Phone_Number        = "0x12a2f"
Secondary_Mobile_Phone_Number = "0x12a30"
Secondary_Pager_Number        = "0x12a31"
Secondary_Fax_Number          = "0x12a32"
```

```
Secondary_User_Defined_1      = "0x12a34"
Secondary_User_Defined_2      = "0x12a35"
Secondary_User_Defined_3      = "0x12a36"
Secondary_User_Defined_4      = "0x12a37"
MOT_Threhold                 = "0x450002c"
MTBF_Threshold                = "0x4500032"
MTTR_Thresshold                = "0x450002f"
Policy_Name_List              = "0x12a4a"
collectionDescription         = "0x12a67"
/>
<!-- ****************************************** -->
<!-- SPECTRUM model type names and handles that  -->
<!-- can be used in topology import XML file.    -->
<!-- ****************************************** -->
<ModelTypes
Universe                      = "0x10091"
Network                       = "0x1002e"
Lan                           = "0x1002d"
IPClassA                      = "0x103d5"
IPClassB                      = "0x103d6"
IPClassC                      = "0x103d7"
LAN_802_3                     = "0x1003c"
LAN_802_5                     = "0x1003d"
ATM_NETWORK                   = "0xaa000f"
EventAdmin                    = "0x3dc0000"
GlobalCollection              = "0x10474"
World                         = "0x10040"
Country                       = "0x10041"
Region                        = "0x10042"
Site                          = "0x10043"
Sector                        = "0x10044"
Building                      = "0x10045"
Section                       = "0x10046"
Floor                         = "0x10047"
Room                          = "0x10048"
Top_Org                       = "0x102cf"
Enterprise                    = "0x102d0"
Subsidiary                    = "0x102d1"
Division                      = "0x102d2"
Department                    = "0x102d3"
Org_Section                   = "0x102d4"
Work_Group                    = "0x102d5"
Org_Owns                      = "0x102da"
Schedule                      = "0x10456"
GnSNMPDev                     = "0x3d0002"
Fanout                        = "0x100ae"
Pingable                      = "0x10290"
WA_Link                       = "0x102e2"
Unplaced                      = "0x103d8"
RTM_Test                      = "0x4560000"
SM_Service                    = "0x1046f"
SM_AttrMonitor                = "0x1046e"
SM_LatencyMon                 = "0x4500001"
```

```
SM_SLA                      = "0x4500002"
SM_Guarantee                = "0x4500003"
SM_Customer                 = "0x1046c"
SM_CustomerGroup            = "0x10477"
SM_ConnectMon               = "0x4500000"
SM_ServiceMgr               = "0x4500006"
CustomerManager             = "0x10478"
SM_Service_Mgt              = "0x4500007"
SM_SLA_Mgr                  = "0x4500008"
Correlation_Domain          = "0x10467"
Correlation_Manager         = "0x10469"
/>
<Relations
Collects                    = "0x10002"
MaintenenceScheduledBy      = "0x10034"
SlmAgreesTo                 = "0x4500000"
SlmGuarantees               = "0x4500001"
SlmHasGuarantee             = "0x4500002"
SlmIsMeasuredBy             = "0x4500003"
SlmMonitors                 = "0x4500004"
SlmOwns                     = "0x4500005"
SlmUses                     = "0x4500006"
SlmWatchesContainer         = "0x4500007"
SlmContains                 = "0x4500008"
SlaPeriod                   = "0x4500009"
SlmSchedulesGuarantee       = "0x450000c"
SlmHasServiceComponent      = "0x450000a"
SlmContainsSLAs             = "0x450000b"
Groups_Customers            = "0x1003e"
GlobalCollect               = "0x1003b"
/>
<!-- When do_not_process_pre_existing_devices_under_container_node    -->
<!-- is set to true, if a device under a container element is found   -->
<!-- pre-existing in Spectrum, Modeling Gateway will not process that -->
<!-- device, like updating attributes, creating connections, etc.     -->
<ImportConfiguration
do_not_process_pre_existing_devices_under_container_node = "false"
import_to_primary_ss_only = "false"
max_device_creation_threads = "50"
/>
<!-- ******************************************************** -->
<!--                                                          -->
<!-- This is the Modeling Gateway export configuration.       -->
<!--                                                          -->
<!-- ExportConfiguration is the configuration that controls   -->
<!-- what to export.                                          -->
<!--                                                          -->
<!-- export_devices: export device models or not              -->
<!--                                                          -->
<!-- export_containers: export container models or not        -->
<!--                                                          -->
<!-- export_port_attributes: export port attributes or not    -->
<!--                                                          -->
```

```
<!-- export_links: export device links or not                -->
<!--                                                         -->
<!-- export_topology_layout: export devices and containers   -->
<!--                          x,y coordinates or not         -->
<!--                                                         -->
<!-- export_annotation: export annotations or not            -->
<!--                                                         -->
<!-- export_WA_Link_models: export WA_Link models or not.    -->
<!--                        If it's not, WA_Link models will -->
<!--                        be treated as transparent.  Link -->
<!--                        between two device made thru a   -->
<!--                        WA_Link will be exported as a    -->
<!--                        direct link.                     -->
<!--                                                         -->
<!-- export_spectrum_settings: export SPECTRUM settings or not-->
<!--                           like the settings for Fault   -->
<!--                           Isolation, Auto Discovery,    -->
<!--                           VNM Control, etc...           -->
<!--                                                         -->
<!-- export_user_models: export SPECTRUM user modeling,      -->
<!--                     user licenses, privileges and etc.  -->
<!--                                                         -->
<!-- export_service_modeling: export SPECTRUM Service modeling-->
<!--                                                         -->
<!-- export_schedules: export SPECTRUM schedules.            -->
<!--                                                         -->
<!-- export_discovery_configs: export Auto Discovery's       -->
<!--                           configurations.               -->
<!--                                                         -->
<!-- ******************************************************* -->
<ExportConfiguration
export_devices           = "true"
export_containers        = "true"
export_port_attributes   = "true"
export_links             = "true"
export_topology_layout   = "true"
export_annotation        = "true"
export_WA_Link_models    = "true"
export_spectrum_settings = "true"
export_user_models       = "true"
export_service_modeling  = "true"
export_schedules         = "true"
export_global_collections = "true"
export_discovery_configs = "true"
export_from_primary_ss_only = "false"
export_policy_manager = "true"
/>
<!-- ******************************************************* -->
<!--                                                         -->
<!-- RootContainerToExport specifies the root container in   -->
<!-- SPECTRUM to be exported.                                -->
<!--                                                         -->
<!-- ******************************************************* -->
```

```
<RootContainerToExport model_type="Universe" model_name=""/>
<!-- ******************************************************** -->
<!--                                                          -->
<!-- DeviceExportAttributes is a list of device attributes to -->
<!-- be exported.  If the attribute ID has not been specified -->
<!-- above, "attribute_id" has to be assigned.                -->
<!--                                                          -->
<!-- ******************************************************** -->
<DeviceExportAttributes>
<name/>
<model_type attribute_id="0x10000"/>
<community_string/>
<agent_port/>
<poll_interval/>
<is_managed/>
<poll_status/>
<Security_String/>
<TimeOut/>
<TryCount/>
<Criticality/>
<Value_When_Orange/>
<Value_When_Red/>
<Value_When_Yellow/>
<Redundancy_Admin_Status attribute_id="0x11d2c" />
<Auto_Reconfigure_Interfaces attribute_id="0x11dd4" />
<Discover_Connection_After_Linkup_Trap attribute_id="0x11d25" />
<Device_Discovery_After_Reconfiguration attribute_id="0x11d27" />
<Generate_Redundancy_Alarms attribute_id="0x11dd6" />
<Create_Sub_Interfaces attribute_id="0x11f3c" />
<Topology_Relocate_Model attribute_id="0x11a80" />
<Disable_Trap_Events attribute_id="0x11cd0" />
<Enable_Spectrum_Management attribute_id="0x1295d" />
<Hibernate_Device attribute_id="0x12aca" />
<Enable_Event_Creation attribute_id="0x129f8" />
<Redundancy_Admin_Status attribute_id="0x11d2c" />
<DeviceCPUUtilization_Threshold attribute_id="0x12ab9" />
<DeviceCPUUtilization_Reset attribute_id="0x12abb" />
<DeviceCPUUtilization_Duration attribute_id="0x12bce" />
<DeviceMemoryUtilization_Threshold attribute_id="0x12aba" />
<DeviceMemoryUtilization_Reset attribute_id="0x12abc" />
<DeviceMemoryUtilization_Duration attribute_id="0x12bcf" />

</DeviceExportAttributes>
<!-- ******************************************************** -->
<!--                                                          -->
<!-- ContainerExportAttributes are the container attributes   -->
<!-- to be exported.  If an attribute ID has not been yet     -->
<!-- specified above, "attribute_id" has to be assigned.      -->
<!--                                                          -->
<!-- ******************************************************** -->
<ContainerExportAttributes>
<name/>
<Security_String/>
```

```
<subnet_address/>
<subnet_mask/>
<subnet_list/>
<Value_When_Orange/>
<Value_When_Red/>
<Value_When_Yellow/>
<SelectMP_port attribute_id="0x118e4" />
 </ContainerExportAttributes>
<!-- ******************************************************* -->
<!--                                                         -->
<!-- PortExportAttributes are the port attributes to be      -->
<!-- exported.  If an attribute ID has not been specified    -->
<!-- above, "attribute_id" has to be assigned.               -->
<!--                                                         -->
<!-- When export_changed_attribute_only = "true", only the   -->
<!-- port attribute whose value is not equal to the default  -->
<!-- value will be exported.  Otherwise, all specified port  -->
<!-- attributes will be exported.                            -->
<!--                                                         -->
<!-- ******************************************************* -->
<PortExportAttributes export_changed_attribute_only="true" >
<poll_interval/>
<poll_status/>
<ok_to_poll attribute_id="0x11dd8" />
<PollPortStatus attribute_id="0x1280a" />
<LockConnection attribute_id="0x129f1" />
<TimeOut/>
<TryCount/>
<is_managed/>
<Enable_Event_Creation/>
<Criticality/>
<Alarm_On_Link_Down_Trap attribute_id="0x11fc2" />
<Assert_Link_Down_Alarm attribute_id="0x12957" />
<Utilization_Threshold attribute_id="0x1294b" />
<Utilization_Reset attribute_id="0x1294f" />
<Utilization_Threshold_Violation_Duration attribute_id="0x12be4" />
<Inbound_Utilization_Threshold attribute_id="0x12d9f" />
<Inbound_Utilization_Reset attribute_id="0x12da0" />
<Inbound_Utilization_Threshold_Violation_Duration attribute_id="0x12da2" />
<Outbound_Utilization_Threshold attribute_id="0x12da3" />
<Outbound_Utilization_Reset attribute_id="0x12da4" />
<Outbound_Utilization_Threshold_Violation_Duration attribute_id="0x12da6" />
<Total_Packet_Rate_Threshold attribute_id="0x12da7" />
<Total_Packet_Rate_Reset attribute_id="0x12da8" />
<Total_Packet_Rate_Threshold_Violation_Duration attribute_id="0x12be3" />
<Error_Rate_Threshold attribute_id="0x1294d" />
<Error_Rate_Threshold_Reset attribute_id="0x12951" />
<Error_Rate_Threshold_Violation_Duration attribute_id="0x12be5" />
<Discarded_Threshold attribute_id="0x1294e" />
<Discarded_Threshold_Reset attribute_id="0x12952" />
<Discarded_Threshold_Violation_Duration attribute_id="0x12be2" />
</PortExportAttributes>
<SpectrumConfigurationExport model_type="VNM">
```

```
<Minimum_Disk_Space attribute_id="0x119d2" />
<Security_String/>
<Unmanaged_Trap_Handling attribute_id="0x11cce" />
<Trap_Storm_Rate attribute_id="0x122db" />
<Trap_Storm_Length attribute_id="0x122da" />
<Auto_Connects attribute_id="0x11f99"/>
<Device_Thresholds attribute_id="0x12acd" />
<Use_Full_Qualified_Host_Name attribute_id="0x12984" />
<Allow_Non_Admin_SNMP_Community_Edit attribute_id="0x12042" />
<Edit_Notes_By_Read_Only_User attribute_id="0x12043" />
<Set_isManaged_By_Read_Only_User attribute_id="0x129f3" />
<Consolidate_Users_In_Group attribute_id="0x12a1d" />
<Copy_Users_When_Copying_Group attribute_id="0x12a5e" />
<VLAN_Configuration attribute_id="0x129ad" />
<Log_When_Device_Cannot_Be_Contacted attribute_id="0x12943" />
</SpectrumConfigurationExport>
<SpectrumConfigurationExport model_type="TopologyWrkSpc">
<Create_WA_Link_Model attribute_id="0x25e0033" />
<Create_LAN_IP_Subnet_Model attribute_id="0x25e000d" />
<Create_Physical_Addresses attribute_id="0x25e000c" />
<Create_Fanout_Models attribute_id="0x25e002e" />
<Run_ATM_Discovery attribute_id="0x25e002d" />
<IP_Route_Tables attribute_id="0x25e0006" />
<Source_Addr_Tables attribute_id="0x25e0025" />
<Spanning_Tree_Tables attribute_id="0x25e0026" />
<Proprietary_Disc_Tables attribute_id="0x25e002b" />
<ARP_Tables attribute_id="0x25e003a" />
<Traffic_Resolution attribute_id="0x25e002f" />
<Unmanaged_SNMP_Disc attribute_id="0x25e0034" />
<New_Device_In_Maint_Mode attribute_id="0x25e0035" />
</SpectrumConfigurationExport>
<SpectrumConfigurationExport model_type="LostFound" >
<Automatic_Model_Destruction attribute_id="0x11de1" />
<Model_Destruction_Interval_Hours attribute_id="0x11de3" />
<Model_Destruction_Interval_Minutes attribute_id="0x11de4" />
</SpectrumConfigurationExport>
<SpectrumConfigurationExport model_type="FaultIsolation">
<ICMP_Support_Enabled attribute_id="0x11d98" />
<ICMP_Timeout attribute_id="0x11dab" />
<ICMP_TryCount attribute_id="0x11dac" />
<Lost_Device_TryCount attribute_id="0x12a0a" />
<Contact_Lost_Model_Destruction attribute_id="0x11fa8" />
<Destruction_Delay attribute_id="0x11fa9" />
<Destruction_Event_Generation attribute_id="0x11faa" />
<Router_Redundancy_Retry_Count attribute_id="0x12a09" />
<Port_Fault_Correlation attribute_id="0x129e6" />
<Unresolved_Fault_Alarm_Disposition attribute_id="0x129f4" />
<WA_Link_Fault_Isolation_Mode attribute_id="0x12adc" />
</SpectrumConfigurationExport>
<SpectrumConfigurationExport model_type="LivePipes" >
<Live_Pipe_Enabled attribute_id="0x11df9" />
<Alarm_Linked_Port attribute_id="0x11fbd" />
<Suppress_Linked_Port_Alarms attribute_id="0x11fbe" />
```

```
<Port_Always_Down_Alarm_Suppression attribute_id="0x129fb" />
</SpectrumConfigurationExport>
<SpectrumConfigurationExport model_type="AlarmMgmt" >
<Generate_Alarm_Event attribute_id="0x11f5f" />
<Add_Event_To_Alarms attribute_id="0x11f5c" />
<Use_Old_Alarm_Event attribute_id="0x11f5d" />
<Alarm_Update_by_Read_Only attribute_id="0x11f5e" />
<Alarm_Ageout_Time attribute_id="0x129ea" />
<Disable_Initial_Alarms attribute_id="0x11f5a" />
<Disable_Suppressed_Alarms attribute_id="0x11f5b" />
<Disable_Maint_Alarms attribute_id="0x11f59" />
<Alarm_Clear_By_Read_Only attribute_id="0x11fb2" />
<Ageout_Residual_Alarm_Only attribute_id="0x129ec" />
</SpectrumConfigurationExport>
<SpectrumConfigurationExport model_type="PolicyManager" >
<Policy_Distribution_Mode attribute_id="0x4ad0007" />
</SpectrumConfigurationExport>
<SpectrumConfigurationExport model_type="GlobalConfig" >
<SNMPv3Profiles attribute_id="0x12bd4" />
<HibernationCommSuccessTries attribute_id="0x12acb" />
</SpectrumConfigurationExport>
</TopologyImportExportResourceFile>
```

Model Type Editor

Introducing the Model Type Editor

This section introduces the Model Type Editor application and the database objects that you can create using it, namely, model types, attributes, relations, and meta-rules.

The section also discusses model type inheritance, which is an important concept to understand before creating and modifying model types.

> **WARNING**
> Before using the Model Type Editor, read Certifications . The *Certification space* provides information on the GnSNMPDev management module. You can use GnSNMPDev to represent a SNMP-compliant network device that lacks a corresponding DX NetOps Spectrum management module. You can also use GnSNMPDev as a toolkit to create new management modules that include new, supporting device model types and application model types.

Modeling Concepts

DX NetOps Spectrum and Model Type Editor

DX NetOps Spectrum is an integrated management system that runs on the Linux and Windows platforms. The DX NetOps Spectrum design is based on a client/server model. The server, called the SpectroSERVER, includes the DX NetOps Spectrum knowledge base, which gets and stores all network information. The client is the DX NetOps Spectrum user interface, called OneClick, which provides a graphical representation of the network environment.

The SpectroSERVER provides intelligence and contains models of the actual network devices and their relationships -- models that continuously collect data about the entities they represent and collectively provide a comprehensive management perspective of the network. Through polling, the DX NetOps Spectrum database gains extensive information about network devices, their relationships, and their performance.

The OneClick client application provides a multi-dimensional picture of the SpectroSERVER database. Users can retrieve and view the information maintained in the SpectroSERVER database, and they can invoke the SpectroSERVER to control objects on the network. Network information can be presented from various perspectives, including topological or geographical views.

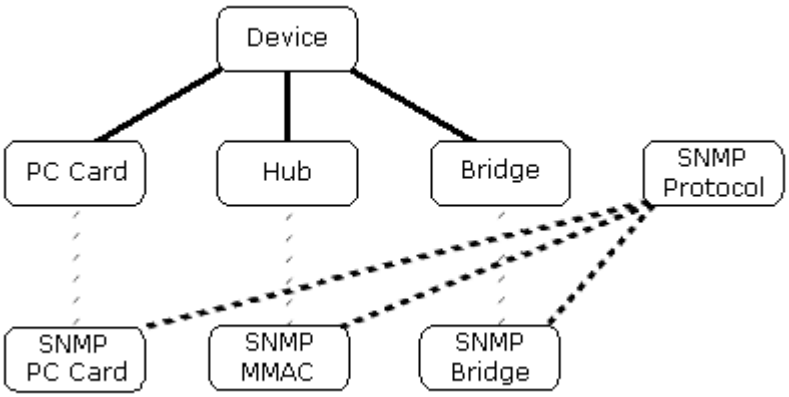
DX NetOps Spectrum deploys model types, models, attributes, relations, and meta-rules to monitor network infrastructure. *Model types* are the templates used to create models, and *models* are specific instances of a model type. (The terms "model type" and "template" are used interchangeably in DX NetOps Spectrum documentation.) *Attributes* are the characteristics of a specific model type, and *relations* and *meta-rules* define how model types interact with each other. While models and associations are defined using OneClick, the Model Type Editor lets you define model types, rules, relations, and attributes.

The Model Type Editor is the Java-based application that you use to create, modify, and delete modeling catalog objects. Because it is a thick-client application that accesses the SpectroSERVER database, you must run it from the SpectroSERVER platform.

Model Types and Attributes

A *model type* is a template that is used to create models, and it is defined by a specific set of attributes. In turn, *attributes* are the database variables that collectively characterize the real-world object represented by the model type. Model types range from simple with few attributes and relationships to very complex with many attributes and relationships.

Complex model types are often derived by inheriting attributes from several, simpler model types. The resulting combination constitutes a hierarchy of model types. Parent model types (model types from which one or more other model types have been derived) are called *base model types*. Child model types (model types that have been derived from one or more other types) are called *derived model types*. The following illustration shows a sample model type hierarchy that illustrates model type derivation.

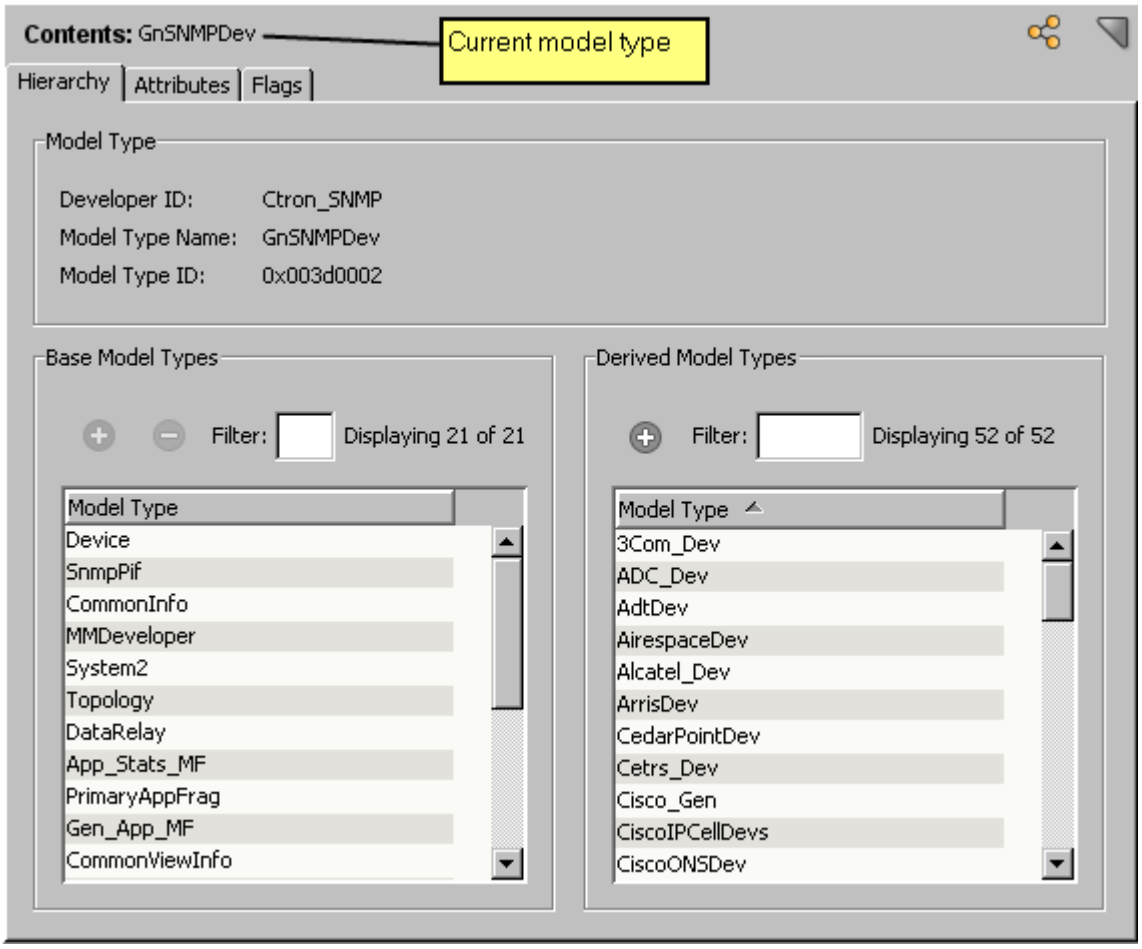


In the illustration, Device is a base model type for three model types: PC Card, Hub, and Bridge. As such, all three derived model types inherit the attributes of Device.

In turn, PC Card, Hub, and Bridge are each used as a base model type for another model type, respectively, SNMP PC Card, SNMP MMAC, and SNMP Bridge. All of these derived model types inherit the attributes of their immediate parent model type, and by extension, the attributes of the Device model type. In addition, they also inherit the attributes of a model type named SNMP Protocol.

In the Model Type Editor, you can identify a model type's position in the model type hierarchy using the Hierarchy tab. The tab shows the following for the current model type:

- The base (parent) model types from which the current model type is derived
- The derived (child) model types that are derived from the current model type



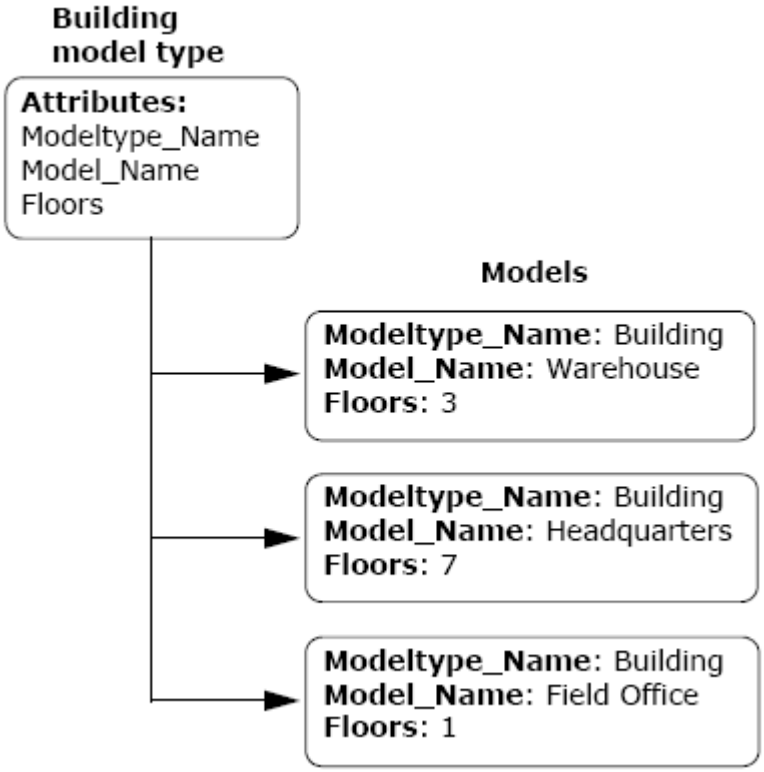
The Model Type Editor also provides an Attribute tab that shows the attributes of a model type and their default values.

There are various rationales for attributes. Certain attributes, such as the name of a router or its IP address, are necessary in order to uniquely identify a resulting model within the DX NetOps Spectrum database. Other attributes relate a model to the SNMP object identifiers (OIDs) supported by the model. Still others support DX NetOps Spectrum functionality. For example, Discovery_Precedence is used by the Discovery program; Value_When_Red is used for alarm roll-up, and Polling_Interval is used to determine how often DX NetOps Spectrum polls the given device for information.

Models

A *model* is an instantiation of a model type by the SpectroSERVER. All models instantiated from the same model type have the same set of attributes and DX NetOps Spectrum intelligence. However, the values for the attributes are unique to each model except in the case of Shared attributes (that is, attributes for which the Shared flag is set).

As a simple example, the following illustration shows three models derived from a model type named Building, which has three attributes: Modeltype_Name, Model_Name, and Floors. Because each model is instantiated from the same model type, each model has the same value for Modeltype_Name (a Shared attribute). However, the values for Model_Name and Floors vary.



NOTE
A network administrator can add, edit, and delete models using OneClick. For more information, see Modeling and Managing Your IT Infrastructure.

Relations and Meta-Rules

A *relation* is a database construct that defines a specific type of relationship between model types. The following are examples of relations:

- Contains
- Collects
- Executes
- HASPART
- Is_Adjacent_to
- Monitors
- Pings_Through
- Schedules

Each relation is defined by one or, more typically, several rules called meta-rules. The *meta-rules* for a relation apply the relation to specific model types, thereby defining how the model types can interact with one another. As an example, the following are several of the meta-rules defined for the Contains relation:

- Building Contains Floor
- Building Contains Room
- Building Contains Section
- Country Contains Building
- Country Contains Region
- Country Contains Site

Each meta-rule must have exactly one antecedent model type (the left-hand entry) and exactly one predicate model type (the right-hand entry). Any model type can be an antecedent or a predicate for any number of meta-rules.

OneClick uses the meta-rules for model types to establish the rules for interaction between specific models instantiated from the model types. In DX NetOps Spectrum terms, instantiation of a meta-rule between two models produces an *association* between the models, and each model can react to the knowledge that it is associated with the other model. For example, consider the following meta-rule:

Country Contains Building

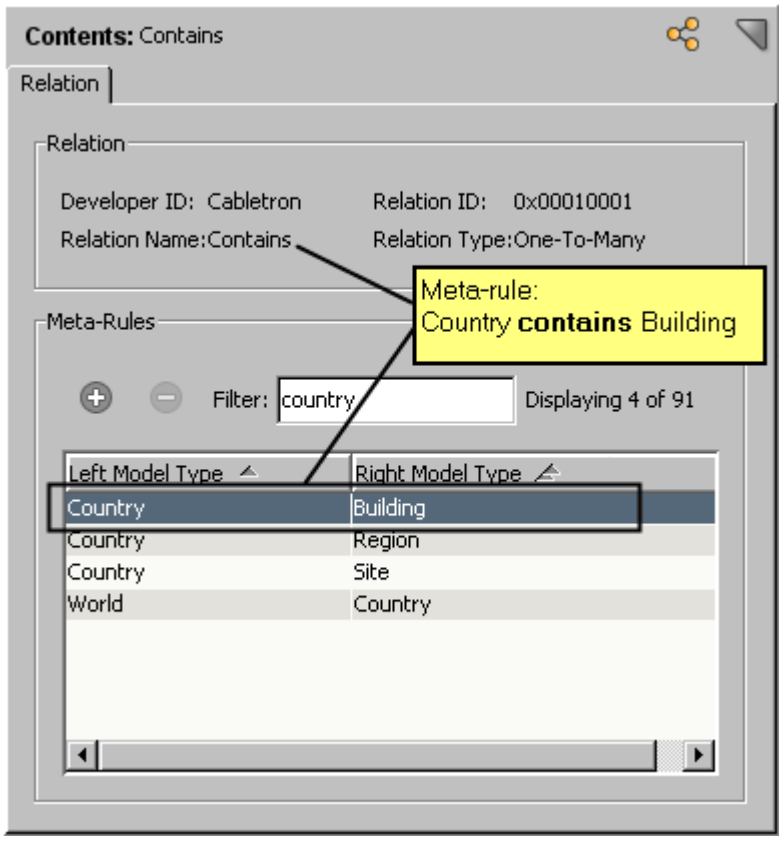
This might result in the following association between two specific models:

France contains Corporation001

There are two types of relations:

- **One-To-Many**
  A relation of this type defines how a single model of one model type relates to multiple models of another model type. For example, Contains is a one-to-many type of relation. As such, it can have one or more meta-rules that define one-to-many relationships between models. For example, it includes a "Country contains Building" meta-rule, which specifies that a *single* model of the Country model type can contain *multiple* models of the Building model type.
- **Many-To-Many**
  A relation of this type defines how multiple models of one model type relate to multiple models of another model type. For example, Is_Adjacent_to is a many-to-many type of relation. As such, it can have one or more meta-rules that define many-to-many relationships between models. For example, it includes a "Device Is_Adjacent_to Device" meta-rule, which specifies that *multiple* models of the Device model type can be adjacent to *multiple* models of the Device model type.

The Model Type Editor provides a Relation tab that shows the meta-rules defined for a specific relation.



Most relations in the modeling catalog that comes with the basic DX NetOps Spectrum package have meta-rules that specify the model types to which the relations can be applied.

Model Type Inheritance

When you derive new model types from existing model types, the specifics of model type and attribute inheritance are important to understand because the functionality of each model type depends on its inheritance.

This section provides information on important concepts related to inheritance, namely, attribute descriptors, standard versus specialized hierarchies, model type precedence, and attribute collapsing.

Attribute Descriptors

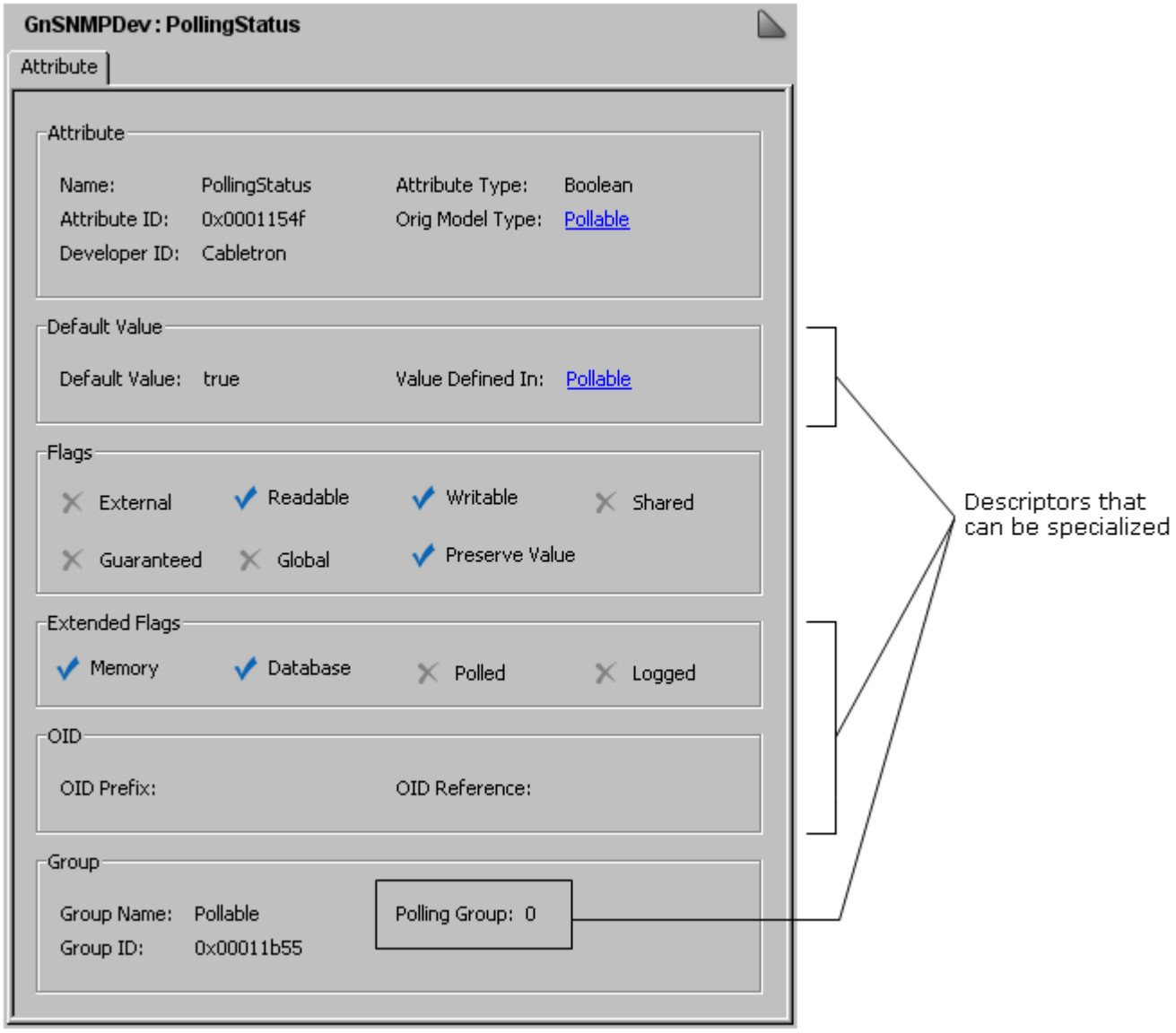
Attribute descriptors are the characteristics that define the attribute, for example, its type (boolean, integer, text string, and so on) and default value.

An attribute has two types of descriptors:

- **Standard descriptors**: The values of these attribute descriptors are inherited by all model types derived directly or indirectly from the base model type in which the attribute was first created (referred to as the *originating model type*). You can only modify the values of standard descriptors in the originating model type; you cannot modify their values in derived model types.
  Name, Attribute ID, and the Shared flag are three examples of standard descriptors.
- **Descriptors that can be specialized**: Like standard attribute descriptors, the values of these descriptors are inherited by all model types derived directly or indirectly from the base model type in which the attribute was first created. However, unlike standard descriptors, you can modify the values of these descriptors at any level in the model type hierarchy. This process is called *specialization*.

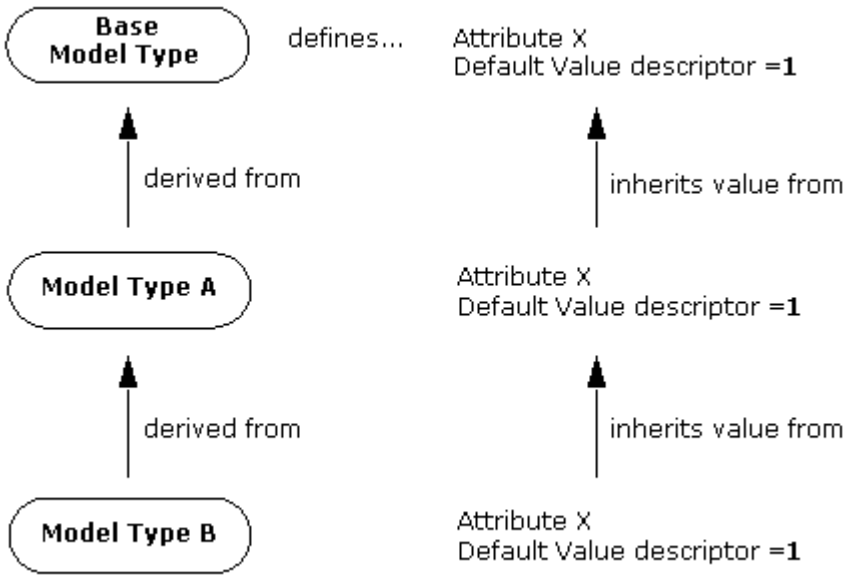
When you modify a descriptor value in a derived model type, the new value overrides the inherited value for that derived model type and any of its own derived (child) model types. Other model types that are derived from the same base model type -- and that are unspecialized with respect to the same attribute descriptor -- continue to inherit their descriptor values; this is also the case for any of their derived (child) model types.

WARNING
The complete list of descriptors that you can specialize includes -- notably, Default Value -- as well as the Extended Flags (Memory, Database, Polled, and Logged), OID Prefix, OID Reference, and Polling Group, as identified in the following image.



Standard Hierarchy

The following illustration shows a model type hierarchy whose attributes and their default values are inherited through a standard hierarchical sequence.



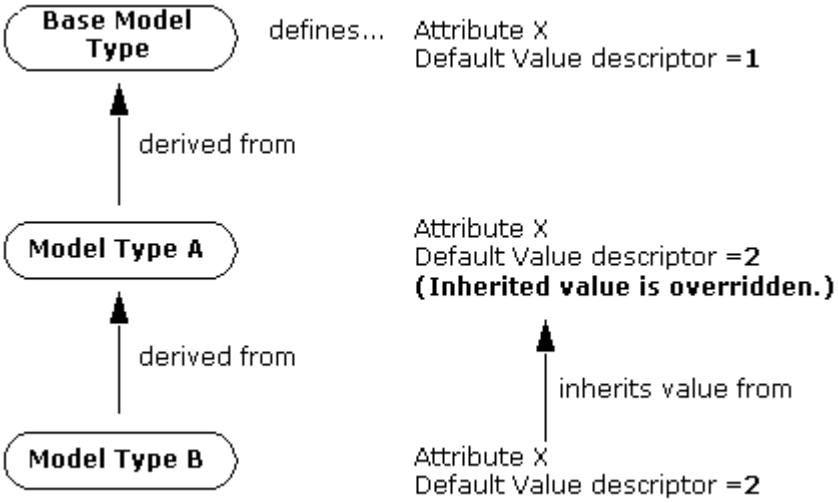
In the illustration, Model Type A is derived from Base Model Type, and, as a result, Model Type A inherits attribute X and its default value of 1 from Base Model Type. In a similar manner, Model Type B is derived from Model Type A, and, as a result, Model Type B inherits attribute X and its default value of 1 from Model Type A.

This hierarchical relationship of inheritance is applied to all other model types derived from Model Type A, Model Type B, or any of their descendants. Moreover, the relationship is maintained by the database so that any change made to the value of the Default Value attribute descriptor in the originating model type is immediately applied to all derived model types that inherit the descriptor value.

As a simple example, assume you add a Technical_Assistance text string attribute to Base Model Type with a telephone number as the default value. Model Type A, Model Type B, and all other model types that derive from them would all inherit the Technical_Assistance attribute and, therefore, the value of the Default Value attribute descriptor (the telephone number). If you then changed the telephone number, all of the model types derived directly or indirectly from Base Model Type would immediately inherit the new telephone number.

Specialized Hierarchy

In contrast to the standard hierarchy, the following illustration shows a model type hierarchy where one derived model type inherits its default value but another does not.

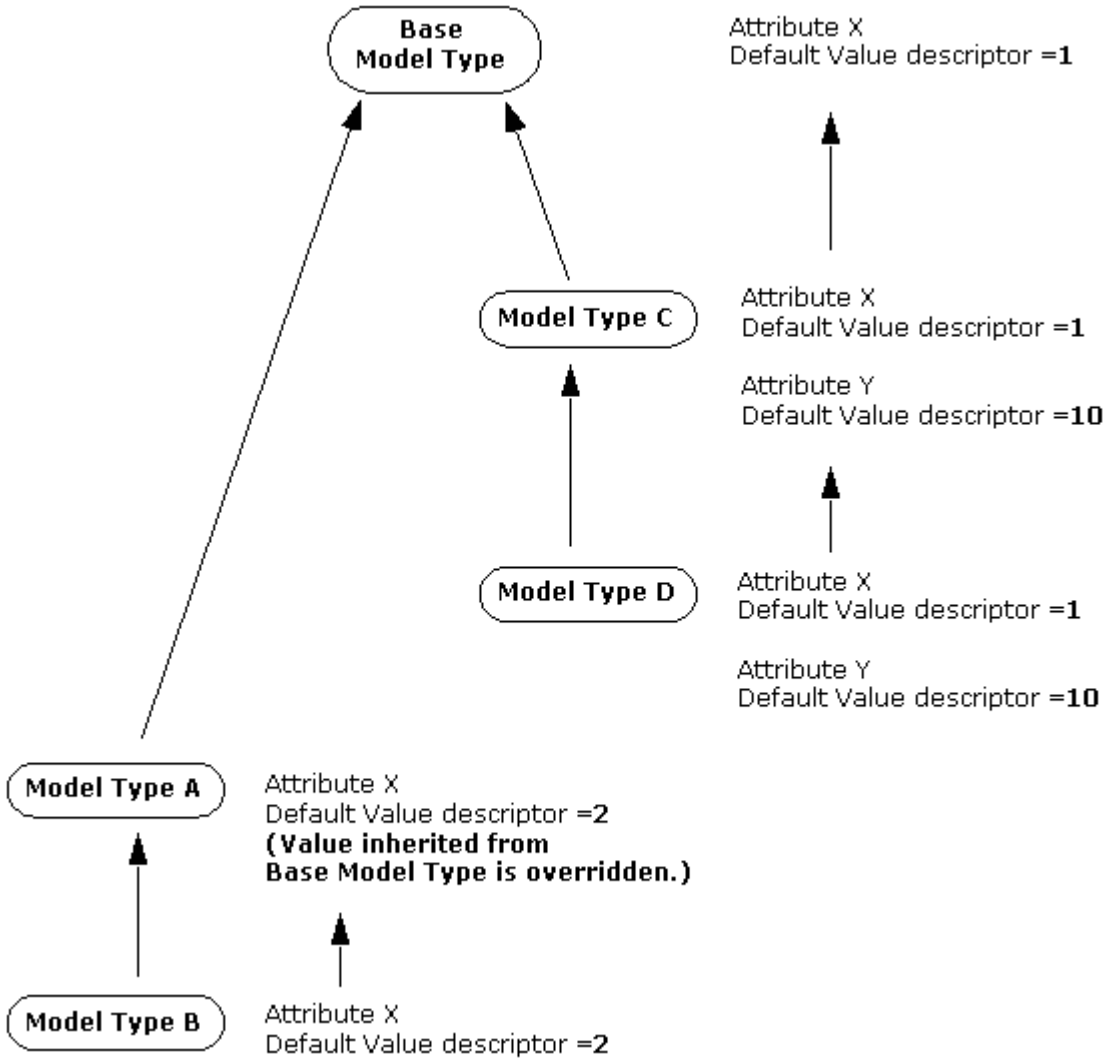


In this illustration, the default value for attribute X (or, more specifically, the value of the Default Value attribute descriptor for attribute X) in Model Type A is specified, not inherited. In other words, the value inherited from Base Model Type is overridden and no longer used. As mentioned previously in this section, this process is called *specialization*.

When you override a descriptor value that is inherited, you break the relationship to the base (parent) model type *for that attribute descriptor only*. The inheritance relationship with respect to the values of all other attribute descriptors is not affected. Furthermore, model types that are derived from specialized model types behave exactly as if they were derived from unspecialized model types. In the illustration, Model Type B inherits its default value for attribute X from Model Type A just as it would if Model Type A were not specialized. If you were to change the default value of attribute x in Model Type A, the default value of the same attribute in Model Type B would immediately change as well. Because the inheritance relationship is in effect, this would be the case even if you derived Model Type B before you made the change.

In reality, the attribute hierarchy often consists of a combination of inherited and specialized values, particularly when multiple model types are derived from a common base model type. Moreover, while specialization is most typical with respect to the Default Value attribute descriptor, it is possible for any other descriptor that can be specialized. The

following diagram illustrates a model type hierarchy that uses both inheritance and specialization for an attribute (attribute



X).

Note that, in the figure that Model Type C is specialized by introducing new attribute Y. Therefore, this attribute is inherited by derived Model Type D.

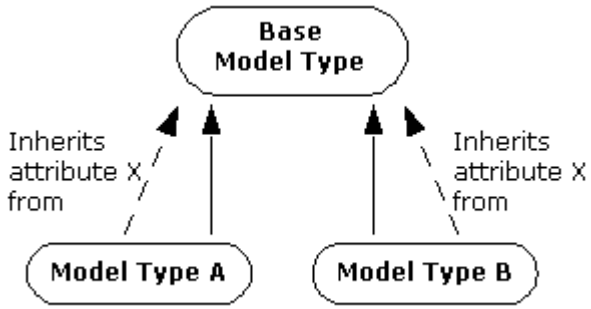
Model Type Precedence

The process of inheriting attributes from base model types means that a derived model type can inherit the same attribute from two or more inheritance paths that have a common originating model type. To avoid ambiguity in this situation, the base model types are ranked according to the order in which they are added as base model types for a derived model type, and this model type ranking determines the inheritance path to use when a derived model type can inherit an attribute from multiple paths. More specifically, the derived model type inherits the attribute from the base model type with the *lowest* ranking.

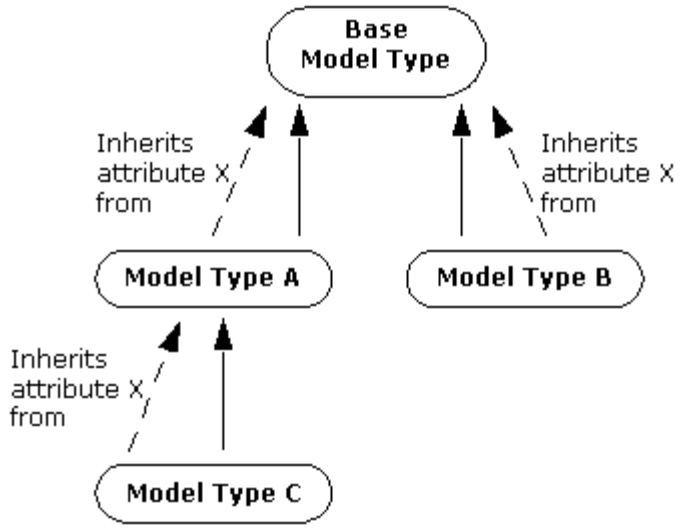
When you first derive a model type from a base model type, that base model type is given a ranking of 1. If you then add a second base model type to the derived model type, that second base model type is given a ranking of 2. Subsequent base model types are assigned rankings in a similar manner.

As an example, consider the following model type derivation workflow:

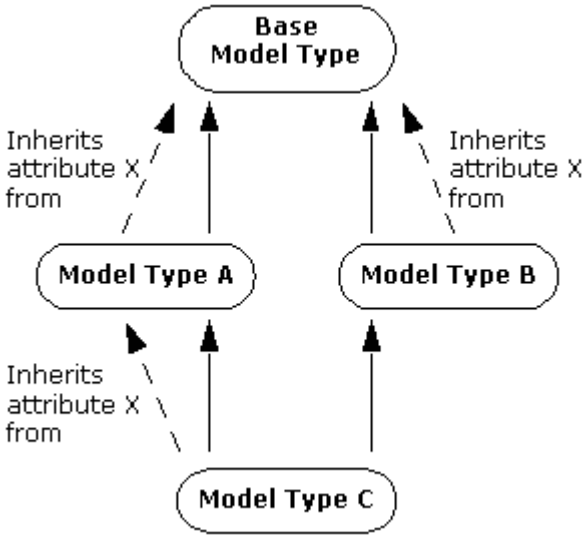
1. You derive model type A and model type B from a common base model type.



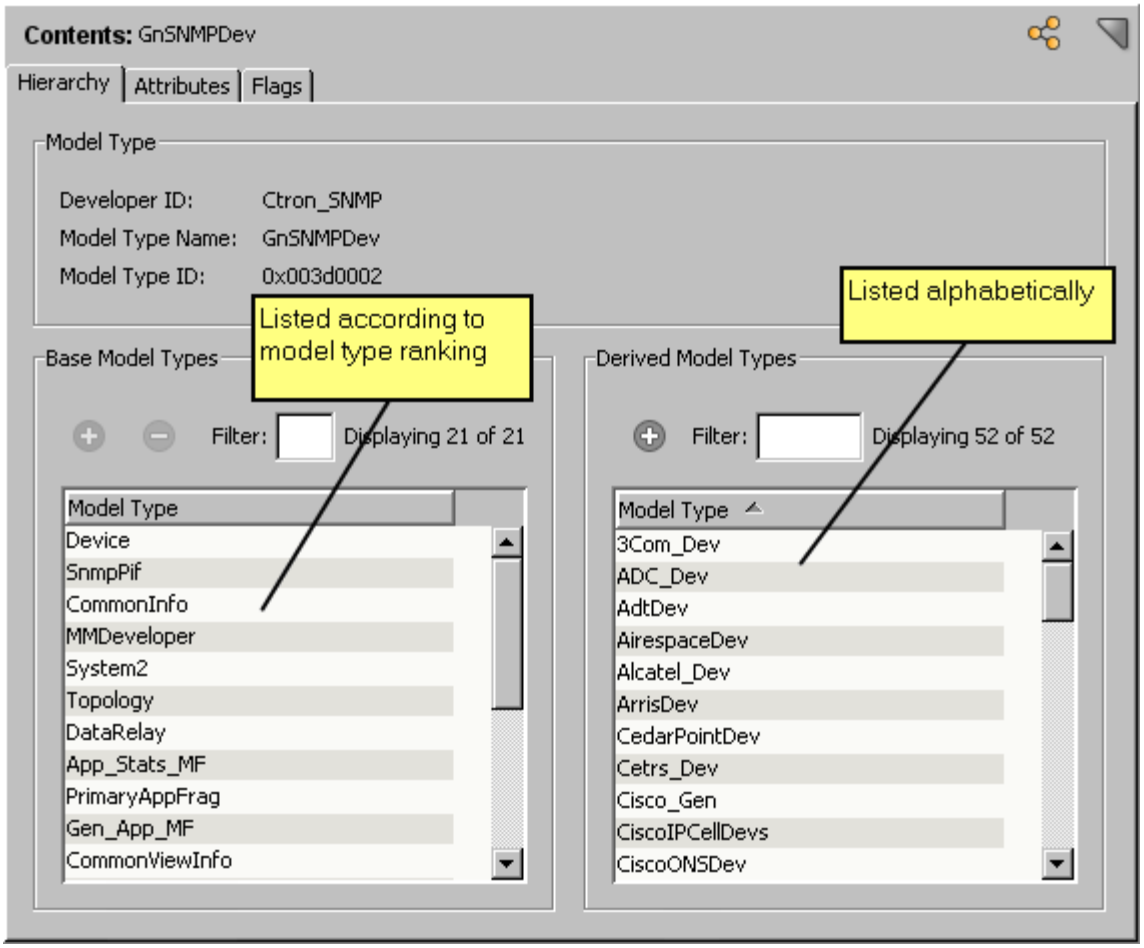
   The base model type defines attribute X, which is inherited by both derived model types. The base model type is assigned a ranking of 1 with respect to both derived model types.
2. You derive model type C from model type A.



   Model type C inherits attribute X from model type A, which, in turn, inherits the attribute from the base model type. Because model type C was created based on model type A, model type A is assigned a base model type ranking of 1 with respect to derived model type C.
3. You add model type B as a base model type of model type C.



Model type B is assigned a base model type ranking of 2 with respect to derived model type C. Because model type A has a lower ranking as a base model type, model type C inherits attribute X from model type A, not from model type B. In the Model Type Editor, the base model types for a given model type are listed in ranked order, so you can identify the order of precedence for attribute inheritance.



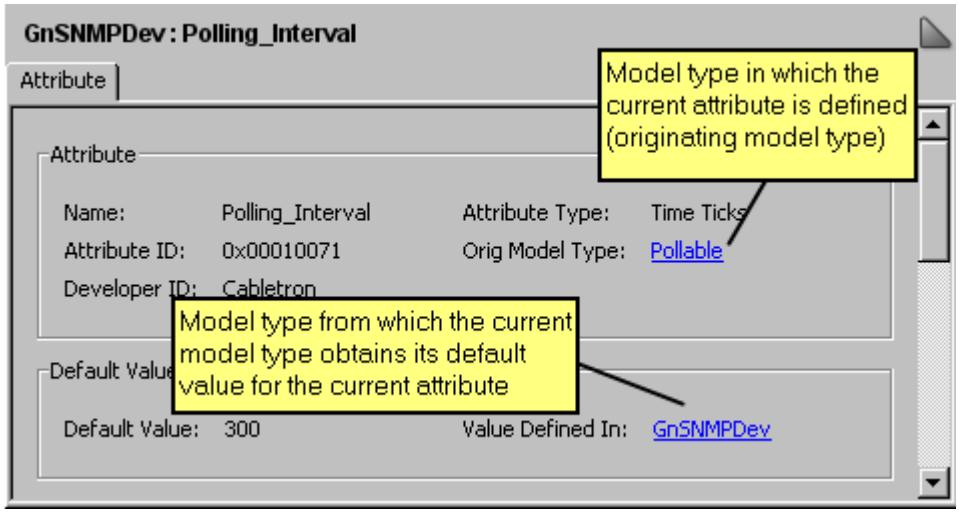
NOTE
You can change a base model type's ranking by removing it and then re-adding it as a base model type in a specific location in the list.

Attribute Collapsing

To maintain an attribute hierarchy that is as simple as possible, the Model Type Editor includes a feature that eliminates unnecessary specialization. That is, if you modify the Default Value descriptor (or any other descriptor that can be specialized) in a derived model type so that its value *matches* the value in the corresponding descriptor in the base model type from which it immediately inherits the attribute (that is, from an immediate parent model type), the database discards the modification and instead inherits the descriptor value from the base model type. This process is called "attribute collapsing;" its goal is to simplify a specialized hierarchy as much as possible by removing unnecessary customizations.

Once the descriptor value reverts to the inherited value, if you subsequently change the descriptor value in the base model type, the derived model type also receives the change (which is normal inheritance behavior).

You can identify the model type from which the current model type obtains its default value in the Value Defined In field in the Component Detail panel, as shown in the following image.



NOTE
Attribute collapsing does not take place for default values if the attribute is Shared (that is, the Shared flag is set).

Getting Started with the Model Type Editor

Before using the Model Type Editor, we recommend that you also read Certification . The section provides information about the GnSNMPDev management module provided with DX NetOps Spectrum. You can use GnSNMPDev to represent an SNMP-compliant network device that does not have a corresponding DX NetOps Spectrum management module. You can also customize GnSNMPDev to extend its support, or you can use it as a toolkit to create new management modules that include new, supporting device model types and application model types.

Using a Developer ID

DX NetOps Spectrum uses developer IDs to help ensure that objects such as model types that are created by users and application integrators have unique identifiers and, therefore, can be distributed to other users without conflict.

There are two types of developer IDs:

- **Registered developer IDs**
  This is a unique, CA-assigned developer ID that protects a developer's model types, attributes, relations, and meta-rules from being edited by other users.
- **The default developer ID**
  This is the developer ID applied when the current user is not registered with CA as a DX NetOps Spectrum developer; the code designation is DF, with a Developer ID value of 0xffff.

Whenever you create a model type, attribute, or relation, the Model Type Editor uses the developer ID that is currently active (loaded in the database) to create the ID or handle for the new object. This ID is either your unique, registered developer ID or the default ID.

The Model Type Editor also uses developer IDs to determine the access privileges of users with respect to creating, modifying, and destroying modeling catalog objects (model types, attributes, relations, and meta-rules).

- Any developer can use the Model Type Editor as the default developer. However, the modeling catalog objects created using the default developer ID are not protected.
- To obtain a developer ID from CA, contact Support.
- To be issued a developer ID, you must have purchased the Level 1 toolkit.

> **NOTE**
> For more information about the toolkit, see Spectrum Integrator .

You can activate your developer ID by loading the developer information file that contains the ID into the SpectroSERVER database. This is a *one-time* operation after you have initialized the database. However, if you reinitialize the database, you must reload the information.

> **NOTE**
> Activate your developer ID using SSdbload with the -d option. For more information, see the discussion about loading developer information in Administrating

Access Privileges With Developer ID

Your developer ID determines your access privileges with respect to model types, attributes, relations, and meta-rules.

If you work in the Model Type Editor using the default developer ID, you can:

- View, create, modify, delete, and export model types and attributes that were created with the default developer ID. This includes modifying the derivation of model types by adding and removing base model types.
- View model types that were created by other registered developers.
- Create, modify and delete relations and meta-rules that were created with the default developer ID.
- Export model types, attributes, relations, and meta-rules that were created with the default developer ID.
- Import model types, attributes, relations, and meta-rules from another compatible database.
- Import a Management Information Base (MIB) text file.

If you work in the Model Type Editor using a registered developer ID, you can:

- View, create, modify, or delete model types and attributes that were created with your registered developer ID. This includes modifying the derivation of model types by adding and removing base model types.
- Create, modify, and delete relations and meta-rules that were created with your registered developer ID.
- Export model types, attributes, relations, and meta-rules that were created with your registered developer ID.
- Import model types, attributes, relations, and meta-rules from another compatible database.
- Import a MIB text file.

> **WARNING**
> We recommend all users register with CA as a DX NetOps Spectrum developer to obtain a registered developer ID.

A registered developer can export model types, attributes, relations, and meta-rules that were created using a registered developer ID. This not only removes ID conflicts, but also protects the objects from accidental or deliberate modification. As indicated previously, if you use the default developer ID, you can export only the model types, attributes, relations, and meta-rules that were created with the default developer ID. It is probable that the IDs (handles) of these objects will conflict with those on another system to which you might want to export the objects. In addition, the exchange of such objects between systems using the default developer ID means that the objects are susceptible to modification or corruption on the receiving system.

SpectroSERVER Database Protection

When a program or process such as the Model Type Editor accesses the SpectroSERVER database, a soft lock file named .VNMDB.LOCK is created. The lock file is a safety feature that protects the database by restricting access to one CA-developed application or process at a time. Because non-CA applications or tools may not check for this lock, exercise care when using these applications to prevent concurrent access to the SpectroSERVER database; this can result in the corruption of the database.

While lock files are removed automatically during normal shutdown of a DX NetOps Spectrum application, an abnormal shutdown can leave behind a lock file erroneously. In rare situations like this, you can manually remove the database lock. For information about how to do this, see Database Management .

Use of the Model Type Editor involves several risks to the SpectroSERVER database, such as the accidental destruction of necessary model types, the inappropriate setting of attribute flags, and the creation of more than one database with different model type derivations. For this reason, it is recommended that you adopt the following strategies to help preserve the database:

- Avoid editing the database that you use to model your network until absolutely necessary. First test your model type changes using a test database.
- Restrict the use of the Model Type Editor to individuals who are familiar with the long-term plans for your model type derivation scheme. This can help to prevent unnecessary modifications to the database.
- Do not permit editing across multiple databases by more than one user using the same developer ID. This practice creates conflicts between the IDs of modeling catalog objects, which can only be corrected by manually recreating the affected objects. If two separate databases are being used, verify that the database files are being modified with different developer IDs.
- Use the database management utilities provided with DX NetOps Spectrum namely SSdbload and SSdbsave to initialize and copy the database. You may get unpredictable results if you use another method.

  NOTE
  For information about using these utilities, see *Database Management* .

Considerations on Database Migration

It is recommended that you record all changes that you make with the Model Type Editor because some changes are not migrated when the database is updated to a later version of DX NetOps Spectrum. Specifically, if you change attributes (for example, flag settings), the model type hierarchy, or relations and associated meta-rules in the model types supplied by CA or another developer (vendor), most likely the changes will not be migrated when the database is upgraded, and you will need to reapply them manually.

To preserve the *default values* of attributes, you can enable the Preserve Value attribute descriptor flag on the relevant attributes. This flag prevents changed default values from being overwritten by subsequent database updates. However, be aware that enabling this flag may prevent you from receiving intended changes pertaining to the same attributes.

About Starting the Model Type Editor

To start the Model Type Editor for the first time, obtain a registered developer ID. You can activate the ID by loading the information into the SpectroSERVER database.

NOTE
For more information, see Database Management .

WARNING
Only one application or process can access the SpectroSERVER database at a time. As a result, after you start the Model Type Editor, all other DX NetOps Spectrum applications, including the SpectroSERVER, are denied access. While CA-developed applications automatically deny access to other CA applications as needed, be aware that some third-party applications do not. Database corruption can result.

In rare situations, the SpectroSERVER database is not closed properly by a process, for example, during a power failure. In these situations, the database lock erroneously remains in effect and prevents you from starting the Model Type Editor. To use the Model Type Editor, you must have read and write permissions to the files in the <*$SPECROOT*>\SS directory.

NOTE
For information about removing a database lock, see the Database Management .

Start Model Type Editor from the Control Panel

You can start the Model Type Editor from the Control Panel.

Follow these steps:

1. Stop the SpectroSERVER if it is running.
2. Verify that no other programs that can access the SpectroSERVER database are running.
3. Open the DX NetOps Spectrum Control Panel and click Configure, Model Type Editor.
   The Model Type Editor opens. The Root model type is set as the current model type. The Root model type is the model type at the highest point in the model type hierarchy.

Start Model Type Editor from the Command Line

You can also start the Model Type Editor from the command line.

Follow these steps:

1. Stop the SpectroSERVER if it is running.
2. Verify that no other programs that can access the SpectroSERVER database are running.
3. Log in to a shell environment if you are running on Unix or Linux, or open a command prompt if you are running on Windows.
4. Change to the directory that contains the SpectroSERVER database that you want to modify using the Model Type Editor.

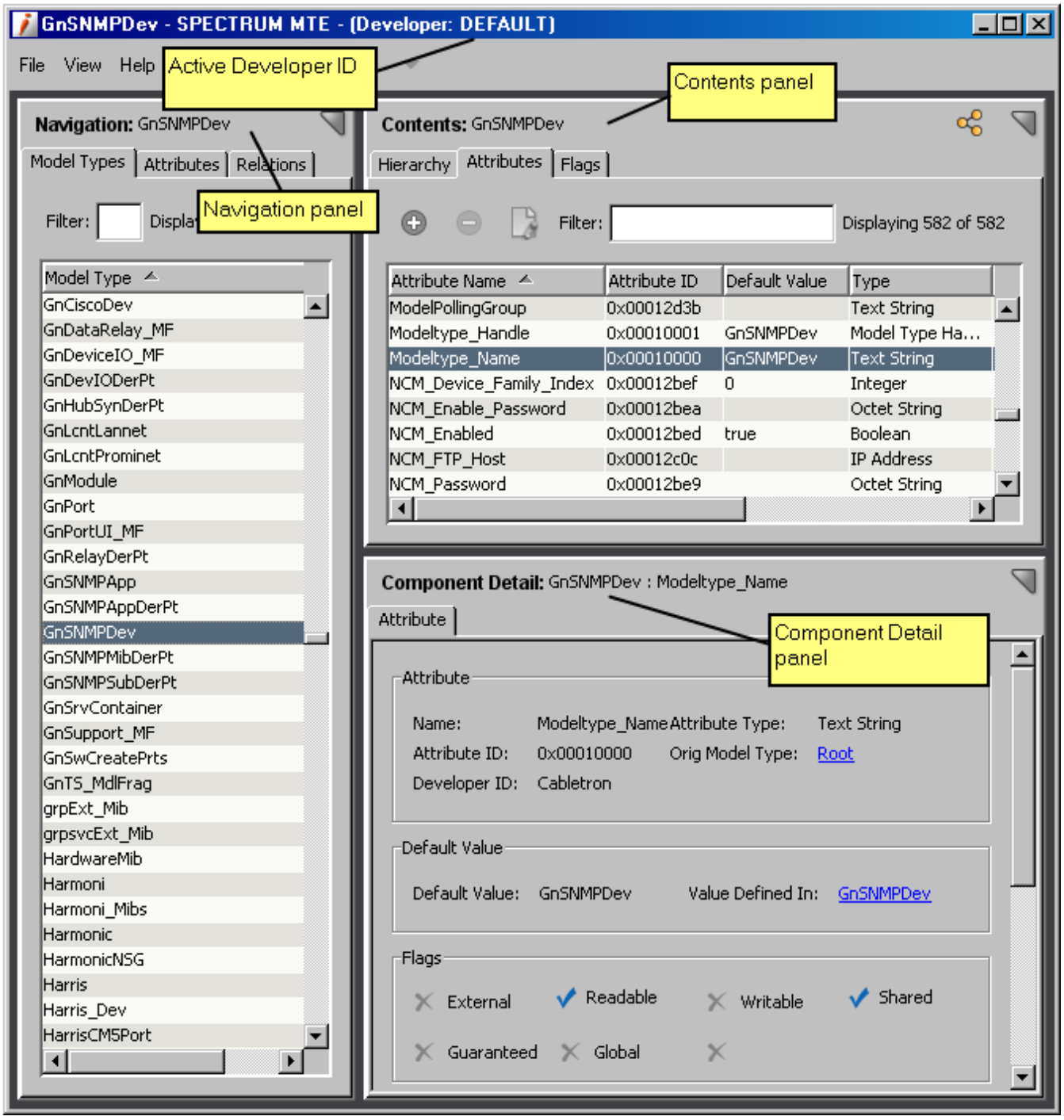
   NOTE
   The executable file for the Model Type Editor is installed in <*$SPECROOT*>/SS-Tools, but it must be called from the directory that contains the SpectroSERVER database that you want to modify. Typically, this directory is DX NetOps Spectrum/SS. The directory should contain the database files, which consist of paired *.db and *.ix files, miscellaneous files, and supporting subdirectories.
5. Enter the following command:

   ```
   ../SS-Tools/mte
   ```

   The Model Type Editor opens. The Root model type is set as the current model type. The Root model type is the model type at the highest point in the model type hierarchy.

Overview of the User Interface

The following image identifies the main work areas, or panels, in the Model Type Editor user interface.



Use the Navigation panel on the left to search for and select a model type, attribute, or relation. Information about a selected object appears in the Contents panel on the right. Use the Contents panel to modify the object, delete the object, or create related objects (for example, to derive a new model type from the current model type).

When you work with attributes, a Component Detail panel lets you view and modify the descriptor values of the current attribute.

Commit Changes to the SpectroSERVER Database

While you work in the Model Type Editor, you work with a temporary cache of database objects called the *working catalog*. The Model Type Editor adds only the database objects that are required for your activities to the working catalog.

The Model Type Editor automatically saves any changes that you make to model types, attributes, relations, and meta-rules to the working catalog. You can add them to the permanent catalog in the SpectroSERVER database on demand, or you can add them when you exit the application. You are prompted to add them when you exit the application to avoid losing your work.

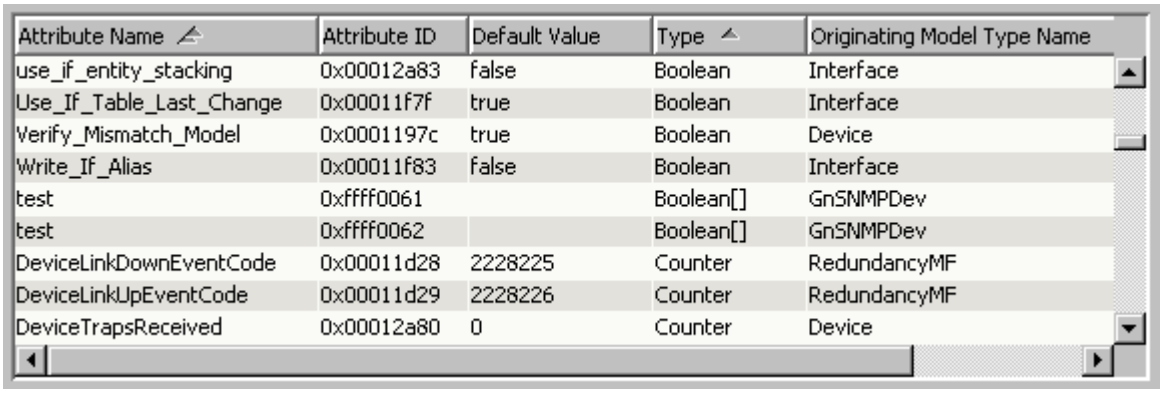
You can commit changes to the SpectroSERVER database on demand.

Follow these steps:

1. Click File, Commit to Database.
   A confirmation dialog opens.
2. Click OK.
   Your changes are saved in the permanent catalog in the SpectroSERVER database. As a result, subsequent calls to the database for the affected objects retrieve the updated versions of those objects.

Sorting and Filtering Lists

To make it easier to work with large lists of modeling catalog objects (for example, model types or attributes), you can sort the lists in tables in ascending or descending order by clicking any column header. You can also sort lists using multiple column headers. For example, the following image shows a list of attributes first sorted in ascending order by Type and then sorted in ascending order by Attribute Name, as indicated by the icons in the column headers.



Attribute Name	Attribute ID	Default Value	Type	Originating Model Type Name
use_if_entity_stacking	0x00012a83	false	Boolean	Interface
Use_If_Table_Last_Change	0x00011f7f	true	Boolean	Interface
Verify_Mismatch_Model	0x0001197c	true	Boolean	Device
Write_If_Alias	0x00011f83	false	Boolean	Interface
test	0xffff0061		Boolean[]	GnSNMPDev
test	0xffff0062		Boolean[]	GnSNMPDev
DeviceLinkDownEventCode	0x00011d28	2228225	Counter	RedundancyMF
DeviceLinkUpEventCode	0x00011d29	2228226	Counter	RedundancyMF
DeviceTrapsReceived	0x00012a80	0	Counter	Device

You can also use the Filter and Search text boxes provided on the various tabs in the Model Type Editor to display only the catalog objects whose names or IDs include a specific character string regardless of case (uppercase or lowercase). In the Filter text boxes, this limits the list of already displayed names. In the Search text box for attributes, this narrows the search criteria applied against the working catalog.

Keep in mind the following as you filter and search for modeling catalog objects:

- To filter lists by ID (for example, model type ID), the ID column must be displayed in the table.
- The Search text box for attributes always searches using both the attribute name and attribute ID as criteria.
- In most cases, a filter or search criterion remains in effect until you clear it by deleting the character string. There are a few actions that automatically clear a filter, for example, when you add a new attribute to a model type.

Add and Remove Columns from Tables

You can modify the information that is displayed in any table in the Model Type Editor by adding and removing columns from the table.

Follow these steps:

1. Right-click the table heading.
   The Table Preferences dialog opens.
2. Click the Columns tab, and select the columns you want to display.
3. (Optional) Change the table sorting and font using the controls on the Sort and Font tabs.
4. Click OK.

Exit the Model Type Editor

When you close the Model Type Editor, you are prompted to save any changes that you have made to the working catalog. Saving them propagates the changes to the SpectroSERVER database.

> **NOTE**
> For information about the difference between the working catalog and the permanent catalog, see Commit Changes to the SpectroSERVER Database.

Follow these steps:

1. Click File, Exit.
   A confirmation dialog opens.
2. Click OK.
   If you have made changes to the working catalog that is stored in cache, you are prompted to save the changes to the permanent catalog in the SpectroSERVER database.
3. Do *one* of the following:
   – To save the changes to the permanent catalog, click Yes.
   – To discard the changes, click No.

   The changes are saved, if appropriate, and the Model Type Editor is closed.

Creating and Modifying Model Types

Creating and Modifying Model Types

This section provides information about how to do the following:

- Extend and customize the default modeling catalog provided with DX NetOps Spectrum by creating and modifying model types.
- Import MIBs.
- Create and manage attribute groups.
  Attribute groups make it easier to work with logically related attributes in the Model Type Editor.

Attributes of Model Types

A model type is defined by the following attributes and classes of attributes:

- Developer ID
- Model type name
- Model type ID (handle)
- Base model types
- Derived model types
- Flags
- Custom attributes

Developer ID

The developer ID that was active when the attribute was created. This can be a registered ID obtained from CA or the default ID. Once a developer ID is assigned to an attribute, it cannot be changed.

After you create an attribute, the access privileges of users for modifying, deleting, and exporting it are determined based on whether the active developer ID matches the developer ID associated with the attribute.

Model Type Name

A descriptive identifier that typically describes the model type's function. Model type names should be a maximum of 128 characters and should only consist of letters, numbers, underscore (_) characters, and dashes (-). Spaces, punctuation, or other symbols should not be used.

> **NOTE**
> A model type name does not need to be unique across the modeling catalog, but you should help ensure it is unique across the model types created under a given developer ID. DX NetOps Spectrum differentiates model types using both the model type name and the developer ID component in the model type ID.

While you can rename a model type, be aware that this affects the AlertMap file that is specific to the model type because the file is located in the following directory:

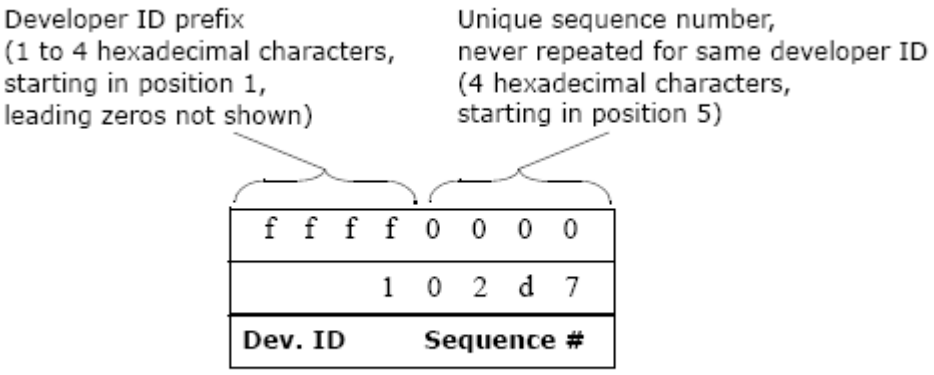
```
SS/CsVendor/<developer name>/<model type name>
```

If you rename a model type, you will need to manually create a new directory based on the new model type name and then move the AlertMap file.

Also be aware that it is possible -- although unusual and discouraged -- that an inference handler depends on the name of a model type rather than its model type ID (handle).

Model Type ID (Handle)

A unique ID that is assigned to the model type when the model type is created. The ID is generated by ORing together the value of the active developer ID and a counter value in the range from 0x0001 and 0xFFFF, as shown in the following illustration.



Structural Makeup of the Model Type ID

Once a model type ID is assigned to a model type, it cannot be modified. Additionally, if the model type is deleted, the ID is not reused.

Base Model Types

A ranked list of parent model types from which the current model type directly inherits attributes and SpectroSERVER intelligence.

Because a model type can inherit the same attribute from two or more inheritance paths that have a common originating model type, the ranking of the base model types is used to determine the inheritance path.

You cannot remove attributes inherited from a base model type, and you have limited capabilities to edit them. To remove an inherited attribute, or to have full editing capabilities, you must use the same developer ID that was active when the model type was created, and you must make the change in the originating model type. The originating model type is the model type in which the attribute was created.

Derived Model Types

An alphabetically sorted list of child model types that are directly derived from the current model type. That is, the derived model types inherit attributes and SpectroSERVER intelligence from the current model type -- and by extension -- from the base model types of the current model type.

Model Type Flags

There are several model type flags that control basic characteristics of a model type.

> **NOTE**
> You can only modify flag settings for a model type if you are using the developer ID that was active when the model type was created.

The following describes each model type flag:

- **Visible**
  If enabled (checked), the model type is exposed in the output when you run a report on the modeling catalog using the reports database utility provided with DX NetOps Spectrum. If disabled (not checked), the model type is only exposed in the output if the model type was creating using the developer ID that is currently loaded in the database. The default value is enabled.

  > **NOTE**
  > The Visible flag does not affect what a user can view in the Model Type Editor or in OneClick, nor does it affect the operation of the SpectroSERVER. However, future releases of DX NetOps Spectrum may respect

the flag and stops access to model types and models when the flag is disabled. For more information about the reports utility, see Database Management .

- **Instantiable**
  If enabled (checked), models of the model type can be created in the SpectroSERVER database by users and inference handlers. If disabled (not checked), models of the model type cannot be created, and the model type is not available in OneClick when creating a model by model type. The default value is disabled.
  Changing the condition of the Instantiable flag for a model type does not affect existing models of that type. For example, if you were to disable the flag after creating models A and B, models A and B would be unaffected. However, you could not create additional models after disabling the flag.
- **Derivable**
  If enabled (checked), the model type can be used as a base model type from which other model types can be derived. If disabled (not checked), the model type cannot be used as a base model type. The default value is enabled.
  DX NetOps Spectrum checks the Derivable flag only as new model types are being created. In other words, if you derive several model types from model type A and then disable the Derivable flag for model type A, the newly derived model types are unaffected, but you are no longer able to derive additional model types from model type A.
- **No Destroy**
  If enabled (checked), models of the model type *cannot* be deleted from the SpectroSERVER database by users and inference handlers. If disabled (not checked), models of the model type can be deleted. The default value is disabled.

  NOTE
  You can only enable this flag if you also enable the Instantiable flag.

- **Unique**
  If enabled (checked), only one model of the model type can exist in the SpectroSERVER database. If disabled (not checked), additional models of the model type can exist. The default value is disabled.
  DX NetOps Spectrum checks the Unique flag only as models are being created. If you create several models of a model type and then disable the Unique flag, the previously created models are unaffected, but you are no longer able to create additional models of the model type.

  NOTE
  You can only enable this flag if you also enable the Instantiable flag.

- **Required**
  If enabled (checked), the SpectroSERVER creates a model of the model type at server startup if a model does not already exist. If disabled (not checked), a model of the model type is created only if requested by a user, inference handler, or application. The default value is disabled.

  NOTE
  You can only enable this flag if you also enable the Instantiable flag.

Custom Attributes

In addition to the model type attributes described earlier in this section (such as model type ID), a model type has many other attributes that are inherited from base (parent) model types or that originate in the model type itself. You can view this list of attributes on the Attributes tab in the Contents panel, as shown in the following image.

You can click the Attribute Name column header to sort the attribute list alphabetically in ascending or descending order.

The Originating Model Type column displays the model type in which each attribute was created.

You can click the Originating Model Type column header to sort the list alphabetically in ascending or descending order. This lets you group together and, therefore, identify all of the attributes in the current model type that originate in the model type itself or in a base (parent) model type.

The values for the attribute descriptors of the selected attribute are displayed in the Component Detail panel.

Standard Attribute Descriptors

Standard Attribute Descriptors

As discussed earlier in this section, every attribute is described by a set of characteristics called *attribute descriptors*. The following are attribute descriptors that you can only modify in the originating model type (the model type in which the attribute was defined):

Developer ID

The developer ID that was active when the attribute was created. This can be a registered ID obtained from CA or the default ID. Once a developer ID is assigned to an attribute, it cannot be changed.

After you create an attribute, the access privileges of users for modifying, deleting, and exporting it are determined based on whether the active developer ID matches the developer ID associated with the attribute.

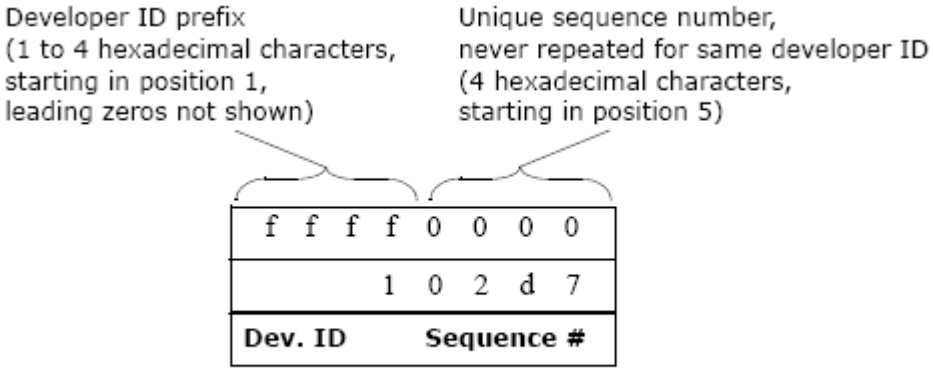
Attribute Name

An attribute name is a descriptive identifier. Attribute names should be a maximum of 128 characters and should only consist of letters, numbers, underscore (_) characters, and dashes (-). Spaces, punctuation, or other symbols should not be used.

> **NOTE**
> An attribute name does not need to be unique across the modeling catalog, but it should be unique across the attributes in the model type that were created using the same developer ID. In other words, within a single model type, two attributes can have the same name if they were created using different developer IDs.

Attribute ID (Handle)

A unique ID that is assigned to the attribute when the attribute is created. The ID is generated by ORing together the value of the active developer ID and a unique sequence number (counter value), as shown in the following illustration.



Structural Makeup of the Attribute ID

Once an attribute ID is assigned to an attribute, it cannot be modified. Additionally, if the attribute is deleted, the ID is not reused.

Type

The attribute type defines the kind of data that the value of the attribute can hold. If the attribute represents a managed object that is defined in the MIB, you should set the type to correspond to the type defined in the MIB.

If an attribute requires a list of values, you must select the List check box.

> **NOTE**
> After you create an attribute, you cannot change its type or whether it stores a single value or a list of values. You must delete the attribute and create a new one of the desired type. There are two exceptions to this. First, you can change the type from Octet String to Text String, and vice versa. Second, if the type is numeric (Integer, Counter, Enumeration, Gauge, or Time Ticks), you can change from one of these numeric types to another.

Flags

Flags -- also referred to as descriptor flags -- inform the SpectroSERVER of the characteristics of the attribute, for example, who can access the attribute's value. The following describes each flag.

- **External**
  Indicates the attribute value is maintained outside of the SpectroSERVER and that update of the value is done either at a polling interval or upon user request.

  NOTE
  If you set this flag, you must supply an OID prefix, which specifies the location of the managed variable in the MIB. Also note that if you set this flag, you cannot set the Shared flag.
- **Readable**
  Informs the SpectroSERVER that a client or other application can read the attribute value from the SpectroSERVER database.
  If you set the External flag, you should set this flag in accordance with the MIB definition of the Readable variable for the attribute. Otherwise, you can set this flag as desired.
- **Writable**
  Informs the SpectroSERVER that a client or other application can write the attribute value to the SpectroSERVER database.
  If you set the External flag, you should set this flag in accordance with the MIB definition of the Writable variable for the attribute. Otherwise, you can set this flag as desired.
- **Shared**
  Declares that only one value exists for the attribute and that all models of the given model type share the same value. The value is *not* duplicated for each model in memory or in the database.

  NOTE
  You can only set this flag if you also set the Database flag or the Memory flag. Also, if you set the External flag or the Polled flag, you cannot set this flag.
- **Guaranteed**
  Guarantees that the attribute will continue to exist and can be used in future model type derivations. Once set, this flag cannot be disabled except by the developer who created the attribute.

  NOTE
  Setting this flag only guarantees the presence of the attribute, not its value or values.

  If this flag is disabled, any Model Type Editor user can enable (set) or disable the Extended flags of the attribute. If this flag is enabled, users having developer IDs other than the one used to create the attribute can only set the Extended flags; they cannot disable them. The user having the developer ID used to create the attribute can set or disable the Extended flags at any time.
- **Global**
  Indicates that the attribute's value will be kept consistent across duplicate models in all landscapes in a distributed SpectroSERVER (DSS) environment.
  Global attributes are only maintained for models of the User and UserGroup model types. These are duplicate model types, that is, across a distributed environment, multiple models of these types effectively represent the same user or user group. As such, a change to a model in one landscape should be propagated to all corresponding, duplicate models in all other landscapes.

  NOTE
  You can only set this flag if you also set the Memory flag or the Database flag.
- **Preserve Value**
  Indicates that imported files will not overwrite the attribute's default value currently stored in the database.
  If you customize the default values of one or more model types to meet specific requirements, and then you set this flag, your customizations (the specialized default values) will remain in place when the model types are updated by subsequent versions of DX NetOps Spectrum.

If you are the owner of the attribute (that is, the attribute was created using the developer ID that is currently active), you can modify all of the flags described in this section *in the originating model type* regardless of whether you are the owner of the associated model type.

If you are not the owner of the attribute, or if the attribute is inherited, you cannot modify these flags. However, you can modify the attribute's extended flags.

Group Name and Group ID

A *group* is a logical collection of related attributes in a model type. Groups make working with related attributes easier because they let you to define and use a user-defined sorting mechanism in the Model Type Editor. You can create groups, assign attributes to them, and then add the Group Name or Group ID as a column header in the table of attributes on the Attributes tab in the Contents panel. This lets you to then click the column header to quickly group together and view together all of the attributes within a group.

Group Name specifies the name of the group to which the attribute is assigned, and *Group ID* specifies the ID of that group.

Like for any standard descriptor, you can change the group to which an attribute is assigned if you are modifying the model type in which the attribute was created, and if you are using the developer ID that was used when the attribute was created (that is, you own the attribute).

By default, an attribute has a Group ID value of 0x00000000, which indicates the attribute is not assigned to a group. If you assign an attribute to a group, and you subsequently decide it should not be assigned to a group, you must reassign the attribute to the Root group. You cannot restore a Group ID value of 0x00000000 in the Model Type Editor.

> **NOTE**
> If you set an attribute's Group Name to the name of another developer's attribute group, and that developer's attribute group is not distributed with the next version of DX NetOps Spectrum, the attribute's Group Name value is reset to <no group>, and its Group ID value is reset to 0x00000000 to indicate it is not assigned to a group.

Special Attribute Descriptors

Special Attribute Descriptors

As discussed earlier in this section, every attribute is described by a set of characteristics called *attribute descriptors*. The following are attribute descriptors that you can specialize; you can modify them at any level of the inheritance hierarchy:

Default Value

The initial value or values for the attribute. An attribute can inherit its default value from a base model type or specify its own default value (a process called specialization). In the latter case, all model types derived from the specialized model type inherit the changed attribute value.

> **NOTE**
> While you can specify a default value for an attribute using this attribute descriptor of a model type, the actual value or values of the attribute are often different in models. This is definitely the case for external attributes (attributes for which the External flag is enabled), since these attributes maintain their values by polling devices at specified intervals or making updates upon user request.

Extended Flags

Extended flags -- also referred to as extended descriptor flags -- inform the SpectroSERVER of additional characteristics of the attribute, for example, whether the attribute should be polled.

You enable and disable an attribute's extended flags using the Model Type Editor, but the flags are used by the SpectroSERVER.

If an attribute's Guaranteed flag is disabled, any Model Type Editor user can enable or disable the extended flags. If the Guaranteed flag is enabled, users having developer IDs other than the one used to create the attribute can only enable the Extended flags; they cannot disable them. The user having the developer ID used to create the attribute can enable or disable the extended flags at any time.

The following list describes each extended flag.

- **Memory**
  Stores a copy of the attribute's value in memory. When the SpectroSERVER is restarted, the value is reset to the default value; the value in memory is not preserved.

  NOTE
  You must set either this flag or the Database flag if you set either the Shared flag or the Global flag.
- **Database**
  Stores a copy of the attribute's value in the database so that it is preserved across SpectroSERVER restarts.

  NOTE
  You must set either this flag or the Memory flag if you set either the Shared flag or the Global flag.
- **Polled**
  Informs the SpectroSERVER that the attribute should be polled at the polling interval in order to update its value. This is only meaningful if the External flag is also set. If the Memory flag is also set for the attribute, the value retrieved by the poll is also stored in memory.

  NOTE
  You cannot set this flag if you set the Shared flag.

  If you set this flag, you should assign the attribute to an appropriate polling group; all attributes of a polling group are polled together. If you set both this flag and the Logged flag, you must group the attribute with other attributes that also have both the Polled and Logged flags set.
- **Logged**
  Causes the value of the attribute to be recorded in the Distributed Data Manager (DDM) database. If you enable this flag, you should assign the attribute to an appropriate polling group; all attributes of a polling group are logged together. If you enable both this flag and the Polled flag, you must group the attribute with other attributes that also have both the Polled and Logged flags enabled.

  NOTE
  Logging occurs at a user-specified interval stored in the Poll_Log_Ratio attribute. By default, this attribute is set to 0 for device model types, which effectively disabled DX NetOps Spectrum's native logging method. If you require the logging of device, attribute, and port statistics, it is recommended that you use [assign the value for sslog in your book] instead of the native method, the latter of which writes the information to the Distributed Data Manager (DDM) database. [assign the value for sslog in your book] is a DX NetOps Spectrum command-line application that lets you to log statistics directly to ASCII files, which reduces the load on the DDM database and eliminates the need to export the data. Of equal importance, [assign the value for sslog in your book] gives you greater control over what data to log and how frequently to log it. For more information, see the *[assign the value for sslog in your book]* User *section.*

External attributes that are set to be polled and logged may return "noSuchName" errors when a management module is based on a more current firmware version than the managed device supports. To reduce unnecessary network traffic, DX NetOps Spectrum automatically suspends normal polling and logging for attributes that return this error, and moves the attribute to the unsupported polling attribute group.

Once an attribute has been moved to the unsupported polling attribute group, DX NetOps Spectrum generates an event (0x10970). By default, this event is not logged and does not generate an alarm. However, you can change this event processing using the Event Configuration application in OneClick. DX NetOps Spectrum attempts to read the attribute at

the interval specified by the unsupported_attr_poll_interval. By default, the value of the unsupported_attr_poll_interval is 12 hours. This value can be changed by manually adding the parameter and the desired interval (in seconds) to DX NetOps Spectrum's .vnmrc file.

When an attribute that had previously been reporting a "noSuchName" error reports a successful poll, DX NetOps Spectrum generates an event (0x10971). By default, this event is not logged and does not clear an alarm. However, you can also change this event processing using the Event Configuration application.

Thus, the unsupported_attr_poll_interval lets normal polling and logging for an attribute to resume automatically without requiring a SpectroSERVER restart or the destruction and recreation of the models that have the attribute.

> **NOTE**
> For more information about the unsupported_attr_poll_interval, see the *Distributed SpectroSERVER* Administrating *section*. For more information about the Event Configuration application, see the Event Configuration *section.*

OID Prefix and OID Reference

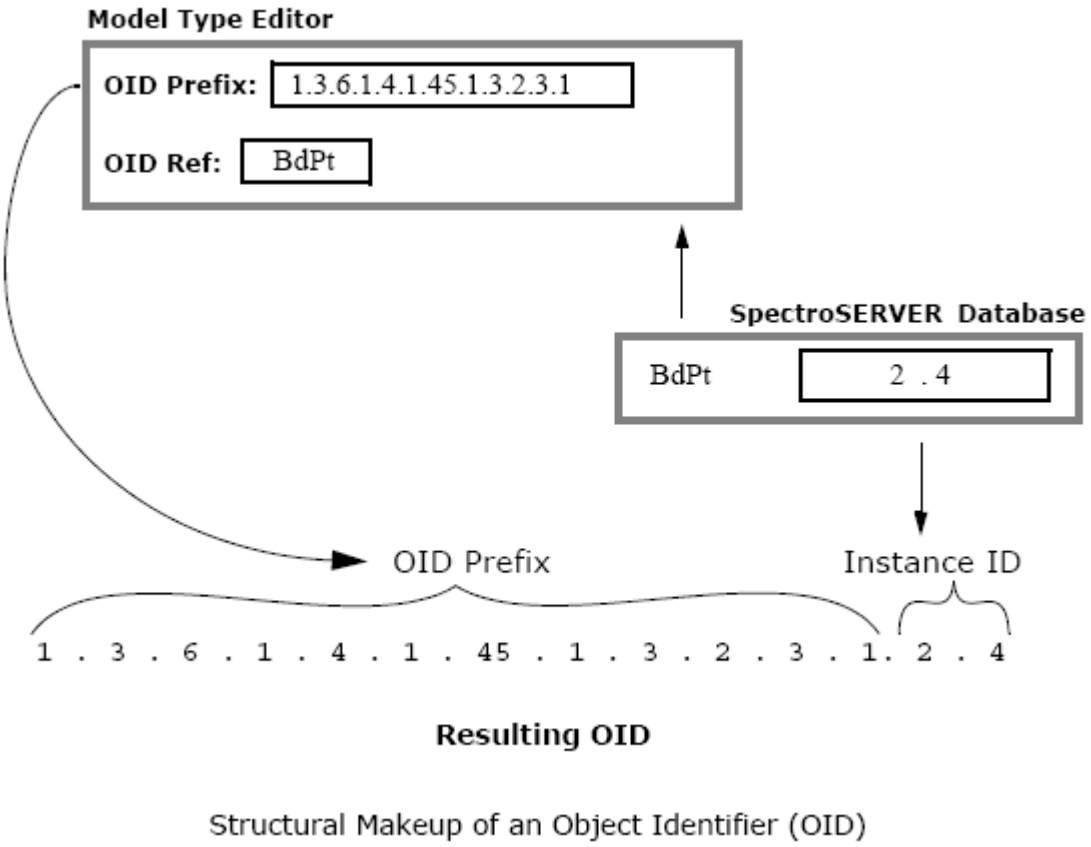
The OID Prefix descriptor and OID Reference descriptor apply only to attributes that have the External flag enabled, that is, attributes that represent managed variables within tables in a MIB.

The *OID Prefix* specifies the column within the MIB table that contains the variable. You must use dotted-decimal notation when entering the OID prefix.

The *OID Reference* (instance ID) specifies the name of an attribute whose value serves as an index used to define the instance of the variable within the column.

The OID prefix is concatenated with the OID reference to define a complete object identifier (OID) for the variable being monitored. As an example, the following image shows the resulting OID formed using the following:

- An OID Prefix set to the OID for EnetPortColls
- An OID Reference set to the ID of an internal attribute, defining the instance ID.



Structural Makeup of an Object Identifier (OID)

NOTE
A managed variable does not always require an OID reference. You can enter a complete OID with instance ID in the OID Prefix field.

Polling Group

The polling group to which the attribute belongs. All of the attributes in a group are polled together and logged together. The SpectroSERVER polls groups of pollable attributes by polling group, one at a time beginning with the group with the lowest number.

Specify a number between 0 (zero) and 255 inclusive. If you do not specify a value for a new or existing attribute, the attribute is automatically given a value of 0.

When attributes are polled but not logged (that is, the Polled flag is enabled, but the Logged flag is disabled), the only limit to the number of attributes within a polling group is the transmission length limits imposed by the following:

- The transmission protocol (Ethernet, FDDI, and so on)
- The management protocol (CA proprietary, SNMP, and so on)

Search for and Display Model Types

You can search for a model type in the working catalog by name or ID.

Follow these steps:

1. In the Navigation panel, click the Model Types tab.
   The names of all of the model types in the modeling catalog are listed.

2. In the list, select the name of the model type that you want to examine.
   To locate and select a specific model type, you can take the following steps:
   – Enter a text string in the Filter text box to filter the list to include only the model types whose names or IDs contain the string. To filter the list by ID, the ID column must be displayed in the table.
   – Click the Model Type bar at the top of the list to change the alphabetical sorting from ascending to descending or vice versa.

   The selected model type becomes the current model type, and information about it is displayed in the Contents panel.
3. To navigate to a base (parent) or derived (child) model type of the current model type, take the following steps:
   a. Click the Hierarchy tab in the Contents panel.
   b. Double-click a base or derived model type to make it the current model type.

      NOTE
      You can filter the list of base or derived model types using the Filter text boxes.
   c. Repeat the preceding step as many times as needed to navigate to the desired model type in the model type hierarchy.

Search for and Display Attributes

You can search for an attribute in the working catalog by name or ID.

Follow these steps:

1. In the Navigation panel, click the Attributes tab.
2. In the Search text box, enter a text string to examine against the attribute names and IDs. (You do not need to display the ID column in the table to search by ID.)
   The attributes with names or IDs that include the string that you entered are displayed in the list. In addition, the *originating model type* for each attribute is displayed. The originating model type is the model type where the attribute was created. Use this model type to modify all of the descriptors for an attribute.
3. Select the name of the attribute that you want to examine from the list.
   The corresponding originating model type is made the current model type in the Contents panel, and information about the attribute you selected is displayed in the Component Detail panel.

Create a Model Type

When you want to represent a new device or some other entity that is not currently defined as a model type in the DX NetOps Spectrum database, you must create a model type.

Follow these steps:

1. Determine the attributes that are required for the new model type.
2. Identify the base model types from which the new model type can inherit its attributes or inherit as many of them as possible.
3. Set the existing model type that has *most* of the attributes that you need for the new model type as the current model type.
   This is the model type from which you will directly derive the new model type.
4. Click the Derive a new Model Type

   icon  .

   NOTE
   If you are not able to click the button, verify that the current model type has its Derivable flag set. You cannot derive a model type from the model type unless this is the case.

   The Create Derived Model Type dialog opens.

5. For Name, enter the name of the new model type.
   The name should be a maximum of 128 characters and should only consist of letters, numbers, underscore characters (_), or dash characters (-).

   WARNING
   A model type name is not required to be unique across the modeling catalog. However, we recommend using unique names across the model types that were created under the currently active developer ID. DX NetOps Spectrum differentiates model types using both the model type name and the developer ID component in the model type ID. In addition, reusing a model type name is not recommended.
6. Click OK.
   The new model type is created and is set as the current model type.
   If you click the Attributes tab, you can examine its attributes, which are those inherited from the base model type that you selected in step 1.
7. If the new model type requires additional attributes that can be inherited from other base model types, add those model types as base model types.
8. If the new model type requires additional attributes that cannot be inherited from other base model types, add those attributes directly to the new model type.
9. Set the model type flags for the new model type as appropriate.
   For example, if the new model type is a final model type (that is, it is meant to be instantiated and used by models represented in OneClick), set the Instantiable flag.
10. Restart the OneClick web server.
    You can now create a model of this new model type in the OneClick console.

Delete a Model Type

Deleting a model type involves deleting all models of that model type, removing all derived model types for the model type, and finally removing all base model types for the model type. In effect, this completely removes all dependencies on the model type from the model type hierarchy so that the model type can safely be destroyed in the database.

Follow these steps:

1. In OneClick, use the Locater tab to find all of the models of the model type you intend to delete, and delete the models.

   NOTE
   For more information, see Modeling and Managing Your IT Infrastructure .
2. Shut down OneClick and the SpectroSERVER, and start the Model Type Editor.
3. Set to current the model type that you want to delete.
4. Examine the hierarchy of the model type in order to identify the consequences of deleting it. For example, check whether any attributes that originate in the model type are critical to any derived model types that inherit them. Then resolve any predictable problems with inheritance factors.
5. Remove all derived model types from the model type that you want to delete:
   a. Make the first derived model type the current model type instead.
      The model type you want to delete is now displayed in the list of base model types for the new current model type.
   b. In the list of base model types, select the model type that you want to delete, and click the Remove selected base Model Type
      icon  .

      NOTE
      The tooltip displayed when you hover over the button indicates whether this action will result in only the removal of the selected model type as a base model type or also the *deletion* of the current model type. If the current model type has no derived model types, it will also be deleted when you remove the last of its base model types because this means it is no longer a part of the model type hierarchy.

A confirmation dialog opens.

c. Click Yes.

d. Repeat the preceding three steps as many times as needed until the model type that you want to delete is no longer being used as a base model type for any model types.

6. Remove all base model types from the model type that you want to delete:

a. Set current the model type that you want to delete.

b. In the list of base model types, select the first model type, and click the Remove selected base Model Type 

NOTE
As previously mentioned, if the current model type has no derived model types, it will also be deleted when you remove the last of its base model types.

A confirmation dialog opens.

c. Click Yes.

d. Repeat the preceding two steps until the model type that you want to delete no longer has any base model types.

NOTE
Removing the last base model type will also delete the model type that you want to delete. You can only remove the last base model type from the current model type when the current model type has no derived model types.

Working with Base Model Types

If a model type requires additional attributes above and beyond those inherited from the model type from which it is first derived, you can add additional base model types. This can add attributes and inference handlers to the model type.

While the addition of attributes typically does not cause any problem, you may add inference handlers that lock attributes, which may make old inference handlers fail. Also, be aware that if a model type has two or more base model types that share a common ancestor model type, the model type has more than one way to inherit attributes and intelligence originating in that common ancestor. As described in Model Type Precedence, DX NetOps Spectrum resolves this type of situation by assigning rankings to base model types. The base model type with the *lower* ranking (that is, the base model type that is *higher* in the list of base model types) is the base model type from which the derived model type inherits the shared attribute.

NOTE
You can change a base model type's ranking by removing the base model type and re-adding it in a specific location in the ranked list of base model types.

How to Determine the Base Model Types for a New Model Type

To identify the model types that you want to use as base model types, use the Hierarchy tab to navigate up and down through the model type hierarchy to specific model types, and then use the Attributes tab to examine their attributes. Continue this process until you have identified the following:

- The existing model type that contains *most* of the attributes needed for the new model type. You should derive the new model type directly from this model type.
- The other model types that can provide some or all of the other attributes needed for the new model type.

As you identify the base model types for the new model type, keep the following guidelines in mind:

- You can only use derivable model types as base model types, that is, model types that have the Derivable flag set.
- Use as few base model types as possible in order to keep the hierarchy simple.
- Avoid adding base model types that do not contribute significantly to the model type being created. And avoid base model types that contain a significant number of unnecessary attributes. Because you cannot remove inherited attributes, ignoring this guideline can quickly add an excessive number of attributes, wasting storage space and perhaps affecting performance.
- You might want to add one or more base model types that provide access to specialized MIB attributes and intelligence. We recommend creating a set of MIB-specific model types, each one containing the intelligence and attributes related to a specific MIB. Create a new model type based on GnSNMPMibDerPt and give it a name that identifies associated MIB. Then import the MIB into the model type. This approach lets you add the MIB-specific model type as a base model type to multiple model types. The associated attribute IDs remain the same across all derived model types.

  WARNING
  For new device model types, the GnSNMPDev model type is often the best starting point. This model type contains the basic attributes and intelligence that are typically required for integration with core DX NetOps Spectrum functionality. For new application model types, select from several possible starting points, such as GnSNMPMibDerPt and GnSNMPAppDerPt. For more information, see the Certifications .

Add a Base Model Type to a Model Type

You can add a base model type to an existing model type.

Follow these steps:

1. Set current the model type for which you want to add a base model type.
2. Do one of the following:
   – To give the new base model type the highest ranking of all of the listed base model types (that is, place the base model type last in the list), proceed to the next step.
   – To give the new base model type a lower ranking (that is, place the base model type higher in the list), select the base model type directly *beneath* the location where you want to insert the new base model type. This step adds the new base model type in that location.

     NOTE
     When a derived model type can inherit the same attribute from two or more inheritance paths that have a common originating model type, the derived model type inherits the attribute from the base model type with the lowest ranking.
3. Under Base Model Types, click the Add a base Model Type icon 
   The Select New Base Model Type dialog opens.
4. Select the model type to add as a base model type.

   NOTE
   To rapidly locate and select a specific model type, enter a text string in the Filter text box to filter the list.
5. Click OK.
   The selected model type is added as a base model type of the current model type.

Remove a Base Model Type from a Model Type

Removing a base model type from a model type is the way in which you remove inherited attributes, inherited meta-rules, or intelligence from the model type. Essentially, this removes the hierarchical relationship between the model type and the base model type in which the undesirable attributes originate (referred to as the *originating model type*).

You can remove base model types from the current model type if the current model type was creating using the developer ID that is currently active.

You cannot remove the last base model type from the current model type if the current model type has derived model types. In order to break such a connection, you must first navigate to the derived model types and use the following procedure to remove the model type of interest as a base model type with respect to the derived model types. You can then navigate back to the model type of interest and remove its last remaining base model type.

> **NOTE**
> Inference handlers are code segments that define the behavior and intelligence of a model type. Problems can occur if you remove a base model type from a derived model type, and the derived model type has associated inference handlers that refer to attributes that used to be inherited from the removed base model type. This sort of dependency can be difficult to detect. Also be aware that removing a base model type also may remove inference handlers that were inherited from that base model type. This may cause anomalies if the removed inference handlers performed some vital function for the model type or for other model types derived from it.

You can remove a base model type from a model type.

Follow these steps:

1. Set current the model type for which you want to remove a base model type.
2. Click the Hierarchy tab, and under Base Model Types, select the model type that you want to remove, and click the Remove selected base Model Type icon  .

   > **NOTE**
   > The tooltip for the button indicates whether this action only removes the selected model type as a base model type or also deletes the current model type. If the current model type has no derived model types, it is also deleted when you remove the last of its base model types because it is no longer a part of the model type hierarchy.

   The selected model type is removed as a base model type.

Import MIBs

DX NetOps Spectrum manages devices according to the requirements and values that are specified in their MIB documents. A *MIB (Management Information Base)* is a database that resides on a network device and represents that device as a hierarchical collection of objects. A MIB object represents an individual element of information, such as the uptime of a device. MIBs themselves are text files with a special syntax. A device MIB defines all of the objects that can be managed on the associated device. The MIB organizes this information in a tree structure with branches that organize the managed objects into logical groups.

You can use the Model Type Editor to import both SMIv1 and SMIv2 MIBs. However, when you import an SMIv2 MIB, the Model Type Editor maps the MIB data type to the corresponding data type that is defined in SMIv1. The Model Type Editor also supports most standard text conventions and associated enumerations that can be used in a MIB.

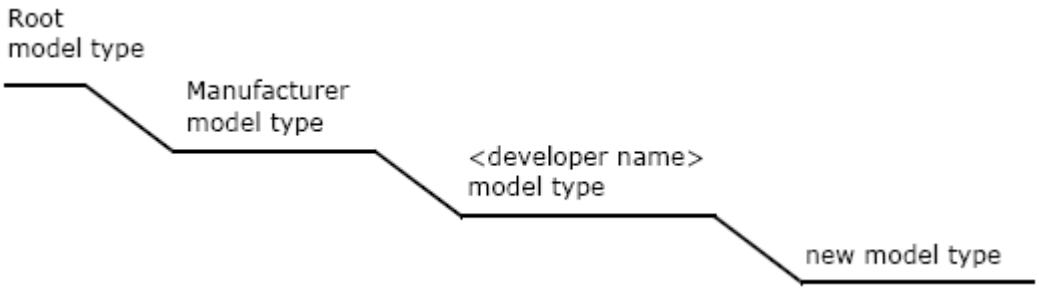
When you create a model type, you typically want to add base model types that provide access to specialized MIB attributes and intelligence. We recommend creating a set of MIB-specific model types, each one containing the intelligence and attributes of a specific MIB. Create a model type from GnSNMPMibDerPt and assign it a name that identifies the associated MIB. Then import the MIB into the model type. This approach lets you add the MIB-specific model type as a base model type to multiple model types and keep the associated attribute IDs consistent across all derived model types.

> **NOTE**
> You can only import a MIB into a model type that was created by the currently active developer ID. This rule prevents you from importing the MIB into a model type that you cannot subsequently export.

Using the Model Type Editor to import a MIB into a model type.

Follow these steps:

1. Select the model type from which you plan to derive a model type for the MIB import.
   Typically, the starting point is some developer-specific (vendor-specific) model type that is derived from the Manufacturer model type, as illustrated in the following figure:



   However, it can also be a model type that was derived from EntityTypes, MMDeveloper, or another model type in the modeling catalog. Another alternative is GnSNMPMibDerPt, which was designed specifically for importing MIBs into the DX NetOps Spectrum database. That model type already contains many needed attributes and relations. Regardless of your selection, you can create device model types at appropriate places in the model type hierarchy later for each device receiving the attributes of the MIB.

   NOTE
   For more information about designing a new model type, see Certifications .
2. Click the Hierarchy tab, and create a model type into which to import the MIB information:
   a. Click the Derive a new Model Type
   icon 
   b. Enter a model type name.
   The name cannot exceed 128 characters and can only consist of letters, numbers, underscore characters (_), or dash characters (-).

      WARNING
      A model type name is not required to be unique across the modeling catalog. However, supply a name that is unique across the model types that were created under the currently active developer ID. DX NetOps Spectrum differentiates model types using both the model type name and the developer ID component in the model type ID. To know more about the 'Developer ID' see, the Getting Started with the Model Type Editor section.
   c. Click OK.
   The derived model type is created and is set as the current model type.
3. Click File, Import MIB.
   The MIB Import dialog opens.

   NOTE
   Sometimes a single file contains multiple MIBs that are delineated with BEGIN and END statements. Include only a single MIB in the file that you import.
4. Click Browse, navigate to the MIB file to import, select the file, and click Open.
5. Click OK to begin the import.
   When the import is complete, the MIB Import Complete dialog is displayed to inform you of the number of attributes and attribute groups that were created.
   If issues are encountered during the import process, a warning is displayed to inform you that the import was successful but issues were encountered. If the import process fails, an error is displayed.
6. Click OK to close the dialog.

WARNING
When you create a model type, typically you also add support for traps, events, and alarms. You can use the MIB Tools utility in OneClick to add trap support and perform initial event configuration. For more information, see Certifications . You can then fully configure the events and alarms using the Event Configuration application in OneClick, as described inManaging Network. For more information about creating a management module (including creating model types), see Certifications .

Set Model Type Flags

To set and change the flags for a model type, you must be using the developer ID that was active when the model type was created.

Follow these steps:

1. Select the model type whose model type flags you want to set.
2. Click the Flags tab and click the Edit
   icon 
   The Edit Flags dialog opens.
3. Select (enable) and clear (disable) the flags as desired.
4. Click OK.

Working with Attributes

You can add or remove attributes from a model type directly and indirectly:

- To add or remove attributes *indirectly*, add or remove base model types, as described in Working with Base Model Types.
- To add or remove attributes *directly*, create or delete the attributes from the model type itself, as described in the topics in this section.

Add an Attribute to a Model Type

Typically, you will add attributes to model types that you own, that is, types that were created using the currently active developer ID. Because you own the model types, you can subsequently export them and their attributes.

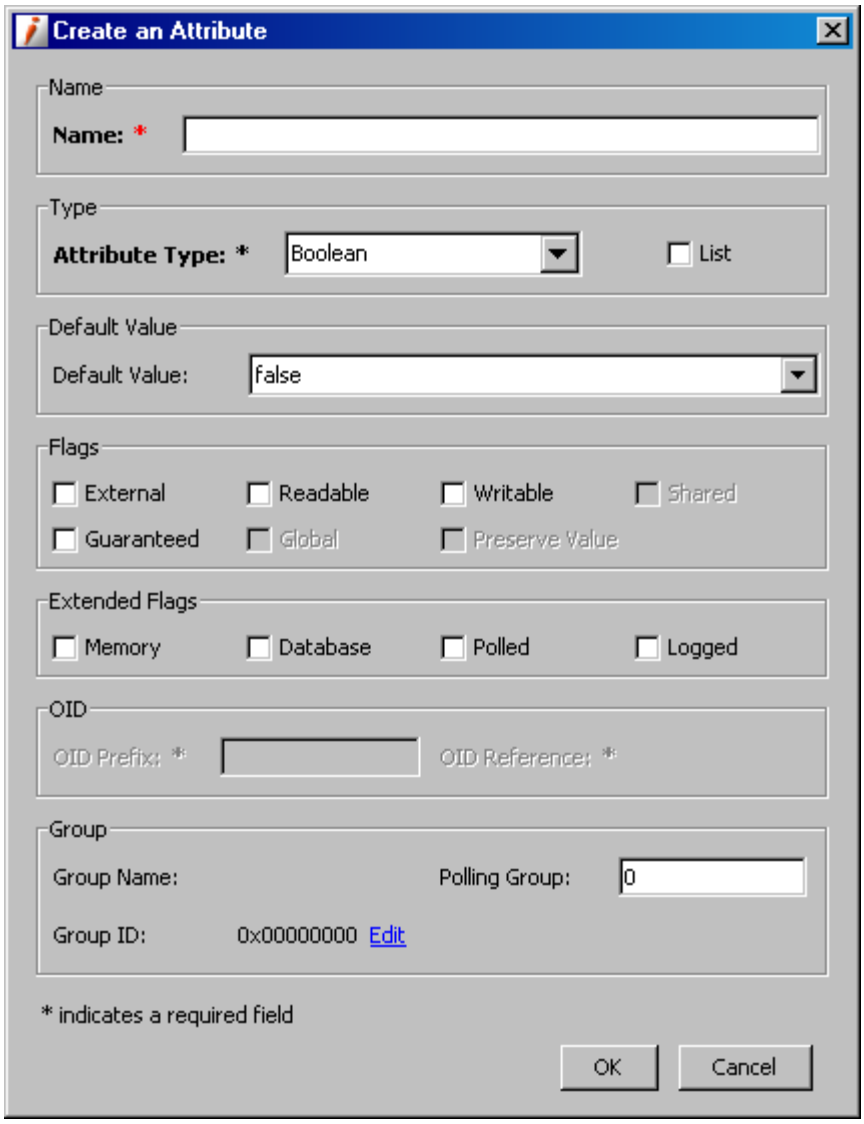
While you can also add attributes to model types that you do not own, you cannot export these model types or their attributes. Typically, exporting model types that you do not own is only required for storing information in the related models when no available attribute is suitable.

WARNING
We recommend recording all changes that you make with the Model Type Editor. Some changes are not migrated when the database is updated to a later version of DX NetOps Spectrum Specifically, if you modify attributes (such as flag settings), the model type hierarchy, or relations and the associated meta-rules in the model types supplied by CA or another vendor, the changes are typically not migrated on database upgrade. You will have to reapply these changes manually. For more information, see *Migrate Changes to a New Version of* DX NetOps Spectrum.

Follow these steps:

1. Select the model type to which you want to add an attribute.

2. Click the Attributes tab and click the Add a new Attribute icon  .
   The Create an Attribute dialog opens.



3. Enter values for the attribute's descriptors.

   NOTE
   An attribute name does not need to be unique across the modeling catalog, but you should enter a name that is unique across the attributes in the model type that were created using the same developer ID. In other words, within a single model type, two attributes can have the same name if they were created using different developer IDs. In addition, while it is allowed, reusing an attribute name is not recommended.

4. Click OK.
   The attribute is created and added to the working catalog, and it is displayed in the list of attributes on the Attributes tab.

Remove an Attribute from a Model Type

You can remove an attribute from the model type in which it was created (referred to as the *originating model type*) if you own the model type, that is, the model type was creating using the developer ID that is currently active.

When you remove an attribute from a model type, it is also removed from all derived model types that inherit it.

> **WARNING**
> We recommend recording all changes that you make with the Model Type Editor. Some changes are not migrated when the database is updated to a later version of DX NetOps Spectrum Specifically, if you modify attributes (such as flag settings), the model type hierarchy, or relations and the associated meta-rules in the model types supplied by CA or another vendor, the changes are typically not migrated on database upgrade. You will have to reapply these changes manually. For more information, see *Migrate Changes to a New Version of* DX NetOps Spectrum.

To remove an attribute from a model type

1. Verify the attribute to be removed is not critical to a derived model type.

   > **NOTE**
   > If necessary, you can provide an alternate path via a different base model type, or you can recreate the attribute at the derived model type level.
2. Select the model type for which you want to remove an attribute.
3. Click the Attributes tab, select the attribute you want to remove, and click the Delete selected Attribute icon 
   A confirmation dialog opens.
4. Click Yes.
   The attribute is deleted from the working catalog.

Edit an Attribute

Typically, you will want to modify the attributes of the model types that you own, that is, that were created using the developer ID that is currently active. Because you own the model types, you can subsequently export them and their attributes.

While you can also modify the attributes of model types that you do not own, you cannot export these model types or the changes to their attributes.

You can edit an attribute by doing the following:

- Changing its default value. See Modifying an Attribute's Default Value.
- Changing its other descriptor values. See Modifying an Attribute's Descriptors.

However, before editing an attribute's characteristics, you should verify that derived model types that inherit the attribute will not be adversely affected by the changes.

> **WARNING**
> We recommend recording all changes that you make with the Model Type Editor. Some changes are not migrated when the database is updated to a later version of DX NetOps Spectrum Specifically, if you modify attributes (such as flag settings), the model type hierarchy, or relations and the associated meta-rules in the model types supplied by CA or another vendor, the changes are typically not migrated on database upgrade. You will have to reapply these changes manually. For more information, see *Migrate Changes to a New Version of* DX NetOps Spectrum.

Modify an Attribute's Default Value

An attribute can inherit its default value from a base model type or specify its own value (a process called specialization). In the latter case, all model types derived from the specialized model type inherit the changed attribute value.

To modify the default value of an attribute

1. Set current the model type that has the attribute with the default value you want to change.
2. Click the Hierarchy tab, note any derived model types, and verify that they will not be adversely affected by the change you want to make.
3. Click the Attributes tab, select the attribute from the displayed list, and click the Edit icon  .
   The Edit an Attribute dialog opens.
4. Do *one* of the following to change the default value:
   - If the attribute requires a single value, modify the value as desired.
   - If the attribute requires a list of values, click Edit, and modify the values as desired in the Edit List Values dialog.
   - To add a value, click the Add icon  .
   - Enter an appropriate Object ID index value (typically, an SNMP object identifier) and enter the value to associate with that index entry.

     The Object ID value is optional, and, if you specify the first one, the Model Type Editor uses that Object ID to generate a default, modifiable Object ID for each subsequent value that you add.

     To delete a value, click the Delete icon  .

     NOTE
     If the Default Value field does not have scroll bars, you can enter a value that fits in the provided area. If there are scroll bars, the field automatically enlarges as needed, and the only limitation is the impact on system performance.
5. Click OK twice.

Modify an Attribute's Descriptors

You can modify two types of attribute descriptors:

- **Standard attribute descriptors**
  When you modify one of these descriptors in a base model type, all derived (child) model types inherit the change.
- **Attribute descriptors you can specialize**
  When you modify one of these descriptors in a base model type, derived model types that have been specialized (specify their own values) do *not* inherit the change, but derived model types that have not been specialized do inherit the change.

If you are the owner of the attribute (that is, the attribute was created using the developer ID that is currently active), you can modify all of an attribute's descriptors *in the originating model type* regardless of whether you are the owner of the model type.

If you are not the owner of the attribute, or if the attribute is inherited, you can specialize a subset of the attribute descriptors. That is, you can modify the values of some of the descriptors at any level of the inheritance hierarchy; you are not limited to modifying the values in the originating model type.

To edit the descriptors of an attribute

1. Set current the model type that has the attribute with the descriptor values you want to change.
2. Click the Hierarchy tab, note any derived model types, and verify that the model types will not be adversely affected by the changes you want to make.
3. Click the Attributes tab, select the attribute from the displayed list, and click the Edit selected Attribute icon 
   The Edit an Attribute dialog opens.
4. Modify the attribute descriptors as needed.

   NOTE
   All changes that you make to flag settings are subject to the relationships described in Flags.
5. Click OK.

Working with Attribute Groups

Working with Attribute Groups

An *attribute group* is a logical collection of related attributes in a model type. Groups make working with related attributes easier in the Model Type Editor because they allow you to define and use a user-defined sorting mechanism. You can create groups, assign attributes to them, and then add the Group Name or Group ID as a column header in the table of attributes on the Attributes tab in the Contents panel. This lets you then click the column header to quickly group together and view together all of the attributes within a group.

Creating an Attribute Group

You can create attribute groups.

NOTE
When you add an attribute group, you add it to a specific model type, and the group is inherited by derived model types in the same way that other attributes of a model type are inherited. Like for other attributes, if you add an attribute group to a model type that you do not own (its developer ID does not match the one that is currently active), you will not be able to export the group.

To Create an Attribute Group

1. Set current the model type in which you want to create the attribute group.
2. Click the Attributes tab, and double-click any attribute that was created using the developer ID that is current active.
   The Edit an Attribute dialog opens.
3. Under Group, click Edit beside the value for Group ID.
   The Select Group dialog opens displaying a table view and a tree view of all of the attribute groups inherited from base model types or originating in the current model type.
   The Model Type column displays the originating model type for each group (that is, the model type in which the group was created).
4. Specify whether the new attribute group has a parent group:
   - If you do not want the attribute group to have a parent group, do nothing. This is the default behavior.
   - If you want the attribute group to have a parent group, select the group using either of the following methods:
     - Click the Table View tab, and use the table to select the desired parent group.

To help you find the group, for Filter, you can enter a text string to filter the list to include only the groups whose names include the string. You can also click any column heading to change the sort order from ascending to descending and vice versa.

- Click the Tree View tab, and navigate the hierarchical tree of parent groups and child groups to select the desired parent group.

5. Click the Add icon  .
   The Create Group dialog opens.
6. Enter a name for Group Name

   NOTE
   Do not enter a name that is being used by another attribute group created using the same developer ID. In addition, use a maximum of 128 characters, and use only numbers, letters, and underscore characters (_).
7. Click OK.
   The new attribute group is created.

Modifying an Attribute Group

You can modify the name or parent group of an attribute group in the originating model type, that is, in the model type in which the group was created. You cannot modify the name or parent group of an inherited attribute group.

In addition, to modify an attribute group, you must be the owner of the model type in which the group was created. In other words, the active developer ID is the one that was used to create the model type.

To modify an attribute group

1. Set current the model type in which the attribute group was created.
2. Click the Attributes tab, and double-click any attribute that was created using the developer ID that is currently active.
   The Edit an Attribute dialog opens.
3. Under Group, click Edit beside the value for Group ID.
   The Select Group dialog opens. The dialog displays a table view and a tree view of all of the attribute groups inherited from base model types or originating in the current model type. The Model Type column displays the originating model type for each group (that is, the model type in which the group was created).
4. Select the attribute group to modify using either of the following methods:
   – Click the Table View tab, and use the table to select the desired group.
   To help you find the group, for Filter, you can enter a text string to filter the list to include only the groups whose names include the string. You can also click any column heading to change the sort order from ascending to descending and vice versa.
   – Click the Tree View tab, and navigate the hierarchical tree of parent groups and child groups to select the desired group.
5. Click the Edit icon  .
   The Edit Group dialog opens.
6. If you want to change the group name, enter a new name for Group Name.

   NOTE
   Do not enter a name that is being used by another attribute group created under the same developer ID. In addition, use a maximum of 128 characters, and use only numbers, letters, and underscore characters (_).
7. If you want to change the group's parent group, do the following:
   a. Click Edit beside the value for Parent Group ID.

b. In the Select Parent Group dialog, select a new parent attribute group using the Table View tab or the Tree View tab, and click OK.

8. Click OK.
   Your changes are saved and the Edit Group dialog closes.

Deleting an Attribute Group

You can delete an attribute group if the following two conditions are met:

- The group has no assigned attributes. If this is not the case, you can delete the group after you have reassigned the attributes.
- The group has no subgroups. If this is not the case, you can delete the group after you have reassigned or deleted the subgroups.

To delete an attribute group

1. Set current the model type in which the attribute group was created.
2. Click the Attributes tab, and double-click any attribute that was created using the developer ID that is currently active.
   The Edit an Attribute dialog opens.
3. Under Group, click Edit beside the value for Group ID.
   The Select Group dialog opens. The dialog displays a table view and a tree view of all of the attribute groups inherited from base model types or originating in the current model type. The Model Type column displays the originating model type for each group (that is, the model type in which the group was created).
4. Select the attribute group to delete and click the Delete icon 
   A confirmation dialog opens.
5. Click Yes.
   The attribute group is deleted.

Working with Relations and Meta-Rules

About Working with Relations and Meta-Rules

The modeling catalog provided with the basic DX NetOps Spectrum package contains a number of predefined relations, many of which have associated meta-rules. These relations provide a framework that can replicate most relationships in a network. However, you can create additional relations as needed for your network design.

If you add a new relation, you must also do the following:

- Create meta-rules in <met> that implement the new relation for specific model types.
- Add intelligence to the model types so that models of those types react appropriately when they are associated based on the new meta-rules. You must implement the intelligence programmatically using the CORBA API.
  As an example, assume that you create the following meta-rule:

  ```
  User Sends_mail_to User
  ```

  When the meta-rule is instantiated (for example, when one User model sends mail to a second User model), the first model may need to react to the fact that it (the user that it represents) has sent mail, and the second model may need to react to the fact that it has received mail. In this case, you must add intelligence to the User model type to implement these reactions.

  NOTE
  For information about using the CORBA API, see the Development API Reference section

Search for and Display Relations

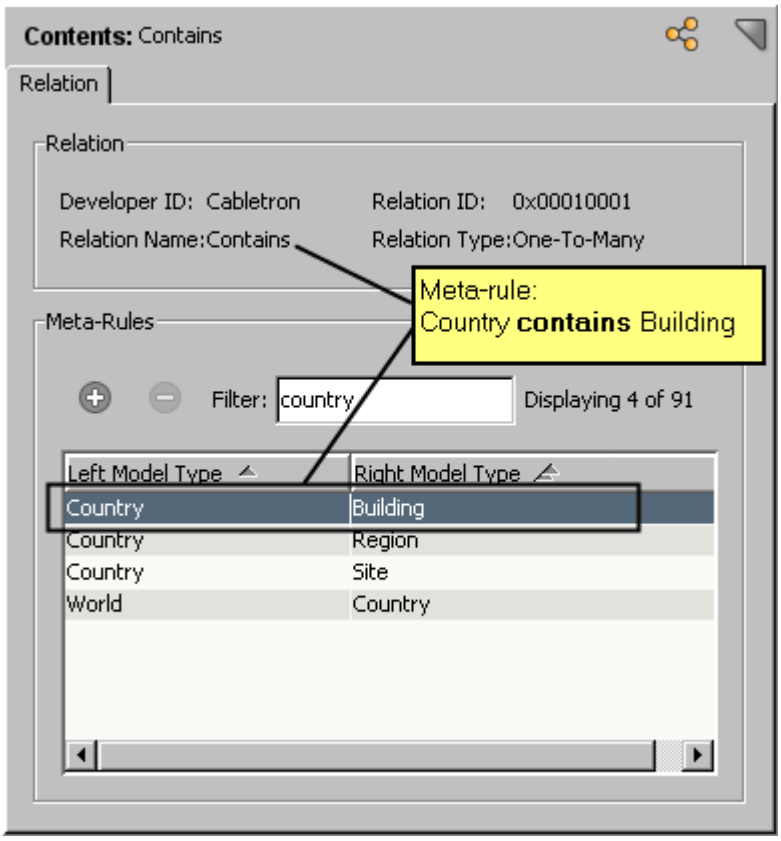
You can search for relation by filtering the list of all relations in the modeling catalog to include only those that contain the text string you specify. By default, the Model Type Editor examines the supplied string against the names of the relations. However, if you display the Relation ID column in the Navigation panel, it will also examine the string against the relation IDs (handles).

Follow these steps:

1. In the Navigation panel, click the Relations tab.
   The names of all of the relations in the modeling catalog are listed.
2. In the list, select the name of the relation that you want to examine.
   To locate and select a specific relation, you can take the following steps:
   - Enter a text string in the Filter text box in order to filter the list to include only the relations whose names or IDs contain the string. To filter the list by ID, the ID column must be displayed in the table.
   - Click the Relation bar at the top of the list to change the alphabetical sorting from ascending to descending or vice versa.

   The selected relation becomes the current relation. The following information about the relationship is displayed in the Contents panel:
   - **Developer ID**
     Specifies the developer ID that was active when the relation was created.
   - **Relation Name**
     Specifies the name of the relation.
   - **Relation ID**
     Specifies the ID (handle) that is assigned to the relation.

     NOTE
     A handle is never reused even if you delete a relation.
   - **Relation Type**
     Specifies the type of relation, either One-to-Many or Many-to-Many.
   - **Meta-Rules**
     Specifies the list of meta-rules that apply the relation to specific model types, thereby defining how the model types can interact with one another.
3. To filter the list of meta-rules to include only those for a specific model type, enter the full or partial name of the model type in the Filter text box:



Create Relations

You can create relations.

To create a relation

1. Click the Relations tab and in the Navigation panel, click the Create Relation icon 
   The Create Relation dialog opens.
2. Enter a name for the relation.
   The name should be a maximum length of 31 characters and should include alphanumeric characters and underscores but not spaces or punctuation.

   WARNING
   The relation name does not need to be unique across the modeling catalog, but you should enter a name that is unique across the relations created under a given developer ID. In addition, while it is allowed, reusing a relation name is not recommended.
3. Select the type of relation to create:
   - **One-to-Many**
     Relations of this type relate one model type to many model types.
   - **Many-to-Many**
     Relations of this type relate many model types to many model types.

     NOTE
     Once you create the relation, you cannot change the relation type. To specify a different relation type, you must delete the relation and create a new one.

4. Click OK.
   The relation is created and assigned a relation ID, and its information is displayed in the Contents panel.

   NOTE
   A handle is never reused even if you delete the relation.
5. Create one or more meta-rules that use the relation.
6. Add intelligence to the relevant model types so that models of those types react appropriately when they are associated based on the new meta-rules. You must implement the intelligence programmatically using the CORBA API.

   NOTE
   For information on using the CORBA API, see the Development API Reference section.

Relation Meta - Rules

Each relation normally has one or more meta-rules, each of which applies the relation to specific model types. Many of the model types in the core database have relations that are supplied without meta-rules; these function as "placeholders" for which you can supply customized meta-rules that are appropriate for your network design.

NOTE
When a new model type is derived from one that is specified as a member of a meta-rule in a relation, the meta-rule automatically applies to the derived model type. For example, the modeling catalog provided with the basic DX NetOps Spectrum package has a relation called Contains and two model types named Room and Device. One of the meta-rules for the Contains relation is the following:

Room Contains Device

The modeling catalog also contains a Workstation model type that is derived from Device. As a result, the "Room Contains Device" also applies to the Workstation model type in the form of "Room Contains Workstation." You do not have to explicitly create a meta-rule that includes the Workstation model type.

About Creating Meta-Rules for General Model Types

Whenever possible, create meta-rules for general model types (model types near the top of the model type hierarchy) in order to maximize their application and reduce the need for more specific meta-rules. The more general the model type that contains the meta-rules, the more derived model types it has, and, therefore, the more model types inherit its meta-rules.

As an example, the Device model type is a general model type that is used in the following meta-rule for the Is_Adjacent_to relation:

Device Is_Adjacent_to Device

Consequently, any model type derived from Device or one of its descendants can be adjacent to any other model type derived from Device or one of its descendants.

Create Meta - Rules

You can create a meta-rule without restriction if you own the corresponding relation, that is, the relation was created using the developer ID that is currently active.

If the relation was created using a different developer ID, you can still create a meta-rule for it if at least one of the model types used in the rule was created using the developer ID that is currently active.

To create a meta-rule

1. Search for and display the relation for which to create the meta-rule.
   The relation is displayed in the Contents panel.

2. On the Relation tab, in the Contents panel, click the Create a new Meta-Rule icon  .
   The Create Meta Rule dialog opens.
3. In the list of model types on the left, select the antecedent (left) model type to include in the meta-rule.

   NOTE
   To help you locate and select a specific model type, you can enter a text string in the corresponding Filter text box in order to filter the list accordingly.
4. In the list of model types on the right, select the predicate (right) model type to include in the meta-rule, and click OK.
   The new meta-rule is added to the list of meta-rules for the current relation; it is inserted into the list alphabetically according to the antecedent model type.

   NOTE
   You now need to add intelligence to the model types so that models of those types react appropriately when they are associated based on the new meta-rule. You must implement the intelligence programmatically using the CORBA API.

Delete Relations

You can delete a relation if the following two conditions are met:

- The relation was created using the developer ID that is currently active.
- The relation does not have any associated meta-rules. If this is not the case, first delete the meta-rules.

To delete a relation

1. Search for and display the relation to delete.
2. On the Relations tab in the Navigation panel, select the relation, and click the Delete selected relation icon  .
   A confirmation dialog opens.
3. Click Yes.
   The relation is deleted.

Delete Meta - Rules

You can delete a meta-rule without restriction if you own the corresponding relation, that is, the relation was created using the developer ID that is currently active.

If the relation was created using a different developer ID, you can still delete the meta-rule if at least one of the model types used in the rule was created using the developer ID that is currently active.

To delete a meta-rule

1. Search for and display the relation that contains the meta-rule to delete.
2. On the Relation tab in the Contents panel, select the meta-rule and click the Delete selected meta-rule icon  .
   A confirmation dialog opens.
3. Click Yes.
   The meta-rule is deleted.

Importing and Exporting Model Types

About Importing and Exporting Model Types

As your network grows, you may need to add new management modules and model types to your database to support additional types of devices. Moreover, you may want to add these new model types to your database without installing a completely new database, so you can keep intact all of the model types you have already created or modified. DX NetOps Spectrum provides two database utilities that lets you to import and export model types:

- **Import and export commands in the Model Type Editor:** These commands let you import and export model types from the *working catalog* for the current session. Because the commands do not operate on the permanent catalog in the SpectroSERVER database, you can make an explicit decision after an import as to whether to commit or discard the changes. Since you can only import or export one catalog file at a time, use these commands when you have only one or a few files to process.
- **A command-line utility named dbtool:** This command-line utility program lets you import and export model types from the *permanent catalog* in the SpectroSERVER database. Because you can specify multiple files as command-line arguments, use this utility to batch process a set of files.

In both cases, the transfer vehicle is a binary export file that has a .e extension. These files are referred to as *catalogs*.

Import Model Types Using the Model Type Editor

You can use the Model Type Editor to import model types, attributes, relations, and meta-rules into the SpectroSERVER database. The modeling catalog objects must be defined in a catalog file (.e file) that was created using the export feature in the Model Type Editor or the dbtool command-line utility.

Import Constraints

> **WARNING**
> You can only import modeling catalog objects that are stored in a compatible DX NetOps Spectrum database (a database that is running the same version of DX NetOps Spectrum as the destination database). For information about updating a DX NetOps Spectrum database, see the *Installation section.*

New model types are imported into the modeling catalog according to the following constraints:

- If the catalog file to import contains a model type that does not exist in the destination database, the model type is imported according to the rest of the constraints described in this section.
- All of the base model types for a new model type must already exist in the destination database. If they do not, you must import them before importing the new model type. Typically, the "core catalog" contains these prerequisite model types, and documentation accompanying any new catalog informs you of any such dependencies.
  If a base model type for a new model type does not exist, the import process is terminated. In this situation, you can identify the missing model type in the DX NetOps Spectrum Control Panel. You must then import a catalog file that contains the missing base model type, and then reinitiate the import process that was terminated.
- If the catalog file to import contains a model type that already exists in the destination database, the existing version is modified to match the version to import, for example:
  - The model type's derivation is updated
  - New attributes are added
  - Existing attributes are updated
  - Attributes in the existing model type that have the same developer ID as the existing model type are deleted if they originate in the existing model type but are not included in the version being imported
- If an error occurs due to an inability to write to the destination database or due to insufficient memory or system resources, the import process is terminated, and you are notified with an appropriate error message. Because this situation can leave the database in an incomplete or corrupted state, you should always back up the database before beginning an import operation.

Import Model Types

When you import model types using the Model Type Editor, the model types and associated catalog objects are imported into the working catalog. This lets you perform the import and then make an explicit decision as to whether to discard the changes or permanently commit them to the SpectroSERVER database.

To import the contents of a catalog

1. Back up the SpectroSERVER database into which you are importing a catalog.

   NOTE
   For information about how to back up the database using the database utilities provided with DX NetOps Spectrum, see the Database Management *section.*
2. Select File, Import Model Types.
   The Open dialog opens.
3. Navigate to the catalog file (.e file) you want to import, select the file, and click Open.
   The modeling catalog information in the selected file is imported into the working catalog for the current session. At this point, you can manually commit the changes to the permanent catalog in the SpectroSERVER database , or you can do so when you are prompted when you exit the application. Alternatively, you can exit the application without committing the changes to discard them.

Export Model Types Using the Model Type Editor

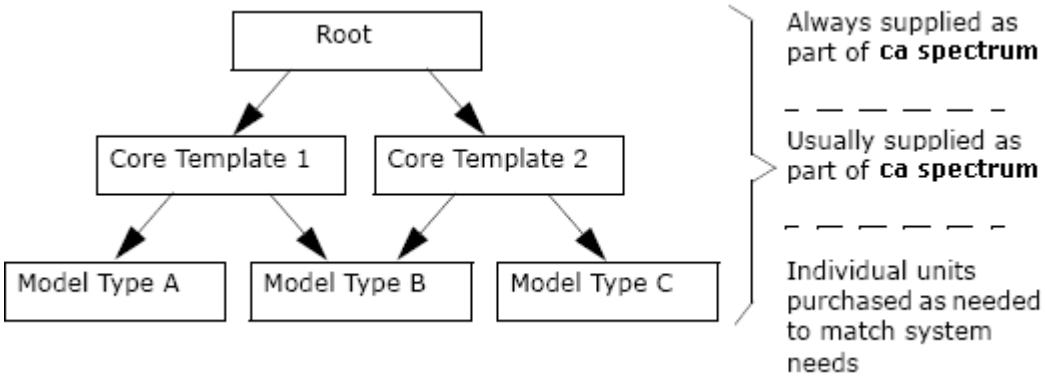
You can use the Model Type Editor to export a specific list of model types from the SpectroSERVER database to a catalog file. A catalog file is a .e file, and it is sometimes referred to as a DX NetOps Spectrum database export file.

When you are exporting model types and associated catalog objects, bear the following in mind:

- The export feature in the Model Type Editor exports the working catalog that is stored in memory. As a result, the resulting export file includes any changes you have made to model types and associated objects during the current session even if you have not yet committed those changes to the SpectroSERVER database.
  To export the permanent catalog only, you can commit the changes and use the export feature in the Model Type Editor, or you can use the dbtool command-line utility instead.
- The .e file produced by the export process contains the following information:
  - The attribute descriptors that originated in the model type being exported.
  - The attribute descriptors that have been specialized (for example, by specifying a default value to override an inherited one).
  - The relations and associated meta-rules in which the model type and/or any ancestor model types participate as an antecedent or a predicate.
- You can only export the model types, attributes, and relations (including their associated meta-rules) that were created using the active developer ID.

To export one or more model types to a catalog file

1. Identify the model types to export.
   Typically, the list of model types to export should include any base model types that are required by the model types being exported and that do not exist in the destination database.
   However, in general practice, dependencies normally are limited to certain commonly-used base model types that are contained in one or more "core" catalogs that are included as part of the basic DX NetOps Spectrum system. These core catalogs are a part of every installation, as shown in the following illustration.



2. Select File, Export Model Types.
   The Model Type Export dialog opens.
   By default, all unmodified model types that were created using the active developer ID are initially displayed in the Model Types column on the left.
   In addition, all model types that were created using the active developer ID and that have been modified during the current session are initially displayed in the Model Types to Export column on the right. *Modified model types* are those you have changed regardless of whether the changes are currently in the working catalog or committed to the database. Note that once you export a model type, it is no longer identified as modified because you have saved it to a catalog, in this case, to a DX NetOps Spectrum database export file.
3. Move the model types that you want to export to the Model Types to Export list.
   To search for the model types to export, in the Filter text box, you can enter a text string to filter the list to include only the model types with names that contain or match the string. The filter is not case-sensitive.
   To move all of the model types available for export to the Model Types to Export list,
   click ▷▷.
   To move a single model type to the list, double-click the model type name, or select the model type and
   click ▷.
   Similarly, you can remove model types from the export list by double-clicking them or using the corresponding left-arrow buttons.
4. Click Browse, and in the Save dialog, navigate to the folder in which to save the catalog file.

   WARNING
   By default, the Save dialog opens to the DX NetOps Spectrum database (modeling catalog) directory. However, it is recommended that you save exported catalog files in a directory outside of the DX NetOps Spectrum installation area in order to prevent the loss of the files during a DX NetOps Spectrum update process.
5. Enter a name for the file (you do not need to include the .e extension) or select an existing .e file to overwrite, and click Save.
6. In the Model Type Export dialog, click OK.
   The model types are exported to the specified catalog file.

Import and Export Model Types Using dbtool

The DX NetOps Spectrum command-line utility program named dbtool can import or export model types from the permanent modeling catalog in the SpectroSERVER database. The dbtool utility lets you specify multiple files as command-line arguments. Therefore, it is a better tool to use than the import and export features of the Model Type Editor for batch processing a set of files.

When you use dbtool, you can export only the model types that you "own," that is, types that were created using the currently active developer ID. If a model type is not owned, an error message describes the relevant model types, and the export process is terminated.

Run the dbtool utility from the directory that contains the SpectroSERVER database that is used in the import or export. In addition, while the import or export is underway, *no* other program or process (for example, a VNM or the Model Type Editor) can access the database. Keep in mind that while CA-developed applications automatically lock out other CA-developed applications, third-party applications may not. The competition can result in database corruption.

NOTE
For more information about running dbtool, see the Database Management *section.* This section also contains information about how to back up the database using the DX NetOps Spectrum database utilities. Perform the backup before you perform an import.

Send an Exported Catalog to a File or Printer

DX NetOps Spectrum includes a command-line utility program named *dbtool* that you can use to send or "dump" the contents of a catalog file (.e file) to a file or printer. You can use this tool to store or print the contents of a catalog file that was created using the export feature of the Model Type Editor or using dbtool itself.

Before running dbtool, you must shut down the SpectroSERVER and any other program that accesses the SpectroSERVER database.

NOTE
For information about running the dbtool utility, see the Database Management *section.*

Running Reports on Model Types and Relations

About Running Reports on Model Types and Relations

DX NetOps Spectrum includes a command-line utility program named reports that you can use to display, print, or export (to a file) information about the model types and relations in the modeling catalog.

You must run the reports utility from the directory that contains the SpectroSERVER database that contains the data you want to access. While the report is being generated, no other program or process (for example, a VNM or the Model Type Editor) can access the database.

NOTE
For information about running the reports utility, see Database Management *section.*

Hide a Model Type Name

OneClick provides a flexible platform for administrators to modify aspects of the application to meet specific requirements.

To Hide a Model Type Name on Containers such as LAN or Network in Topology view:

On the OneClick web server machine, follow these steps:

1. Copy the **$SPECROOT/tomcat/webapps/spectrum/WEB-INF/topo/config/oneclick-container-iconbase-config.xml file** to the **$SPECROOT/custom/topo/config/** directory.

2. Edit the **$SPECROOT/custom/topo/config/oneclick-container-iconbase-config.xml** file and comment out the following line:

NOTE

<!-- <column idref="column-devicetype-config"/> -->

3. Save the file.

4. Launch the OneClick Console.

5. The Modeltype_Name is no longer visible in Topology view for the containers.

OneClick Customization

OneClick provides a flexible platform for administrators to modify aspects of the application to meet specific requirements. For example, you can modify OneClick behavior to support the unique structure of a site, an enterprise and network environment, work processes, and software deployments. Make your modifications using the OneClick UI or by coding the changes in the XML files that are provided for that purpose.

> **WARNING**
>
> ! Do not add customizations to the files in their default location (<*$SPECROOT*>/tomcat/webapps/spectrum/WEB-INF/console/config/). The customizations in that directory are ignored. In addition, these files are overwritten when you perform DX NetOps Spectrum and OneClick upgrades.

Prerequisites for Customizing OneClick XML Files

Before you attempt to customize OneClick files, be aware of the following requirements:

- You must be able to create and modify files on the OneClick server.
- You must be familiar with the fundamentals of XML coding as well as the DX NetOps Spectrum and OneClick directory structure.
- You must know the following:
  - The file whose functionality you want to extend with your modifications.
  - The directory in the <*$SPECROOT*>/custom directory structure in which to create your custom file.

Extend Factory XML Files

You can extend default XML files to accomplish OneClick customizations without overriding the entire factory default file. Customized XML files are not removed during a DX NetOps Spectrum/OneClick software upgrade or reinstallation.

To extend the default OneClick XML configuration files, create a file with the same name as the default file in the appropriate custom directory. Use the XML idref attribute in the new file to refer to the default OneClick file of the same name. Code the new functionality in this file. When OneClick parses the XML files, the changes in the new file are added to the existing factory file referenced using idref.

By extending factory files, you are able to take advantage of new features and functionality available in software updates to the factory XML code while preserving your customizations.

Although you can still override a factory XML file by creating a copy of it in the <*$SPECROOT*>/custom directory and making your changes in the copy, using the IDREF XML attribute provides the ability to inherit and extend the factory file, while maintaining customizations in streamlined files.

Override Factory Files

Override a factory configuration file by copying the original file to the appropriate custom directory, and then adding new XML code or modifying the existing XML code. OneClick reads the files in the custom directory first. If the file exists in the custom directory and does not contain an idref statement referencing the original factory file, OneClick does not read the original factory file, and the new file overrides the original factory file.

WARNING

! Do not add customizations to the files in their default location (<*$SPECROOT*>/tomcat/webapps/spectrum/WEB-INF/console/config/). The customizations in that directory are ignored. In addition, these files are overwritten when you perform DX NetOps Spectrum and OneClick upgrades.

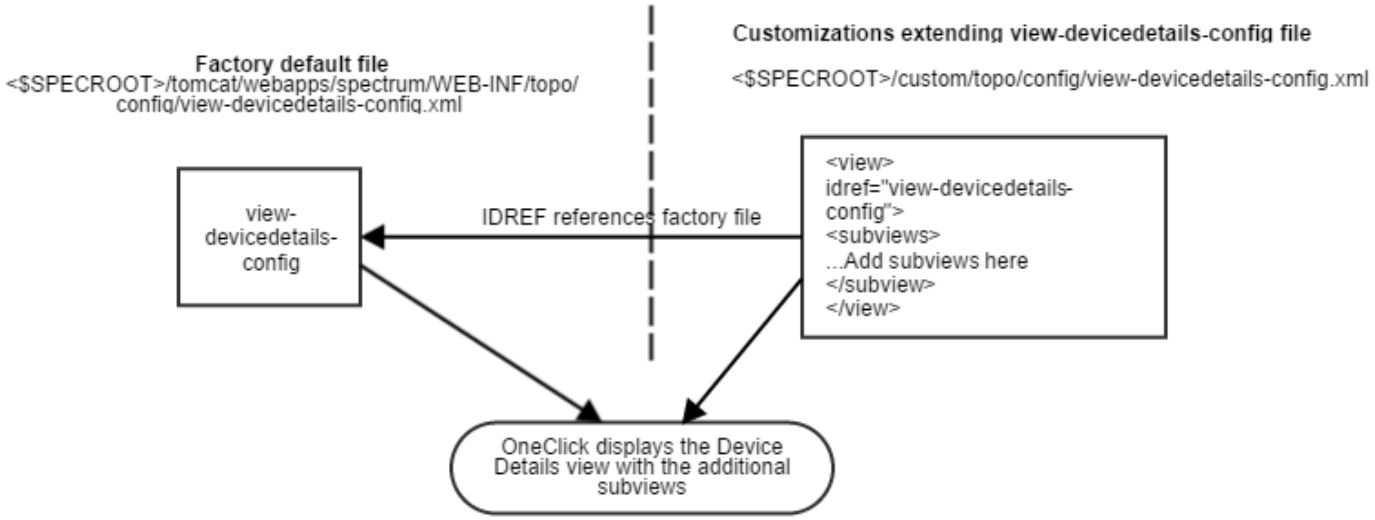
Inherit Features in Factory XML Files

Using idref to extend XML files has applications beyond extending the factory file with the same name. You can use this technique to inherit or reuse features in any file of the same type. For example, you can create your own model types that have a customized details view defined in view-mymtypedetails-config.xml. This model type can also inherit the default device views configured in view-devicedetails-config using idref. The new custom file extends the functionality of the default file while also inheriting the views in the default file.

Example Extending Factory XML File

The example in the following figure extends the functionality of the factory default <*$SPECROOT*>/tomcat/webapps/spectrum/WEB-INF/topo/
config/view-devicedetails-config.xml file by adding the code for the new subviews in <*$SPECROOT*>/custom/topo/config/view-devicedetails-config.xml. The default factory file view-devicedetails-config is specified in an "idref" statement.

Figure 79: Extending Factory XML File



OneClick Directory Structure

This section explains the directory structure of the XML files used to create the OneClick interface. You must be familiar with the structure to find the files necessary for customization and to implement customization in directories that are not overwritten when you upgrade or reinstall DX NetOps Spectrum.

Existing OneClick Files

The OneClick user interface is installed with a default layout, panel, menu, toolbar, and submenu content. The files that reside on the OneClick server controls all of these features. These files and their locations are identified in this section.

The console/config Directory

The files in the *<$SPECROOT>*/tomcat/webapps/spectrum/WEB-INF/console/config directory support menus, topology views, privileges for user interface elements, branding elements, and other aspects of the OneClick user interface. The files that are located in this directory are placeholder files that resemble templates for customizations to OneClick functionality. The files and their functions are described as follows:

- **custom-app-config.xml**
  General OneClick registrations, and topology support for DX NetOps Spectrum model types, including icons and views.
- **custom-branding-config.xml**
  Customizes the following UI branding elements of OneClick:
  - Application brand name
  - Application suite name
  - Image to display in the splash screen
  - Image to display as the logo button in the lower-left corner
  - Name of the root node in the tree in the Navigation panel
  - About dialog

    NOTE
    For information about the XML elements to specify these branding elements, see the comments in the file that is named custom-branding-config.xml in *<$SPECROOT>*/tomcat/webapps/spectrum/WEB-INF/console/config/.
- **custom-menu-config.xml**
  OneClick menus and toolbars.
- **custom-privileges.xml**
  Registers custom privileges that are applied to the menu items, columns, and subviews.

To customize the OneClick user interface, copy these files to the *<$SPECROOT>*/custom/console/config directory and then edit them.

WARNING

Do not add customizations to the files in their default location (*<$SPECROOT>*/tomcat/webapps/spectrum/WEB-INF/console/config/). The customizations in that directory are ignored. In addition, these files are overwritten when you perform DX NetOps Spectrum and OneClick upgrades.

Check the <$SPECROOT>/custom/console/config directory before copying files there. Some actions, such as creating custom searches in the Explorer, automatically create a copy of the custom-app-config.xml if one does not exist. If the config files already exist in the <$SPECROOT>/custom/console/config directory, add your customizations to those existing files.

The topo/config Directory

The files in this directory create the components of the OneClick topology views. These components include icons, subviews, and tables that display data.

All of the table files are named after the functionality that they display. For example, the file that builds the interface table for each model type is table-common-ifconfig-config.xml.

The common/config Directory

The files in this directory create various topology elements that can be used by all of the other files that create the OneClick interface. This includes colors, columns for tables, and tables.

The alarm/config Directory

The files in this directory create the OneClick alarm views and contents, including the Alarms table and the Alarm Details information tab.

Save Customized XML Files

OneClick customization files must be placed in specific “custom” directories so that OneClick finds and reads the customized code and associates it with the correct default factory file. The following lists the custom directories for the OneClick component categories.

- **Alarms**
  <*$SPECROOT*>/custom/alarm/config/
- **Common**
  <*$SPECROOT*>/custom/common/config/

  WARNING
  Do *not* copy the <*$SPECROOT*>/custom/common/config/custom-jnlp-config.xml file to another computer when you migrate and upgrade DX NetOps Spectrum. This file can contain memory settings that are not compatible with the computer where you are copying the custom directories.
- **Console components**
  <*$SPECROOT*>/custom/console/config/
- **Event format and probable cause files**
  <*$SPECROOT*>/custom/Events/
- **Images**
  <*$SPECROOT*>/custom/images/
- **Background images**
  <*$SPECROOT*>/custom/images/Background/
- **Stored SSL certificates**
  <*$SPECROOT*>/custom/keystore/
- **Report Manager**
  <*$SPECROOT*>/custom/repmgr/config/
- **Topologies**
  <*$SPECROOT*>/custom/topo/config/

Preserve XML Customizations

OneClick does not delete or overwrite files in the custom directory during an upgrade of DX NetOps Spectrum or OneClick.

Customized OneClick XML files may be overwritten in the following situation:

- Uninstalling SpectroSERVER
- Reinstalling the same version of SpectroSERVER if you have installed OneClick under the DX NetOps Spectrum installation directory.
  In this case, you should save off the customized files to an area unaffected by the uninstall process, and re-insert them once you have reinstalled SpectroSERVER.

  NOTE
  For more information on upgrades and installation of DX NetOps Spectrum and OneClick, see the Fresh Install section.

Preserve Custom Images

You must place all image files that you create or customize in the *<$SPECROOT>*/custom/images directory. Otherwise, all new or customized images are deleted or overwritten during an upgrade or reinstallation of DX NetOps Spectrum or OneClick.

Customizing the OneClick Login Dialog

Custom Login Message

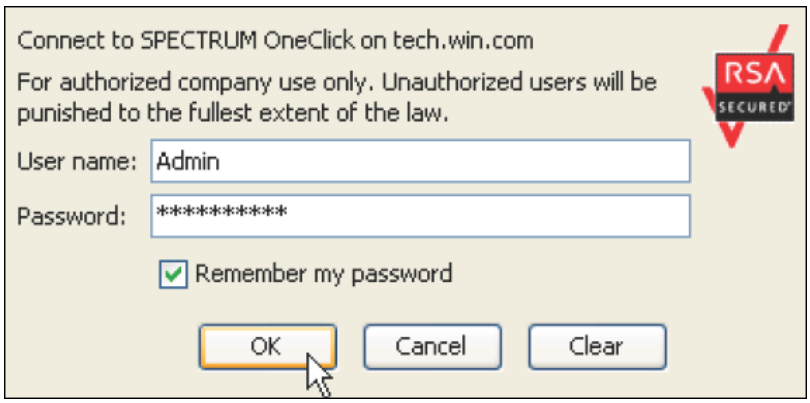
Custom messages can be added to the Login dialog for the OneClick Console. You can use this message to inform OneClick users about your usage policies, legal rights, consequences of unauthorized usage, or other important information they must know before they log in. The custom message appears in the Login dialog for the OneClick Console only.

Add a Custom Message to the OneClick Login Dialog

To inform OneClick users about usage policies, legal rights, or other important information needed before logging in, you can add a custom message to the OneClick Login dialog.

Follow these steps

1. Open the *<$SPECROOT>*/tomcat/webapps/spectrum/oneclick.jnlp file with WordPad.
2. Add the following argument into the <application-desc> section:
   ```
   <argument>-loginTitle Message_Text</argument>
   ```
   For example, you can replace the *Message_Text* variable with your own message, as follows:
   ```
   <argument>-loginTitle For authorized company use only. Unauthorized users will be punished to the fullest
    extent of the law.</argument>
   ```
3. Click File, Save.
   Your custom message is added to the OneClick Login dialog.



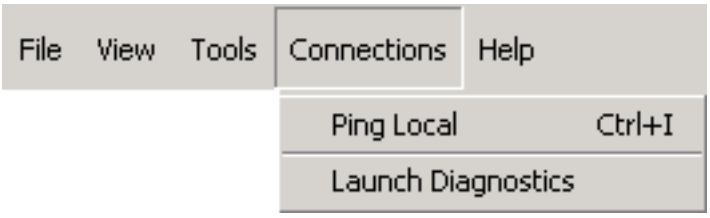
Customizing the OneClick Console Menu

This section describes how to add new menus and new menu items to the OneClick console. You can use new menu items to launch URLs, third-party applications, and scripts, and to pass parameters to them.

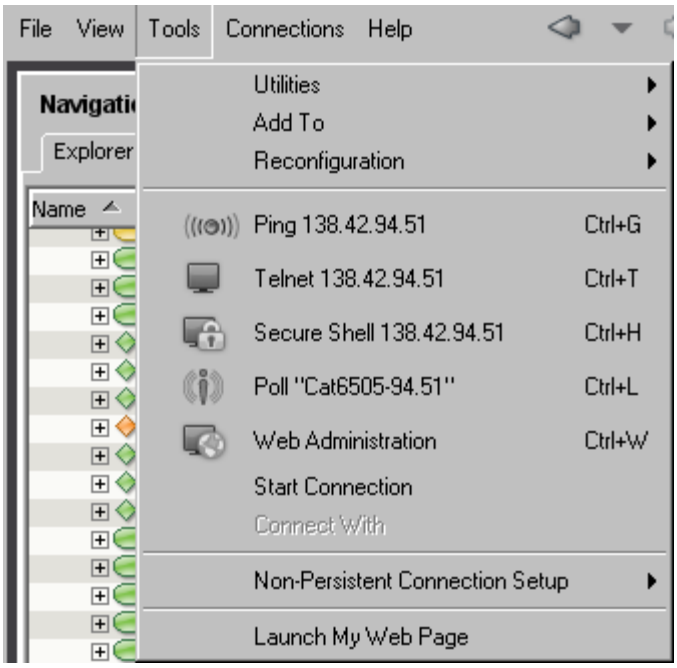
The custom-menu-config.xml File

The <*$SPECROOT*>/tomcat/webapps/spectrum/WEB-INF/console/config/custom-menu-config.xml file contains examples on how to add custom menus and custom menu items to your OneClick console as shown in the images. You need to copy this file in to the <*$SPECROOT*>/custom/console/config/ directory if the file is not already in this directory.

The following image shows the Connections menu and its two new menu items: Ping Local and Launch Diagnostics:



The following image shows that a new menu item called Launch My Web Page, which has been added to the existing Tools menu. This menu item has been created to launch a specified web page.



You create OneClick menus and menu items using the <menu> and <item> XML elements. The <menu> element can enclose one or more <item> elements that define the commands that are available on the menu. The <item> element can enclose several other elements that define how the menu item appears and behaves. See the following table for information about these elements.

Element	Parent Element	Description
<menu>	<root>	Defines the menu. The name attribute is used to define the name of the menu.
<separator>	<menu>	Used just before an <item> element to define a separator line as shown in the first figure in this section.
<item>	<menu>	Defines an item on a specific menu. The name attribute is used to define the name of the item.
<privilege>	<item>	Associates a privilege to the menu item. If the user is not given this privilege, the menu item is not displayed for that user.

<toolbar-image>	<item>	Specifies the image to display for the menu item and its associated toolbar button when the functionality is available to the user.
<toolbar-image-rollover>	<item>	Specifies the toolbar image that is displayed when a user places the cursor over the toolbar button.
<toolbar-image-disabled>	<item>	Specifies the toolbar image that is displayed when the functionality is disabled (not available to the user). A typical representation for this state is an image that is 80percent “grayed out.”
<accelerator>	<item>	Defines a keyboard sequence that executes the menu item.
<action>	<item>	Defines the action that takes place when the user clicks the menu item.
<hot-key>	<item>	Underlines the first instance of the indicated letter and enables the user to activate the menu item using this letter as a keyboard shortcut.

The subsequent sections of this chapter describe how to use the <menu> element to create a new menu and how to use the <item> element and its child elements to add menu items to a new or existing menu.

Add a New Menu

The <menu> element is used to create a OneClick console menu.

To add a new menu

1. Open the existing *<$SPECROOT>*/custom/console/config/custom-menu-config.xml file.
2. If the file does not exist, copy the file *<$SPECROOT>*/tomcat/webapps/spectrum/WEB-INF/console/config/custom-menu-config.xml into the *<$SPECROOT>*/custom/console/config directory, and then open it.
   The <root> element is the root element for this file. Define all new menus inside the <root> element.
3. Use the <menu> element to create new menus. This element has a single attribute, name, which defines the name of the menu.

   NOTE
   Some of the examples in the custom-menu-config.xml file show a fully qualified menu name that references a Java class created by OneClick engineers. For example, com.aprisma.spectrum.app.swing.window.menu.Tool is used as the value for the name attribute in the <menu> element that defines the Tools menu. You don't have to use a fully qualified name to create a new menu or to refer to an existing menu. Simply use the exact text that you would like to appear as the menu name on the toolbar.
4. Add items to the new menu by specifying them using the <item> element and its available child elements. If you do not specify menu items for a menu, the menu is not visible in the OneClick console.
5. Save the changes that you have made to custom-menu-config.xml.
6. To view and test the new menus, restart the OneClick console.

Example: Creating a New Menu

The following lines of XML create the Connections menu shown in The custom-menu-config.xml File.

```
<menu name="Connections">
   <item name="Ping Local">
    .
    .
    .
   </item>
   <item name="Launch Diagnostics">
       .
       .
       .
```

```
    </item>
</menu>
```

Add a New Menu Item

To add an item to an existing OneClick console menu or to a new menu that you created, you must create a new <item> element inside the <menu> element that you are customizing. The <item> element uses the <name> attribute to specify the name of the menu item.

NOTE
The new menu item is also added automatically to the right-click menu.

To add a new menu item

1. Open <*$SPECROOT*>/custom/console/config/custom-menu-config.xml.
2. Find the <menu> element that you created in Add a New Menu that defines the menu to which you want to add items. If the <menu> item does not yet exist, add it using the name attribute to define either an existing or a new menu.

   NOTE
   Some of the examples in custom-menu-config.xml show a fully qualified menu name that references a Java class created by OneClick engineers. For example, com.aprisma.spectrum.app.swing.window.menu.Tool is used as the value for the name attribute in the <menu> element that defines the Tools menu. You do not have to use a fully qualified name to create a new menu or to refer to an existing menu. For example, you can use <menu name="Tools"> to refer to the Tools menu.
3. Use the <item> element to create each new menu item. This element has one attribute, name, which defines the name of the menu item.
4. The <item> element has a series of child elements that enable you to define how the item behaves. These elements are listed in the table in the custom-menu-config.xml File, and they are further defined in the rest of this chapter. Use these elements to define the behavior of the menu item you have added.
5. Save the changes that you have made to custom-menu-config.xml.
6. To view the new menu items, restart the OneClick Console.

Example: Creating New Menu Items

The following example adds a menu item that is called Ping Local to a menu called Connections.

```
<menu name="Connections">
   <item name="Ping Local">
    <accelerator modifiers="2">VK_I</accelerator>
    <action>
        <filter>
            <has-attribute>AttributeID.NETWORK_ADDRESS</has-attribute>
        </filter>
        <context>com.aprisma.spectrum.app.topo.client.render.ModelContext </context>
        <context>com.aprisma.spectrum.app.alarm.client.group.AlarmContext</context>
        <launch-application>
            <platform>
                <os-name>Windows 9x</os-name>
                <command>command.com /c start "Local ping {0}" cmd.exe /c
            "ping.exe {0} &#38;&#38; pause"</command>
            </platform>
            <platform>
                <os-name>Windows</os-name>
                <command>cmd.exe /c start "Local ping {0}" cmd.exe /c "ping.exe
            {0} &#38;&#38; pause"</command>
            </platform>
```

```
            <platform>
                <command>/usr/dt/bin/dtterm -e ping -s {0}</command>
             </platform>
                <param>
                  <attribute>AttributeID.NETWORK_ADDRESS</attribute>
                </param>
        </launch-application>
     </action>
    </item>
 </menu>
```

Add Toolbar Images

To have a toolbar image available for each of the three toolbar image states, you must specify them in your menu item definition. The elements for toolbar states are:

- <toolbar-image>
- <toolbar-image-rollover>
- <toolbar-image-disabled>

You can use the following image formats for OneClick toolbar images: .png, .gif, .jpg, and .jpeg.

The recommended toolbar image size is 24 x 24 pixels. Store custom images in the *<$SPECROOT>*/custom/images directory. When you reference an image that is placed in this directory, specify the path from the images directory, for example, images/myimage.png.

The following line of code specifies a toolbar image using the relative path to the image file.

```
<toolbar-image>images/hints.gif</toolbar-image>
```

For a listing of all of the elements that are used in defining OneClick menu items, see the table in Contextually Apply the Action.

Define a Keyboard Accelerator

The <accelerator> element specifies a combination of keyboard input that executes a corresponding menu item.

Specify the code for the accelerator key using the capitalized letter on the keyboard, which is preceded by “VK_”.

The modifiers attribute indicates the modifier key combinations as an integer where:

- 1 = Shift
- 2 = Ctrl
- 3 = Ctrl+Shift
- 8 = Alt
- 9 = Alt+Shift
- 10 = Ctrl+Alt

You are not required to specify a keyboard accelerator for a customized menu item.

```
<accelerator modifiers="2">VK_L</accelerator>
```

In the preceding example, the specified action of the menu item is performed if the 'L' key is pressed while holding down the Control key (Ctrl+L).

Perform an Action

The <action> element specifies the action that is performed when the menu item is selected. You can use the child elements that are shown in the table in Contextually Apply the Action to specify a particular action.

The <context> element specifies the context in which the menu item is active so that the action can be executed. This applies to both the standard and the right-click menu.

Contextually Apply the Action

Actions do not always apply in all situations, such as an action that is applicable only when the user selects a model. Therefore, you can specify one of the following contexts for your actions:

- **ModelContext**
  Indicates that the action should be available when the user selects a model. The format for this context is as follows:
  ```
  <context>com.aprisma.spectrum.app.topo.client.render.ModelContext</context>
  ```
- **AlarmContext**
  Indicates that the action should be available when the user selects an alarm. The format for this context is as follows:
  ```
  <context>com.aprisma.spectrum.app.alarm.client.group.AlarmContext</context>
  ```
- **TableContext**
  Indicates that the action should be available when the user selects any table. The format for this context is as follows:
  ```
  <context>com.aprisma.spectrum.app.util.table.TableContext</context>
  ```
  If no tablename is specified, context is limited to any table. However, you can also limit context to a single table using the following format:
  ```
  <context>com.aprisma.spectrum.app.util.table.TableContext</context>
      <table-name>TableName</table-name>
  ```

You can specify one or a combination of contexts. If no specified context matches the current window context, the menu item is disabled. If no contexts are specified, the menu item is displayed in all contexts.

The following table describes the elements that are used to implement an action.

Element	Parent Element	Description
<context>	<action>	Limits the context in which the menu item is enabled and can perform the action.
<table-name>	<action>	Used with <context>, specifies the tablename when limiting the action to a single table. Works with TableContext only.
<column-name>	<param>	Used with <context> and <command>, specifies which values in a table column to pass into a script from a selected row in the table. Works with TableContext only.
<filter>	<action>	Limits the availability of menu items.
<has-attribute>	<filter>	Specifies the attribute on which to filter.
<and>, <or>, <value>, <equals>	<filter>	Creates an expression that can be used with a filter.
<launch-browser>	<action>	Launches a browser.
<launch-sso-browser>	<action>	Launches a browser and, if single sign-on is enabled in OneClick, includes a single sign-on token associated with the current session in the URL. This token can be used to reauthenticate the session across integrated web applications instead of prompting the user repeatedly for a username and password. **Note:** For information about how to set up single sign-on in OneClick using CA SiteMinder® or CA Embedded Entitlements Manager, see the Integration section for that application.
<url>	<launch-browser>	Specifies the URL to launch in the browser.

<launch-application>	<action>	Launches an application.
<launch-web-server-script>	<action>	Launches a script available on the web server.
<display-output>	<launch-application>, <launch-web-server-script>	Displays the output from the launched script.
<display-exit-status>	<launch-application>, <launch-web-server-script>	Displays the exit status of a launched script.
<command>	<launch-application>, <launch-web-server-script>, <platform>	Specifies the application or script that the menu item launches.
<run-for-multiple-alarms>	<launch-web-server-script>	Runs a script on web browser for selected multiple alarms. Applicable for alarms only. Starting from 10.3, DX NetOps Spectrum users are allowed to add the following tag, which runs a script for multiple alarms in combined mode. <multiple-contexts-enabled mode= "combined-execution" delimiter= ";"/> You can configure with the following modes: combined-execution - multiple alarms single executionseparate-execution (default value) - multiple alarms multiple executions (one execution per an alarm) For more information, see the Launch a Web Server Script For Multiple Alarms At The Same Time section in this page. **Note:** If you do not mention any mode or the mode name is incorrect or if the <multiple-contexts-enabled mode= /> tag is completely removed, then the default mode (separate-execution) is applied.
<platform>	<launch-application>	Used with <os-name>, specifies the application to launch based on the operating system of the OneClick client.
<validate>	<launch-application>	Used with the <command> element, specifies that the menu item should only be added to the menu if the command exists on the OneClick client and has execute permissions. If either condition is found to be false during OneClick startup, the menu item is not added to the menu. If the <validate> element is not used, the menu item is always added to the menu, but its state is determined by the value of other elements.
<os-name>	<platform>	Used with <platform>, specifies the application to be launched specific to the operating system of the OneClick client.
<param>	<url>, <command>	Specifies a parameter that is passed to a browser, executable, or script.
<attribute>	<param>	Specifies an attribute used as a parameter.

Limit the Availability of Menu Items

The <filter> element specifies a filter that further restricts the enabled state of the menu item. You can filter on any attribute of the selected context.

```
<filter>
   <has-attribute>AttributeID.NETWORK_ADDRESS</has-attribute>
```

```
</filter>
```

In the preceding example, the action needs the IP address of the alarmed model. Therefore, it should only be enabled if the alarmed model has the Network_Address (ID 0x12d7f) attribute.

You can specify complex attribute filters with any combination of nested “and” and “or” filters.

Example: Nesting Filters

The following example enables the item if the selected model has the Network_Address attribute and the Condition (ID 0x1000a) attribute is RED.

```
<filter>
   <and>
    <has-attribute>AttributeID.NETWORK_ADDRESS</has-attribute>
    <equals>
        <attribute id="AttributeID.CONDITION">
            <value>3</value> <!--red-->
        </attribute>
    </equals>
   </and>
</filter>
```

The file *<$SPECROOT>*/tomcat/webapps/spectrum/WEB-INF/common/schema/attribute-filter.xsd contains the complete syntax for attribute filters.

The following table defines commonly used attributes where an attribute ID is expected.

Constant	Attribute
AttributeID.NETWORK_ADDRESS	Network Address (ID 0x12d7f)
AttributeID.MTYPE_ID	Model Type Handle (ID 0x129ab)
AttributeID.MTYPE_NAME	Model Type Name (ID 0x10000)
AttributeID.MODEL_OBJECT	Model Handle (ID 0x11f53)
AttributeID.MODEL_NAME	Model Name (ID 0x1006e)
AttributeID.MODEL_CLASS	Model Class (ID 0x11ee8)
AttributeID.CONDITION	Condition (ID 0x1000a)
AttributeID.DOMAIN_ID	Landscape Handle (ID 0x129ac)
AttributeID.DOMAIN_NAME	Landscape Name (ID 0x11d42)
AttributeID.MAC_ADDRESS	MAC Address (ID 0x110df)
AttributeID.DEVICE_TYPE	Device Type (ID 0x23000e)

You can use the constants defined in the following table for alarm attributes:

Constant	Alarm Attribute
AlarmAttrID.ACKNOWLEDGED	Acknowledged (ID 0x11f4d)
AlarmAttrID.ALARM_FILTER_MH	Alarm Filter (ID 0x12a56)
AlarmAttrID.ALARM_ID	Full Alarm ID (ID 0x11f9c)
AlarmAttrID.INT_ALARM_ID	Integer Alarm ID (ID 0x4820067)
AlarmAttrID.ALARM_SOURCE	Alarm Source (ID 0x11fc4)
AlarmAttrID.ALARM_STATUS	Alarm Status (ID 0x11f4f)
AlarmAttrID.CAUSE_CODE	Cause Code (ID 0x11f50)

AlarmAttrID.CAUSE_LIST	Cause List (ID 0x12a05)
AlarmAttrID.CAUSE_TITLE	Cause Title (ID 0x4820020)
AlarmAttrID.CREATION_DATE	Creation Date (ID 0x11f4e)
AlarmAttrID.CLEARED_BY_USER_NAME	Cleared By User Name (ID 0x11f51)
AlarmAttrID.IMPACT_SEVERITY	Impact Severity (ID 0x1290d)
AlarmAttrID.OCCURRENCES	Occurrences (ID 0x11fc5)
AlarmAttrID.ORIGINATING_EVENT	Originating Event (ID 0x1296e)
AlarmAttrID.PERSISTENT	Persistent (ID 0x12942)
AlarmAttrID.PRIMARY_ALARM	Primary Alarm (ID 0x11f54
AlarmAttrID.SEVERITY	Severity (ID 0x11f56)
AlarmAttrID.TROUBLESHOOTER	Troubleshooter (ID 0x11f57)
AlarmAttrID.TROUBLE_TICKET_ID	Trouble Ticket ID (ID 0x12022)
AlarmAttrID.USER_CLEARABLE	User Clearable (ID 0x11f9b)

If you need to use an attribute other than one of the attributes listed in the 2 preceding tables, specify the attribute using its hexadecimal attribute ID.

Launch a Browser

The <launch-browser> element lets you launch a specified URL in a browser and pass parameters to the URL. These parameters can be hard-coded values or values from model attributes.

Example: <launch-browser> Code

The following example launches the default browser on the client machine. The <url> element specifies the URL pattern. You can specify parameters to substitute in the URL pattern by enclosing the parameter number (starting at 0) in curly braces {}. You then specify <param> elements for each parameter.

```
<launch-browser>
   <url>http://{0}</url>
   <param>
    <attribute>AttributeID.NETWORK_ADDRESS</attribute>
   </param>
</launch-browser>
```

DX NetOps Spectrum processes the <param> elements in order so the first one corresponds to the 0th parameter in the URL pattern. A <param> element has a specific syntax. The most commonly used is the <attribute> element. This element substitutes the value of the specified attribute for the selected context. In the preceding example, the value of the Network Address attribute is substituted in the URL pattern. For more complex parameters, see the definition of <param-type> in the file.

```
<$SPECROOT>/tomcat/webapps/spectrum/WEB-INF/common/schema/basic-config.xsd
```

Important Information About Specifying URLs

Provide the following information when specifying URLs.

Use Standard Characters

Whenever a URL is specified in XML customization code for OneClick, the URL formatting must adhere to the standards published in the Internet Engineering Task Force (IETF) RFC 1738. Use of non-standard characters in URLs results in unreliable browser performance including the browser not locating the specified web page.

URL Encoding of Spaces and Commas

If you are using spaces or commas, or other "reserved" or "unsafe" characters in URLs (see the tables later in this section), convert them to their ASCII equivalent value with the proper URL encoding. URL encoding of a character consists of a "%" symbol, followed by the two-digit hexadecimal representation (case-insensitive) of the ISO-Latin code point for the character. Examples for "space" and "comma" are:

- For spaces, use %20
- For commas, use %2C

> **NOTE**
> Some browsers may encounter problems processing URLs even when using this encoding.

Use of Ampersands

If you are using an ampersand in a URL or in XML customization code, you must convert it to &amp.

Use CDATA in XML

You can place URLs inside a CDATA section so that they are not parsed. This avoids possible problems with URLs and the XML parser.

Be sure to follow the requirements for CDATA, including:

- A CDATA section cannot contain the string "]] >", therefore, nested CDATA sections are not allowed.
- Also ensure there are no spaces or line breaks inside the "]] >" string.

URL Unsafe Characters

Some characters can be misunderstood within URLs for various reasons. These characters should also always be encoded. Unsafe characters and their hexadecimal encoding are provided in the following table.

Character	Code Points (Hex)
Space	20
Quotation marks (")	22
'Less Than' symbol ("<")	3C
'Greater Than' symbol (">")	3E
Pound' character ("#")	23
Percent symbol ("%")	25
Left Curly Brace ("{")	7B
Right Curly Brace ("}")	7D
Vertical Bar/Pipe ("|")	7C
Backslash ("\")	5C
Caret ("^")	5E
Tilde ("~")	7E
Left Square Bracket ("[")	5B
Right Square Bracket ("]")	5D
Grave Accent ("`")	60

URL Reserved Characters

URLs use some characters for special use in defining their syntax. When these characters are not used in their special role inside a URL, they need to be encoded. These characters and their hexadecimal encoding are provided in the following table.

Character	Code Points (Hex)
Dollar ("$")	24
Ampersand ("&")	26
Plus ("+")	2B
Comma (",")	2C
Forward slash/Virgule ("/")	2F
Colon (":")	3A
Semi-colon (";")	3B
Equals ("=")	3D
Question mark ("?")	3F
'At' symbol ("@")	40

Specify a Username

You can pass the OneClick username of the current user to an application, Web browser, or executable requiring a username. Use the following expression to specify the logged-in username of the user:

```
<param>
   <expression>
    com.aprisma.spectrum.app.util.context.DefaultApplicationContext.getGlobal
Parameter(com.aprisma.spectrum.app.util.context.ApplicationContext.
USER_PARAMETER_NAME)
   </expression>
</param>
```

Example: Pass Username to Browser

The following example launches a browser to a specified URL and passes the username to the browser.

```
<launch-browser>
   <url> http://acme.com?user={0}</url>
   <param>
    <expression>
 com.aprisma.spectrum.app.util.context.DefaultApplicationContext.getGlobalParameter(com.aprisma.spectrum.app.util.conte
    </expression>
   </param>
</launch-browser>
```

Launch an Application From OneClick

The <launch-application> element enables you to launch a specified command or executable.

Example 1: <launch-application>

The following example launches an application called myapp on the client machine and passes in the IP address of the selected model. As with the <launch-browser> action, you can substitute any number of parameters.

```
<launch-application>
   <command>myapp {0}</command>
   <param>
    <attribute>AttributeID.NETWORK_ADDRESS</attribute>
   </param>
</launch-application>
```

The <command> element specifies the command or executable to execute. You can provide the path to the command or executable in one of two ways:

- You can specify the path on each client via an environment variable. To create an environment variable in the Windows environment, select My Computer,Properties,Advanced, and then select the Environment Variables button.
- You can specify an absolute path to the command or executable. If you do this, keep in mind that the path must be the same on each OneClick client. Path statements in the Windows environment should use a double backslash instead of a single backslash, for example:
  ```
  C:\\Windows\\system32\\cmd.exe
  ```

  NOTE
  You can use the <validate> element to verify that the command or executable exists on the OneClick client and has execute permissions. If either of these conditions is found to be false during OneClick startup, the associated menu item is not added to the OneClick menu. (If the <validate> element is not used, the menu item is always added to the menu, but its state is determined by the value of other elements.)

If you use the <validate> element, you must specify an absolute path in the <command> element, as shown in the following example:

```
<launch-application>
   <command>c:\\windows\\system32\\notepad.exe</command>
   <validate/>
</launch-application>
```

The <command> element must conform to the following syntax rules:

- The command arguments are delimited by only spaces. If you would like to have a space within an argument, you must either place quotes around the argument or use the escape character '\' prior to the internal space(s).
- If you would like to embed quotes within an argument, you must place the escape character '\' prior to the quote.
- If any of your command arguments contain commas, DX NetOps Spectrum automatically places the argument within quotes. This is important to know in case that you are going to parse an argument that is a numeric value that contains commas.
- DX NetOps Spectrum replaces arguments that return null or have a string length of zero with empty quotes (“ ”).

Example 2: <launch-application>

The following example uses the <platform> element to specify different commands for different platforms. The <os-name> element specifies the operating system name and the <command> element specifies the command to execute on that operating system. The <os-name> element is optional. If you do not specify the <os-name>, the associated command is the default such that if no other platforms match, the default command is executed.

```
<launch-application>
   <platform>
    <os-name>Windows</os-name>
    <command>cmd.exe /c start "ping {0}" cmd /c "ping.exe {0}
    &#38;&#38;pause"</command>
   </platform>
   <platform>
    <os-name>SunOS</os-name>
    <command>>/usr/dt/bin/dtterm -e ping {0}</command>
```

```
    </platform>
    <param>
     <attribute>AttributeID.NETWORK_ADDRESS</attribute>
    </param>
 </launch-application>
```

At runtime, DX NetOps Spectrum compares the specified OS names to the OS name returned by the “os.name” Java property. DX NetOps Spectrum uses a best-match algorithm so only a prefix of the OS name need be specified. You may specify any of the following OS names:

- Windows for all Windows platforms
- Windows XP for Windows XP
- Windows 7 for Windows 7
- Linux for the Linux platform
- Mac for the Macintosh platform

If no specified platforms match, the associated menu item is disabled.

Launch a Web Server Script

The <launch-web-server-script> element launches a script on the web server machine. The <command> element specifies the script to execute. As with the <launch-browser> action, any number of parameters can be substituted. Since the script resides on the web server, which is restricted to a Windows machine, you do not use a <platform> element to denote the platform on which the script is running.

> **NOTE**
> This action can only be used to launch a script; it cannot be used to launch a user interface.

Example: <launch-web-server-script> Code

The following example launches “myscript” on the web server, passing it the model name and model type name of the selected model. Note that the path that is shown in the <command> element is a path for a Windows web server.

```
<launch-web-server-script>
       <command>c:/scripts/myscript {0} {1}</command>
       <param>
        <attribute>AttributeID.MODEL_NAME</attribute>
       </param>
       <param>
        <attribute>AttributeID.MTYPE_NAME</attribute>
       </param>
</launch-web-server-script>
```

Use the <platform> tag with the <launch-web-server-script> as described in Launch an Application From OneClick.

Launch a Web Server Script For Multiple Alarms in Combined Mode

Till 10.2.3, when you use the <run-for-multiple-alarms> tag with <launch-web-server-script>, it runs a script for selected multiple alarms in the OneClick console. Selected alarm attributes are passed as command line arguments to script for each selected alarm. The same script is run multiple times, once per selected alarm.

Starting from 10.3, you can use the following sub-element/tag to run a combined script in combined mode for multiple alarms. You can configure the<multiple-contexts-enabled mode= /> tag to run a script for several alarms at the same time in the alarm console.

<multiple-contexts-enabled mode= "combined-execution" delimiter= ";"/>

By default a space is considered as the 'delimiter' value. You can specify the delimiter value according to your requirement.

If you do not provide mode then the default value is applied as 'separate-execution' (<multiple-contexts-enabled mode=" separate-execution" />).

In combined execution mode, DX NetOps Spectrum creates a temporary file in $SPECROOT/tmp folder, which contains details of specified attributes of all alarms selected separated by delimiter. This file is passed as an argument to the combined execution script.

The temporary file contains the list of alarm attributes separated by a delimiter in a single line. Each line specifies a single alarm.

NOTE

After the script is complete for combined execution, you need to manually delete the temporary file which was created in the $SPECROOT/tmp folder.

Pass Table Values to a Script

Used with the <context> and <command> elements, the <column-name> element lets you add menu items in OneClick that execute a command using data from a selected row in a table. With this feature, DX NetOps Spectrum eliminates the need to look up attributes separately to build the logic. Instead, you can build the logic from selected column headings within a table and can pass the values from any row in the table to the script. This ability to pass values directly from a table is helpful when you need to use the data in an external script. For example, you can build an interface between DX NetOps Spectrum and an issue-tracking system. Then, you can create a menu item that creates trouble tickets from table data in OneClick.

NOTE

The <column-name> element works with TableContext only in the <context> element.

Example: <command> and <column-name> Commands

The following example passes values from three columns (Condition, Status, and Type) to the "NewTicket" command. The <command> element specifies the command pattern. You specify parameters to substitute in the command by enclosing the parameter number (starting at 0) in curly braces {}. You then specify <param> elements for each column that passes values to the command.

```
<context>com.aprisma.spectrum.app.util.table.TableContext</context>
<command>$SCRIPT_PATH/NewTicket.exe {0} {1} {2} {3}</command>
   <param>
    <column-name>Condition</column-name>
   </param>
   <param>
    <column-name>Status</column-name>
   </param>
   <param>
    <column-name>Type</column-name>
   </param>
```

DX NetOps Spectrum processes the <param> elements in order so the first one corresponds to the 0th parameter in the command pattern. By default, DX NetOps Spectrum passes the raw value to the command. To preserve the formatting information from the table, use the <formatted/> option, as follows:

```
<param>
   <column-name>Condition
    <formatted/>
   </column-name>
```

```
</param>
```

NOTE
The formatted option attempts to render the specified column as seen in the table. However, DX NetOps Spectrum cannot pass images as arguments to commands and, therefore, passes the raw value only.

Display the Status of a Launched Application or Script

Use the <display-exit-status> and <display-output> elements with <launch-web-server-script> and <launch-application> to display the exit status and the output from the script or application.

By default <display-exit-status> displays “Success” if the exit code is 0 and “Failed with error code #” otherwise. You can change the default behavior by specifying <status> child tags that map an exit code to a custom message to display.

Example: <display-exit-status> Code

Examine the following example using <display-exit-status>:

```
<display-exit-status>
   <status code="1">Could not open file</status>
   <status code="2">Bad parameter</status>
   <status code="3">Could not connect to the server</status>
   <status default="true">Unknown error code {0}</status>
</display-exit-status>
```

This example maps status codes 1, 2, and 3 to specific message strings. The last status code specifies default=“true”, mapping all other error codes except 0, which by default maps to “Success”. If exit code 0 does not indicate success, you can override it with a <status> tag. The {0} in the message string substitutes the exit code.

By default, <display-output> displays both the standard output and standard error output from the process. You can display only the standard output by specifying:

```
<display-output stdout="t"/>
```

or only the standard error output by specifying:

```
<display-output stderr="t"/>
```

NOTE
The <display-exit-status> and <display-output> elements can only be used for command line applications or scripts and not GUI applications. OneClick waits for the script to complete before being available to the user again.

Customizing OneClick Alarms

You can create custom alarm attributes and add them to the Alarm table and Information views.

Adding customized alarm attributes that display in OneClick is a multi-step process that requires using the Model Type Editor and the Alarm Preferences dialog, in addition to modifying the Alarm table and the Alarm Information view to display the custom alarm attributes.

To create and view custom alarm attributes

1. Create the custom alarm attributes by adding them to the GlobalAlarm model type using the Model Type Editor. The attribute group ID value must be set to equal 11f4c.

   NOTE
   For information on how to add custom alarm attributes to the GlobalAlarm model type see Model Type Editor. The section provides complete instructions on accessing and using the Model Type Editor.

2. Add a column to the Alarm table that displays the new customized alarm attribute by customizing the alarm-table-config.xml file. Modify Table Columns provides an example showing how to add a column to a table configuration file.
3. Add a field to the Alarm Details view configuration that displays the alarm attribute in the alarm's Information tab by customizing the view-alarmdetails-config.xml file. Extend or Modify an Information View provides information and examples on how to add a subview to an existing information view.
   You can also apply a custom privilege to the new alarm attribute by customizing the custom-privileges.xml file. Doing this limits which users or user groups or specific privileges are required to view the customized alarms. The name of the privilege must use the following syntax:
   ```
   alarm-write-<attribute ID in hex>
   ```
   The following example adds the new privilege to the Alarm Management group.
   ```
   <alarm-write-ffff0000 type="write">
       <group scope="alarm-manager">alarm-mgmt</group>
       <label>New Alarm Attr</label>
   </alarm-write-ffff0000>
   ```
4. Save and close the XML files you edited.
5. You must restart the OneClick server in order to apply the new privilege.
6. You must restart the OneClick Console to view the changes made to the Alarms table and Alarm Information view.

Customizing OneClick Tables

This section discusses some of the ways that you can modify existing tables found in the OneClick Console. A list of the table elements available in OneClick and their descriptions are presented in Example: Defining a Table Column. Specific examples for modifying table columns are presented in the sections listed below.

Modify Table Columns

If you want to display additional attributes in a OneClick console table, you can do so by making modifications to the XML using a customization file. The modifications need to be made in a separate XML file in the appropriate directory under <*$SPECROOT*>/custom/.

Extend a Factory Default File Using IDREF

OneClick requires that you write your customization code in a new file located in the <*$SPECROOT*>/custom/topo/config directory that uses the same name as the factory default file that builds the table you are modifying. Use the IDREF attribute to "reference" the factory file and extend it with the new customization file. See Create Customizations for details on creating customization files in OneClick.

Example: Referencing a Column File from a Table Configuration File

The following example shows a portion of an XML file used to define a table. Rather than defining each column in the same file that defines the entire table, the example uses separate files to define the first two columns in the table. The example uses the idref attribute with each <column> element to link to the file that defines the column.

```
<table id="table-licenses-config"
   xmlns="http://www.aprisma.com"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.aprisma.com
   ../../common/schema/table-config.xsd">
   <swing-row-template>
    <enumerated-color idref="alternatingrow-color-config"/>
   </swing-row-template>
   <swing-table-template>
```

```
     <show-vertical-lines>true</show-vertical-lines>
     <show-horizontal-lines>false</show-horizontal-lines>
    </swing-table-template>
    <swing-header-row-template>
     <static-color idref="row-header-color-config"/>
    </swing-header-row-template>
    <column-list>
     <column idref="column-servicestate-config"/>
     <column idref="column-modelname-config">
     <default-width>300</default-width>
     </column>
    </column-list>
    .
    .
 </table>
```

The first column is defined in the column-servicestate-config.xml file. The beginning portion of this file is shown below. Note the id attribute used with the <column> element to define this file as “column-servicestate-config”.

```
 <column id="column-servicestate-config"
    xmlns="http://www.aprisma.com"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.aprisma.com
    ../../common/schema/table-config.xsd">
```

Modify a Table Column

The following steps describe the general process for modifying table columns in OneClick. Specific examples are provided in the following sections.

To modify a table column

1. Identify the default factory XML file that builds the table that you want to modify.
   Many of the table files used to display data in the OneClick console are located in *<$SPECROOT>*/tomcat/webapps/spectrum/WEB-INF/topo/config. All of the table files are named for the functionality that they display. For example, the table used to display interface information for each model is called table-common-ifconfig-config.xml.
2. Create the file in which to add your modifications.
   In this example the default file is *<$SPECROOT>*/tomcat/webapps/spectrum/WEB-INF/topo/configtable-common-ifconfig-config.xml. Create a file with the same name in the *<$SPECROOT>*/custom/topo/config directory. If a file with that name already exists in this directory, that is an indication that previous customizations have been made to this table. In this case, add your customized code to the existing file.
3. Open the file in a text editor in order to make the appropriate modifications.
4. Use idref to reference the table configuration you are extending with this new column configuration (see Step 1).
5. Construct a new column using the XML elements defined in Example: Defining a Table Column. The example that follows this procedure shows how some of these elements are used to define a column.
6. Find the <column-list> element in the XML file. The <column-list> element contains all of the <column> elements used to define each column in the table.
7. Define a <column> element within the <column-list> element. The columns display in the order they appear within the <column-list> element.
8. Insert the <name> element to define the title of the column.
9. Insert a <content> and an <attribute> element to define the contents you want to display in the column.
10. (Optional) Use the <default-width> element to define the default width of the column.
11. Save and close the modified file, and restart the OneClick console to view the changes.

Example: Defining a Table Column

```
<column-list>
    <column>
        <name>Interface</name>
        <content>
            <attribute>0x100c4</attribute>
        </content>
        <default-width>30</default-width>
    </column>
    .
    .
    .
</column-list>
```

The following table describes the elements used for modifying a table.

Element	Parent Element	Description
<table>	Not applicable	This is the root element, and encloses all child elements used to create a table.
<swing-table-template>	<table>	Used to define the appearance of the table.
<show-vertical-lines>	<swing-table-template>	Defines whether to show vertical lines in the table, values are true or false.
<show-horizontal-lines>	<swing-table-template>	Defines whether to show horizontal lines in the table, values are true or false.
<line-color>	<swing-table-template>	Defines the color of the lines used to create the table. The default value is light-background_color. Other values can be found in the *<$SPECROOT>*/tomcat/webapps/spectrum/WEB-INF/console/common/config/common-color-config.xml file.
<show-tree-lines>	<swing-table-template>	Defines whether the table will be shown with dashed lines connecting tree nodes, values are true or false.
<preferred-width>	<swing-table-template>	Defines (in pixels) the default width of the table.
<preferred-height>	<swing-table-template>	Defines (in pixels) the default height of the table.
<swing-header-row-template>	<table>	Used to define the appearance of the header row of the table.
<static-color>	<swing-header-row-template>	Specifies the color for the header row. Use value idref=row-header-color-config for consistency.
<swing-row-template>	<table>	Used to define the appearance for the body rows in the table.
<enumerated-color>	<swing-row-template>	Used to specify different colors for each row of the table, the default value used is alternating row-color-config.
<static-color>	<swing-row-template>	Used to specify a single color used for all of the rows in the table.
<column-list>	<table>	Used to define the list of columns to be used in the table
<column>	<column-list>	Used to define a column in the column list.
<name>	<column>	The name of the column. The text used here will appear in the table header for this column.
<editable>	<column>	Defines whether the value in the table can be edited. If this value is set to true, a set link appears next to the value.
<content>	<column>	Defines the value placed in the column. This is the value used for sorting and filtering. See "Rendering a value" for information on what child elements can be specified. The final displayed text can be further manipulated by defining a <swing-cell-template> tag. See Define How Cells Display in Table Columns for information on <swing-cell-template>.

<renderer>	<content>	Specifies the renderer to be used for the content of the column. See Customize the Port Name Column of the Interface Table for more information.
<dynamic-renderer>	<content>	Enables you to specify a renderer depending on model class or model type. See About <dynamic-renderer> for detailed instructions on usage.
<expression>	<content>	Used to define an expression to produce a value for the column. See XML Usage Common to All Customization Files for more information
<message>	<content>	Used for specifying a plain text value for the column.
<select>	<content>	Used to select a value for the column based on certain criteria. See XML Usage Common to All Customization Files for more information.
<attribute>	<content>	Used to specify an attribute ID. The value of the attribute will be placed in the column.
<swing-cell-template>	<column>	Used to define how the cell in the column is displayed. See Define How Cells Display in Table Columns for detailed information on using this element.
<image>	<swing-cell-template>	Used to display an image in a cell.
<attribute>	<image>	The attribute used to determine the image selection.
<select>	<image>	Used to define the image that is selected. See XML Usage Common to All Customization Files for more information.
<text>	<swing-cell-template>	Used to display a string of text in a cell.
<renderer>	<text>	The renderer used to manipulate the text. See Customize the Port Name Column of the Interface Table for an example. See Manipulate Attribute Output Using Renderers for detailed information on OneClick renderers.
<param>	<renderer>	Specifies any parameters to be used by the renderer. See About Parameters for detailed information.
<renderer-class>	<swing-cell-template>	The class to use for rendering something other than an image or text.
<editable>	<column>	Allows the user to edit the value in a column. See Make a Table Column Editable for more information.
<hidden-by-default>	<column>	Use this element to hide the column by default. The user must add the column to the table using the preferences dialog.
<default-width>	<column>	The default width of the column in pixels.
<default-sort>	<column-list>	Used in conjunction with the <sort-column-list> element to determine how the table is sorted by default.
<sort-column-list>	<default-sort>	A list of columns that will be sorted on (maximum of three).
<sort-column>	<sort-column-list>	The column to be sorted on. Columns will be sorted in the order that they are listed.
<name>	<sort-column>	The name of the column to sort on. This must match the name defined for the column.
<direction>	<sort-column>	The direction of the sort, values can be ascending or descending.

Define How Cells Display in Table Columns

The <swing-cell-template> defines the final presentation of content within the cell of a table column. Use this element in conjunction with the <content> element to specify how the content is presented to users. This element gives the developer flexibility in processing raw OneClick data using buttons, scrollable lists, wrapping text, and other techniques.

If you do not specify <swing-cell-template> in a column cell definition, the raw output from the <content> element is displayed without refinement. The elements you can use in defining <swing-cell-template> are described in the following table.

Element	Parent Element	Description
<image>	<swing-cell-template>	Defines an image displayed in the cell.
<text>	<swing-cell-template>	Defines the text to display in a cell. If no child elements are specified, then the text from the <content> element is displayed.
<renderer-class>	<swing-cell-template>	Specifies a Java class to use for the final presentation of data in a cell. See Use Renderers to Present Data in Column Cells for the renderers available to use with this element.

The following Java classes are available to use with <renderer-class> to render or manipulate and format raw data for display in table column cells.

TextAreaCellRenderer

Use this renderer to display text that wraps at the cell width.

Classname: com.aprisma.spectrum.app.swing.table.render.TextAreaCellRenderer

Supported Parameters: None

Example: Displaying Wrapping Text in a Cell

The following example creates a cell that displays text and allows it to wrap at a defined width, displaying in multiple lines of fixed width. This technique is used in the Notes field on the device's Information tab. Text entered by operators in this field wraps at a predetermined column width.

```
<swing-cell-template>
   <text/>
   <renderer-class>
    com.aprisma.spectrum.app.swing.table.render.TextAreaCellRenderer
   </renderer-class>
</swing-cell-template>
```

ListAttributeRenderer

Displays the values from a list-type attribute in a scrollable list.

Classname: com.aprisma.spectrum.app.swing.table.render.ListAttributeRenderer

Supported Parameters:

- <attrID> - Required parameter. ID of the DX NetOps Spectrum attribute displayed in the cell.
- <maxRowsToDisplay> - Integer; specifies the maximum number of viewable list entries, scrolling is turned on after this maximum is exceeded.
- <order> - True or false; if true, the users can modify the list order if the column is editable.
- <add> - True or false; if true, users can add new entries to the list if the column is editable.
- <remove> - True or false; if true, users can remove entries from the list if the column is editable.
- <valuePrompt> - The text displayed when prompting the user to add a new value to the list. If not specified, a default prompt is used.

Example: Displaying a Scrollable List in a Cell

The following example defines the maximum number of rows displayed in the cell as four; if there are more than four rows, the list becomes scrollable.

```
<swing-cell-template>
   <renderer-class>
    com.aprisma.spectrum.app.swing.table.render.ListAttributeRenderer
    <param name="maxRowsToDisplay">4</param>
    <param name="attrID">0x25e0039</param>
    <param name="add">true</param>
    <param name="remove">true</param>
    <param name="valuePrompt">SNMP Port</param>
   </renderer-class>
</swing-cell-template>
```

ListAttributeOIDRenderer

Displays the OIDs from a list-type attribute in a scrollable list.

Classname: com.aprisma.spectrum.app.swing.table.render.ListAttributeOIDRenderer

Supported Parameters: See ListAttributeRenderer.

- <oidPrompt> - The text displayed when prompting the user to add a new OID to the list. If not specified, a default prompt is used.

ActionButtonCellRenderer

Displays a button in the cell that sends an action to the model when clicked.

Classname: com.aprisma.spectrum.app.topo.client.render.ActionButtonPanelCellRenderer

Supported Parameters:

- <actionID> - Required parameter; the DX NetOps Spectrum attribute ID for the action the button sends.
- <text> - Required parameter; the text displayed on the button.
- <toolTipText> - The text displayed in a tooltip for the button.
- <prompt> - The text displayed when prompting the user to confirm a button click. If not specified, no prompt is displayed.
- <confirmSuccess> - True or false; if true, a message is displayed if the action was successful.

Example: Displaying an Action Button in a Cell

```
<swing-cell-template>
   <renderer-class>
    com.aprisma.spectrum.app.topo.client.render.ActionButtonCellRenderer
    <param name="actionID">0x0001011f</param>
    <param name="text">Reevaluate Model Names</param>
    <param name="toolTipText">
        Reevaluates all model names based on the model naming order of VNM
    </param>
    <param name="confirmSuccess">true</param>
   </renderer-class>
</swing-cell-template>
```

ActionButtonPanelCellRenderer

Displays one or more buttons that each send an action to the model. All parameters are specified as a semi-colon separated list, one per button.

Classname: com.aprisma.spectrum.app.topo.client.render.ActionButtonPanelCellRenderer

Supported Parameters: See ActionButtonCellRenderer.

Example: Displaying Multiple Action Buttons in a Cell

The following example defines two buttons, “Reconfigure Model” and “Discover Connections.” The parameters with a value for each button separate each value with a semi-colon (;).

```
<swing-cell-template>
   <renderer-class>
    com.aprisma.spectrum.app.topo.client.render.ActionButtonPanelCellRenderer
    <param name="actionID">0x1000e;0x25e0022</param>
    <param name="text">Reconfigure Model;Discover Connections</param>
    <param name="confirmSuccess">true;true</param>
   </renderer-class>
</swing-cell-template>
```

AttrToggleButtonCellRenderer

Displays a button with text that can be one of two values based on two specified values of an attribute. When the button is clicked, the opposite value is written to the attribute.

Classname: com.aprisma.spectrum.app.topo.client.render.AttrToggleButtonCellRenderer

Supported Parameters:

- <attrID> - Required parameter. ID of the DX NetOps Spectrum attribute that the button toggles the value of.
- <firstValueMapping> - Required parameter. Specifies the first attribute value and the text to display when the attribute equals this value, separated by a semi-colon.
- <secondValueMapping> - Required parameter. Specifies the second attribute value and the text to display when the attribute equals this value, separated by a semi-colon.
- <promptOnFirstValue> - Specifies the text used to prompt the user for confirmation when the button is clicked and the attribute equals the first value. If not specified, no prompt is displayed.
- <promptOnSecondValue> - Specifies the text used to prompt the user for confirmation when the button is clicked and the attribute equals the second value. If not specified, no prompt is displayed.
- <disableOnFirstValue> - True or false; if true, the button is disabled when the attribute equals the first value.
- <disableOnSecondValue> - True or false; if true, the button is disabled when the attribute equals the second value.

Example: Displaying a Toggle Action Button in a Cell

```
<swing-cell-template>
   <renderer-class>
    com.aprisma.spectrum.app.topo.client.render.AttrToggleButtonCellRenderer
    <param name="attrID">0x11b6e</param>
    <param name="firstValueMapping">true;Start</param>
    <param name="secondValueMapping">false;Stop</param>
    <param name="promptOnSecondValue">Are you sure you want to stop?</param>
   </renderer-class>
</swing-cell-template>
```

BoldAttributeTableCellRenderer

Displays the cell text in bold.

Classname: com.aprisma.spectrum.app.swing.table.render.BoldAttributeTableCellRenderer

Example: Displaying Bold Text in a Cell

```
<swing-cell-template>
```

```
    <text/>
    <renderer-class>
     com.aprisma.spectrum.app.swing.table.render.BoldAttributeTableCellRenderer
    </renderer-class>
 </swing-cell-template>
```

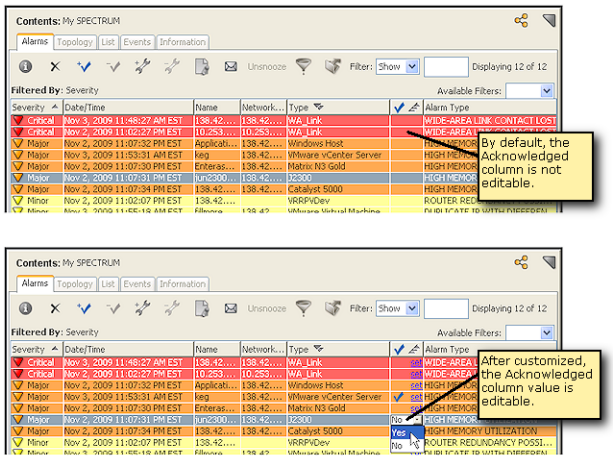
Make a Table Column Editable

In some cases, you can allow the user to edit the value found in a particular table cell in the column. The Acknowledged column in the Alarms table and the Admin Status column in the Interface Configuration table are examples of columns that allow users to edit.

Customize the Alarm Table Acknowledge Field

Some organizations may prefer to see text displayed in the Acknowledged column (“yes” or “no”) instead of the default checkmark or blank space. This section describes how to make the Acknowledged column in the Alarms table editable by extending the factory XML file.

The following figure shows the default Acknowledged field in the Alarms table and the Acknowledged field after customizing it to make it editable.



Check to see if the file *<$SPECROOT>*/custom/alarm/config/alarm-table-config.xml already exists. It exists in this directory if previous customization has been made to this file. If the file does not exist, create it.

To create an editable Acknowledged column

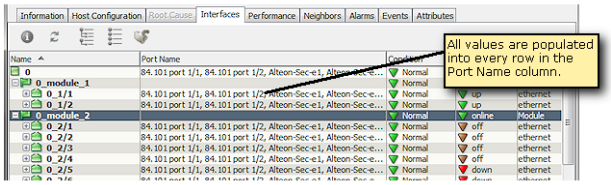
1. Open the file and add the following XML

```
<table idref="alarm-table-config">
    <column-list>
        <column idref="column-alarmacknowledge-config">
            <editable/>
            <default-width>37</default-width>
        </column>
    </column-list>
</table>
```

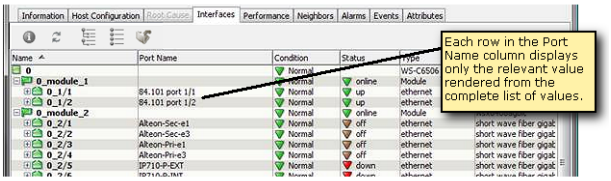
2. Save and close the alarm-table-config.xml file.
3. Restart the OneClick client for the changes to take effect.
   This code overrides the "column-alarmacknowledge-config" entry in the factory alarm-table-config.xml file.

Display Instanced Attribute Values in Separate Table Rows

If you add a new table column for attributes with instanced values (such as, PortName and BoardToPortMap), adding ObjectIDValueListRenderer *and* the <refID> to your XML file helps ensure the data displays correctly. Using this renderer without the <refID> value, DX NetOps Spectrum displays all values returned for the selected OID value list attribute in every row.



When the <refID> value is added to the XML file, DX NetOps Spectrum displays the attribute values correctly, rendering only the particular value from the list that applies to a given row.



The supported parameters include the following:

- <attrID> - Required parameter. ID of the DX NetOps Spectrum attribute displayed in the cell.
- <refID> - Required parameter. Reference ID for the DX NetOps Spectrum attribute that is used for indexing the attribute values.

Example: Displaying a Single OID Value in Each Row of a Column

The following example shows the contents of the column-portname-config.xml file used to create a custom table column for the Port Name attribute.

```
<?xml version="1.0" encoding="UTF-8"?>
<column id="column-portname-config"
    xmlns ="http://www.aprisma.com"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.aprisma.com
         ../../common/schema/column-config.xsd">
   <name>Port Name</name>
   <content>
    <attribute>0x11c0056</attribute>
    <renderer>
         <param name="attrID">0x11c0056</param>
         <param name="refID">0x1297f</param>
         com.aprisma.spectrum.app.util.render.ObjectIDValueListRenderer
    </renderer>
   </content>
</column>
```

Other Customizations in Tables

Customize Alarm Table Row Colors

You can customize the background color displayed in each row of the Alarms table. By default, each row in the Alarms table takes on the background color associated with the alarm listed in the row.

Customize the Alarms table to display in alternating gray and white rows by changing the swing-row-template definition from enumerated-severity-color-config to alternatingrow-color-config.

To customize the Alarm table colors

1. Create an alarm-table-config.xml file (if one does not already exist) in the <*$SPECROOT*>/custom/alarm/config directory.
2. Use the idref attribute to reference the factory file being extended: alarm-table-config.xml.
3. Define <enumerated-color> to reference alternatingrow-color-config.
4. Save and close the file.
5. Close and restart the OneClick Console to view the changes.

Example: Modifying the Alarm Table Row Color

```
<table idref="alarm-table-config">
   <swing-row-template>
    <enumerated-color idref="alternatingrow-color-config"/>
   </swing-row-template>
</table>
```

Set Up a Default Sort

You can create default sort criteria for a table using the <default-sort> element and its child elements. You can sort on a maximum of three columns. DX NetOps Spectrum sorts the columns in the order in which they are listed in <sort-column-list>.

To set up a default sort

1. Identify the appropriate XML file that is used to build the table. All of the table files that are used to display data in the OneClick console are located in the <*$SPECROOT*>/tomcat/webapps/spectrum/WEB-INF/topo/config directory. All of the table files are named after the functionality that they display. For example, the table used to display interface information for each model is called table-common-ifconfig-config.xml.
2. Determine if this file exists in the <*$SPECROOT*>/custom/topo/config directory. It exists in this directory if previous customizations were made to this file. If it is not there, copy the factory table file into the <*$SPECROOT*>/custom/topo/config directory.
3. Open the file in a text editor in order to make the appropriate modifications.
4. Find the closing tag for the column-list element, </column-list>.
5. Insert the XML to create the default sort after the </column-list> using the <default-sort> element.
6. Save and close the XML file.
7. Restart the OneClick client for your changes to take effect.

Example: Using <default-sort>

```
<table>
<column-list>
.
.
```

```
.
</column-list>
<default-sort>
   <sort-column-list>
    <sort-column>
        <name>Name_of_first_column_to_sort</name>
        <direction>ascending</direction>
    </sort-column>
    <sort-column>
        <name>Name_of_second_column_to_sort</name>
        <direction>ascending</direction>
    </sort-column>
    <sort-column>
        <name>Name_of_third_column_to_sort</name>
        <direction>ascending</direction>
    </sort-column>
   </sort-column-list>
</default-sort>
```

NOTE
Include the specific text for the contents of each <name> element based on the column list entries.

Customize the Port Name Column of the Interface Table

You can change the tree column in the interface table to display something other than the model name which is the default attribute displayed. This type of customization requires you to override the factory content-iftree-config.xml file.

To customize the port name column

1. Determine if the file *<$SPECROOT>*/custom/topo/config/content-iftree-config.xml exists. It already exists in this directory if previous customizations have been made to this file. If the file does not exist, create this file in the specified directory by copying it from *<$SPECROOT>*/tomcat/webapps/spectrum/WEB-INF/topo/config and pasting it into *<$SPECROOT>*/custom/topo/config.
2. Find the following code in the content-iftree-config.xml file that creates the column containing the model name attribute:

```
<content id="content-iftree-config"
    xmlns ="http://www.aprisma.com"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.aprisma.com
    ../../common/schema/column-config.xsd">
    <attribute>AttributeID.MODEL_NAME</attribute>
<!-- If the model name is not filled in or it's a GnPort, use Component_OID-->
    <expression>
        (((String)value()).length() == 0 ||
        ((String)value()).equals("GnPort")) ?
        attr(0x1006a).toString(): value()
    </expression>
</content>
```

NOTE
The contents of the <expression> element are specific to the use of the Model_Name attribute and should be removed when specifying other attributes.

3. Customize the attribute displayed in the Port Name column by changing the <attribute><value></attribute> element to contain the attribute you want to display. To display a different attribute, change <value> to the desired attribute ID.

You can specify the integer ID (for example, 0x129e0) or a predefined constant. The table beneath this procedure lists the predefined Port Attributes and their integer IDs.
To display the port description, edit the file as follows:

```
<content id="content-iftree-config"
    xmlns ="http://www.aprisma.com"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.aprisma.com
    ../../common/schema/column-config.xsd">
    <attribute>AttributeID.PORT_DESCRIPTION</attribute>
</content>
```

4. Save and close <*$SPECROOT*>/custom/topo/config/content-iftree-config.xml.
5. Close and restart the OneClick client to see the changes take effect.

Predefined Port Attribute Value Constant	Integer ID
AttributeID.PORT_DESCRIPTION	0x129e0 (ifDescr)
AttributeID.PORT_TYPE	0x129ed (ifType)
AttributeID.COMPONENT_OID	0x1006a

Sort Interfaces Table by ifIndex

DX NetOps Spectrum sorts the interfaces in the Interfaces table by the first column, which is the interface model name. DX NetOps Spectrum sorts this field alphabetically based on the textual values.

To sort the interfaces table numerically by ifIndex value

1. Edit the definition of the interfaces table content definition file acquire the ifIndex data. Check the <*$SPECROOT*>/custom/topo/config directory to see if the content-iftree-config.xml file exists. The file exists in this directory if previous customizations have been made to it.
2. If the file is not in this custom directory, create it by copying it from <*$SPECROOT*>/tomcat/webapps/spectrum/WEB-INF/topo/config and pasting it into the <*$SPECROOT*>/custom/topo/config directory.
3. Open the file and find the line that reads:
   ```
   <attribute>AttributeID.MODEL_NAME</attribute>
   ```
4. Change this line to read:
   ```
   <attribute>0x1006a</attribute>
   ```

   NOTE
   The 0x1006a attribute is Component_OID (which is rolled from ifIndex).The contents of the <expression> element are specific to the use of the Model_Name attribute and should be removed when specifying other attributes.
5. Save your changes to the content-iftree-config.xml file.
   In the following steps, you edit the interface table column definition file to relabel the Model Name column to be the Index column, as it will display the interface index value instead of the model name.
6. Open the <*$SPECROOT*>/custom/topo/config directory and check to see if the column-iftree-config.xml file already exists there. If it does not, go to the <*$SPECROOT*>/tomcat/webapps/spectrum/WEB-INF/topo/config directory and find the column-iftree-config.xml file. Copy it and paste it into the <*$SPECROOT*>/custom/topo/config directory.
7. Find the line that reads:
   ```
   <name>com.aprisma.spectrum.app.util.render.ModelNameColumn</name>
   ```
8. Change this line to read:
   ```
   <name>Index</name>
   ```

9. Save your changes to the column-iftree-config.xml file.
   In the following steps, you edit the interface table configuration file to change the name element value from the model naming java class to the text “Index”.
10. Open the <*$SPECROOT*>/custom/topo/config directory and check to see if the interfaces-table-config.xml file already exists there. If it does not, go to the <*$SPECROOT*>/tomcat/webapps/spectrum/WEB-INF/topo/config/ directory and find the interfaces-table-config.xml file. Copy it and paste it into the <*$SPECROOT*>/custom/topo/config directory.
11. Find the line that reads:
    ```
    <name>com.aprisma.spectrum.app.util.render.ModelNameColumn</name>
    ```
12. Change this line to read:
    ```
    <name>Index</name>
    ```
13. Save your changes to the interfaces-table-config.xml file.
14. Close all of the text files and restart the OneClick console in order for the changes to take effect.

If you would like to retain the interface model name in the table, implement the following additional instructions:

1. Open the <*$SPECROOT*>/custom/topo/config/interfaces-table-config.xml file.
2. Find the following entry within the <column-list> element:
   ```
   <column idref="column-iftree-config">
       <default-width>150</default-width>
   </column>
   ```
3. Add the following three lines to the end of this entry.
   ```
   <column idref="column-ifmodelname-config">
       <default-width>150</default-width>
   </column>
   ```
4. Save your changes and close the XML file.
5. Close and restart the OneClick console in order for the changes to take effect.

Adding Support for Model Types or Model Classes

This section describes how to add or extend OneClick support for a DX NetOps Spectrum model type or model class. For example, you may want to add specific OneClick support for a management module that you have created with the Generic SNMP Toolkit in DX NetOps Spectrum.

Create a Registration

If you have created a new model type or model class, you can use the XML elements described in this chapter to configure the associated model appearance, available information, and views within the OneClick interface.

Register the Model Type or Model Class in custom-app-config.xml

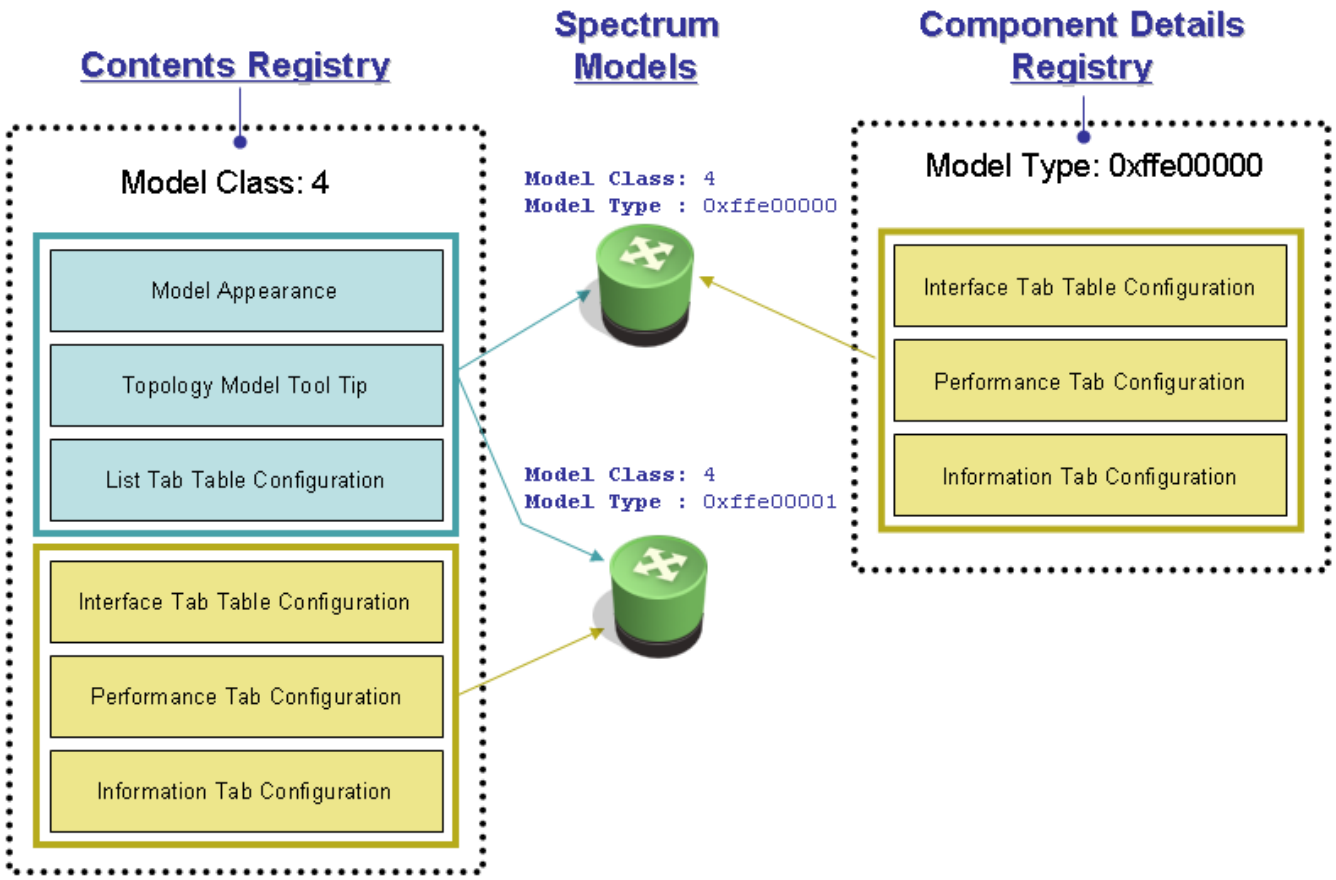
OneClick registration for new model types and model classes is performed within the custom-app-config.xml file. Doing so enables one to configure OneClick support on a per model type and/or model class basis. There are two methods for registration:

- Contents registry
- Component details registry

Both of these registrations work in conjunction with each other to provide effective OneClick support for DX NetOps Spectrum models. The component details registry acts as an extension of the contents registry in order to define general and specific OneClick device support.

Because most models of a given model class will share the same OneClick device support, typically you define a contents registry for the model class to define this general support. However, for a given model class, you might want to expose different information in the Component Details panel based on model type. To accomplish this, you define a component details registry for the applicable model type. As a result, the model receives its general, shared content from the contents registry, and it receives its more specific component details content from the component details registry.

Consider the example illustrated in the following figure. There are two routers of model class 4 (switch-router). They both receive their general information from the contents registry that has been registered with model class 4. However, the model of type 0xffe00000 receives its component details content from the component details registry. Because the model of type 0xffe00001 does not have a component details registry, it receives all of its configuration content from the contents registry.



Define General OneClick Device Support Based on Model Class

The contents registry (denoted by the <oi> element) is primarily used to define the general, shared configuration for a model class or a group of model types. By default, DX NetOps Spectrum defines this general behavior for most model classes. Therefore, if you set the appropriate model class for your custom model type, it automatically inherits this generic content.

If the default configuration meets most of your requirements, but you want to modify the information in the Component Detail panel, you should only create a component details registry for the applicable model type.

However, if you create a new model class, or you want to define a specific model appearance for a model type, you need to create a contents registry entry using the XML elements described in the following table.

Element	Parent Element	Description
<app-config>	ot applicable	The root element for the custom-app-config.xml file.
<contents-registry>	<app-config>	Used to associate a single or group of model classes and/or model types to configuration files that define model appearance and tooltip content. (Optional) You can use the scope attribute to specify the location of the configuration xml files you are using (when you are not using the default locations of console, topo, or common). The scope attribute has to match the directory name where the existing configuration xml files reside, as follows: <$SPECROOT>/tomcat/webapps/spectrum/ WEB-INF/<scope>/config For example, <contents-registry scope="devman"> specifies the devman/config directory, which contains the configuration xml files for device management views and subviews.
<model-class>	<contents-registry>	Registers a model class for this contents registry.
<model-type>	<contents-registry>	Registers a model type for this contents registry. Model types should only be registered for a content registry if you need a unique topology icon or tooltip. If you only want to configure component detail content for a model type, use the <component-details-registry> instead of the <contents-registry>.
<is-derived-from>	<contents-registry>	Registers a parent model type and all of its derived model types for this contents registry. Model types should only be registered for a contents registry if you need a unique topology icon or tooltip. If you want to configure only component detail content for a model type, use the <component-details-registry> instead of the <contents-registry>.
<icon-reg-id>	<contents-registry>	The icon registration ID that identifies the icon configuration to use. The icon configuration defines the appearance of the icon in OneClick.
<tooltip-config>	<contents-registry>	The XML that defines the tooltip content for topology icons.
<table-config>	<contents-registry>	Specifies the XML that configures the columns available on the List tab in the Contents panel.
<information-config>	<contents-registry>	Specifies the XML that defines the content of the model's Information tab.
<performance-config>	<contents-registry>	Specifies the XML that defines the contents of the model's Performance tab, where one or more graphs can be defined to show the values of model attributes over time. The contents of these graphs are real time; the data is only available for the current OneClick client session.
<interface-table-config>	<contents-registry>	Specifies the XML that configures the table on the Interfaces tab.

Define Specific OneClick Device Support Based on Model Type

As mentioned earlier in this chapter, the contents registry defines the general OneClick device support. If you want to define specific device support on a per model type basis in the OneClick client, you should define a component details registry (denoted by the <component-details-registry> element). The component details registry defines the contents for the views within the Component Detail panel.

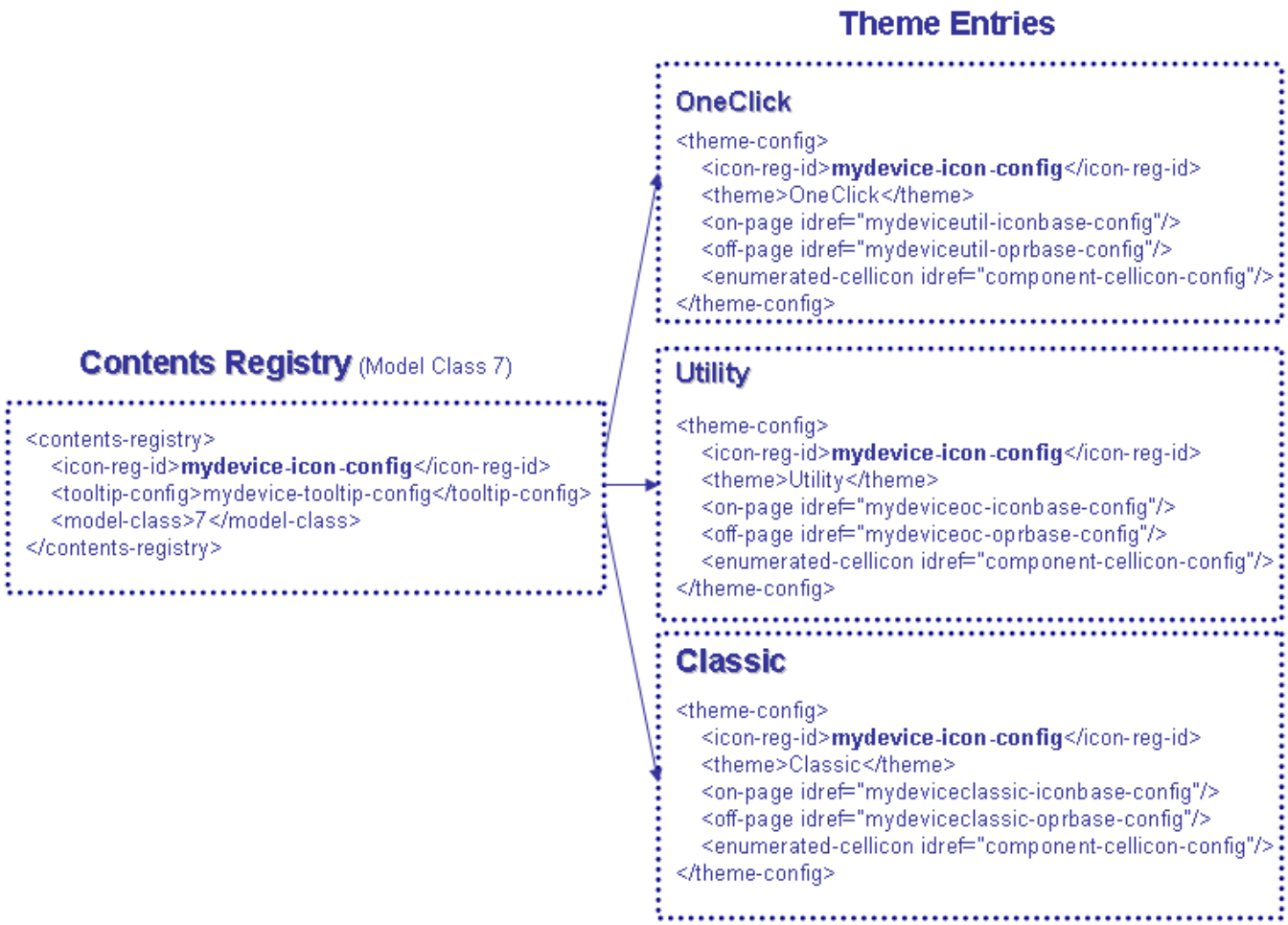
Element	Parent Element	Description
<app-config>	Not applicable	The root element for the custom-app-config.xml file.

<component-details registry>	<app-config>	Used to associate a single or group of model classes and/or model types to configuration files that define the content in the Component Detail panel. The same child elements of <component-details-registry> are supported within <contents-registry>. However, since <component-details-registry> is an extension of <contents-registry>, all elements that are not defined are inherited from <contents-registry>. If both are registered for the same model type or model class, <component-details-registry> takes precedence over <contents-registry>.
<model-class>	<contents-registry>	Registers a model class for this component details registry.
<model-type>	<contents-registry>	Registers a model type for this component details registry.
<is-derived-from>	<contents-registry>	Registers a parent model type and all of its derived model types for this contents registry.
<information-config>	<contents-registry>	Specifies the XML that defines the content of the model's Information tab.
<performance-config>	<contents-registry>	Specifies the XML that defines the contents of the model's Performance tab, where one or more graphs can be defined to show the values of model attributes over time. The contents of these graphs are real time; the data is only available for the current OneClick client session.
<interface-table-config>	<contents-registry>	Specifies the XML that configures the table on the Interfaces tab.

Define Model Appearance

You can define the model appearance as it will look within the OneClick client through XML configuration (for example, topology, Component Detail header, navigation hierarchy, and so on). This is accomplished using the contents registry on a per model class and/or model type basis.

Within the contents registry entry, you need to specify an icon registration ID (<icon-reg-id>). As shown in the following figure, this ID maps to theme configuration entries (<theme-config>) that define the XML files used to construct the model appearance for a specific theme. By default, OneClick includes three icon themes: OneClick, Utility, and Classic.



In the preceding XML example, the contents registry for model class 7 specifies the <icon-reg-id> as mydevice-icon-config. For this ID, a theme configuration is created for each of the three themes included with OneClick: OneClick, Utility, and Classic. Within the theme configurations, the XML files that construct the icons are specified.

The following procedure outlines the process to define a model's appearance in the OneClick user interface using each of these elements.

To define a model's appearance in the OneClick user interface

1. If it does not already exist, create a file named <*$SPECROOT*>/custom/console/config/custom-app-config.xml by copying <*$SPECROOT*>/tomcat/webapps/spectrum/WEB-INF/console/config/custom-app-config.xml.
   If the file already exists, it already contains customized changes to the factory default file, and you should make your additional changes in it.
2. Open the file with a text editor.
3. Create a contents registry entry for the applicable model types and/or model classes.

```
<contents-registry>
    ...
        <model-type>0xffff0000</model-type>
        <model-class>2</model-class>
    ...
</contents-registry>
```

4. Within the contents registry, specify an icon registration ID. This ID will map to theme configuration entries.
5. For each desired theme, create a <theme-config> entry. These entries will be identified using the registration ID from the previous step.
6. Design the appearance of the icon for each theme as described in Design On-Page and Off-Page Reference Icons.
7. Define the tooltips for the icons as described in Define Model Icon Tooltips.
8. Once you have created support for a model's appearance, you may want to customize the views and information available in the model's Component Detail panel.
9. Save and close the <*$SPECROOT*>/custom/console/config/custom-app-config.xml file.
10. Restart the OneClick client for your changes to take effect.

Configure Icons for OneClick Themes

By default, OneClick includes three icon themes:

- OneClick (default theme)
- Utility
- Classic

The default OneClick theme is the most robust and, therefore, is the recommended theme for use.

The elements that you can use to create a theme configuration are described in the following table:

Element	Parent Element	Description
<theme-config>	<app-config>	Defines the theme configuration for a specific icon registration ID.
<icon-reg-id>	<theme-config>	The icon registration ID that maps the theme configuration to the contents registry for one or more model types and/or model classes.
<enumerated-cellicon>	<theme-config>	Specifies that an enumerated cell icon image should be used for the model hierarchy tree. The idref attribute of this element defines the id of the chosen cellicon. Predefined icons exist in the <*$SPECROOT*>/tomcat/webapps/spectrum/WEB-INF/topo/config directory and follow the naming pattern *-color-config.xml.
<dynamic-cellicon>	<theme-config>	Specifies that a dynamic cell icon image should be used for the model hierarchy tree. The idref attribute of this element defines the id of the chosen cellicon. Predefined icons exist in the <*$SPECROOT*>/tomcat/webapps/spectrum/WEB-INF/console/topo/config directory and follow the naming pattern *-color-config.xml.
<static-cellicon>	<theme-config>	Specifies that a static cell icon image should be used for the model hierarchy tree. The idref attribute of this element defines the id of the chosen cellicon. Predefined icons exist in the <*$SPECROOT*>/tomcat/webapps/spectrum/WEB-INF/console/topo/config directory and follow the naming pattern *-color-config.xml.
<theme>	<theme-config>	The name of the theme to which this configuration is applied. OneClick, Classic, or Utility should be specified to configure the existing themes. Specifying a new name generates a new theme.
<on-page>	<theme-config>	The icon configuration for the standard icon for the designated theme.
<off-page>	<theme-config>	The icon configuration for the referenced icon for the designated theme.

The <on-page> and <off-page> elements specify the XML files used to construct the standard version and off-page reference version of the icon as they would appear within the Topology.

The image used within the Navigation panel hierarchy is defined by one of the following three <*-cellicon> elements:

- <enumerate-cellicon>
- <dynamic-cellicon>
- <static-cellicon>

Using the <theme-config> Element to Create Icon Appearance

The following example shows how the theme configuration for the icon registration ID mydevice-icon-config is defined using the <theme-config> element. Notice that the icon's appearance is defined for each of the OneClick themes: Utility, Classic, and OneClick.

Example: <theme-config> Code

```
<theme-config>
   <icon-reg-id>mydevice-icon-config</icon-reg-id>
   <theme>Utility</theme>
   <on-page idref="mydeviceutil-iconbase-config"/>
   <off-page idref="mydeviceutil-oprbase-config"/>
   <enumerated-cellicon idref="component-cellicon-config"/>
</theme-config>
<theme-config>
   <icon-reg-id>mydevice-icon-config</icon-reg-id>
   <theme>Classic</theme>
   <on-page idref="mydeviceclassic-iconbase-config"/>
   <off-page idref="mydeviceclassic-oprbase-config"/>
   <enumerated-cellicon idref="component-cellicon-config"/>
</theme-config>
<theme-config>
    <icon-reg-id>mydevice-icon-config</icon-reg-id>
    <theme>OneClick</theme>
    <on-page idref="mydeviceoc-iconbase-config"/>
    <off-page idref="mydeviceoc-oprbase-config"/>
    <enumerated-cellicon idref="component-cellicon-config"/>
</theme-config>
```

Design On-Page and Off-Page Reference Icons

A model can have two different appearances within the OneClick topology:

- On-page
- Off-page

The on-page representation is the standard appearance of the model and, therefore, the more important appearance. The off-page representation is used when the model is drawn in a topology as a reference to a model that exists in a different topological view. That is, when the model is connected to another model that resides in a different topology.

If the model supports connections (links) within the OneClick topology, you should specify an off-page reference icon (image) in addition to an on-page version.

You specify the on-page and off-page icons within the theme configuration using the <on-page> and <off-page> elements. These elements specify the XML that constructs the corresponding model icon.

The following table defines the elements that can be used within the XML files referenced by the <on-page> and <off-page> elements.

Element	Parent Element	Description
<icon-config>	Not applicable	This is the root element for the icon configuration file.

<static-cellicon>	<icon-config>	A single color foreground/background or image. You can use the idref attribute to refer to another file that defines the image or color. Predefined icons exist in the <*$SPECROOT*>/tomcat/webapps/spectrum/WEB-INF/console/topo/config directory and follow the naming pattern *-color-config.xml.
<dynamic-cellicon>	<icon-config>	More than one color or image. You can use the idref attribute to refer to another file that defines the image or color. Predefined icons exist in the <*$SPECROOT*>/tomcat/webapps/spectrum/WEB-INF/console/topo/config directory and follow the naming pattern *-color-config.xml.
<enumerated-cellicon>	<icon-config>	Displays an image or color based on the value of an attribute. You can use the idref attribute to refer to another file that defines the image or color. Predefined icons exist in the <*$SPECROOT*>/tomcat/webapps/spectrum/WEB-INF/console/topo/config directory and follow the naming pattern *-color-config.xml.
<shape>	<icon-config>	See Define the Icon Shape.
<stroke>	<icon-config>	The type of line used to create the shape. The following values can be used: - Bump1: Bump stroke of width 1. - Bump2: Bump stroke of width 2. - Dash1x1x1: Dashed Stroke of width of 1, and repeats dash of length 1 followed by a blank of length 1. - Dash1x1x3: Dashed Stroke of width of 1, and repeats dash of length 1 followed by a blank of length 3. - Dash1x2x2: Dashed Stroke of width of 1, and repeats dash of length 2 followed by a blank of length 2. - Dash1x3x1: Dashed Stroke of width of 1, and repeats dash of length 3 followed by a blank of length 1. - Dash2x1x1: Dashed Stroke of width of 2, and repeats dash of length 1 followed by a blank of length 1. - Dash2x1x3: Dashed Stroke of width of 2, and repeats dash of length 1 followed by a blank of length 3. - Dash2x2x2: Dashed Stroke of width of 2, and repeats dash of length 2 followed by a blank of length 2. - Dash2x3x1: Dashed Stroke of width of 2, and repeats dash of length 3 followed by a blank of length 1. - DashDot: Dashed Stroke of width of 1, and repeats dash of length 3 followed by a blank of length 1 followed by a dot of length 1 followed by a blank of length 1. - DashDotDot: Dashed Stroke of width of 1, and repeats dash of length 3 followed by a blank of length 1 followed by a dot of length 1 followed by a blank of length 1 followed by a dot of length 1 followed by a blank of length 1. - DashedSeparator: Dashed Stroke used for display a separator lines for the DX NetOps Spectrum look and feel. - PenStroke Ditch1: Ditch stroke with a width of 1. - Ditch2: Ditch stroke with a width of 2. - Ditch4: Ditch stroke with a width of 4. - Hole1: Hole stroke of width 1. - Hole2: Hole stroke of width 2. - Invisible: Invisible Pen Stroke. - Ridge2: Ridge stroke with a width of 2. - Ridge4: Ridge stroke with a width of 4. - Solid1: Solid Pen Stroke with a width of 1. - Solid2: Solid Pen Stroke with a width of 2. - Solid3: Solid Pen Stroke with a width of 3. - StepBump1: Step Bump with a width of 1. - StepBump2: Step Bump with a width of 2. - StepHole2: Step Hole with a width of 2.
<pipe-connection>	<icon-config>	See Define Pipe Connection Location.

<components>	<icon-config>	This element encloses the image, label, and text components that define the icon. The different components are layered on top of each other, and the index value for each of these components determines the drawing order. The lowest number (1) will be drawn first. See Define Image Components, Define Text Components, and Define Selection Components.

Use <on-page> and <off-page> Elements

If you create an icon configuration file called mydevice-utility-iconbase-config.xml that defines the on-page reference for your icon in the Utility theme, you must define the <on-page> reference element in the appropriate theme configuration as in the following example.

Example: <on-page> and <off-page> Code

```
<theme-config >
   <icon-reg-id>mydevice-icon-config</icon-reg-id>
   <theme>Utility</theme>
   <on-page idref="mydevice-utility-iconbase-config"/>
   <off-page idref="mydevice-utility-oprbase-config"/>
   <enumerated-cellicon idref="component-cellicon-config"/>
</theme-config>
```

Example: Icon Configuration File

This example shows an icon configuration file using the XML elements described in the table in Design On-Page and Off-Page Reference Icons.

```
<?xml version="1.0" encoding="UTF-8"?>
<icon-config id="mydevice-utility-iconbase-config">
   <static-color idref="default-iconbase-color-config"/>
   <shape-rectangle >
    <x>0</x>
    <y>0</y>
    <width>139</width>
    <height>84</height>
   </shape-rectangle>
   <stroke>Invisible</stroke>
<!-- ============================================================
- Specify the location of where pipes will connect to the icon.-
 ============================================================ -->
   <pipe-connection>
    <x>73</x>
    <y>36</y>
   </pipe-connection>
   <components>
<!-- ============================================================
- Definition of the model's base image. The color of this
- image is determined by the condition of the model.-
 ============================================================ -->
    <image-component index="2">
        <x>0</x>
        <y>0</y>
        <width>139</width>
        <height>84</height>
        <image idref="oneclick-orgservice-iconbase-image-config"/>
```

```
    </image-component>
   </components>
</icon-config>
```

Define the Icon Shape

The base shape of the icon defines the area for the icon that the user clicks on to activate the model. The base shapes for a OneClick icon are the following:

- Rectangle
- Rounded Rectangle
- Ellipse
- Polygon
- Line

Define each of these shapes using the <icon-config> element.

Rectangle

Elements used to specify a rectangle in OneClick is defined in the following table.

Element	Parent Element	Description
<shape-rectangle>	<icon-config>	Defines a rectangle.
<x>	<shape-rectangle>	The upper left X coordinate of the rectangle.
<y>	<shape-rectangle>	The upper left Y coordinate of the rectangle.
<width>	<shape-rectangle>	The width of the rectangle.
<height>	<shape-rectangle>	The height of the rectangle.

Example: Rectangle Code

```
<shape-rectangle>
   <x>26</x>
   <y>24</y>
   <width>45</width>
   <height>14</height>
</shape-rectangle>
```

Rounded Rectangle

Elements used to specify a rounded rectangle in OneClick are defined in the following table.

Element	Parent Element	Description
<shape-roundrectangle>	<icon-config>	Defines a rounded rectangle.
<x>	<shape-roundrectangle>	The upper left X coordinate.
<y>	<shape-roundrectangle>	The upper left Y coordinate.
<width>	<shape-roundrectangle>	The width.
<height>	<shape-roundrectangle>	The height.
<arcwidth>	<shape-roundrectangle>	The width of the arc that defines the corners.
<archeight>	<shape-roundrectangle>	The height of the arc that defines the corners.

Example: Rounded Rectangle Code

```
<shape-roundrectangle>
   <x>0</x>
   <y>0</y>
   <width>89</width>
   <height>68</height>
   <arcwidth>12</arcwidth>
   <archeight>12</archeight>
</shape-roundrectangle>
```

Ellipse

Elements used to specify an ellipse in OneClick are defined in the following table.

Element	Parent Element	Description
<shape-ellipse>	<icon-config>	Creates an ellipse.
<x>	<shape-ellipse>	The upper left X coordinate of the ellipse.
<y>	<shape-ellipse>	The upper left Y coordinate of the ellipse.
<width>	<shape-ellipse>	The width of the ellipse.
<height>	<shape-ellipse>	The height of the ellipse.

Example: Ellipse Code

```
<shape-ellipse>
   <x>26</x>
   <y>24</y>
   <width>45</width>
   <height>14</height>
</shape-ellipse>
```

Polygon

Elements used to specify a polygon in OneClick are defined in the following table.

Element	Parent Element	Description
<shape-polygon>	<icon-config>	Defines a polygon.
<point>	<shape-polygon>	A point defining an edge of the polygon. The x attribute of this element defines the x coordinate position of the point. The y attribute of this element defines the y coordinate position of the point.

Example: Polygon Code

```
<shape-polygon>
   <point x="0" y="28" />
   <point x="10" y="10" />
   <point x="28" y="0" />
   <point x="116" y="0" />
   <point x="134" y="10" />
   <point x="144" y="28" />
   <point x="144" y="65" />
   <point x="134" y="80" />
```

```
    <point x="116" y="92" />
    <point x="28" y="92" />
    <point x="10" y="80" />
    <point x="0" y="65" />
</shape-polygon>
```

Line

Elements used to specify a line in OneClick are defined in the following table.

Element	Parent Element	Description
<shape-line>	<icon-config>	Defines a line.
<x1>	<shape-line>	X coordinate for the start of the line.
<y1>	<shape-line>	Y coordinate for the start of the line.
<x2>	<shape-line>	X coordinate for the end of the line.
<y2>	<shape-line>	Y coordinate for the end of the line.

Example: Line Code

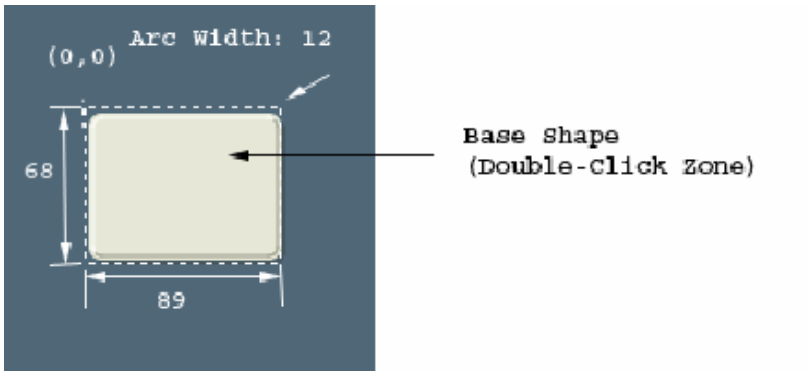
```
<shape-line>
   <x1>5</x1>
   <y1>5</y1>
   <x2>40</x2>
   <y2>40</y2>
</shape-line>
```

Create an Icon Shape

The icon shape defines the area of the icon the user can click in and select the device. The following example defines the rounded-rectangle icon shape shown in the image that follows the example.

Example: Icon Shape Code

```
<icon-config id="device-iconbase-config">
   <static-color idref="device-iconbase-color-config"/>
   <shape-roundrectangle >
    <x>0</x>
    <y>0</y>
    <width>89</width>
    <height>68</height>
    <arcwidth>12</arcwidth>
    <archeight>12</archeight>
   </shape-roundrectangle>
```



X and Y Coordinates

The x and y coordinates define the upper left-hand corner of the image. As the value of x increases, the upper left-hand corner of the image moves from the left to the right. As the value of y increases, the upper left-hand corner of the image moves from the top to the bottom. The image in the preceding section shows the upper left corner of the icon shape at 0,0, defined in Example: Icon Shape Code.

Define Pipe Connection Location

The pipe location defines where pipes get connected to the icon. Elements used to define pipe connection locations are listed in the following table.

Element	Parent Element	Description
<pipe-connection>	<icon-config>	Defines the pipe location on an icon.
<x>	<pipe-connection>	The x coordinate for the pipe location.
<y>	<pipe-connection>	The y coordinate for the pipe location.

Example: <pipe-connect> Code

```
<pipe-connection>
   <x>73</x>
   <y>36</y>
</pipe-connection>
```

Define Image Components

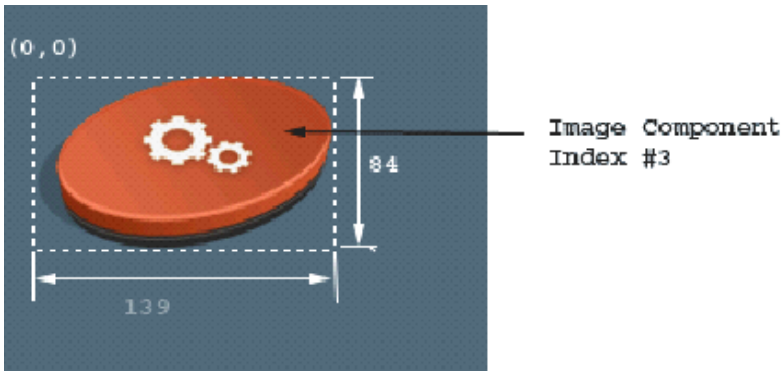
Image components enable you to add static or dynamic images to a model based on the model attributes. Image components are defined in the table after the following example.

Example: <image-component> Code

The following example defines an image component in an icon configuration file:

```
<image-component index="1">
   <x>0</x>
   <y>0</y>
   <width>139</width>
   <height>84</height>
   <image idref="oneclick-orgservice-iconbase-image-config"/>
</image-component>
```

The <image> element references the image definition file shown in Example: Image Definition File. This generates the image component shown in the following figure, showing the model in the CRITICAL (3) state.



The following table describes the elements used for defining image components.

Element	Parent Element	Description
<components>	<icon-config>	Defines all components for the icon.
<image-component>	<components>	Defines the image component. The index attribute of this element defines the order that the image is drawn relative to the other image, text, and selection components defined for this icon. Each index value must be unique, for example you cannot have two <*-component> elements with an index value of 1 in the same file. If you do, only the first <*-component> element is used. Index values must begin at 1.
<x>	<image-component>	The x coordinate of the upper left corner of image.
<y>	<image-component>	The y coordinate of the upper left corner of image.
<width>	<image-component>	The width of the image in pixels. There are no size restrictions on the image, however, it is recommended that you use a size relative to the other images used in OneClick.
<height>	<image-component>	The height of the image in pixels. There are no size restrictions on the image, however, it is recommended that you use a size relative to the other images used in OneClick.
<image>	<image-component>	The image to be rendered. The idref attribute references the XML file that builds the image. It is put in a separate file for organizational purposes only. Images should be in png file format and should be placed in the <*$SPECROOT*>/tomcat/webapps/spectrum/images directory.
<shape-*>	<image-component>	Shape component to be drawn with image.
<enumerated-color>	<image-component>	Colors used for the shape.
<static-color>	<image-component>	Color used for the shape.
<selection-component>	<image-component>	Indicates whether this is to be shown only when the user selects the image component.

Example: Image Definition File

The following XML code defines the image used for <image-component index =“2”> in Example: Icon Configuration File. The example uses the <select>, <case>, and <expression> elements to conditionally select an image based on the value of the condition attribute. The example uses the <yield> to define which image should be used when a condition is met. Note that the image is in PNG file format. Images used for customization purposes should be stored in the <

$SPECROOT>/tomcat/webapps/spectrum/images directory. Express the path to an image placed in this directory from the images directory, as images/myimage.png.

```
<?xml version="1.0" encoding="UTF-8"?>
<image id="oneclick-orgservice-iconbase-image-config">
   <select>
    <case>
        <expression>
            attrInt(AttributeID.CONDITION) == 0
            <!-- GREEN_CONDITION -->
        </expression>
        <yield>
            images/oneclick_org_service_green.png
        </yield>
    </case>
    <case>
        <expression>
            attrInt(AttributeID.CONDITION) == 1
        </expression>
        <yield>
            images/oneclick_org_service_yellow.png
        </yield>
    </case>
    <case>
        <expression>
            attrInt(AttributeID.CONDITION) == 2
        </expression>
        <yield>
            images/oneclick_org_service_orange.png
        </yield>
    </case>
    <case>
        <expression>
            attrInt(AttributeID.CONDITION) == 3
        </expression>
        <yield>
            images/oneclick_org_service_red.png
        </yield>
    </case>
    <case>
        <expression>
            attrInt(AttributeID.CONDITION) == 4
        </expression>
        <yield>
            images/oneclick_org_service_brown.png
        </yield>
    </case>
    <case>
        <expression>
            attrInt(AttributeID.CONDITION) == 5
        </expression>
        <yield>
            images/oneclick_org_service_grey.png
```

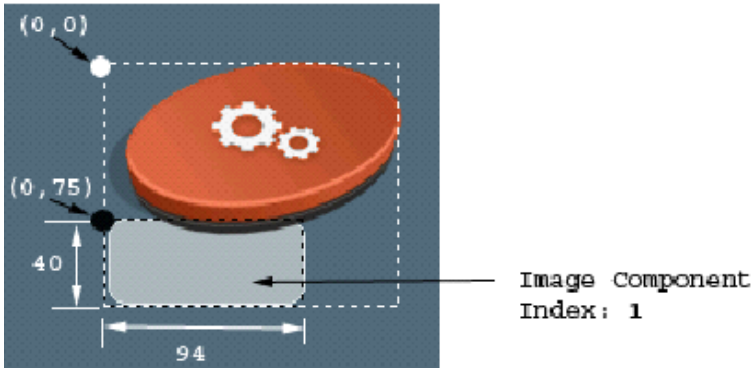
```
        </yield>
    </case>
    <case>
        <expression>
            attrInt(AttributeID.CONDITION) == 6
        </expression>
        <yield>
            images/oneclick_org_service_blue.png
        </yield>
    </case>
        <default>
            images/oneclick_org_service_blue.png
        </default>
       </select>
</image>
```

Example: Icon Configuration File

This icon configuration example generates the image that follows it. The example assumes that the condition of the model is CRITICAL (3).

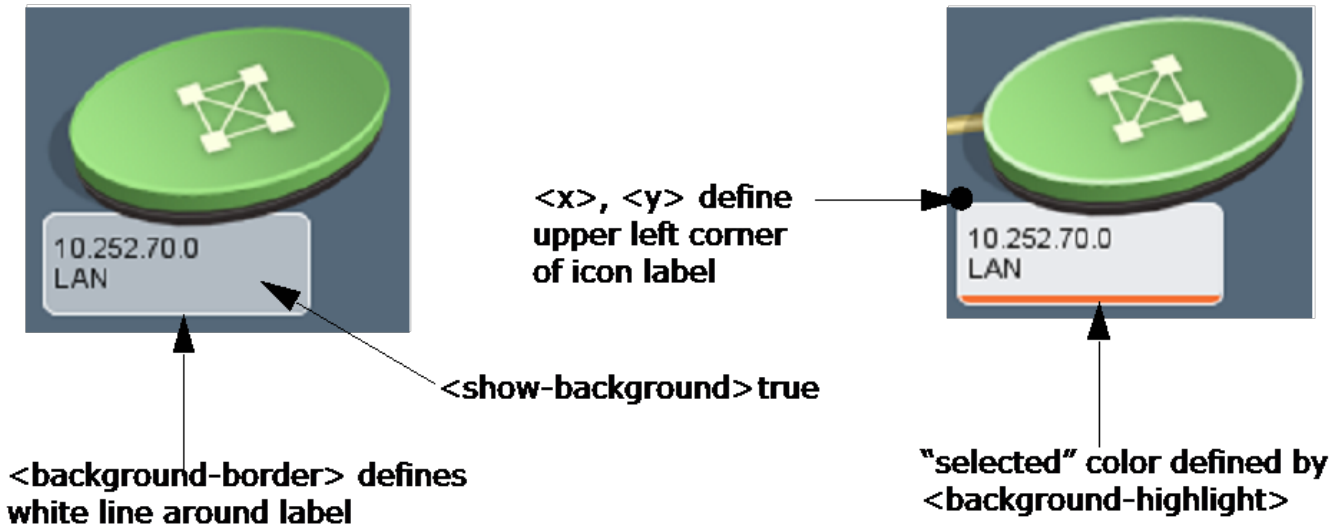
```
<?xml version="1.0" encoding="UTF-8"?>
<icon-config id="oneclick-orgservice-iconbase-config">
   <static-color idref="oneclick-default-iconbase-color-config"/>
   <shape-rectangle >
    <x>0</x>
    <y>0</y>
    <width>139</width>
    <height>84</height>
   </shape-rectangle>
   <stroke>Invisible</stroke>
<!-- ============================================================
Specify the location of where pipes will connect to the icon.
============================================================-->
   <pipe-connection>
    <x>73</x>
    <y>36</y>
   </pipe-connection>
   <components>
<!-- ============================================================
Definition of the model label.
============================================================ -->
    <label-component idref="default-iconlabel-config" index="1">
        <x>0</x>
        <y>77</y>
        <column-list>
            <field-column>
                <column idref="column-modelname-config"/>
                <column idref="column-devicetype-config"/>
            </field-column>
        </column-list>
    </label-component>
<!-- ============================================================
Definition of the model's base image. Image color is determined by model condition.
```

```
========================================================== -->
    <image-component index="2">
        <x>0</x>
        <y>0</y>
        <width>139</width>
        <height>84</height>
        <image idref="oneclick-orgservice-iconbase-image-config"/>
    </image-component>
   </components>
</icon-config>
```



Create an Icon Label

OneClick model or device type icons have labels to identify them to OneClick operators. Use the <label-component> elements listed in the table in the default-iconlabel-config.xml file to add labels to the icons you create. The following figure shows how some of the elements can be used in defining an icon label.



The default-iconlabel-config.xml File

The file *<$SPECROOT>*/SPECTRUM/tomcat/webapp/spectrum/WEB-INF/topo/config/default-iconlabel-config.xml contains examples and more information on using the <label-component> element and its attributes to create icon labels.

Example: <label-component> Code

```
<!-- ================================================================
Definition of the model label.
=================================================================-->
<label-component idref="default-iconlabel-config" index="1">
   <x>0</x>
   <y>67</y>
   <column-list>
    <field-column>
        <column idref="column-modelname-config"/>
        <column idref="column-devicetype-config"/>
    </field-column>
   </column-list>
</label-component>
```

This example defines the model label by extending the functionality of the default-iconlabel-config.xml file. This example creates a label that displays two fields of text defined in the two column statements. The column statements create two rows in the label for the content defined by the column configuration files they reference. This label displays the model name and device type in the icon label. The code does not specify a minimum or maximum column width for the label (see the following table), so it has a fixed width of 95 pixels, the default condition.

Element	Parent Element	Description
<components>	<icon-config>	Defines all components for the icon.
<label-component>	<component>	Defines a label component. The index attribute defines the order that the label is drawn in with respect to other image, text, and label components defined for the same icon. Each index value must be unique. If you have two <*-components> with the same index value - only the second one is drawn. Index values begin at 1.
<x>	<label-component>	Defines the x coordinate of the upper left corner of the label relative to the icon image component.
<y>	<label-component>	Defines the y coordinate of the upper left corner of the label relative to the icon image component.
<column-list>	<label-component>	Constructs a list of columns used to create the labels. Only one column is supported.
<field-column>	<column-list>	Constructs a column of information
<column>	<field-column>	Defines the data for the <field-column>. The idref attribute allows you to associate another XML file with the <column> that defines the data for the column. The data for the <column> does not have to reside in another file.
<max-background-width>	<label-component>	Defines the maximum width of the label background. The label background expands and contracts in width based on the longest column value.
<min-background-width>	<label-component>	Defines the minimum width of the label background.
<default-transparency>	<label-component>	Defines transparency value for label background when the icon is not selected. 0 - 255; 0=completely transparent, 255=completely opaque.
<selected-transparency>	<label-component>	Defines transparency value for label background when the icon is selected. 0 - 255; 0=completely transparent, 255=completely opaque.

<show-background>	<label-component>	Indicate whether or not to show the label's background.
<enumerated-color>	<label-component>	Defines the label background color. <enumerated-color> uses expressions and enumerations to determine the color.
<static-color>	<label-component>	Defines the label background color. <static-color> uses a specific color.
<vertical-spacing>	<label-component>	Specifies the spacing between rows of text in pixels.
<border-spacing>	<label-component>	Defines the border height above the first column and below the last column in pixels.
<background-border>	<label-component>	True or False. If true, a one-pixel wide line is used to outline the label border.
<enumerated-color>	<background-border>	Defines the color of <background-border>. For details, see <enumerated-color> in this table.
<static-color>	<background-border>	Defines the color of <background-border>.
<background-highlight>	<label-component>	Defines the line that displays at the bottom of the label when the icon is selected.
<enumerated-color>	<background-highlight>	Defines label background color when the icon is selected. For details, see <enumerated-color> in this table.
<static-color>	<background-highlight>	Defines the color of <background-highlight>, the label background that displays when the icon is selected.
<field-value>	<label-component>	Defines the values displayed in the icon label.
<enumerated-color>	<field-value>	Defines the color of text of the icon label.
<static-color>	<field-value>	Defines the color of text of the icon label.
<font>	<field-value>	Defines the font used for the icon label text.

Adjust Icon Label Background Width

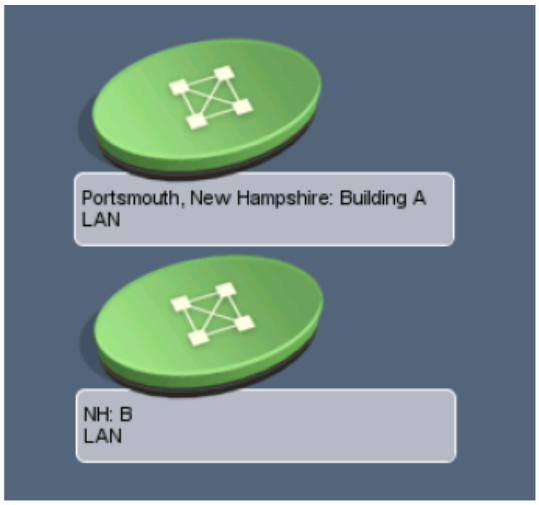
The icon label background widens and narrows according to the length of the longest text entry in the label, up to the maximum width specified in <max-background-width>, and down to the minimum width specified in <min-background-width>. If the label background is not wide enough to accommodate the length of the label text, increase the <max-background-width> value.

Default Label Width Settings

When you create an icon label using label-component, the default label size is fixed at 95 pixels. Both <max-background-width> and <min-background-width> have a default value of 95. This creates a label background with a fixed width of 95. If you do not specify either of these elements, they assume the default value.

Create Fixed Width Icon Labels

To create an icon label that has a fixed width, set <max-background-width> and <min-background-width> to the same value that provides enough line space for the icon label text. The following image shows an icon label with a fixed width of 200.

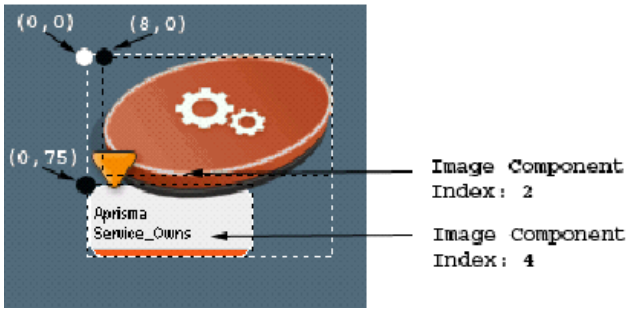


Define Text Components

Refer to versions of this manual for release 8.0 or earlier on the DX NetOps Spectrum support and documentation web site.

Define Selection Components

In order to make the icon standout when selected, you may want to specify images that will appear only during selection. You do this via the <selection-component> element. In the example below there are two components added that are defined as Selection Components (image component #2 and #4). If the user selected the component, these image components would become visible. Otherwise, they remain invisible. The following figure shows the two selection components:




```
<?xml version="1.0" encoding="UTF-8"?>
<icon-config id="oneclick-orgservice-iconbase-config">
<static-color idref="oneclick-default-iconbase-color-config"/>
<shape-rectangle >
   <x>0</x>
   <y>0</y>
   <width>139</width>
   <height>84</height>
</shape-rectangle>
<stroke>Invisible</stroke>
<!-- ============================================================
Specify the location of where pipes will connect to the icon. -
 ============================================================ -->
```

```
<pipe-connection>
   <x>73</x>
   <y>36</y>
</pipe-connection>
<components>
<!-- ============================================================
Definition of the model text background.
============================================================= -->
   <image-component index="1">
    <x>0</x>
    <y>75</y>
    <width>94</width>
    <height>40</height>
    <image>
        <select>
            <default>
                com/aprisma/spectrum/app/topo/images/
                icon_text_background.png
            </default>
        </select>
    </image>
   </image-component>
   <image-component index="2">
    <x>0</x>
    <y>75</y>
    <width>94</width>
    <height>40</height>
    <selection-component>true</selection-component>
    <image>
        <select>
            <default>
                com/aprisma/spectrum/app/topo/images
                icon_selected_text_background.png
            </default>
        </select>
    </image>
   </image-component>
<!-- ============================================================
Definition of the model's base image. The color of this image is
determined by the condition of the model.
============================================================= -->
   <image-component index="3">
    <x>0</x>
    <y>0</y>
    <width>139</width>
    <height>84</height>
    <image idref="oneclick-orgservice-iconbase-image-config"/>
   </image-component>
<!-- ============================================================
Definition of the image to show when the model is selected.
============================================================= -->
   <image-component index="4">
    <x>8</x>
```

```
    <y>0</y>
    <width>131</width>
    <height>72</height>
    <selection-component>true</selection-component>
    <image>
        <select>
            <default>
                com/aprisma/spectrum/app/topo/images/
                oneclick_selected_container.png
            </default>
        </select>
    </image>
   </image-component>
<!--=================================================================
Definition of the model name text field.
================================================================= -->
   <text-component idref="oneclick-default-textfield-config"index="5">
    <x>5</x>
    <y>97</y>
    <width>85</width>
    <height>13</height>
    <horizontal_alignment>left</horizontal_alignment>
    <text>
        <attribute>0x1006e</attribute>
    </text>
   </text-component>
<!-- ================================================================
Definition of the model type name text.
================================================================= -->
   <text-component idref="oneclick-default-textfield-config" index="6">
    <x>5</x>
    <width>85</width>
    <height>12</height>
    <horizontal_alignment>left</horizontal_alignment>
    <text>
        <attribute>AttributeID.DEVICE_TYPE</attribute>
        <expression>
            ( value() == null || ((String)value()).length() ==
            0 ) ? attr(AttributeID.MTYPE_NAME ) : value()
        </expression>
    </text>
   </text-component>
<!-- ================================================================
Definition of the rollup condition symbol.
================================================================= -->
       <image-component index="7">
        <x>0</x>
        <y>54</y>
        <width>19</width>
        <height>19</height>
        <image idref="oneclick-rollup-triangle-image-config"/>
       </image-component>
</components>
```

```
</icon-config>
```

Define Model Icon Tooltips

You can configure the content of a tooltip that displays when a OneClick user moves the cursor over the model icon. In the contents registry of the custom-app-config.xml file, the <tooltip-config> element specifies the file that defines the tooltip. The custom tooltip file, mydevice-tooltip-config.xml must be placed into the *$SPECROOT*/custom/topo/config folder.

A tooltip configuration for models of model class 2, 5, 11, and 12 is registered to use the tooltip that is defined within the mydevice-tooltip-config.xml file. Verify the following example:

```
<contents-registry>
   <reg-id>device-icon-config</reg-id>
   <tooltip-config>mydevice-tooltip-config</tooltip-config>
   <model-class>2</model-class>
   <model-class>5</model-class>
   <model-class>11</model-class>
   <model-class>12</model-class>
</contents-registry>
```

The following example shows the contents of a tooltip file. Each element is explained in the table with examples.

```
<?xml version="1.0" encoding="UTF-8"?>
<tooltip-config id="mydevice-tooltip-config"
xmlns ="http://www.aprisma.com"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aprisma.com../../common/schema/column-config.xsd">
<format><![CDATA[
<html><table>
   <tr>
    <td><b>{0}</b></td>
    <td>{1}</td>
   </tr>
   <tr>
    <td><b>{2}</b></td>
    <td>{3}</td>
   </tr>
   <tr>
    <td><b>{4}</b></td>
    <td>{5}</td>
   </tr>
</table></html>
</format>
<param>
   <localize>com.aprisma.spectrum.app.util.render.ModelNameColumn</localize>
</param>
<param>
   <attribute>AttributeID.MODEL_NAME</attribute>
   <renderer>com.aprisma.spectrum.app.util.render.NullRenderer
   </renderer>
</param>
<param>
   <localize>
   com.aprisma.spectrum.app.util.render.NetworkAddressColumn
   </localize>
```

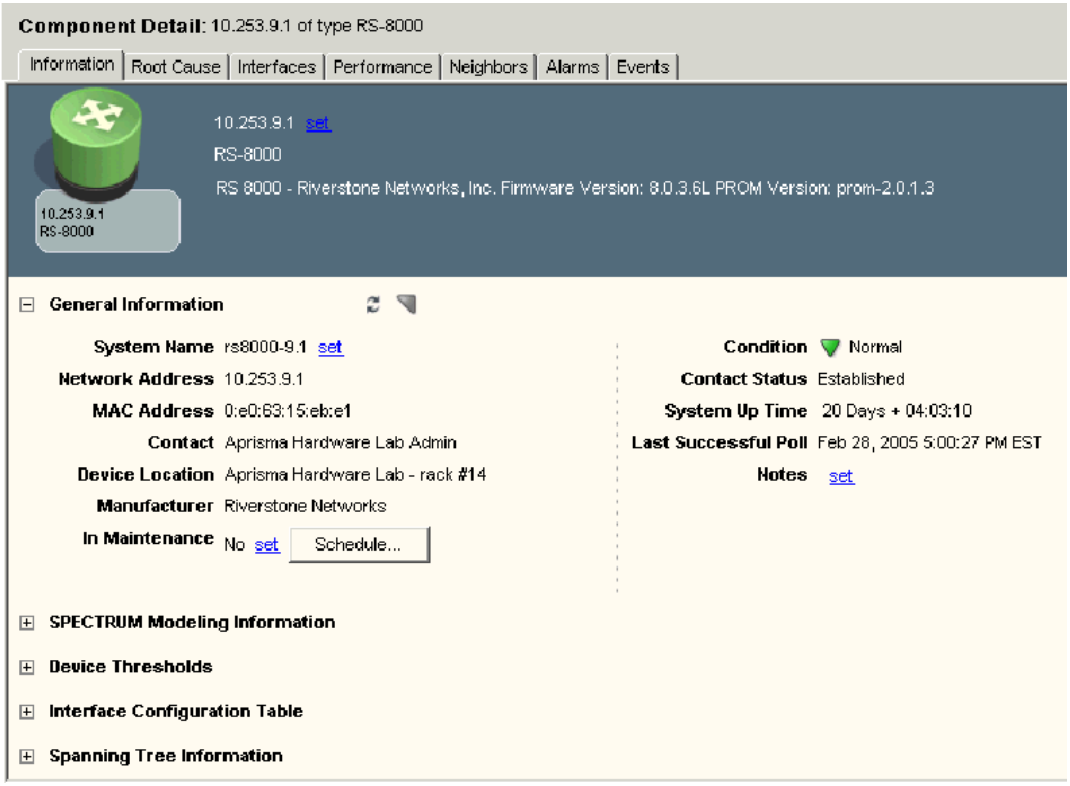
```
</param>
<param>
   <attribute>AttributeID.NETWORK_ADDRESS</attribute>
   <renderer>com.aprisma.spectrum.app.util.render.NullRenderer
   </renderer>
</param>
<param>
   <localize>
   com.aprisma.spectrum.app.util.render.MACAddressColumn
   </localize>
</param>
<param>
   <attribute>AttributeID.MAC_ADDRESS</attribute>
   <renderer>com.aprisma.spectrum.app.util.render.NullRenderer
   </renderer>
</param>
</device-tooltip-config>
```

NOTE
The numbers that are used in the curly brackets reference the parameters that are defined by the following <param> elements. {0} references the first parameter, {1} references the second parameter and so on.

Element	Parent Element	Description
<tooltip-config>	Not applicable	The root element for the file that defines the tooltip. The id attribute for this element must be set equal to the value used for the <tooltip-config> element in the <content-registry> found in the custom-app-config.xml file.
DO NOT USE	<tooltip-config>	Use this to define how the data will be displayed in the tooltip. In the above example, an HTML table is used. The number in the curly brackets, e.g. {3}, references the corresponding parameter, for example, the third parameter defined in the file.
<param>	<tooltip-config>	Use this to define the value to be displayed.
<localize>	<param>	Converts the string specified in the parameter to a localized value. Use this if you are using a parameter value obtained from a OneClick XML file that shipped with DX NetOps Spectrum and begins with “com.aprisma.spectrum”.
<renderer>	<param>	See Customize the Port Name Column of the Interface Table.
<attribute>	<param>	Use this to identify the attribute you want to be displayed.
<message>	<param>	Use this for specifying a plain text value for a parameter.

Customizing a Model's Information View

Each model displayed in OneClick has an Information view as shown in the following figure. You access this view using the Information tab in the Component Detail panel.



Information views are constructed from separate XML files called Information Configuration files. The primary file is the <*$SPECROOT*>/tomcat/webapps/spectrum/WEB-INF/topo/config/topo-app-config.xml file. In this file, for each model type, the <contents-registry> element specifies the <information-config> elements that link an Information Configuration file to the model type.

The <contents-registry> in the following example is found in topo-app-config.xml. It links the 0x2100c (Rtr_Cisco) model type with view-devicedetails-config.xml, which specifies the format for the Information view.

```
<contents-registry>
   <reg-id>router-icon-config</reg-id>
   <tooltip-config>device-tooltip-config</tooltip-config>
   <information-config>view-devicedetails-config</information-config>
   <performance-config>performance-data-rtrcisco-config</performance-config>
   <!-- All Model Types derived from Rtr_Cisco (0x21000c) -->
   <model-type>0x21000c</model-type>
   <!-- Rtr_Cisco -->
</contents-registry
```

NOTE
Several model classes and model types may be specified within the contents or component details registries. Information views can be reused in one or more <contents-registry> elements.

This section describes how to add and edit information displayed in the Information view for a particular model type or model class.

Extend or Modify an Information View

You can modify or create an Information view to display information available in a device MIB that DX NetOps Spectrum and OneClick do not support by default. You can decide whether to add this parameter to one of the existing subviews in the Information view for that device type, or to create a new subview.

When you create or modify an Information view, you must create a new Information Configuration file in the <*$SPECROOT*>/custom/console/config/ directory. This file must have the same name as the factory default Information Configuration file that you need to modify. You then associate the Information Configuration file with the appropriate model type using the <*$SPECROOT*>/custom/console/config/custom-app.config.xml file.

> **NOTE**
> You must create the file <*$SPECROOT*>/custom/console/config/custom-app-config.xml and add your customization code to it. See OneClick Customization .

Follow these steps:

1. If you are modifying or extending an existing Information view for an existing model type or model class, identify the current Information view configuration file that is used to create the Information view.
   a. Open the <*$SPECROOT*>/tomcat/webapps/spectrum/WEB-INF/topo/config/topo-app-config.xml file and find the <contents-registry> element for the appropriate model type or model class. (For an example, see the XML code example at the start of this chapter.)

      > **NOTE**
      > OneClick uses the hierarchy that model_type definitions override the same definition found in a model_class.

   b. Find the <information-config> element within the <contents-registry> element, and note the name of the Information Configuration file that is described by this element. All of the existing Information Configuration files are located in the <*$SPECROOT*>/tomcat/webapps/spectrum/WEB-INF/topo/config directory.
2. To extend an existing information configuration, take the following steps:
   a. Create a new file in the <*$SPECROOT*>/custom/topo/config directory with the same name as the Information Configuration file determined in the next step. Use idref to extend the existing factory file with the contents of this new file.
   b. Open the file using a text editor and use the XML syntax outlined in Create an Information Configuration File to build the file.
   c. Continue to step 4.
3. To modify an existing information configuration:
   a. Copy the Information configuration file identified in step 1 from <*$SPECROOT*>/tomcat/webapps/spectrum/WEB-INF/topo/config directory into the <*$SPECROOT*>/custom/topo/config directory.
   b. Open the file using a text editor and use the XML syntax outlined in Create an Information Configuration File to modify the file.
   c. Continue to step 4.
4. Save and close the file.
5. Associate the new Information Configuration file with the appropriate model types or model classes. Follow the instructions in Associate an Information Configuration File with a Model Class or Model Type.

Create an Information Configuration File

The XML that defines the Information Configuration is split up into two major sections: the header definition and the subview definition. Each define that portion of the model's information tab as shown in the following figure.

The XML elements used to build the header and subview in the Information view are listed in the following table.

Element	Parent Element	Description
[set the view variable for your book]	Not applicable	This is the root element for the Information Configuration file. The ID attribute for this element defines the value that should be used for the <information-config> element in the custom-app-config.xml file.
<view-header>	[set the view variable for your book]	Defines the header portion of the view.
<subviews>	[set the view variable for your book]	Defines the available subviews.

Define the Header

The Information tab header is identified by the <view-header> element. The header specifies the model's graphical depiction and textual information as shown in the image in Create an Information Configuration File.

Example: Code for Information Tab Header

```
<view-header>
   <show-icon>true</show-icon>
   <show-labels>false</show-labels>
   <field-column>
    <column idref="column-modelname-config"/>
    <column idref="column-modeltype-config"/>
   </field-column>
```

```
</view-header>
```

Element	Parent Element	Description
<view-header>	[set the view variable for your book]	Defines the header portion of the view.
<show-icon>	<view-header>	Indicates whether or not to show the icon. Values: true or false.
<show-labels>	<view-header>	Indicates whether or not to show field labels Values: true or false
<field-column>	<view-header>	Constructs a column of information.
<column>	<field-column>	Defines the data for the field column. The idref attribute enables you to associate another XML file, which will define the data for the column. The data for the column does not have to be in another file, it is done for organizational purposes only.

Define the Subview

The subview section defines one or more subviews that display in the Information tab as shown in the image in Create an Information Configuration File. You can define one or more subviews using the <subviews> element and the child elements shown in the following table. As shown in the Example: Subview Definition, all subview definitions are enclosed within one <subviews> element.

Element	Parent Element	Description
<subviews>	[set the view variable for your book]	Encloses all of the elements which define each type of subview.
<field-subview>	<subviews>	Defines a field subview.
<table-subview>	<subviews>	Defines a table subview.
<application-subview>	<subviews>	Defines an application subview.
<related-model-subview>	<subviews>	Defines a related model subview.
<related-model-table-subview>	<subviews>	Defines a related model table subview.
<subview-group>	<subviews>	Groups subviews together under one subview.

Example: Subview Definition

```
<subviews>
   <field-subview>
   .
   .
   .
   </field-subview>
   <application-subview>
   .
   .
   .
   </application-subview>
   <table-subview>
   .
   .
   .
   </table-subview>
```

```
</subviews>
```

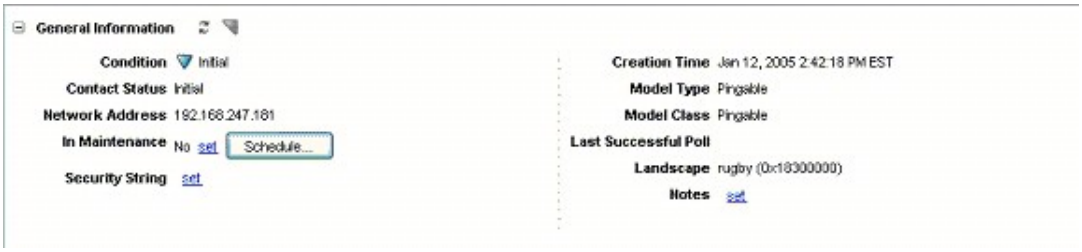
Add a Field Subview

Field subviews are used to display a list of non-list attributes available on the selected device model. The following is an example of XML syntax used to create a field subview.

Example: Field Subview

```
<field-subview>
   <title>General Information</title>
   <privilege>
    <name>GeneralInfo</name>
   </privilege>
   <field-column>
    <column idref="column-condition-config"/>
    <column idref="column-contactstatus-config"/>
    <column idref="column-networkaddress-config"/>
    <column idref="column-ismanaged-config">
        <editable/>
    </column>
    <column idref="column-securitystring-config">
        <editable verifier=
    "com.aprisma.spectrum.app.swing.widget.SecStringInputVerifier"/>
    </column>
   </field-column>
   <field-column>
    <column idref="column-modelcreationdate-config"/>
    <column idref="column-modeltypename-config"/>
    <column idref="column-modelclass-config"/>
    <column idref="column-lastsuccessfulpoll-config"/>
    <column idref="column-landscape-config"/>
    <column idref="column-modelnotes-config">
        <editable/>
    </column>
   </field-column>
</field-subview>
```

This code generates a subview similar to the one shown in the following figure.



You can use the elements shown in the following table to create a field subview.

Element	Parent Element	Description
<field-subview>	<subviews>	Defines a field subview. If you set the expanded attribute of this element to true, the subview will be expanded by default. The idref attribute enables you to associate an XML file that defines the data for this subview. The data for the subview does not have to be in another file, it is done for organizational purposes only.
<title>	<field-subview>	The title of the subview. In our example above, the title is “General Information”.
<display-if>	<field-subview>	Enables the author to specify whether the subview should be displayed via an expression.
<display-if-app-installed>	<field-subview>	Enables the author to specify that the defined view only be added if the specified application is installed.
<privilege>	<field-subview>	Associates a privilege to the subview. If the user is not given this privilege, the subview will not be displayed for that user.
<show-labels>	<field-subview>	Indicates whether or not to show field labels. Values: true or false.
<field-column>	<field-subview>	Constructs a column of information.
<column>	<field-column>	Defines the data for the field column. The idref attribute enables you to associate another XML file, which will define the data for the column. The data for the column does not have to be in another file, it is done for organizational purposes only.
<editable>	<column>	Specifies if the column is editable.

Add Field Subviews Using IDREF

The example in this section shows how to extend the factory default view-devicedetails-config.xml file with a customized field subview using the IDREF attribute. The code shown is in the file *<$SPECROOT>*/custom/topo/config/view-devicedetails-config.xml, the same name as the file it extends, but in the /custom directory.

Example: Field Subview using IDREF

```
<view idref="view-devicedetails-config">
<subviews>
   <field-subview >
    <title>My Subview</title>
    <field-column>
        <column idref="column-networkaddress-config">
        <editable/>
        </column>
    </field-column>
   </field-subview>
</subviews>
</view>
```

This example creates a field subview titled My Subview that displays information defined by the file column-networkaddress-config.xml.

The line <view idref="view-devicedetails-config"> adds the field subview “My Subview” to the factory default view-devicedetails-config.xml file.

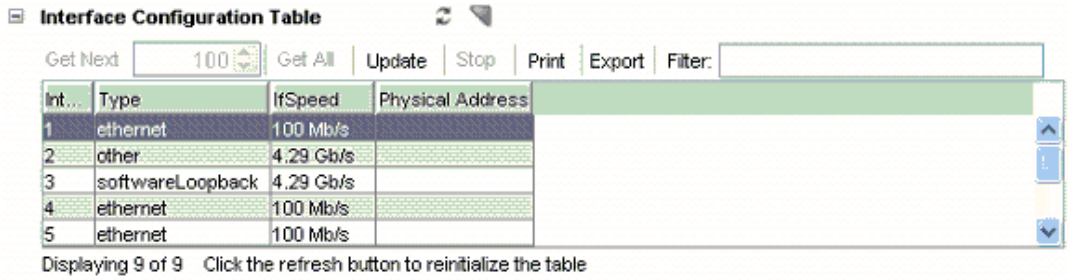
Add a Table Subview

A table subview enables you to display a group of list attributes available from the selected model. These list attributes are displayed in table format. The following is an example of XML syntax used to create a table subview.

Example: Table Subview

```
<table-subview>
   <title>Interface Configuration Table</title>
   <privilege>
    <name>InterfaceConfigurationTable</name>
   </privilege>
   <swing-header-row-template>
    <static-color idref="row-header-color-config"/>
   </swing-header-row-template>
   <swing-row-template>
   </swing-row-template>
   <column-list>
    <column>
        <name>Interface</name>
        <content><attribute>0x100c4</attribute>
        </content>
            <default-width>30</default-width>
    </column>
    <column>
        <name>Type</name>
        <content>
            <attribute>0x100c6</attribute>
            <renderer>
            <param name="attrID">0x100c6</param>
            com.aprisma.spectrum.app.util.render.EnumeratedAttrRenderer
            </renderer>
        </content>
        <default-width>100</default-width>
    </column>
    <column>
        <name>IF Speed</name>
        <content>
            <attribute>0x100c8</attribute> <!-- IfSpeed -->
            <renderer>com.aprisma.spectrum.app.topo.client.
            interfaces.render.IfSpeedRenderer
            </renderer>
        </content>
        <default-width>60</default-width>
    </column>
    <column>
        <name>Physical Address</name>
        <content>
            <attribute>0x100c9</attribute>
        </content>
        <default-width>90</default-width>
    </column>
   </column-list>
</table-subview>
```

The code in this example generates a subview similar to the one shown in the following image.



You can use the elements shown in the following table to create a table subview.

Element	Parent Element	Description
<table-subview>	<subviews>	Adds a table subview. If you set the expanded attribute of this element to true, the subview will be expanded by default. The idref attribute enables you to associate an XML file that defines the data for this subview. The data for the subview does not have to be in another file, it is done for organizational purposes only.
<title>	<table-subview>	The title for the table.
<privilege>	<table-subview>	Associates a privilege to the subview. If the user is not given this privilege, the subview will not be displayed for that user.

NOTE
All of the elements that can be used to modify a table can be used to create the table for the table subview.

Add an Application Subview

An application subview enables you to display attributes affiliated with an application model type related to the selected model type by specified criteria. The example below uses DX NetOps Spectrum's PossPrimApp (0x230000) relation.

Example: Application Subview

```
<application-subview>
   <title>SNMP2 IP Routing Table</title>
   <model-type>0x230010</model-type>
   <subviews>
       <table-subview idref="table-ip2-ip-routingtable-config"/>
   </subviews>
</application-subview>
```

The attribute used within the <model-type> element defines the model type to which the subview pertains. In the example above, the value 0x230010 is used. This attribute value corresponds to the SNMP2_Agent Application model type. This

means that this particular table definition applies only to the SNMP2_Agent application. If the current device does not implement this application, this table will not be visible.

Element	Parent Element	Description
<application-subview>	<subviews>	Creates an application subview. If you set the expanded attribute of this element to true, the subview will be expanded by default. The idref attribute enables you to associate an XML file that defines the data for this subview. The data for the subview does not have to be in another file, it is done for organizational purposes only.
<title>	<application-subview>	The title of the subview.
<model-type>	<application-subview>	The model type to which the subviews will pertain.
<subviews>	<application-subview>	Enables you to add a prebuilt table-subview or field- subview to the detail components that reside within the application subview. Values: table-subview and/or field-subview.
<criteria>	<application-subview>	The search criteria used to find the related models.
<privilege>	<application-subview>	Associates a privilege to the subview. If the user is not given this privilege, the subview will not be displayed for that user.

Add a Related Model Subview

A related model subview enables the user to display the attributes of models related to the current, selected model via a search criteria, for example, models that are related by a specific association. The user can then display attributes of the found models in field or table format. Add a Related Models Table Subview shows how you can display the attributes in table format.

You can use the elements in the following table to create related model subview.

Element	Parent Element	Description
<related-model-subview>	<subviews>	If you set the expanded attribute of this element to true, the subview will be expanded by default. The idref attribute enables you to associate an XML file that defines the data for this subview. The data for the subview does not have to be in another file, it is done for organizational purposes only.
<title>	<related-model-subview>	The title of the subview.
<model-type>	<related-model-subview>	The model type to which the subviews will pertain.
<subviews>	<related-model-subview>	Enables you to add a prebuilt table-subview or field- subview to the detail components that reside within the subview. Values: table-subview and/or field-subview.
<criteria>	<related-model-subview>	The search criteria used to find the related models.
<privilege>	<related-model-subview>	Associates a privilege to the subview. If the user is not given this privilege, the subview will not be displayed for that user.

Add a Related Models Table Subview

You can use the <related-models-table-subview> to display a table of models that are associated with the current selected model based on specified search criteria.

You can use the elements in the following table to create a related model table subview.

Element	Parent Element	Description
<related-model-table-subview>	<subviews>	If you set the expanded attribute of this element to true, the subview will be expanded by default. The idref attribute enables you to associate an XML file that defines the data for this subview. The data for the subview does not have to be in another file, it is done for organizational purposes only.
<title>	<related-model-table-subview>	The title of the subview.
<privilege>	<related-model-table-subview>	Associates a privilege to the subview. If the user is not given this privilege, the subview will not be displayed for that user.
<table>	<related-model-table-subview>	This elements and its sub-elements define the table. See the table in Modify a Table Column for a list of sub-elements.
<criteria>	<related-model-table-subview>	The search criteria used to find the related models.

In the example that follows, the <subviews> element is used to place a view within a view allowing multiple views to be nested within each other. Each column in the table represents the value of an attribute for each of the models that have passed the search criteria.

Example: Related-Model-Table Subview (demo-details-config.xml)

```
<subviews>
   <related-models-table-subview>
    <title>Demo Table Title</title>
    <criteria>demo-search-criteria</criteria>
    <table>demo-table-config</table>
   </related-models-table-subview>
</subviews>
```

Example: Referenced Criteria XML (demo-search-criteria.xml)

```
<search-criteria id="demo-search-criteria"
   xmlns="http://www.aprisma.com"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.aprisma.com
   ../../common/schema/search-criteria-config.xsd">
   <child-models>
    <relation>Collects</relation>
   </child-models>
</search-criteria>
```

Example: Referenced Table XML (demo-table-config.xml)

```
<?xml version="1.0" encoding="utf-8"?>
<table id="demo-table-config"
   xmlns="http://www.aprisma.com"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.aprisma.com
      ../../common/schema/table-config.xsd">
   <swing-header-row-template>
    <static-color idref="row-header-color-config"/>
   </swing-header-row-template>
```

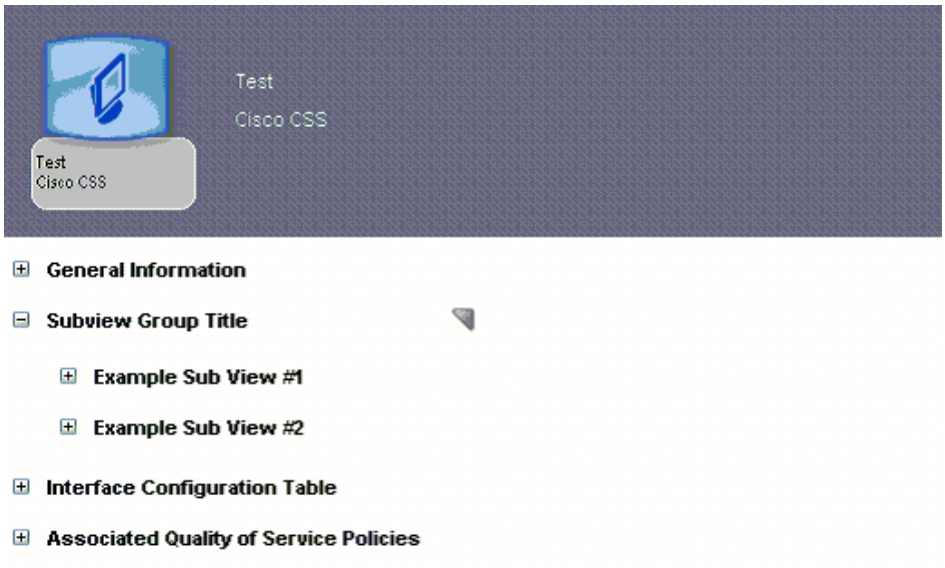
```
    <swing-row-template>
     <enumerated-color idref="alternatingrow-color-config"/>
    </swing-row-template>
    <column-list>
     <column>
         <name>Model Name</name>
         <content>
             <attribute>AttributeID.MODEL_NAME</attribute>
         </content>
     </column>
     <column>
         <name>Condition</name>
         <content>
             <attribute>AttributeID.CONDITION</attribute>
         </content>
     </column>
    </column-list>
 </table>
```

Define a Subview Group

You can group together one or more subviews under a single collapsible group using the <subview-group> element.

```
 <subview-group>
    <title>Subview Group Title</title>
    <display-if>
     <expression>
         attrInt(AttributeID.MTYPE_HANDLE) == 0x3cc0002
     </expression>
    </display-if>
    <subviews>
     <table-subview idref="example-table1-config">
         <title>Example Sub View #1</title>
     </table-subview>
     <table-subview idref="example-table2-config">
         <title>Example Sub View #2</title>
     </table-subview>
    </subviews>
 </subview-group>
```

This example generates a subview group similar to the one shown in the following figure.



Use the elements shown in the following table to create a subview group.

Element	Parent Element	Description
<subview-group>	<subviews>	If you set the expanded attribute of this element to true, the subview will be expanded by default. The idref attribute enables you to associate an XML file that defines the data for this subview. The data for the subview does not have to be in another file, it is done for organizational purposes only.
<title>	<subview-group>	The title of the subview group. In the example above, this is Subview Group Title.
<privilege>	<subview-group>	Associates a privilege to the subview group. If the user is not given this privilege, the subview group will not be displayed for that user.
<display-if>	<subview-group>	Adds an expression that will determine whether or not the group will be visible.
<subviews>	<subview-group>	Adds any type of subview (besides group) to this subview group.

Associate an Information Configuration File with a Model Class or Model Type

Once you have created the Information view with an Information Configuration file, you must follow the instructions below to associate the Information Configuration file with the model type or model class.

To associate an Information Configuration File with a Model

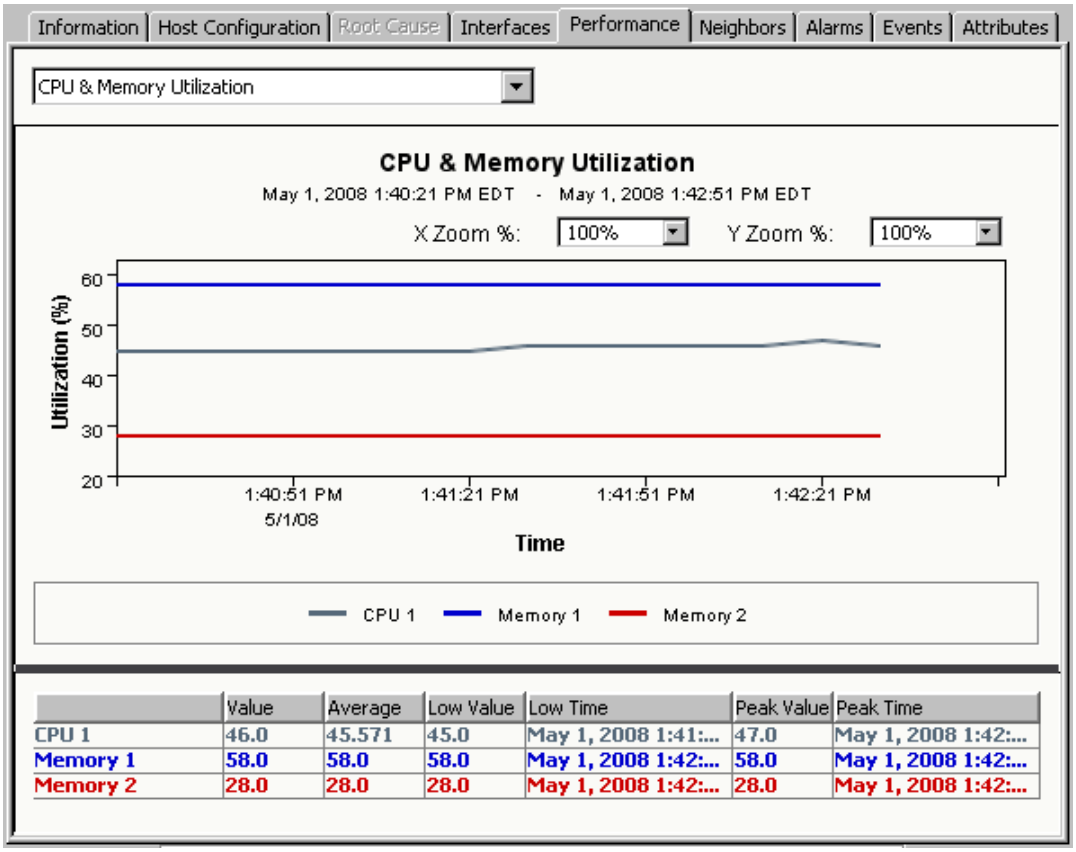
1. If it does not already exist there, copy the *<$SPECROOT>*/tomcat/webapps/spectrum/WEB-INF/console/config/custom-app-config.xml to the *<$SPECROOT>*/custom/console/config directory.
2. Open this file with a text editor.
3. Add the following block of XML code to link the appropriate model type(s) or model class(es) to the Information Configuration file. Use the XML elements shown to define the information appropriate to your model type or model class.

```
<contents-registry>
    <icon-reg-id>your-icon-registration</icon-reg-id>
    <tooltip-config>your-tooltip-config</tooltip-config>
    <information-config>your-information-config-file</information-config>
    <model-class>your-model-class</model-class>
</contents-registry>
```

4. Save and close the custom-app-config.xml file.
5. Restart the OneClick client for your changes to take effect.

Creating a Models Performance View

By default, some model types are configured to have a Performance view that shows changes in attributes, such as CPU utilization or memory, over time. In OneClick, you access this device view by clicking the Performance tab in the Component Detail panel.



A Performance view is composed of two XML files:

- **A performance data configuration file**
  This XML file specifies the data that can be displayed in any of the graphs within the Performance tab. Typically, this data includes attributes of the associated model type.
- **A performance view configuration file**
  This XML file defines the appearance of each graph available within the Performance tab. Each graph can display any of the lines defined within the performance data configuration file.

If the data or the format of one of the default Performance views does not meet your requirements, you can customize it for a particular model type or model class. For example, if you have added support in DX NetOps Spectrum for additional MIBs that are supported by a device, you might want to customize the view to graph some of the data that is available in the MIB. You can also create your own custom Performance views.

If a Performance view is not configured for the model currently selected in OneClick, the Performance tab is disabled.

Create a New Performance View

When you create a Performance view for a model type or model class from scratch, you should place the configuration files that define the view in the <*$SPECROOT*>/custom/topo/config directory. This helps to ensure they are not overwritten during an upgrade of DX NetOps Spectrum.

You associate the view's performance data configuration file with each applicable model type or model class in a file named custom-app-config.xml. While you can use either the <contents-registry> element or the <component-details-registry> element to do this, a best practice is to use the <component-details-registry> element because it configures only the Component Detail panel for the given model type or model class. For a description of the different registries available, see Chapter 5: Add Support for Model Types or Model Classes.

To create a new Performance view

1. In the <*$SPECROOT*>/custom/topo/config directory, create the performance data configuration file that specifies the data to be graphed in the view.
2. In the <*$SPECROOT*>/custom/topo/config directory, create a performance view configuration file that defines the appearance of each graph in the view.
3. In custom-app-config.xml, associate the performance data configuration file with the appropriate model types or model classes:
   a. If it does not already exist there, copy <*$SPECROOT*>/tomcat/webapps/spectrum/WEB-INF/console/config/custom-app-config.xml to the <*$SPECROOT*>/custom/console/config directory.

      NOTE
      Ensure to copy the file to the specified location. Do not modify the default custom-app-config.xml file that is provided with DX NetOps Spectrum because it is overwritten when you upgrade to a newer version.

   b. In custom-app-config.xml, add a block of XML code similar to the following example using a text editor. This code links the appropriate model types and model classes to the performance configuration data file.
      The following XML code example associates a model type whose ID is 0x3250004 to a performance data configuration file named <*$SPECROOT*>/custom/topo/config/performance-data-ciscovoiceapp-config.xml.

```
<component-details-registry>
  <performance-config>performance-data-ciscovoiceapp-config</performance-config>
  <model-type>0x3250004</model-type>
  <!-- CiscoVoiceApp -->
</component-details-registry>
```

      NOTE
      You can specify several model classes and model types within the contents or component details registries. You can also reuse Performance views in one or more <contents-registry> elements.

   c. Save and close the custom-app-config.xml file.
4. Restart the OneClick client for your changes to take effect.

Create a Performance Data Configuration File

The *performance data configuration file* specifies the data that can be displayed in any of the graphs within the Performance tab. A recommended naming convention for this XML file is performance-<descriptor>-data-config.xml.

Use the XML elements described in the following table to create a performance data configuration file.

Element	Parent Element	Description
performance-config	Not applicable	Represents the top-level parent element. You specify multiple graphs for a Performance view using multiple instances of the <graph> element in the performance view configuration file.

display	performance-config	Specifies the XML file that defines the view for presenting the graph data. This name must exactly match the simple file name of the actual performance view configuration file.
line	performance-config	Defines a line in the graph.
name	line	Specifies the label (name) for the line defined by the <line> parent element as it will be seen in the graph. The value for name needs to match its corresponding line definition in the performance graph view configuration file. If you are graphing a list attribute, you can also specify an attr-id attribute for the name element. This specifies an attribute ID whose value is appended to the name of each instance in the list. If not specified, the instance number is appended to the name of each list instance. In the following example, attribute 0x12ac6 represents the list of labels for the multiple lines defined by <list-content>. <line> <name attr-id="0x12ac6"> Memory Utilization </name> <list-content> <attribute>0x12ac6</attribute> </list-content> </line>
content	line	Specifies scalar data to the graph as a single line defined by the <line> parent element, for example: <content> <attribute>0x2100cc</attribute> </content>
list-content	line	Specifies list data to the graph as multiple lines defined by the <line> parent element. In the following example, the attribute 0x12ac6 is an integer list attribute that represents memory utilization. There will be a separate line graphed for each instance in the list. <line> <name attr-id="0x12ac6"> Memory Utilization </name> <list-content> <attribute>0x12ac6</attribute> </list-content> </line>
attribute	content, list-content	Specifies the ID of the attribute to graph as the line defined by the <line> parent element.
expression	content, list-content	Used to define an expression to produce a value for the column, for example: (attrInt(0xd054c) + attrInt(0xd054d))/8
applications	line	You can also graph data from related application models. Within the <applications> element, use the <model-type> element to specify the model type handle of the related application model. If the model in context has an application model related to it of this type, the data is retrieved from that application model. <applications> <model-type>0xc40043</model-type> </applications>
model-type	applications	The model type handle of the application model from which the data is retrieved. If no application models of this type are related to the model in context, the line is not shown.

The following example specifies the data to display in a 2-line performance graph that shows the change in both active and total VoIP calls over time for a Cisco device.

```
<performance-config id="performance-data-ciscovoiceapp-config">
   <display>performance-ciscovoiceapp-config</display>
   <line>
    <name>Active VoIP Calls</name>
<content>
        <attribute>0x325012b</attribute><!-- VoIP_Current_Calls -->
</content>
   </line>
   <line>
    <name>Total VoIP Calls</name>
<content>
        <attribute>0x3250129</attribute> <!-- VoIP_Total_Calls -->
</content>
   </line>
</performance-config>
```

NOTE
For additional, more complex examples of performance data configuration files, see the supporting files for the Performance views included with DX NetOps Spectrum. You can find these files by navigating to the <*$SPECROOT*>/tomcat/webapps/spectrum/WEB-INF directory and searching for files named perf*.

Create a Performance View Configuration File

The *performance view configuration file* defines the appearance of each graph available within the Performance tab. A recommended naming convention for this XML file is performance-<descriptor>-view-config.xml.

Use the XML elements described in the following table to create a performance view configuration file.

Element	Parent Element	Description
performance-view	Not applicable	Represents the top-level parent element.
graph	performance-view	Within the performance view configuration file, you can configure multiple graphs. Each graph is denoted by this <graph> element. The id attribute of this element is used as the graph title and displayed in the pulldown menu on the view (which is used for switching between multiple graphs for a single model). <graph id="CPU Utilization"> <y-axis-label>Utilization</y-axis-label> <y-axis-units>%</y-axis-units> <line> <name>CPU Utilization</name> </line> </graph>
y-axis-label	graph	Specifies the label for the Y axis.
y-axis-units	graph	Specifies the units for the Y axis, for example, % or Bits per Second.
line	graph	Defines a line in the graph, for example: <line color="#ffff00"> Use the color attribute to specify the hexadecimal RGB value of the color to use.

name	line	Specifies the label (name) for the line defined by the <line> parent element. This value must match the value for the same line in the performance data configuration file that defines the view's data. If you are graphing a list attribute, you may also specify an attr-id attribute for the name element. This represents the attribute id of which the value is appended to the name of each instance in the list. If not specified, the instance number is appended to name of each list instance.
display-if	line	Specifies the line should be displayed in the graph only if the expression defined in the <expression> child element evaluates to TRUE.
expression	display-if	Used to define an expression to define a complex condition for whether or not to graph the line. For more information, see Chapter 9: XML Usage Common to All Customization Files.
fill	line	If this element is included, the area below the line is filled in with color.

The following example specifies the format for a 2-line performance graph that shows the change in both active and total VoIP calls over time for a Cisco device.

NOTE
This is the format for the example graph whose data is defined in Create a Performance Data Configuration File.

```
<performance-view id="performance-ciscovoiceapp-config">
   <graph id="VoIP Calls Title">
    <y-axis-label>Calls</y-axis-label>
<y-axis-units>unit</y-axis-units>
<line>
   <name>Active VoIP Calls</name>
</line>
<line>
    <name>Total VoIP Calls</name>
</line>
   </graph>
</performance-view>
```

NOTE
For additional, more complex examples of performance view configuration files, see the supporting files for the Performance views included with DX NetOps Spectrum. You can find these files by navigating to the <*$SPECROOT*>/tomcat/webapps/spectrum/WEB-INF directory and searching for files named perf*.

Customize an Existing Performance View

In general, you customize an existing Performance view by overriding the default configuration files for the view with versions that contain your customizations.

To customize an existing Performance view

1. Identify the default configuration files that define the Performance view you want to customize:
   a. Open <*$SPECROOT*>/tomcat/webapps/spectrum/WEB-INF/topo/config/topo-app-config.xml and find the <contents-registry> or <component-details-registry> element for the appropriate model type or model class.

      NOTE
      If your model qualifies for both a model type and a model class registration, the model type registration takes precedence and is applied. Also, even though you can define the Performance view configuration in both the contents registry and the component details registry, the component details registry takes

precedence. The contents registry is primarily for model appearance and typically is applied to only the model class.

b. Find the <performance-config> element within the <contents-registry> or <component-details-registry> element, and note the name of the specified performance data configuration file.

NOTE
All of the default performance configuration files -- both the data configuration files and the view configuration files -- are located in the *<$SPECROOT>*/tomcat/webapps/spectrum/WEB-INF/*/config directories.

c. Open the *<$SPECROOT>*/tomcat/webapps/spectrum/WEB-INF/topo/config directory, and then open the performance data configuration file you identified in the previous step.

d. Find the <display> element within the <performance-config> element, and note the name of the specified performance view configuration file.

2. Copy over one or both of the performance configuration files that you identified in step 1 to the *<$SPECROOT>*/custom/topo/config directory. You only need to copy over a file if it requires customizations.

NOTE
To override the factory default performance configuration files, the copied files (that will contain your customizations) must have the same names as the original, default files.

3. If necessary, modify the copied performance data configuration file per your requirements, and then save and close the file.
4. If necessary, modify the copied performance view configuration file per your requirements, and then save and close the file.
5. If necessary, in custom-app-config.xml, change the model types or model classes that are associated with the performance data configuration file:

a. If it does not already exist there, copy *<$SPECROOT>*/tomcat/webapps/spectrum/WEB-INF/console/config/custom-app-config.xml to the *<$SPECROOT>*/custom/console/config directory.

NOTE
Make sure to copy the file to the specified location. Do not modify the default custom-app-config.xml file that is provided with DX NetOps Spectrum because it is overwritten when you upgrade to a newer version.

b. In custom-app-config.xml, add a block of XML code similar to the following example using a text editor. This code links the appropriate model types and model classes to the performance configuration data file.
The following XML code example associates a model type whose ID is 0x3250004 to a performance data configuration file named *<$SPECROOT>*/custom/topo/config/performance-data-ciscovoiceapp-config.xml.

```
<component-details-registry>
  <performance-config>performance-data-ciscovoiceapp-config</performance-config>
  <model-type>0x3250004</model-type>
  <!-- CiscoVoiceApp -->
</component-details-registry>
```

NOTE
You can specify several model classes and model types within the contents or component details registries. You can also reuse Performance views in one or more <contents-registry> elements.

The <component-details-registry> element within custom-app-config.xml *overrides* the equivalent registration for the model type or model class within an *-app-config.xml file. These registrations are used in the factory XML files in *<$SPECROOT>*/tomcat/webapps/spectrum/WEB-INF/*/config/*-app-config.xml.

c. Save and close the custom-app-config.xml file.

6. Restart the OneClick client for your changes to take effect.

Creating Custom Privileges

This section describes how to restrict access to menu items, attributes, and subviews using privileges.

Define a Custom Privilege

Define each new privilege in the custom-privileges.xml file. This file registers custom privileges that can be applied to the following components:

- Menu items
- Columns
- Subviews

If an administrator has not assigned the corresponding privilege to a user, that user cannot access the menu item, column, or subview.

Follow these steps:

1. Copy <*$SPECROOT*>/tomcat/webapps/spectrum/WEB-INF/console/config/custom-privileges.xml to the <*$SPECROOT*>/custom/console/config directory.
2. Open this file with a text editor.

   NOTE
   Define all new privileges inside the <privileges> element, which is the root element for the file.
3. Create the privilege. Use the elements shown in the table that follows this procedure.
4. Save and close the custom-privileges.xml file.
5. Restart the Tomcat web server, and then restart OneClick so that changes to the custom-privileges.xml file are available in OneClick.
6. You can now use the privilege to do the following:
   - Create a custom menu item, column, or subview that is accessible to only users who have the privilege. For more information, see Reference a Privilege When Defining a Menu Item, Column, or Subview.
   - Create a search that is accessible only to users who have the privilege.

   NOTE
   For more information, see OneClick Administration .

You can use the elements in the following table to create a privilege:

Element	Parent Element	Description
<privileges>	Not applicable	The root element for the custom-privileges.xml file.
<*your_privilege_name*>	<privileges> or <*your_group_name*>	Defines the privilege. You create an element for each new privilege. The type attribute for this element defines the default role to which you assign the privilege. Possible values for the type attribute are “read” or “write”. If you are grouping privileges, place all defined privileges for that group within the element that defines your group.
<label>	<*your_privilege_name*>	The name of the privilege, which will be shown on the privilege list.
<desc>	<*your_privilege_name*>	The description of the privilege.
<model-view-*attr*>	<privileges>	For more information, see Restrict Access to Attribute Values in Model Subviews.
<model-write-*attr*>	<privileges>	For more information, see Restrict Access to Attribute Values in Model Subviews.

[set the product group or family]	*<your_privilege_name>*	The new privilege appears in one of the existing groups if you use the [set the product group or family] element. The scope attribute defines group scope. The following list includes existing groups and their scope values: • <group scope="alarm"> alarm-manager </group> • <group scope="topo">tools</group> • <group scope="topo">model-tab</group> • <group scope="topo">model-view-group</group> • <group scope="topo">model-write-group</group> For more information see Group Privileges.

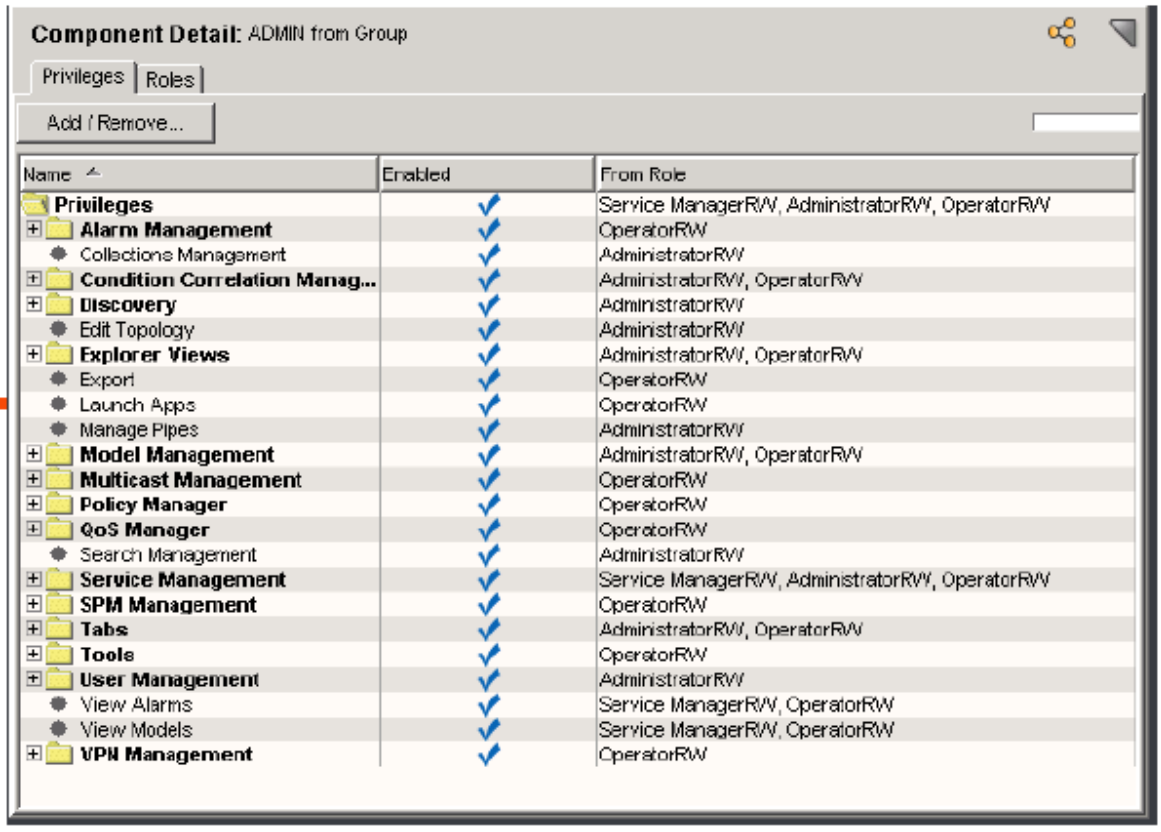
Example: Create a Privilege

NOTE
When you create a privilege, you are creating a new XML element. In the example above, the <launch-app> element creates the launch-app privilege. The type attribute defines the default role to which the privilege is assigned. Two values are possible: "read" and "write". A privilege with the "read" type is assigned to the OperatorRO role, and a privilege with the "write" type is assigned to the OperatorRW role.

The following example defines the launch-app privilege, as shown in the image:

```
<privileges>
   <launch-app type="write">
    <label>Launch Apps</label>
    <desc>Ability to launch application from the tools menu.</desc>
   </launch-app>
</privileges>
```

A newly defined "Launch Apps" privilege is made available in the privileges list.



Restrict Access to Attribute Values in Model Subviews

You can restrict a user's access to certain attributes using the <model-view-attr> and <model-write-attr> elements, where attr is equal to the attribute ID of the attribute you want to restrict. These elements are used in the custom-privileges.xml file and regulate the attributes that show up in the OneClick Privilege list's Model Management>View Attributes folder and the Model Management>Model Write folder.

The <model-view-attr> element enables you to create a privilege that determines whether or not a user can see an attribute. For example, if you added the following XML to the custom-privileges.xml file, you will create a privilege called Community Name. This privilege restricts view access to attribute 10024, community name. This privilege will appear in the Model Management > View Attributes folder as specified with the [set the product group or family] element. If the user does not have this privilege in any access group, they will not be able to see the community name attribute.

```
<model-view-10024 type="read">
   <label>Community Name</label>
   <group scope="topo">model-view-group</group>
</model-view-10024>
```

The <model-write-attr> element enables you to create a privilege that determines whether or not a user can edit an attribute. For example, if you added the following XML to the custom-privileges.xml file, you will create a privilege called Community Name. This privilege restricts write access to attribute 10024, community name. This privilege will appear in the Model Management > Model Write folder as specified with the [set the product group or family] element. If the user does not have this privilege in any access group, they will not be able to edit the community name attribute.

```
<model-write-10024 type="write">
   <label>Community Name</label>
   <group scope="topo">model-write-group</group>
</model-write-10024>
```

Group Privileges

If you want to group privileges together, you can create groups in the custom-privileges.xml file. To specify a group, nest the element that defines the privilege (<*your_privilege_name*>) between the element that defines the group (<*your_group_name*>). The group's <label> element defines the name that represents the group in the privileges tree (see the image later in this section).

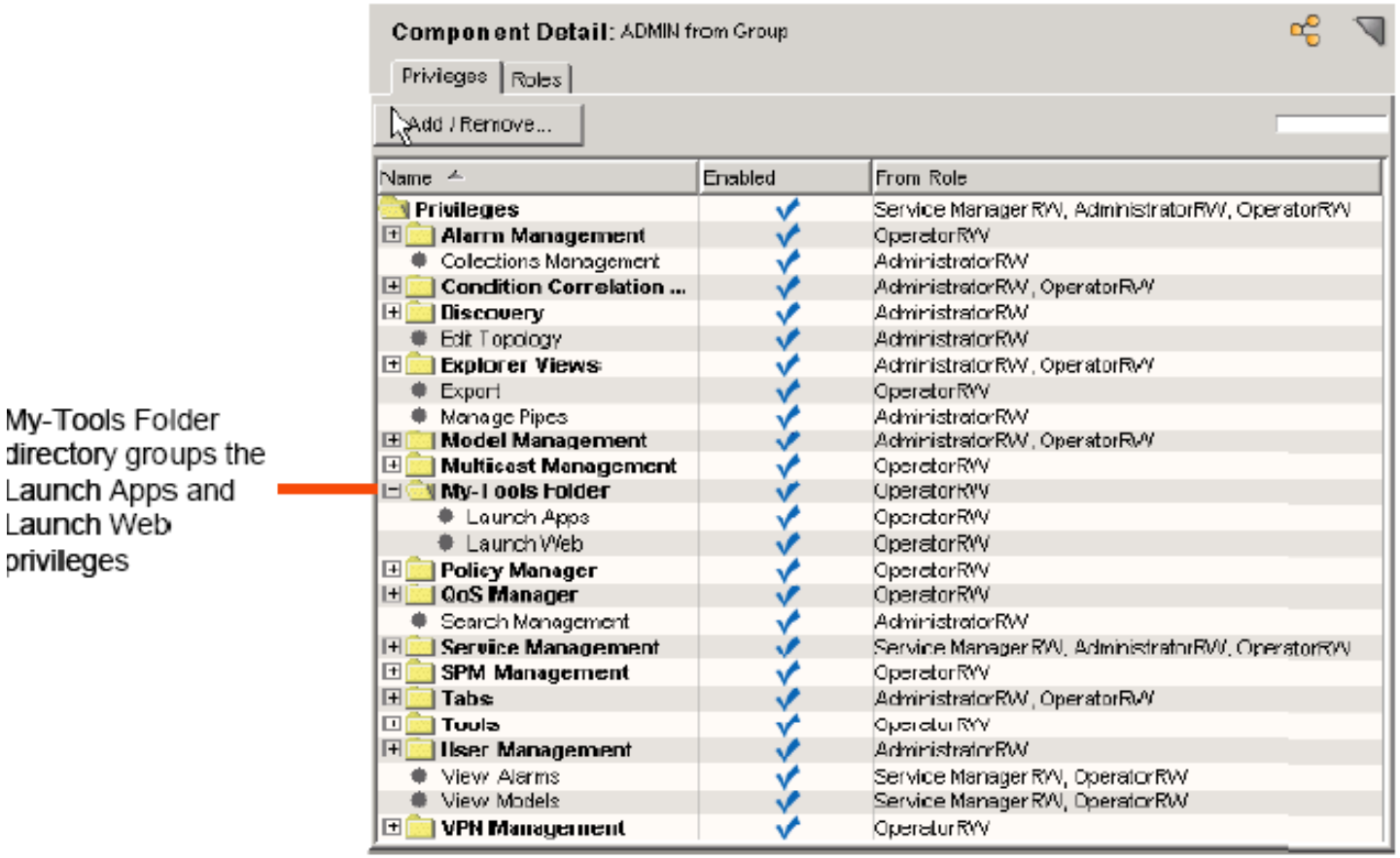
Element	Parent Element	Description
<privileges>	Not applicable	This is the root element for the custom-privileges.xml file.
<your_group_name>	<privileges>	This element defines the group. You create a new element for each group you define.
<label>	<your_group_name>	The name of the group, which will be shown on the privilege list.

NOTE
In order for changes made to the custom-privileges.xml file to be available in OneClick, you must restart the Tomcat web server and then restart OneClick.

In the following example, the <my-tools> element creates a group in which privileges can be nested. The value defined for the group's <label> is "My-Tools Folder". This will create a "My-Tools Folder" group in the privileges list as shown in the image that follows. The <launch-app> and <launch-web> privileges will appear in this group.

```
<privilege>
   <my-tools>
    <label>My-Tools Folder</label>
    <launch-app type="read">
        <label>Launch Apps</label>
```

```
        <desc>Ability to launch Applications.</desc>
    </launch-app>
    <launch-web type="read">
        <label>Launch Web</label>
        <desc>Ability to launch Web URLS.</desc>
    </launch-web>
   </my-tools>
</privilege>
```



Reference a Privilege When Defining a Menu Item, Column, or Subview

When you create a menu item, column, or subview, you can use the <privilege> element to reference a custom privilege. Custom privileges are defined in the custom-privileges.xml file. For example, if you have defined the "launch-app" privilege in the custom-privileges.xml file, you can use the following XML when you define a menu item, column, or subview:

```
<privilege>
   <name>launch-app</name>
</privilege>
```

This XML associates the "launch-app" privilege with the menu item, column, or subview. The user must have an associated role that grants the launch-app privilege in order for the menu item, column, or subview to be displayed. If granted, the menu item is always enabled.

NOTE
For more information, see OneClick Administration .

XML Usage Common to All Customization Files

This section explains common XML elements and strategies that can be used across customization files.

About Parameters

You can use the <param> element in many different instances to reference parameter values within a OneClick XML file. Here are several common cases where you will likely use the <param> element.

- If you need to pass a parameter to a web page, use the <param> element as a child element of the <url> element. See Launch a Browser for an example.
- If you need to pass a parameter to an application, use the <param> element as a child element of the <launch-application> element. See Launch an Application From OneClick for an example.
- If you need to pass a parameter to a command, use the <param> element as a child element of the <command> element. See Launch a Browser, Launch an Application From OneClick, and Launch a Web Server Script.
- If you need to format a series of values, use the <param> element in conjunction with standard HTML formatting elements. See Define Model Icon Tooltips for an example.
- If you need to manipulate the value of an attribute, you may need to use the <param> element when accessing one of the renderers.

See Acquire Data Render a Value for information on what you can specify using the <param> tag.

Acquire Data -- Render a Value

Acquiring data from OneClick about a model type parameter that you then act on is a fundamental process in customizing the OneClick interface. A set of elements provide the ability to acquire or render data from OneClick. These elements or tags are used in acquiring data to display in a table column, a field-subview column, an <param> element for a menu item, and the <render> element in a <dynamic-renderer>, and are shown in the following table.

Element	Description
<attribute>	Used to specify a DX NetOps Spectrum attribute
<select>	Used to specify something based on the value of another attribute, parameter, etc. Used to select a value based on certain criteria being met. Very generally, <select> this <if>condition1, <select> that <if> condition2.
<expression>	Used to define an arithmetic expression.
<renderer>	Used to define or access any number of renderers that process raw data and refine it into a specific format for presentation to the user.
<dynamic-renderer>	Specifies a renderer based on the value of an attribute criteria filter.
<message>	Used for specifying a plain text value for the column.

You can use any number and combination of these elements chained together, with the output from one element serving as the input to the next element in the chain. The <attribute> tag must be first in a chain of elements because it yields an attribute value and does not accept input. The <message> tag must be first in a chain of elements used to render a value. because it does not accept input.

Use a Select Case

If you want to conditionally display something in the OneClick interface, you may use the <select> and the <case> elements to create a decision structure similar to those used in many programming languages. Use the <select> and <case> elements as follows:

```
<select>
   <case>
    <expression>the expression to evaluate</expression>
    <yield>what to yield if the expression is true</yield>
   </case>
   <case>
    <expression>the expression to evaluate</expression>
    <yield>what to yield if the expression is true</yield>
   </case>
    .
    .
    .
   <default>what to yield if no matches are found</default>
</select>
```

Example: Image Definition File shows an example of the <select> and <case> elements used to select the image to be displayed on a OneClick device model icon depending on the model's condition.

Manipulate Attribute Output Using Renderers

There are several built-in attribute renderers that you can use to manipulate how the attributes you have specified in a OneClick table are displayed. You use the <renderer> element to access one of these renderers. The text of the element must be a fully-qualified Java class name; each allowable Java class name is explained below.

NOTE
You will need some background in programming to fully understand the renderer concepts presented below.

You can pass parameters to a renderer using the <param> element. The text of the <param> element is the parameter value. An <param> element must have a name attribute that specifies the name of the parameter.

Example

The following example specifies the BooleanRenderer with parameter trueTag set to No and parameter falseTag set to Yes. Each renderer has a set of parameters, and each renderer is defined differently.

```
<renderer>
   <param name="trueTag">Enabled</param>
   <param name="falseTag">Disabled</param>
   com.aprisma.spectrum.app.util.render.BooleanRenderer
</renderer>
```

Boolean Renderer

The class name for the boolean renderer is com.aprisma.spectrum.app.util.render.BooleanRenderer. This renderer outputs an enumerated String for an input Boolean value. By default, "Yes" is rendered for TRUE and “No” is rendered for FALSE, but other elements or text may be specified via the following parameters:

- trueTag - the tag or text to render for TRUE
- falseTag - the tag or text to render for FALSE

The following example reverses the TRUE/FALSE output:

```
<renderer>
```

```
   <param name="trueTag">No</param>
   <param name="falseTag">Yes</param>
   com.aprisma.spectrum.app.util.render.BooleanRenderer
</renderer>
```

If the input value is TRUE, “No” is rendered and if FALSE, “Yes” is rendered.

Commented Text Renderer

The class name for the commented text renderer is com.aprisma.spectrum.app.util.render.CommentedTextRenderer. This renderer strips off the HTML-commented prefix that is added by some renderers (for example, DateRenderer). It searches for the first occurrence of the ending character sequence of an HTML comment, such as <!---comment text --->, and returns the rest of the string.

Date Renderer

The class name for the date renderer is com.aprisma.spectrum.app.util.render.DateRenderer. This renderer outputs date and time using Java’s DateFormat. If the input to the renderer is a long integer (for example, of type java.lang.Long), it is assumed to represent the date and time in milliseconds. If the input is any other numeric type, it is assumed to be an integer representing the date and time in seconds. Otherwise, the only other valid input type is java.util.Date. The output string is prefixed by the numeric date and time value enclosed in HTML comments (<!---comment text --->). An example output would be:

```
<!--1089808869000-->Jul 14, 2004 8:41:09 AM EDT
```

You use the numeric value prefix in the comments tag for sorting. Without the prefix, DX NetOps Spectrum would sort on the formatted date and time string, and this would not work correctly. Therefore, you should only use the DateRenderer in the <content> element section of a column. To strip off the prefix for display, use the CommentedTextRenderer in the <swing-cell-template> section. For example:

```
<content>
   <attribute>0x11620</attribute>
   <! -- an attribute that contains an integer date/time in seconds >
   <renderer>
    com.aprisma.spectrum.app.util.render.DateRenderer
   </renderer>
</content>
<swing-cell-template>
   <text>
    <renderer>
        com.aprisma.spectrum.app.util.render.CommentedTextRenderer
    </renderer>
   </text>
</swing-cell-template>
```

Enumerated Attribute Renderer

Classname: com.aprisma.spectrum.app.util.render.EnumeratedAttrRenderer.

This renderer outputs an enumerated String for an attribute value. The renderer obtains the enumerations from the DX NetOps Spectrum database. You must specify the attribute ID via the “attrID” parameter. This renderer is most commonly preceded by an <attribute> element with the same attribute ID as the “attrID” parameter. The following sample XML renders the enumerated value for the Model_Class attribute (ID 0x11ee8):

```
<attribute>0x11ee8</attribute>
<renderer>
   <param name="attrID">0x11ee8</param>
   com.aprisma.spectrum.app.util.render.EnumeratedAttrRenderer
```

```
</renderer>
```

List Renderer

The classname for the list renderer is com.aprisma.spectrum.app.util.render.ListRenderer. This renderer outputs the components of a Java Collection or an array of any type as a comma-separated string.

Null Renderer

Classname: com.aprisma.spectrum.app.util.render.NullRenderer.

This renderer outputs a null input value as an empty string.

Object ID Renderer

This renderer outputs an object identifier (OID). The expected input value is type CsObjectID.

Classname: com.aprisma.spectrum.app.util.render.ObjectIDRenderer.

Supported parameters:

- term -- an integer value that specifies the index of a particular term of the OID to render
- startTerm -- an integer value that specifies the index of the first term of the OID to render
- endTerm -- an integer value that specifies the index of the last term of the OID to render

The term indices start at 1. If you specify the startTerm without the endTerm, then the portion of the OID from the startTerm to the last term of the OID is rendered. If you specify the endTerm without the startTerm, then the portion of the OID from the first term to the endTerm is rendered.

The ObjectIDRenderer is most commonly used to render the row instance of a MIB table. You obtain the row instance via the getRowId() method in an <expression> element. You can then pass the result to the ObjectIDRenderer. For example, the following column renders the first term of the row instance:

```
<column>
   <name>com.aprisma.spectrum.app.topo.client.ifIndex</name>
   <content>
    <expression>getRowId()</expression>
    <renderer>
        <param name="term">1</param>
        com.aprisma.spectrum.app.util.render.ObjectIDRenderer
    </renderer>
   </content>
</column>
```

The following example renders terms 5 through 8:

```
<column>
   <name>com.aprisma.spectrum.app.topo.client.NetworkAddr</name>
   <content>
    <expression>getRowId()</expression>
    <renderer>
        <param name="startTerm">5</param>
        <param name="endTerm">8</param>
        com.aprisma.spectrum.app.util.render.ObjectIDRenderer
    </renderer>
   </content>
</column>
```

The following example combines an expression with the ObjectID renderer to enable you to display the last term of an OID value in a table:

```
<column>
   <name>com.aprisma.spectrum.app.topo.client.ifIndex</name>
   <content>
    <expression>
        ((com.aprisma.spectrum.global.CsObjectID)value()).get_sub_oid(
        ((com.aprisma.spectrum.global.CsObjectID)value()).get_term_count(),
        ((com.aprisma.spectrum.global.CsObjectID)value()).get_term_count())
    </expression>
    <renderer>
        <param name="term">1</param>
        com.aprisma.spectrum.app.util.render.ObjectIDRenderer
    </renderer>
   </content>
</column>
```

Round Number Renderer

The classname for this renderer is com.aprisma.spectrum.app.util.render.RoundNumberRenderer. This renderer outputs a number rounded to the nearest 100th (or 2 decimal places).

System Up Time Renderer

This renderer outputs a numeric time value represented in one-hundredths of a second. The time representation is used in MIB objects such as sysUpTime. The output is expressed in days, hours, and minutes (for example, 30 days 1 hr 55 min).

Classname: com.aprisma.spectrum.app.util.render.SysUpTimeRenderer

Byte Renderer

This renderer outputs an integer value in byte units (byte, KB, MB, GB, or TB).

Classname: com.aprisma.spectrum.app.util.render.ByteRenderer

Inet Address Renderer

This renders an MIB object of type InetAddress as defined in RFC-3291.

Classname: com.aprisma.spectrum.app.util.render.InetAddressRenderer

Supported parameters:

- addressAttrID - the ID of the InetAddress attribute
- type - the InetAddressType as defined in RFC-3291
- typeAttrID - the ID of an attribute used to obtain the InetAddressType

List Instance Renderer

Renders the value of a specific instance of a list-type attribute.

Classname: com.aprisma.spectrum.app.util.render.ListInstanceRenderer

Supported parameters:

- oid -- the OID of the instance to render
- index -- the index of the instance to render

You must specify either the oid or index parameter.

Simple Integer Renderer

Renders an integer value without using comma grouping; 123456 instead of 123,456. Use this to substitute an integer value in a URL used in a menu item where commas are not acceptable input.

Classname: com.aprisma.spectrum.app.util.render.SimpleIntegerRenderer

Type Prepended Inet Address Renderer

Renders a MIB object of type InetAddress as defined in RFC-4293 with the type added to the beginning of the address.

Classname: com.aprisma.spectrum.app.util.render.TypePrependedInetAddressRenderer

Supported parameter:

addressAttrID - the ID of the InetAddress attribute

About <dynamic-renderer>

Use the <dynamic-renderer> element to specify a renderer that depends on the value of an attribute criteria such as <model_class>, <model-type>, or other attribute criteria. You select an attribute ID as the key and specify one or more <dynamic-renderer> elements in the custom-app-config.xml file. Each <dynamic-renderer> element defines a criteria and the renderer to use if the criteria is satisfied.

The structure to use with <dynamic-renderer> is as follows:

```
<dynamic-renderer>
   <attribute><KEY_ATTRIBUTE_ID></attribute>
   CRITERIA
    <render>
    .
    .
    .
    </render>
   <default/>
</dynamic-renderer>
```

The following table describes the elements you can use with <dynamic-renderer>.

Element	Usage and Description
<attribute>	Specifies the <KEY_ATTRIBUTE_ID> used to bind or tie together a set of dynamic-renderers.
CRITERIA	Defines an attribute filter criteria used to determine which renderer is used based on the filter output.
<render>	Defines what to render.
<default>	Specifies the dynamic-renderer to use as the default when none of the other dynamic-renderer criteria are met.

Attribute Filter Criteria and <dynamic-renderer>

It is common to use <model-class> and <model-type> for attribute filter criteria. You can use any attribute and any set of complex attribute filters with any combination of nested “and” and “or” filters. The file <*$SPECROOT*>/tomcat/webapps/spectrum/WEB-INF/common/schema/attributefilter.xsd contains the complete syntax for attribute filters.

Specify a Default <dynamic-renderer>

You define the default dynamic renderer for use when none of the conditions for using the <dynamic-renderer> specified in the CRITERIA statement are met. You can specify only one default dynamic-renderer per dynamic renderer set. Do not specify a filter criteria for the default.

Example: Using Attribute Filtering Criteria with <dynamic-renderer>

This example creates a column that displays an attribute based on the value of the model_type attribute. The attribute displayed for the model_type filter criteria conditions are shown in the following table.

Attribute to display...	if model_type is...
<attribute> 0xffff0000	<model-type> 0x12
<attribute> 0xffff0001	<model-type> 0x34 and 0x56
<attribute> 0xffff0002	for all other model types (default)

You must select one of the attributes specified in your filter criteria to be the key. This example uses 0xffff0002. Add the following <dynamic-renderer> elements to the custom-app-config.xml file:

```
<dynamic-renderer>
   <attribute>0xffff0002</attribute>
   <model-type>0x12</model-type>
   <render>
    <attribute>0xffff0000</attribute>
   </render>
</dynamic-renderer>
<dynamic-renderer>
   <attribute>0xffff0002</attribute>
   <or>
    <model-type>0x34</model-type>
    <model-type>0x56</model-type>
   </or>
   <render>
    <attribute>0xffff0001</attribute>
   </render>
</dynamic-renderer>
<dynamic-renderer>
   <attribute>0xffff0002</attribute>
   <render>
    <attribute>0xffff0002</attribute>
   </render>
   <default/>
</dynamic-renderer>
```

Example: Use a Key Attribute ID with <content>

This example creates a column specifying the <dynamic-renderer> element with the key attribute ID defined in the <content> element.

```
<column>
   <name>My Column</name>
   <content>
    <dynamic-renderer>0xffff0002</dynamic-renderer>
   </content>
</column>
```

About Expressions

When you are customizing the OneClick interface, there are several places where you may want to use an expression to display a calculated value. For example, you may want to display a calculated value in a table or subview. The section below explains how to use expressions to manipulate attribute information.

NOTE
Expressions are created using standard Java expressions. You must be familiar with Java code in order to implement the following instructions that create expressions in the OneClick XML files. If you are not familiar with Java code, you should refer to a Java reference before attempting to create expressions when customizing OneClick files.

Manipulate Attribute Information

The most common use of an expression is to manipulate attribute information. The attribute information available is dependent upon the OneClick context in which you are using the expression.

You can use the methods listed in the following table in the context of an expression to retrieve attribute information:

java.lang.Object	**attr (int attrID)**
boolean	attrBoolean (int attrID)
byte	attrByte (int attrID)
char	attrChar (int attrID)
double	attrDouble (int attrID)
float	attrFloat (int attrID)
int	attrInt (int attrID)
long	attrLong (int attrID)
short	attrShort(int attrID)

The following example shows a column configuration that displays the contact person for a device. In this example, an expression displays the attribute 0x23000c (AttributeID.CONTACT_PERSON) if the attribute 0x10b5a (AttributeID.SYS_CONTACT) is null or has no value.

Example: Specifying a Contact for a Device Using an Expression

```
<column id="column-contact-config"
   xmlns ="http://www.aprisma.com"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.aprisma.com
   ../../common/schema/column-config.xsd">
   <name>Contact Person</name>
   <content>
    <expression>
        ( attr( AttributeID.SYS_CONTACT ) == null ||
        ((String)attr(AttributeID.SYS_CONTACT)).length() == 0) ?
        attr( AttributeID.CONTACT_PERSON ) : value()
    </expression>
   </content>
</column>
```

Another way to accomplish the same result is to use the attribute renderer to retrieve the SYS_CONTACT attribute value. You can then access the value returned using an expression that uses the value() method.

Example: Using Attribute Renderer to Retrieve Attribute Value

```
<column id="column-contact-config"
   xmlns ="http://www.aprisma.com"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.aprisma.com
   ../../common/schema/column-config.xsd">
```

```
    <name>Contact Person</name>
    <content>
     <attribute>AttributeID.SYS_CONTACT</attribute>
     <expression>
         (value() == null || ((String)value()).length() == 0) ?
         attr( AttributeID.CONTACT_PERSON ) : value()
     </expression>
    </content>
 </column>
```

The two examples shown above produce the same value for the column.

Append Suffix to Values

Use an expression to append a suffix to values to increase readability of information displayed in tables.

Example

This example appends a “%” character to a value so that the value displays in a table as <value>%.

<expression>value().toString() + "%"</expression>

You can use this method to append a “%” character to a “percentage of disk space used” value, so that the value displays in a table as <percentage of disk spaced used>%, or 64%.

Precautions for Using Expressions

The following list describes major exceptions to the rules for standard Java code that are used to create OneClick expressions.

- **Comparison Operator**
  You cannot use the comparison operator, &&, due to restrictions on XML formatting. In place of && you must use &&.
- **Less Than, Greater Than Operators**
  You cannot use the less than (<) or greater than (>) operators. Instead, you must use < and > respectively.
- **Subtraction Expressions**
  OneClick processes subtraction expressions using non-standard associativity. Subtraction is done using a right-to-left associativity instead of the standard left-to-right associativity.
  OneClick processes subtraction as follows:
  A - B - C = A - (B - C)
  compared with standard subtraction expression processing:
  A - B - C = (A - B) - C

Reference XML Files

As you are customizing OneClick XML files, you may find it necessary to split a single XML file into two or more XML files for the following reasons:

- Some XML files are so complex that they become unreadable. In this case breaking the XML file down into two or more files assists you in keeping your code organized and making it readable and editable in the future.
- You may want to reuse certain sections of XML code. Putting this XML code in a separate file allows you to reference it from multiple files instead of copying and pasting it into new files, or new sections of the same file.

Use the standard XML id and idref attributes to label and reference the split up code.

NOTE
For information on XML standards, including id and idref, see .

Reference Images

You may need to reference image files from within your XML. When you reference image files in either the factory <*$SPECROOT*>/tomcat/webapps/spectrum/images directory or the custom <*$SPECROOT*>/custom/images directory, express the path starting from the images directory, for example, images/myimage.png. See Example: Icon Configuration File for an example.

You must place all image files that you add or customize in the <*$SPECROOT*>/custom/images directory. Otherwise, all new or customized images you add will be deleted or overwritten during a DX NetOps Spectrum or OneClick upgrade or reinstallation.

Verify User Input Using Verifiers

You can verify user input by specifying a verifier class along with the <editable> element before committing the change. If the input is invalid, an error message is displayed. The verifiers available are described in the following section.

Specify the <verifier> element inside the <editable> element. Inside the <verifier>, you specify a verifier Java class and optional parameters to pass to the verifier class.

Example: Using Verifiers

This verifies the input value is from 0-100, inclusive.

```
<editable>
   <verifier>
    <class>
        com.aprisma.spectrum.app.swing.widget.IntegerContainedInRangeInputVerifier
    </class>
    <param name="lowValue">0</param>
    <param name="upperValue">100</param>
   </verifier>
</editable>
```

OneClick Input Verifiers

IntegerContainedInRangeInputVerifier

Description: Verifies the input is an integer value within a specified range.

Class:com.aprisma.spectrum.app.swing.widget.IntegerContainedInRangeInputVerifier

Parameters:

- lowValue - the lower bound of the range
- upperValue - the upper bound of the range

AttrIDInputVerifier

Description: Verifies the user input is a valid attribute.

Class: com.aprisma.spectrum.app.swing.widget.AttrIDInputVerifier

DoubleInputVerifier

Description: Verifies the user input is a valid real number.

Class: com.aprisma.spectrum.app.swing.widget.DoubleInputVerifier

IPAddressInputVerifier

Description: Verifies the user input is a valid IP address.

Class: com.aprisma.spectrum.app.swing.widget.IPAddressInputVerifier

IntegerInputVerifier

Description: Verifies the user input is a valid integer.

Class: com.aprisma.spectrum.app.swing.widget.IntegerInputVerifier

LongInputVerifier

Description: Verifies the user input is a valid long integer.

Class: com.aprisma.spectrum.app.swing.widget.LongInputVerifier

MACAddressInputVerifier

Description: Verifies the user input is a valid MAC address.

Class: com.aprisma.spectrum.app.swing.widget.MACAddressInputVerifier

NonEmptyStringInputVerifier

Description: Verifies the user input is a non-empty string.

Class: com.aprisma.spectrum.app.swing.widget.NonEmptyStringInputVerifier

UnsignedIntInputVerifier

Description: Default verifier for all integer attributes; verifies the user input is an unsigned integer.

Class: com.aprisma.spectrum.app.swing.widget.UnsignedIntInputVerifier

Customizing OneClick for CA Service Desk

For a DX NetOps Spectrum and CA Service Desk integration, you can modify the behavior of finding and creating Service Desk assets from OneClick. This customization is done by changing the attribute mapping between DX NetOps Spectrum models and Service Desk assets. Customizing asset reporting lets you prioritize the information used to identify a device and determine which information to record within Service Desk. How information is recorded in Service Desk can enhance the user's efficiency and reporting capabilities to best suit your organization.

> **NOTE**
> For more information about customizing asset reporting for CA Service Desk, see CA Service Desk and DX NetOps Spectrum .

TL1 Gateway

What Is TL1?

Transaction Language 1 (TL1), occasionally referred to as MML (Man Machine Language), is a widely used protocol in telecommunications management. The TL1 protocol can be used to manage most telecom network elements in North America.

Unlike SNMP, TL1 is a human-machine interface that contains human-readable strings. Also, unlike SNMP, TL1 includes no concept of a MIB. TL1 was originally specified by Bellcore in 1986 and is now maintained by Ericsson.

What Is TL1 Gateway?

The TL1 Gateway for DX NetOps Spectrum translates TL1 events and alarms originating from a TL1 device into DX NetOps Spectrum events and alarms. The gateway acts as a mediator between TL1 devices and DX NetOps Spectrum.

Each TL1 device is represented by a corresponding device model within DX NetOps Spectrum. As a result, you can launch the Enterprise Alarm Manager application for a particular model so that you can check for certain alarm conditions and initiate corrective action. TL1 Gateway also supports TL1 devices that are accessible through proxy devices, also known as Gateway Network Element (GNE) devices.

TL1 Gateway implements the full range of TL1 Condition Types as specified by Telcordia. Telcordia has been acquired by Ericsson. Multiple websites provide lists of current network element and transport surveillance messages.

What Does TL1 Gateway Include?

TL1 Gateway for DX NetOps Spectrum provides the following components:

- TL1-specific inference handlers that plug into a SpectroSERVER.
- A daemon (TL1d) that handles communication between the TL1 devices and the SpectroSERVER.
- A model type (Gen_TL1_Dev) for generic TL1 devices.
- A utility (tl1map) to manage TL1 AlarmMaps.

Installing TL1 Gateway

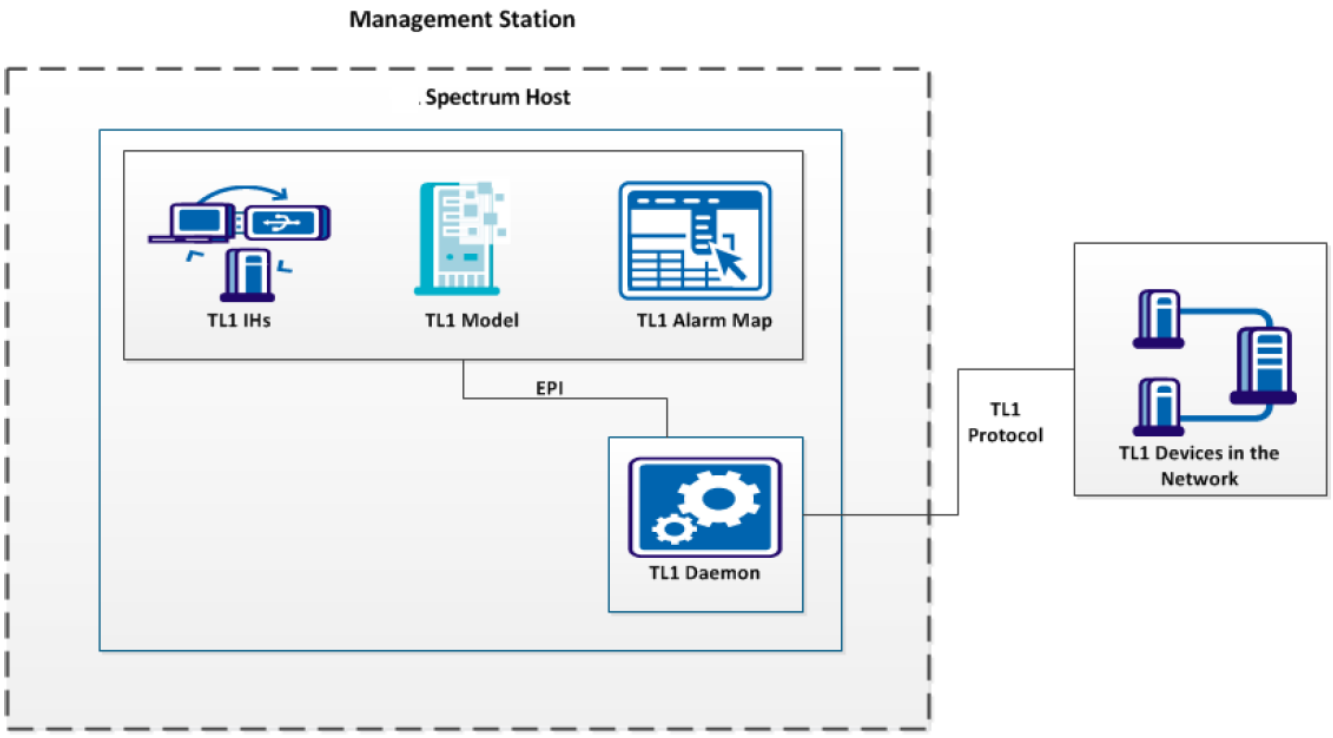
Prerequisites for TL1 Gateway

TL1 Gateway is designed such that it can either be run on the same machine DX NetOps Spectrum is running on, or on a separate machine. Diagrams illustrating both installation models are provided on the subsequent pages. Running TL1 Gateway on a separate machine is recommended if any of the following conditions apply:

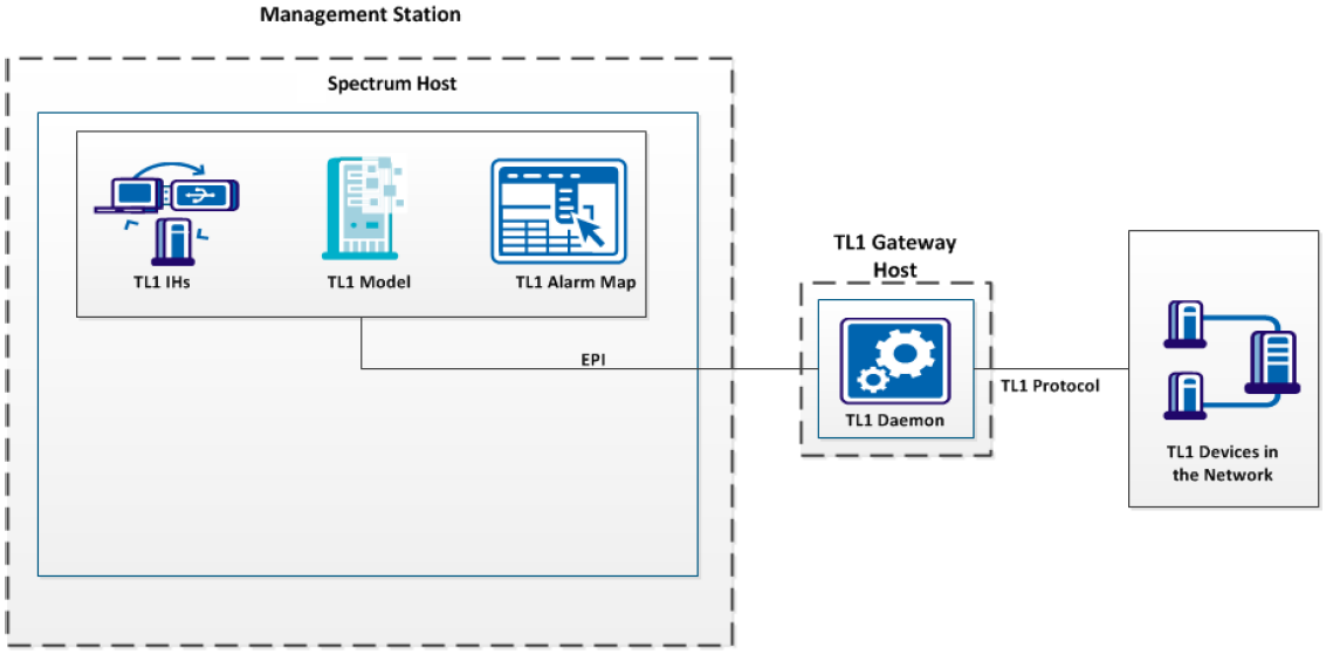
- The DX NetOps Spectrum machine hosts a SpectroSERVER that has a high workload.
- The DX NetOps Spectrum machine is short on resources like CPU and RAM.
- The DX NetOps Spectrum machine is also used by applications other than DX NetOps Spectrum.
- A high volume of TL1 traffic has to be processed.

> **NOTE**
> TL1 Gateway does not support IPv6.

The following diagram shows an example of a TL1 Gateway installed on DX NetOps Spectrum host:



The following diagram shows an example of a TL1 Gateway installed on a remote host:



Installation Options

If your configuration requires TL1 Gateway to be run on the same machine DX NetOps Spectrum is running on, then the gateway has already been installed during the major DX NetOps Spectrum installation. In this case, all required TL1

Gateway components can be found in a subdirectory called TL1Apps, which is located in your top-level DX NetOps Spectrum directory. The TL1 Apps directory contains the following files:

- TL1d -- the gateway daemon
- tl1map -- the AlarmMap utility

Install on a Non-DX NetOps Spectrum Machine

In some cases, where a certain level of performance is needed, it is preferable to install the TL1 daemon component on a separate server that is dedicated to TL1 message processing. The instructions vary slightly depending on the platform you are using. In either case, you must find and run the self-extracting installer file available with the DX NetOps Spectrum application.

The installer launches a series of dialogs that prompt you to supply the following information:

- confirmation of the license agreement
- the extraction key
- installation target area

> **NOTE**
> Regardless of the platform on which DX NetOps Spectrum is running (Windows), the TL1 daemon component can be installed and operated.

Follow these steps on Windows:

1. Navigate to the TL1GW folder on the DX NetOps Spectrum application installer.
2. Run the tl1inst.exe program.
3. Copy the resulting files (listed below) to the proper locations on the NT machine:
   - TL1d.exe
   - tl1map.exe
   - orb.dll
   - orb_p.dll

   Copy the executables TL1d.exe and tl1map.exe to the preferred locations. Copy the DLL files to the system area that is used for third-party DLL files.

Setting up Port Numbers

When you create a TL1 device model, specify the two port numbers that are described below. Before, you assign any port numbers, first check the local system administration or security policies.

- **TL1 Gateway Port**
  Used for communication between the TL1 daemon (TL1d) and the SpectroSERVER. At model creation, a default port number of 64222 is supplied. When choosing a different port number, we urge you not to use Port 65535. Ask a system administrator to find a port number in the dynamic/private range 49152 to 65534.
  The TL1 Gateway port number can be specified as a command-line argument when launching TL1d; otherwise, the default of 64222 is assumed.
- **TL1 Device Port**
  Used for communication between the physical TL1 device and the TL1 daemon. Check with the network administrator for the port number to use; it depends on device configuration. It is likely to be a port number specified in the IANA standard document. Check the “Registered Port Numbers” section. Likely candidates are: 2361, 3081, 3082 and 3083. IANA port assignments are available on the Internet.
  Supply the TL1 port number in the model creation dialog.

Creating a TL1 Device Model

The TL1 Device models are used to check for alarm conditions and initiate corrective action through Enterprise Alarm Manager. The TL1 devices must be modeled manually as they cannot be modeled through Spectrum Discovery. For more information about Discovery, see Modeling and Managing Your IT Infrastructure .

Follow these steps:

1. From any OneClick Topology tab view, select the Model by Type toolbar icon.
2. Select the 'All Model Types' tab and enter TL1 into the Filter field.
3. Select Gen_TL1_Dev to model a generic TL1 device and click OK.
4. Complete the following fields to configure the TL1 device model:
   - **Model Name**
     Specifies a unique name for this model.
   - **TL1 Gateway Address**
     Specifies the IP address of the server that hosts the TL1 daemon (TL1d).
   - **TL1 Gateway Port**
     Specifies the port number for communication between the TL1 daemon and DX NetOps Spectrum.
     Default: 64222
   - **TL1 Device Address**
     Specifies the IP address of the TL1 device.

     NOTE
     Sometimes, this address is the IP address of a dedicated TL1 device that acts as a proxy for all the other TL1 devices in the network.
   - **TL1 Device Port**
     Specifies the port number where the TL1 device agent is listening (also known as the “craft” port).
   - **Max Telnet Sessions**
     Specifies the total number of Telnet sessions TL1 Gateway can make to Gateway Network Element (GNE).
     Default: 1
   - **Max Logins Per Telnet**
     Specifies the number of logins that are permitted to be active simultaneously per telnet session. This field value depends on the GNE.
   - **TID**
     Specifies the “Target ID” for the TL1 device. This parameter is part of the TL1 addressing concept, and is required to uniquely identify a device. The TID is a predefined alphanumeric string. Obtain it from the local Network Administrator.
   - **User Name**
     Specifies the user ID for the maintenance user, as configured in the TL1 device.
   - **Password**
     Specifies the password for the maintenance user.
   - **Security String**
     Prevents selected users from viewing this model.
   - **Poll Interval (sec)**
     Specifies the frequency at which the device is polled for status updates. Change the value as needed, to increase or decrease the polling interval.
     Default: 60 (300 for some model types)

NOTE
If you increase the time between polling intervals, less bandwidth is required for traffic management. However, you receive device status updates less frequently.We recommend using the default polling interval for critical devices and using 600 seconds for less important devices.

– **Log Ratio**
Defines how many times DX NetOps Spectrum polls devices for updates before logging the results.
Default: 10 (DX NetOps Spectrum logs the polling results after it polls the device every tenth time).

NOTE
No entries or adjustments to the defaults are required for the remaining fields. For more information about field settings, see OneClick Administration.

5. Click OK.
A TL1 device model icon is displayed.

Max Telnet Sessions and Max Logins Per Telnet Attribute Considerations

Max Telnet Sessions

The Max Telnet Sessions field enables you to specify the total number of telnet sessions that TL1 Gateway can make to Gateway Network Element (GNE). Default is 1.

TL1 devices exist in a ring or group. The Gateway Network Element acts as a go-between the rest of the network and the TL1 devices within the ring. Therefore, DX NetOps Spectrum makes a telnet connection to the GNE.

This varies with hardware maximum device specifications. DX NetOps Spectrum can use as many sessions as the Max Telnet Sessions value has been set to allow. You can specify Max Telnet Sessions to one less than the hardware maximum to leave one available for the Network Administrator, but it is not required.

For any given ring, the number of telnet sessions that are used is the highest Max Telnet Sessions value among all the Gen_TL1_Dev models that are associated with that ring. So, if you have ten devices that are modeled for a single ring, if all the Max Telnet Sessions values are 1, then the TL1 daemon (TL1d) uses at most one telnet session when communicating with all the devices on that ring. If nine of the models have a Max Telnet Sessions value of 1 and one of the models has a Max Telnet Sessions value of 4, then the TL1d can use up to four concurrent telnet sessions when communicating with ANY of the devices on that ring.

The recommended usage for Max Telnet Sessions is to set it to the desired value on the Gen_TL1_Dev model which represents the TL1 proxy device and to leave the value at its default for other Gen_TL1_Dev models. That way, if you want to change the value, you only have to change it on one model.

Max Logins Per Telnet

The Max Logins Per Telnet field enables you to enter the number of logins that are permitted to be active simultaneously per telnet session. This depends on the GNE. Some authorize one active login at a time, others allocate more.

As with Max Telnet Session, when two or more devices from the same ring are modeled, the TL1 deamon (TL1d) uses the highest Max Logins Per Telnet value among all the Gen_TL1-Dev models that are associated with that ring.

The recommended usage for Max Logins Per Telnet is to set it to the desired value on the Gen_TL1_Dev model which represents the TL1 proxy device and to leave the value at its default for other Gen_TL1_Dev models. That way, if you want to change the value, you only have to change it on one model.

Important Processing Information for Max Telnet Sessions and Max Logins Per Telnet

It is important to understand the difference between having a value for Max Logins Per Telnet of 1 versus a value greater than 1.

When Max Logins Per Telnet Equals 1

When Max Logins Per Telnet equals one, the telnet connections are time-shared. Therefore, polls cannot occur simultaneously, but instead, are carried out in sequence. The maximum number of devices manageable is dependent primarily on the polling interval and the value of Max Telnet Sessions. Congestion and slow device response can adversely affect the management of the devices by DX NetOps Spectrum.

When Max Logins Per Telnet Is Greater Than 1

When Max Logins Per Telnet is greater than one, polls can occur simultaneously. The maximum number of devices manageable is a hard limit of Max Telnet Sessions multiplied with Max Logins Per Telnet. Therefore, congestion and slow device response has less impact on the ability of DX NetOps Spectrum to properly manage the devices.

The TL1 daemon (TL1d) communicates with TL1 devices over shared telnet sessions. That is, if you have say, ten TL1 devices on a single TL1 ring, TL1 Gateway can send ping requests to and can receive messages from all ten devices over a single telnet connection.

The TL1 daemon (TL1d) can also have several concurrent telnet sessions open to the same TL1 ring. So, TL1 Gateway could be using three telnet sessions open simultaneously to communicate with ten TL1 devices on a single ring.

Congestion Considerations

Because communication is shared over the connections, congestion can occur. The amount of congestion is determined through:

- Number of concurrent telnet sessions available to TL1 Gateway for use to the TL1 ring in question
- Number of TL1 devices on the ring that is managed in DX NetOps Spectrum
- Polling interval of the TL1 models
- Frequency of alerts sent to TL1 Gateway from the TL1 devices
- Time delay in the TL1 devices responding to commands from TL1 Gateway

If congestion over the telnet session is bad enough, ping requests that the SpectroSERVER sends does not return in time, and the TL1 models will sporadically go into a contact lost state.

If the network congestion is caused by too many alerts from the TL1 devices, a solution might be to specify an autonomous port for the devices.

If Max Logins Per Telnet equals one, the following two solutions are available:

- Increase the polling interval of the TL1 models -- the devices are polled less often causing less congestion, resulting in better performance.
- Increase the telnet sessions to TL1 devices ratio, by either increasing the number of concurrent telnet sessions the TL1d can use or by decreasing the number of TL1 devices on the ring.

Upgrade Considerations

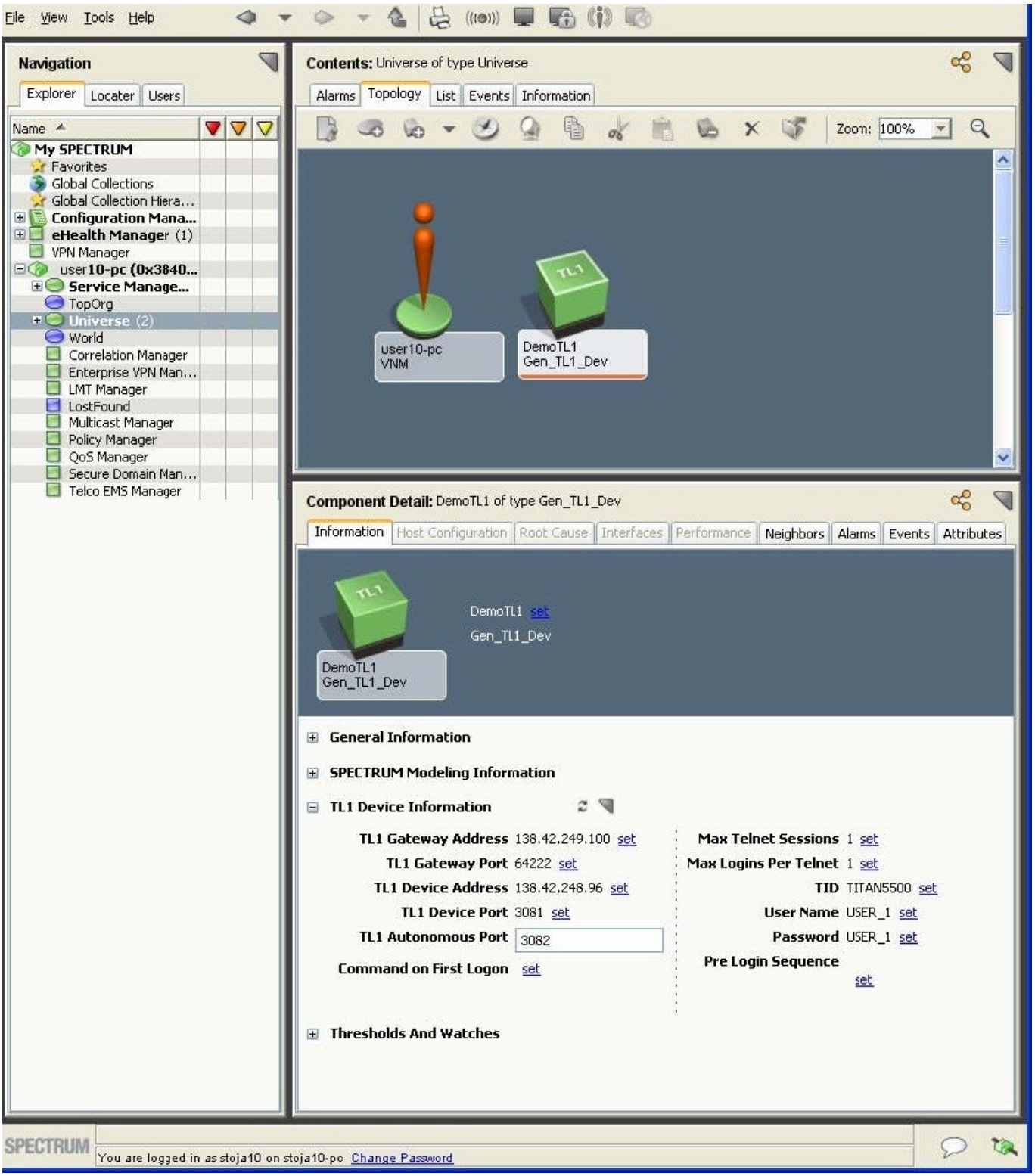
Of particular importance to customers who are upgrading:

- Due to the overhead caused by the new functionality, we have increased the default value for the DCM Timeout attribute (0x110c4) from 3000 to 5000. However, if the customer is upgrading, this change is not made because the Timeout attribute has the 'Preserve Value' flag set.
- If the DMC timeout value is left at 3000, the ping requests sent by the SpectroSERVER will probably time out sporadically. After upgrading, customers change the Timeout value for all Gen_TL1_Dev models that they have currently modeled to be 2000 milliseconds higher than their current value. Likewise, they must also increase the Gen_TL1_Dev model type default value for Timeout by 2000 milliseconds.

TL1 Devices with Autonomous Port

Some TL1 devices support command ports only and have a port dedicated to autonomous messages. TL1 Gateway supports this option.

To use it, highlight the TL1 device icon and select the Information tab in the Component Detail view. The TL1 Autonomous Port can be set in the TL1 Device Information section.



Command on First Logon

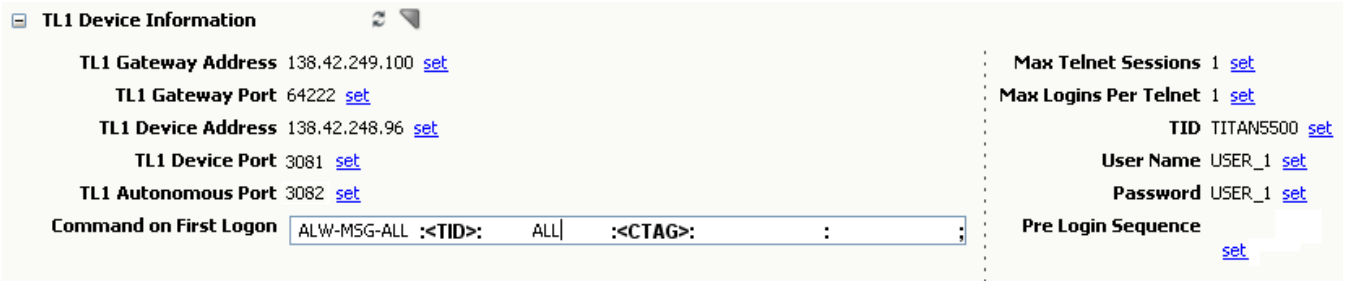
The value that appears in the Command on First Logon field is issued to the device the first time you log into that device. In the image shown below, the ALW-MSG-ALL command enables the device to send alerts to the TL1 Gateway. Note that some devices do not send alerts automatically. The gateway can then relay the alerts to DX NetOps Spectrum as it receives them.

You can customize the Command on First Logon for any command that requires customization. DX NetOps Spectrum automatically replaces the <TID> and <CTAG> values. The <TID> value uses the TID value that you specified in the Create Model Type dialog. The <CTAG> value is automatically generated by the TL1 Gateway for proper communication.

The following image shows an example of the Command on First Logon field in the TL1 Device Information subview.

NOTE

The TL1 Device port and TL1 Autonomous port can be same.



Pre Login Sequence

Before logging into a TL1 device, the TL1 daemon can send a few characters to the device to wake up the telnet session. These characters are stored in the Pre Login Sequence attribute. The default is a single newline character.

For some cases, multiple newlines may be more appropriate. To change the Pre Login Sequence, highlight the TL1 device icon and select the Information tab in the Component Detail view. The Pre Login Sequence can be set in the TL1 Device Information section.

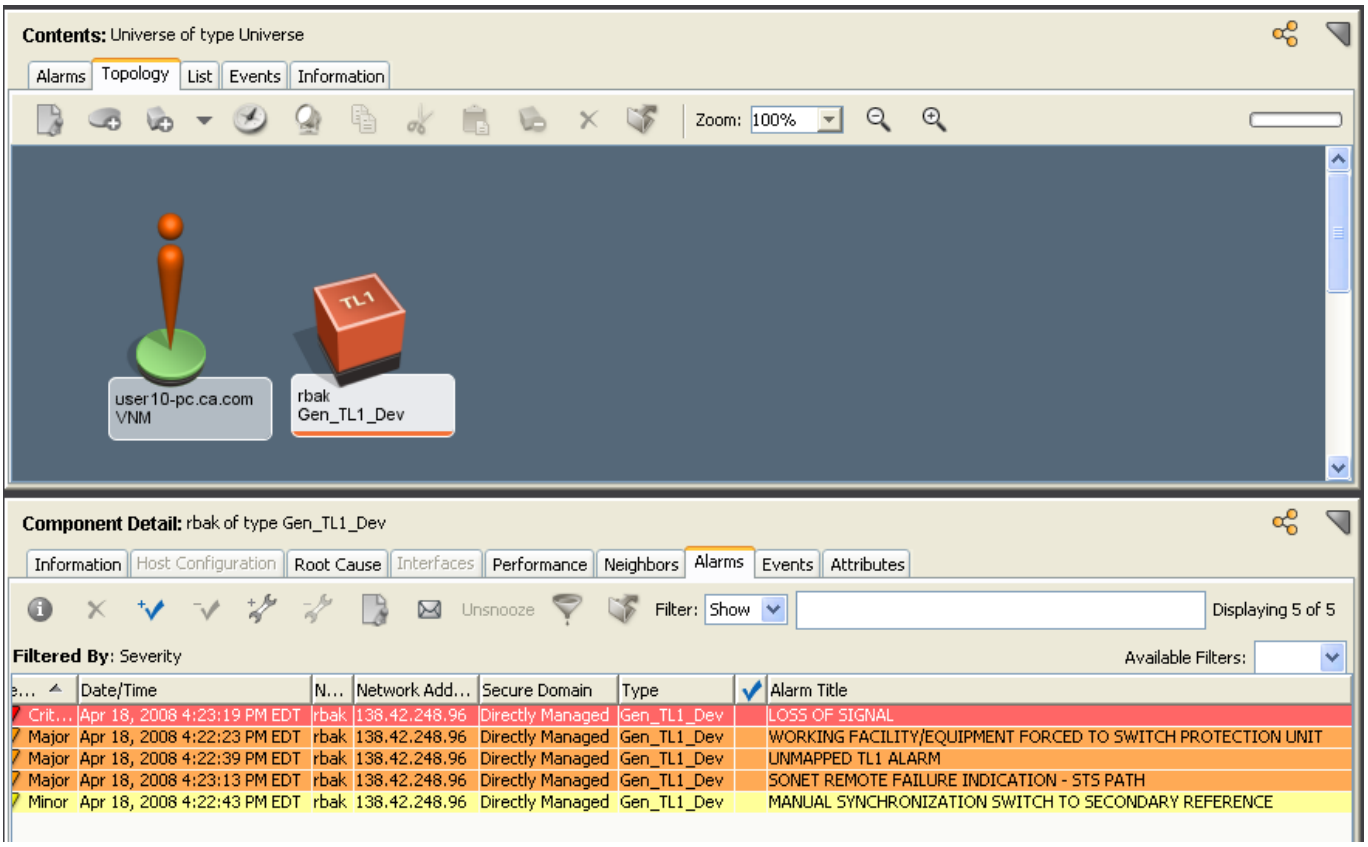
The TL1 AlarmMap

The TL1 Gateway AlarmMap is an internal mapping of TL1 events and alarms to DX NetOps Spectrum alerts. The AlarmMap is used by the TL1 daemon to determine whether an incoming TL1 event or alarm is discarded or be converted to an alert code and forwarded to DX NetOps Spectrum. Thus the AlarmMap functions as a filter to discard unnecessary TL1 events or alarms.

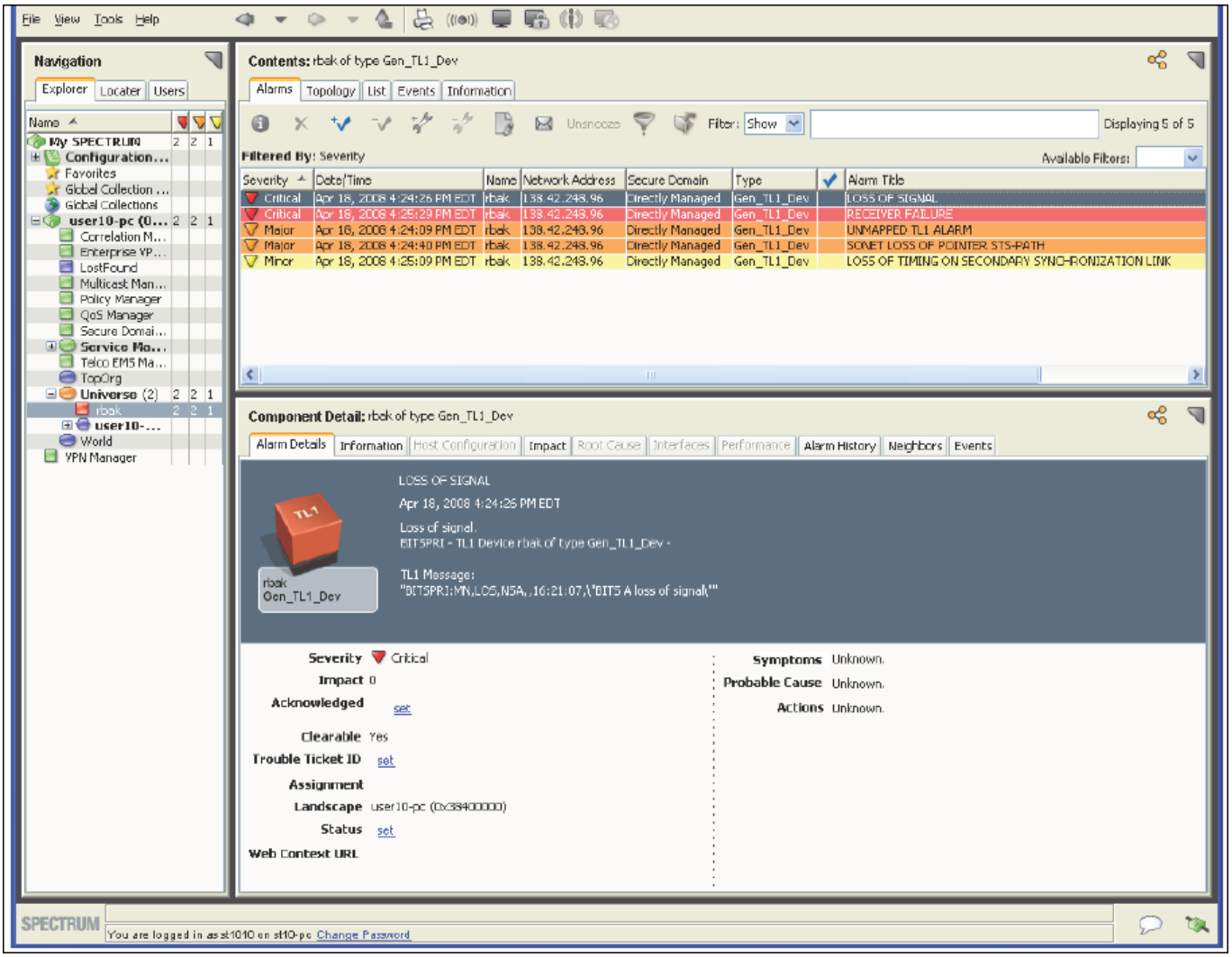
Another mapping file is also required to support the AlarmMap. The alert code that is specified in the TL1 AlarmMap is processed by a corresponding AlertMap file on the SpectroSERVER. If the AlertMap file is missing, TL1 alerts are not properly mapped to events. For more information, see AlertMaps for TL1.

An AlarmMap usually includes a default entry, which is used for any TL1 events or alarms that lack a specific mapping. That is, all unmapped TL1 events/alarms result in the same DX NetOps Spectrum alert that is specified in the default entry.

TL1 alarms are displayed in the following image:



TL1 details per alarms are displayed in the following image:



Format of the AlarmMap

The AlarmMap is a sequence of records, each comprising the following five comma-separated fields:

```
<type>,<condition>,<alert-code>,<clr-alert-code>,<process>
```

- **<type>**
  A text string specifying the entry type as either “ALM” (for alarm) or “EVT” (for event).
- **<condition>**
  A text string identifying a TL1 condition type (TL1 alarms) as specified in the Telcordia standards document GR-833-CORE. If this string is blank, or empty, the record serves as the default entry for the AlarmMap. If the string ends in a hyphen (e.g., OGCCS-), all conditions that start with that string are mapped by this entry. It resembles a “wildcard.”
- **<alert-code>**
  A hexadecimal literal with a leading “0x” that specifies the alert code to be used by DX NetOps Spectrum for this TL1 event/alarm.
- **<clr-alert-code>**

A hexadecimal literal with a leading “0x” that specifies the code that DX NetOps Spectrum uses to clear the TL1 event/alarm.

- **<process>**
  Is a boolean value (TRUE or FALSE) that specifies whether the event/alarm is processed. If TRUE, an alert is generated for DX NetOps Spectrum. If FALSE, the incoming TL1 event/alarm is discarded and nothing is forwarded to DX NetOps Spectrum.
  The following is an example of an AlarmMap:
  ALM, ,0x3d50001 ,0x3d50002 ,TRUE <------ DEFAULT
  ALM,LINETERM ,0x3d500f3 ,0x3d500f4 ,TRUE
  ALM,ECOI ,0x3d501b9 ,0x3d501ba ,TRUE
  ALM,SLOR ,0x3d501c9 ,0x3d501ca ,TRUE
  EVT,NTYDGSP ,0x3d5009d ,0x3d5009e ,TRUE
  The commas can be surrounded by spaces. Everything after the <process> field and a space is treated as comment for better readability.

The contents of the AlarmMap are critical to the performance of TL1 Gateway. Configure the AlarmMap such that only those TL1 alarms in which you are interested are processed. In other words, verify that the <process> value is set to FALSE for all TL1 alarms that you do not want to generate a DX NetOps Spectrum alert.

WARNING

If an “unwanted” TL1 event/alarm does not appear in the map, it is still processed through the default mapping, assuming that the map has a default entry. To prevent this extra processing, you can take one of the following steps:

- Include all unwanted TL1 events/alarms in the map with their <*process*> values set to FALSE.
- Remove the default entry.

The tl1map Utility

TL1 Gateway includes a command-line tool that is called tl1map that lets you inspect the AlarmMap and perform either of the following two operations:

- Extract an AlarmMap from either a TL1 device model or from the TL1 device model type (Gen_TL1_Dev).
- Import an AlarmMap in the form of a text file either to a TL1 device model or to the Gen_TL1_Dev model type.

The tl1map tool is located in your top-level DX NetOps Spectrum directory's TL1Apps subdirectory. You can run tl1map either from the shell command line or from within scripts (if you need to perform more complicated AlarmMap manipulations). In either case, tl1map is a SpectroSERVER client, so SpectroSERVER must be up and running.

NOTE
The tl1map utility is dedicated to models/model types of the Gen_TL1_Dev hierarchy only. It is not intended to be used with any other models/model types and could cause potential data corruption if not used properly.

Syntax

```
.tl1map {load | dump} -f mfile {-t | -m} handle [host]
```

- **dump**
  Extracts the internal AlarmMap in text format and saves it to the file specified by the -f option, in text format.
- **load**
  Processes the input file that is specified by the -f option and writes an AlarmMap to the corresponding attribute.
- **-f mfile**
  Indicates that the next entry will provide the name of a particular map file (mfile). In load mode, the supplied input file is expected to have AlarmMap syntax, and it is converted into an internal TL1 AlarmMap object. When in dump mode, the contents of the already existing, internal AlarmMap is extracted, formatted, and written to the specified mfile. This

file is then suitable for visual inspection and processing by a text editor, so it can be changed, and re-imported using the load option.

- **-t**
  Specifies that the handle provided is the model type handle of the TL1 device model type.
- **-m**
  Specifies that the handle provided will the model handle of a TL1 device model.
- **handle**
  Specifies the handle of either the TL1 device model type (if the -t flag is used) or the TL1 device model (if the -m flag is used).
  Limits: Hexadecimal format
- **host**
  The name of the DX NetOps Spectrum host. This is only needed when accessing a remote host, or if the Unix hostname command reports a mixed-case name, while the “true” host name was advertised in lower-case format. In any case, tl1map should be run on the local host.

NOTE
The -f, -t, and -m -f options can appear in any order.

AlarmMap Extraction

If tl1map is invoked with the dump option, it will read the current AlarmMap attribute value from either a model (-m option), or a model-type (-t option), convert it into a human-readable format, and write it to the file specified by the -f option. That file will also contain an informational header, which includes things like model name, model type name, model handle, model type handle, host name, and the current time stamp (based on the machine where the utility was invoked). The header is formatted as a comment, so it will not affect any subsequent import operation using this machine-generated file.

Typically, you would examine this AlarmMap file with a text editor, make any desired modifications, and then import the file again by invoking tl1map with the load option.

NOTE
The entries in the output file are in no particular order. If you want to compare two different AlarmMaps, you need to do the following:

- Strip off the comment header.
- Sort the files.
- Perform the actual comparison.

Importing an AlarmMap

To import an AlarmMap under any of the following scenarios, use the tl1map utility with the load option:

- Creating an initial AlarmMap based on documents from a TL1 device vendor or on TL1 standards documents.
- Extracting a previously stored AlarmMap, applying some changes, and importing the changed version again.
- Extracting the AlarmMap from one model/model type, and importing it to another model/model type (possibly on another host)

When importing an initial AlarmMap from a file, you have to make sure that the file conforms with the format specifications for AlarmMap files.

The tl1map utility displays limited diagnostic messages if it encounters illegal entries in the AlarmMap file. If there were any errors, no data is written to the model/model type attribute.

Examples of tl1map Usage

The following examples illustrate how the tl1map utility might be used to perform various specific tasks. Note that the examples use /dev/stdin and /dev/stdout. Since these are not available on NT, you would therefore use intermediate files instead of Unix-style pipes.

To import an AlarmMap to model 0x146007

```
tl1map load -f custom.amap -m 0x146007
```

To export the AlarmMap to a file

Exporting the AlarmMap to a file enables it to be modified. The modified file can be reloaded/imported.

```
tl1map dump -f exp.amap -t 0x4010000
```

...then, after you edit and save exp.amap ...

```
tl1map load -f exp.amap -t 0x4010000
```

To copy the AlarmMap from one model to another

```
tl1map dump -m0x1503fcc -f/dev/stdout |
tl1map load -m0x1504003 -f/dev/stdin
```

To delete the “LINETERM” entry from the AlarmMap

```
tl1map dump -m0x1504003 -f/dev/stdout |
sed '/LINETERM/d' |
tl1map load -m0x1504003 -f/dev/stdin
```

To copy a model-based map from one host to a model type-based map on another host:

```
tl1map dump -m0x1504003 -f/dev/stdout hosta |
tl1map load -t0x25040cc -f/dev/stdin hostb
```

The following example shows a shell script that could be used to implement a simple AlarmMap editor:

```
#! /bin/ksh
#
# A simple AlarmMap editor
#
# Syntax:  tl1edit { -m | -f } handle [ host ]
#

# Your favorite text editor
if [[ -z $EDITOR ]] ; then
EDITOR=vi
fi

# Generate a unique, temp file name
MF=`date '+%Y%m%d%H%M%S.almap'`

# Dump the AlarmMap to that temp file
tl1map dump -f $MF $*

# Make a backup copy
cp $MF $MF.orig

# Invoke your favorite editor on that file
```

```
$EDITOR $MF

# If there are no changes, remove temp mapfiles & exit
diff $MF $MF.orig > /dev/null && rm -rf $MF $MF.orig && exit

# If there were changes, load them into the attribute
tl1map load -f $MF $*
```

AlertMaps for TL1

The TL1 AlarmMap converts a TL1 autonomous message into an alert (not an event) that is processed by DX NetOps Spectrum. DX NetOps Spectrum handles these alerts similarly to SNMP traps. Configure the AlertMap with the alert code that is specified in the TL1 Alarm Map.

NOTE
The alert code that is specified in the AlertMap is a decimal number rather than an SNMP trap OID.

Each alert from the TL1 Gateway offers the following standard, unchangeable varbinds:

- Varbind 1: AID
- Varbind 2: Severity
- Varbind 3: Condition
- Varbind 4: Alarm Message
- Varbind 5: Alert Code (as specified in the TL1 AlarmMap)

The AlertMap must map these alert variables to event variables, one for one.

A supported entry in the AlertMap resembles the following syntax:

```
64352257 0x3d5f001 1(1,0) 2(2,0) 3(3,0) 4(4,0) 5(5,0)
```

The previously mentioned entry maps alert code 64352257 (0x3d5f001 in decimal) to event code 0x3d5f001, and copies each of the five alert variables to the event.

For more information about the AlertMap, EventDisp, CsEvFormat, and CsPCause files, see Event Configuration .

Managing TL1 Gateway Daemon

TL1 Gateway Daemon

A TL1 Gateway daemon (TL1d) must be up and running in order to translate alarms from the modeled TL1 device into DX NetOps Spectrum events and alarms. The daemon is located in your top-level DX NetOps Spectrum directory's TL1Apps subdirectory. It has the following syntax:

```
TL1d [-p gw-port]
```

- **-p gw-port**
  Specifies that a TL1 Gateway port (gw-port) other than the one specified at model creation will be designated for communication between the daemon and SpectroSERVER.
  Default: 64222

  NOTE
  CA recommends that you *not* use the value 65535, as it is likely to conflict with other applications.

Start the TL1 Daemon on a DX NetOps Spectrum Server

Start the TL1 daemon to start handling alarms from TL1 devices.

This procedure varies only slightly (see Step 3) for the Windows platforms.

Follow these steps:

1. Launch a command shell window.
2. Navigate to the TL1Apps subdirectory in your top-level DX NetOps Spectrum directory.
3. Enter the command:

   ```
   TL1d
   ```

   or, if you want to use a different gateway port...

   ```
   TL1d  -p <your-gw-port number>
   ```

   As soon as you see “@(#)” followed by the actual port number, the daemon is running. You can minimize the command shell window.

Start the TL1 Daemon on a Remote Server

It is recommended to know the IP address of the server that hosts the CORBA osagent in your environment. This server is most likely the same computer where DX NetOps Spectrum is running.

Take a few steps to start the TL1 Daemon on a remote server on Windows.

Follow these steps:

1. Launch a command shell window.
2. Navigate to the directory where your TL1d.exe file resides.
3. Enter one of the following commands:

   ```
   TL1d
   ```

   Or, to use a different gateway port, enter the following command:

   ```
   TL1d -p <your-gw-port number>
   ```

   As soon as you see “@(#)” followed by the actual port number, the daemon is running. You can minimize the command shell window.

Terminating the TL1 Daemon

The specific command or procedure for gracefully terminating the TL1 daemon depends on the platform the daemon is running on.

- On NT, bring up the Task Manager, select the TL1d process, and then terminate it.

Customize the Tomcat Log Path

From 10.4.2.1, you can customize the OneClick WebServer Tomcat log path to a non-default location.

Follow these steps:

Windows:

1. Open the **OneClickService.conf** file from the `%SPECROOT%/tomcat/bin/` directory.
2. Add a new `jvm_opt` property and set the property value as the new path for tomcat log file.

   ```
   jvm_opt=-DCATALINA_OUT=C:\tomcat-log\stdout.log
   ```

3. Modify the `log_file` path to new log location.

   ```
   log_file=C:\tomcat-log\stdout.log
   ```

4. Save and close the file.
5. Restart the Tomcat.

Linux:

1. Update the `catalina.sh` file located at the `$SPECROOT/tomcat/bin/` directory.
2. Add `CATALINA_OUT` property to the `JAVA_OPTS` property section.

```
JAVA_OPTS="-DOneClick -server -Xmx10000M -XX:+HeapDumpOnOutOfMemoryError -Djava.awt.headless=true
 -Djavax.net.ssl.trustStore=$SPECROOT/custom/keystore/cacerts -Dfile.encoding=UTF-8 -
Dcom.sun.management.jmxremote -Dorg.apache.coyote.USE_CUSTOM_STATUS_MSG_IN_HEADER=true -DCATALINA_OUT=/
var/log/tomcat/catalina.out"
```

3. Set the CATALINA_OUT property value as the new path for tomcat log file.

```
CATALINA_OUT="/var/log/tomcat/catalina.out"
```

4. Save and close the file.
5. Restart the Tomcat.

The Tomcat log file will be generated in the new location post this change.

Integrating

This section contains information about integrating DX NetOps Spectrum with other CA products.

Integration with UIM

NOTE
Previously, Unified Infrastructure Management (UIM)-DX NetOps Spectrum integration sync status was not getting updated for hosts and VMware in the OneClick view. Now, in 10.4.1, the integration sync status is updating correctly under the **UIM Host Sync Configuration** section and **UIM Virtualization Sync Configuration** section (**Navigation** view, **UIM Manager**, **Information**, **Configuration**), as appropriate. For more information, see the related section UIM-DX NetOps Spectrum Sync Status in this article.

WARNING

There are major changes to the GUI for configuring the integration from 10.2.3 and spectrumgtw probe v8.67. Refer to the spectrumgtw AC documentation for configuring the integration.

NOTE
From the 10.1.2 release, the SNMP Gateway probe and SBGW / Southbound gateway are no longer recommended for alarms synchronization from UIM to DX NetOps Spectrum. CA recommends using the Spectrum Gateway (spectrumgtw) probe for alarm synchronization from this release.
We recommend that you go through the DX NetOps Spectrum-UIM Integration using spectrumgtw probe - FACT SHEET to understand the changes to the integration whether you are new to this integration or an existing user.
For more information about the integration of DX NetOps Spectrum with other CA products, see Integration Compatibility.
We recommend you deploy the latest hotfix that is available for the probes.

NOTE

In previous releases of UIM, the nisapi_service_host package was deployed to the service_host probe. As of UIM release 8.47, the service_host probe was deprecated and its functionality was moved to the wasp probe. This change has the following implications for your DX NetOps Spectrum integration: You no longer have to deploy nisapi separately. By default, your DX NetOps Spectrum integration uses the nisapi package deployed in wasp as part of a UIM installation or upgrade.

- By default, the wasp probe uses port 80. However, because the wasp probe is a core component in UIM, this port should already be open in a properly functioning UIM environment. For more information, refer to the wasp probe documentation.

Overview

The DX NetOps Spectrum - Unified Infrastructure Management (UIM) (formerly known as CA Nimsoft or CA Nimsoft Monitor) Integration expands the DX NetOps Spectrum monitoring capabilities of the infrastructure with information and alarms from UIM and provides the following benefits:

- Provides a holistic view of the availability of host servers and VMware environment across the network and their performance data for fault management in a single pane of the application. It also provides end to end root cause and impact analysis across network and server elements, extending DX NetOps Spectrum core capabilities to other infrastructure domains.
- Advance condition correlation between UIM and DX NetOps Spectrum helps in building robust fault management.
- Leverage UIM capabilities for server management and use the DX NetOps Spectrum network management capabilities for an end to end infrastructure management.

The following video describes how you can integrate DX NetOps Spectrum and UIM to monitor your infrastructure and to use DX NetOps Spectrum root cause analysis feature to rapidly troubleshoot and resolve issues:

- Integrating DX NetOps Spectrum and UIM through the Web Server for Server Management
- Integrate DX NetOps Spectrum and UIM for Virtualization Management

DX NetOps Spectrum and UIM bidirectional Integration

You can enable bidirectional integration between CA Spectrum r10.3 and CA UIM r8.5.1 using the **spectrumgtw** probe v8.67.

This integration enables reception of all UIM alarms by DX NetOps Spectrum and reception of DX NetOps Spectrum inventory information and all DX NetOps Spectrum alarms by UIM.

For more on the bidirectional integration see DX NetOps Spectrum and UIM Bidirectional Integration.

Please see the following video to understand more about the bidirectional integration and its capabilities:

Enhancements and Improvements to the DX NetOps Spectrum and UIM Integration

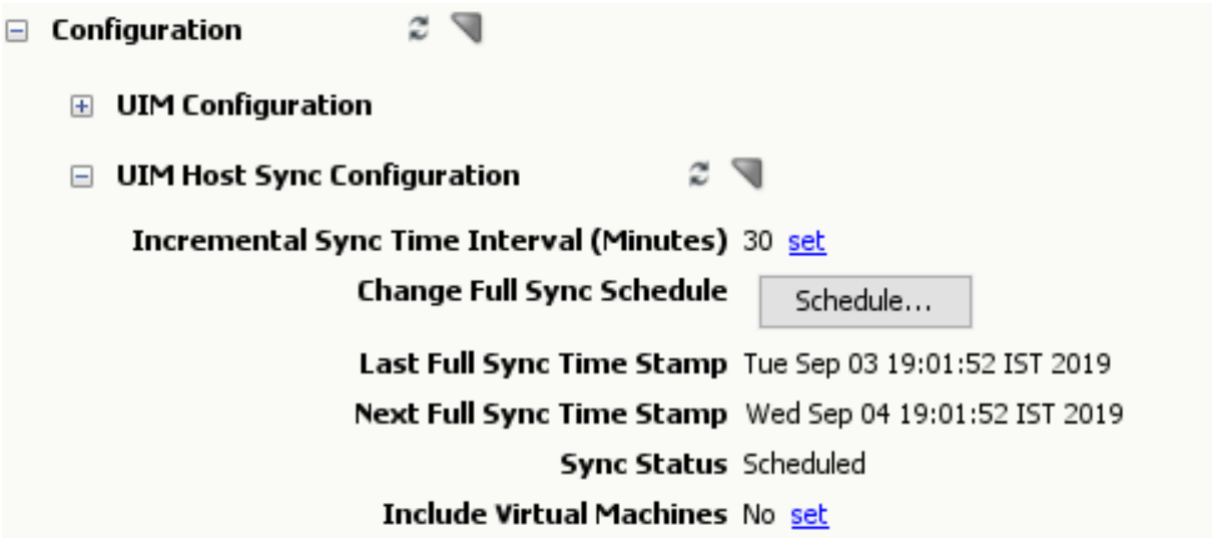
This section includes enhancements and improvements made to the integration.

UIM-DX NetOps Spectrum Sync Status

Previously, UIM-DX NetOps Spectrum integration sync status was not getting updated for hosts and VMware in the OneClick view. Now, in 10.4.1, the integration sync status is updating correctly under the **UIM Host Sync Configuration** section and **UIM Virtualization Sync Configuration** section (**Navigation** view, **UIM Manager, Information, Configuration**), as appropriate.

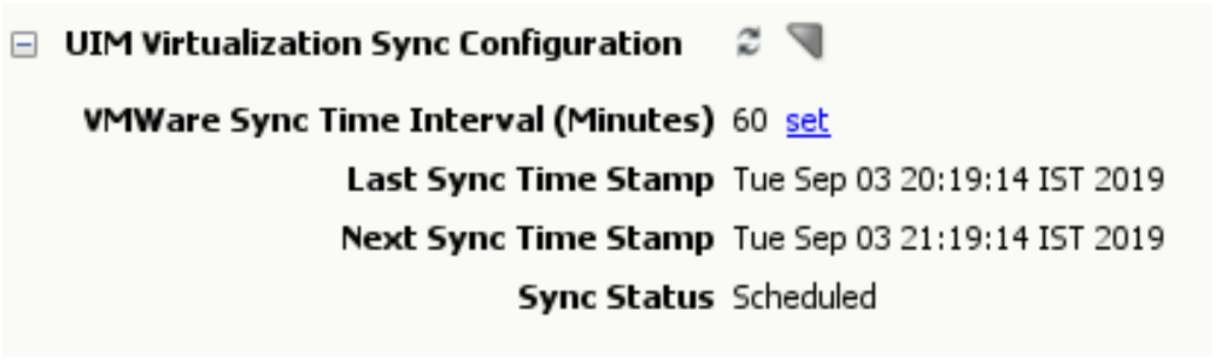
- Under the **UIM Host Sync Configuration** section, the sync status in the following parameters now gets updated correctly:
  - Last Full Sync Time Stamp
  - Next Full Sync Time Stamp
  - Sync Status

  The following screenshot shows the required information:



- Under the UIM Virtualization Sync Configuration section, the sync status in the following parameters now gets updated correctly:
  - Last Sync Time Stamp
  - Next Sync Time Stamp
  - Sync Status

  The following screenshot shows the required information:



GUI Changes to the UIM Configuration page in DX NetOps Spectrum

From the current release of DX NetOps Spectrum, major GUI changes are done to the UIM configuration section in the DX NetOps Spectrum Administrations page. You can choose the integration option to configure it from DX NetOps Spectrum or UIM's spectrumgtw probe. For more details, see the Changes in Integration Architecture section in the Integration Architecture page.

Following are the available configuration options:

- **No Integration** - Select this option to disable DX NetOps Spectrum-UIM integration.
- **Legacy Integration** - Select this option if you are using the spectrumgtw probe v8.67 with CA Spectrum 10.2.3
- **SpectrumGateway Integration** - Select this option if you are using the spectrumgtw probe v8.65. The Spectrum-UIM integration is configured through UIM's spectrumgtw probe.

Configuring Integration through UIM's spectrumgtw Probe v8.65

If you are using UIM's spectrumgtw probe (v8.65) and CA Spectrum r10.2.3, the integration between DX NetOps Spectrum and UIM is configured through UIM's spectrumgtw probe (v8.65) Admin Console. Select the 'SpectrumGateway

Integration' option in the DX NetOps Spectrum OneClick Admin pages > UIM Configuration page. Click 'Save' then go to UIM's spectrumgtw probe (v8.65) Admin Console to complete the configuration. For configuration instructions, see the Deploy and Configure the Probe section, in the spectrumgtw probe documentation.

NOTE

Before switching the existing integration from **Legacy Integration** to **SpectrumGateway Integration**, ensure that you have installed the patch **10.02.03.PTF_10.2.303** on top of 10.2.3. This patch is a solution to retain the UIMEventAdmin Models after moving integration from Legacy Integration to SpectrumGateway Integration.

VHM Enhancements

From 10.2.3, DX NetOps Spectrum can monitor both UIM models and SystemEdge models when DX NetOps Spectrum-UIM Integration is enabled with Multi-tenancy.

When the integration is enabled with Multi-tenancy, UIM origins are mapped to DX NetOps Spectrum landscapes. In this scenario, DX NetOps Spectrum monitors UIM models on the landscapes which are part of Multi-tenancy mapping and can monitor SystemEdge AIM based models on the landscapes which are not part of Multi-tenancy mapping.

When the DX NetOps Spectrum-UIM Integration is enabled without Multi-tenancy, DX NetOps Spectrum monitors UIM models only. All the SystemEdge AIM based models are deleted on all the landscapes.

vMotion Support Enhancements

From r10.2.3, when a VM is migrated (vMotion) in VMware, then the spectrumgtw probe picks the changes and sends it to DX NetOps Spectrum. The changes are shown in the DX NetOps Spectrum hierarchy. The reflection time for the changes in the DX NetOps Spectrum hierarchy is based on the frequency at which the UIM VMware probe publishes the changes to UIM. This functionality is supported for both legacy integration and integration through spectrumgtw probe 8.65 version.

NOTE
After deploying and starting the spectrumgtw 8.6.5, you must restart the UIM VMware probe.

Reconciliation Enhancements

In earlier releases, when DX NetOps Spectrum models UIM host Server if the network devices such as routers or switches have the same IP address then DX NetOps Spectrum reconciles based on IP address and network devices are displayed as part of UIM Manager Hierarchy.

Starting from 10.2.3, for the same scenario, DX NetOps Spectrum will create UIM Host Server/UIM VM models in the UIM Manager Hierarchy and stop reconciling with the network devices with IP Address.

Support for Customized Alarms

From r10.2.3, you can manage customization (changing the severity) for UIM alarms in DX NetOps Spectrum. The spectrumgtw probe 8.65 version is enhanced to allow the Spectrum users for this customization. For more information, see the Support for Customized Alarms section in the spectrumgtw AC Configuration documentation.

Maintenance Schedule Synchronization

From r10.2.3, the Maintenance Mode schedules on devices from DX NetOps Spectrum will be synchronized to UIM and from UIM to DX NetOps Spectrum, using the integration via spectrumgtw probe v8.65. For more information, see the Schedule Maintenance Mode section.

Multi-Tenant support in DX NetOps Spectrum

From r10.2.2, DX NetOps Spectrum supports multi-tenancy by leveraging the multi-tenant capability of UIM through the bidirectional DX NetOps Spectrum-UIM integration enabled by the Spectrum Gateway (spectrumgtw) probe. The Multi-

tenancy model in UIM is based on the Origin tag (Ownership). Using REST calls provided by DX NetOps Spectrum the spectrumgtw probe leverages this origin information. The spectrumgtw probe allows DX NetOps Spectrum to map Origin with Landscape. For more information on multi-tenancy in DX NetOps Spectrum see Multi-Tenant support in DX NetOps Spectrum.

Filter and Synchronize Specific Alarms

The DX NetOps Spectrum and UIM integration support filtering and synchronizing specific alarms from Spectrum to UIM or UIM to Spectrum. The alarm filter solution reduces clutter and eliminates the sync of alarms that you do not want to monitor. After applying the alarm filters you can see only the filtered alarms in Spectrum or UIM. You can configure the alarm filtering in the Spectrum Gateway (spectrumgtw) probe. For more information, see Synchronize Specific Alarms.

Selective Inventory Synchronization

From r10.2.1, the DX NetOps Spectrum and UIM integration support selective synchronization of Server Management entities only. This functionality allows you to avoid full inventory synchronization and select only the server inventory which you want to get synced from UIM to Spectrum. For more information, see 'Selective Inventory Synchronization' section in the Integrating DX NetOps Spectrum and UIM through the Web Server for Server Management page.

Custom Attributes Synchronization

You can synchronize custom attributes from UIM to DX NetOps Spectrum. When you enable the DX NetOps Spectrum-UIM Bidirectional Management integration, the following custom attributes are synced from UIM to DX NetOps Spectrum. For more information see, custom attributes section.

Supporting AWS (Amazon Web Services) Cloud Monitoring

AWS Cloud (Amazon Web Services) Monitoring is supported by DX NetOps Spectrum using the UIM and DX NetOps Spectrum integration. The aws (Amazon Web Services) The monitoring probe deployed in UIM enables the metric data collection from the AWS instances. This data is then synchronized from UIM to DX NetOps Spectrum. For more information, see Supporting AWS (Amazon Web Services) Monitoring.

Supporting Azure (Microsoft Azure Monitoring)

Microsoft Azure Monitoring is supported by DX NetOps Spectrum using the UIM and DX NetOps Spectrum integration. The azure (Microsoft Azure Monitoring) probe deployed in UIM enables the metric data collection from the AWS instances. This data is then synchronized from UIM to DX NetOps Spectrum. For more information, see Supporting azure (Microsoft Azure Monitoring).

ESX Host Server Maintenance Mode enhancement

From DX NetOps Spectrum 10.2.1, you can now choose whether to retain the default correlation between the ESX host and the VMs during Maintenance mode. By default, all the VMs under the ESX Host are automatically placed in maintenance mode. You can now choose to set only the parent ESX host server in maintenance mode and not the child virtual machines that are hosted on the ESX server. For more information see, Place ESX Host in Maintenance Mode or Hibernation.

SNMP profile Synchronization

From this release, SNMP profiles for DX NetOps Spectrum inventory in a Global Collection is synchronized by the Spectrum gateway (spectrumgtw) probe to SNMP Collector so that it can monitor those without re-entering SNMP credentials.

When inventory data is synchronized from DX NetOps Spectrum to UIM via spectrumgtw probe, the SNMP credentials are also synchronized and stored in the UIM database. The SNMP profiles synchronized from DX NetOps Spectrum

are shared with the Discovery server, and once the SNMP Collector probe provides the IP range relevant to the profiles synchronized, you no longer need to re-enter the SNMP credentials while logging in to the UIM interface.
The configuration of integration parameters is done in the Spectrum gateway configuration. This capability is only available for the integration between DX NetOps Spectrum 10.2.1 and CA UIM 8.5.1 using the spectrumgtw probe 8.6 versions.

SSL communication between Spectrum Gateway (spectrumgtw) probe and DX NetOps Spectrum

The DX NetOps Spectrum- UIM integration between DX NetOps Spectrum 10.2.1 and CA UIM 8.5.1 using Spectrum Gateway (spectrumgtw) probe 8.6, supports secure (https) communication the gateway probe and DX NetOps Spectrum. You need to ensure that the DX NetOps Spectrum OneClick Server is configured for SSL (HTTPs) and that https is selected as the communication protocol in the Spectrum Gateway (spectrumgtw) probe configuration.

For more information, see Configure OneClick for Secure Sockets Layer

DX NetOps Spectrum Fault-Tolerant Scenarios

Back-end changes have been made to DX NetOps Spectrum to prevent modeling new entities, reported by UIM, in a particular SpectroSERVER if it is a Secondary SpectroSERVER. However, updating the models and their relationships will still be done in secondary SpectroSERVER.

> **NOTE**
> While triggering DC Migration, ensure that both source and destination SpectroSERVER are running on the primary SpectroSERVER in the Fault-Tolerant environment.

Logging Improvements

Improved the OneClick Server logging for DX NetOps Spectrum-UIM integration to give insights into what has been modeled a what was not, movement of VMs across different ESX hosts, relationship additions and deletions and so on.

Reconciliation Enhancements

Till now DX NetOps Spectrum used to reconcile entities like Vcenters, VMs, and Hosts. DX NetOps Spectrum is able to reconcile logical entities like Datacenters, Clusters and Resource Pools with existing models of corresponding entity types if they were modeled previously to prevent duplication in error scenarios.

DX NetOps Spectrum will now reconcile ESX hosts with existing ESX hosts or UIMHostServer models only. If ESX host is reconciled with a UIM Host Server model then the existing UIMHostServer model will be deleted and a new ESX model will be created for that ESX host. This newly created model will be part of both Virtualization and Server hierarchies.

> **NOTE**
> If an ESX server is already modeled in DX NetOps Spectrum with a model type other than UIMVirtualMachineHost or UIMHostServer, the ESX Server is not reconciled as part of the DX NetOps Spectrum-UIM integration.

Out of box solution for managing VM as a server

You no longer need to deploy the **VMware probe** to synchronize VMware inventory. If VMs are being monitored as a server in UIM it can be pulled into DX NetOps Spectrum for Server management.

> **NOTE**
> The VMware probe allows you to monitor standalone ESX Host Servers and it's corresponding hierarchy in a UIM context. DX NetOps Spectrum does not support this. This is only supported if you are monitoring ESX Servers through VCenter using VMware probe.

Delete Unreported Entities in DX NetOps Spectrum (Virtualization Management)

In previous versions of the DX NetOps Spectrum-UIM integration, the virtual entities/ models that were not reported during a synchronization cycle were auto-deleted. From this release, these models will be placed in the Unreported Entities/ Models container, instead of being deleted.
See Delete Unreported Entities.

Integration Architecture

Changes in DX NetOps Spectrum 10.4.2

In the current release, DX NetOps Spectrum-DX Infrastructure Manager integration has the following changes:

Handling of Unreported Inventory

From 10.4.2, DX NetOps Spectrum identifies unreported inventory from DX Infrastructure Manager entities and deletes the unreported inventory if inventory does not exist in DX Infrastructure Manager. When the inventory exists in DX Infrastructure Manager after the sync, it retains the unreported inventory to verify after the next synchronization. By default, the attribute is enabled. If **deleteUnreportedEntities** attribute is enabled in DX NetOps Spectrum, DX NetOps Spectrum checks the presence of the unreported inventory models in DX Infrastructure Manager and deletes the models that are no longer present in DX Infrastructure Manager.

Follow these steps:

1. Log in to the OneClick WebApp.
2. On the **Locator** tab, navigate to **UIM Manager**.
3. In the **Components Detail** section, select **Attribute** tab.
4. Search **deleteUnreportedEntities** to view the current setting, enabled by default.
5. Right-click the **deleteUnreportedEntities** attribute and select **Edit**.
6. In the **Edit UIM Manager** popup, change the setting if required.
7. Select **OK** to apply the change.
   The **Attribute Edit Results** window shows the status.
8. Close the window, on the successful assignment of the attribute.

Changes in DX NetOps Spectrum 10.4.1

In the current release, DX NetOps Spectrum-UIM integration has the following changes:

Customize the HostSsync Payload Size in OneCLick

From 10.4.1, you can customize the maximum size of the payload that you want to receive from the spectrumgtw probe to DX NetOps Spectrum during the host server sync. Set the org.`apache.cxf.stax.maxInputSizeInMB` parameter to a desired value in the `Spectrum_HOME\tomcat\webapps\spectrum\WEB-INF\web.xml` file.
Default: 8 MB

Include Virtual Machines

When you select the **Include Virtual Machines** checkbox, DX NetOps Spectrum syncs only the non-VMWare VMs

> **NOTE**
> From the spectrumgtw probe 8.68, the vmware probe-managed VMs are not synchronized as part of the host server sync.

Disable the Host Server Integration

When you disable the **Host Server Integration**, the sync stops and deletes the existing entities from DX NetOps Spectrum. You can change the retaining of the models in DX NetOps Spectrum by using the **RetainModelsAndRelations** attribute in the UIM Manager model.

The following are the supported integration modes and the resulting actions in DX NetOps Spectrum:

- DELETE_MODELS: Delete models from DX NetOps Spectrum. **Default value**
- DELETE_RELATIONS: Retain the models and delete the relationships of models.
- RETAIN_ALL: Keep both relations and models in DX NetOps Spectrum

Disable the VMWare Integration

When you disable the **VMWare Integration**, the sync stops and deletes the existing VMWare entities from DX NetOps Spectrum. You can change the retaining of the models in DX NetOps Spectrum by using the **RetainModelsAndRelations** attribute in the UIM Manager model.

The following are the supported integration modes and the resulting actions in DX NetOps Spectrum:

- DELETE_MODELS: Delete models from DX NetOps Spectrum. **Default value**
- DELETE_RELATIONS: Retain the models and delete the relationships of models.
- RETAIN_ALL: Keep both relations and models in DX NetOps Spectrum.

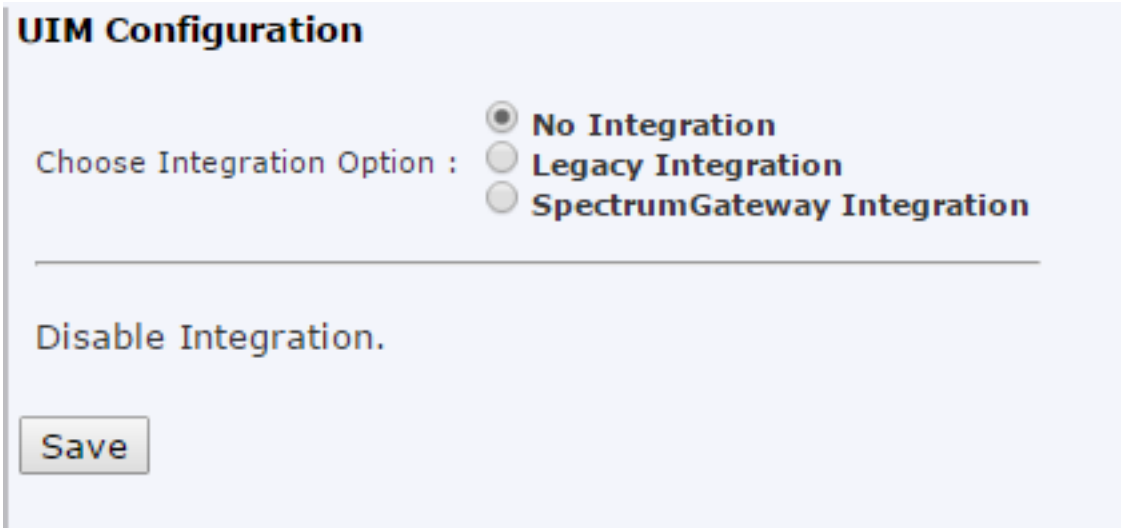
Changes in 10.2.3

From the 10.2.3 release, the DX NetOps Spectrum and UIM integration has undergone major changes. You can now configure the integration between DX NetOps Spectrum and UIM through either DX NetOps Spectrum or CA UIM's spectrumgtw probe (v8.65) Admin Console.

You can see the following options in the UIM Configuration page:

- **No Integration**
- **Legacy Integration**
- **SpectrumGateway Integration** - If you are using the spectrumgtw probe v8.65 or higher, select this option to configure the integration through spectrumgtw probe.

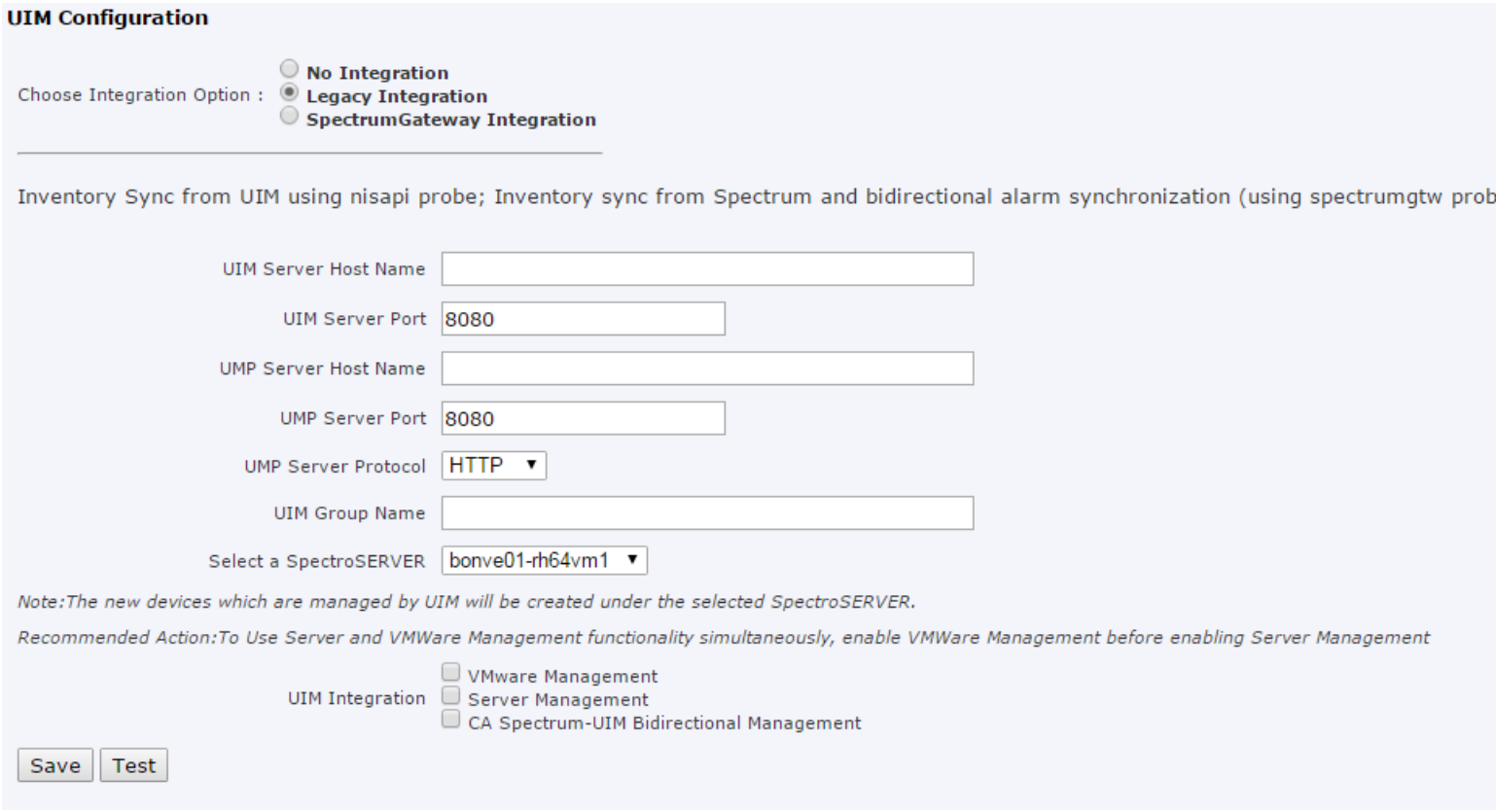
No Integration - Select this option button and click Save to disable the integration.



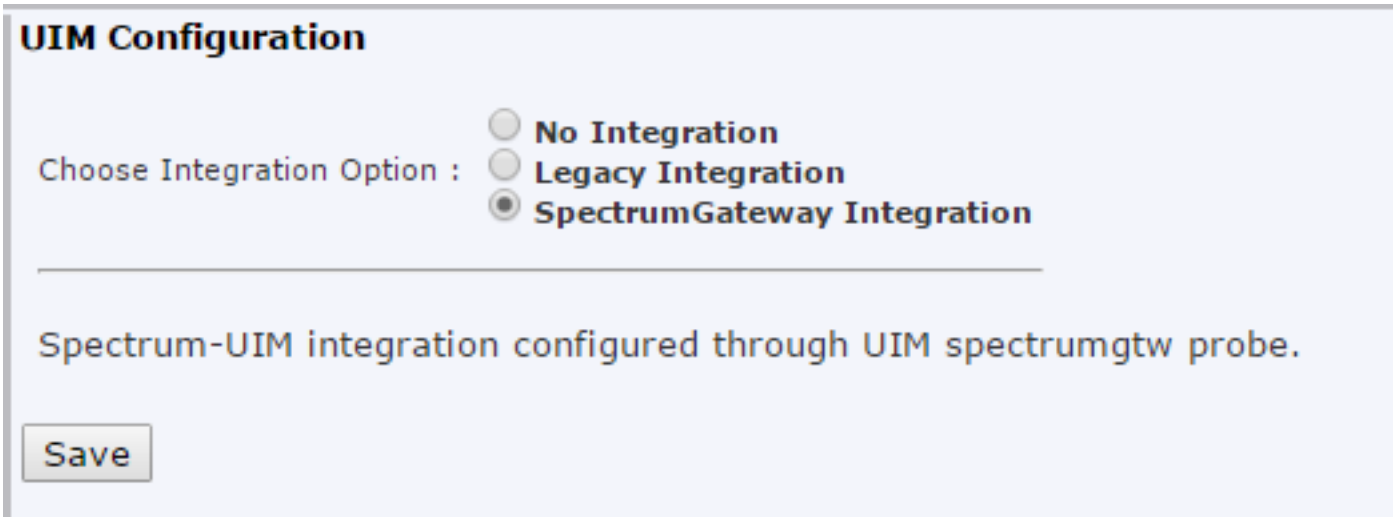
Legacy Integration - Select this option to configure the integration through DX NetOps Spectrum.

NOTE

The Legacy Integration option is used when you are using the spectrumgtw 8.64 for the integration. If the integration is already enabled before upgrading DX NetOps Spectrum, after the upgrade, the 'Legacy Integration' appears as selected by default.



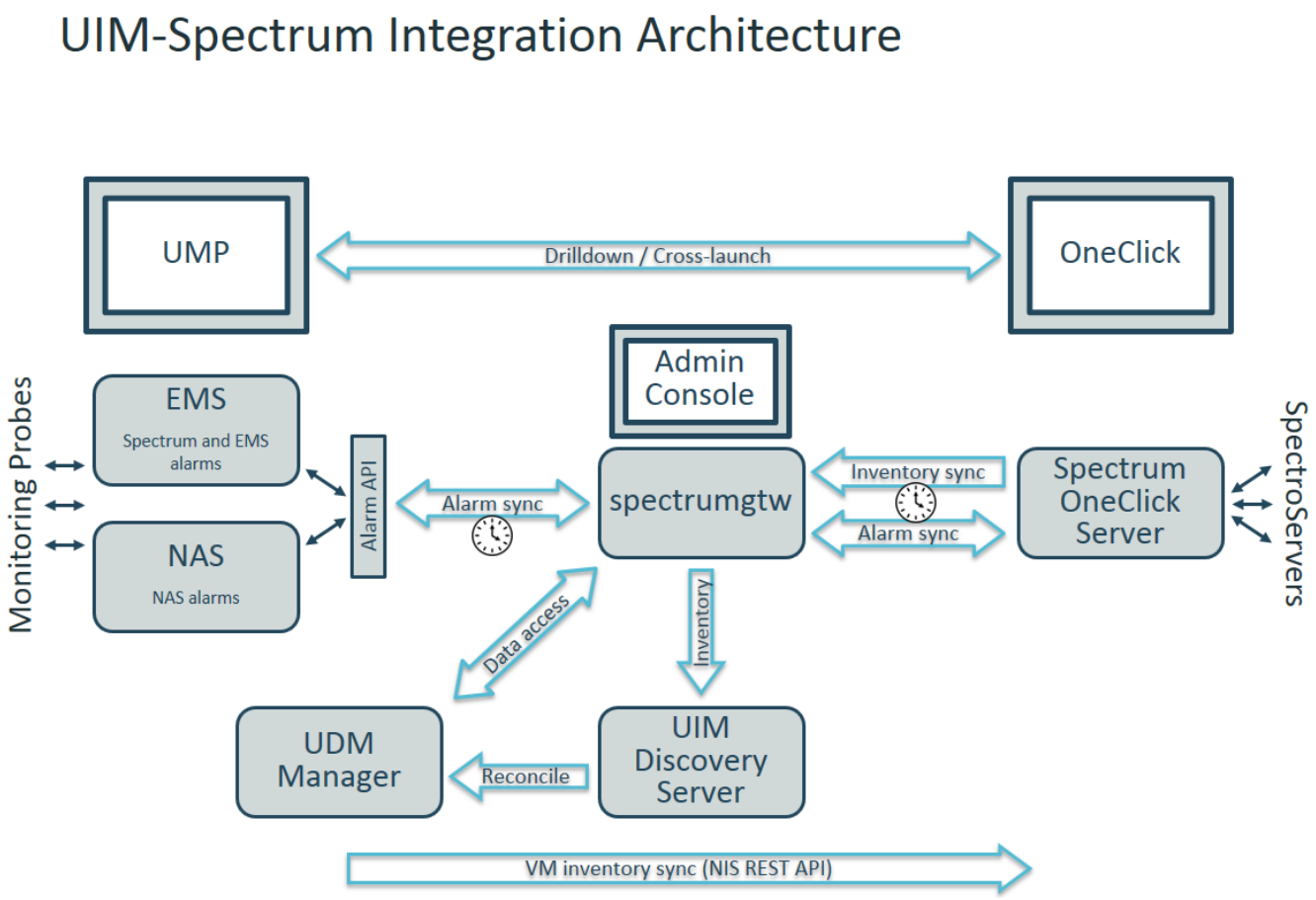
SpectrumGateway Integration - Select this option to configure the integration through UIM spectrumgtw probe v8.65. For instructions, see the Deploy and Configure the Probe section the spectrumgtw probe documentation.



The architecture for the DX NetOps Spectrum and UIM integration has undergone major changes from 10.1.2. We recommend you to go through the DX NetOps Spectrum-UIM Integration using spectrumgtw probe - FACT SHEET to understand the changes to the integration if you are new to this integration or an existing user.
Here are some key changes to the integration architecture:

- Alarms sent from UIM to DX NetOps Spectrum will no longer leverage the SNMP Gateway or DX NetOps Spectrum Southbound gateway. Instead, they will be routed through the new DX NetOps Spectrum Gateway Probe.
- Nisapi-service-host is replaced by nisapi-wasp for inventory sync from UIM to DX NetOps Spectrum
- Server Management
  - Has removed dependency on dedicated role
  - Added ability to monitor VM as a server without dependency on Vmware probe
  - Pulls inventory with one of the roles in the database as DatabaseServer, Host, VirtualMachine, VirtualMachineHost, WebServer, vCenter.
- Vmware Management
  - Handle Flapping inventory/relationships from UIM
  - Unreported inventory by UIM will be kept in a separate container in DX NetOps Spectrum and can be deleted by the user when no more needed

Here's an overview of the updated integration architecture:



Architectural Overview - using SBGW component /snmpgtw probes (till r10.1.1)

When an issue occurs in the infrastructure, alert data is sent from UIM to the SpectroSERVER of DX NetOps Spectrum through the Southbound Gateway component. SpectroSERVER is a primary server for DX NetOps Spectrum. For more information, see the SpectroSERVER and DX NetOps Spectrum Databases section.

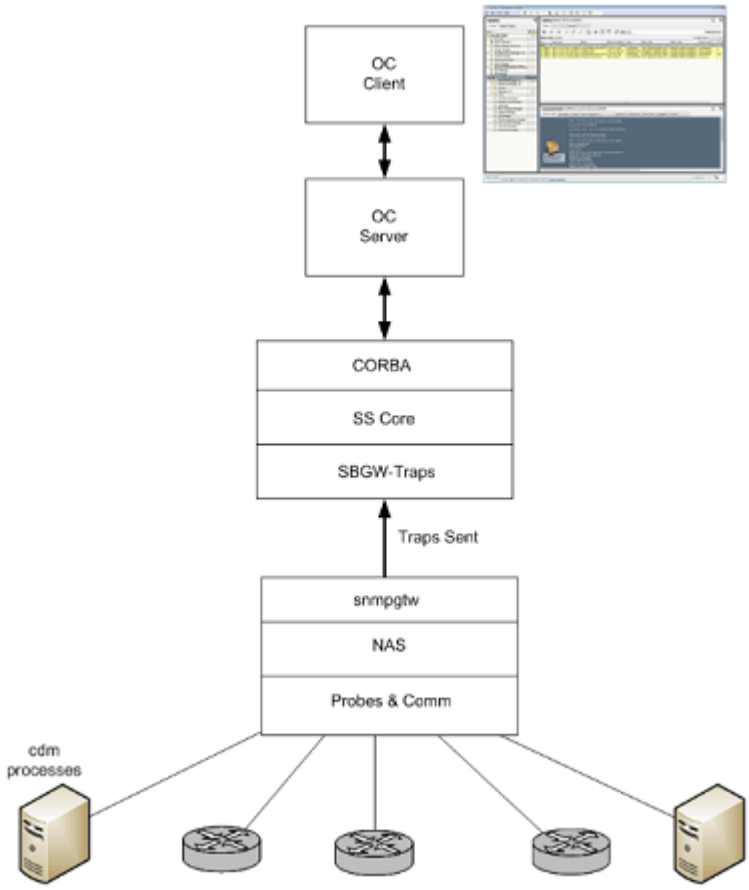
Using the Southbound Gateway, you can centralize network management, allowing DX NetOps Spectrum to capture and display data. Alert data is organized into DX NetOps Spectrum event and alarm data as appropriate and is displayed within OneClick.

The Southbound Gateway can be used with any incoming alert data stream format. The Southbound Gateway provides a simple, non-programmatic integration point for systems that can generate SNMP traps. It is also useful for managing non-SNMP environments. Southbound Gateway supplies an import tool that accepts XML-formatted alert data in case the system with which you are integrating cannot generate SNMP traps. For more information, see the Southbound Gateway Toolkit section.

Once the Southbound Gateway receives the alert data, the data is mapped to a DX NetOps Spectrum event in an AlertMap file. The Southbound Gateway determines the appropriate EventAdmin model to receive the alert data based on the IP address of the host computer that is sending the data. The IP address of the host computer should match the IP address that is used to create the EventAdmin model.

The DX NetOps Spectrum EventAdmin model receives the trap and translates it into a DX NetOps Spectrum event. If the event corresponds to a critical, major, or minor condition, the corresponding alarm is raised on a DX NetOps Spectrum model. The model where the alarm is raised depends on a few factors. We recommend having a previously modeled device in DX NetOps Spectrum. If the device model is present in DX NetOps Spectrum, the alarm is asserted against the existing device model. If the device model does not exist in DX NetOps Spectrum the alarm is asserted against an auto-created EventModel of the Nimsoft Robot that is reporting the condition.

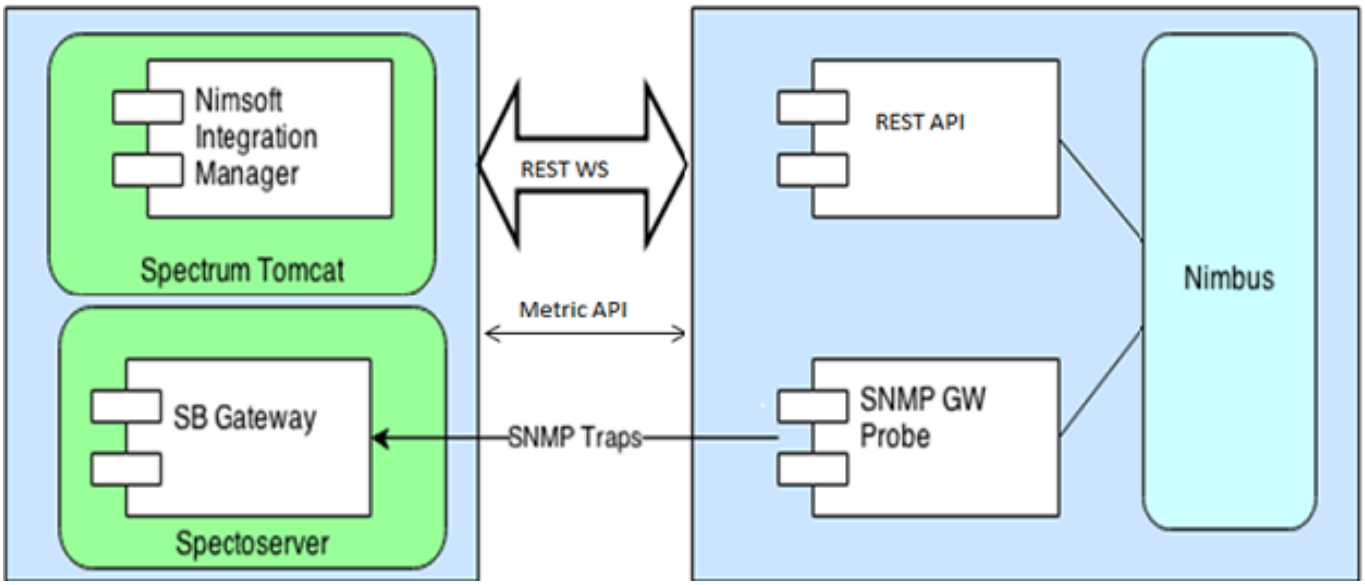
The following diagram illustrates an overview of the UIM - DX NetOps Spectrum Integration Architecture, till 10.1.1:



The integration between DX NetOps Spectrum and UIM through Web Server is designed to support server management. The HostServer models resulting from the integration provides traditional DX NetOps Spectrum capabilities such as layer2 connectivity and fault isolation with the features supported by UIM. You can use the Launch-in-Context feature to launch the CA Unified Management Portal (UMP) view from the server models in DX NetOps Spectrum to view the server information.

When the UIM and DX NetOps Spectrum integration is enabled from the OneClick Administration page, DX NetOps Spectrum receives the data from UIM through the Rest API. Once we get the data from the Rest API, UIM hosts are modeled in DX NetOps Spectrum. The alert data is sent from UIM to the SpectroSERVER of DX NetOps Spectrum through the Southbound Gateway component. The received data is mapped to a DX NetOps Spectrum event. The Southbound Gateway determines the appropriate EventAdmin model to forward the event. Metric violation traps that are coming from UIM are asserted on the respective Nimsoft host models in DX NetOps Spectrum.

The following diagram illustrates the architecture of the UIM - DX NetOps Spectrum Integration for Server Management through the Web Server (till v10.1.1):



- **UIM Probes**
  Provide the intelligence to manage specific components on a managed device. For example, the CDM processes probe is responsible for monitoring CPU, disk, and memory usage on target hosts. Over 135 UIM probes are available, to let you manage the entire IT infrastructure, including servers, network devices, applications, and databases.
- **Nimsoft Alarm Server (NAS)**
  Receives and manages incoming alarm messages. The Nimsoft Alarm Server supports message suppression and provides clients with services such as event updates, message filtering, automated actions, and mirroring capabilities.
- **SNMP Gateway Probe (snmpgtw)**
  Sends out the traps from UIM to DX NetOps Spectrum. This probe converts alarms to SNMP-Trap messages which are readable by any SNMP-based management system. It subscribes to UIM internal alarms and processes these alarms into SNMP traps with all the information about the alarm that is encoded in the trap varbinds.

Bidirectional Integration

This page describes the DX NetOps Spectrum and CA UIM Bidirectional integration by covering the following topics:

> **WARNING**
>
> If you are setting up the integration for the first time using CA Spectrum 10.2.3 and CA UIM 8.5.1/ 8.5.1 (SP1) using the spectrumgtw probe v8.65; you can no longer enable the bidirectional integration from the DX NetOps Spectrum OneClick Administration > UIM Configuration page. You need to first select **SpectrumGateway Integration** from the OneClick Admin > UIM Configuration page, enable the Alarm Integration checkbox and then configure additional parameters in the Alarm Configuration section of the spectrumgtw Admin Console, to initiate the bidirectional alarm integration.
> For more information, see spectrumgtw Admin Console.

Integration Compatibility

For more information about the integration of DX NetOps Spectrum with other CA products, see Integration Compatibility.

NOTE

In previous releases of CA UIM, the nisapi_service_host package was deployed to the service_host probe. As of CA UIM release 8.47, the service_host probe was deprecated and its functionality was moved to the wasp probe.

- By default, the wasp probe uses port 80. However, because the wasp probe is a core component in CA UIM, this port should already be open in a properly functioning CA UIM environment. For more information, refer to the wasp probe documentation.

Overview

You can enable bidirectional integration between CA Spectrum r10.3 and CA UIM 8.5.1 using the spectrumgtw probe v8.66. This bidirectional integration enables the following functionalities:

- Reception of all CA UIM alarms by DX NetOps Spectrum.
  This functionality enables DX NetOps Spectrum users to update (acknowledge, unacknowledge, clear, assign and unassign troubleshooter, and create service desk ticket) CA UIM alarms in OneClick. CA UIM receives such updates when its alarms are updated in OneClick.
- Reception of DX NetOps Spectrum inventory information and all DX NetOps Spectrum alarms by CA UIM.
  This functionality enables:
  - CA UIM to maintain an updated DX NetOps Spectrum inventory information based on a specific global collection in OneClick.
  - CA UIM to receive all DX NetOps Spectrum alarms. As a result, CA UIM users can update (acknowledge, unacknowledge, clear, assign and unassign troubleshooter, and create service desk ticket) DX NetOps Spectrum alarms. DX NetOps Spectrum receives such updates when its alarms are updated in CA UIM.

NOTE

- If you were using the uni-directional integration, you should turn off the associated profile in the snmpgtw probe before you start using the spectrumgtw probe for bidirectional integration.
- Ensure that the snmpgtw probe is not configured with the SpectroSERVER you are configuring, for the bidirectional integration using spectrumgtw probe.

NOTE

Information:

- Inventory is a list of devices from single or multiple landscapes.
- The deletion of DX NetOps Spectrum models in CA UIM is not supported.
- Even if you have enabled the integration for Server Management and VMware Management, you can enable this bidirectional integration provided that you deploy and configure the spectrumgtw probe. In this case, integration for Server and VMware Management, and the bidirectional integration work without any conflict.

Enable the Bidirectional Integration

You need to complete the following steps to enable this integration:

- Verify the following prerequisites:

  - CA Spectrum v10.3 and CA UIM v8.51 are deployed.
  - The spectrumgtw probe (v8.66) is deployed and configured. For more information on the Spectrum Gateway probe, see spectrumgtw AC Configuration.
  - The CA Unified Infrastructure Management Alarm Server (nas) probe v9.00 or higher must be installed and activated.
  - The CA Unified Infrastructure Management Event Management Services (v10.17 or higher) probe must be deployed and configured for the **spectrumgtw** probe to run successfully.
  - The CA Unified Infrastructure Management trellis probe should be active and the **Spectrum UIM Services** service should be running.
- Configure the **UIM Configuration** page on the DX NetOps Spectrum OneClick Administration page.
- Select the bidirectional integration option on the **UIM Configuration** page

NOTE

For more information on configuring the spectrumgtw probe, see spectrumgtw Admin Console Configuration.

To enable the bidirectional integration, follow these steps:

1. Launch the DX NetOps Spectrum OneClick home page, and click **Administration**.
2. From the left Navigation pane, select UIM Configuration.
3. In the UIM Configuration section, enter the following information in the corresponding fields:
   a. UIM Server Host Name
   b. UIM Server Port
   c. UMP Server Host Name
   d. UMP Server Port
   e. UMP Server Protocol
   f. UIM Group Name

NOTE

Please specify the UIM group name in this field if you want selective inventory synchronization (of the UIM group specified) from CA UIM to DX NetOps Spectrum.

4. Select the dedicated SpectroSERVER from the drop-down list.

NOTE

The new devices that are managed by CA UIM will be created under the selected SpectroSERVER.

5. Select the **DX NetOps Spectrum-UIM Bidirectional Management** checkbox.
6. Click **Save.**
7. Verify that the UIMAdmin container model of type EventAdmin is created in **OneClick-> Universe topology**.
   You have enabled the DX NetOps Spectrum and CA UIM bidirectional integration successfully.

Share DX NetOps Spectrum inventory with CA UIM

You need to create a global collection of the devices (and their interfaces) that you want to share with CA UIM. Once the global collection is created with all the required devices, CA UIM receives that inventory information. DX NetOps Spectrum inventory information is synchronized in CA UIM when new devices are included or deleted from that global collection.

DX NetOps Spectrum inventory need not include devices that belong to the following model types or their derivatives:

- Pingable
- ContainerDevice
- ClusterBaseResource
- GenCblModem
- GenSetTop
- VMwareESXHost
- SolarisZonesHost
- IBMLPARHost
- HyperVHost

The following procedure describes the search criteria that exclude these devices. If these devices are included in the global collection, they do not appear in CA UIM.

Follow these steps:

1. From **OneClick->Navigation pane** right-click **Global Collections**, and select **Create Global Collection**.
2. Enter the following details in the **Create Global Collection** dialog for this global collection:
   – Name
   For example, enter "InventoryGC".
   – Security String
3. Click **Landscapes** to include devices from all the required landscapes in a DSS environment.
4. Click **Search Options** to specify the search criteria for including devices in this global collection.
   The Search Options dialog appears.
5. In the **Search Criteria** pane, click the **Show Advanced >>** button.
   The **Hints** pane appears.
6. Click the **Add Existing** button.
   The **Add Existing Search** dialog appears.
7. Expand **Models**, and select **UIM Bidirectional Search**.
8. Click **OK**.
   You come back to the **Search Options** dialog.
9. In the **Update Options** pane, select **Real-Time update** and then Click **OK**.
   You come back to the Create Global Collection dialog.
10. Click OK.
    A global collection called "InventoryGC" is created, and then based on the search criteria devices are added to InventoryGC.

NOTE

- In a Fault-Tolerant environment, to avoid model mismatch between SpectroSERVERs, perform Online Backup from a Primary MLS/Non-MLS SpectroSERVER to the respective Secondary SpectroSERVER.
- You must enable the bidirectional integration and run a successful full synchronization of inventory and alarms, before performing the online backup.

- If the CA UIM-DX NetOps Spectrum integration is enabled in an FT setup, it is recommended that you synchronize the Primary and Secondary SpectroSERVERs after the first successful synchronization. This will ensure that you have the integration data backed up in case the Primary/ dedicated SS (specified in the OneClick Admin > UIM Configuration) goes down.
- It is also recommended that you synchronize the Primary and Secondary SpectroSERVERs, after a successful synchronization job, if you are moving Datacenters from an MLS to a Non-MLS landscape.

Custom Attributes Synchronization

Using the Spectrum and UIM integration you can synchronize the UIM custom attributes from UIM to Spectrum. When you enable the DX NetOps Spectrum-UIM Bidirectional Management integration, the following custom attributes are synced from UIM to DX NetOps Spectrum.

- User Tag 1
- User Tag 2
- Custom 1
- Custom 2
- Custom 3
- Custom 4
- Custom 5

Follow these steps to view the custom attributes in DX NetOps Spectrum:

1. Open the OneClick Administration page.
2. Navigate to **Administration** tab > **UIM Configuration** link in the left panel.
3. Select the **DX NetOps Spectrum-UIM Bidirectional Management** option.
   The integration is enabled.
   After the integration is enabled, the alarms are synced from UIM to Spectrum.
4. In the Alarms table, right-click a column heading to open the Table Preferences dialog.
5. Select the custom attributes that you want to display.
   The selected custom attributes are added to the Alarms Table. The custom attributes are updated in the Alarms Table for the latest updates.

   NOTE

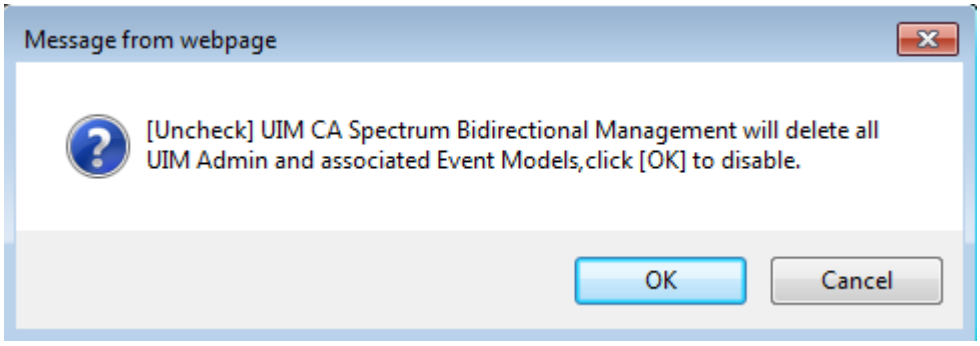
   The Alarm Details tab also displays the custom attributes and description that is synced from UIM. The Alam Details tab will not be updated for the latest updates. You can see the latest updates only in the Alarm Table.

Disable the Bidirectional Integration

You can disable the DX NetOps Spectrum and CA UIM bidirectional integration from the OneClick Administration page.

Follow these steps:

1. Open the OneClick Administration page.
   The OneClick Administration page opens.
2. Navigate to **Administration** tab > **UIM Configuration** link in the left panel.
   The UIM Configuration window opens.
3. Clear the **DX NetOps Spectrum-UIM Bidirectional Management** option.
   The following dialog appears.



4. To confirm, click Ok.

DX NetOps Spectrum-UIM Bidirectional integration is disabled successfully.

Troubleshooting

Symptom: EventModel not created on dedicated SpectroSERVER when any SpectroSERVER within the DSS setup is down
Scenario: When an alarm is raised on a device that is modeled in a SpectroSERVER that is not integrated into UIM, and the device is present in the Global collection. When the SpectroSERVER holding the device goes down, the EventModel is not created on that SpectroSERVER.
Resolution: You need to manually restart the SpectroSERVER which has the EventAdmin modeled.

Symptom: Source attribute is Empty or Null for alarms generated from IP Services related logical models.
Resolution: You need to clear alarms with Source attribute Empty or Null manually, in DX NetOps Spectrum or in CA UIM to maintain consistency across both DX NetOps Spectrum and CA UIM applications.

For more known issues and troubleshooting tips for the integration using the gateway probe, see spectrumgtw Troubleshooting and Known Issues.

Multitenant support

Introduction

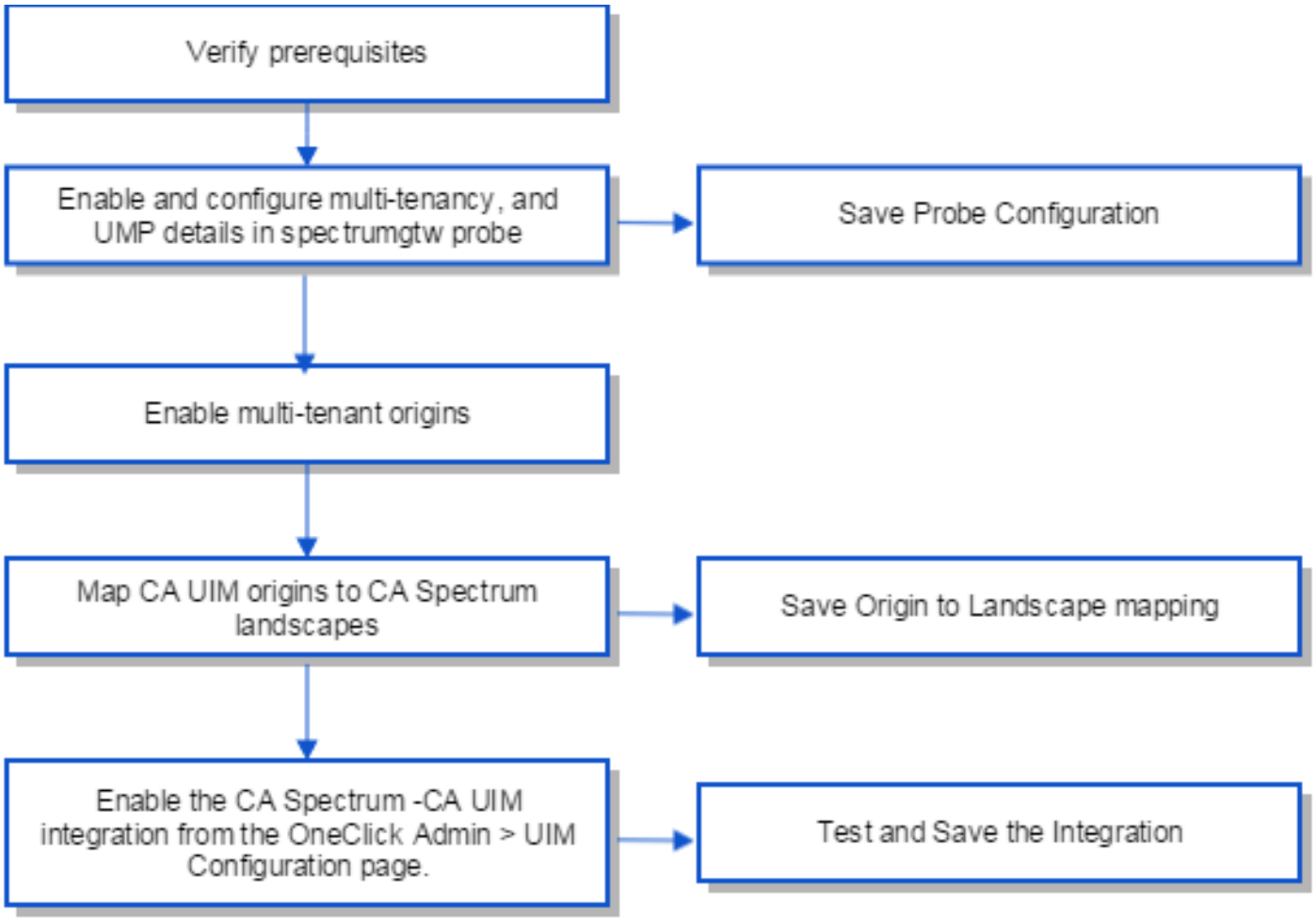
From the release 10.2.2, DX NetOps Spectrum supports multitenancy for DX NetOps Spectrum-UIM integration that is configurable in the Spectrum Gateway (spectrumgtw) probe.

The Multitenancy model in UIM is based on the Origin tag (Ownership). DX NetOps Spectrum-UIM integration leverages this origin tag information. The spectrumgtw probe allows you to map Origin with Landscape. Mapping UIM origins to DX NetOps Spectrum landscapes, ensures that the inventory belonging to a specific origin is modeled in the landscape it is mapped to.

> **WARNING**
>
> To enable and configure multitenancy, you need to install spectrumgtw probe v8.64 on CA UIM 8.51 or 8.51 SP1 This capability is not supported by previous versions of the probe. You also need to deploy the webservices_rest package version 8.51 or later on the same robot where UMP is deployed.

The following diagram outlines the process to enable multitenancy in DX NetOps Spectrum.

Figure 80: Multi-tenancy Configuration workflow



In deployments where there is a clear distinction in terms of UIM and DX NetOps Spectrum installation setups, i.e. one UIM hub and one SpectroSERVER per tenant (where the tenant has one origin), simply mapping origin to landscape is enough.

However, for deployments where there are multiple origins per tenant (with overlapping IP addresses that need to be modeled in same landscape), or where entities of multiple UIM tenants have to be modeled in the same landscape, you need to configure Secure Domain Connectors on the landscape and map the SDC + landscape combination to an origin.

The probe retrieves the list of origins in your UIM environment. You can then map your DX NetOps Spectrum landscapes to UIM origins.

> **WARNING**
>
> While you can map multiple origins to the same landscape, the same origin cannot be mapped to multiple landscapes.

How multitenancy works in the DX NetOps Spectrum-UIM integration

Once you manually map the origins to landscapes, using REST calls provided by DX NetOps Spectrum, spectrumgtw probe sends the Origin to Landscape mapping. These mappings are sent to DX NetOps Spectrum every time DX NetOps

Spectrum OneClick server or spectrumgtw probe restarts. Once the mappings are in place, Spectrum ensures that entities belonging to an origin are modeled on the landscape, its origin is mapped to.

NOTE

The spectrumgtw probe restarts every time you change mapping and save the configuration

When spectrumgtw probe gets alarms from DX NetOps Spectrum, the probe determines the landscape for each alarm and fetches the corresponding origin (by checking against the mapping and filling the origin in the alarm) before sending it to UIM.

Until 10.2.2, at the most two UIM Event admin models were created (one in selected landscape and another in MLS if the selected SS is down) to handle alarms coming from UIM. When origins are mapped to landscapes then one UIM event Admin model is created per landscape as specified in the map. This allows the segregation of alarms into their respective landscapes.

Event models are created for alarms whose IP or hostname are not present in DX NetOps Spectrum. In cases like overlapping IPs or duplicate hostnames, chances are that same event model is used as a place holder for asserting such alarms. Now origin in the alarm will be used to identify the correct event model to assert the alarm.

Configuring Multitenancy for DX NetOps Spectrum-UIM Integration

To enable multitenancy, follow these steps:

1. In the probe Admin Console, navigate to the spectrumgtw probe configuration interface/ GUI.
2. In the **UMP Connection Details** section, enable the **Configure multitenancy** option, update the UMP details and save the configuration.
   The probe configuration interface displays the **Multitenant Mapping** node.
3. In the **Multitenant Mapping** node > **Landscape Mapping** section, click **New** to create a new landscape to origin mapping.
4. Map UIM Origins to DX NetOps Spectrum Landscapes, from the list of options displayed and save the mapping changes.
5. In the OneClick Administration web server, click the Administration tab and navigate to the UIM Configuration page.
6. Update the integration configuration details, enable the integration(s).
7. Test and Save integration.

NOTE

For more information about configuring and enabling multitenancy on the spectrumgtw probe, see spectrumgtw Advanced Configuration.

Upgrade Scenarios and Mapping Behavior

Go through the following scenarios for the expected behavior of models in DX NetOps Spectrum after you complete configuring and enabling multi-tenancy.

Upgrade Scenarios

Enabling Multi-tenancy after upgrading to 10.2.2, from a 10.2.1 DSS environment which is already integrated with UIM

Scenario: I am currently on 10.2.1 and the bidirectional integration with UIM (with VMware and Server management) are enabled in my DSS environment. What happens to the models that already exist in my DX NetOps Spectrum environment in specific landscapes in the Universe view when I configure and enable Multi-tenancy?

Expected Behavior: After you have successfully upgraded to 10.2.2, ensure that all the models exist in the selected Landscape.

1. Map DX NetOps Spectrum landscapes (or Landscape + SDC) to UIM Origins from the **Landscape Mapping** section and enabled multi-tenancy from the spectrumgtw probe Admin Console. The following behavior is expected:

- – If the models already exist in the mapped landscape, then they should be retained and ensure that the UIMOrigin (0x1337f) attribute has the correct value.
  – If the models exist in a different landscape, they should be removed and modeled in the new landscape which is mapped to the origin.

Enabling multi-tenancy on a new 10.2.2 DSS setup (Install 10.2 and upgrade to 10.2.2)
Scenario: I have a new 10.2.2 setup, after I upgraded from 10.2.2 from 10.2.

Expected Behavior: After you complete the upgrade to 10.2.2. from 10.2, validate the modeling behavior before and after receiving the mapping.

1. Integrate 10.2.2 with UIM [In DSS].
2. Enable Server and VMware management for a selected landscape.
3. Map DX NetOps Spectrum landscapes (or Landscape + SDC) to UIM Origins from the **Landscape Mapping** section and enabled multi-tenancy from the spectrumgtw probe Admin Console.
   The following behavior is expected:

- – If the models already exist in the mapped landscape then they should be retained and ensure that the origin attribute has the correct value.
  – If the models exist in a different landscape, they should be removed and modeled in the new landscape which is mapped to the origin.

  NOTE

  If the DX NetOps Spectrum-UIM integration via spectrumgtw probe is already in place, but some of the device models have origin not mapped to a landscape, the related models (coming from from the origin that is not mapped) are not removed from the Landscape they are currently modeled in. You need manually remove those models.

Mapping Behavior
Points to note:

- Once the Origin to Landscape mapping is in place, any new entities coming in from UIM (with an origin that is not mapped to a landscape), are not modeled in DX NetOps Spectrum. Alarms are not be stopped but are asserted to Main Location Server's UIM EventAdmin model. Existing alarms will remain with previous EventAdmin, even after multi-tenancy is enabled. If you do not need the models, you have to remove the models manually.
- If the origin gets mapped to a new landscape, then the existing models corresponding to that origin are deleted from the old landscape and are modeled in the new landscape.
- If a new SDC is specified for an origin, in the origin to landscape map, then all the entities are associated with that SDC. Similarly, if you change the mapped SDC all entities are associated with the SDC most recently mapped.
- If Origin to Landscape mapping is removed, corresponding device models are not removed from DX NetOps Spectrum. If you do not need the models, you have to remove the models manually.
- When a new configuration of Origin to Landscape mapping is added, device models of that origin are synced during the next inventory full synchronization, after the map update is received by DX NetOps Spectrum.
- Till 10.2.1, you could see all the VMware Datacenters in the **UIM Manager** > **Information Tab** > **VMWare Configuration** > **VMWare Datacenter Modeling** table, irrespective of where they were modeled, i.e. across landscapes. From 10.2.2, VMWare datacenter(s) modeled in specific landscapes, will show up under the **UIM Manager** > **Information Tab** > **VMWare Datacenter Modeling** table, view of the landscape where they are modeled.

Reconciling Entity Data Integration Using Web Server for Server Management

WARNING

From 10.1.2, the SNMP Gateway probe and SBGW /Southbound gateway are no longer recommended for alarms synchronization from UIM to DX NetOps Spectrum. It is recommended to use the Spectrum Gateway (spectrumgtw) probe for this purpose. For more information about the integration of DX NetOps Spectrum with other CA products, see Integration Compatibility.

We recommend that you go through the DX NetOps Spectrum-UIM Integration using spectrumgtw probe - FACT SHEET to understand the changes to the integration whether you are new to this integration or an existing user.

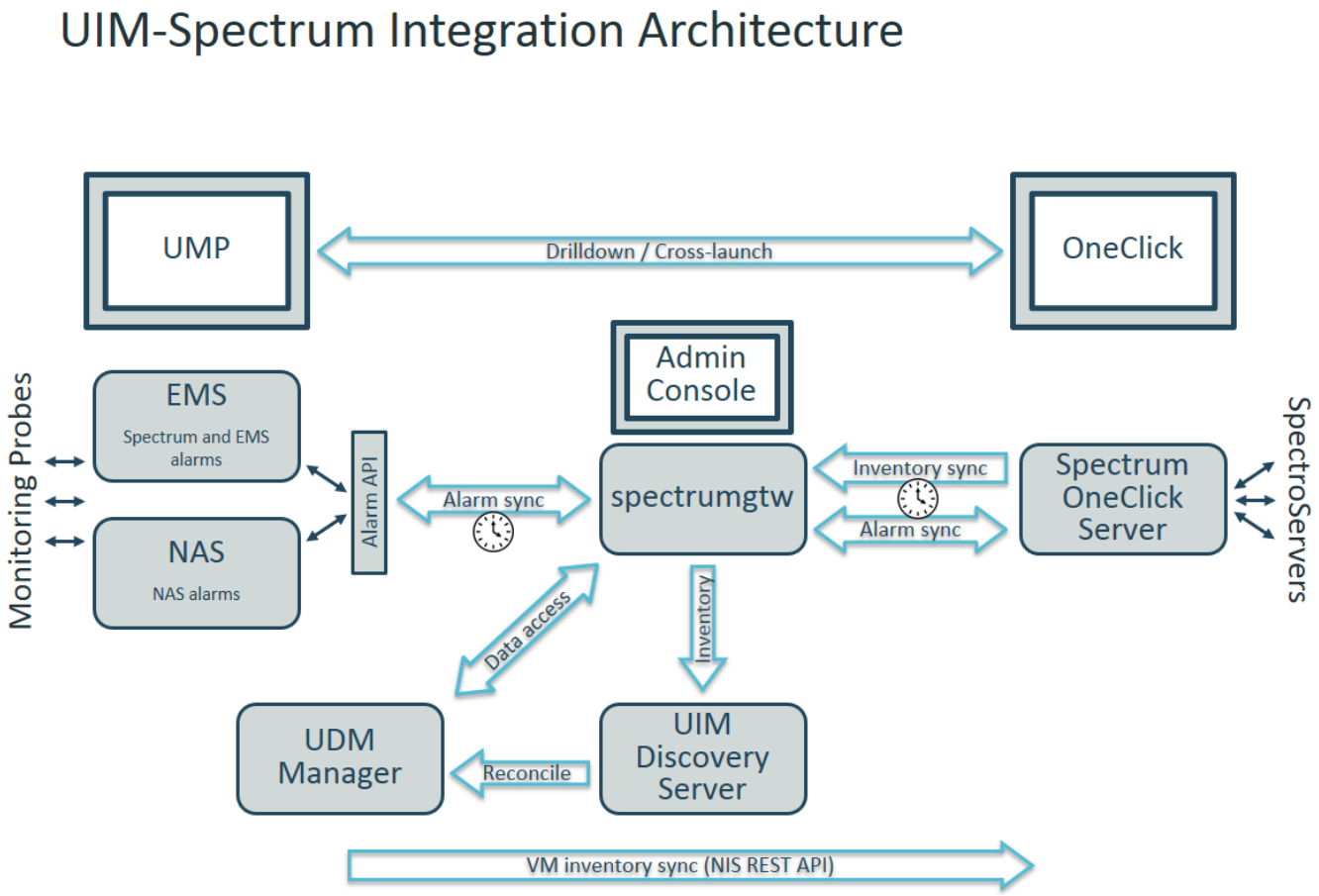
This section describes how to set up the integration between DX NetOps Spectrum and UIM through the webserver for server management. It also describes how to use its features to perform specific tasks such as synchronize inventory, manage alarms and events, and run reports.**Contents**

How to Integrate DX NetOps Spectrum and UIM through the Web Server

The integration between DX NetOps Spectrum and UIM through Web Server is designed to support server management. The HostServer models resulting from the integration provides traditional Spectrum capabilities such as layer2 connectivity and fault isolation with the features supported by UIM. You can use the Launch-in-Context feature to launch the CA Unified Management Portal (UMP) view from the server models in DX NetOps Spectrum to view the server information.

When the UIM and DX NetOps Spectrum integration is enabled from the OneClick Administration page, DX NetOps Spectrum receives the data from UIM through the Rest API. Once we get the data from the Rest API, UIM hosts are modeled in DX NetOps Spectrum.

The following diagram illustrates the UIM - DX NetOps Spectrum Integration Architecture:

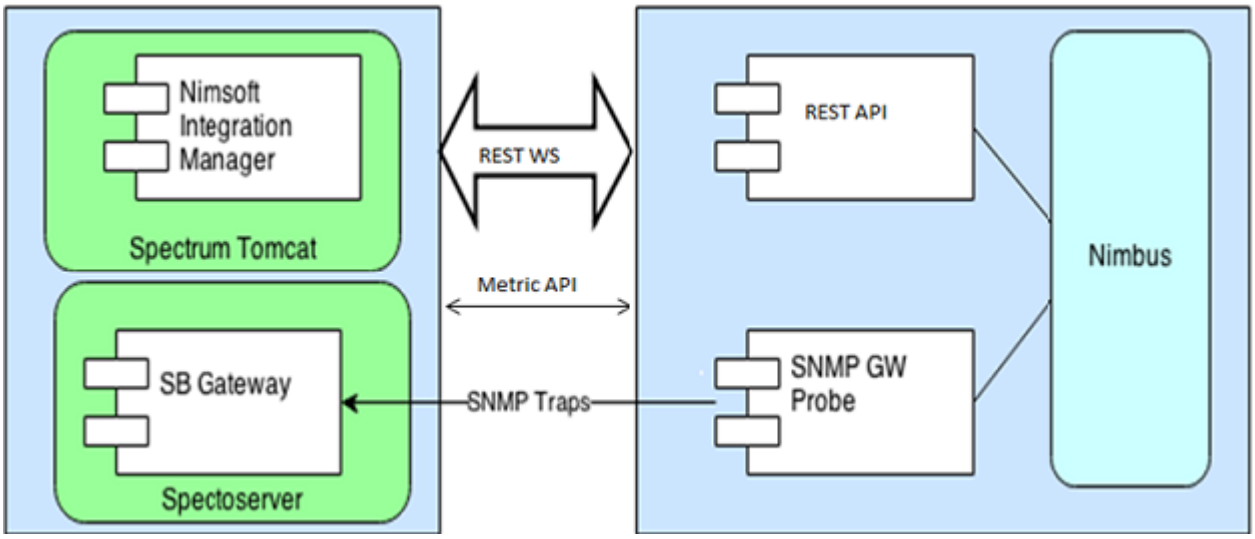


NOTE

Existing DX NetOps Spectrum – UIM integration for alarms through southboundgateway and UIM's snmpgtw probe should be stopped by deleting the associated Spectrum profile from UIM's snmpgtw probe.

Alarms sent from UIM to DX NetOps Spectrum will no longer leverage the SNMP Gateway or Spectrum Southbound gateway. Instead, they will be routed through the new Spectrum Gateway Probe.
Nisapi-service-host is replaced by nisapi-wasp for inventory sync from UIM to DX NetOps Spectrum.

The following diagram illustrates the UIM - DX NetOps Spectrum Integration Architecture till 10.1.1:



The alert data is sent from UIM to the SpectroSERVER of DX NetOps Spectrum through the Southbound Gateway component. The received data is mapped to a DX NetOps Spectrum event. The Southbound Gateway determines the appropriate EventAdmin model to forward the event.

Metric violation traps that are coming from UIM are asserted on the respective Nimsoft host models in DX NetOps Spectrum.

Review the following process to integrate UIM and DX NetOps Spectrum through the Web Server:

1. Review the Prerequisites and Considerations
2. Deploy and Configure Probes
3. Enable the Integration

Prerequisites and Recommendations

Consider the following prerequisites for the DX NetOps Spectrum and UIM integration (for releases prior to 10.1.1):

- Licensed installation of CA Spectrum 10.1 and CA UIM 8.2 or 8.3 provided that nisapi_service_host v8.04 /nisapi-wasp v8.4.2 or higher probe is deployed.
- The snmpgtw, cdm, and net_connect must be deployed and configured before integrating UIM and DX NetOps Spectrum through the Web Server
- For Launch-in-Context to work, UMP must be configured to UIM.

Consider the following prerequisites for the DX NetOps Spectrum and UIM integration (for CA Spectrum r10.1.2 or later):

- UIM Server should be at version 8.4.7 or later.
- Existing DX NetOps Spectrum – UIM integration for alarms through southboundgateway and UIM's snmpgtw probe should be stopped by deleting the Spectrum profile from UIM's snmpgtw probe
- cdm and net_connect must be deployed and configured before integrating UIM and DX NetOps Spectrum.
- ems 8.4.3, nas, and trellis probes should be deployed and running.
- spectrumgtw probe must be deployed on a robot connected to the primary hub.
- spectrum-uim-service-impl has to be deployed on the primary robot where Trellis is installed and Trellis has to be restarted.
- For server/vmware inventory synchronization:

  - nisapi-wasp 8.4.2 or higher
  - vmware probe version 6.72 or higher
- Use the latest ticketing gateway probes for Ticket ID synchronization and ensure the key **create_incidents_for_spectrum_alarms** is set to **“yes”** to allow probe to create tickets for spectrum alarm

Consider the following recommendations for the integration:

- If a new SpectroSERVER is added to the Distributed SpectroSERVER (DSS) setup, you must restart the OneClick server.
- If a child SpectroSERVER is removed from the DSS setup, restart the OneClick server.
- If anyone of the available SpectroSERVER is initialized to a legacy database or another database, restart the OneClick server.
- Specify a SpectroSERVER that has less load as the dedicated SpectroSERVER for UIM integration.
- Use an OneClick web server that is not integrated with Spectrum Report Manager.

Deploy and Configure Probes

To enable the DX NetOps Spectrum and UIM integration, deploy and configure the following UIM probes on UIM server:

- **snmpgtw**
  Sends traps from UIM to DX NetOps Spectrum. The SNMP gateway converts alarms to SNMP trap messages that are readable by any SNMP-based event manager.
- **net_connect**
  net_connect (Network Connectivity Monitoring) probe measures network connectivity that is based on "ping" (ICMP ECHO) and the TCP connections to a list of user-defined services. The service can be NetBIOS, Telnet, FTP, and HTTP. The probe supports the UIM family of solutions by sending the quality of service (QoS) messages.
- **(nisapi) service_host or wasp**
  Queries the CA Nimsoft Manager using the Restfull Services API to retrieve the list of UIM models to be monitored. Deploy service_host v8.04/ wasp v8.4.2 or higher.

  NOTE

  In previous releases of UIM, the nisapi_service_host package was deployed to the service_host probe. As of UIM release 8.47, the service_host probe was deprecated and its functionality was moved to the wasp probe.

  - By default**,** the wasp probe uses port 80. However, because the wasp probe is a core component in UIM, this port should already be open in a properly functioning UIM environment. For more information, refer to the wasp probe documentation.
- **cdm** The CPU, Disk, Memory Performance Monitoring (cdm) probe monitors the performance and resource load on the system with the robot. The probe provides the following benefits:
  - Generate alarms that are based on configured threshold values. The probe generates alarms that can trigger corrective actions immediately.
  - Generate trending Quality of Service (QoS) data. The information is measured and sent to the data_engine probe to process and store in the UIM database. The historical data facilitates capacity planning for monitored systems in the IT environment. Some examples are as follows:
    - View how disks are filling up over time
    - Plan batch jobs according to the CPU utilization
    - Upgrade systems which consistently operate near capacity
- **spectrumgtw** The spectrumgtw (Spectrum Gateway) probe provides a bidirectional (UIM to DX NetOps Spectrum and DX NetOps Spectrum to UIM) integration between CA Unified Infrastructure Management and DX NetOps Spectrum enabling you to perform the following actions:

- Export and correlate DX NetOps Spectrum Inventory from a Global Collection to UIM inventory.
- Bi-directional synchronization of alarms
- Bi-directional Clear / Update of alarms
- View symptomatic DX NetOps Spectrum alarms in the context for the root-cause alarm in UIM
- Ticket ID synchronization between DX NetOps Spectrum and UIM

WARNING

From 10.2.1, all UIM events (apart from events related to VMWare and servers) are no longer considered as generic events.
Specific events raised on UIM from the following probes are mapped to unique events created in DX NetOps Spectrum. This allows you to manage the events (raised on UIM) from DX NetOps Spectrum more efficiently.

Additional event mapping for select metrics from the following probes (along with cdm and net_connect probes) is added from 10.2.1. Additional event mapping for Please refer to Additional event mapping from UIM Probes to DX NetOps Spectrum.

- **logmon** The Log Monitoring (logmon) probe scans ASCII-based systems and application log files by matching specified expressions. Alarms are generated when the log file content matches the defined expression.
- **rsp (Remote System probe)** The Remote System Probe (rsp) allows you to monitor system metrics. The probe collects performance data in an agentless manner without installing proprietary software on the system.
- sqlserver The SQL Server Monitoring (sqlserver) probe constantly monitors the internal performance and space allocation of SQL Server databases. The probe can run locally on the database server or it can be configured to run as a remote client. The probe feeds essential information that is based on predefined criteria to the UIM availability manager for appropriate alert notification, as required. An extensive range of checkpoints can be selected and individually scheduled to meet your monitoring requirements. The probe will run selected SQLs to extract vital information about your SQL Servers. The information is presented to database-administrator as alarms or reports. For more information, see sqlserver Metrics.
- **processes (Process Monitoring)**
  The (Process Monitoring) processes probe monitors the specified processes to detect any error situation. The probe also retrieves information about the process, for example, the CPU usage, memory usage, and so on. For more information, see processes Metrics.

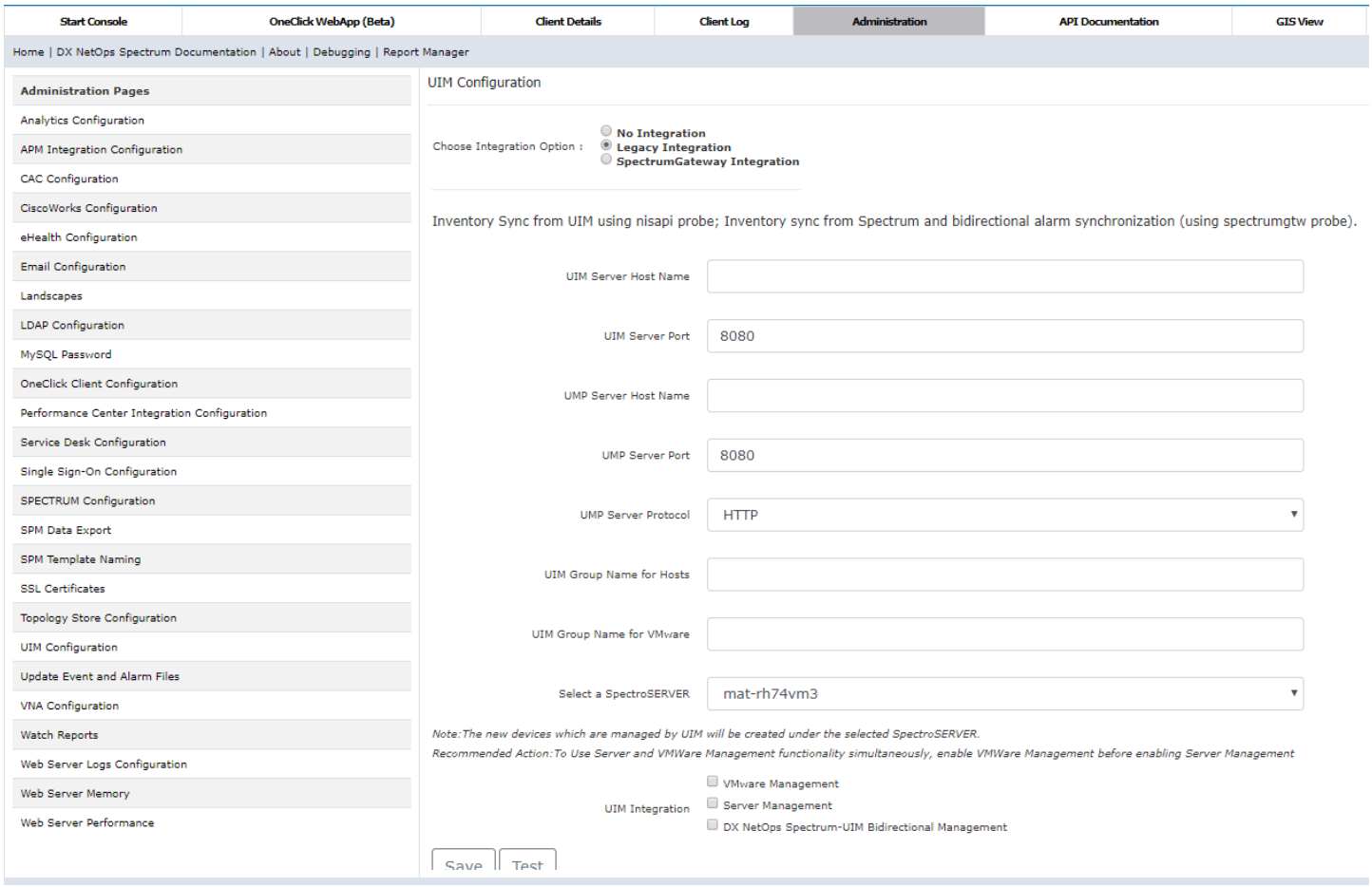
You can configure the UIM probes through the Probe Configuration interface. For more information, see the Unified Infrastructure Management Probe Space.

Enable the Integration

You can enable the DX NetOps Spectrum and UIM integration through the webserver from the OneClick Administration page. Specify the UIM Configuration information such as UIM Server Host Name, UIM Server Port, Unified Management Portal (UMP) Server Host Name, and UMP Server Port to enable the integration.

Follow these steps:

1. Open the OneClick Administration page.
   The OneClick Administration page opens.
2. Click the Administration tab.
   Links to various OneClick web server configuration pages are displayed.
3. Click the UIM Configuration link in the left panel.
   The UIM Configuration page opens.
   The following image illustrates the configuration options that are available in the UIM Configuration window:



If the integration was already enabled before the upgrade, then the **Legacy Integration** radio button would appear as selected by default.

- **UIM Server Host Name**
  Indicates the IP address/hostname of the UIM Server.
- **UIM Server Port**Indicates the server port number of UIM.
- **UMP Server Host Name**
  Indicates the IP address/hostname of UMP.
- **UMP Server Port**
  Indicates the server port number of UMP.
- **UMP Server Protocol:**
  Indicates the network protocol of the CA UMP server.

  NOTE
  For successful encrypted communication (https protocol) between DX NetOps Spectrum and UIM UMP Server, you must import SSL Certificate of UIM UMP Server into Spectrum OneClick Server. To import the certificate, from the OneClick home page select **Administration**, **SSL Certificates**, upload the certificate file then click **Save**. Restart the OneClick web server for the changes to take effect.

NOTE
From spectrumgtw 8.67 onwards, the UIM Group Name field in the UIM configuration section, has been changed to Group Name for Hosts.

- **UIM Group Name for Hosts**: Specify the UIM group name in this field for selective inventory synchronization of hosts (of the UIM group specified) from UIM to DX NetOps Spectrum. This is applicable to Server Management entities only.
- **UIM Group Name for VMware**: Specify the UIM group name in this field for selective inventory synchronization of VMware entities (of the UIM group specified) from UIM to DX NetOps Spectrum. To address the challenges in providing filtering of VMware data from UIM to Spectrum, a 'UIM Group Name for VMware' field is introduced from spectrumgtw v.8.67.

NOTE

- Ensure that you add vCenter and ESX hosts and the corresponding virtual machines to be synced to Spectrum.
- If the vCenter is not part of the group, ESX and VMs are not modeled in Spectrum even though they are part of the group.
- If vCenter and VMs are part of the group and ESX is not, then the VMs are created without any relationship with the vCenter in Spectrum.

NOTE
Inventory from the group you specify in this field is synchronized from UIM to Spectrum. For more information, see Create and Manage Groups in USM.

4. Select the SpectroSERVER under which the new UIM Host models hierarchy is to be created.
5. To enable the integration, select **Server Management** and click Test. (**Till 10.2.2 and spectrumgtw probe v8.65**)
   If the test is successful, Successfully connected to the UIM message appears.

   NOTE
   If you are setting up the integration using 10.2.3 and spectrumgtw probe v8.65, you can no longer enable the Server Management capability from the UIM Configuration page. To enable and configure the Server Management capability for Host servers, please refer to the spectrumgtw AC documentation.
6. Click Save.
   Successfully saved configuration to the database message appears.
   UIM Integration is now enabled.

   NOTE
   If you change any settings on any of the OneClick server other than the integration OneClick server. The message, "Saving details on this OC will change the Integration OC Server. Click [OK] to continue" appears. If you click Save, the integration OneClick server changes and the details are saved. If you click Cancel, the details are saved but the OneClick server remains the same.

Incremental and Full Synchronization

This Integration supports incremental and full synchronization. When the DX NetOps Spectrum and UIM integration is enabled, synchronization occurs automatically with the default scheduled timings displayed in the OneClick view. From 10.2.2, you have the option to configure including or excluding virtual machines (VMs) during the UIM Host Server Synchronization.

WARNING
From 10.2.3 and spectrumgtw probe v8.65, the Incremental and Full Synchronization configuration for Host servers are moved to the spectrumgtw Admin Console. For more information, please refer the spectrumgtw Admin Console documentation.

Incremental Synchronization

NOTE
Additions and modifications of devices in UIM are reflected in DX NetOps Spectrum after incremental synchronization. You can set the Incremental Sync interval in the OneClick view. (Till 10.2.2 and spectrumgtw probe v8.64)

Full Synchronization

Full synchronization occurs when the DX NetOps Spectrum and UIM integration is enabled. Thereafter, the full synchronization occurs at a scheduled time based on the schedule that is selected in the OneClick view. During full sync DX NetOps Spectrum queries UIM for all the hosts that are managed by UIM. Once Spectrum receives this data, reconciliation is performed and new hosts (if any) are modeled in Spectrum. The un-managed hosts are removed.

The minimum schedule time for full sync is one day. The last full sync time in the OneClick view displays the previous full sync completion time. The sync times are based on the OneClick Tomcat server time.

NOTE
Do not schedule the incremental or full sync at smaller intervals, you may experience a performance impact if a large number of servers is being monitored by UIM.

WARNING
Schedule full synchronization during non-business hours.

Follow these steps:

1. Open the DX NetOps Spectrum OneClick console.
2. From the Navigation panel, select **UIM Manager**.
   The contents pane for UIM Manager opens.
3. Click the **Information** tab and select **Configuration** > **UIM Host Sync Configuration**.
   Information on incremental and full synchronization is displayed.
4. To schedule incremental sync, click **Set**.
   The Time interval window opens.
5. Specify the time interval and click Ok.
   Default: 300 minutes
   Minimum: 10 minutes
   Incremental sync is now scheduled.
6. To schedule full sync, click the **Schedule** button that is available in the **UIM Sync Configuration** section.
   The **Create Schedule** window opens.
7. Specify the following Recurrence information:
   – Days
     Default: 7
     Maximum: 31
   – Hours
     Default: 00:01
     Maximum: 23:59
8. Click Ok.
   Full sync is scheduled.
9. In the **Include Virtual Machines** field, do one of the following:

- Select Yes, to include virtual machines.
- Select No, to exclude virtual machines.
  When you select the **Include Virtual Machines** checkbox, spectrum syncs only the non-VMWare VMs.

Reconciling UIM entity data with existing DX NetOps Spectrum models

During full synchronization or incremental synchronization, when a new UIM entity is reported to DX NetOps Spectrum from UIM, DX NetOps Spectrum will perform a search to identify if this entity was modeled during DX NetOps Spectrum

discovery and modeling. If such an existing model is found, DX NetOps Spectrum will reconcile the UIM entity information with the existing model, instead of creating a new model. This search is performed on all landscapes in the Distributed SpectroSERVER (DSS) setup.
In cases where an existing model is not found and reconciliation is not performed, DX NetOps Spectrum will model the entity in the SpectroSERVER specified in the **OneClick Admin Configuration**> **UIM Configuration** page.

> **NOTE**
> DX NetOps Spectrum uses the IP Address reported by the entity to perform this search. If the IP Address is not reported, the search is performed using the reported MAC address.
> If neither (IP or MAC Address) are reported, DX NetOps Spectrum will not be able to reconcile that entity even if the entity is discovered and modeled in DX NetOps Spectrum.

Selective Inventory Synchronization

Starting from the 10.2.1 release, the DX NetOps Spectrum, and UIM integration support selective synchronization of Server Management entities only. This functionality allows you to avoid full inventory synchronization and select only the server inventory which you want to get synced from UIM to Spectrum.

To enable the selective inventory synchronization, follow these steps:

1. Launch the DX NetOps Spectrum OneClick home page, and click Administration.
2. From the left Navigation pane, select UIM Configuration.
3. Contact the UIM Administrator for the following information, and enter it in the corresponding fields:
   a. UIM Server Host Name
   b. UIM Server Port
   c. UMP Server Host Name
   d. UMP Server Port
   e. UIM Group Name
   Inventory from the group you specify in this field is synchronized from UIM to Spectrum. The Synchronization happens only for the specified UIM group. The group name is created in UIM/UMP and it can be static or dynamic.
   > **NOTE**
   > If the Group Name field is empty or does not match with the UIM group name, then DX NetOps Spectrum pulls all server management entities from UIM during the sync.
   > You can only specify a single (static/dynamic) group from UMP at a time.
4. Select the dedicated SpectroSERVER from the drop-down list.

   > **NOTE**
   > The new devices that are managed by UIM will be created under the selected SpectroSERVER.
   > **Recommended Actions:**
   > To use Server and VMWare Management functionality simultaneously, enable VMWare Management before enabling the Server Management.
   > If both functionalities are enabled and you want to use only one of them then you must disable both and enable the functionality which you want to use.
5. Select the **Server Management** checkbox and click Test. (**Till 10.2.2 and spectrumgtw probe v8.65**)
   If the test is successful, Successfully connected to the UIM Server message appears.

   > **NOTE**
   > If you are setting up the integration using 10.2.3 and spectrumgtw probe v8.65, you can no longer enable the Server Management capability from the UIM Configuration page. To enable and configure the Server Management capability for Host servers, please refer to the spectrumgtw AC documentation.
6. Click **Save.**
   Successfully saved configuration to the database message appears.

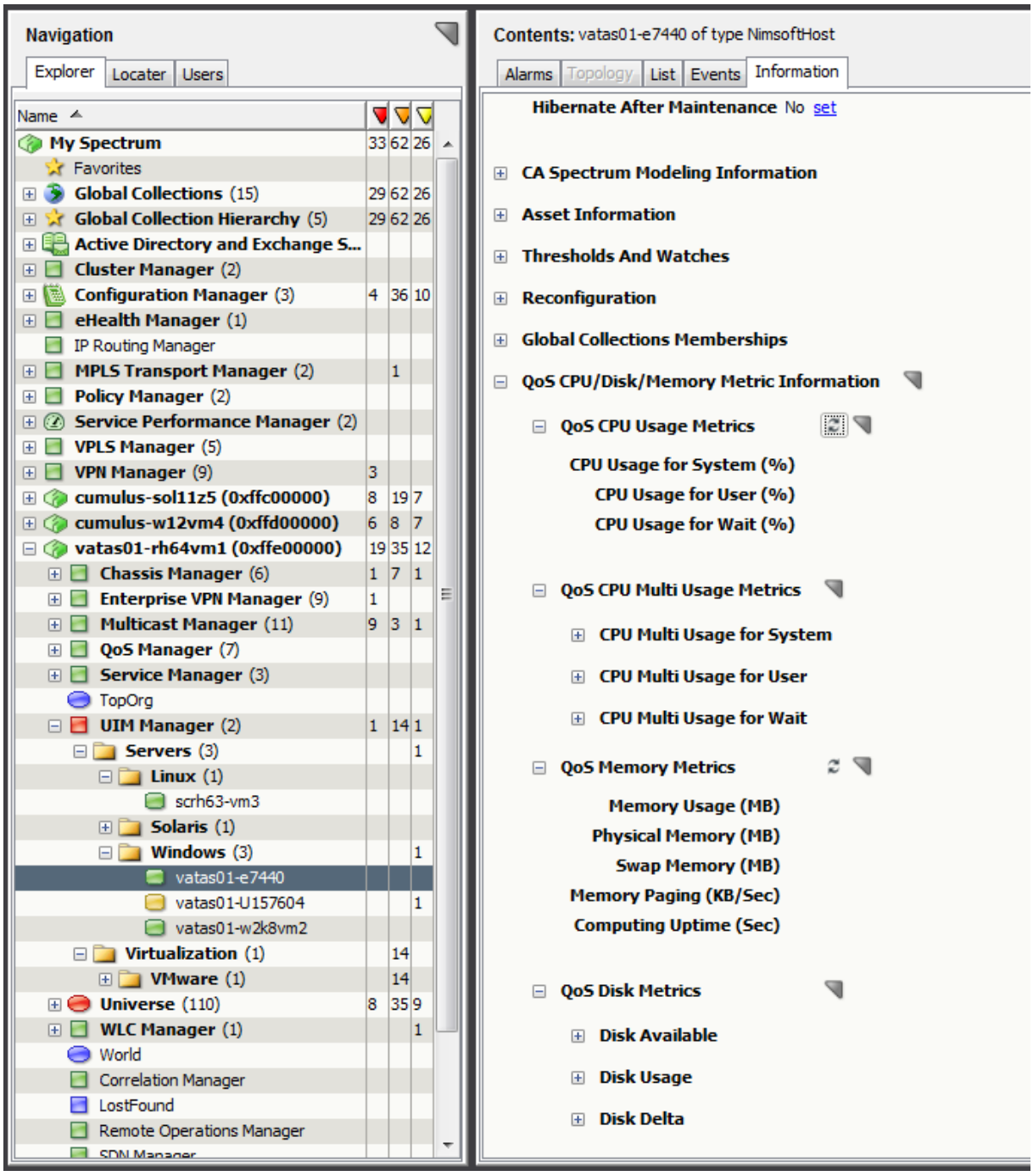
QoS Metrics

QoS Metric Information provides the metrics for both CPU and memory usage. From the Navigation Pane of OneClick Console, you can access the QoS metrics information of the available NimsoftHost Models. The following metrics are available for each UIM Host Model modeled in DX NetOps Spectrum:

- QoS CPU Usage Metrics
- QoS Multi Usage Metrics
- QoS Memory Metrics
- QoS Disk Metrics

Follow these steps:

1. From the OneClick Console, select the UIM Host Model available in the Navigation Panel.
   Contents Pane for the selected UIM Host Model is displayed.
2. Click the Information tab on Contents Pane.
   UIM Host Model information is displayed.
3. Expand QoS Metric Information SubView.
   QoS Metrics for the selected UIM Host Model is displayed.
   The following image displays the QoS Metrics for the UIM Host Model in DX NetOps Spectrum:



The following QoS Metrics are supported in this integration:

- **CPU Usage for System**
  Specifies the time that CPU spends on system tasks in percent.
- **CPU Usage for User**
  Measures the time that CPU spends on user tasks in percent.
- **CPU Usage for Wait**
  Measures the time the CPU waits when accessing external memory or another device in percent.
- **CPU Multi Usage for System**

Measures the time the CPU spends on system tasks in percent.

- **CPU Multi Usage for User**
  Measures the time that CPU spends on user tasks in percent.
- **CPU Multi Usage for Wait**
  Measures the time the CPU waits when accessing external memory or another device in percent.
- **Disk Available**
  Measures the amount of total available disk space for the file system. The Disk Available metrics are populated for only Network file systems.
- **Disk Usage**
  Measures the amount of total used disk space in the file system.
- **Disk Delta**
  Measures the amount of total disk usage change in the file system.
- **Memory Usage**
  Measures the amount of total available memory (physical and virtual memory) used in megabytes
- **Physical Memory**
  Measures the amount of total available physical memory that is used in megabytes.
- **Swap Memory**
  Measures the space on the disk that is used for the swap file in megabytes.
- **Memory Paging**
  Measures the amount of memory that is sent or reads from virtual memory in kilobytes/second.
- **Computing Uptime**
  Measures the computer uptime in seconds every hour.

  NOTE
  You may experience latency in loading QoS Metrics as the values are generated dynamically from UIM server, when queried.

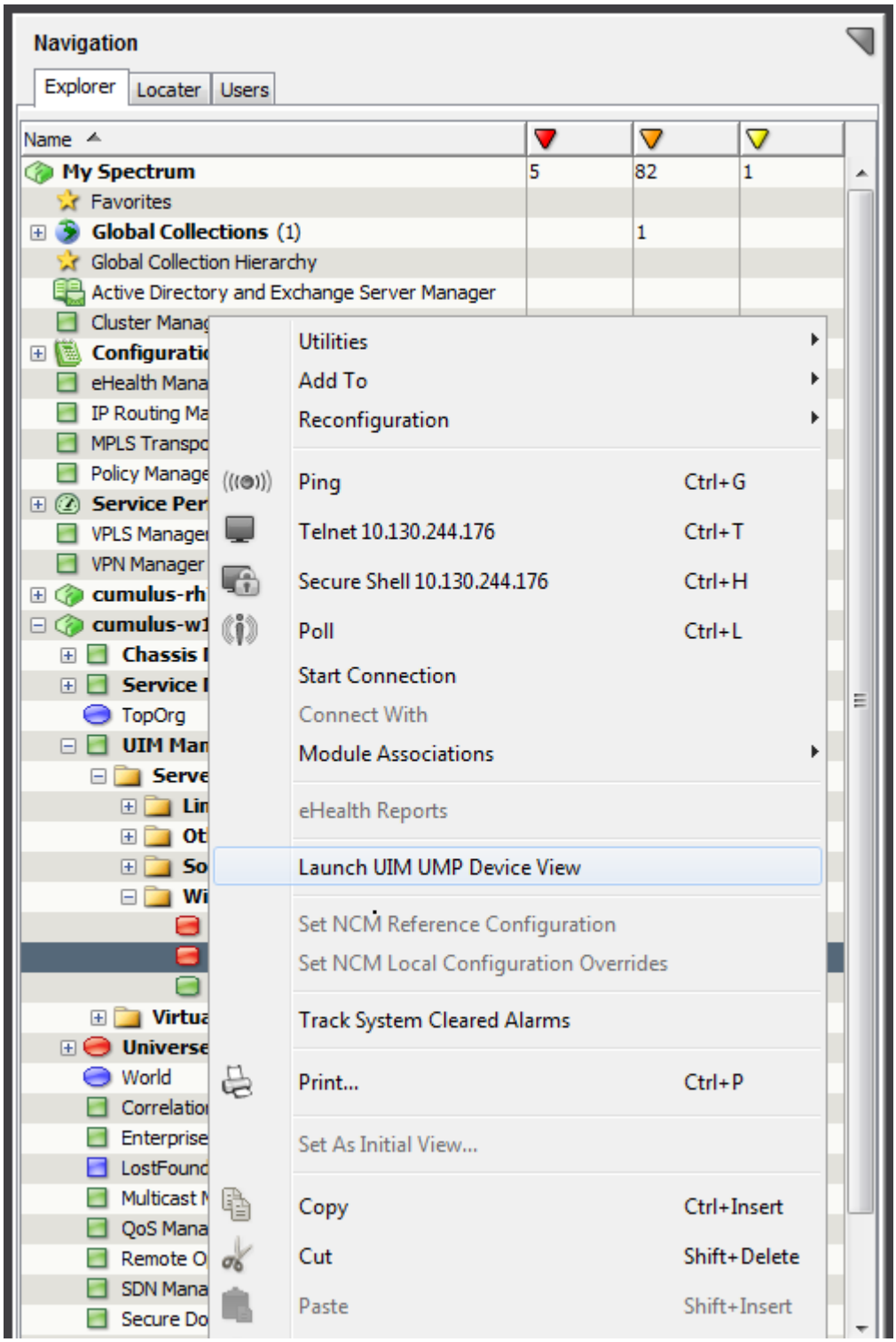
Launch-in-Context

The Launch-in-context feature is used to view the Unified Management Portal (UMP) of the host for the UIM host model. This feature provides detailed information about the UIM host model such as disk usage, CPU usage, processor queue length, paging, and memory usage. The information about the UIM host model is displayed graphically.

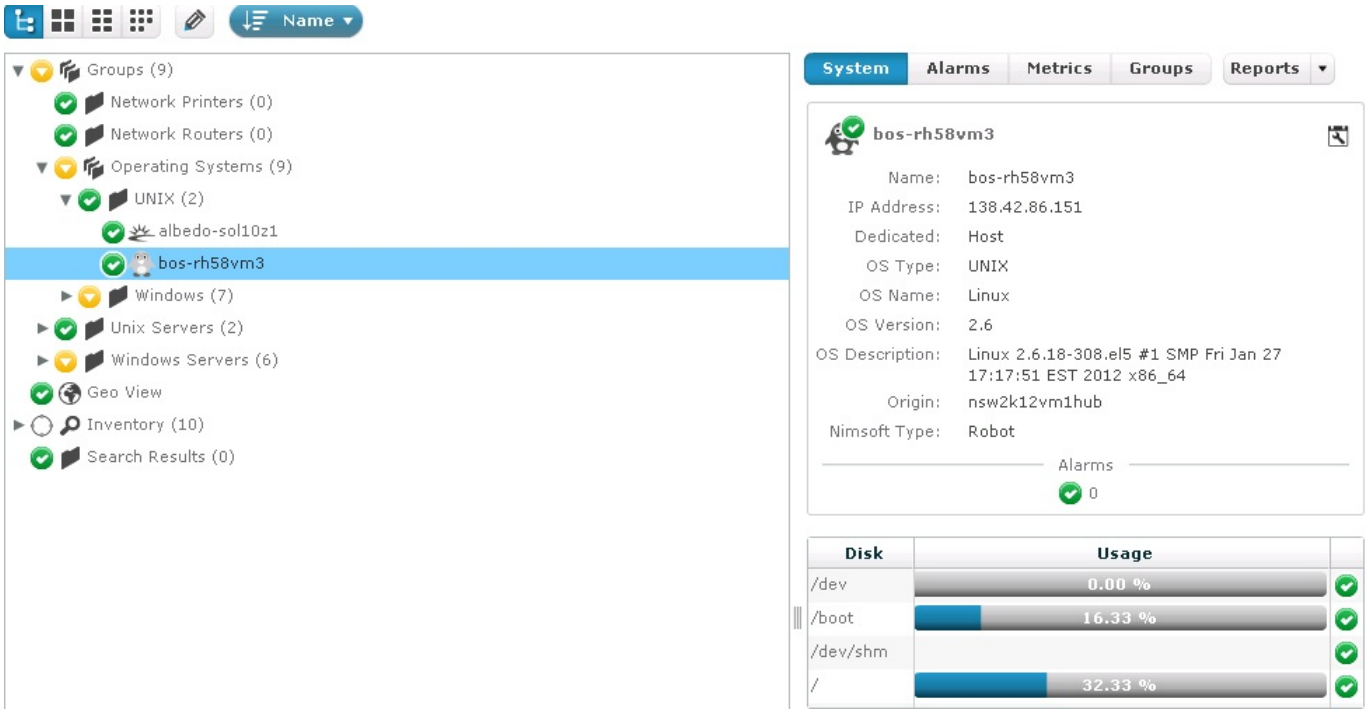
If you are launching the UMP view for the first time in a browser, a dialog for user credentials appears. The user credentials dialog does not appear if you are launching the UMP view using the same browser instance.

Follow these steps:

1. Open the DX NetOps Spectrum OneClick console.
2. From the Navigation panel, select UIM Manager and Servers.
   A complete list of host models is displayed in respective folders.
3. Right-click a host model and select Launch UIM UMP View.
   The UIM UMP login page opens.
   The following image displays the available host models and the option to launch the UMP view from the OneClick console:



4. Enter the UIM UMP credentials and click Login.
   The selected model details are displayed and the UIM UMP login is successful.
   The following image displays the UIM UMP view:

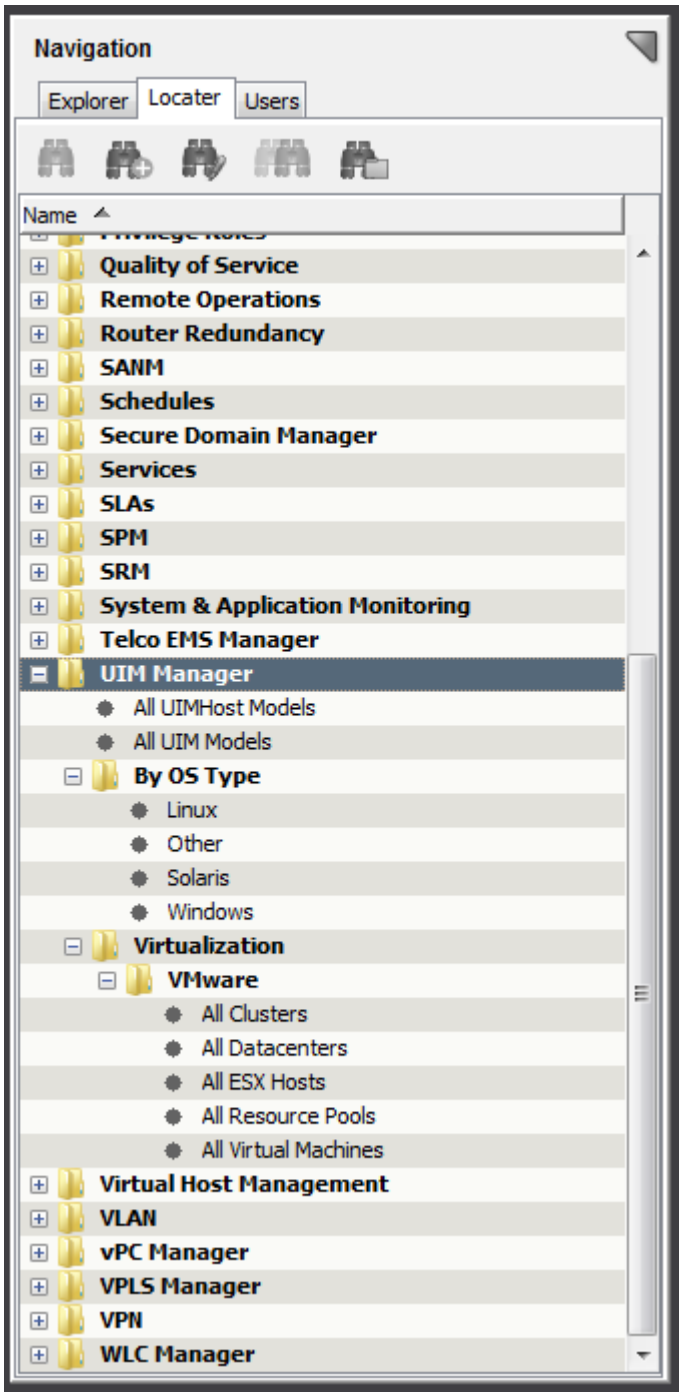


Locater Search

You can use the search functionality in the Locater tab to find the UIM related devices that are available in the DX NetOps Spectrum environment. Search can be performed based on the Operating system type such as Windows, Linux, and Other. Using this functionality, you can also search for all the NimsoftHostServer models and UIM models that are available in DX NetOps Spectrum. The search results appear in the Results tab of the Contents panel. Detailed information for application models that are selected in the results list appears in the Component Detail panel. Access Locater search from the Locater tab of the Navigation Panel.

Follow these steps:

1. Open the DX NetOps Spectrum OneClick Console.
2. From the Navigation Panel, Click the Locater tab.
   The Search Options window opens.
3. Expand UIM Manager and select the models.
   The Locater Search results are displayed in the Contents pane.
   The following figure displays the Locater Search results for UIM Configuration Manager:



Reports

You can generate asset, alarm, availability, and WEBI reports for the UIM hosts. You can access InfoView from the OneClick home page to generate and manage reports. For more information, see the Report Manager section.

Outage Events

This section lists the events that mark the beginning and end of either a planned or unplanned model outage. The following list of events is used for the calculation of availability reports for UIM Host Server Models.

- Up events
  - 0x6330057
  - 0x6330000
- Down events
  - 0x6330003
  - 0x6330056

Standard up and down events are ignored for UIM Host Server Models while calculating the outages. For an existing spectrum model the outage is calculated based on standard up and down events.

Traps and Alarms Support

This integration supports the following alarms:

Generic Alarms

If any threshold violation occurs on UIM hosts, generic alarms are raised.

Event Code Range: 0x6330000 – 0x6330005

Disk Alarms

If the disk usage is high or the disk space availability is low on UIM hosts, disk alarms are raised.

Event Code Range: 0x6330030 – 0x6330035

Memory Alarms

If low memory or any threshold violations are noticed on UIM hosts, raise memory alarms.

Event Code Range: 0x6330040 – 0x6330045

CPU Alarms

If CPU utilization is high on UIM hosts, CPU alarms are raised.

Event Code Range: 0x6330050 – 0x6330055

Condition Correlation and Fault Isolation in UIM Integration

If a managed device stops responding to polls, the DX NetOps Spectrum fault isolation algorithm determines whether to create a critical alarm for the UIM hosts or suppress its alarm state. The unreachable device is the root cause of the alarm.

After the integration of DX NetOps Spectrum and UIM, the events/alarms are received from both Spectrum polling and UIM. Consider the following scenarios to apply condition correlation:

Scenario 1

If a Spectrum event is generated on the UIM host before the UIM event, condition correlation applies and the UIM event suppresses the Spectrum event.

Scenario 2

If a UIM Event is generated on the UIM host before the Spectrum event, condition correlation applies and the events are asserted on their respective host.

Scenario 3

If only Spectrum event is generated on the UIM host, condition correlation cannot be performed until the UIM event is generated and the Spectrum Event is displayed on the UIM host.

Scenario 4

If only UIM event is generated, you cannot perform condition correlation and the Nimsoft Event is displayed on the Nimsoft host.

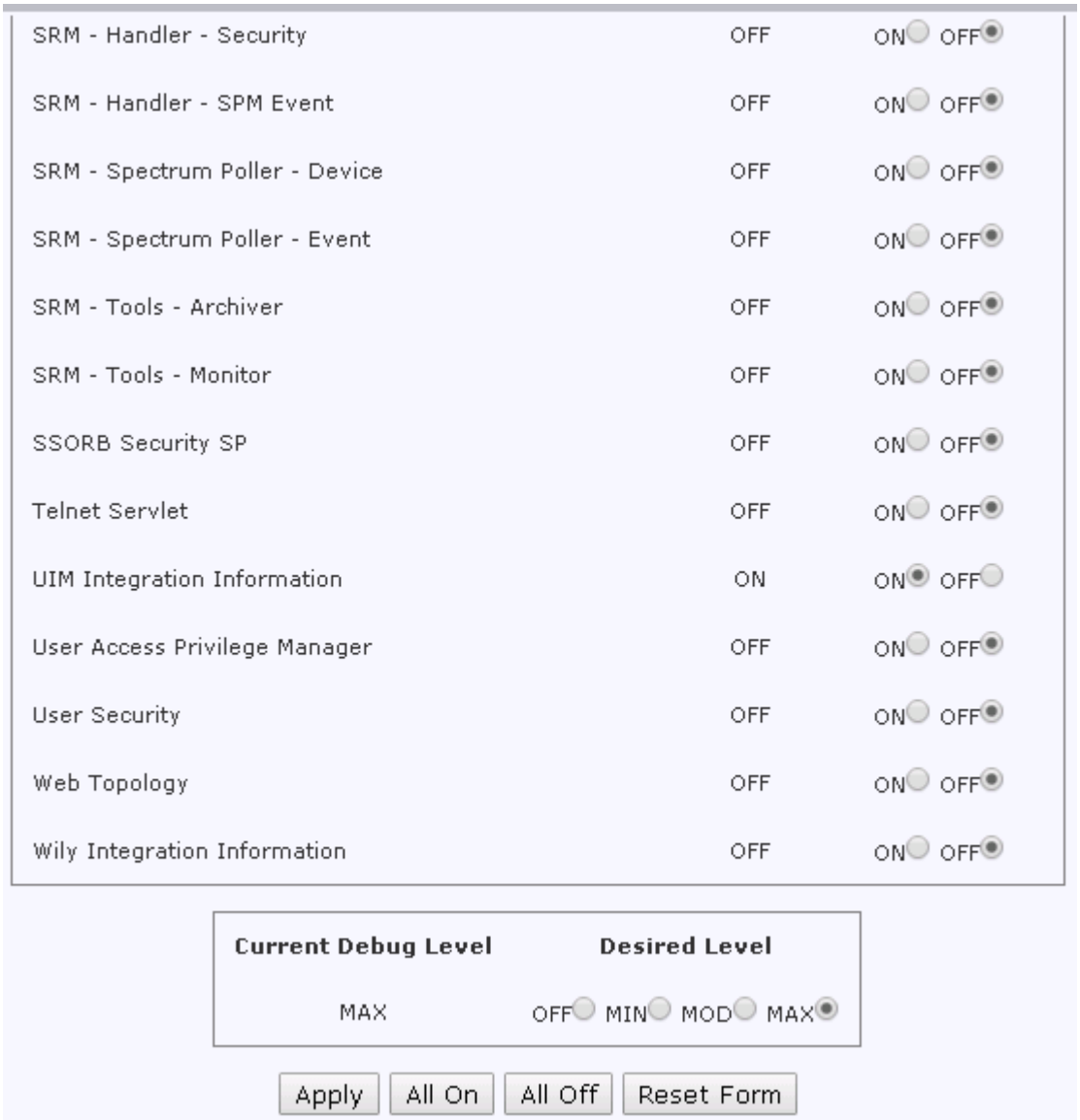
Debugging

Debugging in DX NetOps Spectrum lets you track the data flow from UIM to DX NetOps Spectrum. It investigates and resolves integration related issues. The Start Client Debug Console contains various debug modules. Turn on UIM Integration Information to track alerts and CIs that flow from UIM to DX NetOps Spectrum.

To use Start Client Debug Console, you must first have a running OneClick client. This debug tool lets you turn on debugging output that can be seen in the Java Web Start log.

Follow these steps:

1. Open the OneClick Administration page.
   The OneClick Administration page opens.
2. Click the Administration tab.
   Links to various OneClick web server configuration pages is displayed.
3. Click the Debugging tab.
   A panel with various links to view debugging output opens.
4. Click Web Server Debug Page (Runtime).
   A list of debug modules is displayed.
   The following image displays the list of available debug modules:



5. To enable debug, select On for the UIM Integration Information debug module.
6. Select Max as Desired Level and click Apply.
   Debug is enabled.
7. To disable debug, select OFF and click Apply.
   Debug is disabled.

Disable the Integration

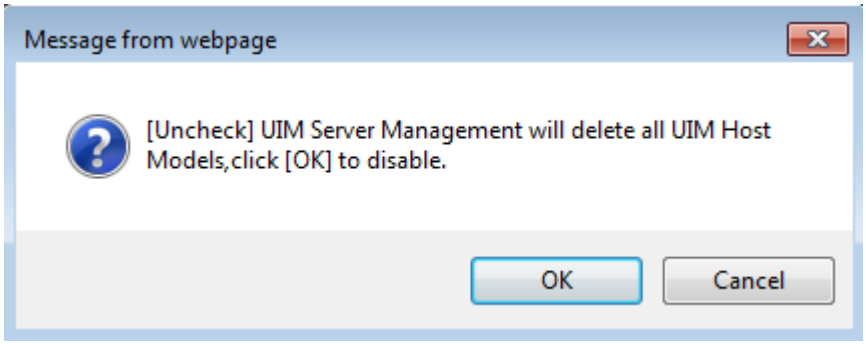
You can disable the DX NetOps Spectrum and UIM Integration from the OneClick Administration page.

NOTE
From 10.2.3, select the No integration option from the UIM Configuration page to disable the integration.

NOTE
You must disable UIM integration from the web server on which the integration is enabled.

Follow these steps (till 10.2.2):

1. Open the OneClick Administration page.
   The OneClick Administration page opens.
2. Click the Administration tab.
   Links to various OneClick web server configuration pages is displayed.
3. Click the UIM Configuration link in the left panel.
   The UIM Configuration window opens.
4. To disable the UIM integration, select Disable and click Save.
   The confirmation dialog appears.
   The following image displays the confirmation dialog that appears after disabling the integration:



5. To confirm, click Ok.
   UIM integration is disabled successfully.

NOTE
Wait for the all the UIM Host server models and the folder hierarchy to be cleared from the UIM Manager in the OneClick view after disabling the integration. To validate, search for any UIM Host Server models, using the search option.

Integrate with UIM for Virtualization Management

WARNING

From the 10.1.2 release, the SNMP Gateway probe and SBGW /Southbound gateway are no longer recommended for alarms synchronization from UIM to DX NetOps Spectrum. Please use the DX NetOps Spectrum Gateway (spectrumgtw) probe for this.

For more information about the integration of DX NetOps Spectrum with other CA products, see Integration Compatibility.

We recommend that you go through the DX NetOps Spectrum-UIM Integration using spectrumgtw probe - FACT SHEET to understand the changes to the integration whether you are new to this integration or an existing user.

This section describes how to set up the integration between the current releases of DX NetOps Spectrum and UIM for Virtualization Management. It also describes how to use its features to perform specific tasks such as synchronize inventory at regular intervals, manage alarms and events, and run reports.

The following video describes how you can integrate DX NetOps Spectrum and UIM for Virtualization Management to monitor virtual datacenters and to use DX NetOps Spectrum root cause analysis feature to rapidly troubleshoot and resolve issues:

Prerequisites for DX NetOps Spectrum and UIM Integration for Virtualization Management

Consider the following prerequisites for the integration:

WARNING

If you have enabled DX NetOps Spectrum and UIM integration for server management in 9.4, then while upgrading to 10.1.2, you must configure the latest UIM REST API (nisapi_WASP).

If you are integrating UIM with CA Spectrum 10.1.1 (or earlier), follow these steps:

1. 8.0.4 probe must be downloaded and configured before enabling the integration.
2. To migrate nisrest to nisapi, take the following steps:
   a. Stop svchost service.
   b. Navigate to C:\Program Files (x86)\CA UIM\probes\service\service_host\catalinaBase\webapps
   c. Remove nisrest.war and the nisrest directory
   d. Add the latest nisapi to archive.
   e. Deploy nisapi and restart svchost service.

   NOTE

   Migrating nisrest to nisapi is a one-time activity that is performed before enabling the integration.
3. Verify that the VMware (**vmware**) and SNMP Gateway (**snmpgtw**) probes are configured before enabling the DX NetOps Spectrum and UIM integration.
4. All the primary SpectroSERVERs should be up and running while enabling the DX NetOps Spectrum and UIM integration.
5. Enable QoS metrics and alarms in UIM for DX NetOps Spectrum OneClick views and condition-correlation to work.
6. UIM integration must be enabled on the OneClick server, which is configured to Main Location Server (MLS) (preferably server which does not have SRM).
7. DX NetOps Spectrum models virtual machines (VMs) whose PrimaryDNSName and PrimaryIPV4address attribute data are not reported to DX NetOps Spectrum by UIM. For UIM to report these attributes, deploy VMware Tools on all the VMs.

   NOTE

   In previous releases of UIM, the nisapi_service_host package was deployed to the service_host probe. As of CA UIM release 8.47, the service_host probe was deprecated and its functionality was moved to the wasp probe.

   - By default**,** the wasp probe uses port 80. However, because the wasp probe is a core component in UIM, this port should already be open in a properly functioning UIM environment.

Deploy and Configure Probes

To optimize the DX NetOps Spectrum and UIM integration, deploy and configure the following UIM probes on UIM server:

- **snmpgtw** Sends traps from UIM to DX NetOps Spectrum. The SNMP gateway converts alarms to SNMP trap messages that are readable by any SNMP-based event manager.
- **vmware (VMware Monitoring)** The vmware probe automates all common monitoring and data collection tasks, and lets you focus on problems as they arise. The vmware probe collects and stores data and information from the monitored components through VMware Virtual Center (vCenter), or by connecting to ESX servers directly, at customizable intervals. You can define alarms to be generated when specified thresholds are breached or when the service is unavailable.
- **net_connect** The net_connect (Network Connectivity Monitoring) probe measures network connectivity that is based on "ping" (ICMP ECHO) and the TCP connections to a list of user-defined services. The service can be NetBIOS, Telnet, FTP, and HTTP. The probe supports the UIM family of solutions by sending quality of service (QoS) messages.
- **(nisapi) service_host or wasp** Queries the CA Nimsoft Manager using the Restfull Services API to retrieve the list of UIM models to be monitored. Deploy service_host v8.04/ wasp v8.4.2 or higher.
- **cdm** The CPU, Disk, Memory Performance Monitoring (cdm) probe monitors the performance and resource load on the system with the robot. The UIM CPU, Disk & Memory (cdm) probe generates alarms that are based on configured threshold values and trending statistics.

The probe provides the following benefits:
  - Generate alarms that are based on configured threshold values. The probe generates alarms that can trigger corrective actions immediately.
  - Generate trending Quality of Service (QoS) data. The information is measured and sent to the data_engine probe to process and store in the UIM database. The historical data facilitates capacity planning for monitored systems in the IT environment. Some examples are as follows:
    - View how disks are filling up over time
    - Plan batch jobs according to the CPU utilization
    - Upgrade systems which consistently operate near capacity
- **spectrumgtw** The spectrumgtw (DX NetOps Spectrum Gateway) probe provides a bidirectional (UIM to DX NetOps Spectrum and DX NetOps Spectrum to UIM) integration between CA Unified Infrastructure Management and DX NetOps Spectrum enabling you to perform the following actions:
  - Export and correlate DX NetOps Spectrum Inventory from a Global Collection to UIM inventory.
  - Bi-directional synchronization of alarms
  - Bi-directional Clear / Update of alarms
  - View symptomatic DX NetOps Spectrum alarms in the context for the root-cause alarm in UIM
  - Ticket ID synchronization between DX NetOps Spectrum and UIM

    WARNING

    From 10.2.1, all UIM events (apart from events related to VMWare and servers) are no longer considered as generic events.
    Specific events raised on UIM from the following probes are mapped to unique events created in DX NetOps Spectrum. This allows you to manage the events (raised on UIM) from DX NetOps Spectrum more efficiently.

Additional event mapping for select metrics from the following probes (along with cdm and net_connect probes) is added from 10.2.1. Additional event mapping for Please refer to Additional event mapping from UIM Probes to DX NetOps Spectrum.

- **logmon** The Log Monitoring (logmon) probe scans ASCII-based systems and application log files by matching specified expressions. Alarms are generated when the log file content matches the defined expression.
- **rsp (Remote System probe)** The Remote System Probe (rsp) allows you to monitor system metrics. The probe collects performance data in an agentless manner without installing proprietary software on the system.
- **sqlserver** The SQL Server Monitoring (sqlserver) probe constantly monitors the internal performance and space allocation of SQL Server databases. The probe can run locally on the database server or it can be configured to run as a remote client. The probe feeds essential information that is based on predefined criteria to the UIM availability manager for appropriate alert notification, as required. An extensive range of checkpoints can be selected and individually scheduled to meet your monitoring requirements. The probe will run selected SQLs to extract vital information about your SQL Servers. The information is presented to database-administrator as alarms or reports.
- **processes (Process Monitoring)** The (Process Monitoring) processes probe monitors the specified processes to detect any error situation. The probe also retrieves information about the process, for example, the CPU usage, memory usage, and so on.

You can configure the UIM probes through the Probe Configuration interface. For more information, see the CA Unified Infrastructure Management Probe Space.

How to Integrate DX NetOps Spectrum and UIM for Virtualization Management

From the OneClick Administration page, you can enable the integration between DX NetOps Spectrum and UIM for Virtualization Management. When you enable the integration all the virtual entities managed by UIM appear in DX NetOps Spectrum.

NOTE

If you want to change any details on the Administration page, refresh the page before making any changes.

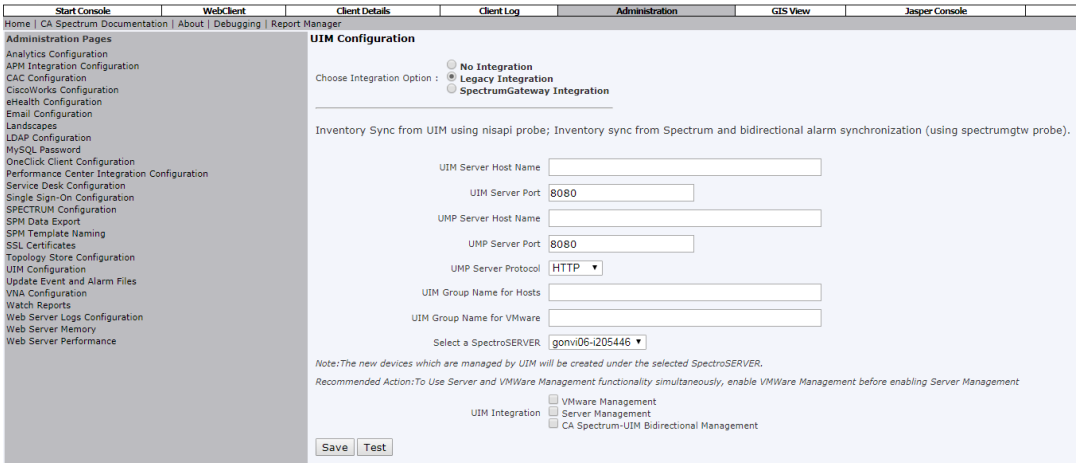
WARNING

- If you enable the integration for VMware Management, all the entities that are managed by Virtual Host Manager (VHM) such as VMware, Hyper-V, Zones, Huawei SingleCloud, and LPAR through CA Virtual Application Insight Module (VAIM) are disabled permanently and all the existing VHM models are deleted. You must reinstall DX NetOps Spectrum for CA VAIM based Virtualization Management.
- From 10.2.3, DX NetOps Spectrum can monitor both UIM models and SystemEdge models when DX NetOps Spectrum-UIM Integration is enabled with Multi-tenancy. When the integration is enabled with Multi-tenancy, UIM origins are mapped to DX NetOps Spectrum landscapes. In this scenario, DX NetOps Spectrum monitors UIM models on the landscapes which are part of Multi-tenancy mapping and can monitor SystemEdge AIM based models on the landscapes which are not part of Multi-tenancy mapping.When the DX NetOps Spectrum-UIM Integration is enabled without Multi-tenancy, DX NetOps Spectrum monitors UIM models only. All the SystemEdge AIM based models are deleted on all the landscapes.

Follow these steps:

1. Open the OneClick Administration page.
   The OneClick Administration page opens.
2. Click the Administration tab.
   Links to various OneClick web server configuration pages is displayed.

   1. a. Click the UIM Configuration link in the left panel.
      The following image illustrates the configuration options that are available in the UIM Configuration window:



      If the integration was already enabled before the upgrade, then the **Legacy Integration** radio button would appear as selected by default.

      UIM Server Host Name Indicates the IP address/hostname of the UIM Server.

      UIM Server Port Indicates the server port number of UIM.

      UMP Server Host Name Indicates the IP address/hostname of UMP.

      UMP Server Port Indicates the server port number of UMP.

      UMP Server Protocol Indicates the network protocol to connect to the CA UMP server.

      NOTE

      From spectrumgtw 8.67 onwards, the UIM Group Name field in the UIM configuration section has been changed to Group Name for Hosts.

UIM Group Name for Hosts: Specify the UIM group name in this field for selective inventory synchronization of hosts (of the UIM group specified) from UIM to DX NetOps Spectrum. This is applicable to Server Management entities only.

UIM Group Name for VMware: Specify the UIM group name in this field for selective inventory synchronization of VMware entities (of the UIM group specified) from UIM to DX NetOps Spectrum. To address the challenges

in providing filtering of VMware data from UIM to DX NetOps Spectrum, a 'UIM Group Name for VMware' field is introduced from spectrumgtw v.8.67.

1. a.

NOTE

1. Ensure that you add vCenter and ESX hosts and the corresponding virtual machines to be synced to DX NetOps Spectrum.

2. If the vCenter is not part of the group, ESX and VMs are not modeled in DX NetOps Spectrum even though they are part of the group.

3. If vCenter and VMs are part of the group and ESX is not, then the VMs are created without any relationship with the vCenter in DX NetOps Spectrum.

NOTE

Inventory from the group you specify in this field is synchronized from UIM to DX NetOps Spectrum.

NOTE

For successful encrypted communication (https protocol) between DX NetOps Spectrum and UIM UMP Server, you must import SSL Certificate of UIM UMP Server into DX NetOps Spectrum OneClick Server. To import the certificate, from the OneClick home page select **Administration**, **SSL Certificates**, upload the certificate file then click **Save**. Restart the OneClick web server for the changes to take effect.

WARNING

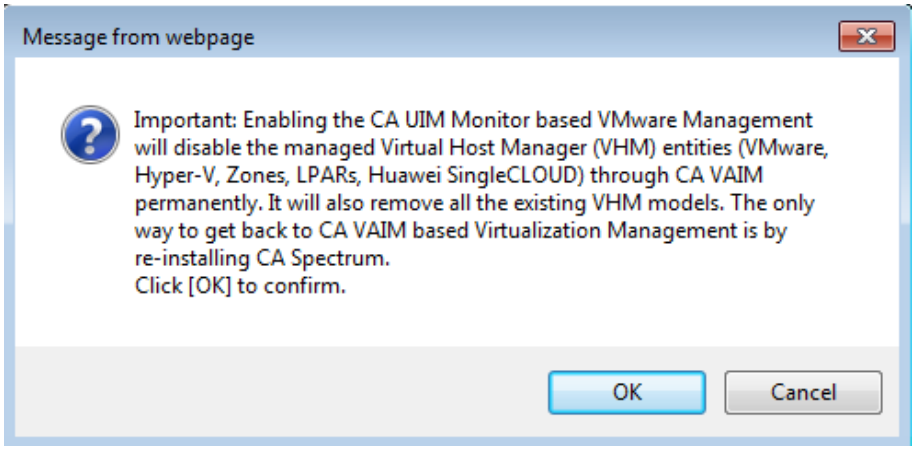
Inventory from the group you specify in this field is synchronized from UIM to DX NetOps Spectrum. If you leave the UIM/UMP Group Name field blank, all inventory from the specified UIM server is synchronized with DX NetOps Spectrum.

3. Select the SpectroSERVER under which the new models for devices managed by UIM are to be created.
4. To enable the UIM integration for VMware, select VMware Management. (**Till 10.2.2 and spectrumgtw probe v8.65**)

NOTE

If you are setting up the integration using 10.2.3 and spectrumgtw probe v8.65, you can no longer enable/disable the VMware Management capability from the UIM Configuration page. To enable and configure the VMware Management capability for VMware, please refer to the spectrumgtw AC documentation.

The following warning message appears:



5. To confirm, click **OK.**

WARNING

In the OneClick Admin page, do not enable or disable the '**VMware Management**' or '**Server Management**' options when the MLS is down. This may cause unexpected errors.

In the DX NetOps Spectrum OneClick console, do not create any container of type '**UIMInventoryContainer**'.

6. Click **Test.** If the test is successful, "**Test connection to UIM Server was successful**", the message appears.
7. Click **Save.**Successfully saved configuration to the database message appears.
   UIM integration is now enabled.

NOTE

If you change any settings on any of the OneClick server other than the integration OneClick server. The message, "Saving details on this OC will change the Integration OC Server. Click [OK] to continue" appears. If you click Save, the integration OneClick server changes and the details are saved. If you click Cancel, the details are saved but the integration OneClick server remains same.

WARNING

If the Main Location Server (MLS) in a Distributed SpectroSERVER (DSS) setup is down and if it does not have a secondary SpectroSERVER, the configuration information that is specified in the UIM Configuration page fails to save to the database. We recommend that you use the Fault Tolerance (FT) setup for MLS to avoid any inconsistent behavior.

WARNING

- If the UIM-DX NetOps Spectrum integration is enabled in a FT setup, it is recommended that you synchronize the Primary and Secondary SpectroSERVERs after the first successful synchronization. This will ensure that you have the integration data backed up in case the Primary/ dedicated SS (specified in the OneClick Admin > UIM Configuration) goes down.
- It is also recommended that you synchronize the Primary and Secondary SpectroSERVERs, after a successful synchronization job, if you are moving Datacenters from a MLS to a Non-MLS landscape.

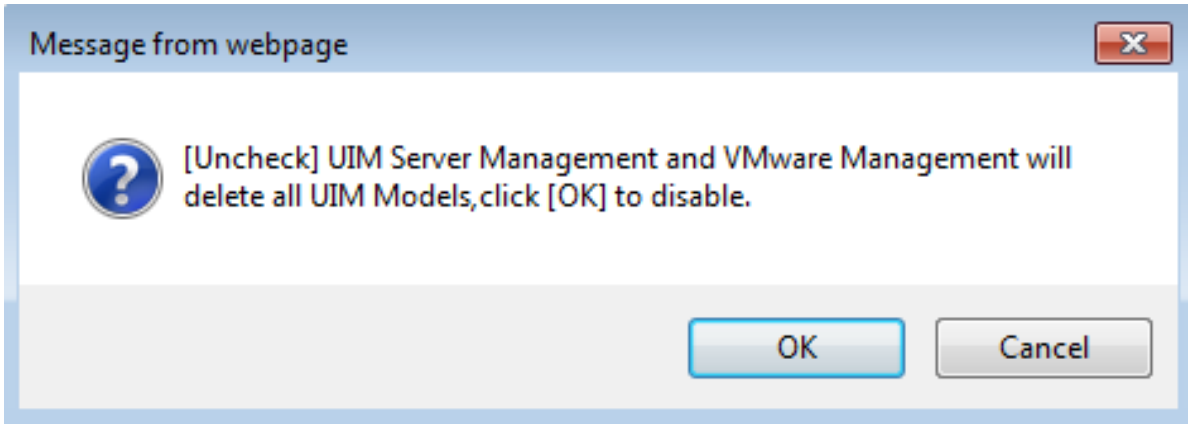
Disable the Integration

You can disable the DX NetOps Spectrum and UIM Integration for Virtualization Management from the OneClick Administration page.

NOTE
From 10.2.3, select the No Integration option from the UIM Configuration page to disable the integration. To disable only VMware integration capabilities you need to disable the VM Server Integration from the VMware Configuration section of the spectrumgtw probe Admin Console. For more information, please refer to the spectrumgtw Admin Console documentation.

Follow these steps(Till 10.2.2):

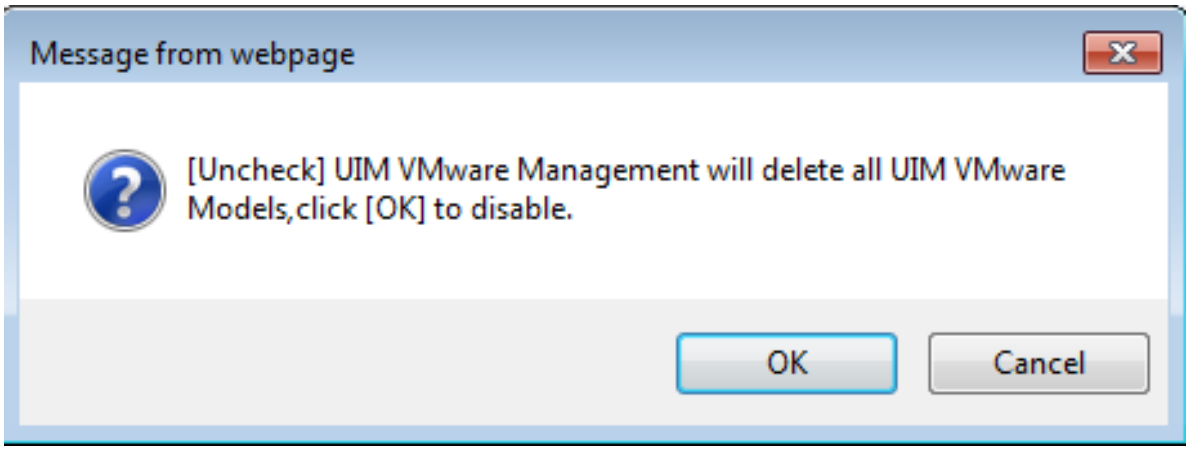
1. Open the OneClick Administration page.
   The OneClick Administration page opens.
2. Click the Administration tab.
   Links to various OneClick web server configuration pages is displayed.
3. Click the UIM Configuration link in the left panel.
   The UIM Configuration window opens.
4. Consider the following scenarios to disable the UIM integration:
5. Clear both Server and VMware Management options and click **Save**.
   The following message appears:



6. Click Ok, DX NetOps Spectrum and UIM integration for both Server Management and Virtualization Management is disabled.

   NOTE
   Wait for the all the UIM Host server models and the folder hierarchy to be cleared from the UIM Manager in the OneClick view after disabling the integration. To validate, search for any UIM Host Server models, using the search option.

7. If you clear the VMware Management checkbox and click **Save.**
   The following message appears:



8. Click OK.
   DX NetOps Spectrum and UIM integration for Virtualization Management is disabled.

Supported Features (UIM Integration)

This section briefly describes the following features that are supported in DX NetOps Spectrum and UIM integration for Virtualization Management:

Models Created for VMware (UIM Integration)

UIM Manager includes the following models and icons for VMware devices:

VMware vCenter Server

Represents a physical or virtual host that contains the vCenter application to manage your VMware virtual environment.

Icon:

ESX Host

Represents an ESX host, as configured in your VMware virtualization technology. An *ESX host* is a physical computer that uses ESX Server virtualization software to run virtual machines. Hosts provide the CPU and memory resources that virtual machines use and give virtual machines access to storage and network connectivity. In the Universe topology, these models group your virtual entities into a separate view while showing how the virtual environment interacts with the physical network. The ESX host cannot be contacted directly for status information. Instead, the status of these models is inferred from the status of the items that it contains.

Icon:

Virtual Machine

Represents a virtual machine, as configured in your VMware virtualization technology. A *virtual machine (VM)* is a software computer that, like a physical computer, runs an operating system and applications. A virtual machine dynamically consumes resources on its physical host, depending on its workload. Because virtual machines are flexible computing units, their deployment comprises a wide range of environments. Examples include environments such as data centers, cloud computing, test environments, or desktops and laptops. In data center implementations, they are used for server consolidation, workload optimization, or higher energy efficiency.

Icon:

Nimsoft Manager also creates models these additional VMware entities that organize the ESX hosts and their virtual machines:

Datacenters

Represents a datacenter, as configured in your VMware virtualization technology. A *datacenter* serves as a container for your hosts, virtual machines, resource pools, or clusters. Depending on their virtual configuration, datacenters can represent organizational structures, such as geographical regions or separate business functions. You can also use datacenters to create isolated virtual environments for testing or to organize your infrastructure. Components can interact within datacenters, but interaction across data centers is limited. A datacenter can contain clusters or hosts.

Icon:

Clusters

Represents a cluster, as configured in your VMware virtualization technology. A *cluster* is a group of ESX hosts and their associated virtual machines. When a host is added to a cluster, the host resources become part of the cluster resources. The cluster manages the resources of all hosts within it. A cluster can contain hosts, resource pools, or virtual machines.

Icon:

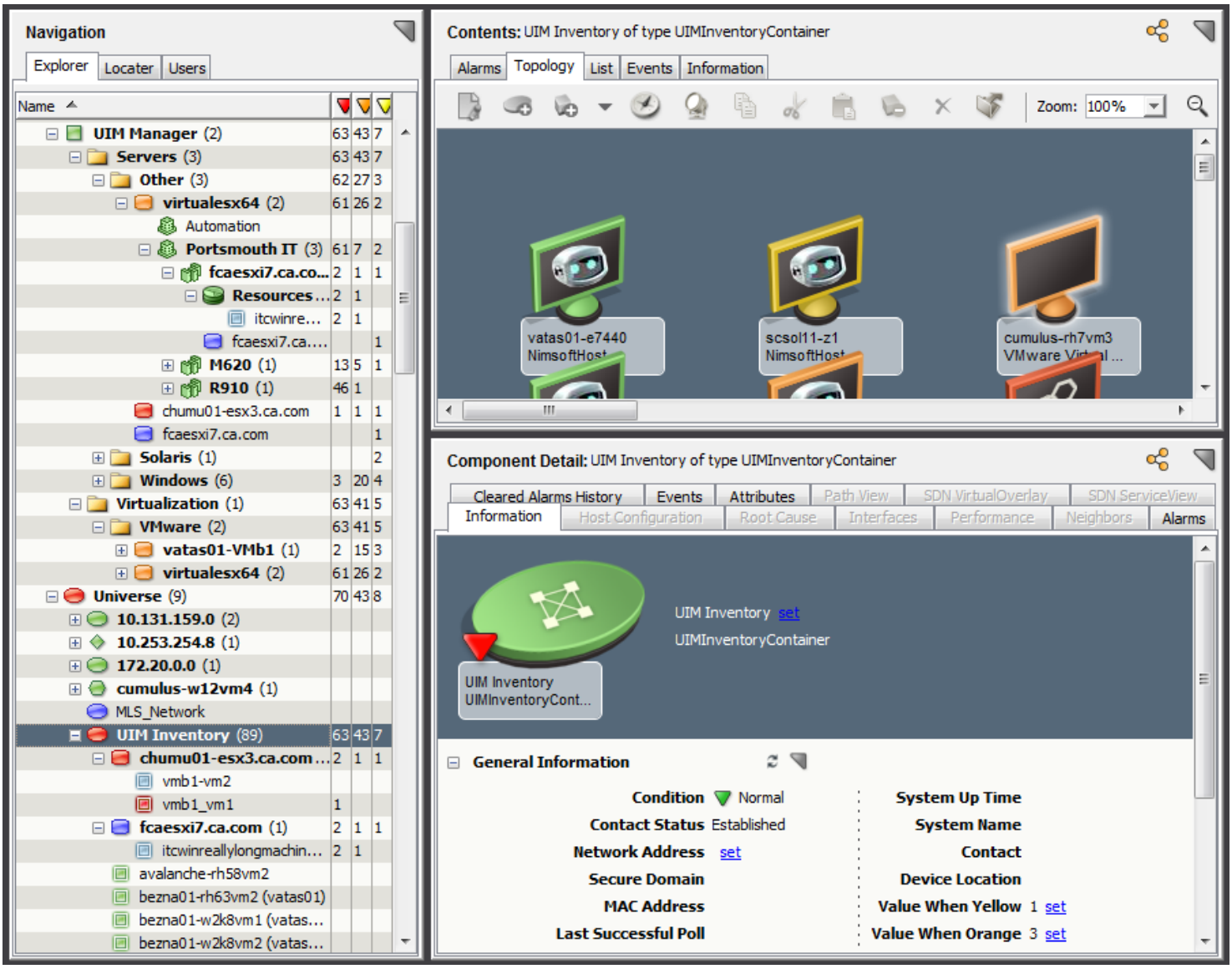
Resource Pools

Represents a resource pool, as configured in your VMware virtualization technology. A *resource pool* defines partitions of physical computing and memory resources of a single host or a cluster. You can partition any resource pool into smaller resource pools to divide and assign resources to specific groups or for specific purposes. You can also hierarchically organize and nest resource pools. A resource pool can contain virtual machines or more resource pools.

Icon:

Hierarchy in the OneClick Navigation Page

You can view the virtual entities such as Virtual Center (VC), and Virtual Machines (VM)s that are created for VMware under **UIM Manager** node in the OneClick console. Logical entities such as Clusters, Data Centers, and Resource Pools can also be viewed in the OneClick Navigation page. A new container '**UIM Inventory**' is created under the Universe view, this container will have the ESX Hosts, Nimsoft Hosts, Vcenter servers and corresponding VMs. ESX hosts are also displayed under the **UIM Manager > Servers** hierarchy. In the Explorer View, **UIM Manager** provides a more detailed hierarchy i.e. ***VCenters > Data Centers > Clusters > Resource Pools > VMs*** , compared to the **UIM Inventory** view, which only displays the hierarchy of ESX Hosts and associated VMs.



ESX Hosts are placed under the UIM Manager hierarchy and in the Universe > UIM Inventory hierarchy, in the OneClick console. To organize the Universe hierarchy, move(cut/paste) the ESX hosts to other containers within the Universe.

NOTE

Do not move containers before running a synchronization job, else, duplicate models are created in DX NetOps Spectrum.

NOTE

If you move (cut/paste) the UIM inventory container model frequently, (and VMs within the UIM inventory Container model are connected to upstream switches), the pipes (connectors) might appear in black color.

To fix the black pipes (connectors) error, follow these steps:

1. Navigate to ***$SPECROOT/tomcat/webapps/spectrum/WEB-INF/web.xml***
2. In the web.xml file, you need to add the below context:

```
<context-param xmlns="">
```

```
<param-name>com.aprisma.topo.Topology.PipeUpdateTardiness</param-name>
<param-value>1000</param-value>
<description>
This parameter defines the amount of time in seconds after which a pipe connection is established.
</description>
</context-param>
```

If you already you have the above context in the web.xml file, ensure that the param-value is greater than or equal to 1000.

Distribution of Virtual Center (VC) Across Landscapes

In a Distributed SpectroSERVER (DSS) environment, DX NetOps Spectrum, and UIM integration has the capability to distribute the Virtual Centers (VCs) across landscapes rather modeling all the VCs on the selected landscape which helps to show the connectivity (layer 2 or 3) between the Virtual Machine (VM) and upstream devices. If no upstream devices are available, the VC and all the entities that are managed by that VC are modeled in the dedicated SpectroSERVER.

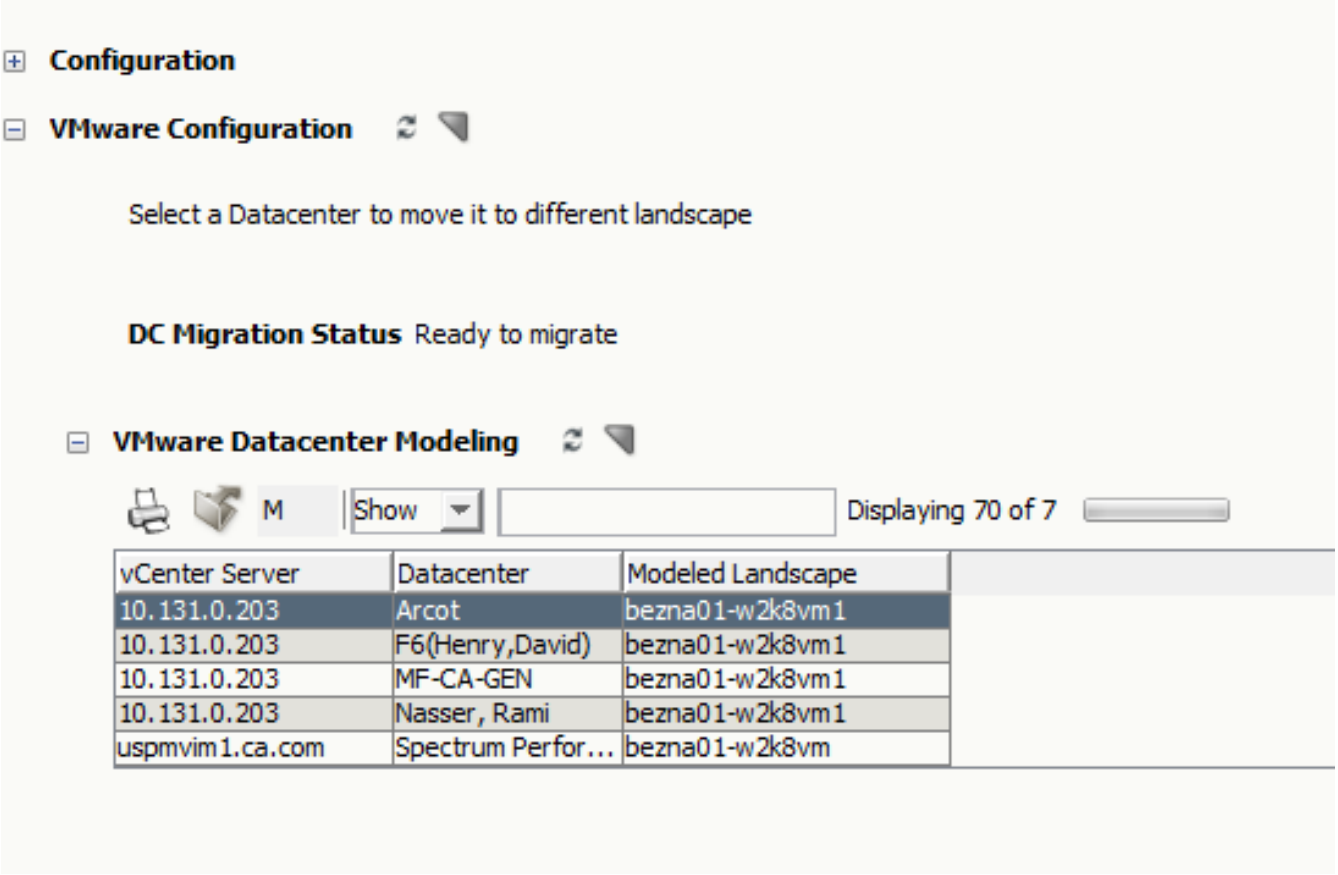
Migrating DataCenters across Landscapes

You can move Datacenters from an existing landscape to a new landscape where the upstream devices are discovered. If a Datacenter migration is in-progress, you cannot migrate another DataCenter. Also, if a virtualization sync is in-progress, the Datacenter migration is not allowed.

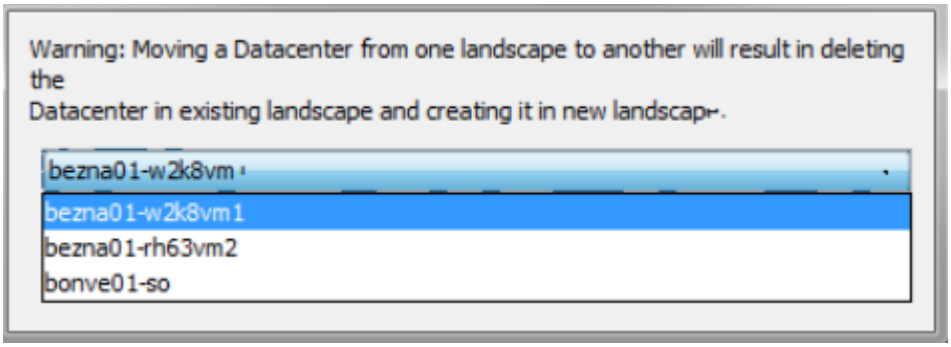
> **NOTE**
>
> If you move a Datacenter from one landscape to another, it results in the deletion of Datacenter from the previous landscape. Any existing events or alarms that are present on the Datacenter and its entities are lost.

Follow these steps:

1. Open the DX NetOps Spectrum OneClick console.
2. Select Nimsoft Manager and select the Information tab on the Content pane.
   VMware Configuration information is displayed.
3. Expand VMware Configuration and VMware Datacenter Modeling.
   A list of datacenters and the modeled landscapes are displayed.
   The following image displays the table with a list of Datacenters and the modeled landscapes:



4. Select a Datacenter and select Move.
   The Select Landscape dialog appears.
   The following image shows the list of landscapes that are available in DX NetOps Spectrum:



5. Select a landscape and select Ok.
   The confirmation window opens.
6. Select Ok.
   The datacenter is moved to the selected landscape successfully.

NOTE

If Datacenters are migrated from landscapes that are down, then the migrated datacenter and its entities are deleted from the old landscape, during the subsequent sync, once that landscape is up.

Migrating Multiple DataCenters across Landscapes Via REST API

Previously, as a user, you could only migrate one Datacenter a time from one landscape to another using the OneClick console. Each time the user migrated the Datacenter, triggering the discover connection, caused a delay in completing the migration. To address this challenge, 10.3.1 introduces a new feature under the DX NetOps Spectrum-UIM integration, to migrate multiple Datacenters at a time from an existing landscape to a new landscape. To achieve this execute a post request through a REST call with the required information payload. The new approach triggers discover connections only once during the migration of multiple Datacenters.

GET Request:

GET action issues an action request to OneClick Server to get complete Datacenters information across landscapes using the get request via REST API.

GET URL

Open any REST Client and perform the below GET request.

```
http://<OC server Name>:<Port number>/spectrum/restful/datacenters
```

Sample Output:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<datacenter-response status="Success">
<dc-elements>
<vcid>62</vcid>
<vcip>10.131.159.5</vcip>
<dcid>C566c8be11d7e9700a63d2cdf5579df06</dcid>
<dcname>ITC-SPECTRUM</dcname>
<currentlandscape>0xc00000</currentlandscape>
<newlandscape>0xc00000</newlandscape>
 </dc-elements>
<dc-elements>
<vcid>109</vcid>
<vcip>138.42.93.28</vcip>
<dcid>Ca0a997f0b109e2e542d8699eb5f718b3</dcid>
<dcname>Engineering</dcname>
<currentlandscape>0xc00000</currentlandscape>
<newlandscape>0xc00000</newlandscape>
</dc-elements>
</datacenter-response>
```

In the above response, each <dc-elements> tag contains an individual data center information. In case you have ten Datacenters, in the response, you get ten <dc-elements> with appropriate details. If you want to move any datacentre from one landscape to another, simply copy the <dcelements> of that specified Datacenter and update the new landscape field name in <dc-elements> with new landscape information.

POST Request:

POST URL

Open any REST Client and perform the below POST request with needed payload information.

```
http://<OC server Name>:<Port number>/spectrum/restful/dcmigration
```

Sample Post Payload:

This payload has the same <dc-elements> information that we received from the get request with an updated new landscape name where you have to move the Datacenter. Following is an example of migrating two data centers from landscape 0xc00000 to another landscape 0x1800000.

```
<?xml version="1.0" encoding="UTF-8"?><datacenter-migration-info xmlns="http://www.ca.com/spectrum/restful/
schema/request">
<dc-migration-list>
<dc-elements>
<vcid>62</vcid>
<vcip>10.131.159.5</vcip>
<dcid>C566c8be11d7e9700a63d2cdf5579df06</dcid>
<dcname>ITC-SPECTRUM</dcname>
<currentlandscape>0xc00000</currentlandscape>
<newlandscape>0x1800000</newlandscape>
</dc-elements>
<dc-elements>
<vcid>109</vcid>
<vcip>138.42.93.28</vcip>
<dcid>Ca0a997f0b109e2e542d8699eb5f718b3</dcid>
<dcname>Engineering</dcname>
<currentlandscape>0xc00000</currentlandscape>
<newlandscape>0x1800000</newlandscape>
</dc-elements>
</dc-migration-list>
</datacenter-migration-info>
```

Schedule Virtualization Sync

Contents

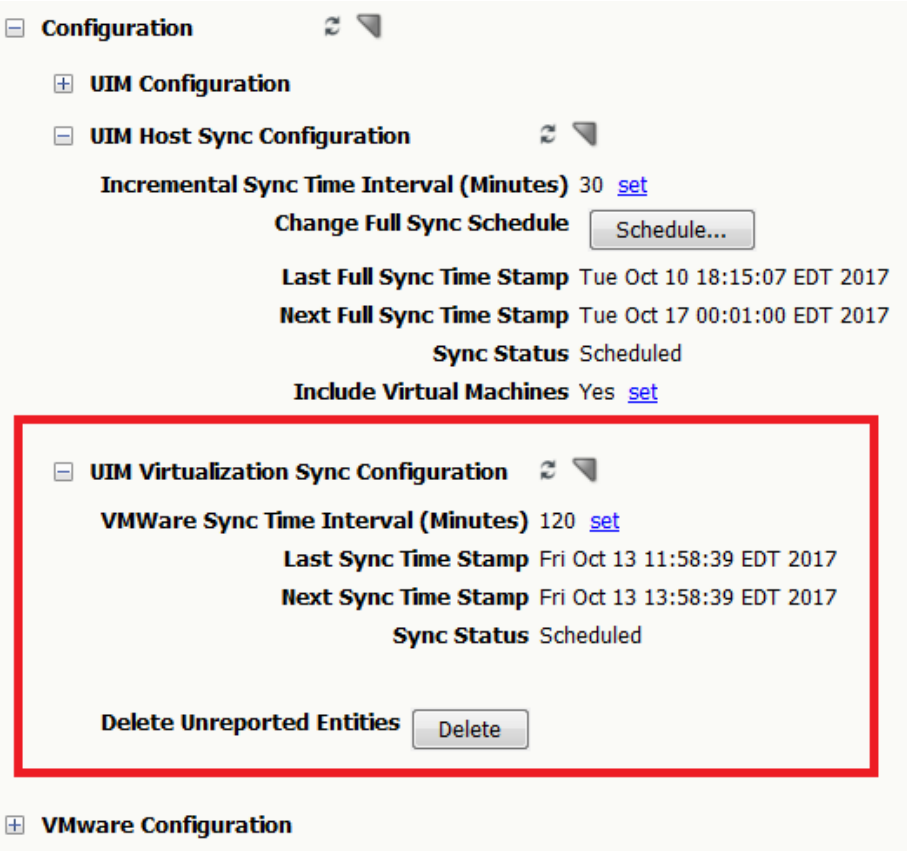
Overview

This integration supports Virtualization sync. When the DX NetOps Spectrum and UIM integration is enabled, synchronization happens automatically at the scheduled time interval displayed in the OneClick view. Additions, deletions and modifications of VMWare entities in UIM are reflected in DX NetOps Spectrum after the Virtualization sync. You can schedule the Virtualization sync at regular intervals through the OneClick view.

Follow these steps:

1. Open the DX NetOps Spectrum OneClick console.
2. Select **UIM Manager** from the Navigation panel on any landscape.
   The **Contents** pane for UIM Manager opens.
3. Click the **Information** tab and expand **Configuration.**
4. Expand **UIM Virtualization Sync Configuration.**
   Information on Virtualization sync is displayed.

   The following image displays the option to schedule Virtualization sync:



5. Click set to schedule Virtualization sync.
   The Time interval window opens.
6. Specify the time interval and click **OK.**
   Default: 300 minutes.
   Minimum: 60 minutes.
   You can confirm whether the Virtualization sync is complete, based on **Sync Status** value.

Delete Unreported Entities

The **Delete Unreported Entities** button, allows you to delete entities/models that are no longer reported as part of virtualization integration.
The entities/ models that were previously reported only as part of virtualization integration will be placed in the Unreported Entities/ Models container, if they are no longer reported. (These entities/ models were auto-deleted previously. You will now have to manually delete these entities using the Delete Unreported Entities button).

1. Click **Delete**.
   A relevant confirmation dialog appears.
2. Select **Yes** to delete the entities/models from this container across landscapes.
   An appropriate confirmation message is displayed.

Reconciling UIM virtual entity data with existing DX NetOps Spectrum models

During virtualization synchronization, when a new UIM virtual entity is reported to DX NetOps Spectrum from UIM, DX NetOps Spectrum will perform a search to identify if this virtual entity was modeled during DX NetOps Spectrum discovery and modeling. If such an existing model is found, DX NetOps Spectrum will reconcile the UIM entity information with the existing model, instead of creating a new model.

This search is limited to the SpectroSERVER in which the parent VCenter UIM entity was destined to be modeled. In cases where an existing model is not found and reconciliation is not performed, DX NetOps Spectrum will model the virtual entity in the SpectroSERVER specified in the **OneClick Admin Configuration**> **UIM Configuration** page.

NOTE
DX NetOps Spectrum uses the IP Address reported by the entity to perform this search. If the IP Address is not reported, the search is performed using the reported MAC address.
If neither (IP or MAC Address) are reported, DX NetOps Spectrum will not be able to reconcile that entity even if the entity is discovered and modeled in DX NetOps Spectrum.

QoS Metrics

From the OneClick console, you can view the QoS VMware Metric information of the logical entities such as Nimsoft ResourcePool and Nimsoft Clusters and the virtual entities such as ESX Host and Virtual Machine (VM).

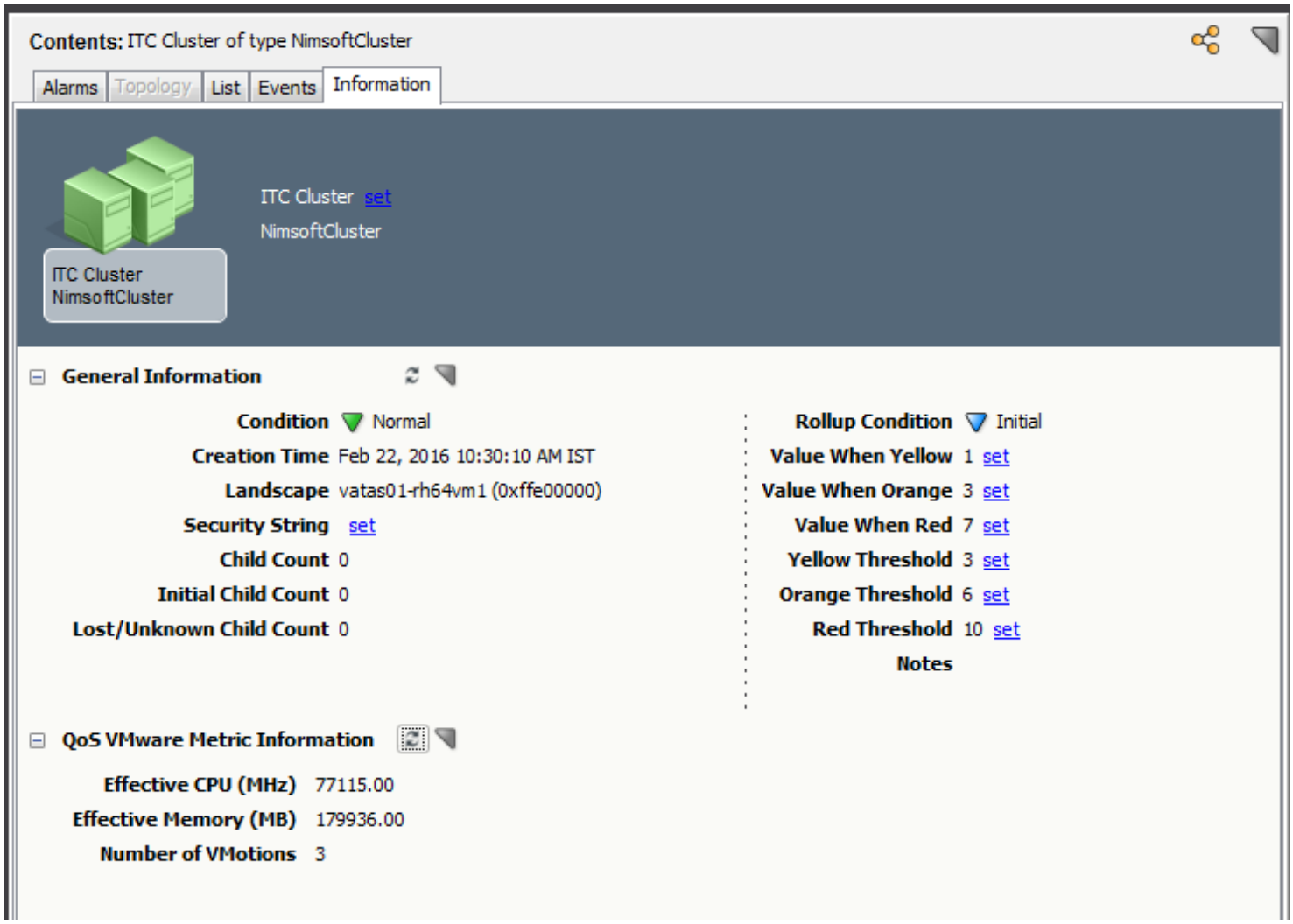
NOTE

You can also view the QoS CPU/Disk/Memory Metric Information for ESX and VMs. Verify that the QoS metrics related monitors are configured on the respective probes such as Vmware and CDM.

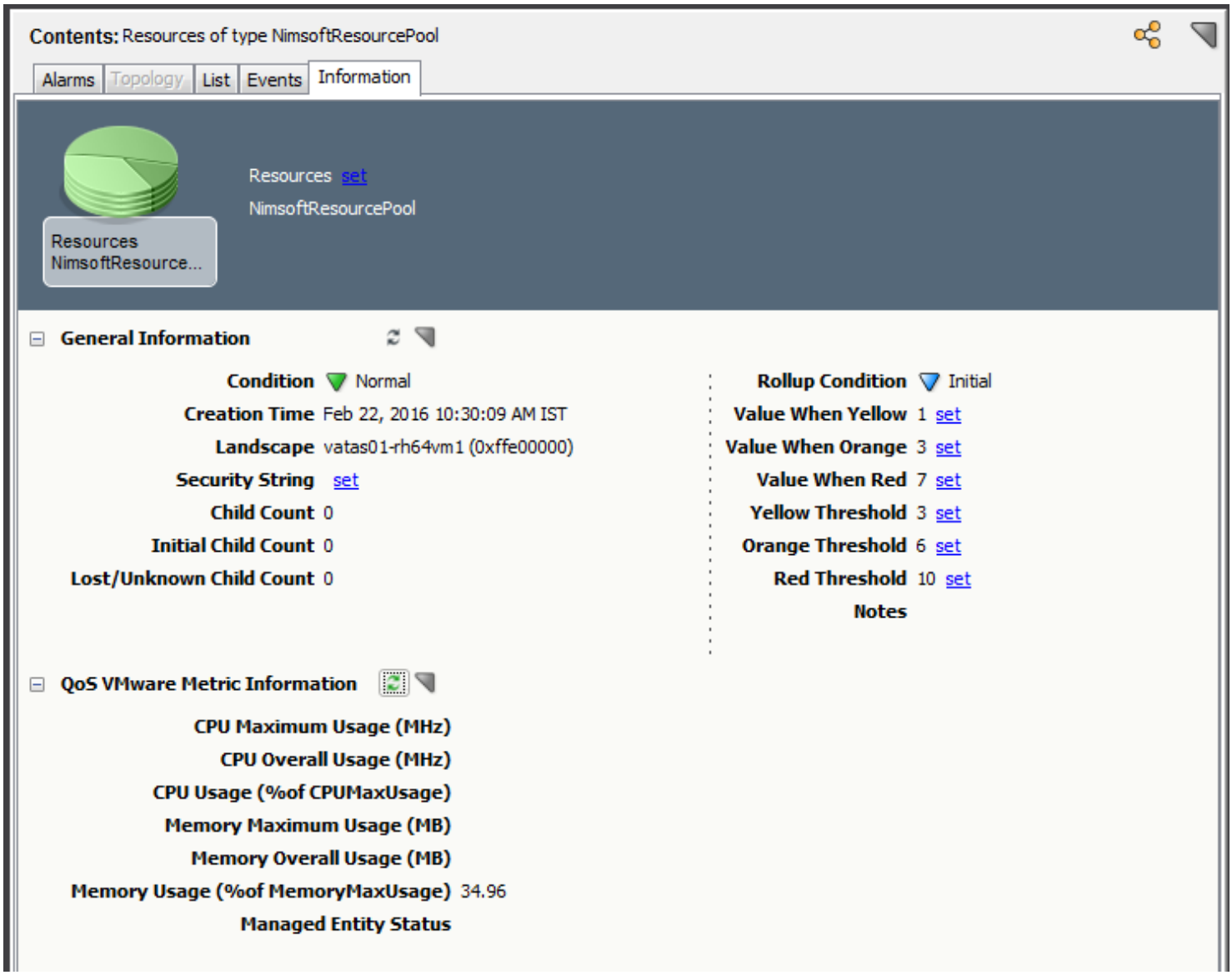
From r10.2.1 you can view QoS metrics information for AWS Cloud (Amazon Web Services) and Microsoft Azure.

Follow these steps:

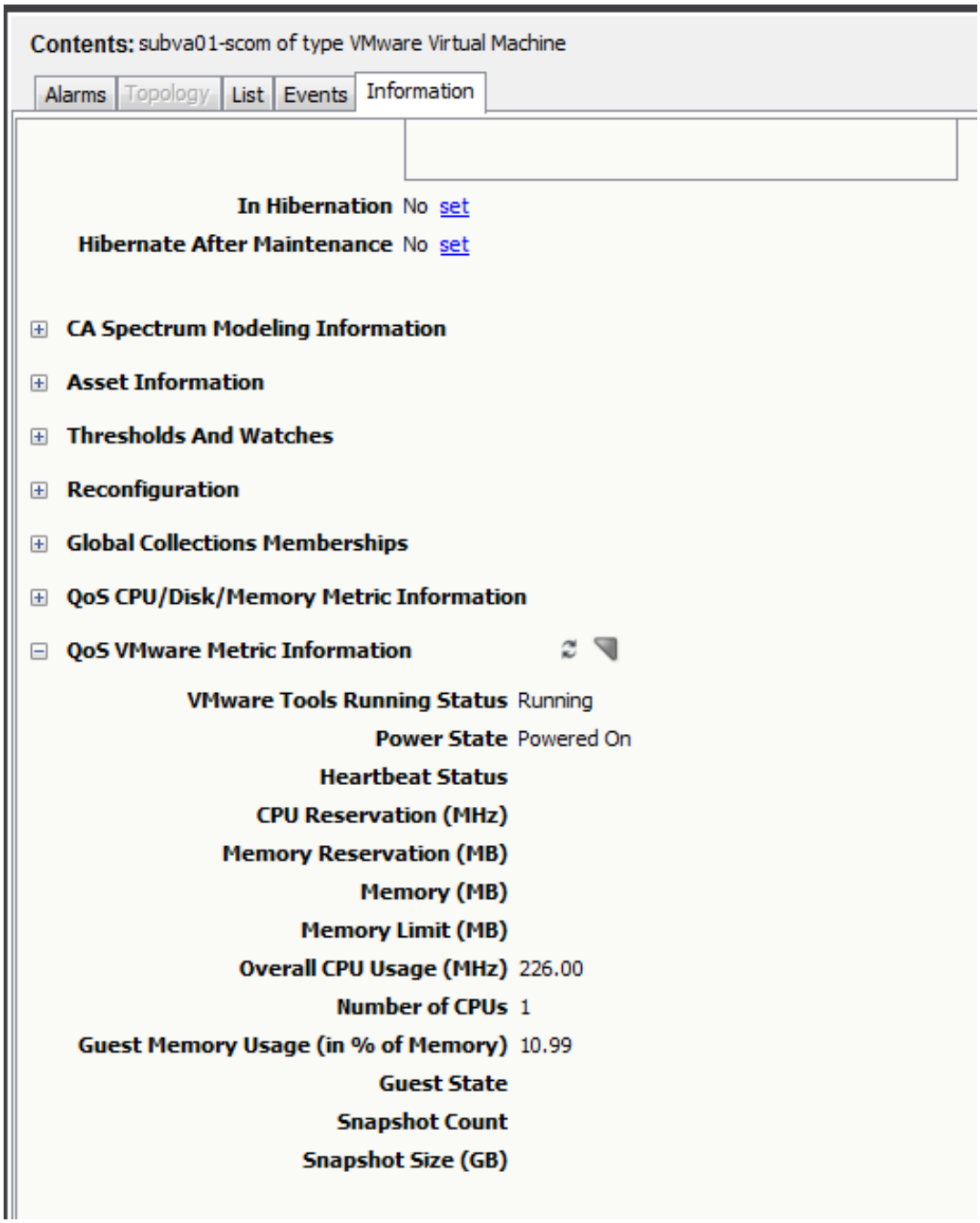
1. Open the DX NetOps Spectrum OneClick console.
2. Select **UIM Manager**, **Virtualization, VMware** and click **Cluster** from the Navigation panel.
   The **Contents** pane opens.
3. Click the **Information** tab.
   NimsoftCluster information is displayed.
4. Expand **QoS VMware Metric Information**.
   QoS Metrics for the selected Cluster is displayed.
   The following image displays the QoS VMware Metric Information of a NimsoftCluster:



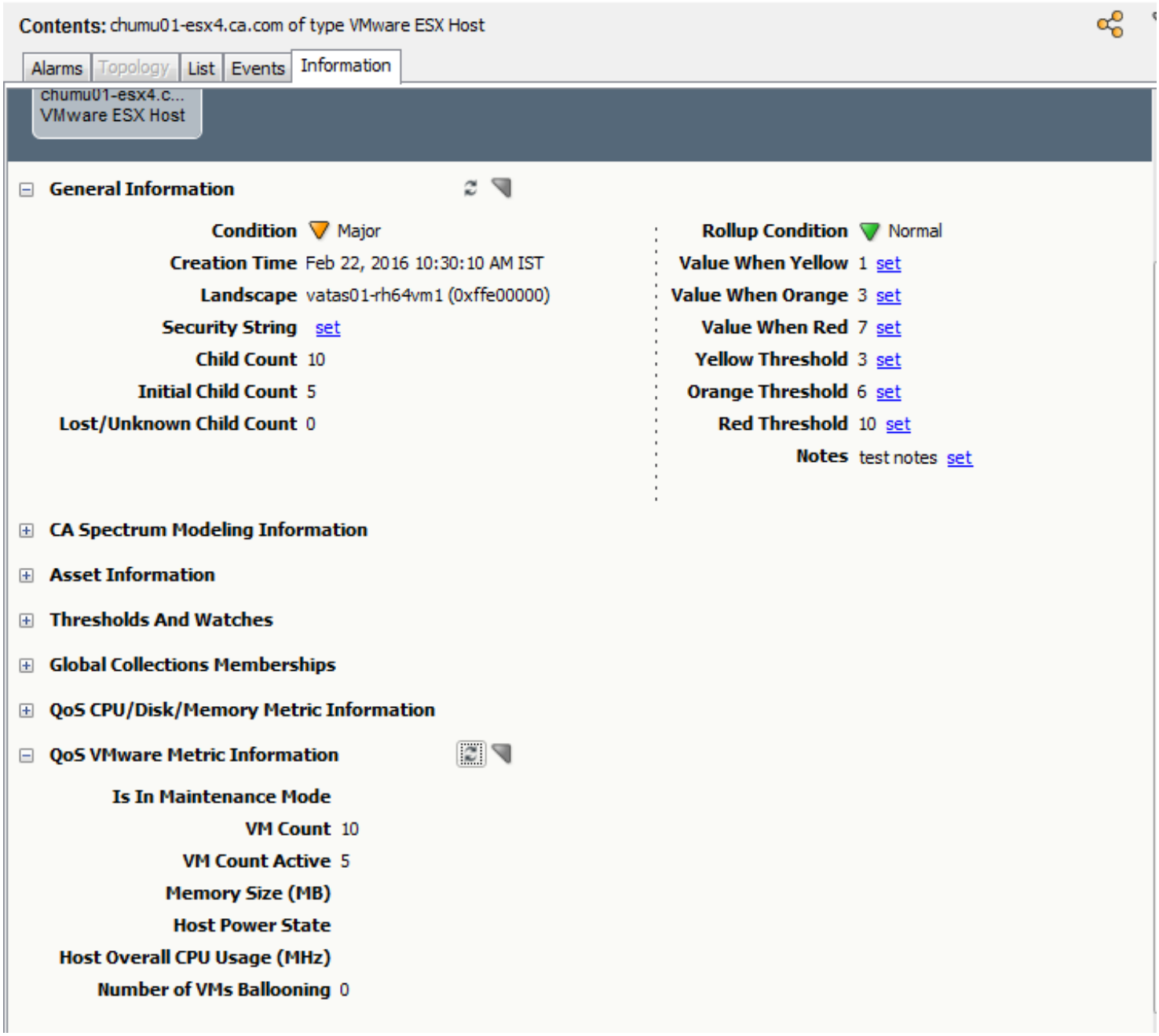
5. **To view QoS VMware Metric Information of a ResourcePool,** select a ResourcePool from the Navigation Panel.
   The **Contents** pane opens.
6. Click the **Information** tab.
   QoS Metrics for the selected NimsoftResourcePool information is displayed.
   The following image displays the QoS VMware Metric Information of a NimsoftResourcePool:



7. To view the QoS VMware Metric Information of a VM, select a VM that is available in Resources from the Navigation Panel.
   The **Contents** pane opens.
8. Click the **Information** tab.
   QoS Metrics for the selected VM is displayed.
   The following image displays the QoS VMware Metric Information of a VM:

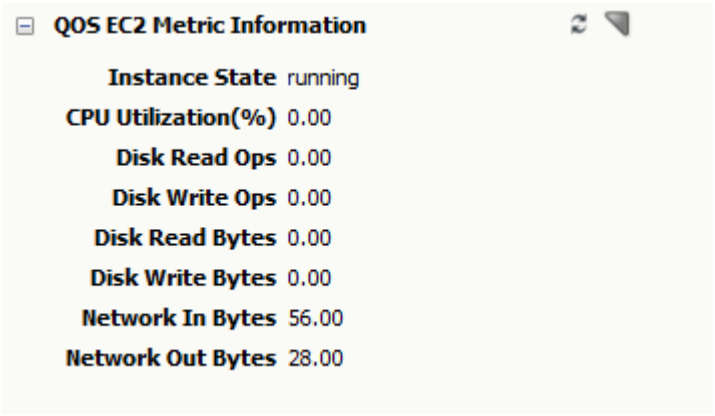


9. To view the QoS VMware Metric Information of an ESX Server, select an ESX or ESXi server that is available in Clusters from the Navigation Panel.
   The **Contents** pane opens.
10. Click the **Information** tab.
    QoS Metrics for the selected ESX server is displayed.
    The following image displays the QoS VMware Metric Information of an ESX server:



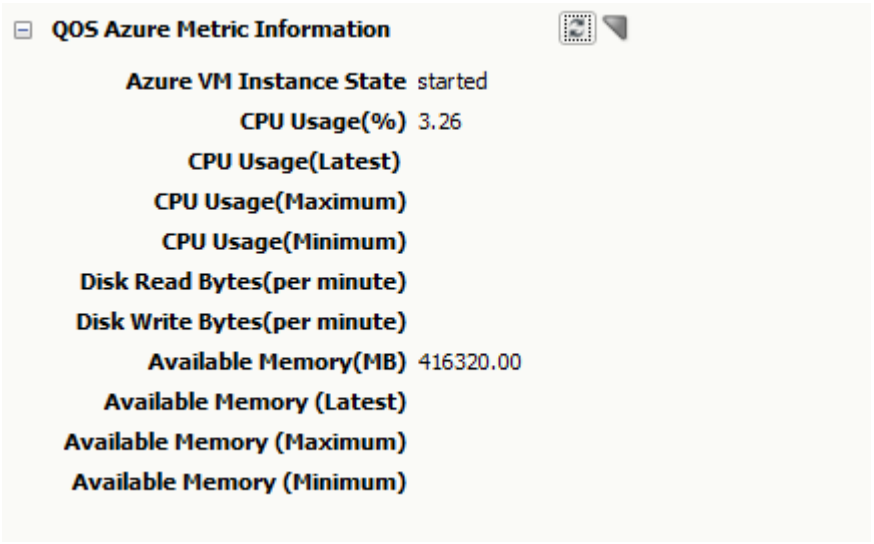
AWS related QOS Metrics

The following QoS Metric information for EC2 Instances are displayed in the **EC2 instance > Information Tab** view:



Azure related QoS metrics

The following QOS Metric information for EC2 Instances are displayed in the **Azure instance > Information Tab** view:



Launch-in-Context

The Launch-in-Context feature allows you to launch a UIM-UMP device view, from DX NetOps Spectrum for performance views of the device, in the given context. The UIM-UMP device view provides detailed information about the IP elements such as Virtual Center (VC), ESX hosts, and Virtual Machines (VM). The information (such as disk usage, cpu usage, processor queue length, paging, and memory usage) about the UIM virtual entities is displayed graphically.

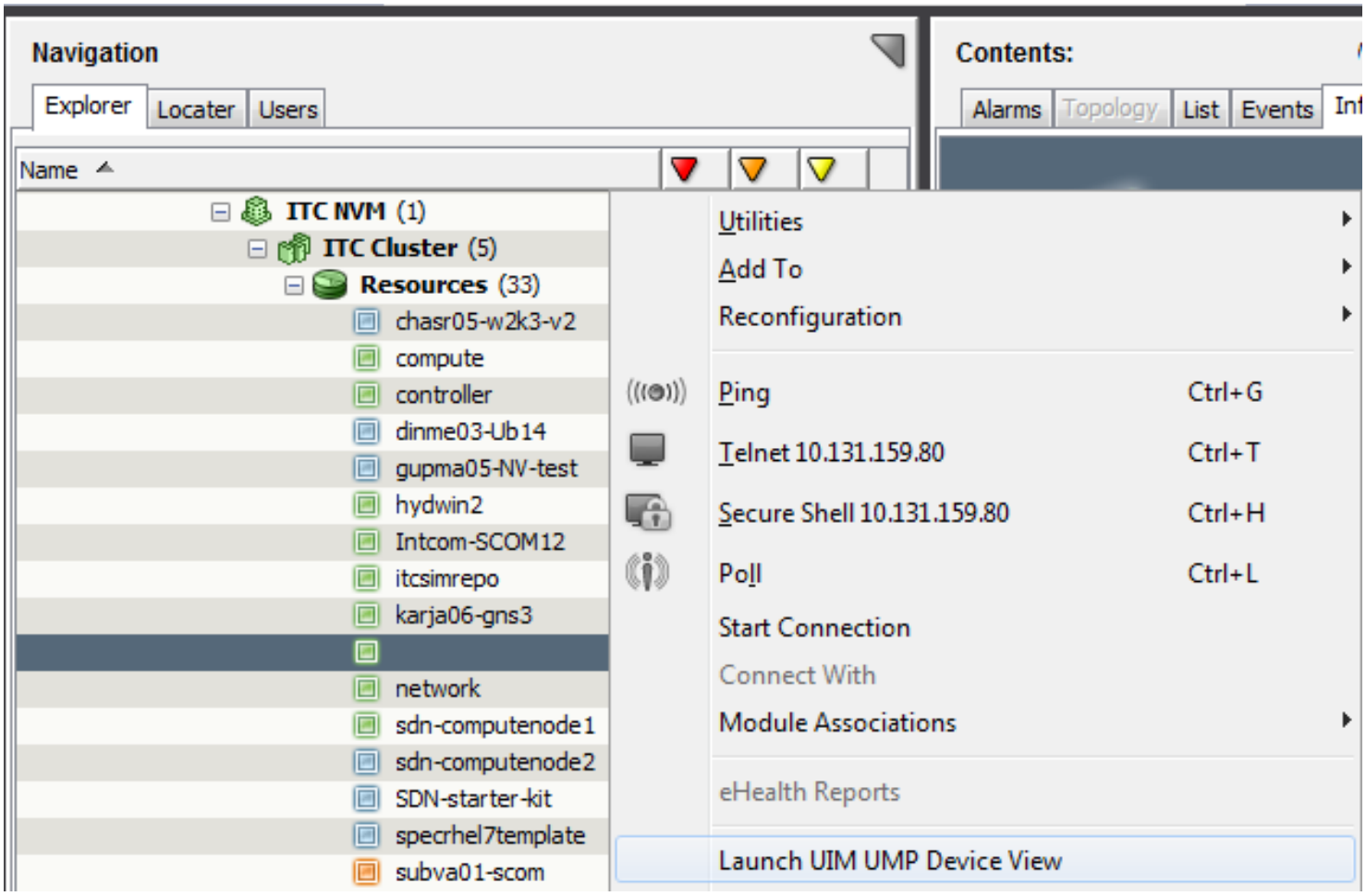
If you are launching the UMP view for the first time in a browser, a dialog for user credentials appears. The user credentials dialog does not appear if you are launching the UMP view using the same browser instance.

> **NOTE**
>
> The Launch-in-Context feature is available only for the IP elements. This feature is disabled for logical entities such as Data Centers, ResourcePools, and Clusters.

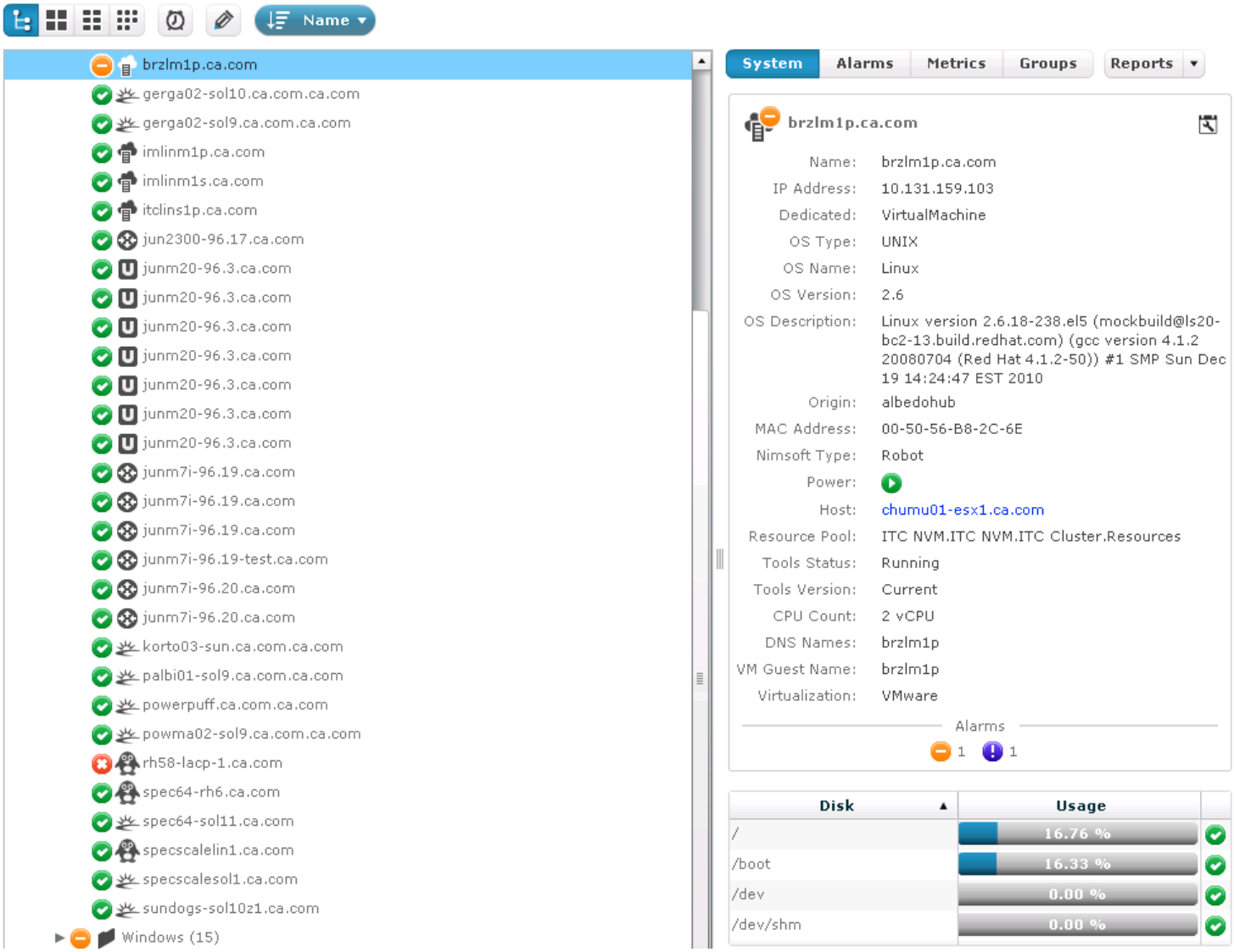
Follow these steps:

1. Open the DX NetOps Spectrum OneClick console.
2. Select **UIM Manager**, **Virtualization,** and click **VMware**.
   A list of all virtual entities such as VC, ESX, and VM is displayed.

3. Right-click one of the IP elements and select **Launch UIM UMP View**.
   The UIM UMP login page opens.
   The following image displays the IP elements and the option to launch the UMP view from the OneClick console:



4. Enter the UIM UMP credentials and click Login.
   The UMP view opens with the detailed information about the IP elements.
   The following image shows the UIM UMP view:



Condition Correlation - (UIM Integration)

If you enable the DX NetOps Spectrum and UIM Integration from the OneClick web page, the VMware inventory is modeled within the Nimsoft Manager hierarchy. The VMs that are modeled are of model type NimsoftVMPingable. Consider the following scenarios to apply Condition Correlation:

Scenario 1:

If you enable Monitors and shut down VM in the UIM with the following settings:

- Message token: EventWarning
- PowerState Monitor: VM
- HostPowerState Monitor : ESX Hosts

Events 0x10d35, 0x6330058, and 0x6330064 are generated in OneClick with Critical alarm. The alarms 0x10d35 and 0x6330058 are made symptom of 0x6330064.

Scenario 2:

Enable Monitors and shut down an ESX server with the following settings:

- Message token: EventWarning
- PowerState Monitor: VM
- HostPowerState Monitor: ESX Hosts

Events 0x10d35, 0x6330058 are generated with Critical alarm and the alarm 0x10d35 is made symptom of 0x6330058.

Scenario 3:

Enable Monitors and shut down ESX and VM with the following settings:

- Message token: EventWarning
- PowerState Monitor: VM
- HostPowerState Monitor: ESX Hosts

Events 0x6330058, 0x6330064 are generated with Critical alarm and the alarm 0x6330058 is made symptom of 0x6330064.

> **NOTE**
>
> If you place the ESX server in maintenance mode, an event 0x6330066 is generated in DX NetOps Spectrum. If the ESX server comes out of maintenance mode, an event 0x6330067 is generated in DX NetOps Spectrum and the VMs in the ESX host Server are moved out of Maintenance mode

Traps and Alarm Support

DX NetOps Spectrum and UIM integration for virtualization management supports the following alarms:

Alarms generated for the ESX Servers,

- If you enable the HostPowerState monitor and shutdown the ESX server, an alarm is generated with the event code 0x6330060.
- If you enable the HostPowerState monitor and the status of the ESX server to Unknown, an alarm is generated with the event code 0x6330061.
- If you enable the IsInMaintenanceMode monitor and the status of the ESX server to True, an alarm is generated with the event code 0x6330066.

Alarms generated for Virtual Machine (VM)s,

- If you enable the PowerState monitor and shutdown the VM, an alarm is generated with the event code 0x6330058.
- If you enable the PowerState monitor and the status of VM to Suspended, an alarm is generated with the event code 0x6330059.
- If you enable the GuestState monitor and the status of VM to notRunning, an alarm is generated with the event code 0x6330058.
- If you enable the GuestState monitor and the status of VM to Unknown, an alarm is generated with the event code 0x6330058.

Common Events for both ESX and VMs,

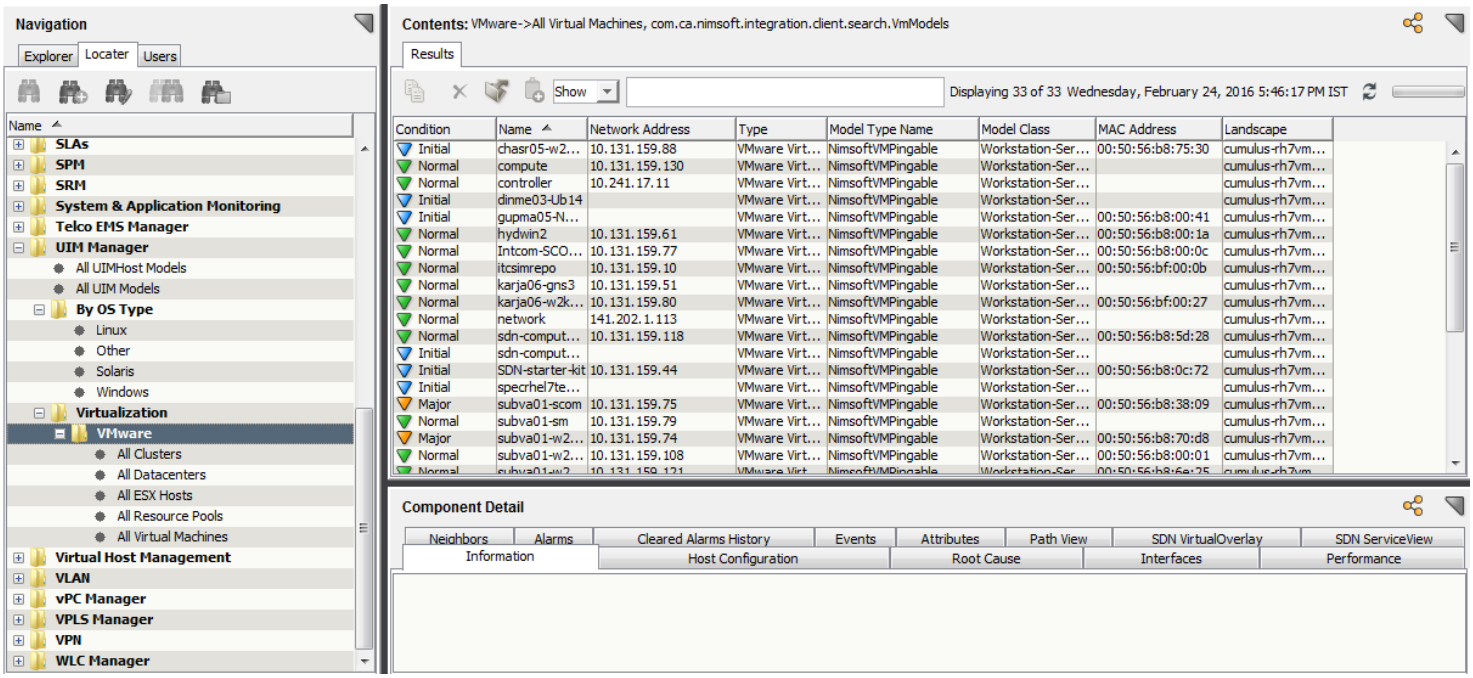
- If you enable the Event monitor and shut down both the ESX server and VM, an alarm is generated with the event code 0x6330064.
- If you enable the Event monitor and shut down both the ESX server and VM, an alarm is generated with the event code 0x6330064.
- If you enable the Status monitor and shut down both the ESX server and VM, an alarm is generated with the event code 0x6330064.

Locater Search

You can use the search functionality in the Locater tab to find both the logical entities (such as Nimsoft ResourcePool and Nimsoft Clusters) and virtual entities (such as ESX Host and Virtual Machine (VM)) that are available in the DX NetOps Spectrum environment. You can access Locater search from the Locater tab of the Navigation Panel. The search results appear in the Results tab of the Contents panel.

Follow these steps:

1. Open the DX NetOps Spectrum OneClick Console.
2. From the Navigation Panel, Click the Locater tab.
   The Search Options window opens.
3. Expand UIM Manager, VMware, Virtualization, and double-click the appropriate VMware entity to search.
   The Locater Search results are displayed in the Contents pane.
   The following figure shows the Locater Search results for all the VMware entities:



Reports (UIM Integration)

You can generate the availability report of all the ESX servers and virtual machines that are available in a DX NetOps Spectrum environment. Access InfoView from the OneClick home page to generate and manage reports. For more information, see the Spectrum Report Manager section.

The following list of events are used for the calculation of availability reports:

ESX Servers

- Up Events
  - 0x06330063
  - 0x6330065
- Down Events
  - 0x6330060
  - 0x6330061
  - 0x6330064

Virtual Machines

- Down Events
  - 0x6330058
  - 0x6330059
  - 0x6330064
- Up Events
  - 0x6330062
  - 0x6330065

Integration with UIM Through the Southbound Gateway

UIM and DX NetOps Spectrum are integrated through the DX NetOps Spectrum Southbound Gateway component (SBGW). This integration is unidirectional (UIM to DX NetOps Spectrum), and supports multiple outstanding alarms, of various types, per device.

The DX NetOps Spectrum - UIM Integration expands the DX NetOps Spectrum model of the infrastructure with information and alarms from UIM and provides the following benefits:

- Receive events and alerts in DX NetOps Spectrum from UIM probes.
- Obtain extended DX NetOps Spectrum monitoring capabilities leveraging the intelligence of UIM probes
- Use the Nimsoft SLA rules to trigger events that create alert conditions in DX NetOps Spectrum.
- Use DX NetOps Spectrum the root cause analysis capabilities to perform basic root cause analysis on events and alerts that are created by UIM.

Integrate with UIM Through the Southbound Gateway

> **WARNING**
>
> From the 10.1.2 release, the SNMP Gateway probe and SBGW /Southbound gateway are no longer recommended for alarms synchronization from UIM to DX NetOps Spectrum. Use the Spectrum Gateway (spectrumgtw) probe for this purpose.
>
> For more information about the integration of DX NetOps Spectrum with other CA products, see Integration Compatibility.
>
> We recommend that you go through the DX NetOps Spectrum-UIM Integration using spectrumgtw probe - FACT SHEET to understand the changes to the integration whether you are new to this integration or an existing user.

UIM and DX NetOps Spectrum are integrated through the DX NetOps Spectrum Southbound Gateway component (SBGW). This integration is unidirectional (UIM to DX NetOps Spectrum), and supports multiple outstanding alarms, of various types, per device. The component asserts the alarm against the existing device model or against an auto-created event model of UIM Robot. DX NetOps Spectrum EventModel is used when a full device model for the network entity does not exist in DX NetOps Spectrum. This integration supports multiple alarms types per model, such as Low Disk, Excessive CPU usage, and Traffic Threshold violation.

Overview

The DX NetOps Spectrum - UIM Integration expands the DX NetOps Spectrum model of the infrastructure with information and alarms from UIM and provides the following benefits:

- Receive events and alerts in DX NetOps Spectrum from UIM probes.
- Obtain extended DX NetOps Spectrum monitoring capabilities leveraging the intelligence of UIM probes
- Use the UIM SLA rules to trigger events that create alert conditions in DX NetOps Spectrum.
- Use DX NetOps Spectrum the root cause analysis capabilities to perform basic root cause analysis on events and alerts that are created by UIM.

As an administrator, configure UIM to send alert data to DX NetOps Spectrum. UIM sends the trap data to the hostname and port where the SpectroSERVER is running. By default, DX NetOps Spectrum uses standard SNMP trap port 162. DX NetOps Spectrum accepts an individual SNMP trap packet to a maximum size of 65467 bytes. You can modify the port by changing the snmp_trap_port parameter in the DX NetOps Spectrum ".vnmrc" file that is located in the DX NetOps Spectrum directory.

Perform the following tasks to integrate UIM and DX NetOps Spectrum through the Southbound Gateway:

Review the Prerequisites and Considerations

Verify the following prerequisites before installing and configuring the DX NetOps Spectrum - UIM Integration:

- Licensed installations of CA Spectrum 10.1 (or later) and CA UIM 8.2 (or later) are required.

  NOTE
  If you plan to install DX NetOps Spectrum as a user other than Administrator, disable User Account Control (UAC) on Windows. For more information, see the Fresh Install section.
- Verify that the system where you want to install DX NetOps Spectrum has a static IP address.
- Standard DX NetOps Spectrum supported platforms and hardware are required.

Verify the following considerations:

- The current integration does not attempt to upgrade previous (that is field-developed) integrations. We plan to support upgrades to future versions of this integration.
- This integration requires DX NetOps Spectrum to use the SNMP Trap port (162) for communication from UIM.
- This integration connects to only a single UIM instance.
- This integration depends on trap reception because typical SNMPv1 traps are unconfirmed. Traps can be dropped in transit and not recognized.
- For the events and alarms to be raised on the correct DX NetOps Spectrum model, use an IP address instead of hostname to model the entity on UIM. If a hostname is used for entities that are modeled in UIM, DX NetOps Spectrum alarms are raised on the EventModel of the robot hosting the probe.

Install and Configure DX NetOps Spectrum

DX NetOps Spectrum installation software requires administrator privileges to evaluate available resources and run custom installation scripts. An initial installation generates residual files with administrator ownership. Subsequent upgrade installations also require administrator privileges.

WARNING
The C:\Program Files\CA directory on Windows platforms and the /opt/CA directory on Linux platforms are automatically created during the DX NetOps Spectrum first-time installation. DX NetOps Spectrum components that are also common to other CA products are intentionally installed into this directory. This directory is automatically updated as needed during a DX NetOps Spectrum upgrade. Do not remove files from this directory.

A DX NetOps Spectrum installation is required to integrate UIM and DX NetOps Spectrum through the Southbound Gateway. You can install DX NetOps Spectrum on Windows and Linux platforms.

Follow these steps:

1. Stop all non-DX NetOps Spectrum running applications.

2. Perform the following actions:
   - Log off from OneClick in the Client Details web page and shut down the OneClick client.

     NOTE
     For more information, see the OneClick Administration section.
   - Click Stop SpectroSERVER to stop the SpectroSERVER and the Archive Manager in the DX NetOps Spectrum Control Panel and then close the DX NetOps Spectrum Control Panel.

     NOTE
     For more information, see the OneClick Administration section.
   - Stop all VnmSh connections.

     NOTE
     For more information, see the Command Line Interface section.
   - Close all Bash shells.

     WARNING
     Disable your antivirus software real-time protection before installing DX NetOps Spectrum. Disabling helps avoid potential problems with files that can be in use by the real-time protection software.
3. Log in as a user with administrator rights.
4. Insert the installation medium into the appropriate drive. If auto-run is disabled, you can double-click the setupnt.exe file from the Explorer view to start the installation.
   The installation starts.
5. Install DX NetOps Spectrum. For more information, see the Fresh Install section.

Deploy and Configure Probes

UIM Probes are small, dedicated applications that monitor specific resources or events. Each probe can be easily configured for your specific monitoring requirements.

The SNMP Gateway probe sends traps from UIM to DX NetOps Spectrum. To integrate UIM with DX NetOps Spectrum, configure the SNMP Gateway probe (snmpgtw) through CA Unified Infrastructure Manager.

The SNMP gateway converts alarms to SNMP trap messages that are readable by any SNMP-based event manager. The SNMP gateway maps the various severity levels to enterprise-specific trap types. For more information, see the Unified Infrastructure Management documentation.

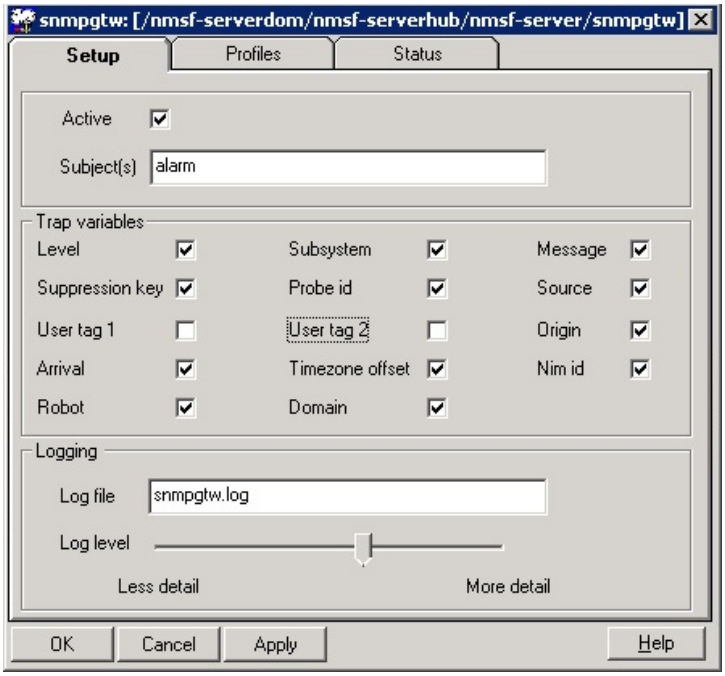
Follow these steps:

1. Open Unified Infrastructure Manager.
2. From the Console window, select Archive, UIM Server hub, and Robot.
   A list of predefined probes is displayed.
3. Select a package name in the archive folder.
4. Drag and drop the package name to the domain/hub/robot.
   A View Distribution Progress dialog opens.
5. Click Close Dialog after distribution has completed.
   The probe is deployed to the specified location.
6. To configure the probe, double-click the probe that you deployed.
   The Probe Configuration window opens.
7. Click the Setup tab.
   The Setup window opens with the following options:
   - **Active**
     Activates or deactivates this probe.
   - **Subject(s)**

Specifies the UIM subject that is transformed. Subject is a text string, that classifies the UIM message for all components of UIM.
Default: Alarm

- **Trap variables**
  Indicates a unique identifier of the SNMP operation where the traps are triggered.
- **Log file**
  Specifies the file where the probe logs information about its internal activity.
- **Log level**
  Sets the level of details for the data that is written to the log-file. We recommend logging as little data as possible during normal operation to minimize disk consumption. You can then increase the amount of detail when debugging.

The following image illustrates the options that are available in the Setup window:



8. Click the Profiles tab.
   The Profile window opens. For more information, see Configure CA Unified Infrastructure Manager.
9. Click Ok.
   The snmpgtw probe is deployed and configured.

Configure CA Unified Infrastructure Manager

The CA Unified Infrastructure Manager is the primary interface for the configuration and management of the UIM system.

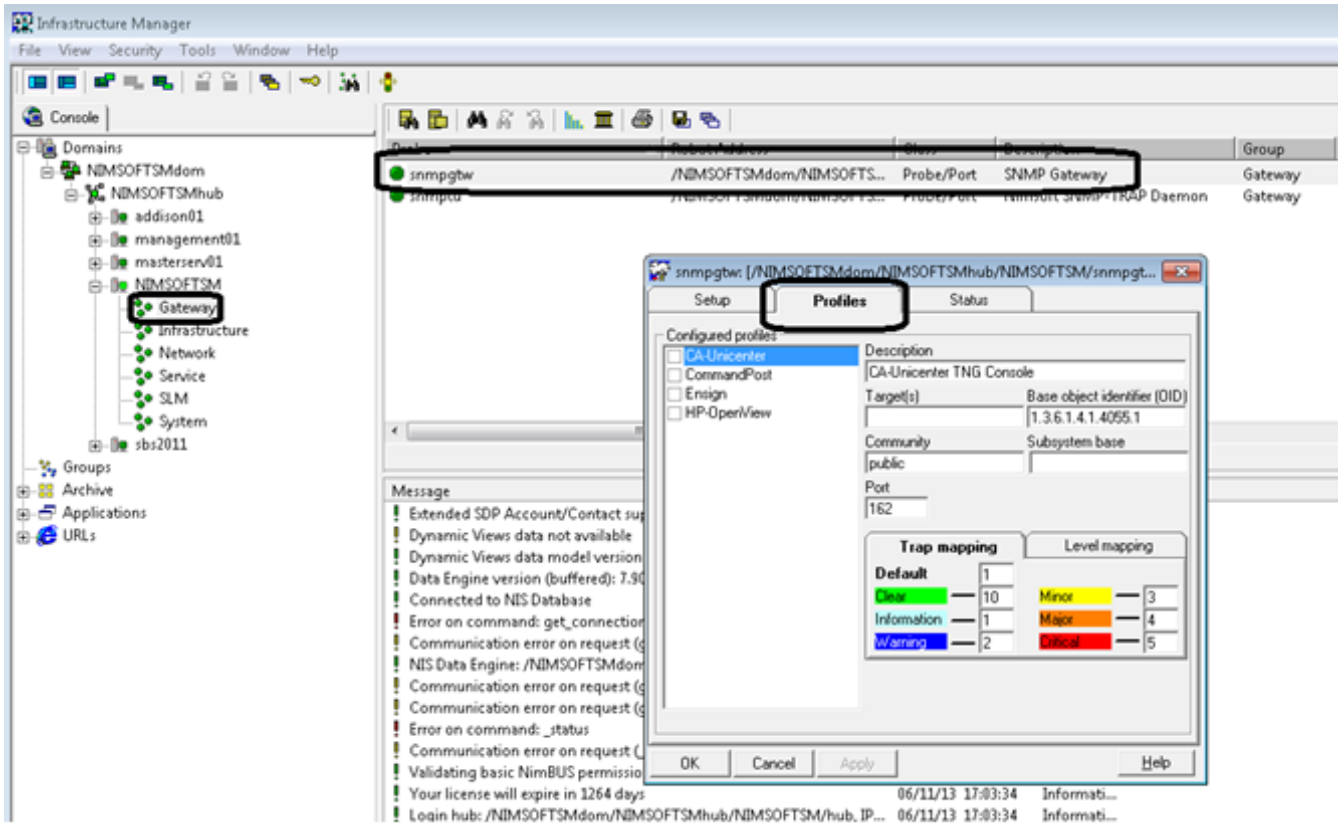
Configure UIM to manage entities on your network through CA Unified Infrastructure Manager or the Unified Management Portal. To integrate UIM with DX NetOps Spectrum, configure the SNMP Gateway probe (snmpgtw) through CA Unified Infrastructure Manager. For more information, see Deploy and Configure Probes.

A profile is created in the SNMP Gateway Probe to communicate to the UIM about the traps to send, the conditions under which to send them, and where to send them.

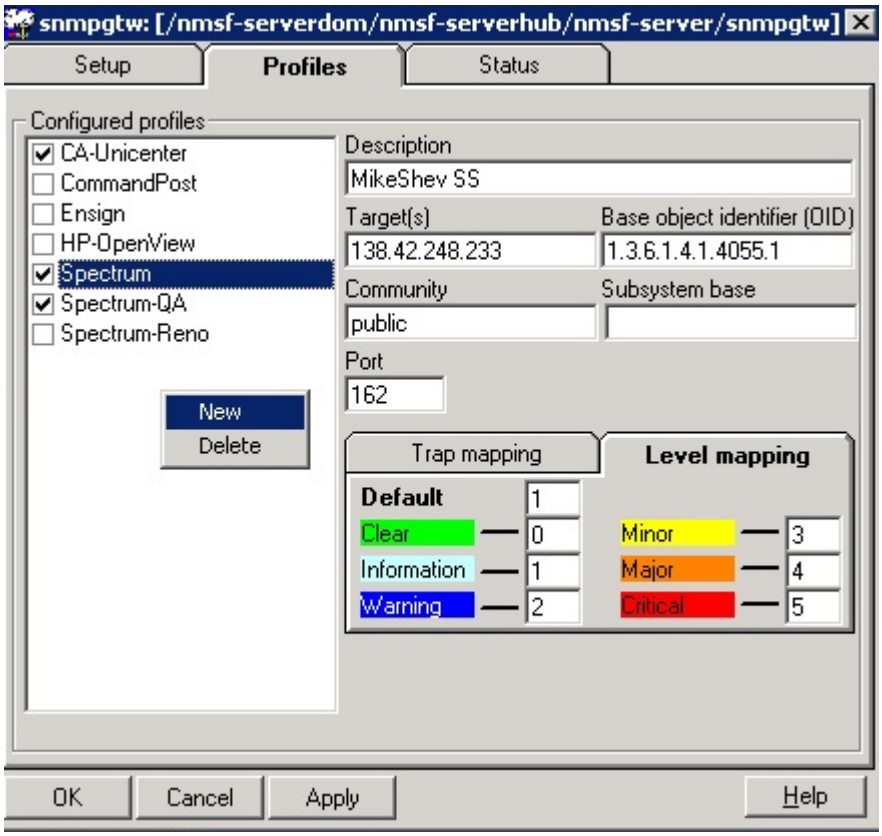
Follow these steps:

1. Open Unified Infrastructure Manager.
2. From the Console window, select Domains, UIM Server Domain, UIM Server Hub, UIM Primary Hub and then Gateway.

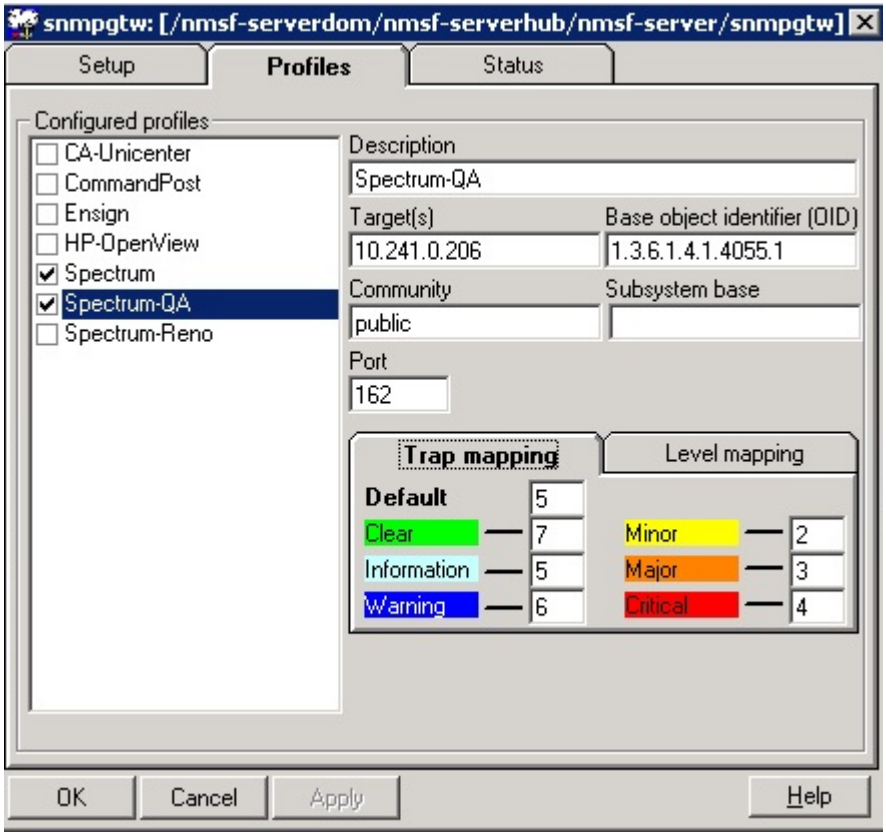
A list of Probes is displayed.
The following image displays the navigation to snmpgtw probe:



3. Double-click the snmpgtw probe.
   The Probe Configuration window opens.
4. Click the Profiles tab.
5. Right-click the Configured Profiles workspace and select New.
   The following image illustrates the procedure to create a new profile:



6. Enter the name of the profile. For example, you can supply Spectrum-*Server name*.
7. To enable the profile, click DX NetOps Spectrum in the list of Configured profiles.
   The following image illustrates the options that are available in the Profiles window.



- **Target(s)**
  Specifies the SpectroSERVER IP address. Indicates the network node where the SNMP traps can be sent.
- **Base Object Identifier (OID)**
  Indicates the SNMP Object identifier to be used in the trap packages generated.
  Default: 1.3.6.1.4.1.4055.1
- **Community String**
  Indicates the SNMP community string that is used in the SNMP traps.
- **Trap Mapping**
  Classifies the incoming traps by trap type and takes different actions for different trap types. You can map the severity levels of the alerts to SNMP traps.
  For example, provide the following values for trap mapping:
  Default: 5
  - Clear: 7
  - Informational: 5
  - Warning: 6
  - Minor: 2
  - Major: 3
  - Critical: 4

  NOTE
  If you want to disable informational and warning messages at the source level, remove the mappings for Default, Warning, and Informational in Trap Mapping.
- **Level Mapping**
  Identifies the severity levels with different codes. You can map the UIM severity levels to the corresponding level in the receiving system by specifying the correct code.
  For example, provide the following values for level mapping:

Default: 1

- Clear: 0
- Informational: 1
- Warning: 2
- Minor: 3
- Major: 4
- Critical: 5

8. Click Apply and Ok.
   Unified Infrastructure Manager is configured to integrate with DX NetOps Spectrum.

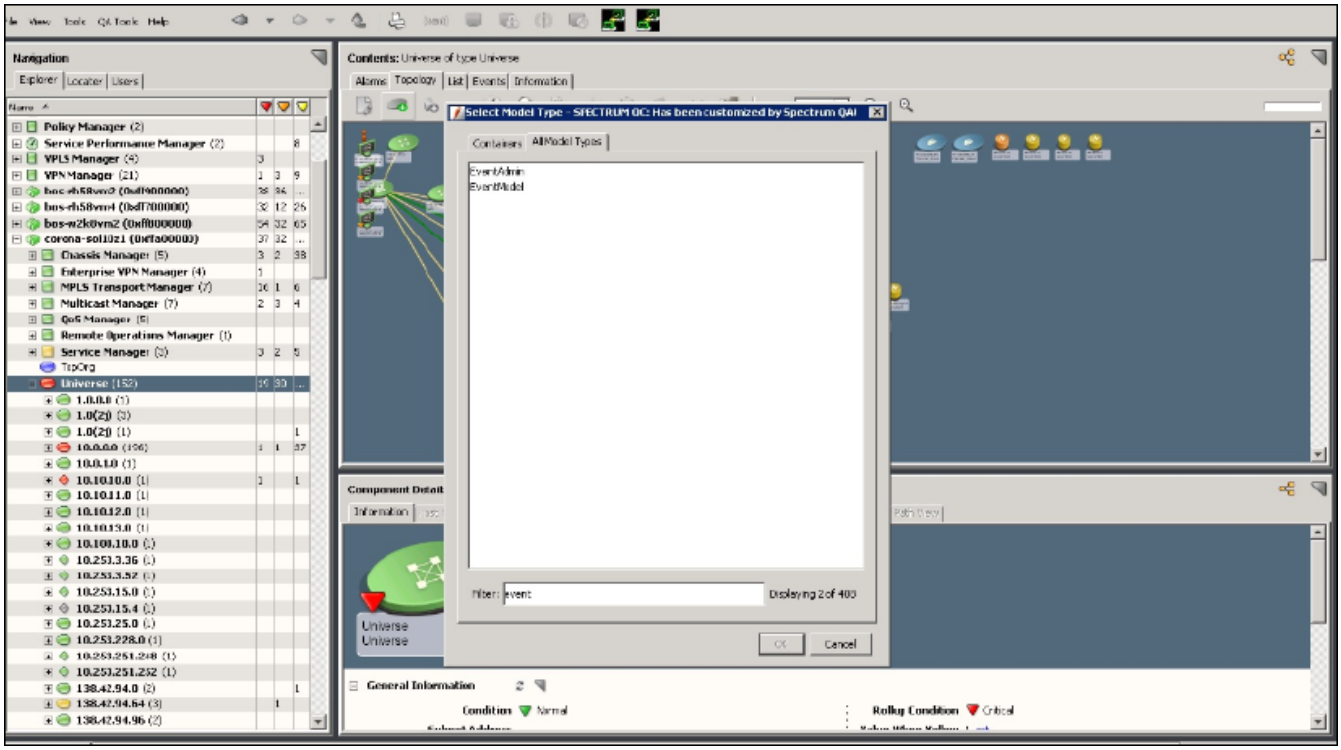
Create an EventAdmin Model for the UIM Server

The DX NetOps Spectrum EventAdmin model receives events from the Southbound Gateway and transfers the event data to EventModels or device models depending on how the integration is configured. Alarms can be created from this event data.

The EventModel is a model type that represents a unique source of event data on the system that is managed by the EventAdmin application. A given EventAdmin model can contain one or many instantiated EventModels. Each event that is received through the Southbound Gateway contains information that uniquely identifies the source of that event. The EventAdmin model receives the event, finds the unique event source, and passes the event to the target destination. Create an EventAdmin model for the UIM server to support the integration.

Follow these steps:

1. Open the DX NetOps Spectrum OneClick Console.
2. From the Navigation Panel, select SpectroSERVER, and then Universe.
3. Click the Topology tab in the Contents panel and click Create New Model by Type.
   The Select Model Type dialog opens.
   The following image displays the model types to be created:



4. In the All Model types tab, click EventAdmin.

5. Click OK.
   The Create Model of Type EventAdmin dialog opens.
6. Configure the following parameters:
   - **Name**
     (Optional) Defines the EventAdmin model name. This model name appears in the field at the top of the EventAdmin icon.
   - **Network Address**
     Specifies the network address of the event source host computer. Required for all integrations that are based on the SNMP traps.
   - **Security String**
     (Optional) Defines who can view and edit this model.
   - **Manager Name**
     When this attribute is set on the EventAdmin model, all EventModels contained within this EventAdmin also have this attribute.
   - **EventModel Prefix**
     Verifies the naming prefix for all EventModels that are associated with a particular EventAdmin model. This field is related to the EventModel Name for all the EventModels contained by this EventAdmin. It is also useful for sorting and filtering.
     Default: 0x06330000
7. Click OK.
   The EventAdmin model is generated. A default EventModel is also created and is contained in the EventAdmin model. This model is used for fault tolerance functionality that represents the unique source.

Verify the Received Events and Alarms in OneClick

The EventAdmin Model receives an event from UIM and sends it to the EventModel in OneClick. The event generates an alarm on this model. To verify that the integration is configured correctly, we recommend viewing the details of the alarm data from the Alarm Details tab in OneClick. The generic and subsystem-specific events are created in OneClick. You can also verify the design pattern of these events/alarms.

Follow these steps:

1. Open the OneClick Console.
2. Select the EventModel in the Navigation panel.
3. To view events, click the Events tab in the Contents panel.
   Events are displayed with the following event types:
   - **Generic Events**
     Indicates the events that are not related to CPU, Disk, and Memory subsystems.
     The range starts from 0x06330000 - 0x6330005.
   - **Subsystem Specific Events**
     Indicates the events that are related to CPU, Disk, and Memory subsystems. You can verify the following event range for the subsystem-specific events:
     - CPU
       0x06330050 - 0x6330055
     - Disk
       0x06330030 - 0x6330035
     - Memory
       0x06330040 - 0x6330045
4. Verify the following design pattern of these events/alarms:

- 0x063300x0 Clear Event
- 0x063300x1 Minor Event / Alarm
- 0x063300x2 Major Event / Alarm
- 0x063300x3 Critical Event / Alarm
- 0x063300x4 Informational Event

NOTE
You can review the following table to know how the UIM message severities are mapped to DX NetOps Spectrum events and alarms:

5. UIM DX NetOps Spectrum
   - Informational Event only
   - Warning Event only
   - Minor Minor Alarm
   - Major Major Alarm
   - Critical Critical Alarm
6. To view alarms, click the Alarms tab.
   Alarms are displayed.
7. Click the Alarm Details tab in the Component Detail panel to view the alarm details.
   Events and Alarms that are generated in OneClick are verified.

NOTE
Alarms that are manually cleared in the UIM Alarm Console do not clear the corresponding alarms in DX NetOps Spectrum. This behavior is caused by a known limitation of the SNMP Gateway probe (snmpgtw). Therefore, when you clear alarms in UIM, the alarms accumulate in DX NetOps Spectrum, causing high alarm counts. These alarms must be manually cleared in DX NetOps Spectrum.

Performance Considerations

UIM - DX NetOps Spectrum integration through the Southbound Gateway supports and implements all severities and traps (such as Informational, Warning, Minor, Major, Critical, Clear).

NOTE
By default, UIM snmpgtw is configured to send alerts (traps) for messages of all severity levels.

The volume of events and alarms that are generated by UIM in DX NetOps Spectrum depends on the number, type, and condition of managed elements. In situations where performance is an issue, you can disable these messages at the Unified Infrastructure Manager.

For example, if the trap storm detection threshold of DX NetOps Spectrum exceeds a certain level, it indicates that performance is degraded. By default, this threshold is configured for 20 traps/second from a single device. In a moderately large UIM installation, the DX NetOps Spectrum default trap storm threshold can be exceeded easily, and when it is exceeded, traps are dropped. To preserve the most critical traps, we recommend disabling the informational and warning messages. In this way, bandwidth is not used in less severe situations and the critical traps can be handled by DX NetOps Spectrum.

To handle this situation, you can disable the informational messages that are sent by UIM. In this way the problem can be resolved at the source level. If the trap storm threshold is exceeded, the warning messages can be disabled and not sent to DX NetOps Spectrum. You can also raise the trap storm threshold to 25 or 30 traps/second, if the SpectroSERVER has sufficient capacity.

If after disabling the informational and warning messages, the number of alerts from UIM still exceeds the trap storm threshold, consult UIM documentation to determine ways to limit the number or types of traps being sent to DX NetOps Spectrum. By default, all alarms are filtered. Therefore, you can change the alarm messages that are filtered by snmpgtw. You can also change the alarm setting to alarm_new and alarm_clear messages, which can reduce the total traffic from UIM to DX NetOps Spectrum.

NOTE
If you change the alarm setting to alarm_new and alarm_clear message, the alarm counts may not be correctly incremented in DX NetOps Spectrum as a single message for each occurrence of an alarm that is received.

Disable the Integration

You can disable the UIM - DX NetOps Spectrum Integration, if you want to stop generating alarms and events in OneClick. On disabling the integration, the EventAdmin model no longer receives events from UIM and the events are not forwarded to the EventModel model in OneClick.

Follow these steps:

1. Open Unified Infrastructure Manager.
2. From the Console page, select Gateway.
   The SNMP Gateway window opens.
3. Click the Profiles tab.
   The Configured Profiles window opens.
4. Right-click a Profile, select Delete.
   Profile is deleted.
5. Click Ok.
   Integration is disabled.

Supporting AWS (Amazon Web Services) Cloud Monitoring

Overview

From 10.2.1, AWS Cloud (Amazon Web Services) Monitoring is supported by DX NetOps Spectrum using the UIM and DX NetOps Spectrum integration. The AWS Monitoring probe deployed in UIM enables the metric data collection from the AWS instances. This data is then synchronized from UIM to DX NetOps Spectrum. Currently, we are synchronizing the AWS resources drilling down to the following hierarchy: **VPC** domain > **Subnet** domain **> EC2 instances**.

All AWS Cloud entities are modeled as UIM Host Servers (model type UIMHostServer).

A *virtual private cloud* (VPC) is a virtual network dedicated to your AWS account. It is logically isolated from other virtual networks in the AWS cloud. You can launch your AWS resources, such as Amazon EC2 instances, into your VPC. You can configure your VPC; you can select its IP address range, create subnets, and configure route tables, network gateways, and security settings.

A *subnet* is a range of IP addresses in your VPC. You can launch AWS resources into a subnet that you select. Use a public subnet for resources that must be connected to the Internet, and a private subnet for resources that won't be connected to the Internet.

Amazon VPC is the networking layer for Amazon EC2. Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the Amazon Web Services (AWS) cloud. Amazon EC2 is hosted in multiple locations worldwide. These locations are composed of regions and Availability Zones. Each region is a separate geographic area. Each region has multiple, isolated locations known as Availability Zones.

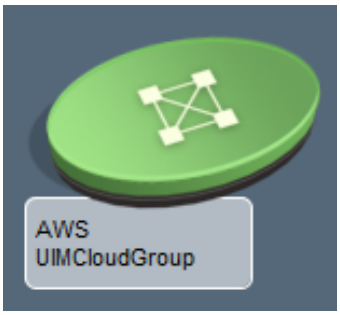
IP addresses enable resources in your VPC to communicate with each other, and with resources over the Internet. Amazon EC2 and Amazon VPC support the IPv4 and IPv6 addressing protocols. By default, Amazon EC2 and Amazon VPC use the IPv4 addressing protocol. When you create a VPC, you must assign it an IPv4 CIDR block (a range of private IPv4 addresses). Private IPv4 addresses are not reachable over the Internet. To connect to your instance over the Internet, or to enable communication between your instances and other AWS services that have public endpoints, you can assign a globally-unique public IPv4 address to your instance.

Each AWS entity will have both Public and Private IP's, provided by the corresponding attributes: **PrimaryIPV4Address** and **OtherIPAddresses.** Some times public and private IPs are not unique and will be the same, thus the VM will be

unreachable by SpectroSERVER. In such cases DX NetOps Spectrum uses the **AWSInstanceState** attribute/ **Instance State** field, to find out the state of the VM and to show the condition in the SpectroSERVER. DX NetOps Spectrum also stops polling that particular device, till public and private IP's are updated.

Topology View

The following icons represent the AWS entities after they are synced from UIM to DX NetOps Spectrum:

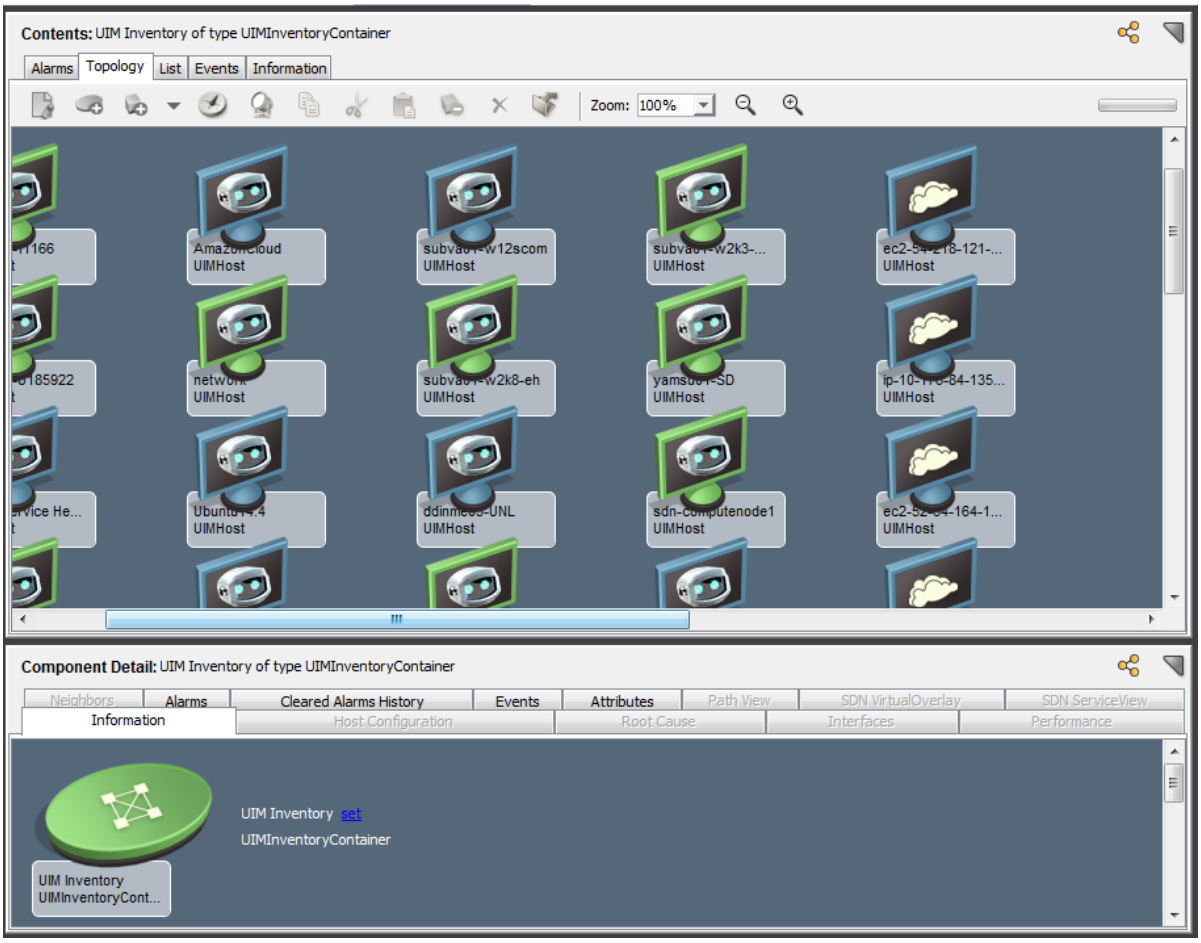


represents the container or network group icon for AWS Cloud entities, VPCs and subnets.

represents the AWS EC2 instances.

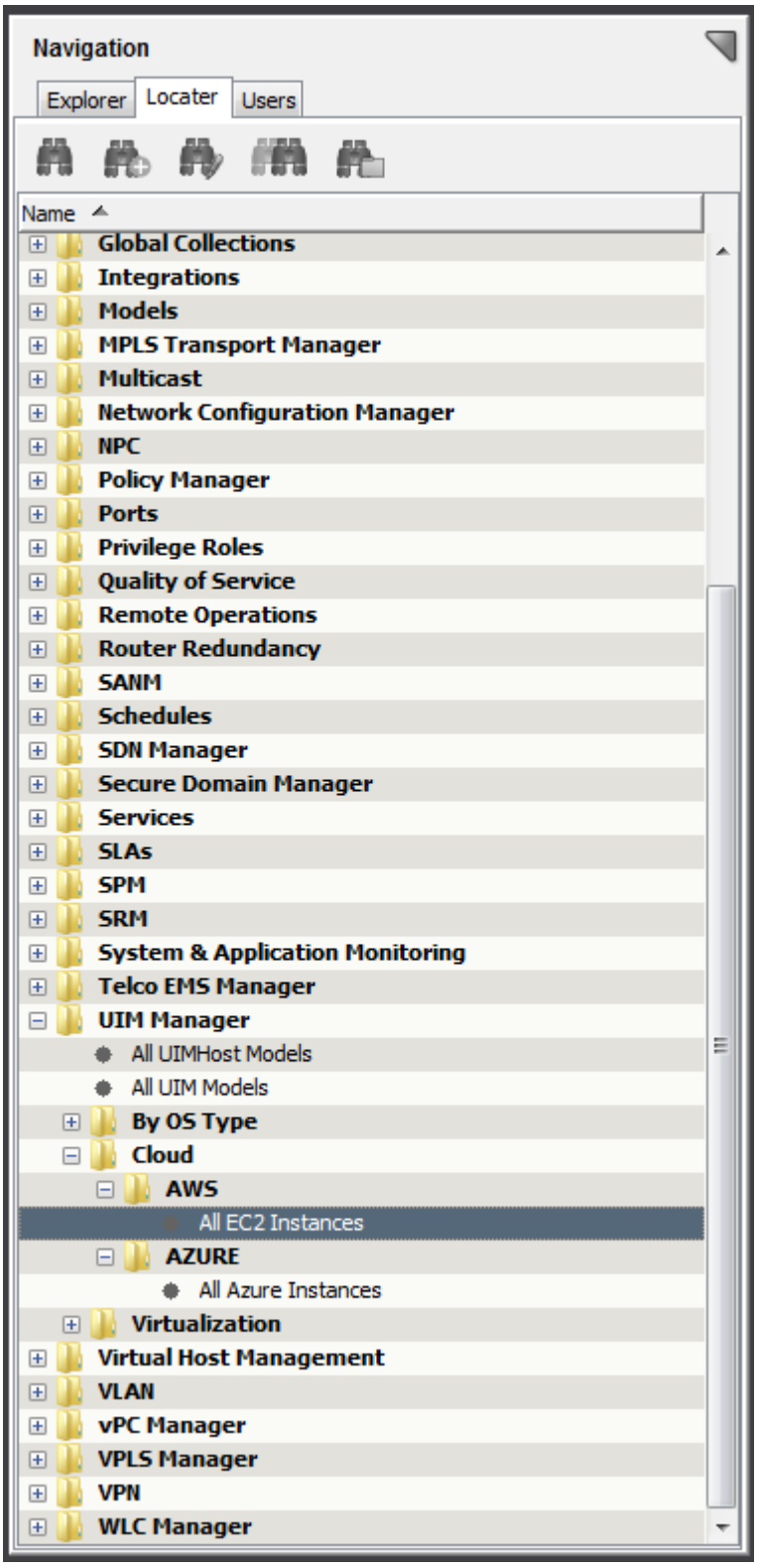
To view the EC2 instance models in its relevant context, follow these steps:

In the DX NetOps Spectrum OneClick Console, **Explorer View**, navigate to the **Universe > UIM Inventory** container, and select the **Topology** tab.
The Topology view displays all the inventory from UIM (which you have configured) that is synced to DX NetOps Spectrum:



Locater Search

You can use pre-configured searches to locate all AWS EC2 instances, in the DX NetOps Spectrum database quickly. The searches are grouped under the **UIM Manager > Cloud > AWS > All EC2 instances** folder in the **Locater** tab of the **Navigation** panel, as shown below:



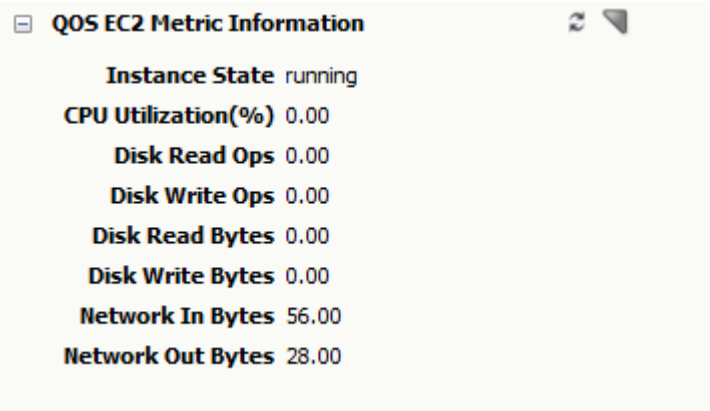
Follow these steps, to view All EC2 instances associated to AWS entities:

1. Navigate to **Locater** tab, **UIM Manager > Cloud > AWS**, and select **All EC2 instances**.

2. Select the landscapes you wish to search against, in the **Select Landscapes to Search** dialog box.
3. Click **OK.**
   The results matching your query is displayed in the **Contents** pane.

AWS related QOS Metrics

The following QOS Metric information for EC2 Instances are displayed in the **EC2 instance > Information Tab** view:



Metric Name	Description	Units
Instance State	This metric is the operational status of the EC2 instance. You can set up the threshold for status using a numeric value between 0 and 2. Each number is assigned a status value, as follows: 0: Instance is executing 1: User has stopped the instance 2: Instance has crashed	State
CPU Utilization (%)	This metric is the percentage of allocated EC2 compute units that are currently in use on the instance. You can use the information to identify the processing power required to execute an application on the selected instance.	Percent
Disk Read Ops	This metric is the number of completed read operations from all ephemeral disks available to the instance. You can use the information to identify the rate at which an application reads from a disk.	Count
Disk Write Ops	This metric is the number of completed write operations to all ephemeral disks available to the instance. You can use the information to identify the rate at which an application writes to a disk.	Count
Disk Read Bytes	This metric is the number of bytes read from all ephemeral disks available to the instance. You can use the information to determine the volume of data that the application reads from the hard disk of the instance.	Bytes
Disk Write Bytes	This metric is the number of bytes written to all ephemeral disks available to the instance. You can use the information to determine the volume of data that the application writes to the hard disk of the instance.	Bytes
Network In Bytes	This metric is the number of bytes received on all network interfaces by the instance. You can use the information to identify the volume of incoming network traffic to an application on the instance.	Bytes
Network Out Bytes	This metric is the number of bytes sent on all network interfaces by the instance. You can use the information to identify the volume of outgoing network traffic from an application on the instance.	Bytes

Supporting azure (Microsoft Azure Monitoring)

Overview

From 10.2.1, Microsoft Azure Monitoring is supported by DX NetOps Spectrum using the UIM and DX NetOps Spectrum integration. The azure (Microsoft Azure Monitoring) probe deployed in UIM enables the metric data collection from the AWS instances. This data is then synchronized from UIM to DX NetOps Spectrum.
Currently, we are synchronizing the Azure resources drilling down to the following hierarchy: **Subscription ID** > **Resource Group > Virtual Network > Virtual machines /Azure Instances**.

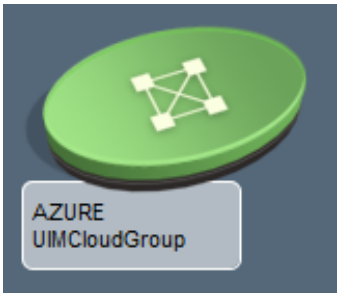
Microsoft Azure is a cloud computing platform and infrastructure, created by Microsoft that provides on-demand computing and storage to host, scale and manage Web applications and services through a global network of Microsoft-managed datacenters.

The Microsoft Azure Monitoring probe remotely monitors the health and performance of Azure infrastructure and services. The probe enables you to connect to Microsoft Azure using certificates and discover Azure resources to be monitored. The probe fetches all the service data from different geographical locations and lets you create profiles that monitor your cloud services including virtual machines (VMs), websites and storage. The probe lets you configure various monitoring parameters for each of these services. For example, you can check the health status of data services and VMs, a number of requests made to the storage service, CPU utilization, and so on. Based on the configured parameters, the probe generates Quality of Service (QoS) metrics.

In DX NetOps Spectrum, the parent/ domain hierarchy (Azure, Subscription ID and Resource Group) entities are modeled as separate containers (model type **UIMCloudGroup**), and all AWS Cloud entities (that is virtual machines) are modeled as UIM Host Servers (model type **UIMHostServer**).

Topology View

The following icons represent the Azure entities after they are synced from UIM to DX NetOps Spectrum:

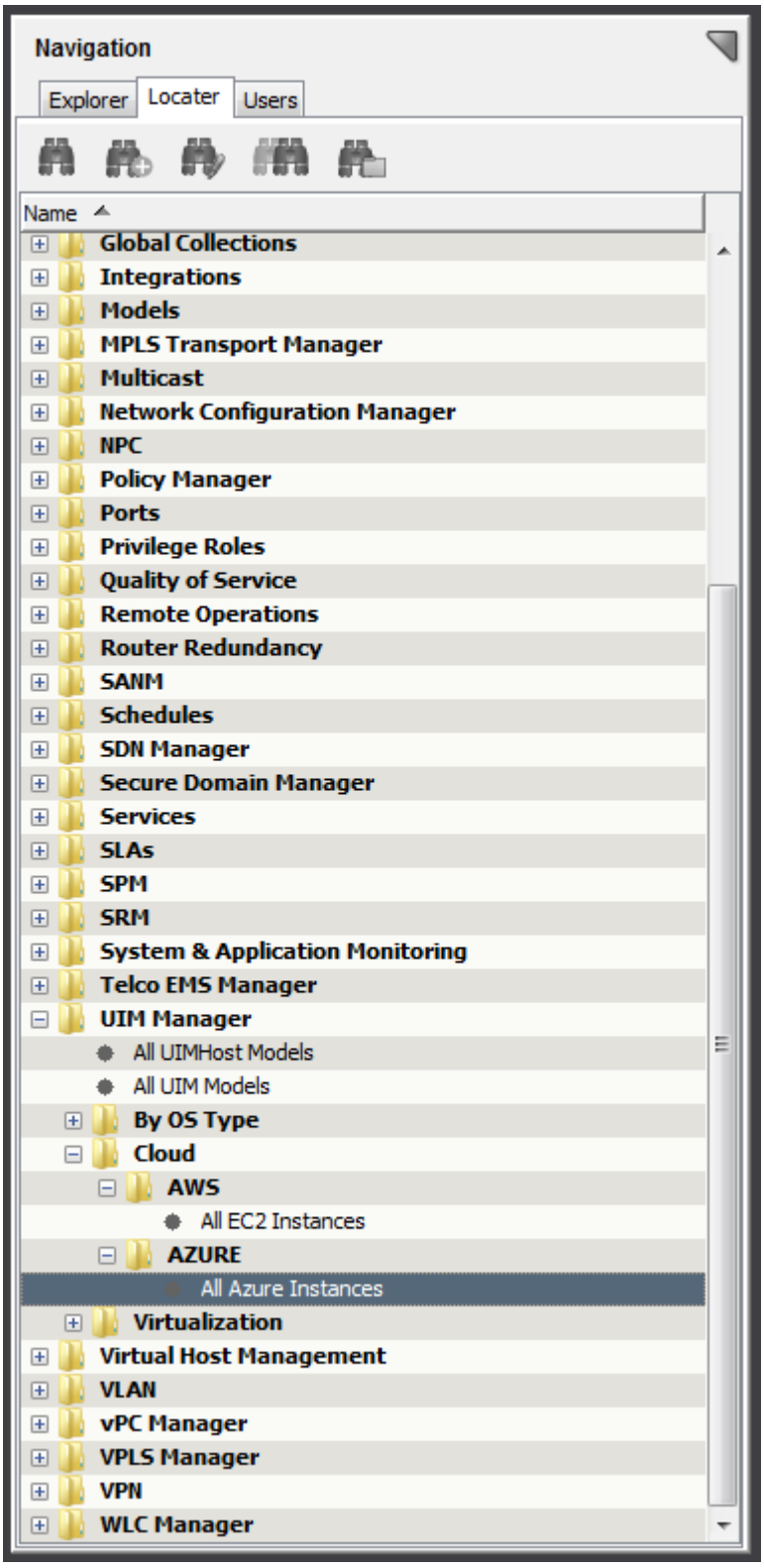


represents the container or network group icon for Microsoft Azure Cloud entities, Subscriptions, and Resource Groups.

represents the Azure VM entity/instance.

Locater Search

You can use pre-configured searches to locate all AWS EC2 instancaes, in the DX NetOps Spectrum database quickly. The searches are grouped under the **UIM Manager > Cloud > Azure> All Azure instances** folder in the **Locater** tab of the **Navigation** panel, as shown below:



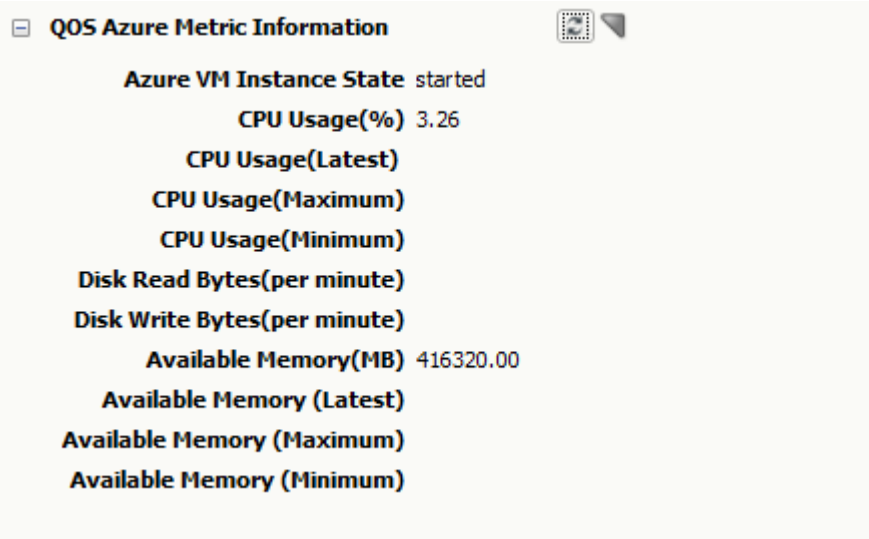
Follow these steps, to view All Azure instances associated to Microsoft Azure entities:

1. Navigate to **Locater** tab, **UIM Manager > Cloud > Azure**, and select **All Azure Instances**.

2. Select the landscapes you wish to search against, in the **Select Landscapes to Search** dialog box.
3. Click **OK.**
   The results matching your query is displayed in the **Contents** pane.

Azure related QOS metrics

The following QOS Metric information for EC2 Instances are displayed in the **Azure instance > Information Tab** view:



QOS Metric	Description	Unit
Azure VM Instance State	This metric is the current state of the virtual machine. The values indicating the state are: 0-Started, 1-Starting, 2-Stopping, 3-Stopped, 4-Unknown	State
CPU Usage (%)	This metric is the average percentage of elapsed time that the processor spends to execute non-idle threads for the virtual machine during the time period specified as the data collection interval.	Percent
CPU Usage (Latest)	This metric is the current percentage of elapsed time that the processor spends to execute non-idle threads for the virtual machine.	Percent
CPU Usage (Maximum)	This metric is the maximum percentage of elapsed time that the processor spends to execute non-idle threads for the virtual machine during the time period specified as the data collection interval.	Percent
CPU Usage (Minimum)	This metric is the minimum percentage of elapsed time that the processor spends to execute non-idle threads for the virtual machine during the time period specified as the data collection interval.	Percent
Disk Read Bytes (per minute)	This metric is the average disk read bytes per minute for the disk partition on the virtual machine during the time period specified as the data collection interval.	Bytes/ Min
Disk Write Bytes (per minute)	This metric is the average disk write bytes per minute for the disk partition on the virtual machine during the time period specified as the data collection interval.	Bytes/ Min
Available Memory (MB)	This metric is the average available memory in megabytes for the virtual machine during the time period specified as the data collection interval.	Megabytes
Available Memory (Latest)	This metric is the current available memory in megabytes for the virtual machine	Megabytes

Available Memory (Maximum)	This metric is the maximum available memory in megabytes for the virtual machine during the time period specified as the data collection interval.	Megabytes
Available Memory (Minimum)	This metric is the minimum available memory in megabytes for the virtual machine during the time period specified as the data collection interval.	Megabytes

Debugging

Debugging in DX NetOps Spectrum lets you track the data flow from UIM to DX NetOps Spectrum. It investigates and resolves integration related issues. The Start Client Debug Console contains various debug modules. Turn on UIM Integration Information to track alerts and CIs that flow from UIM to DX NetOps Spectrum.

To use Start Client Debug Console, you must first have a running OneClick client. This debug tool lets you turn on debugging output that can be seen in the Java Web Start log.

Follow these steps:

1. Open the OneClick Administration page.
   The OneClick Administration page opens.
2. Click the Administration tab.
   Links to various OneClick web server configuration pages are displayed.
3. Click the Debugging tab.
   A panel with various links to view debugging output opens.
4. Click Start Client Debug Console.
   A list of debug modules is displayed.
   The following image displays the list of available debug modules:

5. To enable debug, select On for the Nimsoft Integration Information debug module.
6. Select Max as Desired Level and click Apply.
   Debug is enabled.
7. To disable debug, select OFF and click Apply.
   Debug is disabled.

Troubleshooting Integration with UIM

DX NetOps Spectrum Not Receiving Alerts from UIM

Symptom: The alarm sync functionality of spectrumgtw is not working; previously, it was working without any issue. No changes to the spectrumgtw configuration were made.

Cause:
Check whether the state of the trellis probe is in a failed state. The trellis probe is used in the integration with the spectrumgtw probe. If it is in a failed state, the spectrumgtw service will not work.

Solution: To resolve the issues, 'cold start' (Deactivate-Activate) the trellis probe.

Issues with the spectrumgtw Configuration While Integrating CA Spectrum 10.2 with CA UIM 8.47 and 8.51

Symptom:
An error message is displayed when saving the spectrumgtw probe configuration through the Admin Console.

Cause:
The DX NetOps Spectrum OneClick web server machine is not responding to heartbeat restful post. http://<OneClick_hostname>:<port>/spectrum/restful/heartbeat

> **NOTE**
>
> This issue has been resolved in the CA Spectrum 10.2.1 (and above) integration with CA UIM 8.5.1.

Solution:
There are missing entries in the $SPECROOT/tomcat/webapps/spectrum/WEB-INF/web.xml file on the DX NetOps Spectrum OneClick web server machine.

```
com.ca.spectrum.restful.servlet.OneClickServlet
com.ca.spectrum.restful.servlet.HeartbeatServlet
com.ca.spectrum.restful.servlet.EventAlarmServlet
com.ca.spectrum.restful.servlet.HeartbeatAlarmResponseServlet
com.ca.spectrum.restful.servlet.UniversemhServlet
```

Alarm Forwarding Not Working for UIM

Symptom:

On a Distributed SpectroSERVER (DSS) setup, when the UIM managed hosts are modeled in DX NetOps Spectrum on landscapes other than Main Location Server (MLS), the UIM alarms that are raised on hosts may not get forwarded to the host models in non-MLS landscapes.

Solution:

To fix this issue, create the EventAdmin models manually for UIM Integration on all the landscapes in DSS when the integration is enabled. Review the following scenarios before creating the EventAdmin model manually:

Scenario 1:

For fresh integration or if the UIM models (existing models or UIM host models) are on landscapes other than MLS, create the EventAdmin models manually on other landscapes so that the alarms are forwarded to hosts.

> **NOTE**
>
> Use the default options while creating the EventAdmin model and do not enable the Alert_Forwarding_Enabled attribute.

Follow these steps:

1. To launch the OneClick Console, select Start Console at the top of the OneClick page, and log in as a DX NetOps Spectrum administrator.
2. Select the SpectroSERVER and Universe on the Explorer tab of the OneClick Navigation panel.
3. Select the Topology tab on the Contents panel and click the Create a New Model by Type icon.
   The Select Model Type dialog appears.
4. Click the All Model Types tab.
5. Select EventAdmin and click OK.
   The Create Model of Type dialog appears.

6. Enter the name and IP address of the UIM server and click OK.
   The UIM server is added to the topology as the selected model type.
   For more information about creating a model in OneClick, see the Modeling and Managing Your IT Infrastructure section.

Scenario 2:

If UIM Integration is enabled through the Southbound Gateway and an EventAdmin model already exists on a landscape, the attribute SBG_AlertForwardingEnabled must be enabled for the existing EventAdmin. The EventAdmin models must be created manually on other landscapes.

Follow these steps:

1. To create an EventAdmin model manually, follow the instructions that are documented for Scenario 1.
2. To enable the SBG_AlertForwardingEnabled attribute, select the EventAdmin in the OneClick Topology.
3. Select the Attributes tab in the Component Detail panel.
4. Select SBG_AlertForwardingEnabled in the left window of the Attributes panel.
   The attribute is added to the right window of the Attributes panel.
5. Double-click SBG_AlertForwardingEnabled in the right window, and select Yes. Click OK.
   The SBG_AlertForwardingEnabled attribute is enabled.

   NOTE
   You must delete EventAdmin and the associated event models when the integration is disabled.

Unnecessary minor alarms are noticed on VMware models

Symptom:

You may notice unnecessary minor alarms on VMware models

Solution:

To fix this issue, configure the SNMP gateway probe to avoid unnecessary minor alarms. For more information, see Deploy and Configure Probes.

If DX NetOps Spectrum and CA Unified Infrastructure Management Integration is disabled on second SpectroSERVER, you may notice Stale Nimsoft models on primary SpectroSERVER.

Stale Nimsoft Models on Primary SpectroSERVER

Symptom:

If DX NetOps Spectrum and UIM Integration is disabled on second SpectroSERVER, you may notice Stale Nimsoft models on primary SpectroSERVER.

Solution:

To fix this issue, you can delete the VMs and the rest of the entities in DX NetOps Spectrum OneClick console. To delete the entities, follow these steps:

1. In the DX NetOps Spectrum OneClick console, select the corresponding model, click the attribute tab, and search **“EditModelMask”.**
2. Change the value of **“EditModelMask”** from 255 to 0 and click **Save.**
3. Right-click that entity and select the **Delete** option.

Traps sent from snmpgtw are not being processed in DX NetOps Spectrum after enabling the DX NetOps Spectrum-UIM integration.

Symptom analysis:

This behavior is observed mainly in environments where the snmpgtw (snmp gateway) probe is deployed on a different server, i.e. other than the integrated UIM server (as mentioned in UIM Configuration page).

Solution:
You have to manually create an **EventAdmin** with the IP address of server/machine where the snmpgtw probe is deployed, and the traps will be handled properly using the SBGW probe (South Bound gateway) functionality.

> **NOTE**
>
> You also need to set the **SBGW_AlertForwarding** attribute to **Yes**.

Unknown alerts received for the NimsoftHost model

Symptom analysis: Unknown alert received trap mainly occurs on **NimsoftHostServer** models. The machine on which snmpgtw probe is deployed is also reported as **NimsoftHostServer** model in DX NetOps Spectrum, hence the models are found with the same IP address; as **EventAdmin** and as **NimsoftHostServer** model types.

When a trap is received, DX NetOps Spectrum forwards the trap to both the models with same IP Address.
As SBGW functionality works with EventAdmin model it processes the trap and **NimsoftHostServer** model fails to recognize it. Hence these unknown alerts are generated.

Solution:
Navigate to **NimsoftHost** model > **DX NetOps Spectrum Modeling Information** sub view > Select **Yes** for the **Disable Trap-Based Events** drop-down option.
The default value is **No**.
No unknown alert events are observed after you disable this option.

Missing Inventory due to Vcenter being hosted on a VM

Scenario 1: UIM reports two entities with different CS_ID for the same host which is a VM and has a vcenter deployed on it (Vcenter hosted on VMs managed by other vcenters), which causes the hierarchy in DX NetOps Spectrum to be deleted during the subsequent syncs with UIM.

Solution: Model the Vcenters first, consequently the VM which is acting as a vcenter will not be modeled, thus preserving the hierarchy in DX NetOps Spectrum

Scenario 2: UIM reports one entity for a host which is a VM and has a vcenter hosted on it. Sometimes the relationship for such a VM with its parent ESX host is not reported from UIM in subsequent syncs which causes that VM to be deleted in DX NetOps Spectrum and since it is also a vcenter the hierarchy also gets deleted in DX NetOps Spectrum.

Solution: The relationship between such a VM and its parent ESX host will not be modeled thus preserving the hierarchy in DX NetOps Spectrum.

UIM Models in Lost and Found/Cluttered Universe view

Symptom: Sometimes the relationships between ESX host and child VMs are not reported in some syncs. This causes all such VMs to be moved to the Lost and Found node. Also, All the UIM models are modeled under Universe by default which clutters the topology view of the Universe

Solution: A new container called **UIM Inventory** is now introduced where all the UIM entities will be placed upon creation in DX NetOps Spectrum. Also, in case the ESX to VM relation is not reported then such VMs will be moved to the said container thus preventing the models in the Lost and Found issue.

> **NOTE**
>
> The customer can move this UIM Inventory container across the navigation hierarchy and rename it as per his needs.

Not all UIM models are modeled in DX NetOps Spectrum

Symptom: In some instances, ESX hosts, Resource Pools and VMs are reported as children of an entity whose type is “COMPUTE_RESOURCE”. This is an entity type which DX NetOps Spectrum does not recognize which causes DX NetOps Spectrum to not process the children of such entities.

Solution: DX NetOps Spectrum has been modified to recognize **“COMPUTE_RESOURCE”** entity type, thus enabling modeling the children of such entities

OneClick Client hangs when metric views of UIM models are expanded

Symptom: Sometimes OneClick Client becomes unresponsive when the user clicks the metric views of the UIM models. OneClick Server makes individual REST calls for each metric and till the response is received from the UIM, the OneClick client becomes unresponsive.
Solution: A single **nisapi** call is made to fetch data of all the metric types for each metric family instead of multiple calls in order to improve the performance.

Models found in LostFound container

Symptom: Models found in LostFound container
Probable Cause: If there are any Virtual Machines under vApp, after disabling VMware Management integration, those VMs might be found under LostFound container
Solution: Users should manually delete those models from the LostFound container before integrating DX NetOps Spectrum and UIM (VMware or Server Management) again.

Multiple JSON exception(s) in Tomcat log

Symptom: Users might observe multiple JSONException in tomcat log (See below for exception details) during DX NetOps Spectrum- UIM Integration:

```
Error in getting the property PrimaryOSNameorg.json.JSONException: JSONObject["PrimaryOSName"] not found.
at org.json.JSONObject.get(JSONObject.java:516)
at com.ca.nimsoft.integration.nis.nisservice.NimsoftRESTEntity.getProperty(NimsoftRESTEntity.java:37)
at com.ca.nimsoft.integration.manager.NimosftHostServerModelTask.call(NimosftHostServerModelTask.java:55)
at java.util.concurrent.FutureTask.run(FutureTask.java:266)
at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142)
at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:617)
at java.lang.Thread.run(Thread.java:745)
```

Probable Cause: If any inventory from UIM does not contain Primary OperatingSystem Name, JSONException is thrown in the tomcat log
Solution: Users can ignore these as they do not impact the functionality

Unable to delete UIM Inventory container from source landscape

Symptom: UIM Inventory container doesn't delete from source landscape even though there are no datacenters after migration
Solution: Users should manually delete the 'UIM Inventory' container. However, leaving it as is, without deleting, will not affect any functionality

Server Hierarchy under UIM manager is empty in source landscape

Symptom: Under UIM Manager, Server hierarchy remains empty in source landscape even though there are no datacenters after migration
Probable Cause: If there are many Virtual machines that were part of a datacenter, migration process will take more time than expected if the 'Incremental Sync Time Interval' and 'VMWare Sync Time Interval' are less than 15 minutes

Solution: You should set '**Incremental Sync Time Interval**' and '**VMWare Sync Time Interval**' to its default value. You can manually delete **'Server'** hierarchy. However, leaving it as is, without deleting, will not affect any functionality

Servers does not migrate immediately when Datacenter(s) are migrated.

Symptom: Servers does not migrate immediately when Datacenter is migrated. Server migration happens only after subsequent syncs
Probable Cause: When there are many host servers that were part of a datacenter, and if the '**Incremental Sync Time Interval**' and '**VMWare Sync Time Interval'** are less than 15 minutes, the migration process will take more time than expected,
Solution: You should set '**Incremental Sync Time Interval**' and '**VMWare Sync Time Interval**' to its default value.

Errors in Unidirectional alarms synchronization, when the synchronization options are flipped in the spectrumgtw probe Admin Console

Symptom: Unidirectional alarms sync has a problem when the option is flipped between '**DX NetOps Spectrum to UIM**' and '**UIM to DX NetOps Spectrum**' or vice-versa,
Probable Cause: If users are using any ems version lesser than v8.4.3 this problem is observed
Solution: Deploy latest version on ems probe version 8.4.3

Appendix. Event Management using spectrumgtw probe

DX NetOps Spectrum-UIM Integration Events Mapping (Unidirectional and Bidirectional)

DX NetOps Spectrum receives a trap from UIM. DX NetOps Spectrum uses originating events based on AlertMap mapping to raise different event codes based on event conditions/severity.
DX NetOps Spectrum uses 0x06330010, 0x06330011, 0x06330012,0x06330013 event codes as originating events. For a detailed list of event code mappings based on the originating event, please go through the rest of the page.

If you have integrated DX NetOps Spectrum with UIM using the SNMP Gateway probe and SBGW / Southbound gateway, and want to understand the changes in how events and event codes are managed using spectrumgtw probe, please refer to the mappings below.

If the originating event used is **0x6330010**, subsequently, based on event conditions, appropriate events are generated as shown in the table below:

Event Condition	Generated event Code	Severity	Cleared Cause Codes
If the powerState of UIM Model is OK	0x06330062	Clear	0x06330058, 0x06330059
If the communication of UIM model through Ping is successful	0x06330057	Clear	0x06330056
If the HostPowerState of UIM Model is OK	0x06330063	Clear	0x06330060, 0x06330061
If Power On Event is generated	0x06330065	Clear	0x06330064
if Powered On Event is generated	0x06330065	Clear	0x06330064
If UIM Model is in connected state	0x06330065	Clear	0x06330064
if the Status of UIM Model is Power On	0x06330065	Clear	0x06330064
If the GuestState of UIM Model is OK	0x06330062	Clear	0x06330058, 0x06330059

If the Maintenance Mode of UIM Model is OK	0x06330067	Clear	0x06330066
Default (if above event conditions don't match)	0x06330000	Clear	0x06330001, 0x06330002, 0x06330003
if varbind 102 value matches with octet String 1.1.1.1 (Then Disk alarm is raised)	0x06330030	Clear	0x63300031, 0x63300041, 0x63300051,
if varbind 102 value matches with octet String 1.1.1.2 (Then memory alarm is raised)	0x06330040	Clear	0x63300032, 0x63300042, 0x63300052,
if varbind 102 value matches with octet String 1.1.1.3 (Then CPU alarm is raised)	0x06330050	Clear	0x63300033, 0x63300043, 0x63300053,

NOTE
No event/alarm is raised on DX NetOps Spectrum using 0x06330010, this is used for internal event condition mapping.

If the originating event used is **0x06330011**, later, based on event conditions, appropriate events are generated as shown in the table below:

Event Condition	Generated event Code	Severity
If Powered Off Event is generated	0x6330064	Critical
if UIM Model is in not connected State	0x6330064	Critical
If Powered Off Event is generated	0x6330064	Critical
if UIM Model is in not running state	0x6330058	Critical
if UIM model is in unknown state	0x6330058	Critical
UIM Model's maintenance state is true	0x6330066	Major
if UIM Model has entered maintenance state	0x6330066	Major
if UIM Model has exited maintenance state	0x6330067	Clear
if varbind 102 value matches with octet String 1.1.1.1 (Then Disk alarm is raised)	0x06330031	Minor
if varbind 102 value matches with octet String 1.1.1.2 (Then memory alarm is raised)	0x06330041	Minor
if varbind 102 value matches with octet String 1.1.1.3 (Then CPU alarm is raised)	0x06330051	Minor
Default (if above event conditions don't match)	0x6330001	Minor

NOTE

No event/alarm is raised on DX NetOps Spectrum using 0x06330011, this is used for internal event condition mapping.

If the originating event used is **0x06330012**, later, based on event conditions, appropriate events are generated as shown in the table below:

Event Condition	Generated event Code	Severity
UIM Model communication failed through Ping	0x06330056	Critical
HostPowerState of UIM Model is poweredOFF	0x06330060	Critical
HostPowerState of UIM Model is Unknown	0x06330061	Critical
PowerState of UIM Model is poweredOff	0x06330058	Critical
PowerState of UIM Model is Suspended	0x06330059	Critical
UIM Model is Power Off event generated	0x06330064	Critical
UIM Model is PoweredOff, an event is generated	0x06330064	Critical
The Status of UIM Model is Power Off	0x06330064	Critical
The GuestState of UIM Model is not running	0x06330058	Critical
The GuestState of UIM Model is unknown	0x06330058	Critical
if varbind 102 value matches with octet String 1.1.1.1 (Then Disk alarm is raised)	0x06330032	Major
if varbind 102 value matches with octet String 1.1.1.2 (Then memory alarm is raised)	0x06330042	Major
if varbind 102 value matches with octet String 1.1.1.3 (Then CPU alarm is raised)	0x06330052	Major
Default (if above event conditions don't match)	0x06330002	Major

NOTE

No event/alarm is raised on DX NetOps Spectrum using 0x06330012, this is used for internal event condition mapping

If the originating event used is **0x06330013**, later, based on event conditions, appropriate events are generated as shown in the table below:

Event Condition	Generated event Code	Severity
HostPowerState of UIM Model is poweredOFF	0x06330060	Critical
HostPowerState of UIM Model is Unknown	0x06330061	Critical
PowerState of UIM Model is poweredOff	0x06330058	Critical
PowerState of UIM Model is Suspended	0x06330059	Critical
UIM Model is Power Off event generated	0x06330064	Critical
UIM Model is PoweredOff, an event is generated	0x06330064	Critical
The Status of UIM Model is Power Off	0x06330064	Critical
The GuestState of UIM Model is not running	0x06330058	Critical

The GuestState of UIM Model is unknown	0x06330058	Critical
if varbind 102 value matches with octet String 1.1.1.1 (Then Disk alarm is raised)	0x06330033	Critical
if varbind 102 value matches with octet String 1.1.1.2 (Then memory alarm is raised)	0x06330043	Critical
if varbind 102 value matches with octet String 1.1.1.3 (Then CPU alarm is raised)	0x06330053	Critical
Default (if above event conditions don't match)	0x06330003	Critical

NOTE

No event/alarm is raised on DX NetOps Spectrum using 0x06330013, this is used for internal event condition mapping

NOTE

In case the device is in maintenance mode, event 0x06330066 is raised to put device in maintenance mode and, event 0x06330067 is raised to remove the device from maintenance mode.

DX NetOps Spectrum- UIM Bidirectional Events:
DX NetOps Spectrum receives alarms from spectrumgtw probe, an originating event **0x06330071,** is used in Spectrum to raise events based on event conditions.

Event Condition	Generated event Code	Severity	Cleared Cause Codes
if UIM Model has entered maintenance state	0x6330066	Maintenance	
if UIM Model has exited maintenance state	0x6330067	Clear	0x06330066
if UIM Model maintenance state is true	0x6330066	Maintenance	
if UIM Model maintenance state is OK	0x6330067	Clear	0x06330066

Minor, Major, Critical alarms will be raised based on severity varbind {S 101).

Event	Generated event Code	Severity	Cleared Cause Codes
If Minor event raised	0x0633006e	Minor	0x0633006f, 0x06330070
If Major event is raised	0x0633006f	Major	0x0633006e, 0x06330070
If Critical Event is raised	0x06330070	Critical	0x0633006e, 0x0633006f
Default (If we receive an event from UIM and no event condition or severity is matched)	0x06330076		

NOTE

No event/alarm is raised on Spectrum using 0x06330071, this is used for internal event condition mapping.

Events Codes for unidirectional and bidirectional UIM Integration		
Severity	**Using SouthBound Gateway Functionality (Unidirectional Integration)**	**Using Spectrum Gateway (Bidirectional Integration)**
Major	0x6330066, 0x06330032,0x06330042, 0x06330052, 0x06330002	0x0633006f
Minor	0x06330031, 0x06330041, 0x06330051, 0x06330001	0x0633006e
Critical	0x06330064 , 0x6330058, 0x06330056, 0x06330060, 0x06330061, 0x06330059, 0x06330064, 0x06330033, 0x06330043, 0x06330053, 0x06330003	0x06330070, 0x06330072, 0x06330074
Clear	*Refer below*	Cleared using IH

	event Code	**Cause Codes cleared**
	0x06330062	0x06330058, 0x06330059
	0x06330057	0x06330056
	0x06330063	0x06330060, 0x06330061
	0x06330065	0x06330064
	0x06330065	0x06330064
For Clear	0x06330065	0x06330064
	0x06330065	0x06330064
	0x06330062	0x06330058, 0x06330059
	0x06330067	0x06330066
	0x06330000	0x06330001, 0x06330002, 0x06330003
	0x06330030	0x63300031, 0x63300041, 0x63300051,
	0x06330040	0x63300032, 0x63300042, 0x63300052,
	0x06330050	0x63300033, 0x63300043, 0x63300053,

Additional Event Mapping from UIM Probes

Till 10.2, apart from events related to VMWare and servers, events from UIM were considered as generic events by DX NetOps Spectrum. As a result users were unable to create specific rules for those events due to lack of unique event codes. From the 10.2.1 release those events are also to be mapped to unique event code so that you can customize them and manage the events raised on UIM more effectively.

Please see the tables below to see the mapping of events from the probes to unique event codes in DX NetOps Spectrum:

sqlserver (SQL Server Monitoring) probe

The SQL Server Monitoring (sqlserver) probe constantly monitors the internal performance and space allocation of SQL Server databases. The probe can run locally on the database server or it can be configured to run as a remote client. The probe feeds essential information that is based on predefined criteria to the UIM availability manager for appropriate

alert notification, as required. An extensive range of checkpoints can be selected and individually scheduled to meet your monitoring requirements. The probe will run selected SQLs to extract vital information about your SQL Servers. The information is presented to database-administrator as alarms or reports. For more information, see sqlserver Metrics.

Checkpoint Metric	Description	Metric Type	DX NetOps Spectrum Event ID	Units
active_connection_ratio	Monitors the numbers of active connections.	4.2:51	0x06330077	Percent
active_users	Monitors the number of users having an active transaction at the moment of the snapshot.	4.2:1	0x06330078	Count
agent_job_failure	Monitors failed agent jobs in a defined time interval. **Note:** This monitor does not generate a clear alarm, by default. You can change the value of the **clear_alarms** key to **1**, using the **Raw Configure** option, to generate a clear alarm.	4.2:57	0x06330079	Count
alloc_space	Monitors allocated space.	4.2:2	0x0633007a	Percent
av_fragmentation	Monitors average fragmentation.	4.2:3	0x0633007b	Percent
average_waittime	Monitors average lock wait time interval in ms. High wait time causes performance degradation, consider an increase number of locks available or computer memory.	4.2:4	0x0633007c	Milliseconds
backup_status	Monitors in minutes, since the last database backup has been taken. For all databases that have never been backed up, this checkpoint returns -1 value.	4.2:5	0x0633007d	Minutes
blocked_users	Monitors the number of blocked users.	4.2:48	0x0633007e	Count
buf_cachehit_ratio	Percentage of pages found in the buffer cache without having to read from the disk. The ratio is the interval number of cache hits divided by the interval number of cache look-ups. **Note**: As reading from the cache is less expensive than reading from disk, you want this ratio to be high. Generally, you can increase the buffer cache hit ratio by increasing the amount of memory available to the SQL Server.	4.2:6	0x0633007f	Percent

check_dbalive	Attempts to connect to the server. This checkpoint cannot be deactivated and returns two values with which threshold comparison is done: • Sql Server instance connection failure: 0 • Sql Server instance connection success: 1	4.2:7	0x06330080	State
connection_memory	Monitors the amount of memory in KB that is used to maintain connections to SQL Server.	4.2:8	0x06330081	Kilobytes
database_count	Change in the number of databases on the server.	4.2:9	0x06330082	Count
database_size	Monitors the database status value. The status value is actually a combination of some configuration options and a status, therefore, there can be multiple values set simultaneously (such as "torn page detection" and "loading").	4.2:10	0x06330083	Megabytes
database_state	State of the sql_server database. For example, the database can be in any one of the following states: • Online • Offline • Restoring • Suspect	4.2:11	0x06330084	State
deadlocks	Monitors the number of deadlocks per second in an interval. **Note:** As deadlocks can cause a severe performance penalty, the count must be close to 0. Use trace 1204 or 1205 to identify the deadlocked resources and involved applications. Procedure such as sp_lock delivers useful information about locking.	4.2:12	0x06330085	Count/s
differential_backup_status	Monitors time in minutes since last differential backup.	4.2:50	0x06330086	Minutes
fg_free_space	Monitors the amount of free space in database file groups in percent. If there is at least one file with "unlimited" growth in a file group, the space in this file group is considered as 100 percent free.	4.2:64	0x06330087	Percent
free_connections	Monitors the percentage of free connections to SQL Server instance, specified by parameter 'user connections' (the maximum value must be 32676).	4.2:14	0x06330088	Percent

free_space	Monitors the amount of free space in data files in percentage. **Note:** If there is at least one file with "unlimited" growth, the space in the whole database is considered as 100% free. If you are using file groups, this could be misleading; therefore, you must deactivate this checkpoint and use only the "fg_free_space" checkpoint.	4.2:15	0x06330089	Percent
full_scans	Monitors the number of full table or index scans per second in the interval. If this value is high (2-10), then you must analyze your queries.	4.2:16	0x0633008a	Count/sec
latch_waits	Monitors the number of latch requests in an interval that could not be granted immediately and has to go in a wait state. **Note:** If this number is high, the system experiences a low cache hit ratio and is forced to perform physical I/O operations. Add more memory or increase the bandwidth of your system.	4.2:17	0x0633008b	Request/sec
lock_memory	Monitors the amount of allocated lock memory in KB.	4.2:18	0x0633008c	Kilobytes
lock_requests	Monitors the number of lock requests per second in the interval.	4.2:19	0x0633008d	Requests/Sec
lock_timeouts	Monitors number of lock-timeouts per second in interval with a precision of 0.001sec	4.2:20	0x0633008e	Count/sec
lock_waits	Monitors the number of locks waits per second in the interval.	4.2:21	0x0633008f	Count/sec
locked_users	Monitors the number of users suspended by locks at a given moment. Also, the blocked user and its current SQL are displayed.	4.2:22	0x06330090	Count
locks_used	Monitors the percentage of used lock and lock owner blocks.	4.2:23	0x06330091	Percent
log_cachehit_ratio	Monitors the percentage of pages found in the log cache without having to read from disk. The ratio is the interval number of cache hits divided by the interval number of cache look-ups. **Note**: Reading from the cache consumes less resources than reading from disk. You want this ratio to be high. You can increase the log cache hit ratio by increasing the amount of memory available to the SQL Server.	4.2:24	0x06330092	Percent

log_file_growths	Monitors the number of times in an interval the transaction log for the database has been expanded. If this happens often, you must consider re-sizing your log files.	4.2:25	0x06330093	Count
log_file_shrinks	Monitors the number of times in an interval the transaction log for the database has been decreased. If this happens often, you must consider re-sizing your log files.	4.2:26	0x06330094	Count
log_flush_waits	Monitors the number of commits per second waiting on the log flush in an interval. When commits are waiting for log flushes, the log device is usually the bottleneck.	4.2:27	0x06330095	Count/sec
logfile_size	Monitors the size of transaction log in MB for at least one transaction log file with "unlimited" growth in a database. **Note:** For this checkpoint, wherever the given database is in the recovery or restore mode, no metric values are reported for given interval of execution.	4.2:52	0x06330096	Count
logfile_usage	Monitors the amount of free space in the transaction log in percentage. If there is at least one transaction log file with "unlimited" growth in a database, the space in the transaction log is considered as 100 percent free. **Note:** For this checkpoint, wherever the given database is in the recovery or restore mode, no metric values would be reported for a given interval of execution.	4.2:28	0x06330097	Percent
logic_fragment	Monitors the number of cluster index pages that are out of order. Any number higher than 10% indicates external fragmentation. The index should be rebuilt. **Note:** Non-cluster indexes are not monitored because a table can have only one clustering sequence.	4.2:29	0x06330098	Percent
login_count	Monitors the number of users having an open connection to the server at a given time	4.2:30	0x06330099	Count
long_jobs	Monitors all jobs running longer than the defined threshold in seconds.	4.2:31	0x063300a0	Count
long_queries	Monitors all queries running longer than the defined threshold in seconds.	4.2:32	0x063300a1	None
mirror_sqlinstance	Monitors the availability of SQL server instance hosting the mirror database.	4.2:55	0x063300a2	State

mirror_state	Monitors mirror database state.	4.2:53	0x063300a3	State
mirror_witness_server	Monitors state of the witness server in the database mirror session.	4.2:54	0x063300a4	State
optimizer_memory	Monitors amount of memory in KB that is used for SQL optimizer.	4.2:33	0x063300a5	Kilobytes
page_reads	Monitors the number of physical database page-reads that are issued per second in an interval. **Note:** Since physical I/O is expensive, you can minimize the cost, either by using a larger data cache, intelligent indexes, more efficient queries, or by changing the database design.	4.2:34	0x063300a6	Counts/sec
page_writes	Monitors the number of databases page-writes that are issued per second in an interval. **Note:** Page-writes are generally expensive. Reducing page-write activity is important for optimal tuning. One way to do this is to ensure that you do not run out of free buffers in the free buffer pool. If you do, page-writes occurs while waiting for an unused cache buffer to flush.	4.2:35	0x063300a7	Counts/sec
scan_density	Monitors the ratio between the best number of extents to the actual number of extents. It should be near 100 percent. A lower number indicates external fragmentation and the object must be reorganized.	4.2:36	0x063300a8	Ratio
server_cpu	Monitors the percentage of CPU usage by SQL Server instance in the interval.	4.2:37	0x063300a9	Percent
server_io	Monitors the percentage of I/O busy for SQL Server instance in the interval.	4.2:38	0x063300aa	Percent
server_startup	Number of days the database server is up and running.	4.2:39	0x063300ab	Days
suspect_pages	Monitors suspect pages logged for databases.	4.2:56	0x063300ac	Gauge
sqlcache_memory	Monitors the amount of memory in KB that is used for SQL statement cache.	4.2:40	0x063300ad	Kilobytes
table_space	Monitors the amount of space (in KB/MB) reserved for a particular table in a database. This checkpoint can be used to control the size of fast-growing tables.	4.2:41	0x063300ae	Kilobytes
total_memory	Monitors the total amount of dynamic memory (in KB) that the server uses currently.	4.2:42	0x063300af	Kilobytes

transaction_backup_status	Sends QoS and Alarms for those databases that are running in full or bulk-logged recovery mode. **Note**: This checkpoint does not send QoS and Alarms for databases that are running in simple recovery mode.	4.2:49	0x063300b0	Minutes
transactions	Monitors the number of transactions per second in the interval.	4.2:43	0x063300b1	Transactions/sec
user_cpu	Monitors the percentage of CPU usage by user in interval. **Note:** The checkpoint **user_cpu** reports **$spid.$hostid** in the QoS target. This results in the creation of new data series for each new **$spid** or **$hostid**. CA recommends disabling the QoS for this checkpoint.	4.2:44	0x063300b2	Percent
user_waits	Monitors time in seconds for session spent waiting for a lock and length of blocking. **Note:** You can add schedules in the **Exclude** and **Include** lists. The match expression, which is added will be executed in the given time period mentioned in the schedule.	4.2:45	0x063300b3	Seconds
workspace_memory	Monitors the amount of memory in KB that is used for executing processes such as hash, sort, bulk copy, and index creation operations.	4.2:46	0x063300b4	Percent
ls_primary_status	Monitors the collective status of agents for the primary log shipping database. This checkpoint must run from the primary server or monitor server. The status can be as follows: • healthy and no-agent failures 1 • otherwise 0	4.2:58	0x063300b5	Status
ls_time_since_last_backup	Monitors time in minutes since the last backup.	4.2:60	0x063300b6	Minutes
ls_secondary_status	Monitors the collective status of agents for the secondary log shipping database. This checkpoint must run from a secondary server or monitor server. The status can be as follows: • healthy and no-agent failures 1 • otherwise 0	4.2:59	0x063300b7	Status
ls_time_since_last_copy	Monitors time in minutes since the last copy.	4.2:61	0x063300b8	Minutes
ls_time_since_last_restore	Monitors time in minutes since the last restore.	4.2:62	0x063300b9	Minutes
ls_last_restored_latency	Monitors time in minutes since last restored latency.	4.2:63	0x063300ba	Minutes

fg_freeSpace_with_avail_disk	Monitors the amount of free disk space in database file groups in %. Free space for file groups (with auto-growth enabled) is calculated after considering the available disk size on which the file group is located. **Notes:** • A single query is executed for all the databases of the SQL server. If any of the databases fails to execute the query, the query is considered as failed for the SQL server. • You require System Administrator privileges on the database server to execute this checkpoint.	4.2:64	0x063300bb	Percent
logfile_usage_with_avail_disk	Monitors free space in the database log files after considering the available disk size. **Note:** You require System Administrator privileges on the database server to execute this checkpoint.	4.2:65	0x063300bc	Percent
aag_cluster_members_state	Monitors the state of the nodes of all AlwaysOn availability groups of WSFC. Each number is assigned a value, as follows: • Offline: 0 • Online: 1	4.2.3:2	0x063300bd	State
aag_cluster_quorum_state	Monitors the quorum state of all AlwaysOn availability groups of Windows Server Failover Clustering (WSFC) cluster. Each number is assigned a value, as follows: • Unknown quorum state: 0 • Normal quorum: 1 • Forced quorum: 2	4.2.3:1	0x063300be	State

aag_db_page_status	Monitors the page state of each database in all the AlwaysOn availability groups of the server. Each number is assigned a value, as follows: • Queued for request from partner: 2 • Request sent to partner: 3 • Queued for automatic page repair (response received from partner): 4 • Automatic page repair succeeded and the page should be usable: 5 • Irreparable: 6 (This indicates that an error occurred during page-repair attempt, for example, because the page is also corrupted on the partner, the partner is disconnected, or a network problem occurred. This state is not terminal; if corruption is encountered again on the page, the page will be requested again from the partner.)	4.2.3:9	0x063300bf	State
aag_db_replica_synchronization_state	Monitors the synchronization state of each database replica in all AlwaysOn availability groups of the server. Each number is assigned a value, as follows: • Not synchronizing: 0 • Synchronizing: 1 • Synchronized: 2 • Reverting: 3 • Initializing: 4	4.2.3:8	0x063300c0	State
aag_listener_state	Monitors the listener state of all AlwaysOn availability groups of the server. Each number is assigned a value, as follows: • Offline: 0 • Online: 1 • Pending restart: 2 • Online: 3	4.2.3:10	0x063300c1	State
aag_replica_connected_state	Monitors the connected state of a replica in all AlwaysOn availability groups of the server. Each number is assigned a value, as follows: • Disconnected: 0 • Connected: 1	4.2.3:5	0x063300c2	State

aag_replica_operational_state	Monitors the operational state of a replica in all AlwaysOn availability groups of the server. Each number is assigned a value, as follows: • Pending failover: 0 • Pending: 1 • Online: 2 • Offline: 3 • Failed: 4 • Failed, no quorum: 5 • Replica is not local: NULL	4.2.3:7	0x063300c3	State
aag_replica_recovery_health	Monitors the recovery health of a replica in all AlwaysOn availability groups of the server. Each number is assigned a value, as follows: • Online_in_progress: 0 • Online: 1 • NULL	4.2.3:6	0x063300c4	State
aag_replica_synchronization_health	Monitors the synchronization health of a replica in all AlwaysOn availability groups of the server. Each number is assigned a value, as follows: • Not healthy: 0 (At least one joined database is in the NOT SYNCHRONIZING state.) • Partially healthy: 1 (Some replicas are not in the target synchronization state: synchronous-commit replicas should be synchronized, and asynchronous-commit replicas should be synchronizing.) • Healthy: 2 (All replicas are in the target synchronization state: synchronous-commit replicas are synchronized, and asynchronous-commit replicas are synchronizing.)	4.2.3:4	0x063300c5	State
aag_synchronization_health	Monitors the synchronization health of all AlwaysOn availability groups of the server. Each number is assigned a value, as follows: • Not healthy: 0 (None of the availability replicas have a healthy synchronization_health) • Partially healthy: 1 (The synchronization health of some, but not all, availability replicas is healthy.) • Healthy: 2 (The synchronization health of every availability replica is healthy.)	4.2.3:3	0x063300c6	State

wait_stats_count	Monitors the count of all the wait checkpoints, and displays the delta value of two intervals in the QoS. For example, if the current interval value is 20, and the previous interval value is 15, the delta value displayed is 5 (20-5). The alarm is generated if the delta value breaches the defined threshold.	4.2:71	0x063300c7	Count
wait_stats_time	Monitors the total wait time in milliseconds for each wait checkpoint, and displays the delta value of two intervals in the QoS. For example, if the current interval value is 30 seconds, and the previous interval value is 20 seconds, the delta value displayed is 10 seconds (30-20). The alarm is generated if the delta value breaches the defined threshold.	4.2:72	0x063300c8	Milliseconds

cdm (CPU, Disk, Memory Performance Monitoring) probe

The CPU, Disk, Memory Performance Monitoring (cdm) probe monitors the performance and resource load on the system with the robot. The UIM CPU, Disk & Memory (cdm) probe generates alarms that are based on configured threshold values and trending statistics. For more information see cdm Metrics.

Metric Name	Description	Metric Type	DX NetOps Spectrum Event ID	Units
system_uptime	Details how long the system has been on since its last restart.	1:1	0x063300c9	Seconds
system_reboot		1:2	0x063300ca	
system_disk_usage_mb	Aggregated disk usage in megabytes	1.1:2	0x063300cb	Megabytes
system_disk_usage_pct	Aggregated disk usage in percentage	1.1:3	0x063300cc	Percent
system_inode_usage_cnt	Total number of free file nodes in file system	1.1:4	0x063300cd	Count
system_inode_usage_pct	Total number of free file nodes in file system in percentage.	1.1:5	0x063300ce	Percent
system_shared_folder_availability	Populates the data depending upon the disk availability. The available options are: Missing, New and Ok.	1.1:7	0x063300cf	State
system_disk_usage_delta	Disk Usage Delta Error Warning Threshold - 8, Error Threshold - 200	1.1:8	0x063300d0	
system_disk_size_gb	Total size of the disk	1.1:9	0x063300d1	Gigabytes
system_read_throughput	Disk bytes read per second	1.1:58	0x063300d2	Bytes/Second
system_write_throughput	Disk bytes written per second	1.1:59	0x063300d3	Bytes/Second
system_total_throughput	Disk bytes read and written per second	1.1:60	0x063300d4	Bytes/Second

system_disk_partition_used_pct	Aggregated disk usage in percentage	1.1.2:2	0x063300d5	Percent
system_disk_partition_used_mbytes	Aggregated disk usage in megabytes	1.1.2:11	0x063300d6	Megabytes
system_user_cpu	The sum of CPU time when all CPUs of the system were executing the kernel or operating system	1.5:1	0x063300d7	Percent
system_user_cpu_pct	The percentage of time for which all CPUs of the system were used.	1.5:2	0x063300d8	Percent
system_system_cpu_pct	The CPU Usage Percentage detailis	1.5:3	0x063300d9	Percent
system_wait_cpu_pct	The percentage of time for which all CPUs of the system were waiting for I/O.	1.5:4	0x063300da	Percent
system_idle_cpu_pct	The percentage of time for which all CPUs of the system were idle.	1.5:5	0x063300db	Percent
system_multi_cpu_usage_diff_pct	Multi-CPU Usage Difference in percentage	1.5:6	0x063300dc	Percent
system_multi_cpu_usage_pct	The percentage of time for which an individual CPU of the system was used.	1.5:7	0x063300dd	Percent
system_data_collection	System data collection	1.5:8	0x063300de	
system_processor_queue_length	The current calculated average load of the system.	1.5:9	0x063300df	Processes
system_load_avg_1min	The average system load over the last one minute. **Note:** This metric is supported only on the Linux, AIX and HP-UX platforms.	1.5:68	0x063300e0	Count
system_load_avg_5min	The average system load over the last five minutes. **Note:** This metric is supported only on the Linux, AIX and HP-UX platforms.	1.5:69	0x063300e1	Count
system_load_avg_15min	The average system load over the last fifteen minutes. **Note:** This metric is supported only on the Linux, AIX and HP-UX platforms.	1.5:70	0x063300e2	Count
system_multi_cpu_system_pct	Multi-CPU System Usage in percentage	1.5.1:3	0x063300e3	Percent
system_multi_cpu_idle_pct	Multi-CPU System Idle in percentage	1.5.1:5	0x063300e4	Percent
system_memory_usage_mb	Total memory usage in megabytes	1.6:1	0x063300e5	Megabytes
system_memory_usage_pct	Total memory usage in percentage	1.6:2	0x063300e6	Percent
system_memory_data_collection	System memory usage in percentage	1.6:3	0x063300e7	Percent
system_system_memory_usage_pct	Total System memory utilization in percent. This metric is supported only on the Windows, Linux and AIX platforms.	1.6:57	0x063300e8	Percent
system_user_memory_usage_pct	Total User memory utilization in percent. This metric is supported only on the Windows, Linux and AIX platforms.	1.6:58	0x063300e9	Percent

system_physical_memory_usage_mb	The size of the physical memory used on the system in megabytes. **Note:** For this metric, the buffer cache will be subtracted from physical memory value if the key **mem_buffer_used** is set as **No**. This support is added for Linux, AIX, and HPUX platforms.	1.6:6	0x063300ea	Megabytes
system_physical_memory_usage_pct	Total Swap memory usage in percent	1.6:7	0x063300eb	Percent
system_swap_memory_usage_mb	Total Swap memory usage in megabytes	1.6:8	0x063300ec	Megabytes
system_inbound_traffic	The total number of bytes per second received by the server.	2.1.1:1	0x063300ed	Bytes/Second
system_outbound_traffic	The total number of bytes per second sent by the server.	2.1.1:2	0x063300ee	Bytes/Second
system_aggregated_traffic	The total number of bytes per second sent and received by the server.	2.1.1:3	0x063300ef	Bytes/Second

net_connect

net_connect (Network Connectivity Monitoring) probe measures network connectivity that is based on "ping" (ICMP ECHO) and the TCP connections to a list of user-defined services. The service can be NetBIOS, Telnet, FTP, and HTTP. The probe supports the UIM family of solutions by sending the quality of service (QoS) messages. For more information see net_connect Metrics.

Metric Name	Description	Metric Type	DX NetOps Spectrum Event ID	Units
net_connect_misc		2.2:1	0x063300f0	
net_connect_response_time	Response time for network connectivity.	2.2.1:1	0x063300f1	Milliseconds
net_connect_packet_loss	Percentage of Packet Loss in network connectivity response. Packet Loss = icmp_lost_count/ icmp_packet_count. Refer **Jitter and Latency QoS Calculation** in net_connect Metrics.	2.2.1:2	0x063300f2	Milliseconds
net_connect_nw_connectivity_response_pkt	Delay in time it takes a packet to cross a network. Latency = average(round_trip_latency); if icmp_lost_count is icmp_packet_count, then latency is NaN. (Nan - NULL QoS). Refer **Jitter and Latency QoS Calculation.**	2.2.1:3	0x063300f3	
net_connect_nw_conn_response_packet_loss_jitter	Variable Latency in a network. Jitter = stardard_deviation(round_trip_latency); if icmp_lost_count is icmp_packet_count, the jitter is 0. Refer **Jitter and Latency QoS Calculation in** net_connect Metrics	2.2.1:4	0x063300f4	Milliseconds

net_connect_tcp_response_time	Response time for network connectivity.	2.2.2:1	0x063300f5	Milliseconds

processes (Process Monitoring) probe

The (Process Monitoring) processes probe monitors the specified processes to detect any error situation. The probe also retrieves information about the process, for example, the CPU usage, memory usage, and so on. For more information see processes Metrics.

Metric Name	Description	Metric Type	DX NetOps Spectrum Event ID	Units
ipc_number_processes_cnt	The number of Instances of a process	1.26:1	0x063300f6	Count
ipc_processes_util	The percentage of the number of processes currently running in the system to the maximum number of processes configured in the system	1.26:2	0x063300f7	Percent
ipc_number_semaphore_sets	The number of semaphore sets currently used by the system	1.26:3	0x063300f8	Number
ipc_number_semaphore_sets_util	The percentage of Semaphore Sets currently used by the System to the maximum number of semaphore sets configured in the system	1.26:4	0x063300f9	Percent
ipc_message_queue_cnt	The number of message queues currently used by the system	1.26:5	0x063300fa	Count
ipc_message_queue_util	The percentage of message queues currently used by system to the maximum number of message queues configured in the system	1.26:6	0x063300fb	Percent
ipc_number_shared_mem_segments	The number of Shared Memory Segments currently used by the system	1.26:7	0x063300fc	Count
ipc_number_shared_mem_seg_util	The percentage of Shared Memory Segments currently used by the System to a maximum number of shared memory segments configured in the system.	1.26:8	0x063300fd	Count
processes_instances	The number of processes currently running in system	1.3:2	0x063300fe	Number
processes_state	The availability (up/down) of a process	1.3:3	0x063300ff	State (Up/Down)
processes_mem_usage	The bytes of memory utilized by a process	1.3:4	0x06330100	Bytes
processes_cpu_usage	The percentage of CPU utilization by a process	1.3:5	0x06330101	Percent
processes_threads	The number of threads of a process	1.3:6	0x06330102	Count
processes_unexpected_user	The process not running with expected user	1.3:22	0x06330103	Count
processes_handles_cnt	The number of handles of a process	1.3:42	0x06330104	Count

processes_resident_mem_usage	Process exceeds the expected resident memory usage **Note:** This metric is only supported on Linux.	1.3:45	0x06330105	Kilobytes

logmon (log monitoring) probe

The Log Monitoring (logmon) probe scans ASCII-based systems and application log files by matching specified expressions. Alarms are generated when the log file content matches the defined expression. The probe also extracts and stores metric data from the matched log file entry in the QoS database. For more information, see logmon Metrics.

Metric Name	Description	Metric Type	DX NetOps Spectrum Event ID
logmon_exit_code (only for Command profile)	Exit code threshold breach/ clear	1.2.3:4	0x06330106
logmon_url_response_probe_state (only for URL profile)	Contact Success (Clear) / Contact Failed	1.2.4:4	0x06330107
logmon_url_load_state	URL load success (Clear)/ Failed.	1.2.4:5	0x06330108
logmon_misc		1.2.5:4	0x06330109

rsp (Remote Systems) probe

The Remote System Probe (rsp) allows you to monitor system metrics. The probe collects performance data in an agent-less manner without installing proprietary software on the system. For more information, see rsp Metrics.

Metric Name	Description	Metric Type	DX NetOps Spectrum Event ID	Units
rsp_processes_owner	Alarm to be issued when the configured threshold is satisfied	1.3:1	0x0633010a	
rsp_processes_instances	A number of process instances.	1.3:2	0x0633010b	Number
rsp_processes_state	Process availability.	1.3:3	0x0633010c	State
rsp_processes_memory_usage	Process memory usage.	1.3:4	0x0633010d	Kilobytes
rsp_processes_cpu_usage	Process CPU usage.	1.3:5	0x0633010e	Percent
rsp_processes_threads	A number of process threads.	1.3:6	0x0633010f	Number
rsp_service_state	Network Service Availability	1.4:1	0x06330119	State
rsp_processor_queue_length	Processor queue length. The current calculated average load of the system.	1:3	0x06330116	Processes
rsp_multi_cpu_usage_pct	Difference between highest and lowest CPU usage on Multi-CPU systems. The metric calculates data in percentage for the individual CPU idle time, user time, system time, wait time, and CPU usage.	1.5.1:1	0x06330117	Percent
rsp_cpu_usage_pct	CPU usage. The metric calculates data in percentage for total usage, user time, system time, wait time, and idle time.	1.5:1	0x06330118	Percent

rsp_mem_usage_pct	Total memory usage in percent.	1.6:2	0x06330114	Percent
rsp_mem_paging	Memory paging in kilobytes per second.	1.6:4	0x06330115	Kilobytes/Second
rsp_number_events	Number of events in interval. This metric sends the count of matched events, irrespective of the profile.	1.2.1:1	0x06330113	Count
rsp_disk_availability		1.1:1	0x06330110	
rsp_disk_usage_mb	Aggregated disk I/O rate in megabytes.	1.1:2	0x06330111	Megabytes
rsp_disk_usage_pct	Aggregated disk I/O rate in percentage.	1.1:3	0x06330112	Percent

Integration with DX NetOps Performance Management

10.4 supports the SD-WAN monitored items such as Tunnels and SLA Paths that are reconciled in NetOps across Spectrum and Performance Management. This ensures performance threshold events are associated with the tunnel or path in Spectrum and the alarm console.

DX NetOps Spectrum - DX NetOps Performance Management integration lets you share models, Global Collections, and events between two powerful infrastructure management systems.

> **NOTE**
>
> The integration can be done between a single instance of DX NetOps Spectrum and a single instance of DX NetOps Performance Management. You cannot integrate one instance of DX NetOps Spectrum with multiple instances of DX NetOps Performance Management.

The DX NetOps Spectrum data source contributes the following item types to DX NetOps Performance Management:

- Devices - Like SNMP and non-SNMP (SD-Router)
- Interfaces
- Groups
- SDN_Tunnels
- SLA_Paths

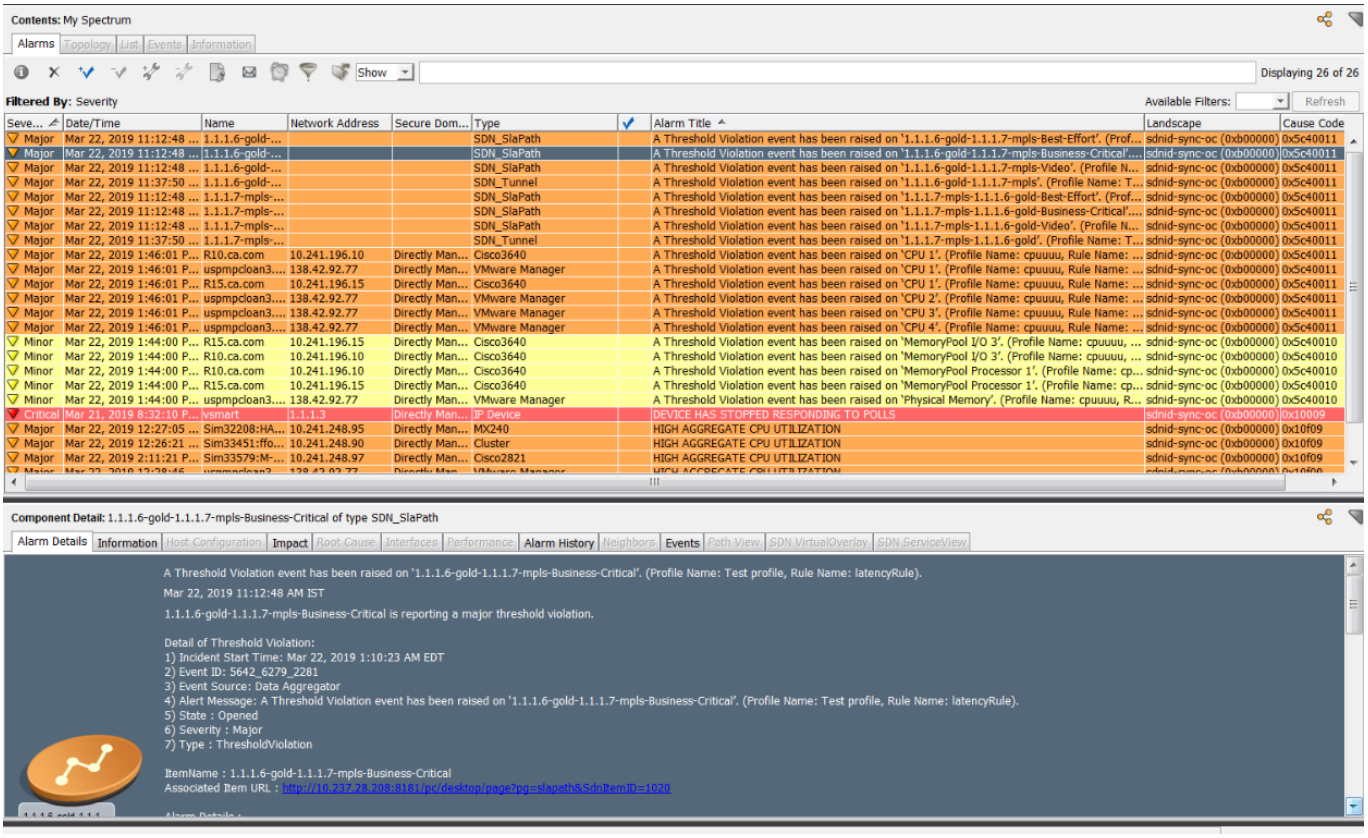
DX NetOps Spectrum also retrieves and displays infrastructure performance events from the DX NetOps Performance Management Event Manager. As a result, you can see performance and fault alarms side-by-side in OneClick.

DX NetOps Performance Management retrieves devices from DX NetOps Spectrum to extend the DX NetOps Performance Management device Inventory. You can determine which devices are retrieved. The interfaces that are associated with each device are added to the Inventory automatically; however, they are subject to CA Infrastructure Management Data Aggregator interface filtering.

DX NetOps Performance Management IP domains are synchronized and displayed in OneClick. You can add device models to them. The items in these IP domains are synchronized with DX NetOps Performance Management, and their data is included in dashboards. Your DX NetOps Spectrum Global Collections become groups in the DX NetOps Performance Management Groups tree.

The integration extracts event data from the DX NetOps Performance Management Event Manager and converts it into DX NetOps Spectrum events. These events then raise DX NetOps Spectrum alarms on models in the SpectroSERVER topology. As clear events are processed, corresponding DX NetOps Spectrum alarms are cleared automatically. Polling for supported events begins when synchronization has completed. These events are converted into DX NetOps Spectrum alarm set or clear events and asserted on models in each landscape.

Alarms based on device performance, such as Time over Threshold and Deviation from Normal events, are generated in DX NetOps Spectrum to supplement fault and availability monitoring. The DX NetOps Spectrum alarms that originate in the Event Manager can be viewed in the OneClick Console. The following image shows the alarm details in the OneClick console:



Finally, the integration enables a Data Aggregator data source to discover DX NetOps Spectrum devices without requiring you to manually create discovery profiles. The system creates discovery profiles, which are scheduled by default but can also be run manually.

Solution Architecture

The following facts describe the architecture of the DX NetOps Spectrum - DX NetOps Performance Management integration:

> **WARNING**
> DX NetOps Performance Management and OneClick Web Server should be in time sync. That is, time difference must not be there between these two machines, otherwise, it causes CA PM alarms missing in DX NetOps Spectrum.

- One SpectroSERVER or a Distributed SpectroSERVER (DSS) can be synchronized with DX NetOps Performance Management by specifying the OneClick web server as a DX NetOps Performance Management data source.
  - Full synchronization occurs when the data source is first added to DX NetOps Performance Management.

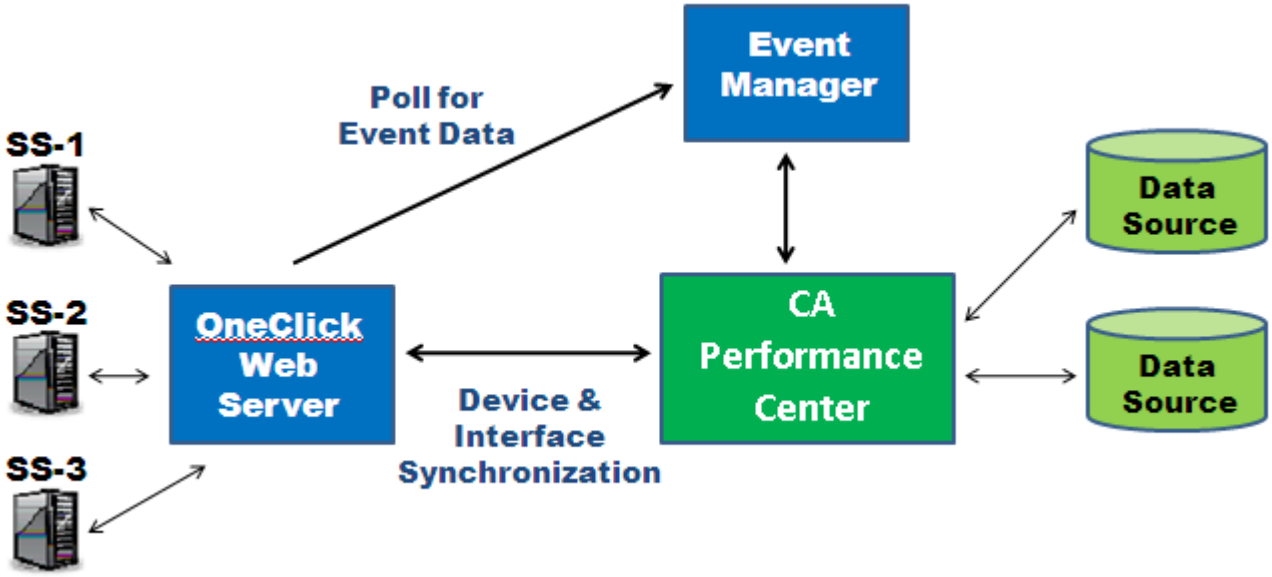
    > **WARNING**
    > If a full synchronization is required after you add the data source, we recommend running it during nonbusiness hours.
  - Incremental synchronization occurs every 5 minutes.

Additions, removals, and modifications of devices, interfaces, and Global Collections in DX NetOps Spectrum are reflected in DX NetOps Performance Management after incremental synchronization.

- Each landscape in the DSS is defined as a DX NetOps Performance Management group.
- Devices and interfaces in the DSS are synchronized with DX NetOps Performance Management and added to the appropriate landscape group.
- OneClick polls the Event Manager for events that are relevant to a specified landscape group. This polling happens every 60 seconds by default. Any retrieved events are then translated to DX NetOps Spectrum events, which can generate or clear alarms.

  WARNING
  DX NetOps Performance Management-related alarm processing in DX NetOps Spectrum applies to device and interface models that have been synchronized to DX NetOps Performance Management. Only events that are related to devices or interfaces that are modeled in DX NetOps Spectrum are processed by OneClick.
- The Event Manager database is polled for supported events at each polling interval. With DX NetOps Performance Management v2.0.00 and later, you can modify supported events.

The following image illustrates the architecture of the DX NetOps Spectrum - DX NetOps Performance Management integration:



Upgrade Considerations

In previous versions of the DX NetOps Spectrum - DX NetOps Performance Management integration, all DX NetOps Spectrum models were contributed to the DX NetOps Performance Management Inventory. And these models were always associated with the Default IP Domain. From DX NetOps Spectrum Release 9.4 and DX NetOps Performance Management Release 2.3, you can precisely control the items that DX NetOps Spectrum contributes to CA Infrastructure Management, and you can control IP domain membership. The OneClick features for creating and maintaining Global Collections can now be applied to IP domains.

Be sure to account for the following key considerations when you are planning for the integration:

- Only those models that you have added to a DX NetOps Performance Management IP Domain model in OneClick are synchronized with DX NetOps Performance Management.
- The devices are added to DX NetOps Performance Management based on the specific DX NetOps Performance Management IP Domain with which the device models are associated in DX NetOps Spectrum.
- If you are upgrading an existing integration, you must at a minimum define the contents of the default DX NetOps Performance Management IP Domain model in DX NetOps Spectrum. You can add additional IP domains as required in the DX NetOps Performance Management Admin pages.
- To move a device from the DX NetOps Performance Management Default IP Domain to another domain, you must first add the device to the desired IP Domain in DX NetOps Spectrum.
  If the device already exists in DX NetOps Performance Management, however, you must delete the device from the Default IP Domain. At the next synchronization, the device will be added to the updated IP domain in DX NetOps Performance Management.

Supported Features

Earlier versions of DX NetOps Spectrum integrate with both CA NetQoS Performance Center v6.1 and DX NetOps Performance Management v2.0.00 through r2.2.00. To integrate with DX NetOps Performance Management versions that predate r2.300, consult an earlier version of this section.

The following list identifies supported features in the integration between CA Spectrum starting from 9.4 and DX NetOps Performance Management 2.3.00.

Events

- ThresholdViolation events: These events from Data Aggregator and CA Network Flow Analysis data sources are integrated and supported by default in DX NetOps Spectrum.
- Other events: By updating an XML file and some event support files, you can instruct OneClick to handle events in the Event Manager database that were reported by other data sources.

IP Domains

- DX NetOps Performance Management IP domains are synchronized as DX NetOps Performance Management IP Domain models in OneClick. Place device models into DX NetOps Performance Management IP Domains manually, or define collection rules to dynamically collect device models. The contents of the DX NetOps Performance Management IP domain model are used to keep the IP domain membership up to date in DX NetOps Performance Management. As new devices are sent to DX NetOps Performance Management from DX NetOps Spectrum, they are assigned to the IP domain that corresponds to their DX NetOps Performance Management IP Domain membership in DX NetOps Spectrum.

Groups

- Your DX NetOps Spectrum Global Collections and landscapes are synchronized with DX NetOps Performance Management and displayed as groups in the DX NetOps Performance Management Groups tree. As part of the Groups tree, you can leverage the synchronized Global Collections in a variety of ways:
  - Create reporting groups
  - Define site membership
  - Drive the content of other custom groups and collections
- DX NetOps Spectrum devices can be added to DX NetOps Performance Management Service Provider groups and shared among multiple tenant users.

Drilldown from OneClick to DX NetOps Performance Management Performance Data

- You can access DX NetOps Performance Management performance data from DX NetOps Spectrum device and interface models. You gain rapid access to information about device performance issues in context.

Synchronized Discovery

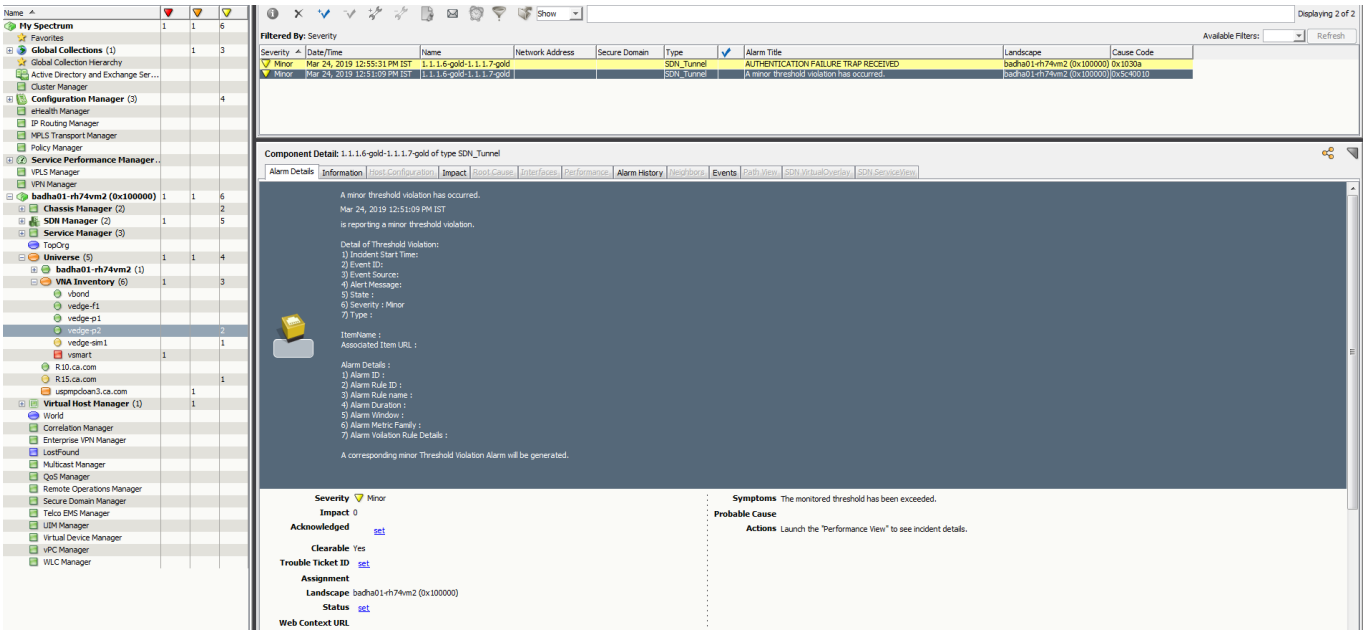
- Shared discovery eases the administrative burden.
- Place CA Infrastructure Management Data Collectors where you require them and leverage discovery data from multiple SpectroSERVERs. Determine the appropriate number of IP Domains that are required, and deploy a Data Collector for each IP Domain.
  At device inventory synchronization, Data Aggregator determines whether each device is new or is already in the inventory. When Data Aggregator encounters a device that it is not monitoring, it adds the IP address to a predefined Discovery profile. A discovery profile is defined for each IP Domain. The IP address for each device is added to the corresponding discovery profile based on its membership in a DX NetOps Performance Management IP Domain model within DX NetOps Spectrum. You can then run this Discovery profile manually, or you can let it run once a day through an automatically configured threshold.

Intelligent Interface Synchronization

- Device monitoring by DX NetOps Spectrum always includes all associated interfaces. DX NetOps Performance Management retrieves information about all interfaces from DX NetOps Spectrum.
  However, the interface inventory in DX NetOps Performance Management does not include interfaces that have only been contributed by DX NetOps Spectrum. Instead, the inventory is filtered to include interfaces that are monitored by a performance monitoring data source, such as Data Aggregator or CA Network Flow Analysis.

SDN_Tunnels and SLA_Paths

Following is a screenshot of the Threshold Alarm on SDN_TUNNEL:



Life cycle management

- Using DX NetOps Spectrum you can control the life cycle state of devices in DX NetOps Performance Management. Changes in DX NetOps Spectrum trigger changes in DX NetOps Performance Management. For more information, see the Manage Device Life Cycles section in the DX NetOps Performance Management - documentation.

Component Requirements

Contents

DX NetOps Spectrum - DX NetOps Performance Management integration requires the following component versions:

For more information about the integration of DX NetOps Spectrum with other CA products, see Integration Compatibility.

Model Synchronization Eligibility

One of the following sets of criteria must be met for a DX NetOps Spectrum model to be eligible for synchronization with DX NetOps Performance Management:

Models that are derived from DX NetOps Spectrum model type *Device* that have:

- A valid IP address
- Membership in a DX NetOps Performance Management IPDomain model
- A Model_State (attribute 0x1007c) of Active.

> **NOTE**
> If Model_State is not Active, the processing of the model is postponed until the next synchronization.

Models that are derived from DX NetOps Spectrum model type *Port* that have:

- A parent device that has been synchronized, which means that the parent device is Active and is a member of a DX NetOps Performance Management IP Domain
- A valid IfIndex value.

Device Model Synchronization

For device model synchronization, DX NetOps Spectrum-DX NetOps Performance Management integration uses Model_Class (attribute 0x11ee8) to determine the DX NetOps Performance Management SubType of a device, as follows:

Model_Class	SubType
Router	Router
Switch-Router	Router
Switch	Switches
Workstation-Server	Workstations

The default SubType of 'Other' is used for models whose Model_Class is not specified in the table.

> **NOTE**
> The table above shows the specific mappings of Model_Class to the device SubType for DX NetOps Spectrum data sources. If the device is contributed by additional data sources, such as Data Aggregator, it is possible that DX NetOps Performance Management will display a different SubType for the device.

How to Integrate with DX NetOps Performance Management

This section discusses the steps to configure the DX NetOps Spectrum and DX NetOps Performance Management integration:

If you plan to have Data Aggregator discover the devices that DX NetOps Spectrum contributes (the optional Step 5, below), register the Data Aggregator data source first, and enable the option to Discover Devices from other Data Sources. Data Collectors must also be installed for each IP Domain to which DX NetOps Spectrum will contribute to device models.

Follow these steps:

1. Configure DX NetOps Spectrum as a Data Source in DX NetOps Performance Management.
2. Enable Event Polling in DX NetOps Spectrum.

3. Configure SNMP Profiles in DX NetOps Performance Management.
4. Add Models to IP Domain Global Collections.
5. (Optional) Enable Discovery Synchronization with CA Infrastructure Management Data Aggregator.

The optional step to enable discovery synchronization involves some configuration in DX NetOps Performance Management and in the Data Aggregator component.

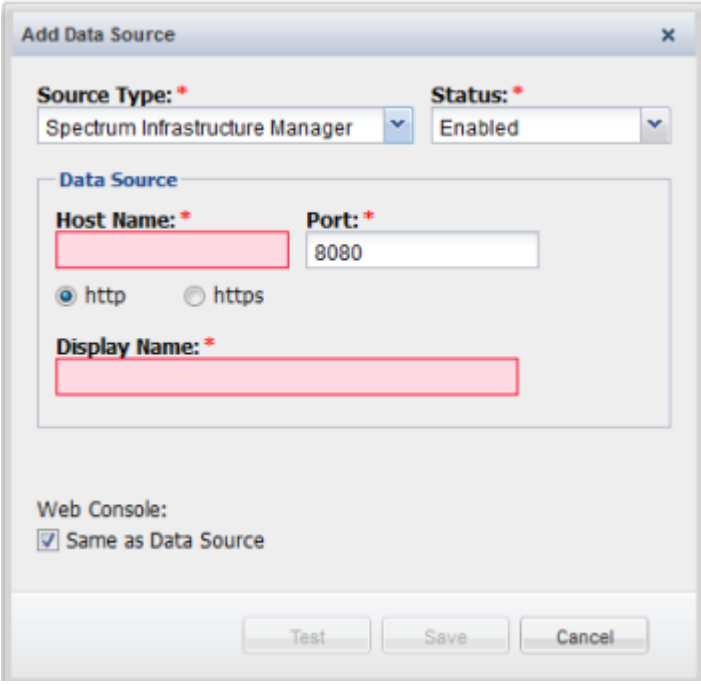
> **NOTE**
> After the integration, if you cannot view the DX NetOps Performance Management events in DX NetOps Spectrum, check whether any firewall settings block the communication between DX NetOps Performance Management and DX NetOps Spectrum OneClick server.

Configuring as a Data Source in DX NetOps Performance Management

Add DX NetOps Spectrum as a data source in DX NetOps Performance Management so that these components can share information.

Follow these steps:

1. Launch the **DX NetOps Performance Management** console and click **Admin, Data Sources**.
   The **Manage Data Sources** page opens.
2. Click **Add**.
   The **Add Data Source** dialog opens.
3. Select '**Spectrum Infrastructure Manager**' in the **Source Type** field.



4. Complete the following fields:
   – **Status**. Select **Enabled** in the **Status** field.

NOTE
You can select Disabled to disable the data source without deleting it.

- **Host Name**. Provide the IP address or DNS hostname of the OneClick server.
- **Port**. Provide the port number to use when contacting the OneClick server.
- **Protocol**. Select the protocol to use to contact the data source. Select the **https** option if your network uses SSL for communications. Verify that you have configured the system correctly before you select the https **option**.

  NOTE
  Before you select **https**, ensure that you have enabled SSL on your OneClick web server host by configuring the "server.xml" and "axis2.xml" file appropriately. For more information about enabling SSL on OneClick web server host, see OneClick Administrator section.
- **Display Name**. Provide a name for the data source. By default, the data source type and the hostname are combined to create the display name.
- **Same as Data Source**. Select the checkbox if the Web Console is on the OneClick server.
  Or clear the Same as Data Source checkbox if the Web Console is on a different server. Then complete the following fields:
  - **Host Name**. Provide the IP address or DNS hostname of the Web Console server.
  - **Port**. Provide the port number to use when contacting the Web Console server.
  - **Protocol**. Select the protocol to use to contact the Web Console server: http or https.

5. Click **Test** to verify that DX NetOps Performance Management can contact the OneClick server and the Web Server.
6. Click **Save.**
   You have added DX NetOps Spectrum as a data source and the synchronization process is initiated.

NOTE

When you delete a data source, the deletion does not remove everything from the system. If you register the same data source, unexpected behavior might occur. To remove a data source temporarily, edit the data source, and set the status to disabled. The changes will be applied to the system only after the next synchronization.

Enable Event Polling

Enable event polling that takes place between the SpectroSERVER and DX NetOps Performance Management. You can specify how often DX NetOps Spectrum queries the DX NetOps Performance Management Event Manager component for events. Perform this step in the OneClick Administration pages.

Follow these steps:

1. Click Administration on the OneClick home page.
   The Administration Pages open.
2. Click Performance Center Integration Configuration in the left panel.
   The Performance Center Integration Configuration page opens.
3. Enter the desired polling interval, in seconds, in the Event Polling Interval field.
   The default value is 60 seconds. Enter a value greater than or equal to 30 seconds.
4. Select the Enabled option in the Event Polling field.
5. Click Save.
   DX NetOps Spectrum - DX NetOps Performance Management integration is now enabled. The event polling settings take effect on the next polling cycle.

Enable Device Monitoring with CA Infrastructure Management

The integration between DX NetOps Spectrum and DX NetOps Performance Management lets a Data Aggregator monitor DX NetOps Spectrum devices. This configuration is optional and requires some additional configuration. We recommend performing these optional steps before you register the DX NetOps Spectrum data source.

To enable CA Infrastructure Management monitoring of the devices and interfaces that DX NetOps Spectrum discovers, take the following steps:

1. Create SNMP profiles in DX NetOps Performance Management for the SNMP-capable devices that are modeled in DX NetOps Spectrum.
2. Create IP domains in DX NetOps Performance Management. For more information, see the online Help for DX NetOps Performance Management.
   When database synchronization occurs, all IP domains are sent to DX NetOps Spectrum, where they appear in OneClick as special Global Collections, with a unique icon.
3. Install and assign a Data Collector for each IP domain.
4. Enable the CA Infrastructure Management Data Collector to discover DX NetOps Spectrum devices.
   Synchronized discovery requires less administration and enables your SpectroSERVERs to provide data to the Data Collectors that you can position where they are required.
5. Add the models to the appropriate IP Domains in OneClick.
   During synchronization, all DX NetOps Spectrum models that are associated with the DX NetOps Performance Management IP Domains are passed to the CA Infrastructure Management Data Aggregator. The associated DX NetOps Spectrum devices are discovered for CA Infrastructure Management Data Aggregator monitoring.
6. Create custom monitoring profiles for DX NetOps Spectrum devices and apply them to collections in the Data Aggregator administration pages.
   Monitoring profiles for a limited set of metrics are applied to the device items as Data Aggregator discovers them. These profiles determine how devices are polled for performance data. For more information, see the online Help for CA Infrastructure Management Data Aggregator.

As DX NetOps Spectrum adds devices to DX NetOps Performance Management, they are added to discovery profiles. Each discovery profile is automatically configured to run with a daily schedule. You can also run them manually or adjust the schedule as required to keep the Data Aggregator up-to-date with new devices from DX NetOps Spectrum.

If SNMP throttling is configured on DX NetOps Spectrum, it does not apply to ongoing polling by Data Aggregator. This feature protects critical devices from failing in case too many polling flows are configured. The throttling mechanism applies to any monitoring or discovery activities. Therefore, if you have configured DX NetOps Spectrum to throttle SNMP requests for a given device, apply the same setting in Data Aggregator.

Add an SNMP Profile to Gather Performance Data

To supply the information that is required to enable SNMP polling for performance metrics, create SNMP profiles in DX NetOps Performance Management. Global administrators and tenant administrators can create SNMP profiles to let DX NetOps Performance Management data sources query devices for performance data. You can create these profiles for SNMPv1/v2c or for SNMPv3.

Device models are contributed to DX NetOps Performance Management with specific IP Domains. The IP domain may be part of the default tenant or a user-created tenant. When you create SNMP Profiles to discover devices that DX NetOps Spectrum contributes, verify that the SNMP Profiles are created in the appropriate tenant space

Follow these steps:

1. Log in to DX NetOps Performance Management as a global administrator or tenant administrator.
2. (Optional) If you have logged in as a global administrator, administer a selected tenant.
3. Select Admin, SNMP Profiles in the menu bar.
   The Manage SNMP Profiles page displays the current list of SNMP profiles.
4. Click New.
   The Add SNMP Profile dialog opens.

5. Complete the following fields and change any default settings as needed. Some fields apply only to SNMPv3.
   - **Profile Name**
     Defines a name for the SNMP profile. Profile names must be unique, cannot be duplicated across SNMP versions, and are not case-sensitive.
   - **SNMP Version**
     Specifies the version of SNMP that the profile uses. Because SNMPv1 and SNMPv2C are similar from a security standpoint, they share a single option. SNMPv3 is a separate option.
   - **Port**
     Identifies the port that is used to make SNMP connections to devices associated with this profile.

     NOTE
     Optional parameter for SNMPv1/v2C.

     Default: 161.
   - **User Name**
     (SNMPv3 Only) Identifies the user for the profile, whose secret keys were used potentially to authenticate and encrypt the SNMPv3 packets. The User Name is a character string.
   - **Context Name**
     (SNMPv3 Only) Identifies the collection of management information that is accessible by an SNMP entity. An octet string that is necessary for providing end-to-end identification and for retrieving data from an SNMPv3 agent.
   - **Community Name**
     (SNMPv1/v2C Only) Defines a secure string that lets the data source query the MIB of the associated device. The community that you supply must provide read-only access to the device MIB.

     NOTE
     In the default SNMP profile, the community is 'public'.
   - **Verify Community Name**
     Confirms the secure community string (name).
   - **Authentication Protocol**
     (SNMPv3 Only) Specifies the authentication protocol to use when contacting devices associated with this profile. The following algorithms for authenticating SNMPv3 packets are supported:
     - None (do not attempt authentication)
     - MD5 (Message Digest 5)
     - SHA (Secure Hash Algorithm)
   - **Authentication Password**(SNMPv3 Only) Specifies the password for authentication using SNMPv3 and the selected authentication protocol.

     NOTE
     Supply an authentication password that contains at least eight characters. Some data sources do not support authentication passwords or privacy passwords that fall below this minimum length. They treat the SNMP profile as invalid, and some data is not collected. Blank passwords are not supported for SNMPv3 profiles with MD5 or SHA as the Authentication Protocol.
   - **Verify Authentication Password**Confirms the authentication password.
   - **Privacy Protocol**
     (Optional) Specifies the encryption protocol to use for data flows sent to any devices or servers associated with this profile, as follows:
     - None (do not encrypt communications)
     - DES
     - AES 128
     - Triple DES

     NOTE
     The privacy protocol option is not enabled until authentication is enabled for this profile.

- **Privacy Password**
  Defines the password used when exchanging encryption keys. See the Note for a possible length requirement.
- **Verify Privacy Password**
  Defines the password used when exchanging encryption keys.
- **Use by default for new devices**
  Specifies whether the information in this profile is used by default. DX NetOps Performance Management uses this information to contact any new items that are discovered from monitored traffic. If it fails, the next profile in priority order is used. Disable this parameter to exclude a profile from discovery

6. Click Save.
   The Manage SNMP Profiles page opens. The new profile appears in the list.
   DX NetOps Performance Management automatically performs a global synchronization to send the profile information to all registered data sources.

Add Device Models to DX NetOps Performance Management IP Domain Models

After database synchronization occurs, the IP domains that you created in DX NetOps Performance Management are displayed as DX NetOps Performance Management IP Domains in OneClick. The device models that you add to these IP Domains are associated with IP domain definitions in DX NetOps Performance Management, polled by Data Aggregator, and included in dashboards.

You can also apply Search criteria to an IP Domain or Global Collection so that its membership is updated dynamically.

NOTE
Although DX NetOps Performance Management IP Domain models are not Global Collections, they share many properties. Consequently, the configuration of a DX NetOps Performance Management IP Domain uses many of the common Global Collection dialogs. But these IP Domain models are designated with a special icon in OneClick.

NOTE
DX NetOps Spectrum devices can only be members of a single IPDomain model type. If you attempt to add a model to multiple IP domains, you see an error message.

Follow these steps:

1. In any topology, take *one* of the following steps to select a device model to add to an IP Domain:
   - **Single model selection:** In the Navigation panel, right-click a modeled element and select Add To, Global Collection(s).
     The Select Global Collections dialog opens. DX NetOps Performance Management IP Domain models appear in the list among your Global Collections.

     NOTE
     Or you can right-click a single model in a topology view and select Add To, Global Collection(s).
   - **Multiple model selection:** To multi-select models in a topology view, take the following steps:
   a. Press and hold the SHIFT key and individually select the modeled elements.
   b. While the SHIFT key is pressed, right-click the last selected modeled element and select Add To, Global Collection(s).
2. Select the name of the IP Domain model where you want to add the models.
   IP Domains are designated with a special icon:

3. Click OK.
   If Data Aggregator has been configured to discover devices from other data sources, the devices from DX NetOps Spectrum are discovered by Data Aggregator at the next synchronization. The devices are added to DX NetOps Performance Management with an IP domain association that mirrors the DX NetOps Performance Management IP Domain membership in DX NetOps Spectrum.

Verify that SNMP profiles for monitored devices are available in DX NetOps Performance Management and create them if necessary.

Update IP Domain Membership Dynamically

In addition to selecting individual models for DX NetOps Performance Management IP Domain models, you can populate IP Domains with dynamic members. Adding dynamic members to the models that represent DX NetOps Performance Management IP domains lets you precisely populate IP domains with devices that belong to them.

Dynamic membership is based on rules and on search criteria that you specify. Dynamic members of a DX NetOps Performance Management IP Domain only remain in the DX NetOps Performance Management IP Domain as long as they meet the specified search criteria. Changes are automatically synchronized with DX NetOps Performance Management.

Follow these steps:

- a. In the Explorer tab of the Navigation panel, navigate to the Global Collections node, and locate the DX NetOps Performance Management IP Domains.
  b. Right-click a DX NetOps Performance Management IP Domain, and select Edit Global Collection.
  IP Domains are designated with a special icon:
  
  The Edit Global Collection dialog opens.
  c. Click 'Search Options'.
  d. Complete any of the following fields to create a single search expression:
    - **Attribute**
      Specifies the attribute of a device to filter. From the drop-down list of commonly used attributes, select the attribute that you want to use. The predefined list might not include the attribute that you want. In this case, click Attribute to specify the model type (device, port, or other) and its associated attribute that you want to find.

      NOTE
      If you choose an alphabetic attribute value, you can either clear (ignore) or select (include) the Ignore Case checkbox.
    - **Comparison Type**
      Specifies the type of comparison to be made against the value that is specified in the Attribute field. Only the comparison types appropriate to the attribute data type are available.
    - **Ignore Case**
      Determines whether the comparison is case-sensitive. If you do not select this checkbox, the comparison is case-sensitive. This selection is only enabled when it is appropriate for the data type of the attribute you selected.
    - **Attribute Value**
      Enter the desired attribute value to search.
    - **Devices Only**
      Specifies that the search results list includes only devices.
  e. (Optional) To use a wildcard character or regular expression in the Attribute Value field, select a valid attribute in the Attribute field. Select 'Matches Pattern' in the Comparison Type field. Then select one of the following options:
    - **Specify Wildcard Now**
      Lets you search for a value using a wildcard. For more information about the available wildcards, see the Modeling and Managing Your IT Infrastructure section.
    - **Specify RegExp Now**
      Lets you create a search using Perl Compatible Regular Expression (PCRE) matching on attributes of the type 'text string'. Text string searches are available only for Matches Pattern comparison types. PCRE matching helps you to find and group models using specific pattern searches that are more advanced than existing searches or wildcard searches.

f. (Optional) Click the Show Advanced button to create a search that is based on a compound clause. For example, you can choose to populate the DX NetOps Performance Management IP Domain based on existing Global Collections, or Secure Domain Connector information. For more information, see the Modeling and Managing Your IT Infrastructure section.
g. (Optional) Select the Real-Time Update checkbox.
This option disables the update interval. It also adds or removes models from the DX NetOps Performance Management IP Domain when they meet or no longer meet the search criteria.
h. (Optional) Supply a value in the 'Run search to update Global Collection membership every <> hours' field.
This value determines how often OneClick conducts a search to update the dynamic membership of the DX NetOps Performance Management IP Domain.
i. Click OK.
The Search Options dialog closes and the Global Collection dialog opens.
j. Click Landscapes to identify the landscapes to include when searching models to populate the DX NetOps Performance Management IP Domain.
k. Click OK.
The DX NetOps Performance Management IP Domain model now has a dynamic membership. Any adjustments that are automatically applied to the membership of this DX NetOps Performance Management IP Domain are synchronized to DX NetOps Performance Management.

Enable Synchronized Discovery

When you enable the Data Aggregator component to discover DX NetOps Spectrum devices, you enable a large set of functionality. The integration can enhance DX NetOps Spectrum fault and availability monitoring with device performance alerts that are based on the analysis of historical data. You can also drill down into device performance data in context from OneClick. And you can monitor the same devices with two different infrastructure management systems without additional discovery administration.

Follow these steps:

1. Launch the DX NetOps Performance Management console and click Admin, Data Sources.
The Manage Data Sources page opens.
2. Select the Data Aggregator data source in the list, and click Edit.
The Edit Data Source dialog opens.
3. Select the option to Discover devices from other data sources.
4. Click Save.
Enabling the Data Aggregator component to discover devices from other data sources initiates a synchronization of the CA Infrastructure Management device inventory with DX NetOps Spectrum. If CA Infrastructure Management determines that a device known to DX NetOps Spectrum is not in the CA Infrastructure Management inventory, the IP address of the device is added to an automatically created discovery profile. This discovery profile is configured to run with a daily schedule by default. You can run the discovery manually or adjust the schedule to meet your requirements. For more information, see the CA Infrastructure Management Data Aggregator online Help.

Smart Interface Filtering

DX NetOps Spectrum can contribute interfaces to DX NetOps Performance Management that does not appear in the DX NetOps Performance Management Inventory unless they are monitored by another data source, such as Data Aggregator or Network Flow Analysis. This behavior results from the Data Aggregator interface filtering feature.

DX NetOps Spectrum does not collect performance data from interfaces. As a result, an interface that DX NetOps Spectrum discovers and sends to Data Aggregator does not have any visible data in DX NetOps Performance Management dashboards unless another data source is also monitoring that interface.

If you right-click an interface model in OneClick, the option to drill down into the matching data context in DX NetOps Performance Management is not available to you if that interface is filtered out of the corresponding monitoring profile in CA Infrastructure Management.

Maintaining the Integration

Modifying Data Sources After Integration

WARNING
The following guidelines must be followed to enable data synchronization between DX NetOps Spectrum and DX NetOps Performance Management.

Use the following guidelines when restoring a data source in DX NetOps Performance Management after DX NetOps Spectrum-DX NetOps Performance Management integration has already been configured:

- If DX NetOps Spectrum exists as a data source in DX NetOps Performance Management when you restore another data source, restart tomcat.
- If DX NetOps Spectrum does not exist as a data source in DX NetOps Performance Management, add other data sources first and then add DX NetOps Spectrum as a data source.

NOTE

For more information refer to Manage Data Sources and Synchronize Data Sources in DX NetOps Performance Management documentation.

Restoring the SpectroSERVER Database

To restore the SpectroSERVER database to a previous state after DX NetOps Spectrum - DX NetOps Performance Management integration, complete the following steps.

WARNING
The correct procedure must be followed to enable data to synchronize correctly between DX NetOps Spectrum and DX NetOps Performance Management.

Follow these steps:

1. Remove DX NetOps Spectrum as a data source in DX NetOps Performance Management.
2. Restore the SpectroSERVER database.

   NOTE
   For information about restoring the SpectroSERVER database, see Database Management .
3. Drop the netqos_integ database on the OneClick server

   ```
   SPECROOT/mysql/bin/mysql --defaults-file=$SPECROOT/mysql/my-spectrum.cnf -uroot -
   proot -e "drop database netqos_integ;"
   ```
4. Restart the OneClick server.
5. Add DX NetOps Spectrum as a data source in DX NetOps Performance Management.

Remove DX NetOps Spectrum as a Data Source in DX NetOps Performance Management

When restoring the SpectroSERVER database to a previous state, unregister the DX NetOps Spectrum data source. This process helps to establish correct synchronization between DX NetOps Spectrum and DX NetOps Performance Management after the database has been restored.

Deleting selected data sources from DX NetOps Performance Management can have negative consequences. Only administrators with the Delete Data Sources role right can delete a data source from DX NetOps Performance Management. This role right is not granted by default and must be assigned to the role as a separate step.

Follow these steps:

1. Log in as a user with the Administrator role.
2. Navigate to the Manage Roles page.
   The page displays the current list of roles.
3. Select the Administrator role, and click Edit. The role right to Delete Data Sources is only available to this predefined role.
   The Edit Role Rights dialog opens.
4. Select Performance Center, and click Edit.
   The Edit Role Rights dialog lets you select individual access rights for this role.
   Assigned role rights are grayed out because they are read-only for this role.
5. Select Delete Data Sources. Click the right arrow to move it from the Available Rights list to the Selected Rights list.
6. Click OK. Then click Save to save your change to the role.

   NOTE
   You must be logged in with a user account that has the Administrator role that you have just edited.
7. Click Admin, Data Source Settings, and select Data Sources.
   The Manage Data Sources page opens.
8. Select the DX NetOps Spectrum data source, and click Delete.
   The Delete Data Source page opens.
9. Click Delete, and then click Yes to confirm the deletion.
   The data source is successfully deleted, and synchronization between DX NetOps Spectrum and DX NetOps Performance Management no longer occurs.
   DX NetOps Performance Management IP Domain models in OneClick are automatically deleted.

Enabling Debug Logging

To facilitate the investigation of issues, enable DX NetOps Performance Management integration debug logging on the OneClick Web Server.

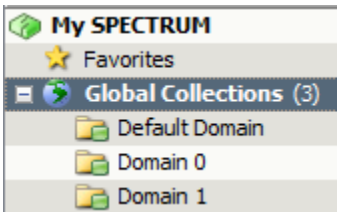
Follow these steps:

1. Navigate to the OneClick home page.
2. Click Administration, Debugging, Web Server Debug Page (Runtime) on the OneClick home page.
3. Select ON for the 'Performance Center Integration' option.
4. Select ON for the 'Performance Center Integration Sync' option.

   NOTE
   Enabling DX NetOps Performance Management integration debug logging can generate a large volume of data in a short amount of time.

Support for DX NetOps Performance Management IP Domains

When you register the DX NetOps Spectrum data source in DX NetOps Performance Management, database synchronization occurs. DX NetOps Spectrum retrieves a list of IP domains from DX NetOps Performance Management. All IP domain definitions are sent over, regardless of their association with individual tenants. OneClick displays these DX NetOps Performance Management IP Domain models in the same area as DX NetOps Spectrum Global Collections in the OneClick Navigation panel. The DX NetOps Performance Management IP Domain models have the same names as the DX NetOps Performance Management IP domain definitions:



Use these IP domains to determine which models are synchronized with CA Infrastructure Management. To include a device model in CA Infrastructure Management monitoring and make it available in DX NetOps Performance Management dashboards, add it to an IP domain in OneClick.

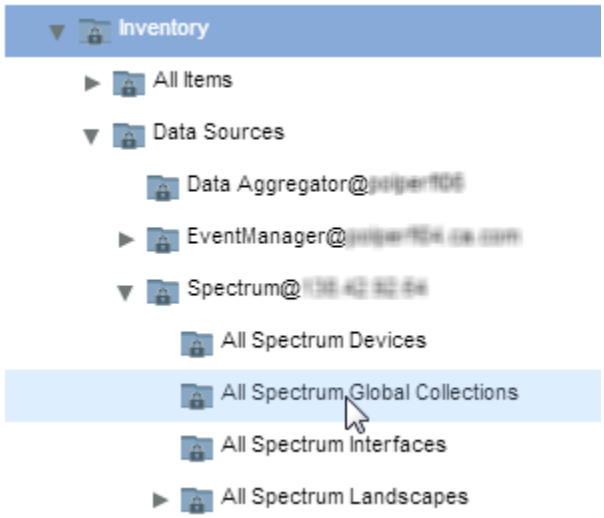
Take care to add only device models that should be synchronized with DX NetOps Performance Management. When the device models are synchronized, they are associated with the corresponding IP domain in DX NetOps Performance Management. The DX NetOps Performance Management IP domain may belong to the Default Tenant, or to any custom tenant. Do not add interface models. Device interfaces are automatically added to the IP domain with which their device is associated.

DX NetOps Spectrum devices can only be members of a single IPDomain model type. If you attempt to add a model to multiple IP domains, you see an error message.

Group Synchronization

If you already have DX NetOps Spectrum Global Collections that you want to continue to use, they are synchronized with DX NetOps Performance Management. All managed items in your Global Collections become members of groups, which are displayed in the DX NetOps Performance Management Groups tree.

NOTE
From 10.4.1, the LocalID for Landscape is sent as Landscape-Handle instead of Landscape Name.



You can also add those managed items to DX NetOps Performance Management Service Provider groups, which enable tenant users to manage them and view their data.

Enable Tenant Access to Data

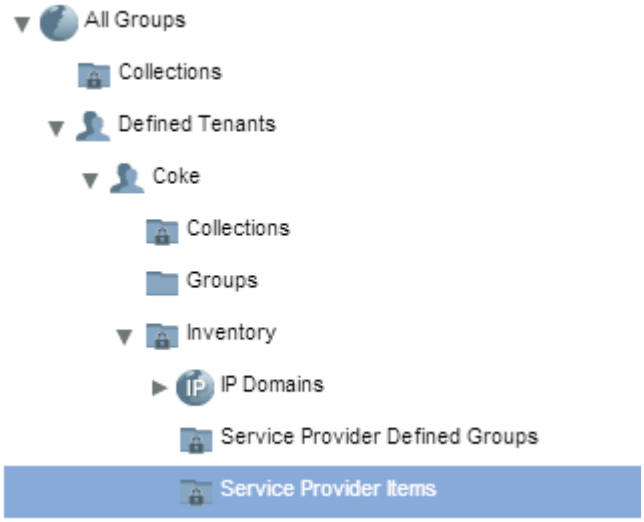
To monitor devices with both DX NetOps Spectrum and CA Infrastructure Management, you must explicitly add models to DX NetOps Performance Management IP Domains. At the first synchronization, DX NetOps Spectrum creates DX NetOps Performance Management IPDomain models in OneClick from all IP domains in DX NetOps Performance Management,

including the Default Domain. Device models within DX NetOps Spectrum can then be added to DX NetOps Performance Management IP Domains, enabling them to be synchronized with DX NetOps Performance Management. Those DX NetOps Spectrum devices are always associated with the same IP domain in CA Infrastructure Management and are also associated with the tenant that owns the IP domain.

If you choose to associate DX NetOps Spectrum device models with IP domains in the Default Tenant space only, you can leverage DX NetOps Performance Management Service Provider groups. These groups enable specific tenants to access the devices in DX NetOps Performance Management. You can thus grant access to device data to other tenant users. Set up Service Provider groups for tenant users so that they can monitor devices and components that are associated with the Default Domain.

Follow these steps:

1. In OneClick, add the device models that you want additional tenant users to monitor to the Default Domain Global Collection.
2. Log in to DX NetOps Performance Management as a user with the predefined Administrator role.
3. Initiate a manual synchronization of the DX NetOps Spectrum data source.
   For more information, see theDX NetOps Performance Management documentation.
4. Select Admin, Groups in the menu bar.
   The Manage Groups page displays current groups in a tree structure.
5. Expand the Defined Tenants group in the Groups tree.
6. Locate the tenant to which you want to grant access to selected DX NetOps Spectrum devices.
7. Expand the Inventory group under the tenant group.
8. Select the Service Provider Items group.



9. Click the Items tab in the right pane, and click Add Item Type.
   The Add Items dialog opens. You can begin adding the items to the Service Provider Items group.
10. Add all DX NetOps Spectrum models that you want the users who are associated with this tenant to be able to monitor.
11. Click Close when you have finished adding items.

Now the tenant users can see items that are being managed in the Default Tenant space in their inventory. Tenant users can also add these items to tenant groups to organize reporting.

Drill Down into DX NetOps Performance Management Performance Data

You can navigate to DX NetOps Performance Management performance data directly from models in OneClick. Drill down from any device or interface model that is also available in DX NetOps Performance Management.

Although DX NetOps Spectrum can contribute an interface model to DX NetOps Performance Management, an interface item only appears in DX NetOps Performance Management when a data source other than DX NetOps Spectrum is contributing performance data for it. If the interface is not being monitored by another data source, such as Data Aggregator or CA Network Flow Analysis, the right-click drill-down option is not available. For more information, see Smart Interface Filtering.

Follow these steps:

1. In the OneClick Navigation panel, expand the DX NetOps Performance Management IP Domains that were synchronized from CA Infrastructure Management.
2. Select a model in one of these collections.
3. Right-click the model, and select the option to navigate to DX NetOps Performance Management.

> **NOTE**
> This option is also available when you select the model on a Topology map or from Locater search results.The DX NetOps Performance Management user interface opens in a separate window. The device context associated with the selected model is preselected.

Known Anomalies - DX NetOps Performance Management Integration

DX NetOps Spectrum - DX NetOps Performance Management integration has the following known anomalies:

- Event processing behavior is undefined when a device is modeled as a nonproxy model on more than one landscape.
- Some data sources do not support IPv6 addresses. Proper mapping does not occur when DX NetOps Spectrum uses an IPv6 primary address for a device but the reporting data source does not support IPv6 addresses. To correct the mapping, destroy the DX NetOps Spectrum model and rediscover it using an IPv4 address. After the next incremental synchronization, the model will be mapped properly.
- When a DX NetOps Performance Management event is received for a model that is in maintenance mode in DX NetOps Spectrum, the event is not processed, in accordance with expected maintenance mode behavior.
- DX NetOps Spectrum can fall out of synchronization when the Event Manager data source is removed. If this situation occurs, follow these steps:
  a. Remove the DX NetOps Spectrum data source.
  b. Register the Event Manager data source again.
  c. Register the DX NetOps Spectrum data source again.

> **WARNING**
> This situation only applies to CA NetQoS Performance Center v6.1. It does not apply to DX NetOps Performance Management. Do not delete the data source from DX NetOps Performance Management unless product support advises you to do so.

How to Configure Events for Integration with DX NetOps Performance Management

With the integration of r9.2.2 and later and CA Performance Management v2.0.00 and later, events for which DX NetOps Spectrum polls are specified in an XML file. Based on the contents of the default XML file, DX NetOps Spectrum polls for ThresholdViolation events automatically. If you do not modify the XML file, information is obtained from the Event Manager database for ThesholdViolation events only.

You can also configure DX NetOps Spectrum to poll for any event in the Event Manager database. To do so, modify the XML file and set up other event support files in DX NetOps Spectrum. In addition, for DX NetOps Spectrum to process a modified event, the device or port must be modeled in DX NetOps Spectrum and included in the synchronization process.

To configure DX NetOps Spectrum to poll for specific events, take the steps that are listed here. A complete example is provided.

NOTE
To poll for ThresholdViolation events only, no action is required.

1. Obtain a developer ID to create event codes.
2. Update the netqos-integration-application-config.xml file to specify additional events and alarms.
3. Update the event disposition file to map the events to DX NetOps Spectrum event files.
4. Create an event format file for each event.
5. Create a probable cause file for each alarm code.
6. Deploy the changes by restarting the SpectroSERVER and OneClick servers.

Obtain a Developer ID

When defining events for the DX NetOps Spectrum - DX NetOps Performance Management integration, you use identifying event codes. The first 2 bytes of any event code contains a developer ID. You can obtain a registered developer ID from CA so that you can specify unique codes for your events. Using a unique developer ID lets you easily recognize your new codes in OneClick and prevents potential conflicts with other DX NetOps Spectrum event codes.

To obtain a developer ID from CA, contact CA Technical Support.

Update the netqos-integration-application-config.xml File

DX NetOps Spectrum uses the netqos-integration-application-config.xml file to determine the events for which to poll. DX NetOps Spectrum polls for ThresholdViolation events by default. To poll for more events, modify the netqos-integration-application-config.xml to define the event codes and associated alarms for each event.

The netqos-integration-application-config.xml file is located in the following directory:

```
$SPECROOT\tomcat\webapps\spectrum\WEB-INF\netqos\
config\container
```

Define Events

The eventTypeManager bean defines the events for which DX NetOps Spectrum polls. The entries for ThresholdViolation events appear in the file by default. You can manually add more events.

```
<bean id="eventTypeManager"
   class="com.ca.im.netqos.integration.event.type.EventTypeManager">
   <property name="interestingEventTypes">
     <map>
       <entry key="ThresholdViolation" value-ref="thresholdViolationAlarmCodes" />
       <entry key="TestEvent" value-ref="TestEventAlarmCodes" />
     </map>
   </property>
   <property name="alarmClearCodes">
     <map>
       <entry key="ThresholdViolation" value="0x5c40009" />
       <entry key="TestEvent" value="TestEventAlarmClearCode" />
     </map>
   </property>
```

```
</bean>
```

Update the following property elements to add events that DX NetOps Spectrum can include in polling:

- **interestingEventTypes**
  Specifies the types of events to include in polling. Each entry element identifies a specific event type and an alarm code map value. The ThresholdViolation entry is included by default. Add an entry element, as follows:
  ```
  <entry key="TestEvent" value-ref="TestEventAlarmCodes" />
  ```
  – ***TestEvent***
    Specifies the name of an event in the Event Manager database.
  – ***TestEventAlarmCodes***
    Specifies the value of the map that identifies the alarms for this event.

    NOTE
    The alarm code map is described in the next section.
- **alarmClearCodes**
  Specifies the alarm clear codes for polled events. The default alarm clear code for the ThresholdViolation event is 0x5c40009. For each event, add an entry element, as follows:
  ```
  <entry key="TestEvent" value="TestEventAlarmClearCode" />
  ```
  – ***TestEvent***
    Specifies the name of the event that was added for polling.
  – ***TestEventAlarmClearCode***
    Specifies the alarm clear code for the event.

Define Alarms

An alarm map defines the alarm code values that are associated with a particular event. For each polled event (or, each interestingEventTypes entry), a corresponding alarm map must be defined. The alarm map for the ThresholdViolation event appears in the file by default, and an alarm map for each custom event must be added manually.

```
<bean id="thresholdViolationAlarmCodes"
    class="org.springframework.beans.factory.config.MapFactoryBean">
    <property name="sourceMap">
      <map>
        <entry key="1" value="0x5c40010" />
        <entry key="2" value="0x5c40011" />
        <entry key="3" value="0x5c40012" />
      </map>
    </property>
</bean>
<bean id="testEventAlarmCodes"
    class="org.springframework.beans.factory.config.MapFactoryBean">
    <property name="sourceMap">
      <map>
        <entry key="alarmSev1" value="alarmCode1" />
        <entry key="alarmSev2" value="alarmCode2" />
        <entry key="alarmSev3" value="alarmCode3" />
      </map>
    </property>
</bean>
```

To add alarm maps for custom events, add a bean element for each event and update the following values:

- ***testEventAlarmCodes***

Specifies the alarm code map value for a particular event. This value is established on the interestingEventTypes entry and must match that value.

- ***alarmSev1 - alarmCode1, alarmSev2 - alarmCode2, alarmSev3 - alarmCode3***
  Specifies the *alarmSeverity* - *alarmCode* pairs for a particular event. For example, for the default ThresholdViolation event, the Minor (1), Major (2), and Critical (3) alarm codes are 0x5c40010, 0x5c40011, and 0x5c40012, respectively.

Update the Event Disposition File

The Event Disposition (EventDisp) file is used to determine how to process the events configured in the netqos-integration-application-config.xml file. Each event entry maps an event to a DX NetOps Spectrum event file.

The EventDisp file for DX NetOps Spectrum-DX NetOps Performance Management integration is located in:

```
<$SPECROOT>\SS\CsVendor\netqos
```

For the default ThresholdViolation event, the following entries map the alarm codes to individual DX NetOps Spectrum event files:

```
#PC Threshold
0x5c40010 E 50 A 1,0x5c40010,107
0x5c40011 E 50 A 2,0x5c40011,107
0x5c40012 E 50 A 3,0x5c40012,107
0x5c40009 E 50 C 0x5c40010,107 C 0x5c40011,107 C 0x5c40012,107
```

For each custom event, add new event map entries to the file. The following example shows syntax that generates or clears alarms that are based on the event code.

```
#New Event
alarmCode1E 50 A 1, alarmCode1_filename,107
alarmCode2E 50 A 2, alarmCode2_filename,107
alarmCode3E 50 A 3, alarmCode3_filename,107
alarmClearCode4E 50 C alarmCode1,107 C alarmCode2,107 C alarmCode3,107
```

For more information about using Event Disposition files, including syntax and examples, see Event Configuration.

Create Event Format Files

An event format file contains the message about the event that is displayed to users on the Events tab in OneClick. Each new event that is defined in the netqos-integration-application-config.xml file requires an event format file. The file enables the event to display correctly in the OneClick Events view.

The file name must match the alarm code (for example, alarm code 0x5c40010 uses the file "Event05c40010"). And the file must exist in the following directory:

```
<$SPECROOT>\SG-Support\CsEvFormat
```

The following is an example of the file format:

```
{d "%w- %d %m-, %Y - %T"} - {S 109} is reporting a minor threshold violation.
Detail of Threshold Violation:
        1) Incident Start Time: {D 111}
        2) Event ID: {S 107}
        3) Event Source: {S 113}
        4) Alert Message: {S 76620}
A corresponding minor Threshold Violation Alarm will be generated.
(event [{e}])
```

For more information, see Event Configuration .

Create Probable Cause Files

A probable cause file defines the symptoms, probable causes, and recommended corrective actions for an alarm. Each new alarm code requires a probable cause file so that the alarm displays correctly in the OneClick Alarms view.

The file name must match the alarm code (for example, alarm code 0x5c40010 uses the file "Prob05c40010"). And the file must exist in the following directory:

```
<$SPECROOT>\SG-Support\CsPCause
```

The following is an example of the file format:

```
A minor threshold violation has occurred.
SYMPTOMS:
The monitored threshold has been exceeded.
PROBABLE CAUSES:
RECOMMENDED ACTIONS:
Launch the "Performance View" to see incident details.
```

For more information, including syntax, see Event Configuration .

Deploy the Changes

After all configuration changes have been made, restart the SpectroSERVER and OneClick servers.

Event polling now reflects any changes that you have made.

Example

This example shows how to configure DX NetOps Spectrum to poll for a specific event in the Event Manager database. The event in this example identifies when a router device experiences high memory usage.

1. Identify a device or port for which you want DX NetOps Spectrum to poll for in the Event Manager database. If the device or port is not modeled in DX NetOps Spectrum, model the element. For example, to monitor specific events for a particular router, the router must be modeled in the DX NetOps Spectrum database.
2. Obtain a developer ID from CA Technical Support for use with DX NetOps Spectrum-DX NetOps Performance Management integration. This example uses the default developer ID value, 0xffff.
3. Identify the events for which DX NetOps Spectrum polls. For example, you can identify any occurrence when the router device experiences high memory utilization. This example refers to this event as "RouterHighMemory".
4. Define the event by modifying the XML file:
   a. Open the following file for editing:
      ```
      <$SPECROOT>\tomcat\webapps\spectrum\WEB-INF\netqos\config\container\netqos-integration-application-config.xml
      ```
   b. Define the custom event. Update the existing eventTypeManager element as follows: Add the RouterHighMemory event to the list of events for which to poll. Establish an alarm map value. Specify a default alarm clear code. The following code shows these changes. Notice that the alarm clear code uses a developer ID.
      ```
      <bean id="eventTypeManager"
         class="com.ca.im.netqos.integration.event.type.EventTypeManager">
         <property name="interestingEventTypes">
           <map>
             <entry key="ThresholdViolation" value-ref="thresholdViolationAlarmCodes" />
             <entry key="RouterHighMemory" value-ref="RouterHighMemoryAlarmCodes" />
           </map>
         </property>
         <property name="alarmClearCodes">
           <map>
      ```

```
        <entry key="ThresholdViolation" value="0x5c40009" />
        <entry key="RouterHighMemory" value="0xffff0004" />
      </map>
    </property>
  </bean>
```

c. Define the alarm map by adding the following new bean element:

```
<bean id="RouterHighMemoryAlarmCodes"
    class="org.springframework.beans.factory.config.MapFactoryBean">
    <property name="sourceMap">
      <map>
        <entry key="1" value="0xffff0001" />
        <entry key="2" value="0xffff0002" />
        <entry key="3" value="0xffff0003" />
      </map>
    </property>
</bean>
```

d. Save and close the file.

5. Specify how DX NetOps Spectrum processes the encountered event by updating the Event Disposition file:
   a. Open the following file for editing:

```
<$SPECROOT>\SS\CsVendor\netqos\EventDisp
```

   b. Add the following map entries for the RouterHighMemory event:

```
#RouterHighMemory Event
0xffff0001E 50 A 1, 0xffff0001,107
0xffff0002E 50 A 2, 0xffff0002,107
0xffff0003E 50 A 3, 0xffff0003,107
0xffff0004E 50 C 0xffff0001,107 C 0xffff0002,107 C 0xffff0003,107
```

   c. Save and close the file.
6. Create an event format file for each of the alarm codes using the following naming convention (*AlarmCode* - *EventFormatFile*):
   – 0xffff0001 - Eventffff0001
   – 0xffff0002 - Eventffff0002
   – 0xffff0003 - Eventffff0003
   – 0xffff0004 - Eventffff0004
   – Create a text file containing content similar to the following text:

```
{d "%w- %d %m-, %Y - %T"} - {S 109} is reporting a minor threshold violation.
Detail of Threshold Violation:
        1) Incident Start Time: {D 111}
        2) Event ID: {S 107}
        3) Event Source: {S 113}
        4) Alert Message: {S 76620}
A corresponding minor Threshold Violation Alarm will be generated.
(event [{e}])
```

     NOTE
     When creating Eventffff0004, use appropriate wording for clearing an alarm.

   – Save the file to the following location:

```
<$SPECROOT>\SG-Support\CsEvFormat
```

   – Repeat steps a and b for each alarm code.
7. Create a probable cause file for each of the alarm codes using the following naming convention (*AlarmCode* - *ProbableCauseFile*):

- 0xffff0001 - Probffff0001
- 0xffff0002 - Probffff0002
- 0xffff0003 - Probffff0003
- 0xffff0004 - Probffff0004
- Create a text file containing content similar to the following text:

```
A minor threshold violation has occurred.
SYMPTOMS:
The monitored threshold has been exceeded.
PROBABLE CAUSES:
RECOMMENDED ACTIONS:
Launch the "Performance View" to see incident details.
```

- Save the file to the following location:

```
<$SPECROOT>\SG-Support\CsPCause
```

- Repeat steps a and b for each alarm code.

8. Restart the SpectroSERVER and OneClick servers.
   When the integration is complete, DX NetOps Spectrum uses the updated files to poll for the RouterHighMemory event, generating events and alarms as specified.

Event Handling in FT Scenario 10.4.2.1

When DX NetOps Spectrum is set up in FT mode, and the primary host fails, DX NetOps Spectrum runs on the secondary host. In such a situation only events are pulled from DX Performance Management to DX NetOps Spectrum. However, inventory sync is not supported.

Troubleshooting DX NetOps Performance Management Integration

After DX NetOps Spectrum Upgrade, Integration Breaks with DX NetOps Performance Management in SSL Environment

Symptom:

DX NetOps Spectrum and DX NetOps Performance Management integration breaks after DX NetOps Spectrum upgrade. This happens when SSL is enabled and DX NetOps Spectrum is connected to DX NetOps Performance Management using **https**.

Solution (for 10.2.2 and previous versions):

Verify whether SSL is enabled on the OneClick web server host in the DX NetOps Spectrum environment. If so, verify that the "axis2.xml" file is updated with HTTPS protocol and appropriate port number.

Follow these steps:

1. Open the "axis2.xml" file in an editor from "$SPECROOT/tomcat/webapps/axis2/WEB-INF/conf".
2. Add the following section in axis2.xml after 'http' transportReceiver:

```
<transportReceiver name="https" class="org.apache.axis2.transport.http.AxisServletListener"> <parameter
 name="port">8443</parameter> </transportReceiver>
```

3. Restart the Webserver for these changes to take affect.

Solution (for 10.2.3)

The "axis2.xml" file contains the following section in commented format. After DX NetOps Spectrum upgrade, you can uncomment this code and update the port number to integrate DX NetOps Spectrum with DX NetOps Performance Management using **https**. You must restart the Webserver for these changes to take affect.

```
<transportReceiver name="https" class="org.apache.axis2.transport.http.AxisServletListener"> <parameter
 name="port">8443</parameter> </transportReceiver>
```

Unable to Add Spectrum Data as Source to DX NetOps Performance Management

Symptom:

I am unable to add DX NetOps Spectrum as a data source in NetOps Portal when I select **https** as the protocol.

Solution:

Verify whether SSL is enabled on the OneClick web server host in the DX NetOps Spectrum environment. If so, verify that the "axis2.xml" file is updated with HTTPS protocol and appropriate port number.

Follow these steps:

1. Open the "axis2.xml" file in an editor from "$SPECROOT/tomcat/webapps/axis2/WEB-INF/conf".
2. Locate the following section in axis2.xml:
```
<transportReceiver name="http"
                      class="org.apache.axis2.transport.http.AxisServletListener">
</transportReceiver>
```
3. Change the section as follows:
```
<transportReceiver name="https"
                      class="org.apache.axis2.transport.http.AxisServletListener">
        <parameter name="port">8443</parameter>
</transportReceiver>
```

Error When Adding Device to IP Domain

Symptom:

I tried to add a device to a NetOps Portal IP domain in OneClick. I received an error message that stated, "The following models could not be added to the Global Collection *Domain Name*." An additional statement claimed that the models did not exist. But I have verified that the models do exist in the landscape.

Solution:

You see this message if you attempt to add a device that is already associated with an existing IP domain to another IP domain. This error can occur if, for example, the device was added to the IP domain either manually or dynamically, using a Global Collection rule. The portion of the error message stating that "The model does not exist" is inaccurate. We plan to address this issue in a future version of the DX NetOps Spectrum software.

Pingable Devices in NetOps Portal Have Little Data

Symptom:

After synchronization, some devices that DX NetOps Spectrum contributed to NetOps Portal appear to have a Subtype of Pingable in the Inventory view. They should be classified as Router or Switch. They are legitimate devices that should be reporting lots of performance data.

Solution:

A *pingable device* refers to a device that does not allow SNMP polling and is therefore contacted using ICMP ping tests for status and reachability statistics. Devices that are sent over from DX NetOps Spectrum can appear to be Pingable and have only status and availability data because of configuration issues. Take the following steps:

- Make sure that NetOps Portal has an SNMP profile with the appropriate credentials to collect SNNP data from the device. For more information, see Add an SNMP Profile to Gather Performance Data.
- Make sure that the correct SNMP profiles are specified in the discovery profile that corresponds to the device.
- Check firewall configuration. If the SNMP and discovery configurations are correct, a network ACL or firewall adjustment can be required to enable the Data Collector to collect SNMP data for the device.

Make sure that the Data Collectors that are assigned to the IP domain of each pingable device have network access to the devices. The level of access must be comparable to that of the SpectroSERVER or Secure Domain Connector that monitors the device in DX NetOps Spectrum.

DX NetOps Performance Management alarms are missing in DX NetOps Spectrum

Symptom:

CAPM alarms are missing in DX NetOps Spectrum due to DX NetOps Performance Management and DX NetOps Spectrum machines are not in time sync.

Solution:

DX NetOps Performance Management and DX NetOps Spectrum machines need to be in time sync, contact the network administrator.

Integration with CA Service Desk

Overview of Functionality

The DX NetOps Spectrum and CA Service Desk Manager integration provides the following features:

- Associates DX NetOps Spectrum alarms with the CA Service Desk tickets in the following ways
  - Creates tickets when manually requested by OneClick operators.
  - Automatically creates tickets for each alarm type.
  - Automatically creates tickets using the DX NetOps Spectrum Alarm Notification Manager (SANM) functionality.
- Maintains the consistency of the following information that is shared between a DX NetOps Spectrum alarm and its associated CA Service Desk ticket:
  - Status of alarms and associated tickets
  - The current assignee (troubleshooter) assigned to tickets
- Provides a link to launch a CA Service Desk Manager view of a particular ticket directly from within the OneClick console.
- Provides an approval system for host configuration change requests that are initiated in Network Configuration Manager.
- Supports multiple CA Service Desk Manager servers.

> **NOTE**
> Starting with version 12.5 of CA Service Desk, the product name has changed to CA Service Desk Manager. The DX NetOps Spectrum integration supports earlier versions that used the previous name. As a result, the two product names are used interchangeably in this section. The name "CA Service Desk" is used to describe generic product features, such as tickets and assets.

Integration Details

Once the integration has been successfully configured, DX NetOps Spectrum and CA Service Desk Manager share data.

- OneClick uses CA Service Desk web services to:

  - Create CA Service Desk tickets.
  - Update alarm owners (the troubleshooters who are assigned to alarms).
  - Close the tickets.
- CA Service Desk Manager uses a custom notification that issues HTTP requests to OneClick to:
  - Update the assigned troubleshooter.
  - Clear the alarms.
- DX NetOps Spectrum and the CA Service Desk Manager integration work with SANM to provide automatic ticket creation. You configure the SANM automatic ticket creation using SANM policies. You can then specify which alarms create tickets by configuring alarm properties, such as the date, time, alarm severity, alarm cause, IP address, and the device type.
- DX NetOps Spectrum and the CA Service Desk Manager integration also provides an approval system for host configuration changes that are initiated in Network Configuration Manager. When a host configuration change is requested in NCM, a CA Service Desk ticket is created for the request. The ticket requires approval before it can be implemented.

Fault Tolerance

You can specify a list of multiple CA Service Desk Manager servers to enable fault tolerance when you configure the integration. When DX NetOps Spectrum detects a loss of connectivity to a CA Service Desk Manager, it attempts to connect to the next server in the list. DX NetOps Spectrum continues to step through the list to establish a connection until one is successful.

> **IMPORTANT**
> If CA Service Desk web services are down, DX NetOps Spectrum cannot create Service Desk tickets. Or if an alarm is cleared when CA Service Desk web services are down, that ticket is not closed, but remains open.

How to Install and Configure the Integration

To install and configure the DX NetOps Spectrum and CA Service Desk Manager integration successfully, complete the following procedures:

Verify Integration System Requirements

To perform the procedures in this section, administrator-level permission is required to the CA Service Desk Manager server host computer and the OneClick web server host computer. You must also be a CA Service Desk Manager and OneClick administrator on these computers.

Before you begin, verify that your servers meet the following requirements:

- **DX NetOps Spectrum OneClick server software** -- CA Spectrum 10.1. For system requirements, see System Requirements for Installing DX NetOps Spectrum.
- **CA Service Desk software** -- CA Service Desk 12.7 or 14.1. For system requirements, see Integration Compatibility.

> **NOTE**
> Starting with version 12.5 of CA Service Desk, the product name has changed to CA Service Desk Manager. The DX NetOps Spectrum integration supports earlier versions that used the previous name. As a result, the two product names are used interchangeably in this section. The name "CA Service Desk" is used to describe generic product features, such as tickets and assets.

For information about installing CA Service Desk software, see Service Desk and DX NetOps Spectrum . For information about configuring CA Service Desk software, see CA Service Manager .

- **A supported web browser** -- CA Service Desk and CA Service Desk Manager only support Microsoft Internet Explorer v10 if the browser is running in Compatibility Mode. Other versions of Internet Explorer are fully supported. Mozilla Firefox and Google Chrome are also supported.

Configure the CA Service Desk Manager Server

This section describes the procedures that are required to set up the CA Service Desk server for integration.

1. Download and Install Integration Components on the CA Service Desk Server
2. Create a Contact, Service Desk Ticket Template, Service Desk Web Services Policy, and Service Desk Notification Method
3. Configure the CA Service Desk Ticket Notifications

Download and Install Integration Components on the CA Service Desk Server

Before you configure CA Service Desk Manager and to communicate with each other, download and install the integration components on your CA Service Desk Manager server. Use one of the following methods, depending on your operating systems.

CA Service Desk and OneClick Web Server Use Different Operating Systems

Perform the following steps when your CA Service Desk Manager server uses a different operating system than the OneClick web server (Windows vs Linux):

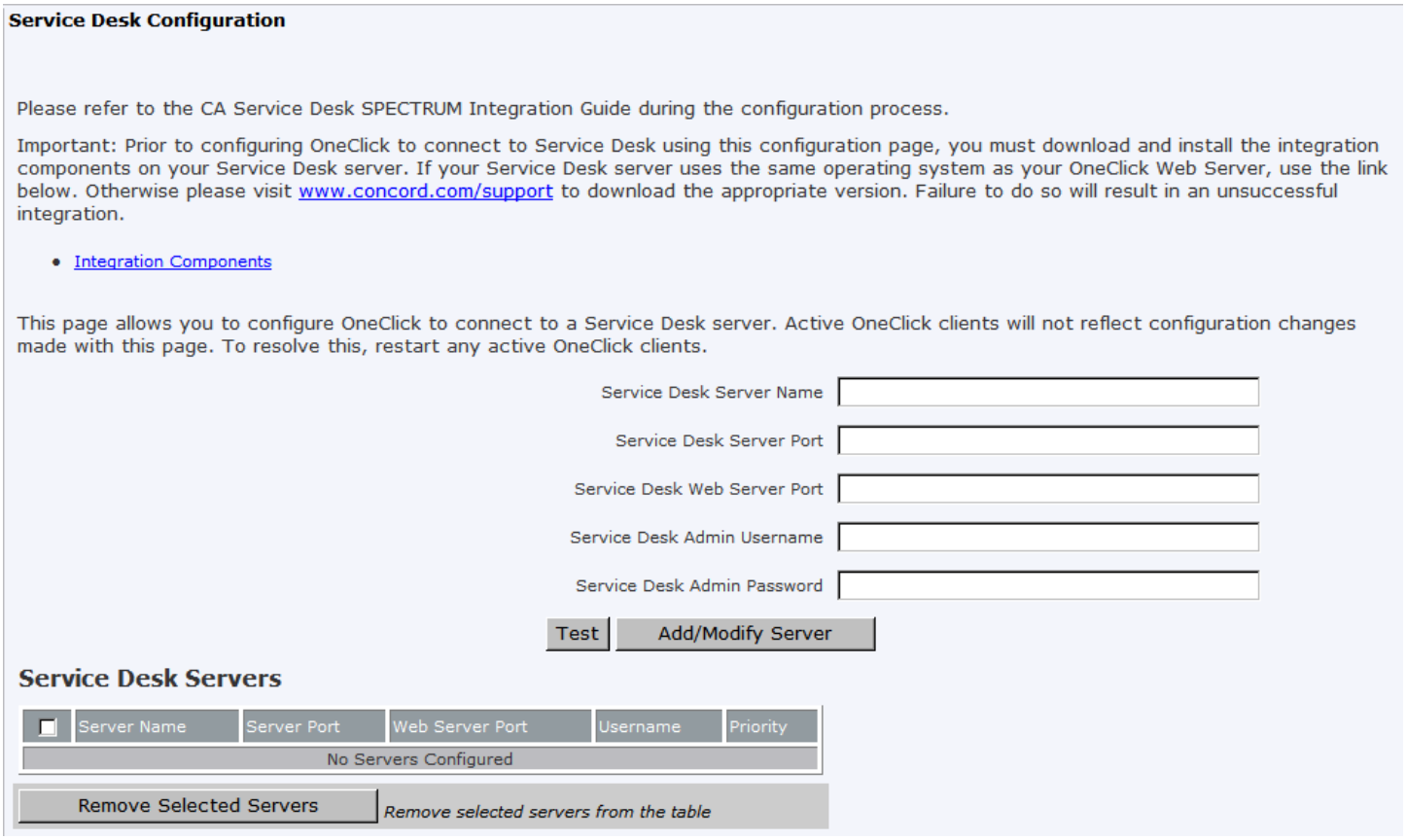
1. Visit support.broadcom.com to locate a version of the CA Service Desk Manager integration components that are appropriate for your CA Service Desk Manager server.
2. Download and save the appropriate version of the integration components for your operating system to the following directory on your CA Service Desk Manager server:
   ```
   Service_Desk_Installation_directory/bin
   ```
3. After you have saved the integration components to your CA Service Desk Manager server, follow the instructions for installing and configuring the integration components using the same operating system, beginning with Step 5.

CA Service Desk and OneClick Web Server Use the Same Operating System

Download the and CA Service Desk Manager integration components from the OneClick web server and install them on the CA Service Desk Manager server.

Follow these steps:

1. From your CA Service Desk Manager server, navigate to the OneClick Administration pages.
   ```
   http://OneClick Web server/spectrum/admin/index.jsp
   ```
2. Click the Service Desk Configuration link in the left panel of the Administration page.
   The Service Desk Configuration administration page opens in the right panel, as shown:



1. Click the Integration Components link to download the oc_components.exe file. This self-extracting archive file contains the executable programs to configure the CA Service Desk Manager server.
2. Save the oc_components.exe file to the directory on your CA Service Desk Manager server:
   ```
   Service_Desk_Installation_directory/bin
   ```
3. Log in to your CA Service Desk Manager server host computer and navigate to the *Service_Desk_Installation_directory*/bin directory.
4. Locate the oc_components.exe file you downloaded.
5. **Linux:** Run the following command to make the oc_components.exe file executable:
   ```
   chmod 755 oc_components.exe
   ```
   Windows: Do not edit the permissions of the downloaded file for the file to be executable.
6. Run the oc_components.exe file.
   The OneClickIntegrationSetup(.exe) file is extracted to the <*Service_Desk_Installation_directory*>/bin directory.
7. Run the *Service_Desk_Installation_directory*/bin/OneClickIntegrationSetup(.exe) configuration program. At each prompt, enter the requested information and press Enter to continue. The following table describes each prompt and the required information:

OneClick Integration Setup Prompt	Description
OneClick Server name?>	Enter the hostname of your OneClick Web server.
OneClick Server port?>	Enter the port of the OneClick Web server.

OneClick Homepage path [default="spectrum"]?>	If your OneClick home page URL uses the default value of http://<*oc*> *Web server*/spectrum, press Enter to accept it. Otherwise, enter the correct home page path portion of the OneClick home page URL at this prompt. This home page path value is <path>, as in the example http://*OneClick Web server*>/*path*. The default value in OneClick is “spectrum.”
Username?>	Enter the username of the OneClick Administrator. This name is the "super user" who installed the OneClick Web server.
Password?>	Enter the password of the OneClick Administrator.
Confirm password?>	Re-type the password of the OneClick Administrator and press Enter.
Enable logging? [yes|no]>	Type *yes* to enable logging or type *no* to disable logging and press Enter. We recommend enabling logging only when you are troubleshooting an integration problem. An active integration can create a large log file. When logging is enabled, the integration creates a file named oc-notification.log in the *Service_Desk_Installation_directory*/bin directory. When enabled, logging writes information about Service Desk notifications to this log file. Information is logged each time the notification occurs and includes the type of activity and whether or not it was a success. In the case of a failed notification, this log file may contain a possible solution such as an invalid port or that the OneClick web server is unavailable.
Enable SSL? [yes|no]>	Type *yes* to enable SSL or type *no* to disable SSL and press Enter. **Note:** To enable SSL in Tomcat, install OneClick and see OneClick Administration. To enable SSL in CA Service Desk Tomcat, see *Administration* in *CA* Service Desk documentation.
Path to the JRE root installation directory?>	Type the JRE root installation directory path and press Enter. Specify the JRE root installation directory so that appending "bin/java - version" becomes a successful command. **Note:** Java 2 Runtime Environment (JRE) version 1.5.0 or later is required.
Close keyword [default="Closed"]?>	If you created customized CA Service Desk notification messages that do not use the default keyword for Close (“Closed”), specify those custom keywords at this prompt. To use default values press Enter at the prompt without specifying a keyword. If you use custom notification messages for the Close action, type the associated keywords at the prompt. You can specify multiple keywords and they will be searched for sequentially. Keywords are case-sensitive. When you have finished entering keywords, leave the line blank and press Enter. **Note:** For more information, see Using Custom Keywords for CA Service Desk Notifications section in the References page.
Transfer keyword [default="Transfer"]?>	If you created customized CA Service Desk notification messages that do not use the default keyword for Transfer ("Transfer"), specify those custom keywords at this prompt. To use default values press Enter at the prompt without specifying a keyword. If you use custom notification messages for the Transfer action, type the associated keywords at the prompt. You can specify multiple keywords and they will be searched for sequentially. Keywords are case-sensitive. When you have finished entering keywords, leave the line blank and press Enter.

The OneClick Integration Setup program creates a file named NotifyOneClick(.bat or .sh, depending on the operating system), in the *Service_Desk_Installation_directory*/bin directory.

NOTE
To reconfigure this information later, run the OneClickIntegrationSetup program again or manually edit values in the *Service_Desk_Installation_directory*/bin/oc-integration.cfg configuration file. Any changes that are made to this file take effect immediately. No additional restart is required.

NOTE

If you have ennabled SSL during the OneClick Integration Setup, follow these steps to set One Click Keystore in the CA Service Desk Manager (SDM):

1. Copy the OC Keystore file: **<SPECROOT>/custom/keystore/cacerts** to the CA Service Desk Manager (SDM) machine. For example: **c:/spectrum_keystore/cacerts**
2. Update NotifyOneClick (.bat or .sh, depending on the operating system) as shown:
   ```
   <JRE_ROOT_PATH> \bin\java.exe -DFILE_NAME="%FILE_NAME%" -jar OCNotify.jar
   ```
   For example:
   ```
   C:\Program Files\Java\jre1.8.0_172\bin\java.exe" -
   Djavax.net.ssl.trustStore=c:/spectrum_keystore/cacerts -DFILE_NAME="%FILE_NAME
   %" -jar OCNotify.jar
   ```
   A customized setting in case OC is SSL enabled, can be for example:
   ```
   C:\Program Files\Java\jre1.8.0_172\bin\java.exe -DFILE_NAME="%FILE_NAME%" -jar
    OCNotify.jar2nd box
   ```

Create a Contact, Service Desk Ticket Template, Service Desk Web Services Policy, and Service Desk Notification Method

Create a DX NetOps Spectrum Contact on the CA Service Desk Manager Server

To enable CA Service Desk Manager to communicate with DX NetOps Spectrum, create a special DX NetOps Spectrum contact on the CA Service Desk Manager server.

Follow these steps:

1. Navigate to your CA Service Desk Manager home page:
   ```
   http://<Service Desk server>/CAisd/pdmweb.exe
   ```
2. Click the Service Desk tab.
3. Click File, New Contact.
   The Create New Contact window opens.
4. Enter *spectrum* in both the Last Name and System Login (or User ID) fields.
5. Select at least the Analyst option from the Contact Type list so that tickets are assigned to the user.
6. Click Save.

Create a CA Service Desk Ticket Template on the CA Service Desk Manager Server

Create a service desk ticket template for the CA Service Desk tickets that are created from DX NetOps Spectrum alarms. This ticket template specifies the format of CA Service Desk tickets that are created from OneClick alarms.

Follow these steps:

1. From the CA Service Desk Manager server home page, select the Service Desk tab.
2. Select File, New Issue.
   The Create New Issue window opens.

NOTE
You can configure CA Service Desk Manager to use Issues, Requests, or Incidents as the default ticket type that OneClick creates. The procedures in this section refer to Issues. To use Requests or Incidents instead (to support Incident and Problem type requests), replace all references to "Issues" with "Requests" or "Incidents". For example, in this step, replace 'File, New Issue' with 'File, New Request,' or with 'File, New Incident'. For an example using Requests, see the topic that describes creating CA Service Desk tickets automatically for a single alarm type.

3. In the Create New Issue window, type *spectrum* in the Affected End User field.
4. (Optional) Take the following actions to configure the integration to assign all CA Service Desk tickets that DX NetOps Spectrum creates to a specific troubleshooter by default:
   a. Click the Assignee link.
      The Analyst List page opens.
   b. Search for the contact name of the person you want to set as the default troubleshooter.
   c. Click the link in the Name column of the desired troubleshooter.
      The troubleshooter is added to the Assignee field as the default.
5. Select the Template tab at the bottom of the Create New Issue page.
6. Type *SPECTRUM_TEMPLATE* in the Template Name field.
7. (Optional) Type a description for this template in the Description field.
8. Click Save.
   The template is saved.

NOTE
You can customize your DX NetOps Spectrum and CA Service Desk Manager integration to use more than the default template. By editing the SPECTRUM_POLICY Web Services Policy, you can add different problem types that refer to different ticket templates. For more information about adding problem types (error types) to the Web Services Policy in CA Service Desk Manager, see CA Service Desk Manager Web Services .

Create a CA Service Desk Web Services Policy and Problem Type for DX NetOps Spectrum

Create a Web Services Policy and problem type (error type) for DX NetOps Spectrum. This policy controls how CA Service Desk Manager processes the requests for the creation of tickets from DX NetOps Spectrum alarms. CA Service Desk Manager uses the problem type to specify the SPECTRUM_TEMPLATE as the basis for new tickets when CA Service Desk Manager receives a DX NetOps Spectrum alarm.

Follow these steps:

1. From the CA Service Desk Manager home page, select the Administration tab.
2. Expand Web Services Policy and click Policies.
3. Click Create New.
   The Create New Web Services Access Policy window opens.
4. Take the following actions:
   a. Enter *SPECTRUM_POLICY* for Symbol.
   b. Enter *SPECTRUM_POLICY* for Code.
   c. (Optional) Type a description for the DX NetOps Spectrum policy.
   d. Click Save.
   The Create New Web Services Access Policy window closes.
5. Select the *SPECTRUM_POLICY* Web Service Policy that you created.
   The Web Services Access Policy Detail window for the *SPECTRUM_POLICY* opens.
6. Click Edit.
7. Click the Error Types tab.
8. Click Add an Error Type.

a. Enter *SPECTRUM_PT* in the Symbol field.
b. Enter *SPECTRUM_PT* in the Code field.
c. Select the Default check box.
d. Select Issue from the Ticket Template Type drop-down list.
e. Enter *SPECTRUM_TEMPLATE* in the Ticket Template Name field.
f. (Optional) Enter a description for the problem type (error type).
g. Select the Duplicate Handling tab, and take one of the following actions:
   - To create a unique ticket regardless of underlying cause, select Create Ticket (do not detect duplicates).
   - To avoid creating multiple tickets for the same underlying cause, select one of the following options:
     - Add Activity Log (do not create ticket)
     - Create Standard Log (do not create ticket)
     - Attach As Child (create a child ticket)

     Specify a value in the 'Maximum time interval for searching duplicates' field. This value must be at least 00:01:00 (one minute). The value the amount of time CA Service Desk Manager looks for duplicates.

     NOTE
     These options require support for the CA Service Desk Manager duplicate handling feature. To use this feature, make further modifications to the CA Service Desk Manager and OneClick servers.

     For more information, see the following topics:
   - Configure CA Service Desk Manager Duplicate Handling Feature
   - Disable Automatic Closing of Ticket

h. Click Save.

The window closes.

9. Click Save in the Policy Detail window.
   The policy and problem type (error type) are created.

   NOTE
   You can customize your DX NetOps Spectrum and CA Service Desk Manager integration to use more than the default template. By editing the SPECTRUM_POLICY Web Services Policy, you can add different problem types that refer to different ticket templates. For more information about adding problem types (error types) to the Web Services Policy in CA Service Desk Manager, see CA Service Desk Manager Web Services .

Create a Custom CA Service Desk Notification Method

Create a custom notification method for CA Service Desk Manager to send notifications to OneClick. This method communicates the CA Service Desk ticket changes to OneClick.

Follow these steps:

1. From the CA Service Desk Manager home page, click the Administration tab.
2. Expand the Notifications folder and click Notification Methods.
   The Notification Method List opens.
3. Click Create New.
   The Create New Notification Method dialog opens.
4. Enter the following information:

   1. Type **SPECTRUM_Notification** in the Symbol field.
   2. Specify the notification method as follows:
      For Windows: Type **NotifyOneClick.bat** for the Notification Method.
      For Linux: Type the full path to the NotifyOneClick script, such as **/opt/CA/ServiceDesk/bin/NotifyOneClick.sh**.

      NOTE
      The NotifyOneClick file must be present in the *Service_Desk_Installation_directory*/bin directory.
   3. (Optional) Enter a description of the notification method.

NOTE
Do not select *write to file* because the integration uses web services instead of reading the information from a file.

4. Click Save.

The notification method is created.

Configure the CA Service Desk Ticket Notifications

CA Service Desk Manager can send OneClick notifications when a ticket that is associated with a OneClick alarm changes. These notifications update the alarm that is associated with a ticket in OneClick to reflect changes to the ticket. The CA Service Desk Manager integration can be configured to generate an automatic Ticket Closed notification that causes OneClick to clear the associated alarm, when a ticket is closed. Similarly, when a ticket has been transferred, a Ticket Transfer notification causes OneClick to update the troubleshooter information for the associated alarm.

NOTE
OneClick only clears an associated alarm when a ticket is closed in the CA Service Desk Manager if the alarm is user-clearable.

These notifications use CA Service Desk "keywords," which must match keywords that are set in the integration for OneClick. Keywords are case-sensitive. By default, the keyword for the close action is "Closed", and the keyword for the transfer action is "Transfer" (in both CA Service Desk Manager and the integration setup). These keywords can be customized in CA Service Desk Manager and in the integration setup.

To configure the CA Service Desk notifications for the integration with CA Service Desk Manager, complete these procedures:

1. Enable notifications for the DX NetOps Spectrum contact.
2. Enable CA Service Desk notifications for "Ticket Close".
3. Enable CA Service Desk notifications for "Ticket Transfer" actions.

For more information, see the example of customizing CA Service Desk keywords. This example describes configuring a custom keyword for the Close action after you have completed and enabled the integration.

Enable Notifications for the Contact

Enable the CA Service Desk notifications for the special contact (*spectrum*) on the CA Service Desk Manager server:

Follow these steps:

1. From the CA Service Desk Manager home page, select the Service Desk tab.
2. Select Search, Contacts.
   The Create New Contact window opens.
3. Type **spectrum** in the Last Name field, and click Search.
   The Contact List opens.
4. Click the *spectrum* contact.
   The Detail dialog for the *spectrum* contact opens.
5. Click Edit and click the Notification tab.
6. For each of the Notification types (Low, Normal, High, and Emergency) in the list, select *SPECTRUM_Notification* for Method.
7. Click Save.
   Notifications are now enabled.

Enable CA Service Desk Notifications for “Ticket Close”

Configure CA Service Desk Manager to send a notification to when a ticket has been closed. A Ticket Closed notification causes to clear the associated alarm.

Follow these steps:

1. From the CA Service Desk Manager home page, select the Administration tab.
2. Expand the Notifications folder and select Activity Notifications.
   The Notification List opens.
3. Select Close activity.
   The Close Activity Notification Detail dialog opens.
4. Verify that the Object Type is set to the appropriate value. By default, this field is set to Requests. If you opted to create CA Service Desk Issues, select Issues.
5. Click Edit.
6. Take the following steps:
   a. Click the Notification Rules tab, and then click the name of the notification rule that the close activity uses.
   b. Click the message template for the transfer activity, and click Edit.
   c. Select Auto Notification.
7. Click the Contacts tab, and then click Update Contacts.
   The Contact Search window opens.
8. Click Search.
   The Notification Recipients Update window opens.
9. Add the *spectrum* contact from the Contacts list to the Notification Recipients list, and click OK.
   The Close Update Activity Notification window opens.

   NOTE
   Do not remove the information in the 'Description: @{call_req_id.description}' field. uses this information to associate alarms in with the CA Service Desk ticket. Without this information, alarms are not cleared in when a CA Service Desk ticket is closed.

1. Click Save.
2. Close the Close Activity Notification Detail window.
   CA Service Desk Manager sends notifications when tickets are closed.

Enable CA Service Desk Notifications for "Ticket Transfer" Actions

CA Service Desk Manager can notify when a ticket has been transferred. To configure this feature, enable CA Service Desk Manager to send notifications to when a ticket that has a alarm is transferred.

Follow these steps:

1. From the CA Service Desk Manager server home page, select the Administration tab.
2. Expand the Notifications folder and select Activity Notifications.
   The Notification List opens.
3. Select Transfer activity.
   The Transfer Activity Notification Detail window opens.
4. Verify that the Object Type is set to the appropriate value. The default is Requests. To create CA Service Desk Issues, select Issues.
5. Click Edit.
6. Take the following steps:
   a. Click the Notification Rules tab, and then click the name of the notification rule that the transfer activity uses.
   b. Click the message template that the transfer activity uses, and click Edit.
   c. Select Auto Notification.

7. In the Notification Message body, change the assignment information from:

   ```
   "Assigned to: @{call_req_id.assignee.combo_name}"
   ```

   to:

   ```
   "Assigned to: @{issue_id.assignee.userid}"
   ```

   NOTE
   Do not remove the information in the 'Description: @{call_req_id.description}' field. uses this information to associate alarms in with the CA Service Desk ticket. Without this information, alarms are not cleared in when a CA Service Desk ticket is closed. The 'issue_id' portion of the assignment is a variable that must match the type of ticket that you are using (in this case, an issue). For requests, use 'call_req_id'.
8. Click the Contacts tab, and then click Update Contacts.
   The Contact Search window opens.
9. Click Search.
   The Notification Recipients Update window opens.
10. Add the *spectrum* contact to the Notification Recipients list from the Contacts list, and click OK.
    The Transfer Update Activity Notification window opens again.
11. Click Save.
12. Close the Transfer Activity Notification Detail window.
    CA Service Desk Manager sends notifications when tickets are transferred.

Configure CA Service Desk Duplicate Handling

CA Service Desk Manager provides the duplication handling for scenarios where multiple tickets are created with the same underlying cause within a specified amount of time. To take advantage of this feature, make some modifications to the CA Service Desk Manager server.

NOTE
To support the duplication handling,enable the Specify Reported By Field when configuring the OneClick server. See Configure Communication and Enable Integration on the OneClick Server.

Configure support for CA Service Desk duplication handling on the CA Service Desk Manager server.

Follow these steps:

1. Log in to your CA Service Desk Manager server.
2. Take one of the following steps:
   For CA Service Desk r12.1:

   1. Open the following file: $SD_ROOT\bopcfg\majic\cm.maj
   2. Change the following line:

      ```
      log_agent SREL cnt WRITE_NEW REQUIRED SERVICE_PROVIDER_ELIGIBLE
      ```

      to

      ```
      log_agent SREL cnt REQUIRED SERVICE_PROVIDER_ELIGIBLE
      ```

   3. Save changes and close the file.

   For CA Service Desk Manager r12.5, and later versions:

   - Create the mod file in $SC_ROOT/site/mods/majic.
   - Include the following line:

     ```
     MODIFY cr log_agent NOT_WRITE_NEW;
     ```

   - Save changes and close the file.
3. Restart the CA Service Desk Manager Server service after making either of these changes.

Configure the OneClick Server

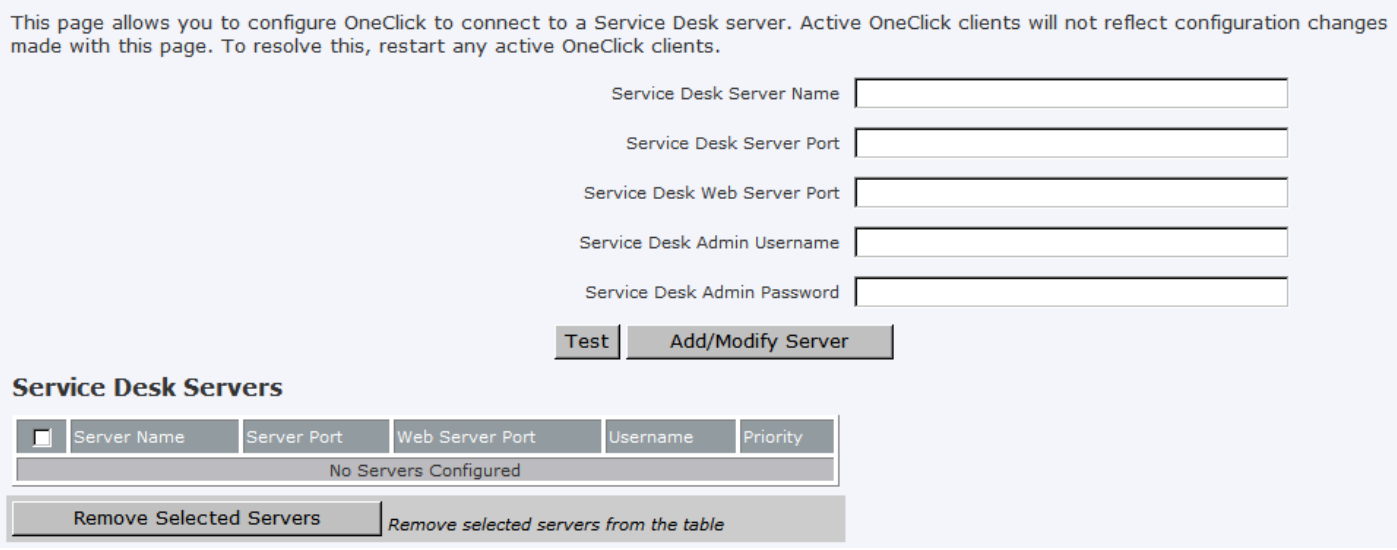
This section describes the procedures to set up the OneClick server for integration.

Configure Communication and Enable Integration on the OneClick Server

Complete the following procedure to configure and CA Service Desk Manager to communicate with each other.

Follow these steps:

1. Navigate to the OneClick Administration pages:
   `http://<oc> Web server/spectrum/admin/index.jsp`
2. Click the Service Desk Configuration link from the left panel of the Administration page.
   The Service Desk Configuration administration page opens.
3. Configure OneClick to connect to a CA Service Desk Manager server by entering valid values for the following fields:



- **Service Desk Server Name**
  The hostname of the CA Service Desk Manager server.
- **Service Desk Server Port**
  The HTTP port of the CA Service Desk Tomcat Port. uses this port to create CA Service Desk tickets.
- **Service Desk Web Server Port**
  The HTTP port of the CA Service Desk Web Server. This port is used when launching the CA Service Desk interface from OneClick.
  - To integrate with a CA Service Desk Manager server that is running on both a web server (IIS or Apache) and Tomcat, this port is *different* from the port that you specified in the previous Service Desk Server Port field.
  - To integrate with a CA Service Desk Manager server that is only running on Tomcat, this port is the *same* as the port that you specified in the previous Service Desk Server Port field.
- **Service Desk Admin Username**
  The user name of the CA Service Desk Manager server administrator.
- **Service Desk Admin Password**
  The password of the CA Service Desk Manager server administrator.

  NOTE
  OneClick clients that are running when you make configuration changes on this page do not reflect the changes. To resolve this problem, restart any active OneClick clients. If you change to a different CA Service Desk Manager server, restart the tomcat server for the changes to take effect.

4. Click Test to verify the connection between the CA Service Desk Manager server and OneClick.

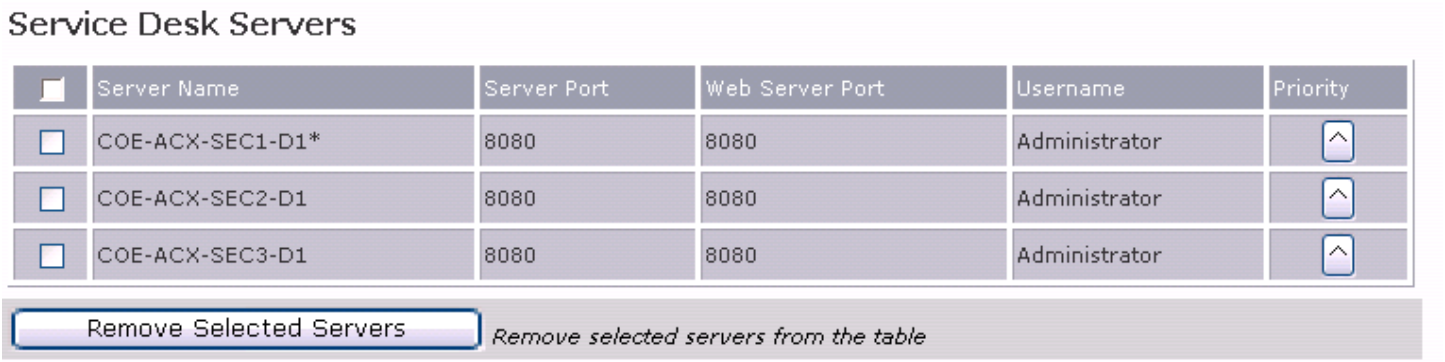
A successful test displays the following information:

```
Successfully connected to Service Desk web services and interface on server Service Desk Manager Server Name.
```

5. Click Add/Modify Server to add these settings to the Service Desk Manager Server table.

The server is added to the Service Desk Servers table.

NOTE
The settings are not saved until you click Save.



The table in this image displays all CA Service Desk servers that have been configured for this integration. The asterisk (*) indicates the connection that is currently in use.

6. Click Save to save these settings.

7. To add another Service Desk Manager server, specify the server details, click Add/Modify Server, and then click Save.

8. You can perform the following tasks from the Service Desk Servers table:

- Edit an existing Service Desk Manager Server by taking the following steps:
  a. Select the server to modify by highlighting the row in the table.

  NOTE
  Do not use the check boxes for the Modify operation.
- To test a server that has already been added, click that row in the server list, specify a password, and click the Test button.
- To increase the priority of a CA Service Desk Manager server, click the up arrow icon to move the server up by one row in the table. Connections are tried in the order that the servers appear in the table. Click Save.
- To remove a CA Service Desk Manager server, select the server, click 'Remove Selected Servers', and then click Save.

9. Supply values for the following additional fields:

- **SSL Support**
  Select the Enabled option to enable SSL.
- **Specify Reported By Field**
  Select Yes to include the submitting user in the Reported By field when you manually submit Request, Incident and Problem tickets.

  NOTE
  Modifications are required on the CA Service Desk Manager server for this feature to function correctly. For more information, see Configure CA Service Desk Duplication Handling.
- **Assign Assets/Configuration Items**
  If enabled, lets you associate assets with the CA Service Desk tickets from OneClick.

- **Reload Asset/CI Mapping**
  When Assign Assets/Configuration Items are enabled, click this button if you have modified the Service Desk asset/CI mapping in service-desk-asset-mapping.xml and you want to apply the changes without restarting the OneClick server.

  NOTE
  The exception "Unable to start ServiceDeskAssetMapping: java.lang.NullPointerException" is logged in the stdout.log file at $SPECROOT\tomcat\logs when you start OneClick and the SpectroSERVER is inactive. This problem is resolved automatically as performs Service Desk Asset Mapping when the first Service Desk ticket is generated. For information on setting alarm types, see Choose DX NetOps Spectrum Alarm Types in OneClick for Automatic Ticket Creation.

10. Click Save. The following message appears:

```
Successfully saved configuration.
```

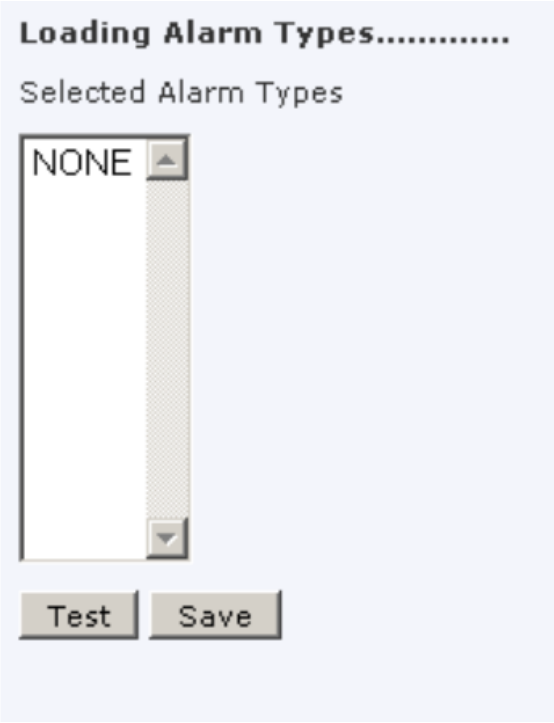
Select Alarm Types in OneClick for Automatic Ticket Creation

Configure the and CA Service Desk Manager integration to create CA Service Desk trouble tickets when generates alarms of certain types that you specify.

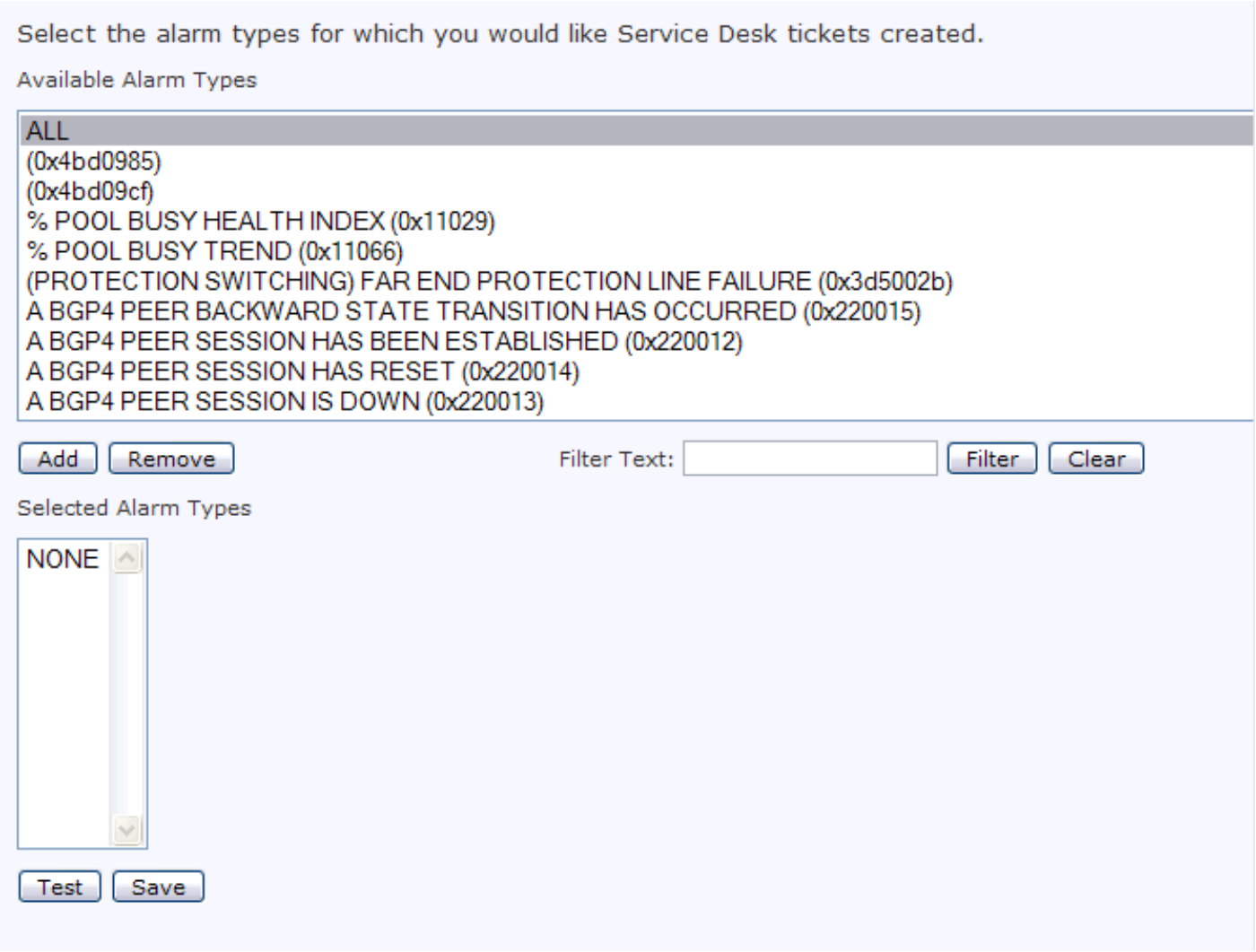
Note: Automatic ticket creation is an optional feature. Operators can instead create all tickets manually. By default, the integration does not automatically create tickets for any alarms.

Follow these steps:

1. Navigate to your OneClick Administration pages:
   `http://<oc> Web server/spectrum/admin/index.jsp`
2. Click the Service Desk Configuration link in the left panel of the Administration page. The Service Desk Configuration administration page opens.
3. Scroll to the bottom of the Service Desk Configuration administration page to the Alarm Types section. The first time that you open the Service Desk Configuration administration page and anytime the cache timeout (one hour) of the page expires, the “Loading Alarm Types” message is displayed as shown:



The available alarm types are displayed in the Available Alarm Types section of the Service Desk Configuration administration page:



4. In the Available Alarm Types section, select the alarms for which you want OneClick to create CA Service Desk tickets, and click Add.

   NOTE
   A delay can occur the first time you add alarm types. The delay occurs when all of the probable cause files are loaded for display in the Available Alarm Types section of the Service Desk Configuration administration page.

   To generate the CA Service Desk tickets for all alarms, select ALL from the Available Alarm Types list, and click Add.

   NOTE
   To select an individual alarm type, enter some text from the desired alarm type into the Filter Text field and click Filter.

5. Click Save when you have finished adding alarms.

Customizing Ticket Creation and Closure

Using Custom Ticket Creation Rules

Using custom ticket creation rules, you can extend the functionality of CA Service Desk Manager tickets. Custom rules are available for tickets that are created by DX NetOps Spectrum Alarm Notifier (SANM), or that are created manually through the OneClick interface. You can add contacts to CA Service Desk Manager or problem types to the SPECTRUM_POLICY Web Services Policy. Tickets that DX NetOps Spectrum creates can then use different templates or can be assigned to different end-users.

Set up custom ticket creation rules to notify the right person who must respond to an alarm. Apply a template with the required information to help IT staff resolve the issue more quickly.

With custom rules, the problem type or affected end user that is assigned to a CA Service Desk ticket is based on information from DX NetOps Spectrum. The following parameters can be used to determine the assignment:

- Alarm attribute
- Model attribute
- Model association

The CA Service Desk tickets that DX NetOps Spectrum creates use default settings if they are not associated with any of the ticket creation rules. These tickets use the following defaults, which are defined for the SPECTRUM_POLICY Web Services Policy:

- Default problem type (error type) (SPECTRUM_PT)
- Default end-user (DX NetOps Spectrum).

Add a Ticket Creation Rule

By editing the ticket configuration file (service-desk-ticket-config.xml), you can customize how the CA Service Desk tickets are created from DX NetOps Spectrum. You can add a ticket creation rule that is based on any attribute of the DX NetOps Spectrum alarm, any attribute of the DX NetOps Spectrum model, or any association of the DX NetOps Spectrum model.

Any attribute of the DX NetOps Spectrum alarm, any attribute of the DX NetOps Spectrum model, or any association of the DX NetOps Spectrum model can have a certain problem type, affected end user, or both, assigned to the ticket in CA Service Desk Manager. Creating CA Service Desk tickets that are based on custom ticket creation rules enhances the ability of the CA Service Desk Manager user to troubleshoot issues.

Follow these steps:

1. Copy the service-desk-ticket-config.xml and service-desk-ticket-config.xsd files from <*$SPECROOT*>/tomcat/webapps/spectrum/WEB-INF/svdsk/config to <*$SPECROOT*>/custom/svdsk/config.
2. Open the service-desk-ticket-config.xml file in a text editor.
3. Create an <sd-ticket> element inside the <service-desk-ticket-config> element for each rule that you want to define. Follow the instructions that are provided in the XML file.
   You can create rules that are based on the following information:
   - Alarm attribute
   - Model attribute
   - Model association

   For each rule, you can specify a problem type, a user ID, or both -- at least one of these attributes must be provided for each rule.

   NOTE
   If an alarm is generated for a model where multiple ticket creation rules can be applied, the rules that are defined earliest in the XML file take precedence over the rules that follow.
4. Save the file.
   Your ticket creation rules are added to CA Service Desk Manager.
5. Click the Reload Ticket Rules button that exists on the OneClick Administration page for the CA Service Desk Manager integration.

Your custom rules are applied, and CA Service Desk tickets that are created from DX NetOps Spectrum use your custom creation rules.

NOTE
If you restart Tomcat, the custom rules are applied automatically.

Examples: Create a Ticket Creation Rule Based on Alarm Attribute

The following examples show three ticket creation rules that are based on alarm attribute (alarm code). These examples demonstrate how to specify a problem type, a user ID, and both, respectively. For these examples, the SPECTRUM_MGT_PT is a problem type added to the SPECTRUM_POLICY Web Services Policy, and northeast_IT is the system name of a contact in CA Service Desk Manager.

```
<sd-ticket>
   <alarm-attribute>
    <attr-id>0x11f50</attr-id> <!-- attribute ID for alarm code -->
    <value>0x10701</value> <!-- alarm code attribute value -->
   </alarm-attribute>
   <sd-problem-type>SPECTRUM_MGT_PT</sd-problem-type>
</sd-ticket>
<sd-ticket>
  <alarm-attribute>
    <attr-id>0x11f50</attr-id> <!-- attribute ID for alarm code -->
    <value>0x119d3</value> <!-- alarm code attribute value -->
  </alarm-attribute>
  <sd-userid>northeast_IT</sd-userid>
</sd-ticket>
<sd-ticket>
  <alarm-attribute>
    <attr-id>0x11f50</attr-id> <!-- attribute ID for alarm code -->
    <value>0x10306</value> <!-- alarm code attribute value -->
  </alarm-attribute>
  <sd-problem-type>SPECTRUM_MGT_PT</sd-problem-type>
  <sd-userid>northeast_IT</sd-userid>
</sd-ticket>
```

Examples: Create a Ticket Creation Rule Based on Model Attribute

The following examples show three ticket creation rules that are based on model attribute (model handle). These examples demonstrate how to specify a problem type, a user ID, and both, respectively. For these examples, the SPECTRUM_MGT_PT is a problem type added to the SPECTRUM_POLICY Web Services Policy, and northeast_IT is the system name of a contact in CA Service Desk Manager.

```
<sd-ticket>
  <model-attribute>
    <attr-id>0x129fa</attr-id> <!-- attribute ID for model handle -->
    <value>0x1000d5</value> <!-- model handle attribute value -->
  </model-attribute>
  <sd-problem-type>SPECTRUM_MGT_PT</sd-problem-type>
</sd-ticket>
<sd-ticket>
  <model-attribute>
    <attr-id>0x129fa</attr-id> <!-- attribute ID for model handle -->
    <value>0x100012</value> <!-- model handle attribute value -->
  </model-attribute>
  <sd-userid>northeast_IT</sd-userid>
```

```
</sd-ticket>
<sd-ticket>
  <model-attribute>
    <attr-id>0x129fa</attr-id> <!-- attribute ID for model handle -->
    <value>0x100023</value> <!-- model handle attribute value -->
  </model-attribute>
  <sd-problem-type>SPECTRUM_MGT_PT</sd-problem-type>
<sd-userid>northeast_IT</sd-userid>
</sd-ticket>
```

Examples: Create a Ticket Creation Rule Based on Model Association

The following examples show three ticket creation rules that are based on model association (models dynamically collected by a global collection named 'switch routers collection', models that are monitored by the service container 'northeast service container', or models connecting to another model named 'northeast firewall'). These examples demonstrate how to specify a problem type, a user ID, and both, respectively. For these examples, the SPECTRUM_MGT_PT is a problem type added to the SPECTRUM_POLICY Web Services Policy, and northeast_IT is the system name of a contact in CA Service Desk Manager.

```
<sd-ticket>
  <model-association>
    <relation>0x1003a</relation> <!-- the relation ID for dynamicGlobalCollects -->
    <left-model-name>switch routers collection</left-model-name> <!-- model name of the lhs -->
   </model-association>
  <sd-problem-type>SPECTRUM_MGT_PT</sd-problem-type>
</sd-ticket>
<sd-ticket>
  <model-association>
    <relation>0x4500004</relation> <!-- the relation ID for SlmMonitors -->
    <left-model-name>northeast service container</left-model-name> <!-- model name of the lhs -->
   </model-association>
  <sd-userid>northeast_IT</sd-userid>
</sd-ticket>
<sd-ticket>
  <model-association>
    <relation>0x10005</relation> <!-- the relation ID for Connects_to -->
    <right-model-name>northeast firewall</right-model-name> <!-- model name of the rhs -->
   </model-association>
  <sd-problem-type>SPECTRUM_MGT_PT</sd-problem-type>
  <sd-userid>northeast_IT</sd-userid>
</sd-ticket>
```

Using the CA Service Desk Duplicate Handling Feature

CA Service Desk Manager provides the duplication handling for scenarios where multiple tickets are created with the same underlying cause within a specified amount of time. To take advantage of this feature, make the following modifications to both the CA Service Desk Manager server and the OneClick server.

The following actions serve as a checklist for enabling this feature:

- Configure the CA Service Desk Manager server to support DX NetOps Spectrum use of this feature. For more information, see Configure CA Service Desk Duplicate Handling Feature.
- Modify the OneClick server to disable automatic closing of tickets when the associated alarm is cleared. For more information, see Disable Automatic Closing of Ticket.
- Make sure that the error type specifies the appropriate duplicate handling action. For more information, see Create a CA Service Desk Web Services Policy and Problem Type (Error Type) for DX NetOps Spectrum.

  NOTE
  If the Error Type has already been created, you can edit the Error Type directly.
- Include the Reported By value when manually creating tickets. See the description of the 'Specify Reported By Field' in Configure Communication and Enable Integration on the OneClick Server.

Disable Automatic Ticket Closure

To use the CA Service Desk Duplicate Handling feature, disable automatic closing of tickets when the associated alarm is cleared. Otherwise, when tickets are closed automatically, duplicate handling does not occur because the ticket is closed before a new ticket request is received.

Follow these steps:

1. Shut down your Tomcat server if it is running.
2. Open the following file in a text editor:
   ```
   <$SPECROOT>/custom/svdsk/config/service-desk-config.xml
   ```

   NOTE
   This file is created after you have configured your OneClick server for integration with CA Service Desk Manager. If this file does not exist, open the following file instead:

   ```
   <$SPECROOT>/tomcat/webapps/spectrum/WEB-INF/svdsk/config/service-desk-config.xml
   ```
3. Remove the following lines:
   ```
   <handler-action>
    <action-tag>service-desk-clear</action-tag>
    <action-class>com.aprisma.spectrum.app.sd.handler.ServiceDeskClearAction</action-class>
   </handler-action>
   <alarm-handler-clear-action>
    <service-desk-clear />
   </alarm-handler-clear-action>
   ```
4. Save and close the file.
5. Restart the Tomcat server.

Configure Ticket Status

When an alarm is cleared in DX NetOps Spectrum, the ticket status in CA Service Desk Manager is updated to "Closed" status by default. You can change the status of the ticket in a DX NetOps Spectrum configuration file.

CA Service Desk tickets are created when an alarm is raised in DX NetOps Spectrum, and when it is cleared, Service Desk tickets for alarms that are cleared are set to the status that is defined in the file 'service-desk-config.xml'. If nothing is defined in that configuration file, the default status, “Closed” (CL), is used.

WARNING
Be sure to verify the code of the status that you want to set. For example, RE is the Code for the status “Resolved”; CL is the code for the status “Closed”. If you specify an incorrect code, the CA Service Desk tickets are not updated. They remain open, even after the corresponding alarm is cleared in DX NetOps Spectrum.

Follow these steps:

1. Log in to the OneClick server.
2. Open the following file in a text editor:
   `<$SPECROOT>/custom/svdsk/config/service-desk-config.xml`

   NOTE
   This file is created after you have configured your OneClick server for integration with CA Service Desk Manager.
3. Perform a search for the “ticket-status” tag.
4. If “ticket-status” is not found, add the following tag:
   `<ticket-status>RE</ticket-status>`

   NOTE
   RE is the Code for the Resolved status in CA Service Desk.
5. If the <ticket-status /> tag is present, replace it with the following tag:
   `<ticket-status>RE</ticket-status>`
6. Save the service-desk-config.xml file.
7. Navigate to the OneClick Administration pages.
8. Click the Service Desk Configuration link from the left panel of the Administration page.
   The Service Desk Configuration administration page opens.
9. Click Save.
   The changes to the XML can now take effect. When an alarm is cleared in DX NetOps Spectrum, the ticket status in CA Service Desk Manager is updated to "Resolved" status.

   NOTE
   You can instruct DX NetOps Spectrum to clear the alarm when you update the corresponding ticket status to “Resolved” in CA Service Desk Manager. For more information, see Example: Using Custom Keywords for CA Service Desk Notifications.

Reset Ticket Status to Default Settings

If you have customized the settings that determine the ticket status of CA Service Desk tickets, you can revert these changes. To reset ticket status to the out-of-the-box setting, edit the same configuration file that you used to select a custom status.

1. Follow these steps:
2. Log in to the OneClick server.
3. Open the following file in a text editor:
   `$SPECROOT/custom/svdsk/config/service-desk-config.xml`
4. Perform a search for the “ticket-status” tag.
5. Replace the complete tag (<ticket-status>RE</ticket-status>) with the following tag:
   `<ticket-status />.`
6. Save the service-desk-config.xml file.
7. Navigate to the OneClick Administration pages.
8. Click the Service Desk Configuration link from the left panel of the Administration page.
   The Service Desk Configuration administration page opens.
9. Click Save.
   The changes to the XML can now take effect. Ticket settings are restored to their defaults.

Configure Ticket Summary

When a ticket is created in CA Service Desk Manager for an alarm in DX NetOps Spectrum, the Originating Event or Cause Code text in the Summary field of the ticket is populated automatically. You can customize the text that is used to populate the Summary field by modifying a DX NetOps Spectrum configuration file.

When an alarm is raised in DX NetOps Spectrum, a ticket is created in CA Service Desk Manager. The Summary field of the ticket is populated with the Originating Event text or Cause Code that is defined in the 'service-desk-config.xml' file.

> **WARNING**
> If nothing is configured for this parameter, or if an incorrect attribute is supplied, the Summary field is not updated. CA Service Desk Manager supplies the default Description text in the Summary field.

Follow these steps:

1. Log in to the OneClick server.
2. Open the following file in a text editor:
   ```
   $SPECROOT/custom/svdsk/config/service-desk-config.xml
   ```
   > **NOTE**
   > This file is created after you have configured your OneClick server for integration with CA Service Desk Manager.
3. Perform a search for the “ticket-summary” tag.
4. If “ticket-summary” is not found, add *one* of the following tags:
   - Originating Event :
     ```
     <ticket-summary>0x1296e</ticket-summary>
     ```
   - Cause Code :
     ```
     <ticket-summary>0x11f50</ticket-summary>
     ```
     > **NOTE**
     > Only the Originating Event or Cause code text in the Summary field of the CA Service Desk ticket can be customized. The above identifiers include the corresponding required attribute IDs.
5. If "ticket-summary" is found, replace the existing tag with one of the tags that are supplied in the previous step.
6. Save the service-desk-config.xml file.
7. Navigate to the OneClick Administration pages.
8. Click the Service Desk Configuration link from the left panel of the Administration page.
   The Service Desk Configuration administration page opens.
9. Click Save.
   The changes to the XML can now take effect. The Summary field of each CA Service Desk ticket that is initiated by a DX NetOps Spectrum alert is populated with the Originating Event text or Cause Code that you specified.

Reset Ticket Summary to Default Settings

If you have customized the settings that determine the summary text that appears in CA Service Desk tickets, you can revert these changes. To reset ticket summary to the out-of-the-box setting, edit the same configuration file that you used to select a custom summary.

1. Follow these steps:
2. Log in to the OneClick server.
3. Open the following file in a text editor:
   ```
   $SPECROOT/custom/svdsk/config/service-desk-config.xml
   ```
4. Perform a search for the “ticket-summary” tag.
5. Replace the complete tag with the following tag:
   ```
   < ticket-summary />
   ```

6. Save the file.
7. Navigate to the OneClick Administration pages.
8. Click the Service Desk Configuration link from the left panel of the Administration page.
   The Service Desk Configuration administration page opens.
9. Click Save.
   The changes to the XML can now take effect. Ticket settings are restored to their defaults.

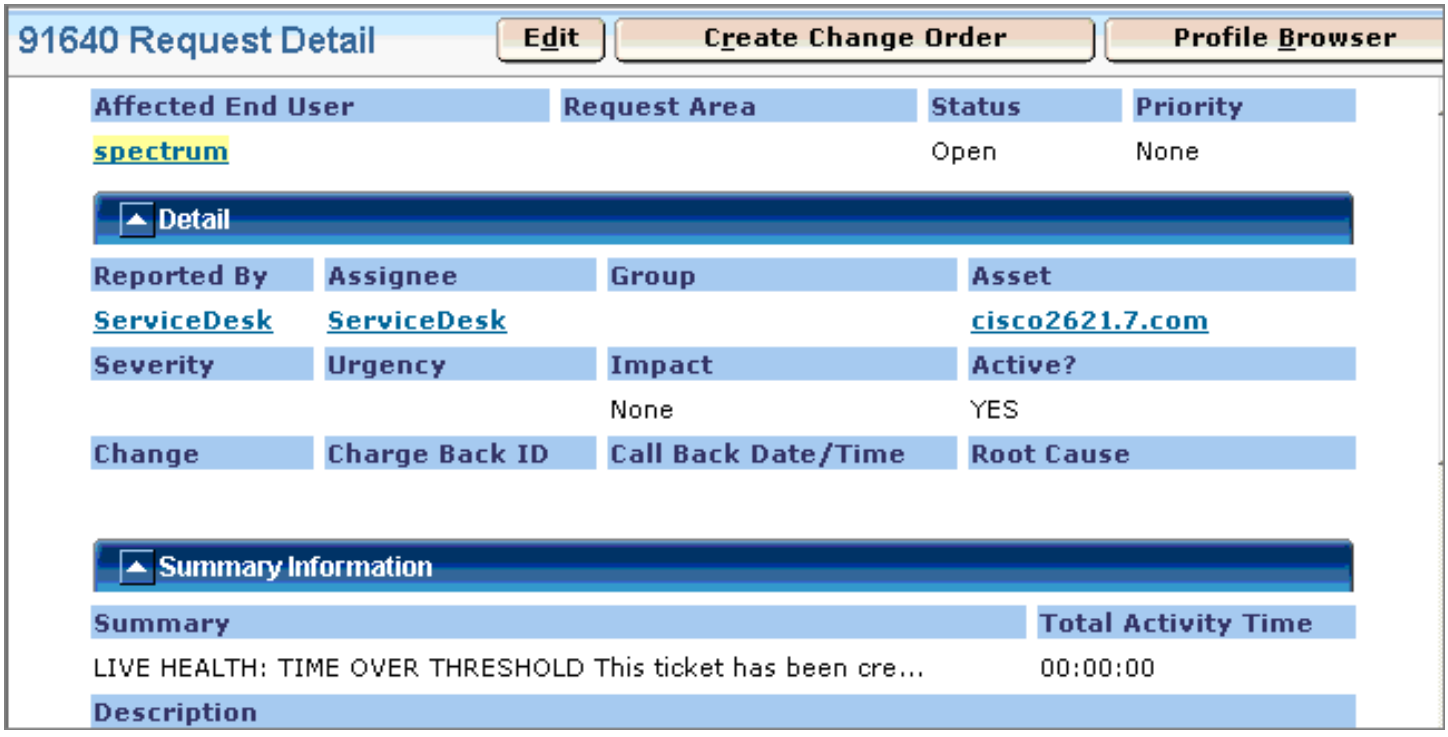
About Asset Assignment

You can configure the DX NetOps Spectrum and CA Service Desk Manager integration to assign an asset to a ticket submitted from OneClick. Automatically assigning assets to a service ticket helps CA Service Desk Manager users to work efficiently. Through a single click in CA Service Desk Manager, these users can view details about the device responsible for submitting the ticket, such as device attributes and a list of other tickets that are submitted for the same device. Assign an asset to the ticket so that the CA Service Desk Manager user can see that multiple trouble tickets are open for the same device.

When this option is enabled, only DX NetOps Spectrum device model types (0x1004b) or a port of a DX NetOps Spectrum device model type assigns assets to trouble tickets. Assets are assigned to a DX NetOps Spectrum device model type only when it, or one of its ports, submits a trouble ticket. DX NetOps Spectrum does not attempt to find or create an asset for a DX NetOps Spectrum device model before submitting a trouble ticket.

NOTE
You can still submit trouble tickets for other DX NetOps Spectrum model types, but no asset is assigned to them.

When OneClick submits a trouble ticket with an assigned asset, a CA Service Desk Manager user sees the Asset field that is populated with a link in the information about the trouble ticket, as shown:



When you click the linked device in the Asset field, details about the asset appear.

Assigning Assets in CA Service Desk Manager

This section describes asset assignment in CA Service Desk Manager. If CA Service Desk Manager is configured to run in ITIL mode, substitute the word “asset” with “configuration item” or “CI” throughout this section.

How Assets are Added to CA Service Desk Tickets

Assign an asset to a trouble ticket, which the DX NetOps Spectrum submits. Once the asset assignment is enabled, device details are accessible from the CA Service Desk trouble tickets that are submitted by DX NetOps Spectrum. The details that are provided after creating a trouble ticket vary, depending on the information available in DX NetOps Spectrum. Determine how the assets are added to trouble tickets to understand what to expect in the CA Service Desk tickets. If necessary, troubleshoot communication issues between CA Service Desk Manager and DX NetOps Spectrum.

Assets are added to CA Service Desk tickets according to the following workflow:

1. An alarm that is generated by a device model submits a trouble ticket to CA Service Desk Manager.
2. The ServiceDesk_Asset_ID attribute of the DX NetOps Spectrum device model is read.
3. If the ServiceDesk_Asset_ID attribute is not set for the DX NetOps Spectrum device model, a web service call to CA Service Desk is made to search for an asset that matches the device model.
   - If an asset is found, its identifier is returned to DX NetOps Spectrum and is written to the ServiceDesk_Asset_ID attribute of the device model.
   - If no match is found in CA Service Desk Manager, an asset that represents the DX NetOps Spectrum device model is created in CA Service Desk Manager. The identifier of the asset is returned to DX NetOps Spectrum and written to the ServiceDesk_Asset_ID attribute of the device model. As a result, future alarms on the model do not require additional web service calls to locate the asset.
4. The trouble ticket is created in CA Service Desk Manager using the asset identifier. This identifier is used in the future when creating trouble tickets for the same device.

Assign Assets in CA Service Desk Tickets

Assign the assets in the CA Service Desk tickets that OneClick creates. Once the asset assignment is enabled, device details are accessible from the CA Service Desk trouble tickets that DX NetOps Spectrum submits. Automatically assigning assets to a service ticket helps CA Service Desk Manager users perform efficient troubleshooting.

Follow these steps:

1. Navigate to the OneClick Administration pages:
   ```
   http://<oc> Web server/spectrum/admin/index.jsp
   ```
2. Click the Service Desk Configuration link from the left panel of the Administration pages.
   The Service Desk Configuration administration page opens.
3. Select the Enabled option in the Assign Assets/Configuration Items field.
4. Click Save.
   The following message appears:
   ```
   Successfully saved configuration to the service-desk-config.xml file.
   ```
   Assets are assigned with the CA Service Desk tickets that are submitted from DX NetOps Spectrum.

How Asset Details are Created in CA Service Desk Manager

Once the asset assignment is enabled, device details are accessible from the CA Service Desk trouble tickets submitted by DX NetOps Spectrum. The details provided upon creating a trouble ticket vary, depending on the information available in DX NetOps Spectrum. Knowing how asset information is gathered before adding them to trouble tickets can help you understand what to expect in the CA Service Desk tickets.

The process for creating asset details in CA Service Desk Manager can be customized. However, the following process explains the default method for locating and creating asset details in CA Service Desk Manager:

1. To locate the device details, DX NetOps Spectrum searches for the Model_Name of the device model, MAC_Address, and sysName attributes. It matches them to the name of the CA Service Desk asset, mac_address, and system_name attributes, respectively.

   NOTE
   If a device model attribute that is defined in the search has no value, it is excluded from the asset query.
2. To create an asset in CA Service Desk Manager for a DX NetOps Spectrum device model, DX NetOps Spectrum and CA Service Desk Manager perform the following steps:
   – DX NetOps Spectrum writes the Model_Name of the device model, Network_Address, MAC_Address, Serial_Number, and sysName attributes to the name of the CA Service Desk asset, alarm_id, mac_address, serial_number, and system_name fields, respectively.
   – CA Service Desk Manager uses "Discovered Hardware" for the asset class and "Device asset" for the asset description.

   NOTE
   If the model attribute defined in mapping has no value, the corresponding asset field is assigned an empty (blank) value.

Clear the Asset ID from All DX NetOps Spectrum Models

Clear out the ServiceDesk_Asset_ID attribute (0x12db9) for each device model. You can then search for and recreate assets in CA Service Desk Manager. For example, if you switch CA Service Desk Manager databases, these values must be repopulated to enable the integration with DX NetOps Spectrum. Clear this attribute by using the Attribute Editor in DX NetOps Spectrum or by using the DX NetOps Spectrum Command Line Interface (CLI). Using the CLI lets you create scripts to automate this procedure.

Follow these steps:

1. Connect to the CLI.
2. Run the following command on each device model that has the attribute set:

   ```
   update mh=<device model handle> attr=0x12db9,val=
   ```

   The ServiceDesk_Asset_ID attribute is cleared.

   NOTE

   For more information about the CLI, see the Command Line Interface section.

How to Customize Asset Assignment

By editing the asset mapping file, you can customize how CA Service Desk Manager assets from OneClick are found and created. Customization can help CA Service Desk Manager users to troubleshoot issues efficiently.

To change the attribute mapping between DX NetOps Spectrum models and CA Service Desk Manager assets, perform the following tasks:

1. Copy the service-desk-asset-mapping.xml and service-desk-asset-mapping.xsd files from *$SPECROOT*/tomcat/webapps/spectrum/WEB-INF/svdsk/config to *$SPECROOT/custom/svdsk/config.*
2. Modify the copy of the service-desk-asset-mapping.xml file, located in *$SPECROOT/custom/svdsk/config.*
   The XML file includes <asset> elements. An <asset> element defines the asset mapping for a DX NetOps Spectrum model type that is defined in the mtype_h attribute of the element. Each <asset> element has a Default child element and multiple <constant> and <mapping> children.

3. Apply your mapping changes.

Edit the Asset Mapping XML File

To change the attribute mapping between DX NetOps Spectrum models and CA Service Desk Manager assets, edit the service-desk-asset-mapping.xml file. Editing the file changes how asset information is found and which information about each asset is provided to CA Service Desk Manager.

Follow these steps:

1. Open the service-desk-asset-mapping.xml file in a text editor.
2. Locate the mtype_h attribute for the asset information you want to modify.

   NOTE
   Only DX NetOps Spectrum device model types assign assets to trouble tickets, and the mtype_h attribute for the device model type is "0x1004b."
3. Create an asset query by modifying the Default parameters.
4. Define the asset mapping for the asset creation. Modify the <constant> and <mapping> attributes of the selected asset.
   Note: When creating an asset, the name and class attributes are always required. Additionally, specify any other asset attributes that are stored in the "nr Object" in CA Service Desk Manager. For a complete list of attributes that are defined in the "nr Object," see the CA Service Desk Manager section.
   The information to be recorded in the CA Service Desk trouble ticket is defined.
5. Save the file.
   Your asset mapping file edits are complete.

   NOTE
   Your changes do not go into effect immediately. Apply the changes in DX NetOps Spectrum after your modifications are complete.

Customize Asset Search and Assignment

The service-desk-asset-mapping.xml file is used to customize how asset attributes are mapped between DX NetOps Spectrum models and CA Service Desk assets. You can troubleshoot issues with the help of this customization.

The XML file is located in the *$SPECROOT*/tomcat/webapps/spectrum/WEB-INF/svdsk/config folder of any OneClick server. This file includes <asset> elements. Each <asset> element has a Default child element and multiple <constant> and <mapping> children. The following sections describe the XML syntax, Elements, and Notes that are required to customize asset search and assignment.

XML Syntax

The XML file has the following basic syntax:

```
- <asset mtype_h="asset_ID">
  - <search>
    - <and>
      - <equals>
        <sd-attribute>SD_name</sd-attribute>
        <model-attribute>SPEC_name_attribute</model-attribute>
        </equals>
      - <equals>
        <sd-attribute>SD_name2</sd-attribute>
        <model-attribute>SPEC_name_attribute2</model-attribute>
        </equals>
      </and>
    </search>
```

```
 - <constant>
  <sd-attribute>constant</sd-attribute>
  <value>constant_value</value>
  </constant>
- <mapping>
  <sd-attribute>mapping_name</sd-attribute>
  <model-attribute>mapping_name_attribute</model-attribute>
  </mapping>
</asset>
```

Elements

The XML file syntax includes the following elements:

- **<asset mtype_h="*asset_ID*">**
  Defines the asset mapping for a DX NetOps Spectrum model type that is defined in the mtype_h attribute (*asset_ID*) of an element. The only supported <asset> element is for the device model type, which is "0x1004b." This element includes the following child elements:
  - Default
  - <constant>
  - <mapping>

  Example: <asset mtype_h="0x1004b">
- **Default**
  Defines the query to locate an asset in CA Service Desk for the DX NetOps Spectrum model. In this element you can define and, or, and equals logic to your asset query by embedding the <and>, <or>, and <equals> child elements respectively. The Default element can contain multiple child elements. Their hierarchy determines the order of operations and placing parenthesis around logical elements when required.
  - **<and>**
    Indicates that this attribute is required when searching for an asset match between CA Service Desk Manager and DX NetOps Spectrum. The <and> element appears between the child elements of the Default element.
  - **<or>**
    Indicates that this attribute is optional when searching for an asset match between CA Service Desk Manager and DX NetOps Spectrum. The <or> element appears between the child elements of the Default element.
  - **<equals>**
    Defines the CA Service Desk Manager and DX NetOps Spectrum attribute mapping for your search. Each <equals> relationship that you define must include the following child elements:
    - <sd-attribute>*SD_name*</sd-attribute> -- Defines that asset attribute (*SD_name*) for which the query searches. You can use any asset attribute that is defined in the "nr Object" in CA Service Desk Manager for your asset query.
    - <model-attribute>*SPEC_name_attribute*</model-attribute> -- Defines the model attribute (*SPEC_name_attribute*) text to match when searching.

      NOTE
      If a model attribute is not set, the parameter is not used in the query. You can verify the complete list of *nr Object* attributes that is available. For more information, see the CA Service Desk Manager section.
- **<constant>**
  Assigns a specified value (*constant_value*) to the asset attribute (*constant*) when creating an asset, regardless of the attribute values of the model. You can define multiple <constant> rules for the asset creation. Each <constant> element that you define must include the following child elements:
  - <sd-attribute>*constant*</sd-attribute>
  - <value>*constant_value*</value>
- **<mapping>**

Assigns the value of the model specified attribute (*mapping_name_attribute*) to the specified asset attribute (*mapping_name*) when creating an asset. You can define multiple <mapping> rules for asset creation, but they must appear after all <constant> rules. Each <mapping> element includes the following child elements:

- <sd-attribute>*mapping_name*</sd-attribute>
- <model-attribute>*mapping_name_attribute*</model-attribute>

Notes

When creating an asset, the name and class attributes are always required. In addition, you can specify any other asset attributes that are stored in the "nr Object" in CA Service Desk Manager. For example, you can include the following CA Service Desk attributes in the <sd-attribute> element:

- **class**
  (Required) Determines the asset class in CA Service Desk Manager. The asset class must exist before creating an asset.
- **name**
  (Required) Defines the asset name.
- **description**
  Sets the Notes section text.
- **alarm_id**
  Defines the IP address of the asset.
- **mac_address**
  Defines the MAC address of the asset.
- **serial_number**
  Defines the serial number of the asset.
- **system_name**
  Defines the hostname of the asset.

OneClick does not create objects in CA Service Desk tables besides the nr Object. Therefore, certain attributes that are references to other objects (such as the vendor) must be created manually in CA Service Desk Manager. The object identifier is then used to assign each attribute an asset mapping. For a complete list of nr Object attributes, see the CA Service Desk Manager section.

NOTE
You can use DX NetOps Spectrum search criteria to identify the existing asset. If a search fails to find the asset with the given search criteria, it results in a "No match found" message. A new asset ID is created based on the specified parameters. However, if you change the properties, such as host name, serial number, MAC address, DNS name, or asset label, DX NetOps Spectrum does not create a new asset ID. For more information, see Asset Matching Logic and About Asset Assignment.

Example: Create a Custom Asset Query

This example shows how to locate an asset in CA Service Desk Manager that has a name attribute matching the model name, or an alarm_id attribute matching the model network address (IP address) and a mac_address attribute that matches the model MAC address. The logic here can be represented in the following logical statement:

```
(name='<MODEL_NAME>') OR
(
(alarm_id='<NETWORK_ADDRESS>') AND
(mac_address='<MAC_ADDRESS>')
)
```

Given that the <MODEL_NAME>, <NETWORK_ADDRESS>, and <MAC_ADDRESS> model attributes are 0x1006e, 0x1027f, and 0x110df respectively, modify the Default element for the asset as follows:

```
<search>
  <or>
```

```
    <equals>
      <sd-attribute>name</sd-attribute>
      <model-attribute>0x1006e</model-attribute>
    </equals>
    <and>
      <equals>
        <sd-attribute>alarm_id</sd-attribute>
        <model-attribute>0x1027f</model-attribute>
      </equals>
      <equals>
        <sd-attribute>mac_address</sd-attribute>
        <model-attribute>0x110df</model-attribute>
      </equals>
    </and
  </or>
</search>
```

Example: Create a Custom Attribute Mapping for Asset Creation

In this example, you want all device model types in DX NetOps Spectrum to assign the model attributes MODEL_NAME and MAC_ADDRESS to asset attribute names and a mac_address, respectively. You also want to assign each asset description to read "Device modeled by SPECTRUM OneClick." And you want the class to be "SPECTRUM Device" (previously created manually in CA Service Desk Manager and defined as cr:9).

The following code shows how to configure these parameters by defining two <constant> and two <mapping> elements to the device model asset mapping. Assume that the model attributes for MODEL_NAME and MAC_ADDRESS are 0x1006e and 0x110df, respectively:

```
<asset mtype_h="0x1004b">
  ...
  <constant>
    <sd-attribute>description</sd-attribute>
    <value>Device modeled by SPECTRUM <oc></value>
  </constant>
  <constant>
    <sd-attribute>class</sd-attribute>
    <value>cr:9</value>
  </constant>
  <mapping>
    <sd-attribute>name</sd-attribute>
    <model-attribute>0x1006e</model-attribute>
  </mapping>
  <mapping>
    <sd-attribute>mac_address</sd-attribute>
    <model-attribute>0x110df</model-attribute>
  </mapping>
</asset>
```

Asset Reporting Customization

When integrating CA Service Desk Manager and DX NetOps Spectrum, you can modify the behavior of finding and creating the CA Service Desk Manager assets. You can modify this behavior when creating a trouble ticket from OneClick. This customization is done by changing the attribute mapping between DX NetOps Spectrum models and CA Service Desk Manager assets.

Customizing the asset reporting lets you prioritize the information that is used to identify a device. You can determine which information to record within CA Service Desk Manager. Customization can enhance the efficiency and reporting capabilities of individual CA Service Desk Manager users.

Apply Asset Mapping Changes

When you modify the asset mapping file, your changes do not immediately take effect when you save the file. Apply the changes in DX NetOps Spectrum after your file modifications are complete. Restarting the OneClick clients is not required.

Follow these steps:

1. Navigate to the OneClick Administration pages:
   `http://<oc> Web server/spectrum/admin/index.jsp`
2. Click the Service Desk Configuration link from the left panel of the Administration pages.
   The Service Desk Configuration administration page opens.
3. Click the Reload Asset/CI Mapping button.
4. Click Save.
   The following message appears:
   `Successfully saved configuration to the service-desk-config.xml file.`
   Your asset mapping changes are now applied.

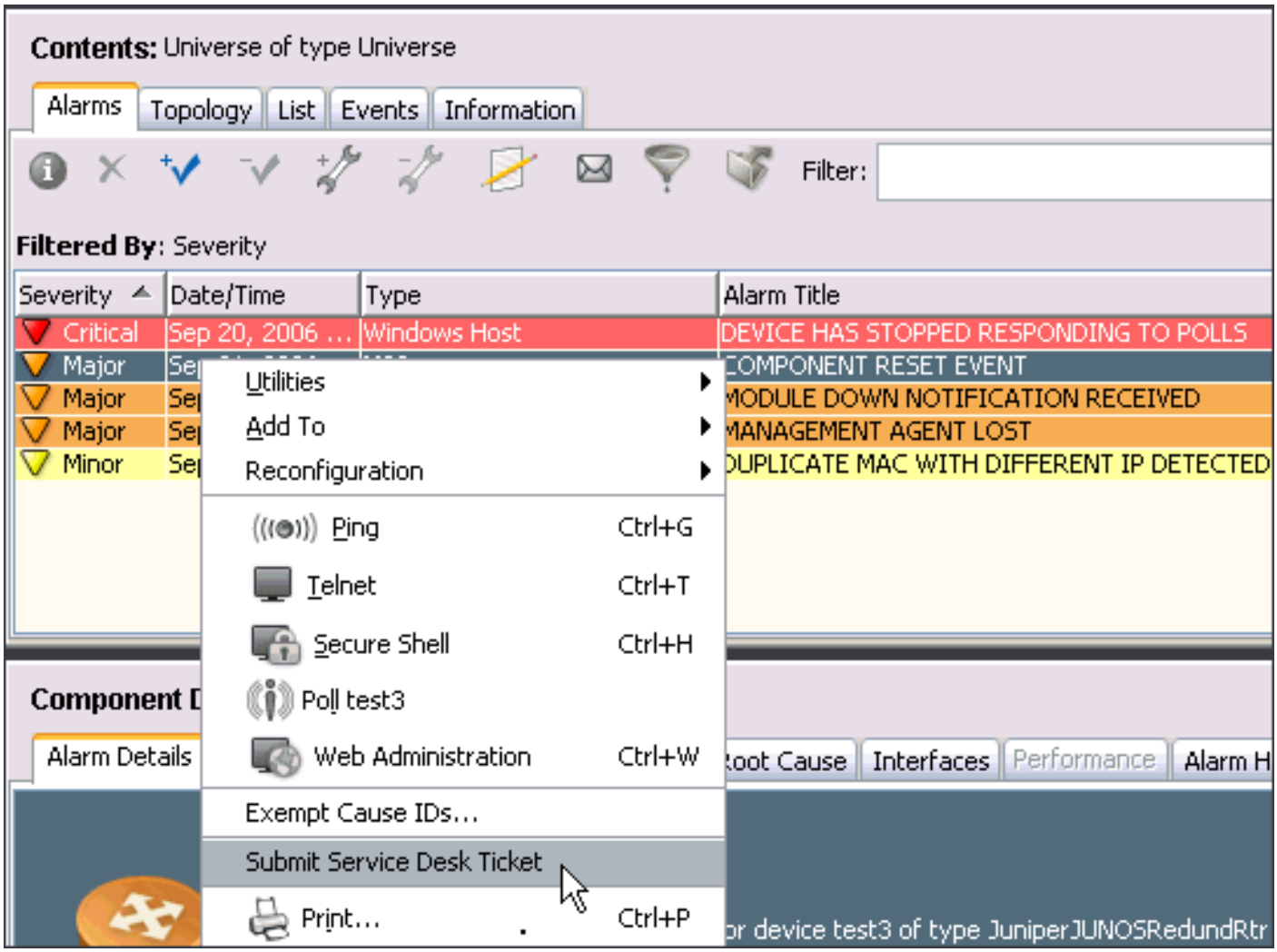
Using the Integration

Submit CA Service Desk Tickets from the OneClick Console Manually

You can manually create a CA Service Desk ticket from OneClick.

Follow these steps:

1. In the OneClick Console, right-click an alarm that you want to submit to CA Service Desk.
2. Select Submit Service Desk Ticket, as shown in the following figure. The ticket is sent to CA Service Desk Manager. The alarm is updated with the Service Desk Trouble Ticket ID. The ID provides a link from the DX NetOps Spectrum alarm back to the ticket in CA Service Desk Manager.



NOTE
After an alarm is submitted to CA Service Desk Manager, right-click the alarm. Select “Service Desk Ticket Information” to view the ticket details.

View CA Service Desk Tickets from the OneClick Console

After a CA Service Desk ticket is created for a DX NetOps Spectrum alarm, you can open the ticket in CA Service Desk Manager from within OneClick. Alarms that are associated with CA Service Desk Manager trouble tickets contain a trouble ticket ID link. Once an alarm has been submitted to CA Service Desk Manager, you can view detailed information about the ticket in OneClick.

Follow these steps:

1. In the OneClick Console, right-click an alarm that has an associated CA Service Desk ticket.
2. Select Service Desk Ticket Information. The Service Desk Request Detail window opens for that ticket ID.

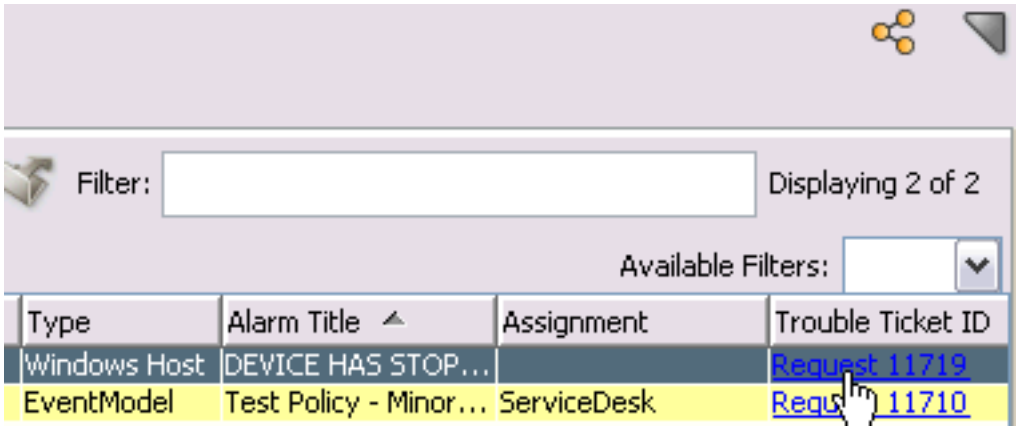
You can also view CA Service Desk tickets by adding the Trouble Ticket ID column to the OneClick Alarms tab view.

Follow these steps:

1. In the OneClick Console, locate an alarm with an existing CA Service Desk ticket.

2. Configure the OneClick client to display the Trouble Ticket ID column in the Alarms tab view as follows:
   a. Right-click the top of any column in the Alarms tab to launch the Table Preferences dialog.
   b. In the Columns tab of the dialog, select Trouble Ticket ID.
   c. Click OK.

   NOTE
   For more information about customizing columns, see Using OneClick.
3. Click the Trouble Ticket ID of the alarm in which you are interested.



The Service Desk Request Detail window opens for that ticket ID.

Using SANM with CA Service Desk Manager

This section provides instructions for configuring and using Alarm Notification Manager (SANM) with the and CA Service Desk Manager integration**.**

NOTE

This section assumes basic knowledge of the use of the AlarmNotifier and SANM functionality. For more information about SANM, see the Alarm Notification Manager section.

CA Service Desk Manager and SANM Overview

The Alarm Notification Manager (*SANM*) is a component that enhances the functionality of alarm-processing applications. You can take advantage of the SANM alarm filtering capabilities to configure the alarms that create CA Service Desk Manager trouble tickets. To enable this feature, deploy special Alarm Notifier scripts that create, clear, and update CA Service Desk Manager tickets. The following scripts let you customize alarm parameters:

- ServiceDeskSetScript
- ServiceDeskClearScript
- ServiceDeskUpdateScript

Best Practices for Automatic Trouble Ticket Creation Using OneClick or SANM

When deploying the CA Service Desk Manager integration, an important best practice is to configure alarm creation using only *one* of the following methods:

- Use the Selected Alarm Types filter in OneClick. To configure the Selected Alarm Types filter on the Service Desk Configuration Administration page to create alarms, take the following steps:
  a. Verify that the Alarm Notifier integration component for CA Service Desk Manager (SDNotifier) is not enabled.

b. Follow the instructions to select alarm types for which to create alarms.

- Configure the CA Service Desk Manager integration Alarm Notifier component and SANM to generate alarms. Take the following steps:
  a. Verify that the Selected Alarm Types filter of the Service Desk Configuration Administration page is set to NONE.
  b. Configure the SANM CA Service Desk integration component to generate CA Service Desk trouble tickets by completing these tasks:
     a. Configure the Alarm Notifier Integration component for CA Service Desk Manager.
     b. Configure SANM to create CA Service Desk tickets.

     WARNING
     When using SANM to generate trouble tickets, first set the Selected Alarm Types parameter on the Service Desk Configuration Administration page to NONE. Be sure to save the change.

If the and CA Service Desk Manager integration is configured to generate alarms using both of these methods, the integration can create unwanted, redundant trouble tickets.

Configuring the AlarmNotifier Integration Component for CA Service Desk Manager

The DX NetOps Spectrum and CA Service Desk Manager integration includes the Alarm Notification Manager support files that are required to configure the SANM functionality. These files are saved in the <*$SPECROOT*>/Notifier/ and <*$SPECROOT*>/Notifier/sd_notifier/ directories.

Follow these steps:

1. Copy the <*$SPECROOT*>/Notifier/AlarmNotifier.exe file to the <*$SPECROOT*>/Notifier/sd_notifier/ directory, renaming the file as <*$SPECROOT*>/Notifier/sd_notifier/SDNotifier.exe.

   NOTE
   The executable files that are referenced in this procedure (for example, SDNotifier) do not have an extension on Linux systems.
2. Copy the <*$SPECROOT*>/Notifier/.alarmrc file to the <*$SPECROOT*>/Notifier/sd_notifier/directory as .alarmrc.
3. In the <*$SPECROOT*>/Notifier/sd_notifier/.alarmrc file, modify the Set script, Clear script, and Update script entries to point to the CA Service Desk-specific scripts found in <*$SPECROOT*>/Notifier/sd_notifier. The actual scripts are named ServiceDeskSetScript, ServiceDeskClearScript, and ServiceDeskUpdateScript.

   NOTE
   Use caution and look for relative pathnames to these scripts. Verify that the pathnames point to the correct directory.
4. In the <*$SPECROOT*>/Notifier/sd_notifier/.alarmrc file, change the application name entry to 'SDNotifier'.
5. Run <*$SPECROOT*>/Notifier/sd_notifier/ServiceDeskIntegrationSetup.exe.
6. Run SDNotifier.exe.
7. Launch the SANM Policy Manager.
8. Create a policy that uses your preferred filters. For more information, see Alarm Notification Manager .
9. Create an application that is named "SDNotifier", and apply the policy to your application.

Configure SANM to Create CA Service Desk Tickets

A few more steps are required to support the SANM functionality in the DX NetOps Spectrum and CA Service Desk Manager integration. Configure SANM to create tickets in CA Service Desk Manager. You can add and modify alarm policies and filters in OneClick.

Follow these steps:

1. Start an instance of AlarmNotifier and verify that it is using the ServiceDeskSetScript.

2. Apply a filter policy to your AlarmNotifier instance.

   NOTE

   Note: Whenever an alarm matches the applied filter, a Service Desk ticket is created.

   For information about applying a filter policy, see Alarm Notification Manager (SANM) .

Provide CA Service Desk Ticket Request Number in SDNotifier Output

You can configure the CA Service Desk Manager integration to provide the CA Service Desk ticket request number in the SDNotifier output. Edit the .alarmrc file and the ServiceDeskUpdateScript to include more instructions.

Follow these steps:

1. Navigate to the sd_notifier directory.
2. Edit the .alarmrc file to add the following new lines:

```
EXTRA_ATTRS_AS_ENVVARS=0x12022
UPDATE_ATTRS=0x12022
```

3. Add the following lines to the ServiceDeskUpdateScript:
   On Windows:

```
ServiceDesk_Request=$SANM_0X12022
echo "ServceDesk_Request:
echo $SANM_0X12022 | cut -f3 -d'>'
```

   On Linux:

```
ServiceDesk_Request=$SANM_0x12022
echo "ServceDesk_Request:
echo $SANM_0x12022 | cut -f3 -d'>'
```

Example: CA Service Desk Ticket Request Number in SDNotifier Output

The following Ticket Request Number is a sample of successful output:

```
Alarm Notification from SPECTRUM
Alarm UPDATED:
Date:              11/10/2006
Time:              11:24:16
DeviceType:        6G306-06
Mtype:             6G3xx
ModelName:         1.2.4.5
AlarmID:           16600
ServiceDesk_Request:
Request 283</a
Severity:          MINOR
ProbableCauseID:   1030a
RepairPerson:
AlarmStatus:
SpectroSERVER:     ratchet.ca.com
Landscape:         0x1e00000
ModelHandle:       0x1e0004d
ModelTypeHandle:   0x3d20001
IPAddress:         1.2.4.5
SecurityString:
AlarmState:        NEW
Acknowledged:      FALSE
UserClearable:     TRUE
```

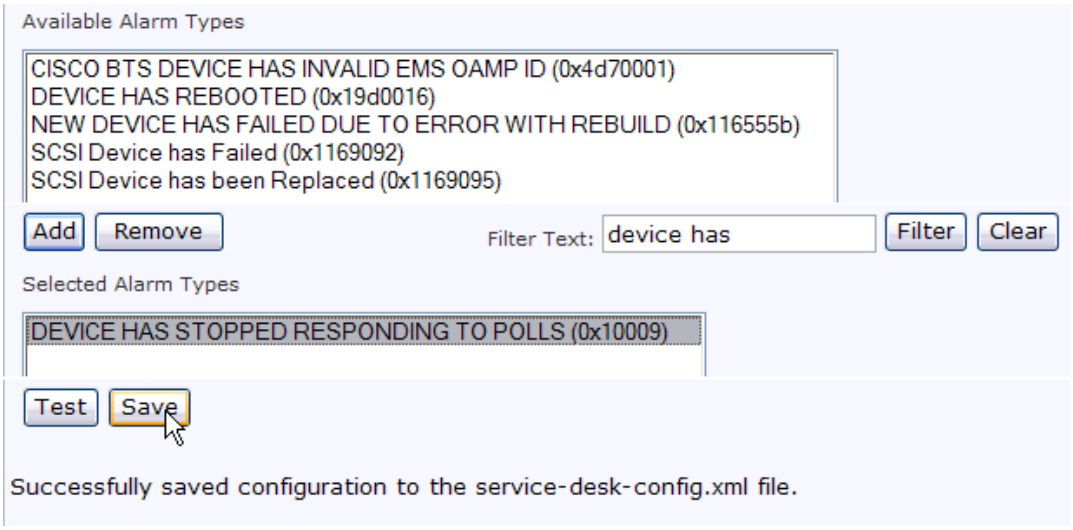
```
Location:          6C107
AlarmAge:          0
NotificationData:

ProbableCause:     No Associated Text

EventMessage:      No Associated Event Message
```

References

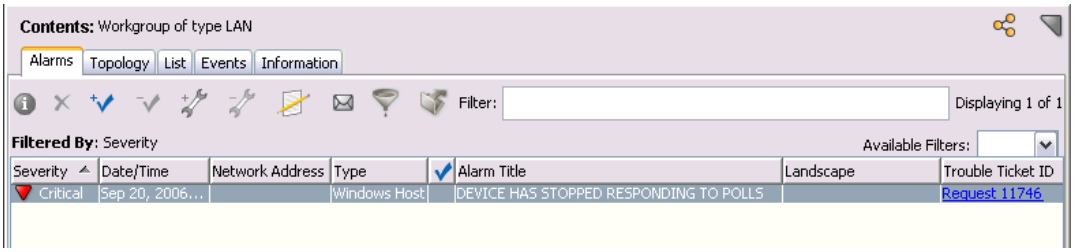
Example Create CA Service Desk Tickets Automatically for a Single Alarm Type

This section describes a simple example of how to configure the integration to create tickets automatically.

The following figure shows the DX NetOps Spectrum alarm type DEVICE HAS STOPPED RESPONDING TO POLLS (0x10009) has been added to the Selected Alarm Types list from the Available Alarm Types list and that this configuration change has been saved:

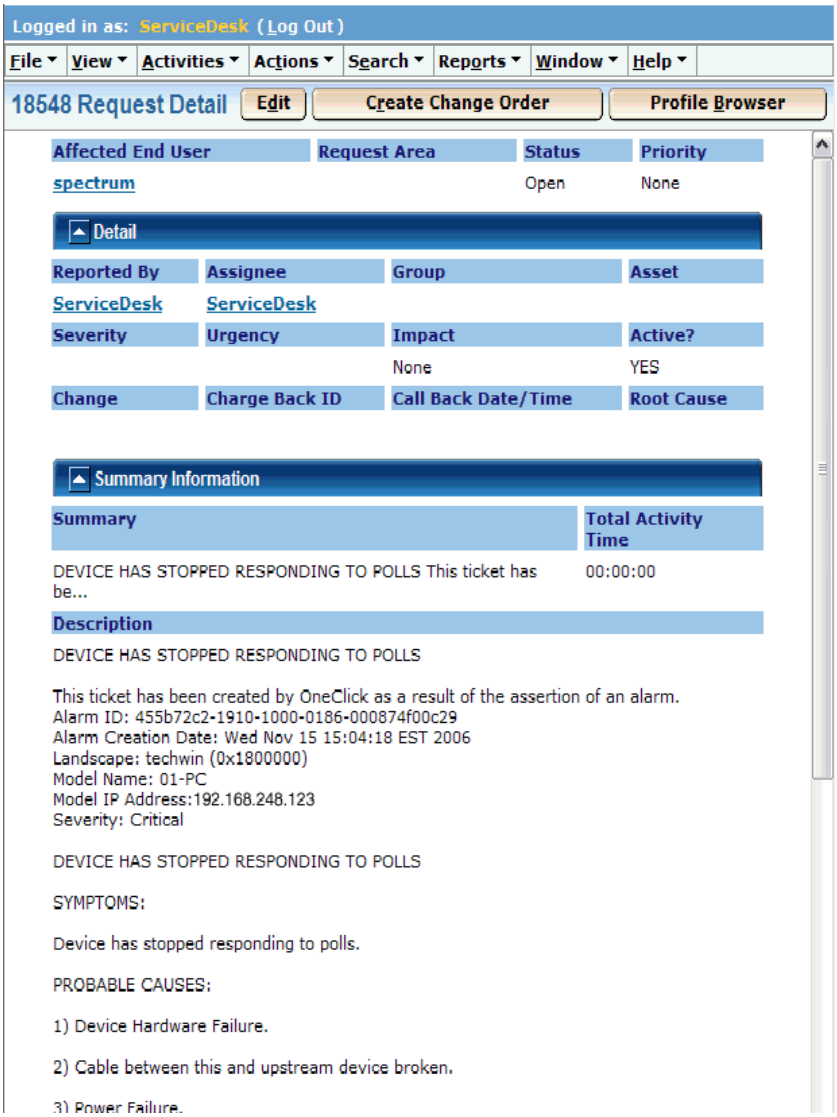


In this example, DX NetOps Spectrum alarms of type DEVICE HAS STOPPED RESPONDING TO POLLS (0x10009) generate corresponding CA Service Desk tickets. The following figure shows an instance of a DX NetOps Spectrum alarm of type 0x1009. The Trouble Ticket ID column contains a link to the CA Service Desk ticket that was automatically created for this alarm.



Click the Trouble Ticket ID link to open the web login page for your CA Service Desk Manager server in a browser.

After you log in to your CA Service Desk Manager server, the Service Desk Request Detail page for the OneClick ticket opens, as shown:



Example Using Custom Keywords for CA Service Desk Notifications

You can customize the keywords that CA Service Desk Manager uses for notifications of ticket actions. Custom keywords must be configured both in CA Service Desk Manager and in DX NetOps Spectrum. The following example explains how to configure both CA Service Desk Manager and DX NetOps Spectrum to use “Fixed” as the custom keyword for the ticket closed action.

Follow these steps:

1. From the CA Service Desk Manager server home page, select the Administration tab.
2. Expand the Notifications folder, and select Activity Notifications.
   The Activity Notification List opens.
3. Select the Close activity.
   The Close Activity Notification Detail dialog opens.
4. Click Edit.

5. Take the following actions:
   a. Click the Notification Rules tab and then click the name of the notification rule that is being used by the transfer activity.
   b. Click the message template that is used by the transfer activity, and click Edit.
6. In the Notification Message title field, change the assignment information from (for example):

   ```
   Request @{call_req_id.ref_num} Closed
   ```

   to

   ```
   Request @{call_req_id.ref_num} Fixed
   ```

   Fixed
   Is the keyword that you want to use for the ticket closed action.

   NOTE
   Do not remove the information in the 'Description: @{call_req_id.description}' field. DX NetOps Spectrum uses this information to associate alarms in DX NetOps Spectrum with the CA Service Desk ticket. Without this information, alarms are not cleared in DX NetOps Spectrum when a CA Service Desk ticket is closed.
7. Click Save in the Close Update Activity Notification window.
8. Close the Close Activity Notification Detail window.
9. Log in to your CA Service Desk Manager server host computer and navigate to the <*Service_Desk_Installation_directory*>/bin directory.
10. Change the Close keyword in DX NetOps Spectrum using one of the following methods:
   – Run the <*Service_Desk_Installation_directory*>/bin/OneClickIntegrationSetup(.exe) configuration program and enter *Fixed* at the Cloase keyword prompt.
   – Manually edit the value for the Close keyword in the <*Service_Desk_Installation_directory*>/bin/oc-integration.cfg configuration file to reflect the *Fixed* keyword for the close action.

   WARNING
   If the keywords that are configured in the DX NetOps Spectrum integration for the Close and/or Transfer actions do not match the keywords set in the CA Service Desk Notification Message Title for those actions, closing or transferring CA Service Desk tickets that are associated with DX NetOps Spectrum alarms does not clear or assign (respectively) the alarm to a troubleshooter in DX NetOps Spectrum.

Troubleshooting with Service Desk

Troubleshoot the Ticket Creation Rules

Sometimes CA Service Desk tickets generated by DX NetOps Spectrum assign the problem type and affected end user, or both. Tools in DX NetOps Spectrum can help you identify problems with ticket creation and correct them.

Ticket rules must be loaded through the CA Service Desk Integration page or by a Tomcat restart. Then the object that manages the ticket rules displays the actual rules that are loaded in memory in the proper order.

Follow these steps:

1. Navigate to the OneClick Administration Debugging Page.
2. Select Context Factory as the option on the left menu.
3. Select **com.aprisma.sd.ServiceDeskTicketConfig**

You can also see how the rules are analyzed and examine the actual web service call that is made to CA Service Desk Manager when a ticket is created. Set the CA Service Desk Integration debugging level to MAX. This debugging level lets you see whether a rule with a higher priority is being unexpectedly applied to your ticket. You can also determine whether the actual web service call results in the expected problem type and affected end-user values.

Troubleshooting Redundant Trouble Ticket Issues

Symptom:

We are seeing two CA Service Desk trouble tickets that are generated for each alarm in OneClick.

Solution:

Duplicate tickets indicate a misconfiguration of the DX NetOps Spectrum and CA Service Desk Manager integration. You probably configured the alarm generation using both the OneClick Selected Alarm Types filter and a SANM policy.

If both OneClick and SANM are configured to generate alarms, one of the trouble tickets has a ticket number that matches the Trouble Ticket ID attribute of the associated alarm in DX NetOps Spectrum. The ticket number of the redundant trouble ticket does not match the ticket number of the Trouble Ticket ID attribute of any alarm in DX NetOps Spectrum.

Even though the redundant trouble ticket does not appear to be associated with a particular alarm, closing it clears the alarm that created it. For the steps to ensure that duplicate tickets are not created by SANM and OneClick, see Best Practices for Automatic Trouble Ticket Creation.

Service Desk Tickets Not Created After Switching Servers

Symptom:

I recently switched my DX NetOps Spectrum and CA Service Desk Manager integration to use a new CA Service Desk Manager server. I restarted Tomcat, but now DX NetOps Spectrum is unable to create tickets in CA Service Desk Manager. The error in my tomcat log is as follows:

```
Oct 15, 2007 1:24:24 PM (AlarmNotifier) (SDIntegration) - SDAlarmHandler -
received alarm SET
Oct 15, 2007 1:24:24 PM (AlarmNotifier) (SDIntegration) - SDAlarmHandler -
attempting to create ticket for alarm 4713a247-0167-1000-0183-0080102af61e
Oct 15, 2007 1:24:24 PM (AlarmNotifier) (SDIntegration) - SDSetAction -
gathering info to create ticket for alarm 4713a247-0167-1000-0183-0080102af61e
Oct 15, 2007 1:24:24 PM (AlarmNotifier) (SDIntegration) - ServiceDesk_Asset_ID
attribute found in <sp>: nr:2929BAB6C548A34FA64FB06A5811A414
Oct 15, 2007 1:24:31 PM - Error occurred while attempting to create a ticket in
Service Desk. Internal err with update_lrel with handle
nr:2929BAB6C548A34FA64FB06A5811A414: NOT FOUND
```

When I try to submit an alarm manually, a "creation failed" message appears. Why can I not create a CA Service Desk ticket from DX NetOps Spectrum?

Solution:

If you enable Assign Assets in OneClick, and then switch to a new CA Service Desk Manager server, verify that the new server uses the same database as the old server. If you switch to a server that uses a different database, CA Service Desk Manager cannot create the ticket. The reason is that the DX NetOps Spectrum model in the original database is aware of the assets that were created for it in CA Service Desk Manager. The new database does not have information about that asset, causing an error when CA Service Desk Manager attempts to assign the asset from the DX NetOps Spectrum alarm.

To continue assigning assets after you switch to a new CA Service Desk Manager database, take one of the following actions:

- Configure the new CA Service Desk Manager server to use the original database.
- Manually clear the ServiceDesk_Asset_ID fields of all the DX NetOps Spectrum models.

Integration with Service Desk

In this release, DX NetOps Spectrum enables integration with third-party service desks (apart from the existing CA Service Desk Manager integration), by adopting an enhanced integration framework acting as an intermediary between DX NetOps Spectrum and the Service Desk(s) which is part of the DX NetOps Spectrum install package. The installation is part of the Spectrum install and the default Integration Parameters are auto-configured, when you enable this integration.

WARNING

We recommend using this integration capability only for fresh installations of DX NetOps Spectrum.

You can select and configure the Service Desk application as well as enable the integration with DX NetOps Spectrum from the OneClick Administration Pages.

For more information, see Select Service Desk and enable integration on the DX NetOps Spectrum OneClick Server.

The new integration enables better communication between the operations center and the service desk, helping automate an efficient workflow process, reduce mean-time-to-resolution, and, lower overall management costs. This integration provides automated and real-time updates as problems are triaged and resolved. We get real-time visibility into the current status of problem conditions. This integration framework also supports policy-based automated alarm ticket submission, and on-demand ticket submission from the DX NetOps Spectrum OneClick.

By default, DX NetOps Spectrum uses polling mechanism to get the ticket update (Assignee and Status) information from the configured Service Desk, based on your configuration in **OneClick Administration** Page > **Service Desk Configuration Integration Parameters** section > **Polling** field.

NOTE

You can configure notifications to OneClick from ServiceNow. See Configuring OneClick notifications from ServiceNow.

WARNING

The enhanced integration framework for integrating DX NetOps Spectrum with Third-party Service Desk applications is not supported on Solaris Operating System.

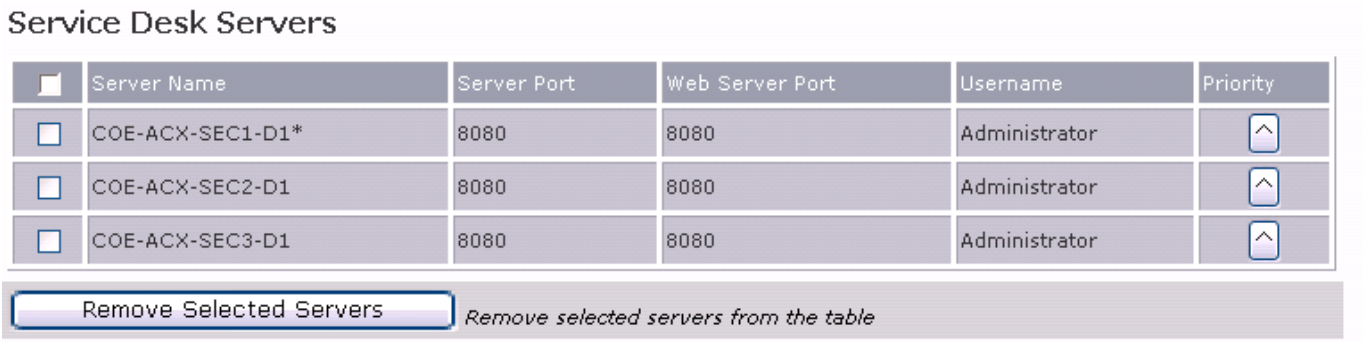
10.3 allows you to integrate with the following Service Desk applications, in addition to CA Service Desk Manager:

- BMC Remedy ITSM
- ServiceAide
- HP ServiceManager
- ServiceNow

WARNING

You cannot configure any other Service Desk application if you have already integrated CA Service Desk Manager with DX NetOps Spectrum on the same OneClick Server.

NOTE
If you have already integrated DX NetOps Spectrum with CA Service Desk Manager, you need to disable the integration and remove the entries, from the OneClick Administration Page > Service Desk Configuration > Service Desk Servers table, as displayed below.



Currently, the integration supports incident creation, update and assignment from both the Service Desk interface as well as the DX NetOps Spectrum OneClick Console.
The following features are supported:

- Incident Creation / Submit Service Desk Ticket
- Launch in Context for Service Desk ticket created in Spectrum
- Incident Assignment (Add/Update Assignee)
- Close / Resolve Incident / Tickets. (Reflects immediately on DX NetOps Spectrum, fetches the data from the Service Desk based on polling interval)

Reproduce-*-

Installing and Configuring the Integrations

To install and configure the DX NetOps Spectrum and Service Desk applications/ MDRs integration successfully, complete the following procedures:

1. Verify integration system requirements for OneClick and Service Desk servers.
2. Select Service Desk and enable integration on the DX NetOps Spectrum OneClick Server
3. Configuring the Service Desk/ MDR Application server.

Verify integration system requirements for OneClick and Service Desk servers.

Before you begin, verify that your DX NetOps Spectrum and Service Desk/ Incident Manager servers meet the following requirements:

- **DX NetOps Spectrum OneClick server software** -- CA Spectrum 10.3. For system requirements, see System Requirements for Installing DX NetOps Spectrum.
- **Service Desk software** – For the list of supported Service Desk/ MDR software and compatible versions, see the following table:

Service Desk / Incident Management Application	Supported Version(s)
BMC Remedy ITSM	8.1, 9.1
ServiceAide	Goldfish
HP ServiceManager	9.32
ServiceNow	See Integration Compatibility

- **Supported platform / Operating Systems**

WARNING

The enhanced integration framework for integrating DX NetOps Spectrum with Third-party Service Desk applications is not supported on Solaris Operating System.

- Ensure that during Service Desk/MDR configuration, User/Troubleshooter/ Assignee should exist on both applications for the assign option to work from both ends.

NOTE

This integration will support only AdoptOpenJDK 8u212 or higher versions of Java.

WARNING

The DX NetOps Spectrum integration with Third-party Service desk(s) is configured and enabled from the DX NetOps Spectrum OneClick Administration> Service Desk Configuration Page.
The following section specifies any special considerations/ exceptions that are applicable for configuring the integration on the Service Desk server side.

Configuring the Service Desk/ MDR Application server

The enhanced integration framework which acts as the intermediary between DX NetOps Spectrum and Service Desk(s) is part of the DX NetOps Spectrum installation package, and installs the required components and configuration during installation. However, for the following Service Desk applications, there are certain considerations and exceptions, which you need to consider:

HP Service Manager

To open an HP Service Manager ticket URL from DX NetOps Spectrum (Launch-In-Context feature), you will need to update/make certain configuration changes on the server where HP Service Manager is deployed/installed.

To launch the HP Service Manager ticket URL from DX NetOps Spectrum; follow these steps:

1. On the machine where HP Service Manager is deployed, stop the Tomcat server.
2. Navigate to the following location *:\apache-tomcat-7.0.67\webapps\webtier-9.32\WEB-INF*
3. Open the ***web.xml*** file.
4. In the web.xml file, do one of the following:
   - Search for the 'querySecurity' parameter and if the parameter exists ensure that the <parameter-value> = false.
   - If the parameter does not exist, follow these steps to add the parameter:

```
<init-param>
      <param-name>querySecurity</param-name>
      <param-value>false</param-value>
    </init-param>
```

5. Save the web.xml file
6. Restart the Tomcat server.

BMC Remedy ITSM

To enable the communication between DX NetOps Spectrum and BMC Remedy ITSM, copy the SDK jar files from the BMC Remedy System to the DX NetOps Spectrum library.

The integration user needs to have the privileges for the following roles in the BMC Remedy ITSM > Roles list:

License Type: Fixed
Application Permission: Incident Master, Incident User

Follow these steps:

1. In the machine where BMC Remedy ITSM is installed, navigate to:
   `\\BMC Software\ARSystem\Arserver\api\lib`
2. Copy the following SDK jar files:
   `arapi8*.jar`
   `arutil81*.jar`
3. Get the following jar file from the location where BMC Remedy is installed or you can download it from the internet.
   `log4j*.jar`
4. Save all the jar files in the following location:
   `\\win32app\Spectrum\tomcat\webapps\ca-nim-sm\WEB-INF\lib`
5. Restart the Tomcat server.

WARNING
Ensure that you copy the specified BMC Remedy SDK jar files and log4j*.jar file to the DX NetOps Spectrum Library before you execute any DX NetOps Spectrum operation on BMC remedy.

NOTE

- If BMC ITSM is configured with an IP address instead of its Host name, the ticket open/launch in context feature is not supported by BMC Remedy.
- It is recommended for the BMC Remedy server to be on the domain as the DX NetOps Spectrum OneClick Server for better response time.

ServiceAide

DX NetOps Spectrum alarms automatically create incident tickets in the ServiceAide (formerly known as CA Cloud Service Management). Operators can also manually create an incident ticket from an alarm in the OneClick Console. When an incident ticket is closed in ServiceAide, the corresponding alarm in DX NetOps Spectrum is also cleared if you enabled Polling in the Service Desk Configuration page - Integration Parameters.

In order for this integration to work, you must configure a Web Services user account in ServiceAide.

Follow these steps:

1. Create an account in either https://csm3.serviceaide.com or https://csmstaging.serviceaide.com
2. Set a password and a role for the new account you created.
3. Check if you can successfully log in to the ServiceAide web page.
4. In the OneClick Administration - Service Desk Configuration, supply the information and click on the Test button.
5. Click the Add/Modify Server button to add the Service Desk Server Name in the table. You can click on the Save button now, or in the next step.
6. Enable the Service Desk Integration, select the alarm types for which you would like Service Desk tickets created and click on the Save button.
   When you select the Enable option, the Integration Parameters will show up. Enable the Polling if you want to clear the alarm in DX NetOps Spectrum side when the ticket is closed.
7. Once a new alarm is asserted, the Service Desk ticket is created. Click on the Trouble Ticket ID URL for the Ticket details.

WARNING

In ServiceAide, you cannot have a user with the user name 'administrator' as the system user in ServiceAide is 'administrator'.

ServiceNow

You have to select and configure ServiceNow from the OneClick Administration Server > Service Desk Configuration to enable the integration.

The default integration behavior is that DX NetOps Spectrum polls ServiceNow to get the ticket update (Assignee and Status) information from the configured ServiceNow instance.
If you wish to enable OneClick (push) notifications to DX NetOps Spectrum from ServiceNow, refer to the Configuring OneClick Notifications for ServiceNow section.

The integration module makes calls to fetch data from ServiceNow for the properties of an incident on the following tables:

- sys_db_object
- sys_dictionary
- incident
- task
- sys_journal_field

Please ensure that the user which is used in the integration has the privileges to access these tables and fetch data from these tables.

The integration user also needs to have the privileges for the following roles in the ServiceNow > Roles list:

- soap
- soap_create
- soap_delete
- soap_ecc
- soap_query
- soap_query_update
- soap_script
- soap_update
- web_service_admin
- odbc
- midserver
- catalog_admin
- u_journal_entry_user

> **NOTE**
> Additionally, the integration user must have the read & write privileges for the roles which are assigned in the ACLs of the tables sys_db_object, sys_dictionary, incident, task, and sys_journal_field. Contact the ServiceNow administrator to know roles defined in ACLs.

Once the integration user has access to these tables.

Please click the Save button on the ServiceDesk Integration page in DX NetOps Spectrum OneClick Administration.

> **NOTE**
>
> If you have already enabled the integration prior to enabling access to the tables and roles listed above, you will need to restart the Tomcat server.

Configuring the OneClick Server for Service Desk Integration

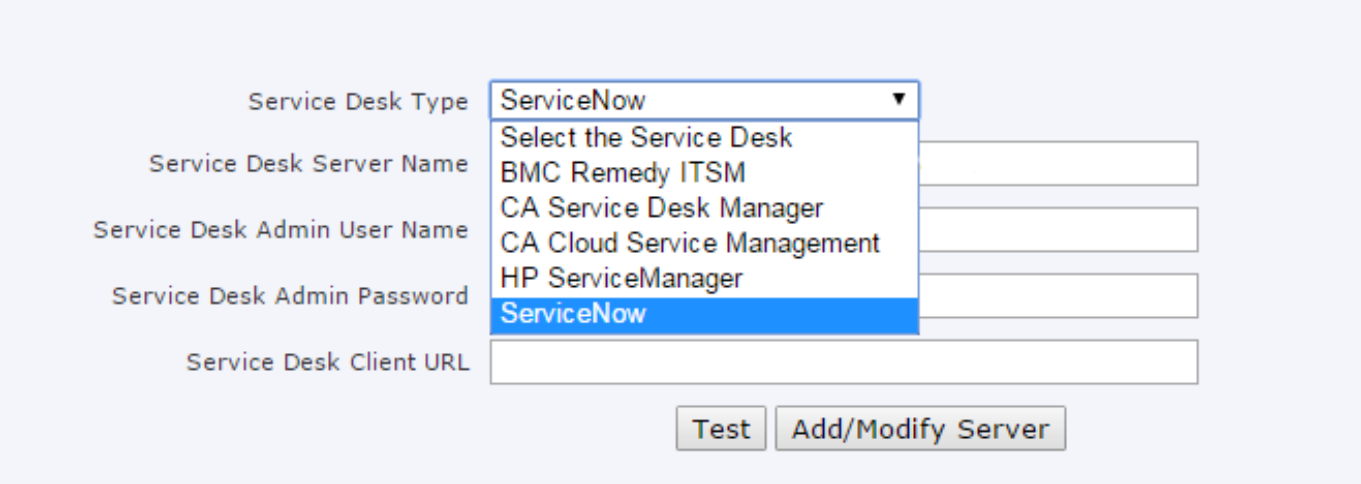
WARNING

The following instructions are applicable for configuring all the Service Desk applications that are supported by DX NetOps Spectrum (including CA Service Desk Manager).

Select Service Desk and enable integration on the DX NetOps Spectrum OneClick Server

To configure the Service Desk integration with Spectrum, follow these steps:

1. Navigate to the **OneClick Administration pages** > **Administration> Service Desk Configuration.**
2. Select the Service Desk you want to integrate with DX NetOps Spectrum, from the **Service Desk Type** field.



3. Configure OneClick to connect to a Service Desk server by entering valid values for the following fields:

   NOTE

   Based on the Service Desk you are integrating the fields may vary, please specify the values applicable for your selected Service Desk.

   - **Service Desk Server Name**
     The host name of the Service Desk Manager server.
   - **Service Desk Server Port**
     The HTTP port of the Service Desk Port. DX NetOps Spectrum uses this port to create Service Desk tickets.
   - **Service Desk Web Server Port**
     The HTTP port of the Service Desk Web Server. This port is used when launching the Service Desk interface from OneClick.
   - **Service Desk Admin Username**
     The user name of the Service Desk server administrator.
   - **Service Desk Admin Password**
     The password of the CA Service Desk Manager server administrator.
   - **Service Desk Client URL** The URL to access the Service Desk web client.
   - **Service Desk Protocol**
     The network access protocol used by the Service Desk
4. Click Test to verify the connection between the Service Desk server and DX NetOps Spectrum OneClick server. A successful test displays a relevant confirmation message.

   NOTE

   The settings are not saved until you click Save.

NOTE

We recommend that you do not add multiple Service Desk applications/ servers.
At any point in time, DX NetOps Spectrum can be integrated with only one Service Desk/ Incident Management application which is enabled in the GUI from the OneClick Admin > Service Desk Configuration > Select Service Desk Type field.

5. Supply values for the following additional fields:

- **SSL Support**
  SSL Support is used in the case of CA Service Desk. If the CA Service Desk is running in 'https' mode, use this option to connect with it using this option.
- **Specify Reported By Field**Select Yes to include the submitting user in the Reported By field when you manually submit Request, Incident and Problem tickets.

NOTE

The user you specify in the **Reported By** the field needs to be an existing user in the Service Desk application, you have configured. If the specified user does not exist in Service Desk application ticket creation and ticket assignment from DX NetOps Spectrum will fail.

- **Assign Assets/Configuration Items**
- **Reload Asset/CI Mapping**
- **Reload Ticket Rules**

NOTE

The above fields (**Assign Assets/Configuration Items**/ **Reload Asset/CI Mapping**/ **Reload Ticket Rules**) are currently not supported in this integration.

- **Service Desk Integration** Select the Enabled option to allow DX NetOps Spectrum to communicate with the specified Service Desk server.

NOTE

The **Integration Parameters** field(s) are displayed only after you enable the Service Desk Integration in the previous option.

Integration Parameters

An enhanced integration framework acting as an intermediary between DX NetOps Spectrum and the Service Desk(s) is part of the CA Spectrum 10.2 install package. The framework installation is part of the Spectrum install and the default Integration Parameters are auto-configured, when you enable this integration. We recommend using this integration capability only for fresh installations of DX NetOps Spectrum.

The Integration Parameters section displays default values for the following fields:

WARNING

It is recommended to use the default values that are displayed in the **Integration Parameter** fields. The Integration Parameter fields are used by the enhanced framework to enable communication between DX NetOps Spectrum and the Service Desk application you are integrating with.

- **Integration Access Protocol**
  Specifies the Access Protocol of the NIM server. This value is the same as the OneClick Server protocol as NIM is deployed on the same machine.
- **Integration Host**
  Specifies the Integration Hostname (Hostname of the machine where the Integration Server is deployed.)

This is the hostname where the OneClick server is deployed. By default, it will be the localhost as the NIM is deployed on the same machine.

- **Integration Port**
  Specifies the Port number for Integration Host
- **Integration Username**
  Specifies the user name (Default is nimadmin)
- **Integration Password**
  Specifies the password of the integration user (Default is nimadmin)
- **Polling**
  By default, Polling is disabled.
  Enabling polling fetches the ticket update (Assignee and Status) information from the configured Service Desk.
  Select the Enabled option to enable polling.

  NOTE

  Time should be in sync for One Click with all Service Desk. Time zones can be different, however, if the time (of the Service Desk Server and Spectrum OneClick server) is not in sync, polling will not work.
- **Polling Interval**
  Specifies how frequently DX NetOps Spectrum queries the configured Service Desk server for updates. If you modify this value, the new polling interval takes effect at the next polling cycle.
  This field specifies the polling interval in seconds. The default value is 300 (seconds). You can configure this field to a value of your choice (in seconds).
- **Status To Check**Specifies the ticket **status to check** from the Service Desk ticket.
  DX NetOps Spectrum looks for the tickets in the configured Service Desk application with the following status': Resolved and Closed. If the ticket status is Resolved or Closed, the corresponding alarms will be cleared from the list of alarms in DX NetOps Spectrum, based on the polling interval defined.
  This feature applies only to alarms that are clearable in DX NetOps Spectrum, and does not apply to non-clearable alarms.

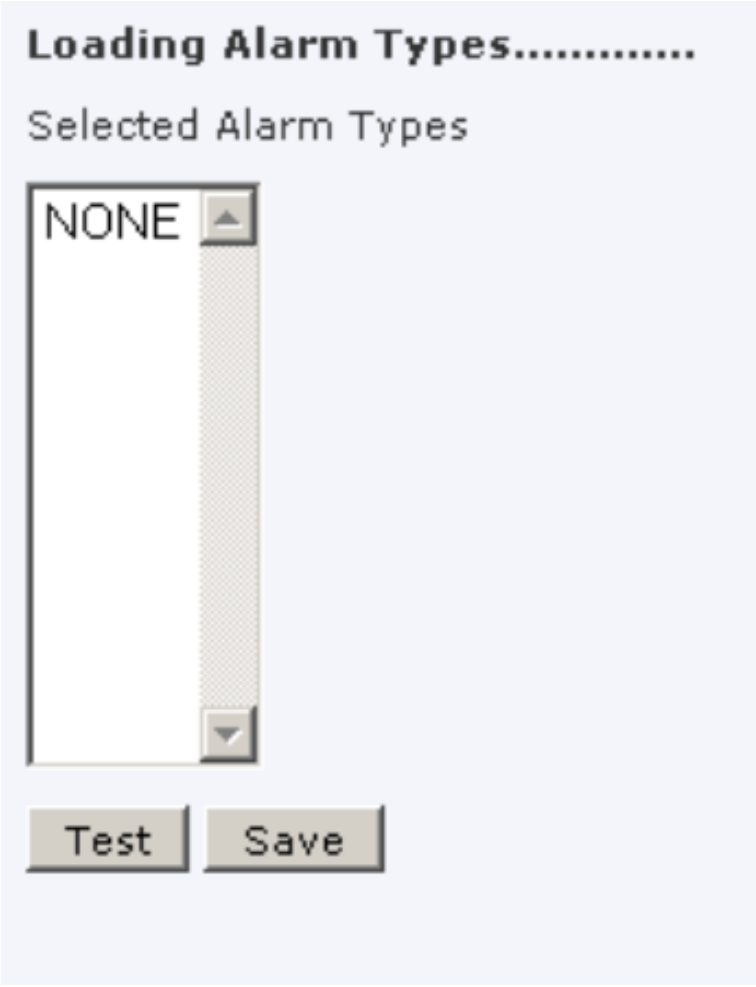
Select Alarm Types to create Service Desk Tickets

Configure the DX NetOps Spectrum and Service Desk integration to create CA Service Desk trouble tickets when DX NetOps Spectrum generates alarms of certain types that you specify.

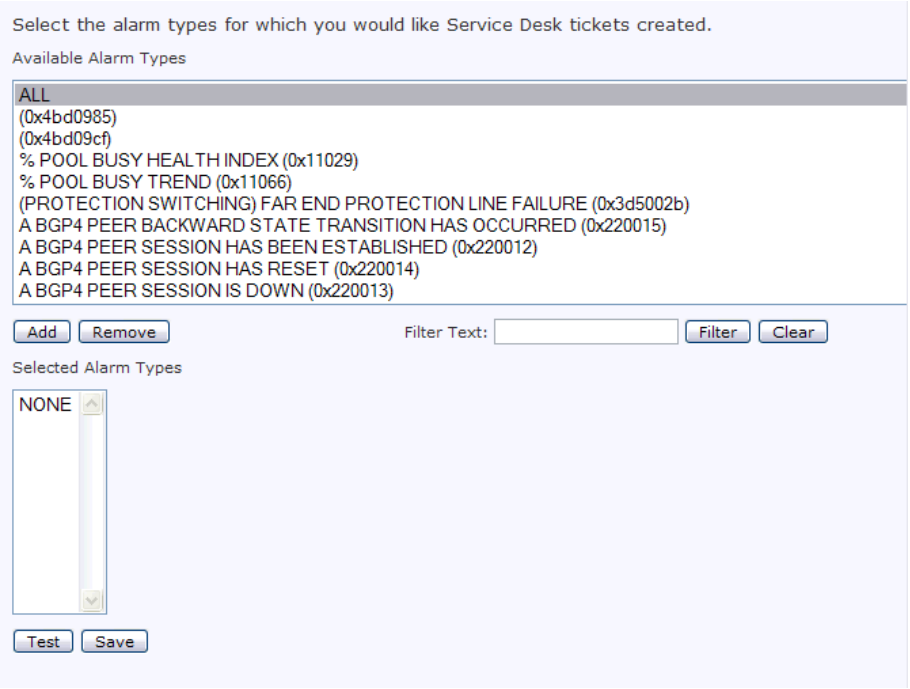
NOTE

Automatic ticket creation is an optional feature. Operators can instead create all tickets manually. By default, the integration does not automatically create tickets for any DX NetOps Spectrum alarms.

1. From the **Available Alarm Types** field, select the alarm types for which you would like Service Desk tickets created. The first time that you open the Service Desk Configuration administration page and anytime the cache timeout (one hour) of the page expires, the “Loading Alarm Types” message is displayed as shown in the following image:



The available alarm types are displayed in the Available Alarm Types section of the Service Desk Configuration administration page:



2. In the Available Alarm Types section, select the alarms for which you want OneClick to create Service Desk tickets, and click Add.

   NOTE
   A delay can occur the first time you add alarm types. The delay occurs when all of the probable cause files are loaded for display in the Available Alarm Types section of the Service Desk Configuration administration page.
3. To generate Service Desk tickets for all DX NetOps Spectrum alarms, select ALL from the Available Alarm Types list, and click Add.

   NOTE
   o select an individual alarm type, enter some text from the desired alarm type into the Filter Text field and click Filter.

   NOTE

   After selecting the alarms, click the Test button to see if your selection is successful. Relevant Confirmation / Error messages are displayed.
4. Click Save.
   A relevant confirmation message appears.

   NOTE

   Click the Save button after modifying the list of selected alarms to ensure that your changes will take effect.

Configuring OneClick Notifications for ServiceNow using MID Server

Overview

ServiceNow can send notifications to OneClick when a ticket that is associated with OneClick alarm changes. These notifications update the alarm that is associated with a ticket in OneClick to reflect changes to the ticket. The ServiceNow integration can be configured to generate an automatic Ticket Closed notification that causes OneClick to clear the associated alarm when a ticket is closed. Similarly, when a ticket has been transferred, a Ticket Transfer notification causes OneClick to update the troubleshooter information for the associated alarm.

> **NOTE**
>
> OneClick only clears an associated alarm when a ticket is closed/resolved in ServiceNow if the alarm is user-clearable.

These notifications use "keywords" which must match keywords that are set in the integration for OneClick. Keywords are case-sensitive.
By default, the keyword for the close action is "Closed", and the keyword for the transfer action is "Transfer" (in both ServiceNow and the DX NetOps Spectrum integration setup). These keywords can be customized in ServiceNow and in the DX NetOps Spectrum integration setup.

Configure ServiceNow to push notifications to OneClick

Follow these steps:

Before you configure ServiceNow and DX NetOps Spectrum to communicate with each other, download and install the integration components on your MID application server. The Management, Instrumentation, and Discovery (MID) Server is a Java server that facilitates communication and movement of data between the ServiceNow platform and external applications, data sources, and services.

Download and Install Integration Components on the MID Server

1. Download and install the MID server from ServiceNow (on the same server where DX NetOps Spectrum is installed or a different server). The Management, Instrumentation, and Discovery (MID) Server is a Java server that facilitates communication and movement of data between the ServiceNow platform and external applications, data sources, and services.
   Refer ServiceNow documentation for Installing MID
   > **NOTE**
   > Please keep handy the name you assign to the MID server during installation and configuration, for use during Creating Business Rules.
2. Download and install integration components from the OneClick Server in the MID server installation directory.
   You can download the integration components on the same machine where the MID agent is installed.
   To achieve this use one of the following methods, depending on your operating systems.
   > **WARNING**
   > Please ensure that **Polling** is disabled from the **OneClick Admin Pages**> **Service Desk Configuration**> **Integration Parameters** before you initiate the OneClick notifications from ServiceNow..
3. Navigate to your ServiceNow server instance and create the following Business Rules:
   - **Update**
   - **Clear**

   For more information, see Creating Business Rules. This example describes creating Business Rules for the Update and Close actions after you have completed and enabled the integration.
   > **NOTE**
   > Please refer ServiceNow™ documentation for information on:
   >
   > - Business Rules
   > - Creating a Business Rule

MID Server and OneClick Web Server Use Different Operating Systems

Perform the following steps when your ServiceNow MID server uses a different operating system than the OneClick web server:

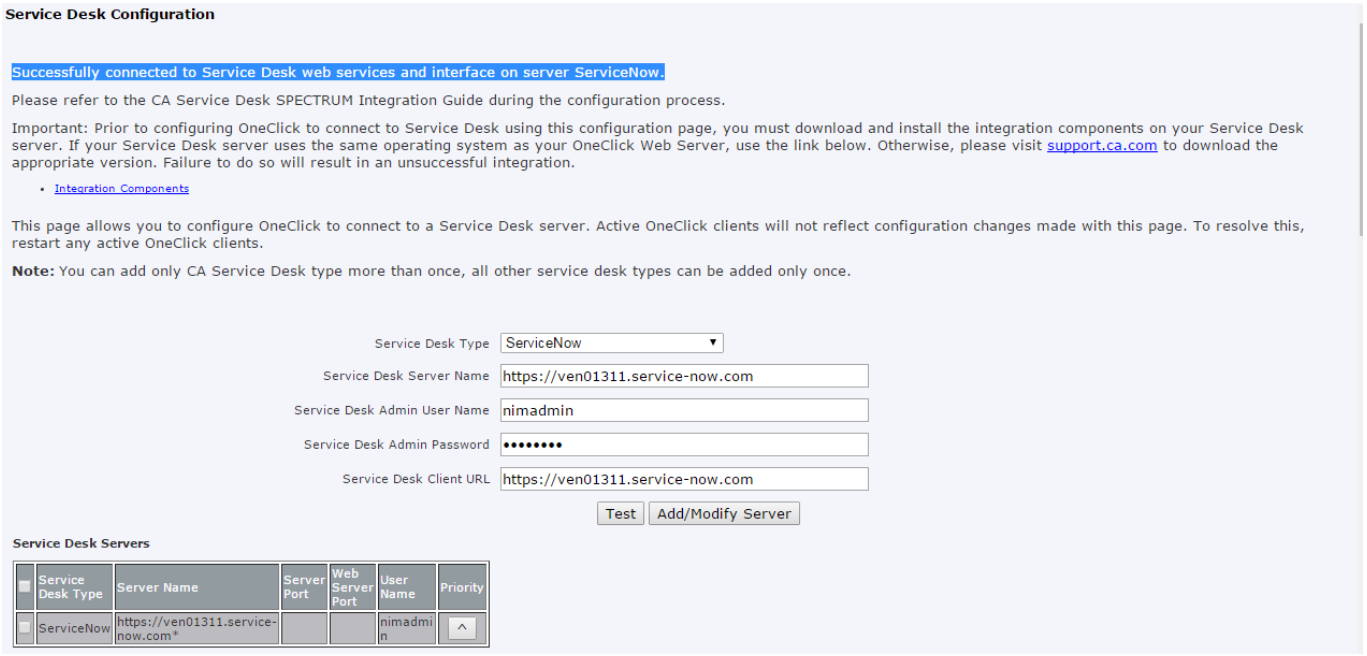
1. Visit the Support website to locate a version of the integration components that are appropriate for your MID server.
2. Download and save the appropriate version of the integration components for your operating system to the following directory on your MID server:
   ```
   MID Server_Installation_directory
   ```
3. After you have saved the integration components to your MID server, follow the instructions for installing and configuring the integration components using the same operating system, beginning with Step 5.

MID server and OneClick Web Server Use the Same Operating System

Download the integration components from the OneClick web server and install them on the MID server.

Follow these steps:

1. From your Service Desk server, navigate to the OneClick Administration pages.
   ```
   http://OneClick Web server/spectrum/admin/index.jsp
   ```
2. Click the Service Desk Configuration link in the left panel of the Administration page.
   The Service Desk Configuration administration page opens in the right panel, as shown:



1. Click the Integration Components link to download the oc_components.exe file. This self-extracting archive file contains the executable programs to configure the Service Desk server.
2. Save the oc_components.exe file to the directory on your MID server:
   ```
   MID Server_Installation_directory
   ```
3. Log in to your MID server host computer and navigate to the *MID_Server_Installation* directory.
4. Locate the oc_components.exe file you downloaded.
5. **Linux:** Run the following command to make the oc_components.exe file executable:
   ```
   chmod 755 oc_components.exe
   ```
   Windows: Do not edit the permissions of the downloaded file for the file to be executable.

6. Run the oc_components.exe file.
   The OneClickIntegrationSetup(.exe) file is extracted to the <*MID Server_Installation_directory*> directory.
7. Run the MID Server_*Installation_directory*/OneClickIntegrationSetup(.exe) configuration program. At each prompt, enter the requested information and press Enter to continue. The following table describes each prompt and the required information:

OneClick Integration Setup Prompt	**Description**
OneClick Server name?>	Enter the hostname of your OneClick Web server.
OneClick Server port?>	Enter the port of the OneClick Web server.
OneClick Homepage path [default="spectrum"]?>	If your OneClick home page URL uses the default value of http://<*oc*> *Web server*/spectrum, press Enter to accept it. Otherwise, enter the correct home page path portion of the OneClick home page URL at this prompt. This home page path value is <path>, as in the example http://*OneClick Web server*>/*path*. The default value in OneClick is “spectrum.”
Username?>	Enter the username of the OneClick Administrator. This name is the DX NetOps Spectrum "super user" who installed the OneClick Web server.
Password?>	Enter the password of the OneClick Administrator.
Confirm password?>	Re-type the password of the OneClick Administrator and press Enter.
Enable logging? [yes|no]>	Type *yes* to enable logging or type *no* to disable logging and press Enter. We recommend enabling logging only when you are troubleshooting an integration problem. An active integration can create a large log file. When logging is enabled, the integration creates a file named oc-notification.log in the *Service_Desk_Installation_directory*/bin directory. When enabled, logging writes information about Service Desk notifications to this log file. Information is logged each time the DX NetOps Spectrum notification occurs and includes the type of activity and whether or not it was a success. In the case of a failed notification, this log file may contain a possible solution such as an invalid port or that the OneClick web server is unavailable.
Enable SSL? [yes|no]>	Type *yes* to enable SSL or type *no* to disable SSL and press Enter. To enable SSL in DX NetOps Spectrum Tomcat, install OneClick and see Administrating. To enable SSL in CA Service Desk Tomcat, see Service Desk Administration.
Path to the JRE root installation directory?>	Type the JRE root installation directory path and press Enter. Specify the JRE root installation directory so that appending "bin/java - version" becomes a successful command. Java 2 Runtime Environment (JRE) version 1.8.0 or later is required.
Close keyword [default="Closed"]?>	To use default values press Enter at the prompt without specifying a keyword. Leave the line blank and press Enter.
Transfer keyword [default="Transfer"]?>	To use default values press Enter at the prompt without specifying a keyword. Leave the line blank and press Enter.

The OneClick Integration Setup program creates a file named NotifyOneClick(.bat or .sh, depending on the operating system), in the *MID Server_Installation_directory*.

> **NOTE**
> To reconfigure this information later, run the OneClickIntegrationSetup program again or manually edit values in the M*ID Server_Installation_directory*/oc-integration.cfg configuration file. Any changes that are made to this file take effect immediately. No additional restart is required.

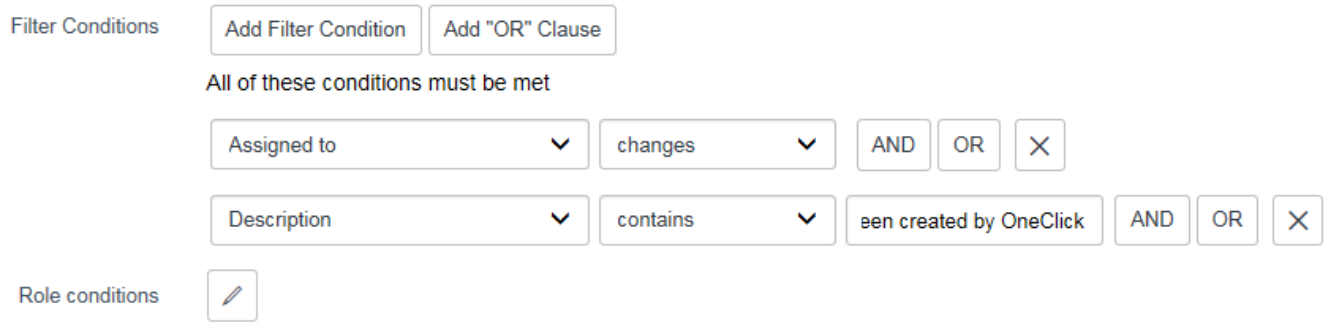
Creating Business Rules

1. Navigate to **Business Rules** (under **System Definition** tab) and click **New** to create a new rule.
2. Provide a relevant name to the Business Rule, for e.g. 'Update'.

3. Select **“Incident”** from the **Table** drop down.
4. Check **“Active”** and **“Advanced”** check box(es).
5. In the **When to run** section, do the following:
   a. Select “after” from the **When** drop down
   b. Enter 100” in the the **Order** field for update rule.
   c. Check the **“Update”** checkbox.
   d. Add filter conditions.
   For **Update** rule :

Assigned to + Changes AND Description + contains + **“This ticket has been created by OneClick”**.

The following image displays how Filter Conditions show up for this Business Rule in ServiceNow



Update Business Rules

In In the **Advanced** section at the bottom of the page give the below script for **Update** by changing the relevant details.

See comments in the script below:

```
function onBefore(current, previous) {
 //This function will be automatically called when this rule is processed.
myFunction();}
 //Update
function myFunction() {
 //Set the path of the OCNotify.jar
 var jarLocation = "C:\\MID_Server\\OCNotify.jar";//OCNotifier.jar"; \\example: "C:\\Users\\abcuser\\Desktop\
\OCNotifier.jar";
 //change the name of the spectrum_mid to be the name of your midserver you specified during mid server
 configuration/installation
 var midHost = "mid.server.spectrum_mid";
 var ecc = new GlideRecord('ecc_queue');
 ecc.initialize();
 ecc.agent = midHost;
 ecc.topic = "Command";
 ecc.queue = "output";
 var description = current.description;
 var re = new RegExp("Alarm ID: ((\\w+)(-\\w+){4})");
 var expResult = re.exec(description);
 var assignee = current.assigned_to.getDisplayValue();
```

```
 var cmd = "java -jar " + jarLocation + " " + "update " + expResult [1] + " " + encodeURI(assignee);
 ecc.payload='<parameters><parameter name="name" value="'+cmd+'"/></parameters>';
 ecc.insert();
}
```

Update Script for New York Version

Use the following script for the New York version.

```
(function executeRule(current, previous) {
//This function will be automatically called when this rule is processed.
myFunction();
})(current, previous);
 //Update
function myFunction() {
 gs.log('in function'+current.number);
 //Set the path of the OCNotify.jar
 var jarLocation = "D:\\SNMidServer\\MidServer01\\agent\\OCNotify.jar";
 //change the name of the MidServer01 to be the name of your midserver you specified during mid server
 configuration/installatio
 var midHost = "mid.server.MidServer01";
 var ecc = new GlideRecord('ecc_queue');
 ecc.initialize();
 ecc.agent = midHost;
 ecc.topic = "Command";
 ecc.queue = "output";
 var description = current.description;
 var re = new RegExp("Alarm ID: ((\\w+)(-\\w+){4})");
 var expResult = re.exec(description);
 var assignee = current.assigned_to.getDisplayValue();
 var cmd = "java -jar " + jarLocation + " " + "update " + expResult [1] + " " + encodeURI(assignee);
 gs.log('in function cmd' +cmd);
 ecc.payload='<parameters><parameter name="name" value="'+cmd+'"/></parameters>';
 gs.log('in function payload' + ecc.payload);
 ecc.insert();
 gs.log('in function inserted');
}
```

Close Business Rules

For creating the **Close** rule, follow the steps mentioned for Creating Business Rules,

In step 5 b. enter “200” as the value in the **Order** field, and in step 5d. add the following filter conditions for **Close** rule:

State + is + Closed AND

Description + Contains + **“This ticket has been created by OneClick”** AND

State + Changes

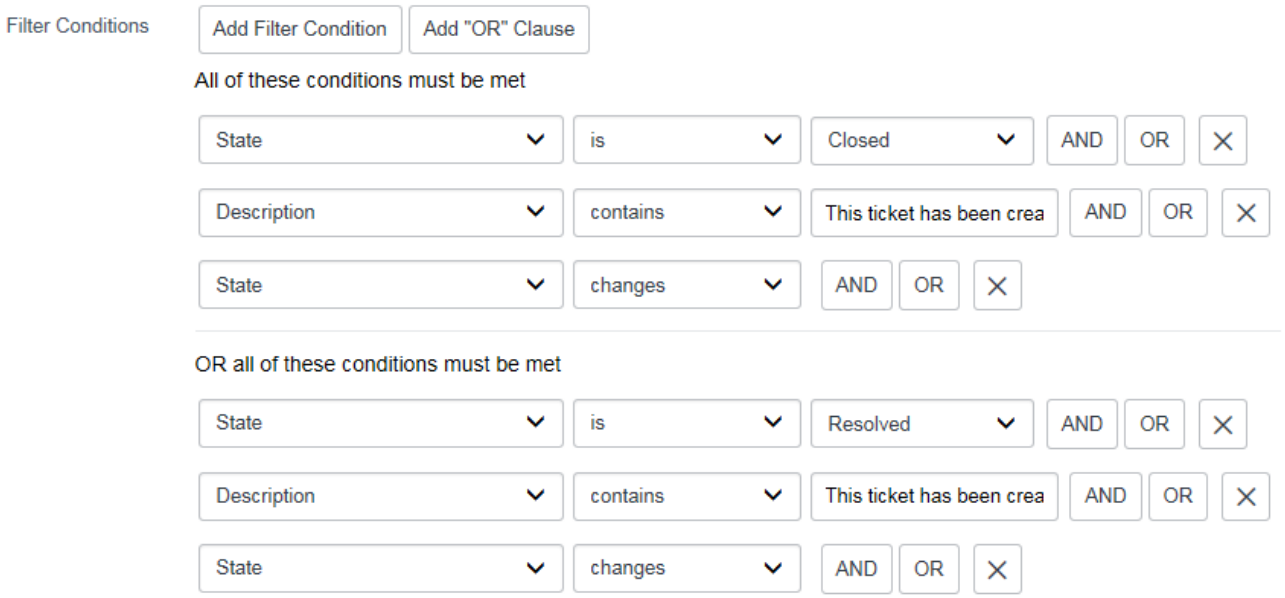
Click ‘Add “OR” Clause’ button and add another filter as:

State + is + Resolved AND

Description + Contains + “**This ticket has been created by OneClick**” AND

State + Changes

The following image displays how Filter Conditions show up for this Business Rule in ServiceNow



In the **Advanced** section at the bottom of the page give the below script for Close by changing the relevant details. See comments in the script, below

```
function onBefore(current, previous) {
 //This function will be automatically called when this rule is processed.
 myFunction();}
//Clear
function myFunction()
{
 //Set the path of the OCNotify.jar
 var jarLocation = "C:\\MID_server\\OCNotify.jar";
 //change the name of the spectrum_mid to be the name of your midserver you specified
 during mid server configuration/installation
 var midHost = "mid.server.spectrum_mid";
 var ecc = new GlideRecord('ecc_queue');
 ecc.initialize();
 ecc.agent = midHost;
 ecc.topic = "Command";
 ecc.queue = "output";
 var description = current.description;
 var re = new RegExp("Alarm ID: ((\\w+)(-\\w+){4})");
 var expResult = re.exec(description);
 var status = current.state.getDisplayValue();
 var cmd = "java -jar " + jarLocation + " " + "clear " + expResult [1] + " " +
 encodeURI(status);
 ecc.payload='<parameters><parameter name="name" value="'+cmd+'"/></parameters>';
```

```
 ecc.insert();
}
```

Clsoe Script for New York Version

Use the following script for the New York version.

```
(function executeRule(current, previous) {
 //This function will be automatically called when this rule is processed.
myFunction();
})(current, previous);
//Clear
function myFunction()
{
 //Set the path of the OCNotify.jar
 var jarLocation = "D:\\SNMidServer\\MidServer01\\agent\\OCNotify.jar";
 //change the name of the MidServer01 to be the name of your midserver you specified during mid server
 configuration/installation
 var midHost = "mid.server.MidServer01";
 var ecc = new GlideRecord('ecc_queue');
 ecc.initialize();
 ecc.agent = midHost;
 ecc.topic = "Command";
 ecc.queue = "output";
 var description = current.description;
 var re = new RegExp("Alarm ID: ((\\w+)(-\\w+){4})");
 var expResult = re.exec(description);
 var status = current.state.getDisplayValue();
 var cmd = "java -jar " + jarLocation + " " + "clear " + expResult [1] + " " + encodeURI(status);
 ecc.payload='<parameters><parameter name="name" value="'+cmd+'"/></parameters>';
 ecc.insert();
}
```

The setup is now complete.
For updates made to tickets in ServiceNow, you should see the changes being reflected in the associated alarms in OneClick Console, after the changes made to the ticket are completed on the ServiceNow side .

> **NOTE**
>
> To disable OneClick notifications from ServiceNow to OneClick, go to the Business Rule, clear the **“Active”** checkbox for both the Business Rules (Update and Clear) and click **Save.**

Alarm Fields, REST Examples and Attribute Mapping

DX NetOps Spectrum Alarm Fields

By default, the enhanced integration framework for integrating DX NetOps Spectrumwith the third-party Service Desk applications uses a polling mechanism to communicate with all the Service Desks.

The following DX NetOps Spectrum Alarm Model fields data is passed to Service Desks to create an incident.

- Summary
- reportedByUserID
- Severity
- Alarm_Status
- assetID
- TroubleShooter
- Creation_Date
- ModifiedTime
- affectedEndUserID
- resolution

Mapping Alarm Model Fields to Incident Fields

The following REST API returns mappings of the above-mentioned DX NetOps Spectrumalarm model fields to the incident fields:

URL	**Sample**: http://<serverhost>:<serverport>/ca-nim-sm/api/v2/config/integration/caspectrum/mapping/incident
Method	**GET**
Request Body	**None**
Headers	**Authorization**: Basic bmltYWRtaW46bmltYWRtaW4=
	Accept: application/json
	Content-type: application/json

Response


```
 [
  {
    "NIMField": "reportedByUserID",
    "MDRField": "reportedByUserID",
    "Description": "UserName or ID of the user who reported the
 Incident"
},
  {
   "NIMField": "name",
    "MDRField": "summary",
    "Description": "short description of the Incident"
},
  {
   "NIMField": "severity",
    "MDRField": "severity",
    "Description": "Severity of the Incident"
  },
  {
    "NIMField": "description",
    "MDRField": "description",
    "Description": "Description of the Incident"
  },
  {
    "NIMField": "status",
    "MDRField": "alarmStatus",
    "Description": "Status of the Incident"
  },
  {
    "NIMField": "affectedCIID",
    "MDRField": "assetID",
    "Description": "CI on which Incident is created"
  },
  {
    "NIMField": "assigneeUserID",
    "MDRField": "troubleShooter",
    "Description": "User to which the incident is assigned"
  },
  {
    "NIMField": "creationTimeStamp",
    "MDRField": "creationDate",
    "Description": "Time at which the incident is created/
opened"
  },
  {
    "NIMField": "updatedTimeStamp",
    "MDRField": "modifiedTime",
    "Description": "Last updated time stamp for the incident"
  },
```



```
  {
    "NIMField": "affectedEndUserID",
    "MDRField": "affectedEndUserID",
    "Description": "UserName or ID of the user impacted by the
 Incident"
```

Mapping Incident Fields and Service Desk Incident

To view field mappings between integration incident and Service Desk incident, use the below REST API:

URL	**Generic**:http://<serverhost>:<serverport>/ca-nim-sm/api/<version>/config/integration/<mdrName>/mapping/<ciName> **Sample**: *http://<Spectrum_OC_Host>:<OC_Port> /ca-nim-sm/api/v2/config/integration/<service_desk_id>/mapping/incident*
Method	**GET**
Request Body	**None**
Headers	**Authorization**: Basic bmltYWRtaW46bmltYWRtaW4=
	Accept: application/json
	Content-type: application/json
Example	To see mappings between normalized incident and ServiceNow incident fields, following REST API should be invoked – *http://<serverhost>:<serverport>/ca-nim-sm/api/v2/config/integration/<service_desk_id>/mapping/incident*

Verifying or Looking at Log Levels

This REST API is used to get log levels of the integration components.

URL	**Generic**: http://<serverhost>:<serverport>/ca-nim-sm/api/<version>/Loglevel **Sample**: http://<SPECTRUM_OC_HOST>:<OC_PORT>/ca-nim-sm/api/v2/Loglevel
Method	**GET**
Request Body	**None**
Headers	**Authorization**: Basic bmltYWRtaW46bmltYWRtaW4=
	Accept: application/json
	Content-type: application/json

Response


```
[
  {
    "name": "root",
    "level": "INFO"
  },
  {
    "name": "cacsm",
    "level": "INFO"
  },
  {
    "name": "omodule",
    "level": "INFO"
  },
  {
    "name": "oapi",
    "level": "INFO"
  },
  {
    "name": "resources",
    "level": "INFO"
  },
  {
    "name": "model",
    "level": "INFO"
  },
  {
    "name": "HPServiceManager",
    "level": "INFO"
  },
  {
    "name": "common",
    "level": "INFO"
  },
  {
    "name": "configuration",
    "level": "INFO"
  },
  {
    "name": "com.ca.integration.normalization.mdr.ca.cmdb",
    "level": "INFO"
  {
    "name": "BMCRemedy",
    "level": "INFO"
  },
  {
    "name": "im",
    "level": "INFO"
  },
  {
    "name": "ServiceNow",
    "level": "INFO"
  },
  {
```

Enable Loglevel in Integration Component to Debug

The following REST API is used to set the log level for the given integration component.

URL	**Generic**: http://<serverhost>:<serverport>/ca-nim-sm/api/<version>/Loglevel/<name> **Sample**: http://<SPECTRUM_OC_HOST>:<OC_PORT>/ca-nim-sm/api/v2/Loglevel/common
Method	**PUT**
Request Body	**None**
Headers	**Authorization**: Basic bmltYWRtaW46bmltYWRtaW4=
	Accept: application/json
	Content-type: application/json

Payload	<pre>{
	"name": "common",
	"level": "DEBUG"
	}</pre>

Response	<pre>{
	"name": "common",
	"level": "DEBUG"
	}</pre>

Encoded/ Plain User Password Using REST API in 10.4.2.1

URL	http://<serverhost>:<serverport>/spectrum/restful/model/modelhandle?attr=0x11f9a&val=<encodedpassword>
Method	PUT
Request Body	None
Header	application/xml
Example	http://<serverhost>:<serverport>/spectrum/restful/model/0x1200077c?attr=0x11f9a&val=9D.1.0.25.43.1.0.6.0.0.0.20.0.0.0.66.53.dc.e9.54.65.03.de.c7.1e.3c.9c.10.a8.de.0f.72.5e.25.f0.18.07.d3.0d.df.e2.1c.fe.26.79.c4.80

Attribute Mapping

The following table displays the attribute mappings between DX NetOps Spectrum(Alarms and Incidents) and the supported Service Desk applications.

Any updates made to these attributes from either DX NetOps Spectrumor the configured Service Desk will be updated and reflect in the other, based on the polling/ OneClick notification mechanism you configure.

NOTE

The OneClick notification through the push mechanism is enabled/supported only for ServiceNow. By default, DX NetOps Spectrumpolls the Service Desk servers for updates.

Alarm Field Name	Incident Field Name	MDR FIELD NAMES	MDR FIELD NAMES	MDR FIELD NAMES	MDR FIELD NAMES
		CACSM	**ServiceNow**	**HPSM**	**BMC Remedy ITSM**
reportedByUserID	reportedByUserID	RequesterID	opened_by		Submitter
Summary	name	Description	short_description	Title	Description
Severity	severity	Severity	severity		
description	description	DescriptionLong	description	Description	Detailed Description

Alarm_Status	status	TicketStatus	state	Status	Status
assetID	affectedCIID	RelatedCI	cmdb_ci	AffectedCI	
Troubleshooter	assigneeUserID	AssignedContactID	assigned_to	Assignee	Assignee Login ID
Creation_Date	creationTimeStamp	CreationTimeStamp	opened_at	OpenTime	Submit Date
ModifiedTime	updatedTimeStamp	LastModTimeStamp	sys_updated_on	UpdatedTime	Last Modiied Date
affectednEnduserID	affectednEnduserID	RequestedForID	caller_id	TicketOwner	
resolution	resolution	Resolution	close_notes	Solution	Resolution

Customize DX NetOps Spectrum to CA NIM Field Mapping

From 10.4.1, you can create a ticket with five more DX NetOps Spectrum attributes (either Alarm or Model) in the ticketing system using the CA Normalized Integration Management (NIM) custom attributes. You can map DX NetOps Spectrum attributes to NIM custom attributes and populate the same in the ticketing system.

Before adding a new attribute mapping, copy the nim-mappings-config.xml file from the `<SPECROOT>\WEB-INF\svdsk\config` directory to the `<SPECROOT>\custom\svdsk\config` directory.

Prerequisites:

- Ensure that you maintain a unique mapping; that is, one DX NetOps Spectrum attribute to one NIM Field.
- Avoid duplicate entries in either DX NetOps Spectrum attribute or NIM Field.
- A maximum of five custom mappings are supported, which is a limitation in NIM.
- Ensure that the NIM custom fields are mapped to the corresponding Service Desk fields in the NIM Customization page to populate the DX NetOps Spectrum attributes to the ticketing system.

The following are the custom fields in CA NIM:

- customField1
- customField2
- customField3
- customField4
- customField5

Sample Configuration for Reference:

The below example shows the customization of the DX NetOps Spectrum attribute to the CA NIM Field Mapping attributes to the ticketing system.

```
<custom-mappings>

<incident>
----
---
<!--
##### Attribute Custom Mapping Support #####
Below example describes - Spectrum Model Name (0x1006e) mapped to NIM customField1 - Spectrum Alarm Symptom
 Count(0x12a06) mapped to NIM customField2
Please make sure maintain unique mappings,i.e. 1 Spectrum Attribute to 1 NIM Field.
There should not be duplicate entries in either Spectrum Attribute or NIM Field.
Maximum 5 custom mappings are supported, which is current limitation in NIM
Please make sure the NIM custom field mapped to the corresponding Service Desk field in the NIM Customization
 page to populate the spectrum attributes to the ticketing system.
      -->
```

```
<attribute-mappings>
   <Mapping>
      <model-attribute>0x1006e</model-attribute>
      <NIMField>customField1</NIMField>
    </Mapping>
    <Mapping>
      <alarm-attribute>0x12a06</alarm-attribute>
      <NIMField>customField2</NIMField>
   </Mapping>
</attribute-mappings>
</incident>
</custom-mappings
```

Troubleshooting Service Desk Integrations (other than CA Service Desk Manager)

Service Desk connection is not successful

Symptom: Service Desk connection is not successful /callback to DX NetOps Spectrum is not working/ticket creation is not successful.

Workaround: Follow these steps

- Verify that the direct login via the browser with respective Service Desk is working.
- To ensure that the integration component is up and running, check if the following url is accessible or not:
  `http://<OC_HOST>:<OC_PORT>/ca-nim-sm/ui/login.jsp`
- Verify the logs (after log level change to MAX) for the payload that DX NetOps Spectrum is sending to NIM and make sure that the values that are mapped correctly.
  To see the mappings refer: Attribute Mapping

  NOTE

  If the Service Desk application is down, logs will be updated in DX NetOps Spectrum > Tomcat logs.

To enable NIM logging run an API call on OneClick server for the following:

- Get Loglevels
- Update Loglevels

For more information refer to the REST APIs example section.

Select the following options to enable the log level to MAX in DX NetOps Spectrum.

- NIM Polling Services - enables the callback logging when we enable polling
- NIM Rest Services - enables the logging and can find the payload that is being sent to NIM

.

Unable to Integrate ServiceNow, get an error - Please check the configuration for ServiceNow(ServiceNow)

Symptom: While integrating DX NetOps Spectrum with Servicenow, we get an error - **Unable to Integrate ServiceNow, get an error - Please check the configuration for ServiceNow(ServiceNow)**

Cause: We get this error if there is a Proxy configured for the browsers. At this point of time, we do not support proxy configuration for this integration.

Workaround:
You need to have a direct connection to the Servicenow servers for the integration to work. This is being considered as an enhancement in the future releases

Integration with Layer7 SiteMinder

Overview

The DX NetOps Spectrum and CA SiteMinder® the integration lets OneClick use CA SiteMinder® Single Sign-On security management capabilities to authenticate DX NetOps Spectrum users. The integration also supports users of applications such as CA Portal and CA eHealth Reports that are integrated with DX NetOps Spectrum

This release of DX NetOps Spectrum, integrates with CA SiteMinder® 12.52 SP01.

OneClick users experience single sign-on in the integrated application environment as follows:

- Users are not prompted for login credentials when they start the OneClick Console from the OneClick home page.
- Users who are logged in to the CA Portal can start the OneClick Console without having to explicitly log in to OneClick.

About DX NetOps Spectrum User Models

Before users can access OneClick in a single sign-on environment, they must have corresponding DX NetOps Spectrum user models. The administrator creates a model for each user in DX NetOps Spectrum.

For more information, see OneClick Administration section.

How OneClick Is Integrated with Layer7 SiteMinder

Complete the following procedures to integrate DX NetOps Spectrum OneClick with CA SiteMinder®:

1. Configure the required OneClick web server parameters on the SiteMinder Policy Server.
2. Register the OneClick web server as a trusted host with Policy Server from the OneClick Administration Pages.

See also:
Configure OneClick Web Server Settings
Disable or Re-enable the Integration
Troubleshooting the Integration Related Issues

Configure OneClick Web Server Parameters in Policy Server

This section describes procedures for setting up the following DX NetOps Spectrum OneClick integration components in Policy Server:

> **NOTE**
> See the Layer7 SiteMinder help if you require more information on Policy Server concepts, components and User Directory creation.

Access the SiteMinder Administration Window

The Administration window is the user interface to the SiteMinder Policy Server.

Follow these steps:

1. Access the Single Sign On web page using the site minder url provided by a administrator: **https://<HOSTNAME>:18443/iam/siteminder/adminui**
2. The Administration Login window opens. Log In as a Policy Server administrator for the domain specified in the Main Policy Server web page address using the credentials provided by the administrator.
   The Administration window opens.

Create a DX NetOps Spectrum OneClick Host Configuration Object

When you create a DX NetOps Spectrum OneClick web server host configuration object, you specify the parameters the DX NetOps Spectrum OneClick web server host uses when it connects to the SiteMinder Policy Server.

Follow these steps:

1. Go to the **Infrastructure > Host >Host Configuration Objects**.
2. Select **Create Host Configuration object**
3. Select Create a new object and type **Host Configuration** and click **OK**.
4. The Site Minder Create Host Configuration window opens. Type '**spectrum_oneclick_server**' in the name of the **Host Configuration Object** field.
5. Under the Configuration Values Section type '**Hostname**'.
6. Enter **44441** as the port value (for all ports) and click **Submit**.

   NOTE

   You can ignore the default parameter values.

   The host configuration object is now created.

Create a DX NetOps Spectrum OneClick Custom Agent Type

The DX NetOps Spectrum OneClick custom agent type defines the actions that can be performed by the DX NetOps Spectrum OneClick custom agent.

NOTE

Verify that the View, Agent Types option is selected.

Follow these steps:

1. Go to **Infrastructure > Agent >AgentTypes**.
2. Select **Create Agent Type**.
   The SiteMinder Agent Type Window opens.
3. Type '**Spectrum OneClick Custom Agent'** in the **Name** field.
4. Define an action by Clicking **Create** in the Agent Type Definition tab.
5. The New Agent Action box opens.Type '**Authenticate**' and Click **Submit**.
   The action name appears in the Actions List.
   The new agent type is now created.

Create a DX NetOps Spectrum OneClick Custom Agent

The DX NetOps Spectrum OneClick custom agent enforces the Policy Server actions on the OneClick web server.

Follow these steps:

1. Go to **Infrastructure > Agents.**
2. Select **Create Agent.**
   The SiteMinder Agent Windows opens. Select Create a new object of type Agent and click **OK**.

3. In the **Name** field, type ‘s**pectrum oneclick custom agent**’. This field specifies the the name of the web agent.
4. You can select either ‘**CA Single Sign On**’ or ‘**RADIUS**’ as the **Agent Type Settings**.
5. You can select the **Agent Type** from the list of the CA Single Sign On or Radius Vendors depending on the Agent style. If you have selected ‘s**pectrum oneclick custom agent**’ in the Agent Type settings then select **'Spectrum OneClick Custom Agent'** from the Agent Type List.
6. Ensure that the **'Support 4.x agents'** option is not selected. This field specifies the IsAgent4x attribute for this Agent instance.
7. Click **Submit**.
   The new agent is now created.

Create Domain

To create a domain, follow these steps:

1. Go to **Policy>Domain>Domains**
2. Select **Create Domain**. The Create Domain window opens.
3. Type ‘**spectrumdomain**’ in the **Name** field. This field specifies the name of the policy domain.
4. The Global Policy Apply box is to be kept checked.
5. To create the **User Directory**, select **Create** and see the SiteMinder User Directory section for details. If the User Directory is already created, then click on the **Add/Remove** button, select the directory and add to the selected list on the right. Click **OK**.
6. Click **Submit**.
   The domain is now created.

Create a DX NetOps Spectrum OneClick Realm

The DX NetOps Spectrum OneClick realm specifies the resources on the OneClick web server that are protected and that require single sign-on authentication to be accessed.

When deploying Single Sign-On for DX NetOps Spectrum OneClick and applications with which it is integrated, the DX NetOps Spectrum OneClick realm and the other application realms should be included in the same domain object. Also, the domain object user store should include users who require access to OneClick.

Follow these steps:

1. Got to **Policies>Domain>Realms**
2. Select **Create Realm**.
   The SiteMinder Create Realm: Select Domain window opens.
3. Select the Domain Name and click **Next**.
4. The Define Realm window opens.
5. Type ‘**spectrum_oneclick’** in the Name field and ‘spectrum’ in the Description field.
6. Select the **Agent** from List of Agents and click **OK**.
7. Type **‘/spectrum/’** in the Resource Filter field. This field specifies the path of the resource filter. The filed functions as a root for locating resources.
8. Select the **'Protected'** option in the Default Resource Protection section.
9. Select the authentication scheme which protects the realm. You can set it to Basic, HTML forms, proxy UI.
10. To create Rules, see Create Rules section.
11. Click **Finish**.
    The realm is now created.

Create Rules

To create rules, follow these steps:

1. Go to **Policies>Domain>Rules**
2. Select **Create Rules**. The Create Rules: Select Domain window opens.
3. Select the already existing Domain which was created and click **Next**.
   If the Domain is not created, see the Create Domain section.
4. The Create Rule: Select Realm window opens.
5. Select the existing Realm and click **Next**. If the Realm is not created, see the Create Realm section.
6. The Create Rule: Define Realm window opens.
7. Enter '**spectrumrule'** in the name field. This field specifies the name of the rule.
8. Select **Authenticate**, in the Action section.
9. Click **Finish**.
   The rules are now created.

Create Domain Policies

To create domain policies, follow these steps:

1. Go to **Policies>Domain>Domain Policies**
2. Select **Create Policy**. The **Create Policy: Select Domain** window opens.
3. Select the existing Domain name and click **Next**. The **Create Policy:General** window opens.
4. Enter '**spectrumpolicy**' in the **Name** field. This field specifies the name of the policy.
5. Click **Next**.The **Create Policy:Users** window opens.
6. Under the User Directories section, select **Add All** and click **Next**. The **Create Policy: Rules** window opens.
7. Select the existing Rule and click **Next**. The **Create Policy: Expression** window opens.
8. Click **Finish**.
   The domain policies are now created.

Configure OneClick Web Server Host Registration Settings

This section includes configuration procedures for setting up log on authentication by CA SiteMinder® for users who access OneClick.

Register the OneClick Web Server with the CA SiteMinder® Policy Server

This section describes how to register DX NetOps Spectrum OneClick Web Server as a trusted host in the Policy Server. The term *trusted host* refers to the webserver host.

When you register the webserver host, initialization parameters that enable the host to connect to the Policy Server are saved to a local configuration file, SmHost.conf. Once the host connects to the Policy Server, the host uses the settings that are specified in the corresponding host configuration object in Policy Server.

Follow these steps:

1. Access the OneClick home page.
2. Click Administration.
   The Administration Pages open.
3. Click Single Sign-On Configuration.
   The Single Sign-On Configuration page opens.
4. Select the SITEMINDER option.
   The SITEMINDER Configuration form opens.

5. Specify the following registration settings in the OneClick Host Registration section:
   - **Policy Server IP Address**
     Specifies the IP address of the Policy Server where you configured the OneClick host configuration object.
   - **Policy Server Port**
     Specifies the Policy Server port number.
     Default: 44441
   - **Policy Server Admin Username**
     Specifies the username of the CA SiteMinder® administrator with privileges in the domain.
   - **Policy Server Admin Password**
     Specifies the administrator password.
   - **Trusted Host Name**
     Specifies the fully qualified domain name of the OneClick web server host.
   - **Host Configuration Object**
     Specifies the OneClick web server host object that is configured on the Policy Server (spectrum_oneclick_server).
     The OneClick host configuration should look like the following:

6. Click Register.

Initialization parameters specified in the registration are saved to the SmHost.conf file. A new form with the “Successfully registered with the Policy Server” message opens. The form includes configuration panels where you can specify additional OneClick registration settings for the Policy Server.

Configure OneClick Web Server Settings

After you have registered the OneClick Web Server host with Policy Server, you can set additional parameters:

- In the Policy Server Settings panel, you can specify backup Policy Servers that have been set up for failover and the maximum amount of time (in seconds) the DX NetOps Spectrum OneClick Web Server waits before it drops a connection request from an unresponsive Policy Server.
- In the OneClick Agent Settings panel, specify the OneClick web server agent and cookie domain parameters. You can also instruct the webserver agent to check IP addresses in cookies so that it can reject unauthorized web server requests if the IP address that is stored in a cookie does not match the IP address of the requester.
- In the Authentication Logging panel, you can specify whether to log authentication information to the Tomcat log file or another log file. You can also disable logging.
- In the DX NetOps Spectrum Authentication Failover panel, you can specify whether to allow OneClick authentication when Single Sign-On authentication fails because the Policy Server cannot be reached. This means that DX NetOps Spectrum users would be able to log on to OneClick as they normally would without a single sign-on.

These settings do not take effect until you save the settings and enable the integration.

The following sections describe the configuration panels and OneClick registration settings in more detail.

Policy Server Settings

The Policy Server Settings panel is available on the OneClick Administration, Single Sign-On Configuration page in the SITEMINDER Configuration form. This form is available after you have successfully registered the OneClick Web Server host with the Policy Server. For more information, see Register the OneClick Web Server with the SiteMinder Policy Server.

In the Policy Server Settings panel, you can specify one or more backup Policy Servers that have been configured for failover. The OneClick Web Server attempts to connect to a backup Policy Server if it detects that the primary Policy Server that is specified in the Registration form is down.

The following shows the Policy Server Settings panel:

You can also specify the timeout interval for connection requests by the OneClick web server to the Policy Server. Set an interval in the Policy Server Settings form. The OneClick web server drops the request if the Policy Server does not respond within the interval. The default interval is 60 seconds. Increase the interval if your connection requests result in frequent drops in high-data traffic or network slowdowns.

NOTE
The OneClick web server does not attempt to connect to a backup server after a request drop because a connection failure alone does not necessarily indicate that the primary server is down.

Follow these steps:

1. To add a backup server, enter the IP address and a port number for the backup server if it differs from the default port, 44441, and click Add.

The backup server and port number are added to the Policy Servers box.

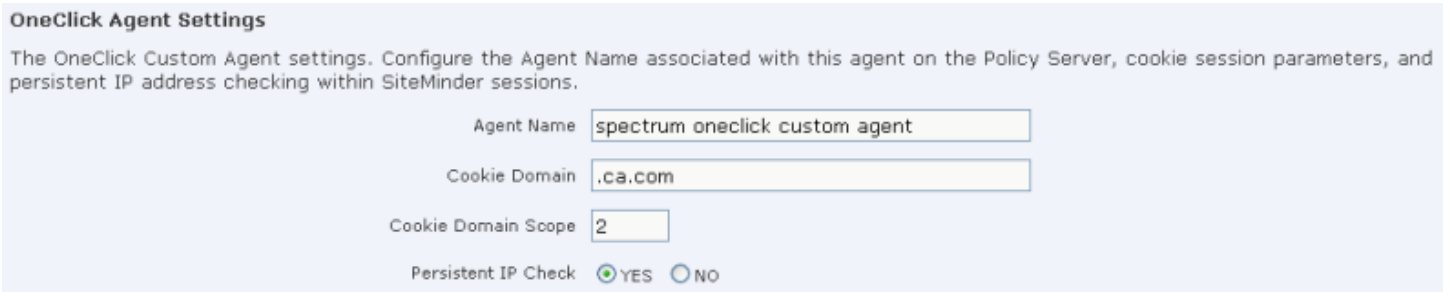
2. To remove a backup, select the server entry in the Policy Servers box, and click Remove.
   The backup entry is removed from the Policy Servers box.
3. To modify the Request Timeout interval, enter a new interval value.
4. Specify whether the Policy Server skips validation of the session IP. Validation if the IP address of the session is changed by a reverse proxy.

Configure OneClick Agent Settings

The OneClick Agent Settings panel is available on the OneClick Administration, Single Sign-On Configuration page in the SITEMINDER Configuration form. This form is available after you have successfully registered the OneClick Web Server host with the Policy Server. For more information, see Register the OneClick Web Server with the SiteMinder Policy Server.

In the OneClick Agent Settings panel, you can specify web agent configuration settings for this trusted host (the OneClick webserver).

The following image displays the OneClick Agent Settings panel:



Follow these steps:

1. From the Single Sign-On Configuration page, navigate to the OneClick Agent Settings panel and configure the following settings:
   - **Agent Name**
     Indicates the name of the DX NetOps Spectrum OneClick custom agent that is created in Policy Server.
   - **Cookie Domain**
     Indicates the cookie domain for the OneClick web server agent. Use the following format for the domain value:

     ```
     .your_company.com
     ```

     NOTE
     If you are trying to inter-operate between CA eHealth and DX NetOps Spectrum using CA EEM or CA SiteMinder®, a second-level domain or greater is required for the cookie domain.

     Cookies are restricted to a certain domain level for security reasons. According to "RFC 2901" and "RFC 2965", cookies cannot be set to a top-level domain (such as .com, .org, .gov). A minimum of the second-level domain is required. For more information, consult the RFC documentation.
     If a domain name ends with a two-letter country code, a minimum of a third-level domain is required. A cookie that is set to a second-level domain is visible at all of its third-level domains. However, a cookie that is set to a third-level domain is not visible at its parent second-level domain or at other subdomains. If no domain name is specified when a cookie is written, the cookie domain attribute defaults to the domain name where the application resides.
   - **Cookie Domain Scope**
     Indicates a cookie domain scope value. The scope determines the number of sections, which are separated by periods, that make up the domain name. Consider the following example:

- Scope = 0, the most specific scope for a given host. (*Not supported in this release.*)
- Scope = 2, .your_company.com
- Scope = 3, your_division.your_company.com

NOTE
A scope value of 1 is not allowed by the HTTP specification.

Default: 2

– **Persistent IP Check**
Enables (Yes) or disables (No) the agent to verify that a single sign-on session token originates from an IP address that differs from the IP address where it is created. If the agent detects a mismatch, it denies the session request.

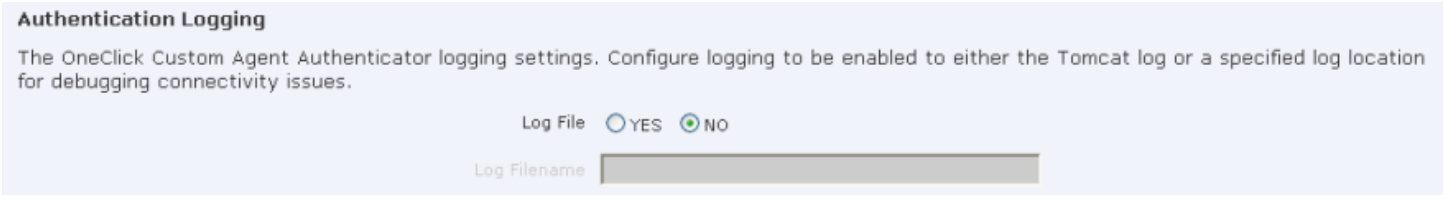
2. Click OK.
OneClick agent settings are configured.

Configure Authentication Logging

The Authentication Logging panel is available on the OneClick Administration, Single Sign-On Configuration page in the SITEMINDER Configuration form. This form is available after you have successfully registered the OneClick Web Server host with the Policy Server. For more information, see Register the OneClick Web Server with the SiteMinder Policy Server.

In the Authentication Logging panel, you can enable verbose logging of authentication and authorization activities for all authentication requests. By default, log files are written to the Tomcat log file (stdout.log on Windows, Catalina.out on Linux). However, you can specify another file and location. Log files include information that can help you troubleshoot authentication and authorization problems. For example, the log indicates whether a user was authenticated properly in the Policy Server and whether a user had the appropriate role associated with a DX NetOps Spectrum user model for the OneClick application.

The following shows the Authentication Logging panel:



NOTE
Because of the large amount of information that is written to the log file, only enable verbose logging as required for troubleshooting purposes. Do not leave it enabled for an extended period of time.

Follow these steps:

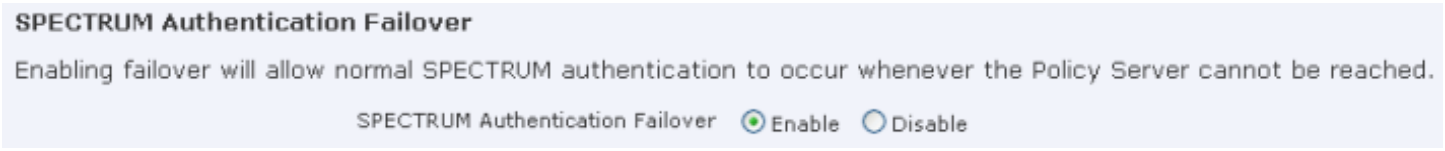
1. To enable logging to the Tomcat log, select YES.
2. To enable logging to a specific file (not the Tomcat log), select YES and enter the full log file path and the log file name in the Log Filename box.
3. To disable logging, select NO (default).

DX NetOps Spectrum Authentication Failover

The DX NetOps Spectrum Authentication Failover panel is available on the OneClick Administration, Single Sign-On Configuration page in the SITEMINDER Configuration form. This form is available after you have successfully registered the OneClick Web Server host with the Policy Server. For more information, see Register the OneClick Web Server with the SiteMinder Policy Server.

In the DX NetOps Spectrum Authentication Failover panel, you can specify whether users are able to log on to OneClick as they normally would (without Single Sign-On) if the OneClick Web Server connection to the Policy Server fails. Do not enable failover if your organization prefers for personnel to access OneClick only through Single Sign-On.

The following shows the DX NetOps Spectrum Authentication Failover panel:



NOTE
DX NetOps Spectrum user passwords may differ from those used by Policy Server to authenticate those users.

To enable authentication failover, select Enable.

User logon requests are authenticated by DX NetOps Spectrum if Single Sign-On fails. Users who have logged on to OneClick through Single Sign-On are prompted to provide DX NetOps Spectrum logon credentials when authentication failover occurs. Conversely, when a connection to the Policy Server is established, those users are prompted for Single Sign-On login credentials.

To disable authentication failover, select Disable (default).

User logon requests are not authenticated by DX NetOps Spectrum if Single Sign-On fails.

Test and Save the Integration

Options to test and save the integration configuration settings are available in OneClick. In OneClick Administration, a Single Sign-On Configuration page is included in the SITEMINDER Configuration form. This form is available after you have successfully registered the OneClick Web Server host with the Policy Server. For more information, see Register the OneClick Web Server with the CA SiteMinder® Policy Server.

Execute an integration test after you have set integration parameters and before you save and enable the integration. The test determines whether you configured the connection parameters correctly and the test username and test password match an entry in the domain. If the test fails, you can reconfigure parameters and re-test. After you have successfully tested the settings, you can save them, automatically enabling the integration.

The following shows the Test and Save options at the bottom of the SITEMINDER Configuration form:

Test the integration settings from OneClick Administration.

Follow these steps:

1. Enter a username and password from the Policy Server domain user directory in the Test Username box and Test Password field. CA SiteMinder® administrators manage the user directory and provide the password for Single Sign-On authentication.

   NOTE
   The username in the Policy Server user directory must match the username created in the applications (DX NetOps Spectrum, CA eHealth) that the user plans to access.
2. Click Test.

If the test is successful, the following message appears:
"Successfully established connection with the Policy Server"

You can also save the integration settings.

Follow these steps:

1. Click Save to save the integration settings.
   You are prompted to restart Tomcat.
2. Click OK to restart Tomcat to apply the configuration.
   SiteMinder single sign-on integration in OneClick is enabled.

Disable or Re-enable the Integration

When you initially configure the CA SiteMinder® integration settings, you automatically enable the integration upon saving. You can also disable and re-enable the integration if necessary. When you disable the integration, users are authenticated by the integrated applications that they access from OneClick.

Disable the CA SiteMinder® integration on the Single Sign-On Configuration page in OneClick Administration.

Follow these steps:

1. Select 'No Single Sign-On' and click Save.
   You are prompted to restart Tomcat.
2. Click OK to restart Tomcat and disable the CA SiteMinder® integration.

When the integration is disabled, you can re-enable it.

Follow these steps:

1. On the OneClick Administration, Single Sign-On Configuration page, select the SITEMINDER option.
   The SITEMINDER Configuration form opens.
2. Verify that the existing settings are correct and click Save.
   You are prompted to restart Tomcat.
3. Click OK to restart Tomcat and apply the CA SiteMinder® integration configuration.

Troubleshooting the Integration Related Issues

This section provides solutions to problems you may encounter with the integration. If you cannot find a solution in this section to your particular problem or you need additional assistance, contact Technical Support.

Debugging Options

OneClick provides multiple debugging options that can help you pinpoint problems with the integration.

- Users cannot log in to the OneClick home page.
  You can enable logging of authentication activities to the Tomcat log (stdout.log on Windows, catalina.out on Linux) using the Authentication Logging option in Single Sign-On Configuration. Authentication logging indicates whether Policy Server is denying a user access to DX NetOps Spectrum OneClick or if the user role is not being retrieved from DX NetOps Spectrum.
- The OneClick web server cannot connect to Policy Server during a single sign-on configuration.

You can enable logging of information about integration parameters to the Tomcat log (stdout.log on Windows, catalina.out on Linux) by enabling the Web Server Debug Page (Runtime)/Single Sign-On Integration option available from the Debugging link on the Administration page.

- Particular users cannot access the OneClick Console from the OneClick home page without being prompted for credentials.
  You can enable the Debug Console for Single Sign-On in the OneClick Console. It provides information about single sign-on token recognition. You can also enable this option directly in the oneclick.jnlp file.

How to disable the EEM integration with Spectrum

Perform the following steps to disable EEM and Spectrum integration:

1. Have all users log out from OneClick.
2. Shut down tomcat on the OneClick server.
3. On the OneClick web server, in the **$SPECROOT/custom/ directory** rename the SSO directory to sso.bak.
4. Edit the web.xml in the directory **$SPECROOT/tomcat/webapps/spectrum/WEB-INF** as follows**:**
   a. Change the following parameters:
   ```
   <login-config>
   <auth-method> EXTERNALSSO </auth-method>
   <realm-name>SPECTRUM</realm-name>
   </login-config>
   ```
   to:
   ```
   <login-config>
   <auth-method> BASIC </auth-method>
   <realm-name>SPECTRUM</realm-name>
   </login-config>
   ```
   b. Comment this entry out that should show at the top of the file:
   ```
   <listener>
   <listener-class>com.aprisma.tomcat.authenticator.ExternalSSOAuth</listener -class>
   </listener>
   ```
   to make it:
   ```
   <!--
   <listener>
   <listener-class>com.aprisma.tomcat.authenticator.ExternalSSOAuth</listener-class>
   </listener>
   -->
   ```
5. Save the changes that are made to the **$SPECROOT/tomcat/webapps/spectrum/WEB-INF/web.xml file.**
6. Go to the **$SPECROOT/tomcat/conf/context.xml** file and add the following:
   ```
   <Valve className="org.apache.catalina.authenticator.BasicAuthenticator"
    changeSessionIdOnAuthentication="false" /></Context>
   ```
   and comment out the following:
   ```
   <Valve
    className="com.aprisma.tomcat.authenticator.ExternalSSOAuth"changeSessionIdOnAuthentication=
   ></Context>
   ```
   So that it looks like:
   ```
   <!--
   ```

```
<Valve
 className="com.aprisma.tomcat.authenticator.ExternalSSOAuth"changeSessionIdOnAuthentication=
></Context>
-->
```

7. Save the file changes to the **$SPECROOT/tomcat/conf/context.xml** file.
8. Start the OneClick Tomcat web server and attempt to log in with a non-ldap account.

Re-Registering the OneClick Web Server

In some troubleshooting scenarios (current registration is invalid, for example), you may have to remove the current registration and re-register the OneClick web server with the CA SiteMinder® Policy Server.

Follow these steps:

1. Stop Tomcat.
2. Remove the Trusted Host Name for the server from the Policy Server.

   NOTE
   For more information, see the Policy Server Help.
3. Remove the *$SPECROOT*/custom/sso directory.
4. Specify DX NetOps Spectrum OneClick Authentication in the web.xml file.
5. Restart Tomcat.
   Register the webserver.

See also:
Specific Problems and Solutions

Specific Problems and Solutions

This section describes specific problems with DX NetOps Spectrum SiteMinder integration and recommended procedures for solving the problems.

Host Registration Fails

Valid on Linux and Windows

Symptom:

When you test the registration, an error message indicates that the OneClick Web Server was unable to connect to the Policy Server.

When you test the registration, an error message indicates that the OneClick Web Server was able to connect to the Policy Server, but the login credentials were invalid.

Solution:

If you received an “Invalid credentials” message, make sure the username and the password you specified in Single Sign-on Configuration are correctly configured in CA SiteMinder®. Consult the SiteMinder/Policy Server administrator for assistance.

If you received an “Unable to connect” message, verify that the Policy Server is up and running and then check settings in Single Sign-on Configuration and Policy Server.

Single Sign-on Configuration:

- Verify that the Policy Server IP Address and Policy Server Port settings are correct.
- Verify that the Policy Server Admin Username and Policy Server Admin Password settings are correct. Also, verify that the credentials provide administrative privileges.

Policy Server:

- Verify that the DX NetOps Spectrum OneClick host configuration object has been correctly created on the Policy Server.
- Verify that the name you specified in Single Sign-On Configuration for the Trusted Host Name setting is not a duplicate of a name that is already included as a Trusted Host in the Policy Server. If it is a duplicate, use another name or delete the name in Policy Server and retry the registration.

Authentication Server (Policy Server) Cannot Be Contacted

Valid on Linux and Windows

Symptom:

An error message states that the authentication server cannot be contacted when a user attempts to invoke the OneClick Console.

Solution:

- Verify that the Policy Server is up and running.
- If the message mentions that the user failed to authorize, do the following:
  - On the Single Sign-On Configuration page, set Log File to “YES” and Save.
  - Look at the Tomcat log after the user attempts to log on. The log indicates whether the Policy Server is denying the user or if the user role is not being retrieved from DX NetOps Spectrum.

Prompted for DX NetOps Spectrum Logon Credentials When Invoking OneClick Console

Valid on Linux and Windows

Symptom:

Even though the integration is enabled, the user is prompted for a DX NetOps Spectrum username and password when attempting to start the OneClick Console from the OneClick home page.

Solution:

An error might have occurred in the communication of Single Sign-On parameters to OneClick from the webserver. To display how Single Sign-On information is being transferred, enable OneClick to display the java debug console.

Follow these steps:

1. Edit the JNLP file (located at *$SPECROOT*/tomcat/webapps/spectrum/oneclick.jnlp).
2. Find the following line:
   ```
   <!--<argument>-debug Poller=on</argument> -->
   ```
3. Add the following line below it:
   ```
   <argument>-debug SSOConsoleDebug=on</argument>
   ```
4. Launch into the OneClick Console with the java debug console displayed.
   If you detect that the -ssoToken parameter is not being passed or it does not have a value associated with it, there is a problem with how your Single Sign-On cookies are being written. Make sure your cookie settings (located in the OneClick Agent Settings area of the Single Sign-On Configuration administration page) coincide with how you are accessing your OneClick web server.

Example: Incorrect Cookie Setting Causes Authentication Problem

The following is an example of an incorrect cookie setting that would produce the authentication problem:

- In the OneClick Agent Settings page, the cookie domain is set to ".ca.com" and the cookie scope is set to "2". This means that cookies will be written to.ca.com in the browser.
- You access your web server using http://someuser/spectrum in your URL. This violates the cookie settings because "someuser" is out of scope with the.ca.com domain. Instead, you should use http://someuser.ca.com/spectrum to access your web server.

Cannot Access the Disable Integration Option in Single Sign-On Configuration

Valid on Linux and Windows

Symptom:

You cannot access the Single Sign-On Configuration page and disable the integration.

Solution:

The OneClick access authorization method is specified in the web.xml file. You can disable the integration (and restore standard DX NetOps Spectrum login access) by editing the <auth-method> element in the file.

Application Web Agents Ignore OneClick Single Sign-On Tokens

The OneClick implementation of CA SiteMinder® uses its own customized agent, rather than an installed web server agent, to communicate with the SiteMinder policy server. If you are sharing single sign-on tokens between OneClick and installed web agents of other applications, you may need to modify the agent configuration of each installed web agent to recognize, or not ignore, OneClick agent tokens.

Set the following web agent parameter from NO to YES for the other applications:

```
AcceptTPCookie=YES
```

Integration with DX APM

> **NOTE**
>
> Starting from the 10.3 release, the **Spectrum SNMP Trap** plugin is deprecated and will not be shipped with DX NetOps Spectrum. However, you can use the **SNMP Alert Action** plugin of DX APM for the trap generation between DX NetOps Spectrum and DX APM.
>
> The **Spectrum SNMP Trap** plugin of DX NetOps Spectrum and **SNMP Alert Action** feature of DX APM both are used for the same purpose (for trap generation). Since, the **SNMP Alert Action** feature of DX APM has more enhanced functionality, we recommend you to use the **SNMP Alert Action** in DX APM for the trap generation.
>
> To configure the SNMP Alert Action plugin, see the Create and Configure Notification Actions section. To migrate from **Spectrum SNMP Trap** to **SNMP Alert action**, see Migrate from Spectrum SNMP Trap to SNMP Alert Action of DX APM.

Integration Prerequisites

Beginning with version 9.1 of DX Application Performance Management (DX APM), the name "Wily" was dropped from the CA Wily Introscope® product. This section retains the name where it helps to clarify compatibility with previous versions of DX APM.

For DX NetOps Spectrum and DX APM integration information, see the Integration Compatibility page.

How the Integration Works

DX Application Performance Management (DX APM) consists of an Enterprise Manager and one or more CA Introscope agents. Introscope agents are installed on servers running Java virtual machines (JVMs). The agents deliver information about numerous metrics, such as servlet response time and report metrics, to the Enterprise Manager.

Enterprise Manager uses a Management Module to organize and report on the metric data provided by an agent. By configuring thresholds on reported metrics, Enterprise Manager creates an alert when a threshold is violated and clears the alert when the threshold is no longer violated. Introscope agents are polled every 60 seconds to determine whether a threshold has been violated or whether a violated threshold has been cleared. An alert is created when a threshold event occurs.

DX NetOps Spectrum models the CA Introscope infrastructure using two model types, the IntroscopeAdmin model and the WilyAgent model.

- The IntroscopeAdmin model represents one Enterprise Manager.
- The WilyAgent model represents an application container (JVM or CLR). When you initiate a Discovery, or when a Discovery is initiated when you restart Tomcat, DX NetOps Spectrum requests a list of application containers from the Enterprise Manager and models them as WilyAgent models.

When CA Introscope generates an alert, data about the metric and its threshold is forwarded to DX NetOps Spectrum. An alarm is generated on a WilyAgent model. If the alert is cleared, DX NetOps Spectrum is notified and clears the associated alarm.

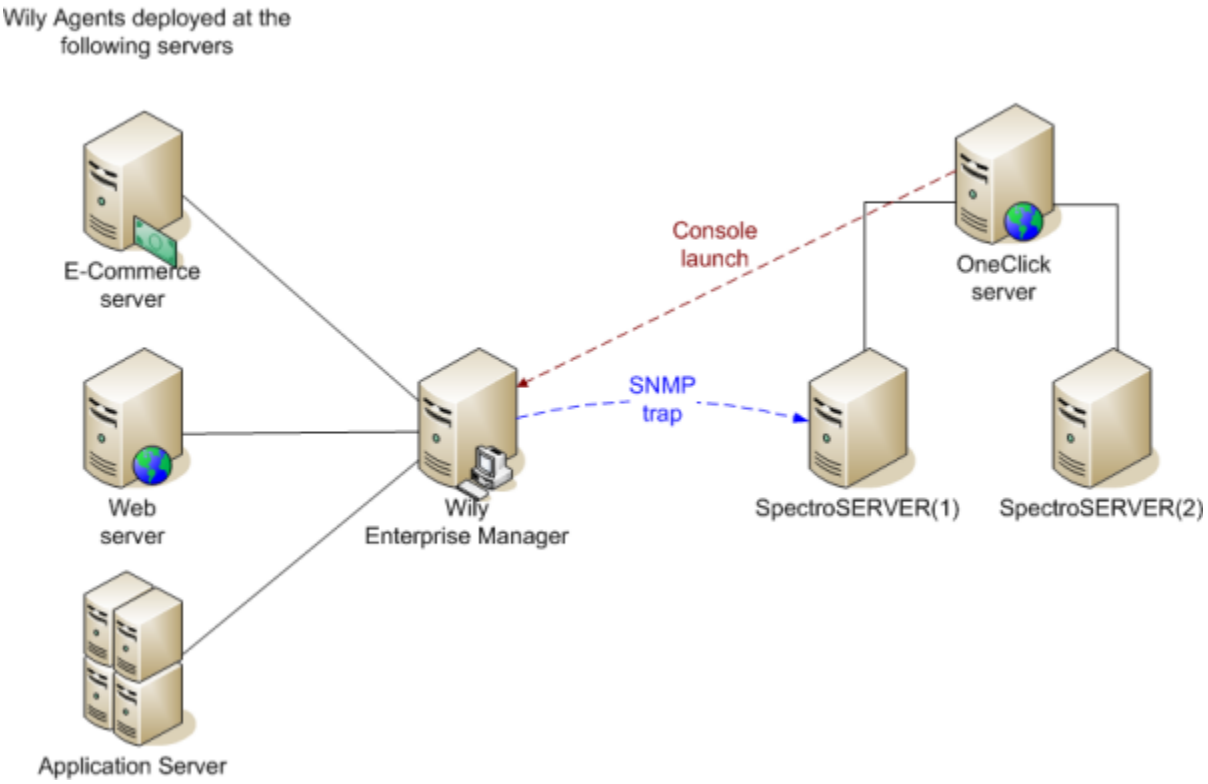
The DX NetOps Spectrum integration with DX APM also actively monitors the inventory of application containers on an Enterprise Manager.

- When an Enterprise Manager starts to monitor a new application container, DX NetOps Spectrum is notified and automatically creates a WilyAgent model to represent the application container.
- When an Enterprise Manager no longer monitors a discovered application container, DX NetOps Spectrum generates an alarm on the corresponding WilyAgent model.

> **NOTE**
> If you move an Introscope agent from one Enterprise Manager to another, a new IntroscopeAgent model is created and associated with the IntroscopeAdmin model. That model represents the Enterprise Manager to which the agent was moved. This activity results in two identical IntroscopeAgent models associated with two distinct IntroscopeAdmin models. Destroy the original IntroscopeAgent model so that the new IntroscopeAgent model receives all subsequent events.

Integration Architecture

The following diagram depicts the architecture of the DX APM and DX NetOps Spectrum integration, and identifies the direction of data transfers.



Integration Considerations

Review the following considerations:

- The integration combines web services with trap notifications:
  - A traditional axis 1.4 polling web service that provides inventory information, such as the Management Module, the agent, alert definitions, alerts, and Management Module/Agent pairs that help determine the DX NetOps Spectrum model. DX NetOps Spectrum uses only the agent information.
  - A bi-directional subscription web service, introscope-wssdk-consumer, uses the Apache Muse framework, which is deployed inside *$SPECROOT*/ tomcat/webapps. The introscope-wssdk-consumer web service listens for and processes asynchronous updates from CA Introscope.
- Trap notifications require the installation of a trap generation plugin on the CA Introscope Enterprise Manager.
- All communication between DX NetOps Spectrum and CA Introscope passes through the Tomcat web server for agent inventory.
- The integration can be enabled on only one server.
- In a distributed SpectroSERVER environment, designate a OneClick server as the integration server.

Maintenance Mode

You can put WilyAgent models into maintenance mode. However, the status of a WilyAgent model is dependent on event updates that the Enterprise Manager sends through SNMP traps. Therefore, before putting WilyAgent models into maintenance mode, consider the following:

- If the WilyAgent model is in an alarm state, and the Enterprise Manager posts an alarm clear update during maintenance mode, the model remains in an alarm state when it comes out of maintenance.
- If the WilyAgent model is in a normal state, and its counterpart on the Enterprise Manager enters an alarm state during maintenance mode, the model remains in a normal state when it comes out of maintenance.

How to Integrate with DX APM

Contents

The following sections describe the process of installing and configuring the integration of DX APM and DX NetOps Spectrum:

Install the Trap Generation Plugin on the Enterprise Manager

> **NOTE**
>
> Starting from the 10.3 release, the **Spectrum SNMP Trap** is deprecated and will not be shipped with DX NetOps Spectrum. However, you can use the **SNMP Alert Action** plugin of DX APM for the trap generation between DX NetOps Spectrum and DX APM.
>
> The **Spectrum SNMP Trap** plugin of DX NetOps Spectrum and **SNMP Alert Action** feature of DX APM is used for the same purpose (for trap generation). Since, the **SNMP Alert Action** feature of DX APM has more enhanced functionality, we recommend you to use the **SNMP Alert Action** in DX APM for the trap generation.
>
> To configure the SNMP Alert Action plugin, see the Create and Configure Notification Actions section. To migrate from **Spectrum SNMP Trap** to **SNMP Alert action**, see Migrate from Spectrum SNMP Trap to SNMP Alert Action of DX APM.

Create and Configure Notification Actions

Contents

> **NOTE**
>
> Starting from the 10.3 release, the **Spectrum SNMP Trap** is deprecated and is not shipped with the installer. However, you can use the **SNMP Alert Action** plugin of DX APM for the trap generation between and DX APM.
>
> The **Spectrum SNMP Trap** plugin of and **SNMP Alert Action** feature of DX APM both is used for the same purpose (for trap generation). Since, the **SNMP Alert Action** feature of DX APM has more enhanced functionality, we recommend you to use the **SNMP Alert Action** in DX APM for the trap generation.
>
> To migrate from **Spectrum SNMP Trap** to **SNMP Alert action**, see Migrate from Spectrum SNMP Trap to SNMP Alert Action of DX APM.

Use the DX APM Workstation to create and configure notification actions for the alert that forwards data to . Configure the SNMP Alert Action Plugin of DX APM allows the APM Catalyst Connector to get Introscope alert data and supply it to DX NetOps Spectrum.

Create SNMP Alert Action

Follow these steps:

1. Open the DX APM Management Module Editor.
2. Select **Elements**, **New Action**, **New SNMP Alert Action**.
3. Provide a name for the new action and select the Management Module that contains the alert that forwards data to .
4. Click OK.
   The new SNMP alert action gets created.
5. Select the **Active** checkbox to activate the action.

6. Complete the details for **SNMP Destination** and **Introscope** in the **SNMP Trap Configuration** panel:
   SNMP Destination:
   - **Host IP**. The IP address of the SpectroSERVER.
   - **Trap Port**. The port number of the SNMP trap for the SpectroSERVER. The default value is '162'.
   - **Community**. the SNMP community string for the SpectroSERVER. The default value is 'public.'

   Introscope:
   - **EM/MOM Host IP**. The IP address of an Enterprise Manager or the MOM Enterprise Manager in a cluster environment.
     - Only IPv4 is supported.
     - The Host IP address must be set to the same as the Enterprise Manager IP address.
   - **WebView Protocol**. The port number of the Enterprise Manager WebView. Select one of the following protocols:
     - Http
     - Https
   - **WebView Host IP**. The IP address of the Enterprise Manager WebView.
   - **WebView Port**. The port number of the Enterprise Manager WebView. The default value is '8080'.
   - **Management Module**. The name of the Management Module that contains the alert that forwards data to . This is the same Management Module that you selected in step 3.
   - **Dashboard Name**. The name of the dashboard for the Alarm/Event launches back.
   - **Domain Name**. The domain name (or Superdomain name) that corresponds with the dashboard name that you specified.
7. Click **Apply**.
   The trap action is configured.
8. Click **Test** to verify the communication between the Enterprise Manager and the APM connector.
   The SNMP action alert configuration is set.

For more detailed configuration details, see the Create an SNMP Alert Action section in the DX APM documentation.

Alerts that are associated with the trap actions must have their 'Notify by Individual Metric' check box selected. You can create Management Modules and Dashboards and Alerts. For more information, see the CA Introscope documentation.

Migrate from Spectrum SNMP Trap to SNMP Alert Action of DX APM

If you are already using the the **Spectrum SNMP Trap** plugin for trap generation, you can migrate to **SNMP Alert Action** of DX APM using the following steps:

1. Login to DX APM.
2. Navigate to **Workstatiion**, **Management Module**.
3. Select the management module that you want to edit.
4. Expand the **Alerts** of selected management module.
5. Select the alert for which you want to assign the SNMP Alert Action. The right side window displays, the **Configuration** and **Settings** details.
6. Select the Spectrum SNMP Trap action that is already applied to the selected alert. Click **Remove**.
7. Click **Add** to assign the **SNMP Alert Action** for the selected alert. The **Choose Action** window appears. This window allows you to either to choose an already created SNMP Alert Action or create a new action using the **New Action** button. For creating new SNMP Alert Action, see the Create SNMP Alert Action section.
8. Choose an SNMP Alert Action from the list of actions.
9. Click **Choose**. The SNMP alert action is associated with the selected alert.

NOTE

For further help regarding the migration contact the DX APM Support team.

Configure Web Services on the Enterprise Manager

You can configure web services on the Enterprise Manager. For more information, see DX APM Introscope Web Services .

Designate the Integration Server and Enable the Integration

Designate an OneClick Tomcat server as the integration server host. You can designate the server you use to access OneClick or you can designate a headless server, which is dedicated to processing DX APM data. You can designate any OneClick Tomcat server within the distributed environment. Therefore, select a server that can accommodate the extra load of the DX APM data.

Important: The DX APM integration becomes disabled when you designate a headless server as the integration server host and then stop and restart an OneClick Tomcat server that is not the integration server host. In such a situation, re-enable the integration.

You designate the integration server and enable the integration at the same time.

Follow these steps:

1. Select Administration from the OneClick home page.
   The OneClick Administration page opens.
2. Select APM Integration Configuration from the left panel.
   The APM Configuration page opens.
3. Complete the fields as follows:
   - **Integration Server Host Name**. The integration server hostname.
   - **Integration Server Port**. The port number for the integration server.
4. Select Enabled in the APM Introscope Integration field.
5. Click Save.
   The Successfully saved configuration message appears.

   NOTE
   You can disable the integration at any time.

Follow these steps:

1. Select Administration from the OneClick home page.
   The OneClick Administration page opens.
2. Select APM Integration Configuration from the left panel.
   The APM Configuration page opens.
3. Select Disabled in the APM Introscope Integration field.
4. Click Save.
   The Successfully saved configuration message appears.

   NOTE
   Disabling the integration disables all IntroscopeAdmin models in a Distributed SpectroSERVER (DSS) environment. DX NetOps Spectrum requires a single IntroscopeAdmin model modeling a single Enterprise Manager in a DSS environment.

Create IntroscopeAdmin Models

After you enable the integration, create an IntroscopeAdmin model to represent the connection to an Enterprise Manager. The Enterprise Manager can be modeled in any landscape in a Distributed SpectroSERVER environment.

NOTE
You can create an IntroscopeAdmin model in each landscape in a distributed SpectroSERVER environment to monitor a singular Enterprise Manager. However, only a single IntroscopeAdmin model can monitor a single Enterprise Manager in a distributed SpectroSERVER environment. Disabling the integration disables all IntroscopeAdmin models.

Follow these steps:

1. Select the Universe subview from the OneClick Navigation panel and select the Topology tab from the Contents panel.
   The OneClick Topology view is displayed in the Contents panel.
2. Click the Model by Type icon

   The Select Model Type dialog opens.
3. Select IntroscopeAdmin from the list, and then click OK.
   The 'Create Model of Type IntroscopeAdmin' dialog opens.
4. Enter the unique Model name and network IP address of the Introscope host system, and then click OK.
   The IntroscopeAdmin model is created.

Discover Introscope Agents

The final step in the configuration process is for DX NetOps Spectrum to discover the Introscope agents that are configured to send information to DX NetOps Spectrum.

Follow these steps in DX APM:

The Spectrum Discovery process for the integration finds and creates the Management Modules as DX APM Agent models in Spectrum. If there are no Management Modules present in DX APM EM, the discovery process completes successfully but no DX APM Agent models are created in DX NetOps Spectrum.

1. Add the following parameter in the 'IntroscopeEnterpriseManager.properties' file, which is located in the '<EM_Home>/config' directory to add the Management Modules in DX APM EM.
   'introscope.alerts.extension.managementmodules.enable=ALL'

   NOTE

   - If the parameter exists, change its value to 'ALL'.
   - The value 'ALL' must be in all capital letters else it is not acknowledged.
2. Restart the EM.

Follow these steps in DX NetOps Spectrum:

1. Select the Universe subview from the OneClick Navigation panel and select the IntroscopeAdmin model.
   Information about the IntroscopeAdmin model is displayed in the Contents panel.
2. Click the Information tab.
   IntroscopeAdmin configurations and information are displayed.
3. Expand the Introscope Integration Administration node.
   Integration configurations are displayed.
4. Verify the settings.
   To change a setting, click the 'set' link next to the setting, enter the appropriate information in the field, and press Enter.
5. When your settings are correct, click Discover Agents.
   The Introscope agents are discovered, and the Discovery status is displayed in the Status window.

Alarms, Events, and Application Statistics

View Introscope Alarms and Events

You can view Introscope alarms and events in OneClick.

NOTE
For more information, see Modeling and Managing Your IT Infrastructure .

Follow these steps:

1. Launch OneClick.
2. Expand the Universe subview in the Navigation panel, and select the IntroscopeAdmin model.
   Information about the IntroscopeAdmin model is displayed in the Contents panel.
3. Click the Alarms tab to display alarms or click the Events tab to display events.
   Introscope alarms or events are displayed in the Contents panel. Information about the agent that caused the event or alarm is displayed in the Component Detail panel when you select an alarm or event in the Contents panel.
4. (*Optional*) Launch the Introscope Dashboard to view more information about the alert in the DX APM interface:
   a. Select an alarm.
   b. Click the URL on the Alarm Detail tab.
      The Introscope Dashboard launches.

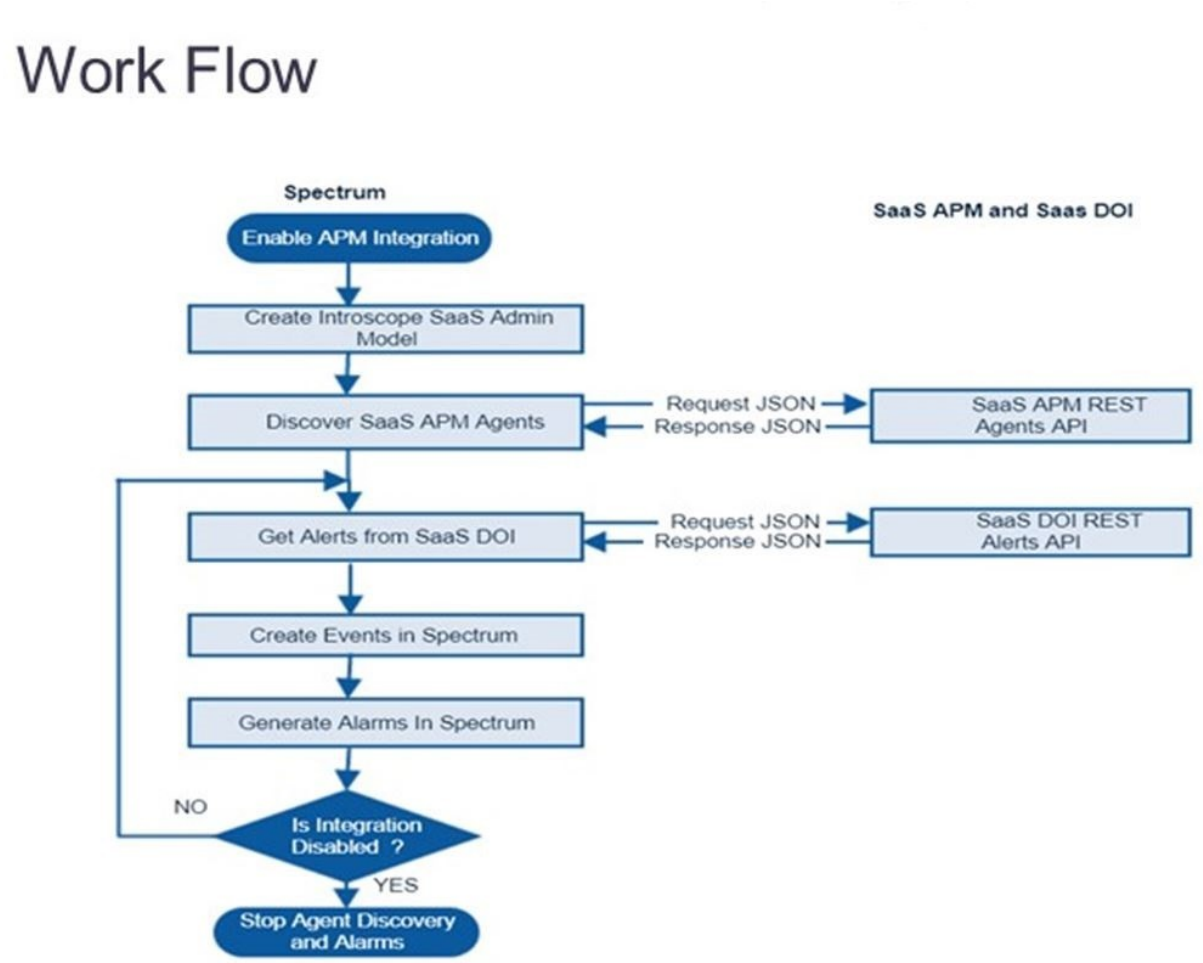
Support for DX NetOps Spectrum Integration with APM SaaS and DXI

About the Integration

With Spectrum 10.3.1, users can now leverage the new APM SaaS integration! The CA APM (SaaS) integration is available through the CA Digital Experience Insights (DXI) platform. With this integration, users can now, discover models/agents from APM SaaS, and get APM alerts via SaaS DOI. APM and (Digital Operational Intelligence) DOI are part of SaaS-enabled DXI. The Spectrum with APM SaaS and DXI platform enabled solution, address key challenges for customers such as managing APM agents in SaaS, discovering APM, inventory/models from SaaS, and fetching the APM alerts from SaaS.

How the Integration works

Following is the screenshot of the APM SaaS integration works:

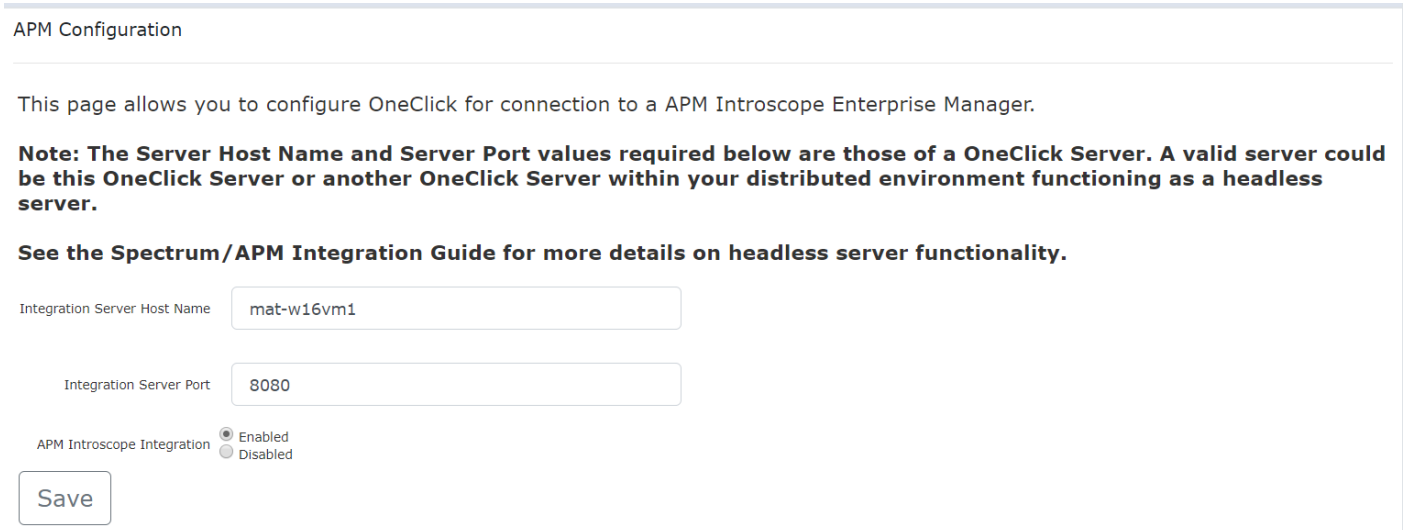


Generate APM SaaS Security Token

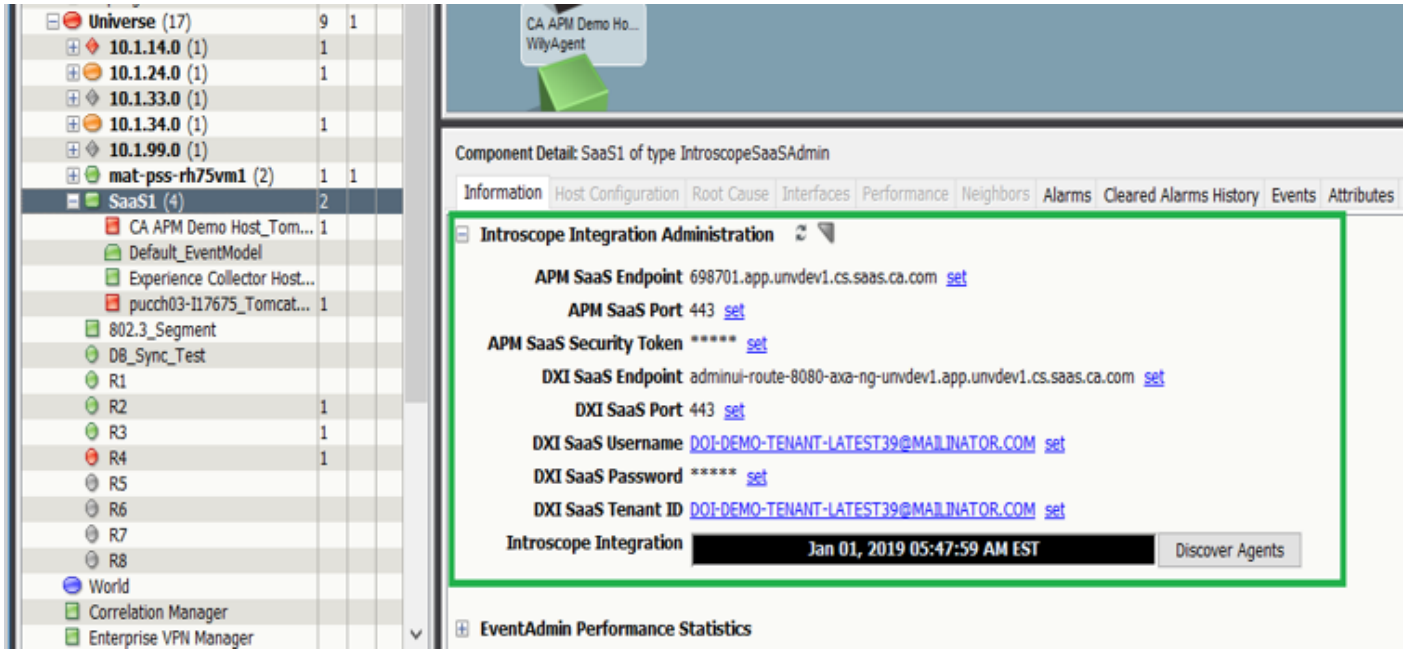
A security token is a randomly generated text string and is roughly equivalent to a text password. This token gives the API access to the APM Web Service. Connect the Enterprise Manager to Command Center. Generate a security token in Team Center. For instructions to generate the token, see the Generate a Security System Token in Team Center section in the CA APM documentation.

Spectrum and APM SaaS Configuration

1. Ensure that the DX NetOps Spectrum and CA APM integration is enabled. Enable the integration from the **OneClick Administration** page. Only after enabling the APM integration you find the configFile **apm-saas-config.xml** created and at **$SPECROOT\custom\wily\config** .This file allows you to configure the polling interval and throttle count values. By default, the value for the polling interval is configured as a maximum of 60 seconds and a minimum value of 15 seconds. For the throttle count, the maximum value is 500.
   For instructions, see Designate the Integration Server and Enable the Integration.



2. Create IntroscopeSaaSAdmin Model from the Topology with details of the APM SaaS Endpoint and the DXI SaaS Endpoint.
   For instructions, see the *Create IntroscopeSaaSAdmin Model* section on this page.



3. From the Introscope Integration Administration window, discover the Introscope Agents and provide details such as the:
   - APM SaaS Security token
   - Port
   - DXI SaaS Port
   - Username
   - Password
   - Tenant ID

   NOTE
   For instructions, see the *Discover Introscope Agents* section on this page.

Create IntroscopeSaaSAdmin Model

After you enable the integration, create an IntroscopesaasAdmin model to represent the connection to an Enterprise Manager. The Enterprise Manager can be modeled in any landscape in a Distributed SpectroSERVER environment.

> **NOTE**
>
> You can create an IntroscopeSaaSAdmin model in each landscape in a distributed SpectroSERVER environment to monitor a singular Enterprise Manager. However, only a single IntroscopeSaaSAdmin model can monitor a single Enterprise Manager in a distributed SpectroSERVER environment. Disabling the integration disables all IntroscopeSaaSAdmin models.

Follow these steps:

1. Select the **Universe** subview from the OneClick **Navigation** panel and select the **Topology** tab from the **Contents** panel.
   The OneClick Topology view is displayed in the **Contents** panel.
2. Select the Model by Type icon

   
3. Select **IntroscopeSaaSAdmin** from the list, and then select **OK.**
   The 'Create Model of Type IntroscopeSaaSAdmin' dialog opens.
4. Enter the unique Model name.
5. Provide APM SaaS Endpoint and DXI SaaS Endpoint details then select **OK**.
   The IntroscopeSaaSAdmin model is created.

Discover Introscope Agents

Follow these steps in DX NetOps Spectrum to discover the Introscope agents.

1. Select the **Universe** subview from the OneClick **Navigation** panel and select the **IntroscopeSaaSAdmin** model.
   Information about the IntroscopeSaaSAdmin model is displayed in the **Contents** panel.
2. Select the **Information** tab in the **Component Detail** panel.
   IntroscopeSaaSAdmin configuration is displayed.
3. Expand the Introscope Integration Administration node and verify the following settings:
   - Protocol (introduced in 10.4.2.1)
     Supports both HTTP and HTTPS (Default).
   - APM SaaS Endpoint
   - APM SaaS Port – (default-443)
   - APM SaaS Security Token
   - DXI SaaS Endpoint
   - DXI SaaS Port – (default-443)
   - DXI SaaS User Name - User name that is used for DXI login
   - DXI SaaS Password - User password that is used for DXI login
   - DXI SaaS Tenant ID - Provide DXI SaaS Tenant Name used to fetch alarms
4. To change a setting, select the **Set** link next to the setting, enter the appropriate information in the field, and press **Enter**.
5. Select **Discover Agents**. If there is no alarm/error on the model "IntroscopeSaaSAdmin" then the configuration is complete.
   The Introscope agents are discovered and modeled in DX NetOps Spectrum.

How the Sync Works

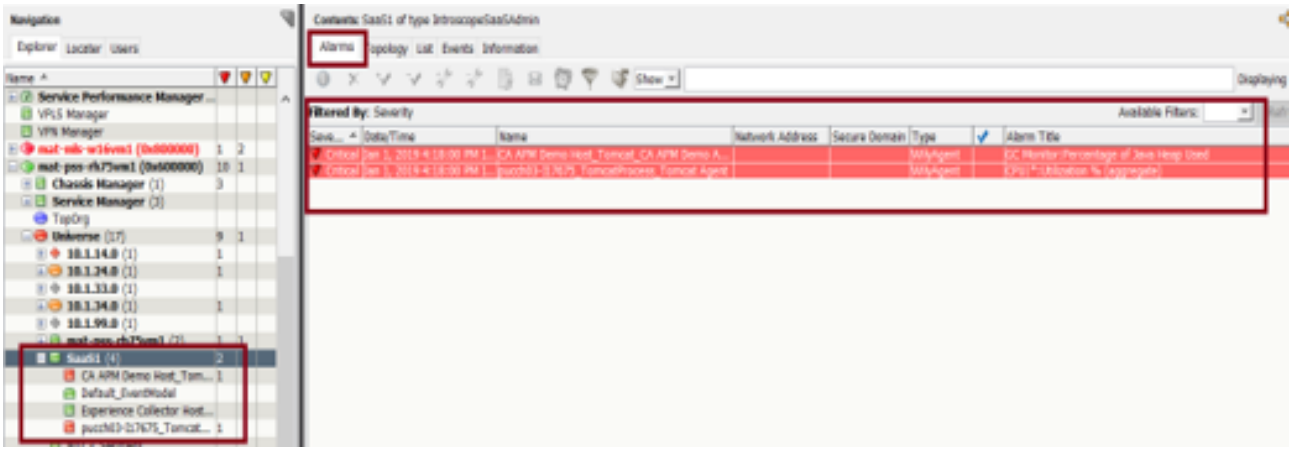
Once the APM Inventory sync and the Alert sync start, you can see the APM Inventory/Models/Agents under the model type 'IntroscopeSaaSAdmin'. If there are any APM alerts that are raised, those can be seen on the models. To perform a full sync for alerts select the Discover Agent under the Information tab in the OneClick. You can also run the discover agents on demand and OneClick server startup. Incremental Sync, that are APM alerts based on the polling interval is specified in the **apm-saas-config.xml** and is applicable for only alerts sync and not the inventory sync.

View Introscope Alarms and Events

You can view Introscope alarms and events in Spectrum OneClick, follow these steps:

1. Launch the OneClick.
2. Expand the **Universe** subview in the **Navigation** panel, and select the **IntroscopeSaaSAdmin** model. Information about the IntroscopeSaaSAdmin model is displayed in the **Contents** panel.
3. Select the **Alarms** tab to display alarms or select the **Events** tab to display events. The CA Introscope alarms or events are displayed in the **Contents** panel. Information about the agent that caused the event or alarm is displayed in the **Component Detail** panel when you select an alarm or event in the **Contents** panel.

Following is the screenshot of Alarms being generated in the IntroscopeSaaSAdmin model:



Troubleshooting

For the following error messages these are the recommended actions you need to take as a Spectrum user:

1. Error Message while running discovery 'Unable to contact APM Endpoint' The recommended action, if you cannot communicate to APM SaaS Endpoint is to verify the configuration for the Endpoint, Security Token, and the Port.
2. Error Message 'Unable to contact DXI Endpoint' The recommended action if you cannot communicate to DXI SaaS Endpoint is to verify the configuration for the Endpoint, Tenant ID, Username, Password, and the Port.

Troubleshooting Tips

Consider or check the following for potential issues:

- “Discover Agents” button is disabled
- Alarms that are created on the APM inventory model and IntroscopeSaaSAdmin model that is related to the connection lost.
- Alerts not syncing into Spectrum from APM/DOI

To trace and correct faults in the system:

- Ensure that APM integration is enabled in the OneClick Administration page
- Check for the tomcat log for connection loss and see the recommended action under the Troubleshooting section.
- If the alerts are not generated in APM, then verify the same in the APM configuration.
- If alerts do not pass in DOI, then verify the same in the APM/DOI configuration.
- If alerts do not pass in Spectrum, then enable the APM Integration debug logs and verify in the tomcat logs whether Alerts are actually coming from APM/DOI.

Common Access Card Authentication

Within secure environments, the use of a single point of entry is required for easy access management. Without a single point of entry, administrators of secure environments must manage several programs with different security levels and requirements in addition to user access. Common Access Cards (CACs) provide a single point of entry by requiring the use of the CAC for access to all controlled resources.

The following topics are covered under this section:

How CACs Work

Each CAC contains certificates that are issued by certificate authorities. A managing authority, such as the ActivIdentity® ActivClient®, controls access to the CAC certificates. Managing authorities prevent the certificates that are contained within the CACs from being used outside of the CAC without first verifying the CAC owner.

The managing authority verifies the card owner by prompting the user of the CAC for a Personal Identification Number (PIN). If the PIN is verified, the managing authority allows access to the CAC Certificates.

The presentation of a CAC Certificate, however, is not sufficient to gain access to a resource. The CAC Certificate must first be checked for both authenticity and validity.

Certificate authenticity is checked by building a certificate path from the CAC Certificate through any intermediate certificates to a trusted root certificate. A chain of trust thus links the user certificate to the trusted root certificate. If this chain is properly built, the certificate is authentic. Certificate authorities provide intermediate and root certificates to administrators for this purpose.

Once a certificate is verified as authentic, validity checks are required. Certificate authorities can revoke CAC Certificates. Revocation invalidates the certificate, which renders the CAC that holds the certificate useless. This verification can be accomplished in *one* of the following two ways:

- Certificate Revocation Lists (CRLs) are the most common way to verify that a CAC is valid and are the industry standard.
  CRLs are flat files that contain the serial numbers of revoked certificates. They become outdated frequently because of constant additions. As a result, they expire at predetermined times and must be refreshed. CRLs also consume a great deal of memory and must be placed on the local file system. Generally administrators choose this option when they have no access to an OCSP server/responder.
- Online Certificate Status Protocol (OCSP) eliminates the load times of CRLs by abstracting the information that is stored within them into a database.
  An OCSP server accepts requests to verify certificates. OCSP servers and responders are rarely outdated because administrators can revoke a certificate at any time. OCSP can be placed on a separate server from DX NetOps Spectrum.

Once a certificate has been verified as both authentic and valid, the CAC can be accepted.

How CAC Authentication Works

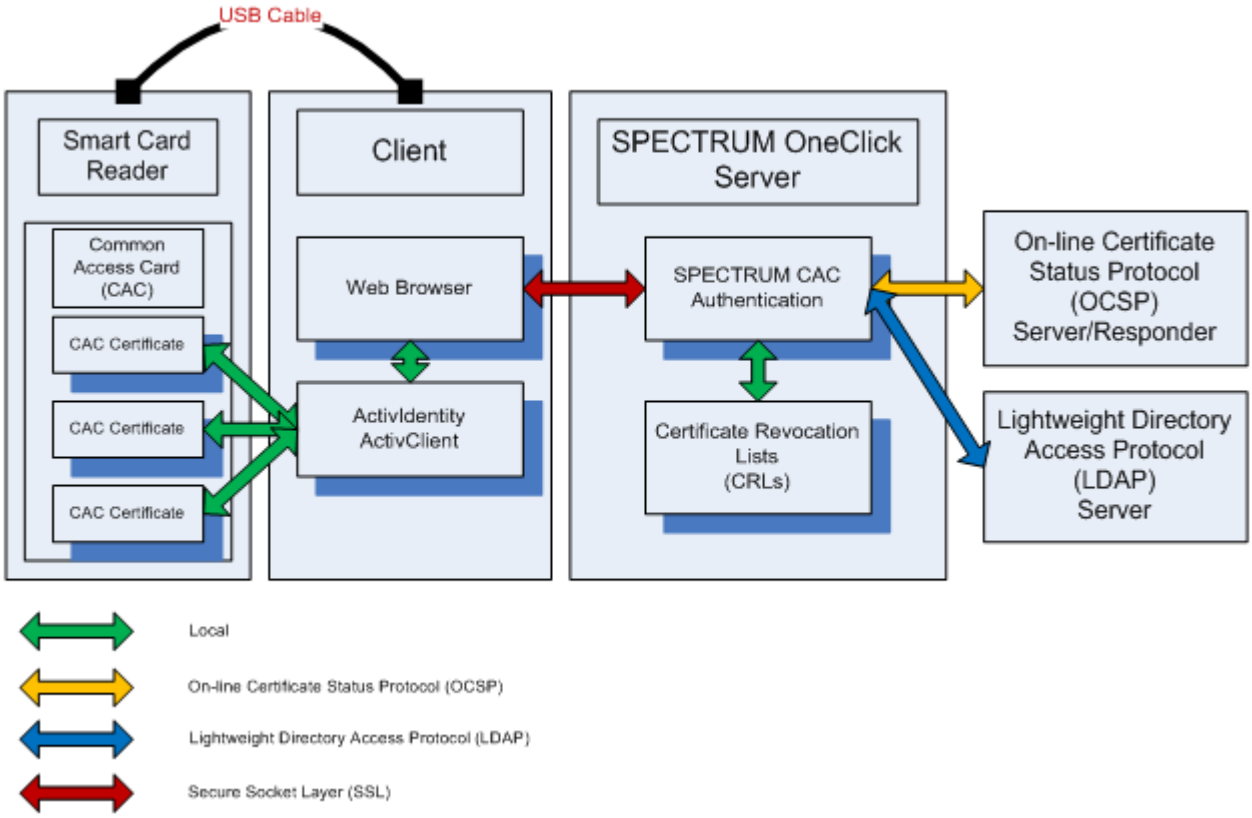
DX NetOps Spectrum CAC Authentication works by leveraging SSL, the Java PKIX Library, and any middleware that exposes PKSC11 to provide a complete CAC authentication solution.

> **NOTE**
> ActivIdentity ActivClient is an example of middleware that exposes PKSC11. We use this client to illustrate the DX NetOps Spectrum CAC authentication solution throughout this section.

As part of SSL negotiation (if SSL is configured), clients that are connecting to the OneClick web server with ActivIdentity ActivClient installed have their CAC Certificates made available for authenticating automatically. Each user provides a CAC PIN to ActivIdentity. Once the PIN is verified, ActiveIdentity releases the CAC Certificate to the OneClick server. DX NetOps Spectrum CAC Authentication verifies that the uploaded CAC Certificate is authentic and valid. Verification consists of building a certificate path and validating the certificate against either Certificate Revocation Lists (CRLs) or an Online Certificate Status Protocol (OCSP) Server.

SSL negotiation is now complete. If LDAP for CAC has been configured, the directory is queried using a unique identifier that is derived from the CAC Certificate. The unique identifier acts as a key to acquire a username to enable login. If LDAP is not used, the user is prompted for a username and password to log in to DX NetOps Spectrum.

The following image shows the architecture for DX NetOps Spectrum CAC Authentication.



Supported Platforms

DX NetOps Spectrum CAC Authentication is available on the OneClick client for the Microsoft Windows and Red Hat Enterprise Linux platforms that DX NetOps Spectrum supports. DX NetOps Spectrum CAC Authentication can be configured on the OneClick server for all DX NetOps Spectrum-supported platforms.

NOTE
DX NetOps Spectrum CAC Authentication for the OneClick client on the Oracle Solaris platform is not supported. If you require CAC Authentication on Solaris, contact CA Support to open a product Enhancement Request.

The following list displays the tested combinations of OneClick client on Windows and the OneClick servers running on Windows and Red Hat Enterprise Linux platforms:

OneClick Clients: See System Requirements for Windows page.	**OneClick Servers:** See System Requirements for Windows page.
Middleware on Windows: HID Global ActivIdentity® ActivClient® 6.2 and 7.0	**Java Runtime Environment (JRE) versions:** JRE 1.6 and 1.7
Web Browser: Microsoft Internet Explorer version 8.0	

NOTE
The table summarizes the platform combinations that we tested. Our testing was constrained by our available resources. Other combinations might also be supported. For more information, contact CA Support.

How to Configure SSL and CAC Authentication

NOTE
The command-line actions that you perform during this process must be executed using the installation owner account. This account owns the DX NetOps Spectrum files.

To configure DX NetOps Spectrum for SSL and CAC Authentication, use the following process:

1. Gather and record the appropriate security certificate information.
2. Verify that ActivIdentity ActivClient™, or other middleware that can expose PKSC11, has been installed on any client that will access OneClick.

   NOTE
   For specific information about installing middleware, see the instructions for the middleware product.
3. Install DX NetOps Spectrum on your server and then reboot.

   NOTE
   For more information about installing DX NetOps Spectrum, see the Fresh Install section.
4. Add DX NetOps Spectrum users.
5. Add Intermediate and Root certificates to DX NetOps Spectrum.
6. Configure the secure socket on the OneClick server.
7. Configure CAC Authentication from the OneClick client.
8. Enable DX NetOps Spectrum CAC Authentication.
9. Verify that CAC is functional. Log in to the CAC Authentication web page and launch the OneClick Console on Windows.
10. (Optional) Configure Linux clients.

Gather Security Certificates and Information

Before you can start setting up DX NetOps Spectrum CAC Authentication, verify that you have the appropriate security certificates and security information readily available to you. Start by gathering the required certificate and security information.

Follow these steps:

1. Import the certificate for the OneClick server as follows:
   - (Optional) Generate a self-signed certificate for the OneClick server if you do not already have one.
   - (Optional) Import an existing self-signed certificate for the OneClick server.
   - (Optional) Import an existing private key and certificate for the OneClick server.
2. Gather root and intermediate certificates for the CACs.
3. Determine the method that you plan to use for CAC verification. Record the information that is indicated for your selection as appropriate:
   - **OCSP AIA**
     Retrieves the parameters of the OCSP server from the certificate on the Common Access Card from the “AIA extension” of the certificate. The OCSP responder certificate is required.
   - **OCSP Server**
     Uses a URL to access the OCSP server and a certificate for the specified server. The OCSP responder certificate and the OCSP responder URL are required.
   - **CRL Directory**
     Uses a path to the directory which contains CRL files. The full path to the directory containing CRLs is required.
   - **CRL URL**
     Specifies a list, separated by spaces, of full URLs to the CRL files that are provided by the webserver. The full URL to each CRL is required.
   - **CRL Distribution Point**
     Specifies that DX NetOps Spectrum retrieves the information about the web location of the CRL files from the certificate itself.

     NOTE
     For more information about these options, see How CACs Work.
4. (Optional) If you are using Lightweight Directory Access Protocol (LDAP), collect the following information:
   - Hostname
   - Port
   - Base distinguished name
   - User distinguished name
   - User password
   - EDIPI attribute name
   - LDAP server certificate (if you plan to enable SSL)
   - Field name to map ID to
   - Field from which to extract the DX NetOps Spectrum Username
   - **For mapping from the certificate to LDAP:**
   - Decide whether ID information on the card certificate (EDIPI or another type of the ID) will come from the subject, alternative name, or rfc822 name.
   - Create a parsing rule to extract ID information from the card.

Generate a Self-Signed Certificate

If you do not already have a certificate, generate a self-signed certificate on the OneClick server.

Follow these steps:

1. Open a command prompt/shell and change the directory to: <*SPECROOT*>/Java/bin.
2. Run the "keytool" program with the following arguments:
   ```
   -genkey -alias tomcatssl -keyalg RSA -keystore <SPECROOT>/custom/keystore/cacerts
   ```
3. Enter **changeit** for the -keystore password.

NOTE
The word 'changeit' is the default password for the keystore.

4. Complete the fields. The following fields are not self-explanatory:
   - **First+Last name**
     Specifies the fully qualified domain name of your OneClick server. For example, "myhostname.mydomain".
   - **Organizational Unit**
     Specifies your company division. For example, Spectrum Engineering.
   - **Organization**
     Specifies the company name. For example, CA Inc.
5. Verify that your information is correct, and type 'yes' to accept.
6. Press Enter to use the same password as the keystore.

Import an Existing Self-Signed Certificate for the OneClick Server

If you already have a certificate, you must import it for DX NetOps Spectrum CAC Authentication.

To import the existing certificate for DX NetOps Spectrum CAC Authentication

1. Change the directory to: <SPECROOT>/Java/bin.
2. Run the following command:
   ```
   ./keytool -importcert -alias tomcatssl -file cert_file -keystore <SPECROOT>/custom/keystore/cacerts
   ```
   - ***cert_file***
     Specifies the existing OneClick certificate file.
3. Type **changeit** for the keystore password.
4. Press Enter to use the same password as the keystore.

Import an Existing Private Key and Certificate for the OneClick Server

WARNING
This procedure destroys your existing cacerts keystore and creates a new one with your private key and certificate. At present, you cannot force a private key into an existing keystore. This procedure is the only way to create a new keystore with a preexisting private key.

Follow these steps:

1. Gather the private key and the certificate files.
2. Change the directory to a temporary directory.
3. Execute the following command:
   ```
   openssl pkcs12 -export -inkey <private_key_file> -in <server_cert_file> -out mycert.pfx -name "default"
   ```
4. Change to the following directory:
   <SPECROOT>/Java/bin
5. Execute the following command:
   ```
   keytool -importkeystore -srckeystore <path_to_mycert.pfx> -srcstoretype pkcs12
   -destkeystore <SPECROOT>/custom/keystore/cacerts -srcalias default -destalias
   tomcatssl -destkeypass changeit
   ```
   Your private key and server certificate are now stored in the keystore, which is located in the following directory: <*SPECROOT*>/custom/keystore/cacerts

Add Users

You can add DX NetOps Spectrum users from the DX NetOps Spectrum Control Panel.

NOTE
For more information, see the OneClick Administration section.

Follow these steps:

1. Open the DX NetOps Spectrum Control Panel using *one* of the following methods depending on the platform you are using:
   - Linux: /usr/SPECTRUM/bin/SCP
   - Windows: Start, Program Files, CA, DX NetOps Spectrum Control Panel.
2. Click Start SpectroSERVER.
   The SpectroSERVER starts.
3. Select Control, Users from the main menu.
   The Users dialog opens.
4. Click Create to create the desired user.
   The Create dialog opens.
5. Type the user's name in the User Name field.

   NOTE
   If you are using LDAP, user names in DX NetOps Spectrum must exactly match those in LDAP.
6. Type a password in the New Password and Confirm New Password fields.

   NOTE
   This password is only used if LDAP is not enabled.
7. Click OK.
   The Create dialog closes.
8. Click Close.
   The DX NetOps Spectrum user is created.

Add Intermediate and Root Certificates

Use the SSL Certificates administration page to load root and intermediate certificate authority certificates for the CACs.

NOTE
For more information, see the OneClick Administration section.

Follow these steps:

1. Click Administration in the OneClick home page.
   The Administration Pages open.
2. Click SSL Certificates in the left side panel.
   The SSL Certificates page opens.
3. Load the root or intermediate certificate authority Certificates for the CACs in the 'File with Certificate' field.
4. Enter an appropriate alias name of your choice.
5. Click Save.

   NOTE
   Restarting the OneClick server is not required after you load each separate certificate. You can wait until you have loaded all of the desired certificates.
6. Repeat Steps 3-5 for every certificate you want to load.
7. (Optional) Load the Online Certificate Status Protocol (OCSP) Responder Certificate if you are using (OCSP).

   NOTE
   Record the name of the certificate alias that is associated with this certificate. The alias name is a requirement for a later step.
8. (Optional) Load the LDAP certificate if you are using SSL to connect to the LDAP server.

9. Click Restart OneClick Server after you have loaded all of the appropriate certificates.

How to perform key functions:

1. Go to $SPECTRUM/Java/bin and run ./keytool.exe -help Key and Certificate Management Tool and run the following commands to perform any specific function:

```
-certreq Generates a certificate request
-changealias Changes an entry's alias
-delete Deletes an entry
-exportcert Exports certificate
-genkeypair Generates a key pair
-genseckey Generates a secret key
-gencert Generates certificate from a certificate request
-importcert Imports a certificate or a certificate chain
-importpass Imports a password
-importkeystore Imports one or all entries from another keystore
-keypasswd Changes the key password of an entry -list Lists entries in a keystore
-printcert Prints the content of a certificate
-printcertreq Prints the content of a certificate request
-printcrl Prints the content of a CRL file
-storepasswd Changes the store password of a keystore
```

NOTE

Refer to Configure the Secure Socket on the OneClick Server page on how to use the keytool!

Configure the Secure Socket on the OneClick Server

As a final step in configuring the OneClick web server for SSL, configure the secure socket on the OneClick web server host.

Follow these steps:

1. Shut down the OneClick web server:
   - **Linux**
     As root: <*$SPECROOT*>/tomcat/bin/stopTomcat.sh
   - **Windows**
     Enter the following command from a command prompt:
     ```
     C:\> net stop spectrumtomcat
     ```
2. Open <*$SPECROOT*>/tomcat/conf/server.xml in a text editor.
3. Locate the following section in the server.xml file:

```
<!-- Define a SSL Coyote HTTP/1.1 Connector on port 443 -->
    <!--
    <Connector
           port="443" minProcessors="5" maxProcessors="75"
           enableLookups="true" disableUploadTimeout="true"
           acceptCount="100" debug="0" scheme="https" secure="true"
           clientAuth="false" sslProtocol="TLS"
           keystoreFile="<SPECROOT>/custom/keystore/cacerts"
           keystorePass="changeit">
     </Connector>
```

```
-->
```

By default the <Connector> element in the section is commented out. Uncomment this section and change clientAuth="false" to clientAuth="true".

NOTE
The preceding XML fragment is Windows-specific. This example specifies 443 as the default port where the OneClick web server listens for SSL communications. You can omit the port from the URL for accessing the OneClick home page:

```
https://<fully_qualified_host_name>/spectrum
```

On a UNIX-based installation, the OneClick web server is not run as root, and the default port is 8443 (it must be greater than 1024).

4. Specify the port number in the web browser when you enter the URL to access the OneClick home page:
   ```
   https://<fully_qualified_host_name>:8443/spectrum
   ```
5. Remove the comments around the Connector definition. Make the definition active by deleting the “<!--” and “-->” tags that surround this section.
6. Replace <*$SPECROOT*> with the actual path as follows:
   - **Linux**
     /usr/SPECTRUM
   - **Windows**
     C:/win32app/SPECTRUM
7. Change clientAuth to “true”.

   NOTE
   Changing this setting to "true" is a key component of the DX NetOps Spectrum Common Access Card solution. You can configure DX NetOps Spectrum for SSL without ClientAuth. However, this parameter must be set to "true" to enable DX NetOps Spectrum Common Access Card authentication. For more information, see OneClick Administration section.
8. Save and close the server.xml file.
9. Start the OneClick web server using one of the following commands, depending on the platform you are using:

   - **Linux**
     As root:
     <*$SPECROOT*>/tomcat/bin/startTomcat.sh
   - **Windows**
     From a command prompt:
     ```
     C:\> net start spectrumtomcat
     ```

The secure socket is now configured.

Configure CAC Authentication on the OneClick Client

After configuring DX NetOps Spectrum for SSL, configure CAC Authentication on the OneClick client. You can set up CAC Authentication by entering all of the relevant security information you have gathered.

Follow these steps:

1. Log in to the OneClick client that has ActivIdentity ActivClient installed.
2. Open the OneClick Administration pages using the SSL port as follows:
   - For Linux, type **https://your_hostname:8443/**
   - For Windows, type **https://your_hostname/**
3. Provide your CAC Certificate.

NOTE
The CAC certificate that you provide here is used to verify that the CAC Configuration is valid.

4. **(Optional)** Accept the OneClick Server Certificate if prompted.
5. Enter the installation owner user name and password when prompted.

   NOTE
   If you have not changed the password for the installation owner, it is *spectrum*.

   The OneClick home page opens.
6. Click Administration.
   The Administration Pages open.
7. Click CAC Configuration in the left side panel.
   The CAC Configuration page opens.

   NOTE
   If you do not see a full web page here with options available, you did not use the SSL port. Repeat Steps 1 through 6.

Setting Up the eHealth Integration with CAC Enabled

You can set up the CAC Integration with DX NetOps Spectrum using the following steps:

- Disable **TLS 1.2** in Java Console (this is to be able to launch JNLP applications) on **Windows client** system, where the card-reader is attached.
- Stop Spectrum Tomcat and eHealth httpd and SCARVES
  - **SPEC:** $SPECROOT/tomcat/bin/stopTomcat.sh
  - **EH:** $NH_HOME/bin/nhHttpd stop
  - **EH (AS ROOT):** $NH_HOME/bin/nhSmartCard.sh stop
- **SPEC:** cd $SPECROOT/CERTS/certs
- **SPEC:** sftp nhuser@<EH Machine Name>
  - cd **<path_to_EH_home>/**web/httpd/conf/cacerts
  - put SpectrumRootCA.crt
  - bye
- **EH:** Import the Spectrum certificates into the eHealth trust stores
  - cd $NH_HOME/bin
  - ./nhSmartCard.sh trust -import -storepass 123456 -alias SpectrumRootCA -file $NH_HOME/web/httpd/conf/cacerts/SpectrumRootCA.crt
  - ./nhWebProtocol -mode https -hostname <EH FDQN> -strongCipher -disableSSLv2 -enableSCAuth -SCCAdir $NH_HOME/web/httpd/conf/cacerts -SCCAsvrFile $NH_HOME/web/httpd/conf/myDaemonCert3.pem -SCServerIPs "<EH FDQN>:8888" -certificate $NH_HOME/CERTS/<HOSTNAME>.crt -key $NH_HOME/CERTS/<HOSTNAME>.key -intermediate $NH_HOME/web/httpd/conf/myDaemonCert3.pem –fips
- – $NH_HOME/bin/nhHttpd start
  - **AS ROOT:** $NH_HOME/bin/nhSmartCard.sh start
  - Verify you can still log into eHealth with CAC.
- **SPEC:** Import the eHealth certificates into the Spectrum OneClick trust store
  - Stop tomcat
    - $SPECROOT/tomcat/bin/stopTomcat.sh
  - cd $SPECROOT/CERTS
  - sftp nhuser@<EH HOSTNAME>

  - cd $NH_HOME/web/httpd/conf
  - get myDaemonCert3.pem eHealth.pem
  - bye
- $SPECROOT/Java/bin/keytool -import -alias EH_CAC -file eHealth.pem -keystore $SPECROOT/custom/keystore/cacerts -storepass changeit
- Edit $SPECROOT/lib/SDPM/partslist/**TOMCAT.idb** and add the following command like parameters to tomcat (applies for LINUX Spectrum 9.4.0 and earlier versions)
- su
- chmod –R a+rwx $SPECROOT/lib/SDPM
- Edit $SPECROOT/tomcat/bin/**catalina.sh** and add the following command like parameters to tomcat JAVA_OPT variable (applies for LINUX Spectrum 9.4.0 and earlier versions):
  - -Djavax.net.ssl.keyStore=$SPECROOT/custom/keystore/cacerts
  - -Djavax.net.ssl.keyStorePassword=changeit

  Example for Spectrum 9.4.2 and later : **catalina.sh** change

```
# This needs to be done after setclasspath.sh as it sets the JAVA_OPTS as well
if [ "$HOST" = "SunOS" ]; then
JAVA_OPTS="-DOneClick -Xmx1024M -XX:PermSize=128M -XX:MaxPermSize=128M -XX:+HeapDumpOnOu
tOfMemoryError -server -Djava.awt.headless=true -Djavax.net.ssl.trustStore=$SPECROOT/cus
tom/keystore/cacerts -Dfile.encoding=UTF-8 -Dcom.sun.management.jmxremote -Dorg.apache.c
oyote.USE_CUSTOM_STATUS_MSG_IN_HEADER=true -Djavax.net.ssl.keyStore=$SPECROOT/custom/key
store/cacerts -Djavax.net.ssl.keyStorePassword=changeit"
else
JAVA_OPTS="-DOneClick -Djava.compiler=NONE -Xmx1024M -XX:PermSize=128M -XX:MaxPermSize=1
28M -XX:+HeapDumpOnOutOfMemoryError -server -Djava.awt.headless=true -Djavax.net.ssl.tru
stStore=$SPECROOT/custom/keystore/cacerts -Dfile.encoding=UTF-8 -Dcom.sun.management.jmx
remote -Dorg.apache.coyote.USE_CUSTOM_STATUS_MSG_IN_HEADER=true -Djavax.net.ssl.keyStore
=$SPECROOT/custom/keystore/cacerts -Djavax.net.ssl.keyStorePassword=changeit"
fi
```

- Edit $SPECROOT/tomcat/bin/**OneClickService.conf** and add the following command like parameters to the end of file (applies for LINUX Spectrum 9.4.0 and earlier versions):
  - jvm_opt=-Djavax.net.ssl.keyStore=C:/win32app/Spectrum/custom/keystore/cacerts
  - jvm_opt=-Djavax.net.ssl.keyStorePassword=changeit
- Start tomcat
  - $SPECROOT/tomcat/bin/startTomcat.sh
- Attempt to configure the eHealth integration from the OneClick Administration page.

CAC Configuration Page

Use the CAC Configuration page to configure OneClick to use Common Access Cards (CAC) for authentication.

- **Choose CAC Option**
  Specifies whether to enable or disable the CAC authentication solution.
  - **Disable CAC**
    Disables CAC authentication.
  - **Enable CAC**
    Enables CAC authentication.

The Trusted Keystore section contains the following fields:

- **Trusted Keystore password**
  Specifies the password to use for accessing the Trusted Keystore: **changeit**.
- **Re-enter Trusted Keystore password**
  Confirms the password for accessing the Trusted Keystore.

The Revocation System section specifies how you want DX NetOps Spectrum to determine whether a CAC has been revoked. Select *one* of the following options:

- **Enable OCSP AIA**
  Instructs DX NetOps Spectrum to retrieve the parameters of the OCSP server from the certificate on the Common Access Card from the “AIA extension” of the certificate.
- **Enable OCSP Server**
  Specifies that the user must provide a URL to access OCSP server and a certificate for this server.
- **Enable CRL Directory**
  Specifies that a path to the directory that contains CRL files is required.
- **Enable CRL URL**
  Specifies a list, separated by spaces, of full URLs to the CRL files that the web server provides.
- **Enable CRL Distribution Point**

Specifies that DX NetOps Spectrum retrieves the information about the web location of the CRL files from the certificate itself.

The OCSP AIA Connectivity section appears when you select Enable OCSP AIA in the Revocation System section. This section contains the following option:

- **Test OCSP AIA**
  Verifies that OCSP AIA is working properly.

The OCSP Server Connectivity section appears when you select Enable OCSP Server in the Revocation System section. This section contains the following options:

- **OCSP Server URL**
  Specifies the complete URL for accessing the OCSP Responder. The complete URL is used because many OCSP Responders are servlets running on a larger OCSP server.
- **OCSP Server Certificate Alias**
  Specifies the certificate for the specified OCSP server.
- **Test OSCAP Server**
  Tests the connection to the OCSP server based on the credentials that you entered.

The Certificate Relocation Lists appears when you select Enable CRL Directory or Enable CRL URL in the Revocation System section. It contains the following settings, depending on the CRL option that you selected:

- **CRL Directory**
  Specifies the full path to the directory that contains the CRL files for verifying user certificates.
- **CRL URL**
  Specifies a list of full URLs, separated by spaces, to the CRL files that the web server provides.
- **Test CRL Availability**
  Attempts to load the CRLs in the specified directory.

The LDAP Username Lookup section contains the following settings:

- **Enable LDAP**
  Enables LDAP.
- **LDAP Server Hostname**
  Specifies the host name of an LDAP server that contains users that correspond to user certificates.
- **LDAP Server Port**
  Specifies the port number for accessing the LDAP server.
- **Enable SSL**
  Enables secure connecting to the LDAP server using SSL.

  NOTE
  Load the LDAP server certificate if you enable SSL.
- **LDAP Base DN**
  Specifies the LDAP base distinguished name.
- **LDAP User DN**
  Specifies the distinguished name (DN) of the user that is used to query the LDAP server.
- **LDAP User Password**
  Specifies the password of the user that is used to query the LDAP server.
- **Re-enter LDAP User Password**
  Confirms the LDAP user password.
- **Certificate's EDIPI Field**
  Specifies the source for user ID information. Select *one* of the following options, which describe the format in which EDIPI is stored in the CAC certificate:

- Subject
- SubjectUniqueId
- AltName.otherName
- AltName.rfc882Name

- **EDIPI Extraction Rule**
  Specifies the rule to use to extract EDIPI from the CAC certificate field.
  Type: Java regular expression
  Example: The default value for this field is as follows:

```
"CN=\w*\.\w*\.(\d+),";
```

  This string defines a rule that matches a string that resembles the following example:

```
CN=aaaa.bbbbbb.1233454,xxxxxxxxxxxxxxxx
```

  - Literal “CN=”
  - Any word (possibly empty) \w*
  - Literal “.”
  - Any word (possibly empty) \w*
  - Literal “.”
  - Integer number (non-empty) \d+
  - Literal “,”
  - Anything can follow.

    NOTE
    Regex *capturing group* must be defined in the regular expression. DX NetOps Spectrum uses the first defined group in the expression to extract unique user ID information. More information about capturing groups is available on the Internet.

- **LDAP EDIPI Attribute Name**
  Specifies the name of the LDAP field that is used to store EDIPI (or other unique identifier) information.
- **LDAP Username Attribute Name**
  Specifies the name of the LDAP field that is used to store DX NetOps Spectrum user name information.
- **LDAP Referral Setting**
  Specifies how OneClick handles LDAP referrals.
  - **follow**
    (Default) Instructs OneClick to automatically follow any referrals.
  - **throw**
    Instructs OneClick to throw an exception for each referral. The request is likely to fail with an "Unprocessed Continuation Reference(s)" error.
  - **ignore**
    Instructs OneClick to ignore referrals. The request is likely to fail with an "Unprocessed Continuation Reference(s)" error.

    NOTE
    LDAP Referral Setting is hidden by default on the CAC Configuration page. To display this field and change its value, see Modify LDAP Referral Setting.
- **Test LDAP Server**
  Attempts to connect to the LDAP server using the credentials that you supplied.

Modify LDAP Referral Setting

The LDAP Referral Setting specifies how OneClick handles LDAP referrals. When an LDAP server cannot locate a requested object, the server returns a referral to the client. The referral directs the request to another server to locate the object. By default, OneClick automatically follows referrals to obtain the requested information.

You can also specify to ignore referrals or to throw an exception for each referral. The LDAP Referral Setting is hidden by default on the CAC Configuration page. To modify its value, change the configuration file to display the field.

Follow these steps:

1. Open <*$SPECROOT*>/tomcat/webapps/spectrum/WEB-INF/cac/cac-config.jsp in a text editor.
2. Uncomment the LDAP Referral Setting display code:
   a. Add "-->" after <!-- BEGIN HIDDEN REFERRAL SETTING SECTION
   b. Add "<!--" before END HIDDEN REFERRAL SETTING SECTION -->
3. Save and close the file.
4. Refresh the CAC Configuration page.
   The LDAP Referral Setting field appears.
5. Select a value from the LDAP Referral Setting drop-down list, and click Save.
   The new LDAP Referral Setting takes effect.

Enable CAC Authentication

After you have configured DX NetOps Spectrum CAC Authentication on the OneClick server and have finished setting up security, you can enable it from the CAC Authentication page.

Follow these steps:

1. Click Administration in the OneClick home page.
   The Administration Pages open.
2. Click CAC Configuration in the left panel.
   The CAC Configuration page opens.
3. Click Enable CAC.
   The available configuration options for enabling CAC are displayed.
4. Enter the Keystore password in the Keystore password fields.

   NOTE
   If you have not changed the Keystore password, it is *changeit*.
5. Select one of the following options in the Revocation System section. These options specify how DX NetOps Spectrum determines whether a CAC has been revoked. Complete the resulting fields:
   - **Enable OCSP AIA**
     Instructs DX NetOps Spectrum to retrieve the parameters of the OCSP server from the certificate on the Common Access Card from the “AIA extension” of the certificate.
   - **Enable OCSP Server**
     Specifies that the user must provide a URL to access the OCSP server and a certificate for this server.
   - **Enable CRL Directory**
     Specifies that a path to the directory that contains CRL files must be specified.
   - **Enable CRL URL**
     Specifies a list, separated by spaces, of full URLs to the CRL files that are provided by the webserver.
   - **Enable CRL Distribution Point**
     Specifies that DX NetOps Spectrum retrieves the information about the location of the CRL files from the certificate itself.

   The CAC Configuration page changes to display the fields that relate to the option that you selected. For more information, see CAC Configuration Page.

6. (Optional) If you are using LDAP, select the Enable LDAP checkbox, and complete the fields as described in CAC Configuration Page.
7. Click the individual test buttons to test your information.
8. Click Save to save your selections.
   A full test of your CAC configuration options runs. If the test is successful, the CAC information is saved, and the OneClick server restarts. If you are using CRLs, they are loaded immediately after the restart. Depending on the number of CRLs and their size, this process can take several minutes. During this time, attempts to access the server using a web browser do not always provide feedback.
9. (Optional) Track the progress of the load operation by viewing one of the following logs:
   – *$SPECROOT*/tomcat/logs/catalina.out for Linux
   – *$SPECROOT*/tomcat/logs/stdout.log for Windows

Configure Linux Clients

If any users are accessing DX NetOps Spectrum from Linux clients, configure those clients for DX NetOps Spectrum CAC Authentication.

Follow these steps:

1. Run mkoctar script on the OneClick server by doing the following:
   1. Set the environment variable SPECROOT to the SPECTRUM root directory.
   2. Navigate to <*$SPECROOT*>/tomcat/webapps/spectrum/
   3. Run the following command:
      ./mkoctar - servercert <*oneclick_certificate_alias*> -cert <*root_alias*> -cert <*int_alias_1*> -cert <*int_alias_2*>
   4. **- servercert <oneclick_certificate_alias>**
      a. Specifies the alias for the OneClick web server certificates. If you created a self-signed certificate, the OneClick certificate alias is "tomcatssl".
   5. **-cert <root_alias>**
      a. Specifies the alias for the root certificate, as defined in Add Intermediate and Root Certificates to DX NetOps Spectrum.
   6. **-cert <int_alias_1>**
      a. Specifies the alias for the first intermediate certificate, as defined in Add Intermediate and Root Certificates to DX NetOps Spectrum.
   7. **-cert <int_alias_2>**
      a. Specifies the alias for the second intermediate certificate, as defined in Add Intermediate and Root Certificates to DX NetOps Spectrum.
   8. (Optional) Run the following command if you see a Permission Denied error:
      chmod +x mkoctar
   9. (Optional) Run the following command to view additional options:
      ./mkoctar -h

   This command produces the file "oc.tar" in the same directory. You can now copy oc.tar to a temporary directory on Linux clients that will access OneClick.
2. On the Linux client, extract oc.tar as follows:
   a. **Note:** For performance and security reasons, extract this file to a local disk, not to a network drive.
   b. Run the following command:
      tar xvf oc.tar
   c. Edit the line in card.config.Linux that begins with "library=". Change it to point to the ActivIdentity pkcs library. For example, change it to the following line:
      ```
      library = /usr/local/ActivIdentity/ActivClient/lib/libacpkcs211.so
      ```

d. Run the runoc script to launch OneClick. Take *one* of the following steps:

- LDAP: Run the following command:
  ./runoc
- Non-LDAP: Run the following command:
  ./runoc - noldap

When it is first run, runoc installs a JRE in the current directory.

NOTE
Users are always prompted for their CAC Personal Identification Number (PIN). If you are not using LDAP, users are also prompted for their user name and password.

Working with CAC Authentication

Once DX NetOps Spectrum CAC Authentication is enabled, the only visible change to the user interface is the new CAC Configuration page in the OneClick Administration pages.

Non-SSL access is completely disabled after DX NetOps Spectrum CAC Authentication has been enabled. Any attempts to access the server from a non-secured port will cause a redirection to the SSL port.

Memory Consumption Using CAC Authentication with CRLs

A single CRL can hold more than half a million revoked certificate serial numbers. Some environments require 20 or more CRLs to cover all potential user certificates. Loading this much data consumes a great deal of time and memory. For example, a set of CRLs that is approximately 100 MB on disk consumes about 1.5 GB of memory and takes several minutes to process.

Take this increased memory consumption into account when you decide where to install OneClick. An easy way to calculate the memory consumption is to take the total size of your CRLs and multiply by 15. One MB of CRLs consumes 15 MB of memory, and 50 MB consumes 750 MB of memory. If your total exceeds 400 MB, consider increasing the amount of memory that is available to OneClick. If your total exceeds 500 MB, OneClick typically requires increased memory to function.

WARNING
If your total CRL memory requirement exceeds 1 GB, do not run DX NetOps Spectrum CAC Authentication with CRLs on a Windows server.

The steps to take to increase the amount of memory that is available to the OneClick server are described in Configure OneClick Web Server Memory Settings.

To alleviate the time impact on end-users, CRLs are loaded while OneClick is initializing. Users of OneClick do not experience significant delays once OneClick is running. You can track the progress of the CRL load operation by viewing the OneClick log file at one of the following locations:

- **Linux**
  `<SPECROOT>/tomcat/logs/catalina.out`
- **Windows**
  `<SPECROOT>\tomcat\logs\stdout.log`

Configure OneClick Web Server Memory Settings

By default, the OneClick web server needs 4 GB dedicated to OneClick Web Server. If the OneClick web server is using more than 75 percent of its configured maximum memory, consider increasing the maximum memory value.

NOTE
The actual memory requirement depends on your OneClick server environment and usage, like configuration and number of managed devices.

- Default: 4 GB
- For production environments: minimum of 8 GB
- For OneClick integrations: minimum of 16 GB
  (for example, integration with CAPM, UIM, ServiceDesk, VNA and so on)

If the webserver runs out of memory, an OutOfMemory error appears in the following log files:

- tomcat/logs/stdout.log (for Windows)
- tomcat/logs/catalina.out (for Linux).

You can change the memory allocations on the Web Server Memory Administration page. We recommend allocating 50% to 75% of the available OS memory to the tomcat process. For example, if you have 32GB then, set tomcat memory to 16GB or 24GB. The steps in this procedure are an optional method for addressing out-of-memory issues.

NOTE
Restart the OneClick web server for these changes to take effect.

Follow these steps:

1. Verify the OneClick web server memory usage:
   a. Click Administration in the OneClick home page.
      The Administration Pages open.
   b. Click Web Server Memory in the left panel.
      The Web Server Memory page opens.
   c. Check the OneClick Server Memory Usage field to verify if memory usage is greater than 75 percent of the configured maximum.
2. Configure the maximum OneClick web server memory usage:
   a. In the Maximum Memory the Server Can Use (In MB) field, enter the new value.
      NOTE
      Do not set the maximum memory to a value larger than the available memory for the system.
   b. Click Save.
      A dialog prompts you to commit your changes and restart the OneClick web server.
   c. Click OK.
      Your changes are saved, and the OneClick web server is restarted.

Changes to Access Security

If any part of the CAC certificate verification process fails, the user cannot be granted access. Allowing a user to log in without a validated CAC would defeat the purpose of enabling CAC Authentication.

If LDAP is up, but a user name is not returned properly or the returned user name is not in DX NetOps Spectrum, the user cannot be granted access. This is because user access is still controlled by DX NetOps Spectrum, and allowing someone who has a valid CAC but an invalid user name to log in as anyone is not secure.

Troubleshooting CAC Authentication

This section includes troubleshooting tips to solve the specified problems while you use CAC authentication.

Java Heap Size and OutOfMemory Errors

Symptom:

I am seeing errors such as "Java Heap Size" and "OutofMemory" when I use DX NetOps Spectrum CAC Authentication with CRLs.

Solution:

The memory of your OneClick server is set too low. Complete the following procedure to avoid seeing these errors.

1. Remove a few CRLs from your CRL directory.
2. Restart the OneClick server.
3. Increase the amount of memory available on the server as described in Configure OneClick Web Server Memory Settings.
4. Try using DX NetOps Spectrum CAC Authentication again.

> **NOTE**
> If you continue to get this error, you may not have enough memory to run DX NetOps Spectrum CAC Authentication with CRLs.

OneClick Client Locks Up After Authentication

Platform: Linux only

Symptom:

I am using CAC authentication for my OneClick clients, which I launch on a Linux server. After I invoke the runoc script, the client fails soon after startup. The OneClick client locks up and becomes unusable. Sometimes, I see a certificate error. I am forced to exit the application. After a few attempts, the client starts and runs as expected.

Solution:

This situation occurs when the SSL connection between the OneClick client and the Tomcat server times out before the SSL handshake has completed.

To resolve this issue, increase the timeout period that is set on the client and on the Tomcat server. On Linux, clients use a script (runoc) to launch OneClick. You can modify the -ssltimeout parameter in this script so that SSL has more time to complete the handshake. The longer timeout lets the handshake complete while the connection is still established.

First, reconfigure the server to use a five-minute connection timeout.

Follow these steps:

1. Navigate to *$TOMCAT_ROOT*/conf/
2. Open the file server.xml for editing, using your preferred text editor.
3. Locate the SSL Connector element in the file.
4. Change the connectionTimeout parameter to use a value equivalent to five minutes in milliseconds:

```
connectionTimeout="300000"
```

> **NOTE**
> The server setting is in milliseconds.

5. Restart the OneClick server so that the change takes effect.

Now launch the OneClick client using the new -ssltimeout parameter.

Follow these steps:

1. Navigate to the directory from which you launch the runoc script on the OneClick client. For example, /opt/CA_OC.
2. Invoke the runoc script, using the -ssl timeout parameter. For example, enter the following command:

```
./runoc  - ssltimeout <value>
```

In this example, use a value of 300 (the equivalent of five minutes in seconds).

NOTE
The client setting is in seconds.

The OneClick client now has a five-minute timeout setting, which matches the timeout that you set on the tomcat server.

Poor OneClick Client Performance

Platform: Windows

Symptom:

The OneClick clients take a long time to start up. Once the clients have started, users experience long wait times for the user interface to react to mouse clicks and navigation tasks. They are so slow that we can hardly use them.

Solution:

This behavior results from the default setting for the "javaws.reuseConnections" Java System property, which is "false". In previous versions of DX NetOps Spectrum, the default value was "true". A change was made to facilitate out-of-the-box connectivity for users with web proxies or load balancers in their environment. Without reusing connections, SSL certificate verification is performed for every request from the client to the server. This work is expensive, in terms of round-trip times.

Change the value of the "javaws.reuseConnections" Java Runtime System property to "true".

To change the property setting, edit the oneclick.jnlp file.

Follow these steps:

1. Navigate to the following directory:
   ```
   <$SPECROOT>/tomcat/webapps/spectrum/
   ```
2. Open the oneclick.jnlp file for editing using your preferred text editor.
3. Add the following line, immediately below the "<resources>" line:
   ```
   <property name="javaws.reuseConnections" value="true"/>
   ```
4. Restart all open OneClick clients.

Poor OneClick Client Performance

Platform: Linux

Symptom:

The OneClick clients take a long time to start up. Once the clients have started, users experience long wait times for the user interface to react to mouse clicks and navigation tasks. They are so slow that we can hardly use them.

Solution:

This behavior results from the default setting for the "javaws.reuseConnections" Java Runtime System property, which is "false". In previous versions of DX NetOps Spectrum, the default value was "true". A change was made to facilitate out-of-the-box connectivity for users with web proxies or load balancers in their environment. Without reusing connections, SSL certificate verification is performed for every request from client to server. This work is expensive, in terms of round-trip times.

Change the value of the "javaws.reuseConnections" Java runtime property from "false" to "true".

To change the property setting for Linux clients, edit the runoc script.

Follow these steps:

1. Navigate to the directory where you installed the runoc script.
2. Add "-Djavaws.reuseConnections=true" to the last line of the script as follows:
   ```
   $JRE_HOME/bin/java -Djavaws.reuseConnections=true -classpath
   ```
3. Save the script.
4. Restart all open OneClick clients.

OneClick console does not launch Start Console

Platform: Windows

Symptom:

After enabling CAC, the ‘Start Console’ is not lauched from the OneClick console.

Solution:

This behavior results from the setting of ‘Use TLS 1.2’ in Java Control Panel.

Before opening the OneClick console, you must disable the ‘Use TLS 1.2’ option from ‘Advanced Security Settings’ in Java Control Panel.

Follow these steps:

1. Open the command prompt.
2. Type the following command to launch the Java Control Panel:
   ```
   javaws -viewer
   ```
3. Click the ‘Advanced’ tab.
4. Navigate to ‘Advanced Security Settings’ section and disable the ‘Use TLS 1.2’ option.
5. Click ‘Ok’ to save the changes.

Unable to Build Certificate Path

Symptom:

I am seeing the error "Unable to Build Certificate Path".

Solution:

This error occurs when SSL is set up, but the option for Client Authentication is not set to “true”. In this case when someone attempts to access a OneClick server with DX NetOps Spectrum CAC Authentication, they do not present a CAC certificate ahead of time. As a result, DX NetOps Spectrum CAC Authentication attempts to build a certificate path, but no certificates are available.

Refer to Configure the Secure Socket on the OneClick Server. Verify that you have set clientAuth to “true”, and restart the OneClick server.

Unable to Determine Certificate Status

Symptom:

I am seeing the error "Unable to Determine Certificate Status".

Solution:

This error is usually presented when CRLs have expired. Acquire new CRLs from the Certificate Authority and replace the old CRLs. Then restart the OneClick server.

Certificate Has Been Revoked

Symptom:

I am seeing the error "Certificate Has Been Revoked".

Solution:

This error indicates that the CAC Certificate has been revoked. The revocation occurred either by an entry in a CRL or because the OCSP Responder has indicated that the certificate is no longer valid.

Microsoft MOM and SCOM

This section describes how to integrate Microsoft® Operations Manager (MOM) and Microsoft® System Center Operations Manager (SCOM) with DX NetOps Spectrum.

- **MOM Connector**: A DX NetOps Spectrum executable that enables alert forwarding from Operations Manager to DX NetOps Spectrum.
- **SCOM Connector:** A DX NetOps Spectrum executable that enables alert forwarding from System Center Operations Manager to DX NetOps Spectrum.

> **NOTE**
>
> The DX NetOps Spectrum MOM and SCOM connectors are compatible with Microsoft® Operations Manager 2005 and Microsoft® System Center Operations Manager (SCOM) 2012, respectively.

MOM/SCOM Connector Comparison

Both the MOM Connector and the SCOM Connector support the following functionality:

- Extracting alerts from Microsoft Operations Manager or System Center Operations Manager and creating alarms on the appropriate system models in DX NetOps Spectrum.
- Supporting drill-down from DX NetOps Spectrum alarms. You can configure DX NetOps Spectrum alarms to display a URL that calls up the appropriate alert in the MOM or SCOM web console.
- Bidirectional alert/alarm clearing.
- Bidirectional alert/alarm acknowledgment.

The MOM Connector and the SCOM Connector have the following differences:

- Alert/alarm acknowledgments: The connectors handle acknowledgments differently.

System Center Operations Manager does not support a resolution state for acknowledging an alert. Therefore, you must create one manually. You can then configure the SCOM Connector to use the new resolution state to synchronize acknowledgments of alerts and alarms. System Center Operations Manager does not provide acknowledgment synchronization by default.

For more information, see Create an Acknowledged Resolution State.

By default, Microsoft Operations Manager 2005 provides an acknowledgment resolution state and acknowledgment synchronization.

- Deployment options: The SCOM Connector can be deployed to almost any Windows host in your environment. But the MOM Connector must run on the MOM server.
- The MOM Connector is not fully supported in a DSS environment. For more information, see MOM Connector in a DSS Environment.

Some versions of the SCOM connector are not fully supported in a DSS environment. The extra configuration is required to support distributed deployments. For more information, see SCOM Connector in a DSS Environment.

- Distinct configuration files: The MOM configuration file is named ".momrc". The SCOM configuration file is named ".scomrc." For more information, see Install the MOM Connector or Install the SCOM Connector.

Install and Run the MOM Connector

This section assumes the following:

- You have an installed and configured Microsoft Operations Manager 2005 Management Server.
- You are integrating your MOM environment with DX NetOps Spectrum network management software.

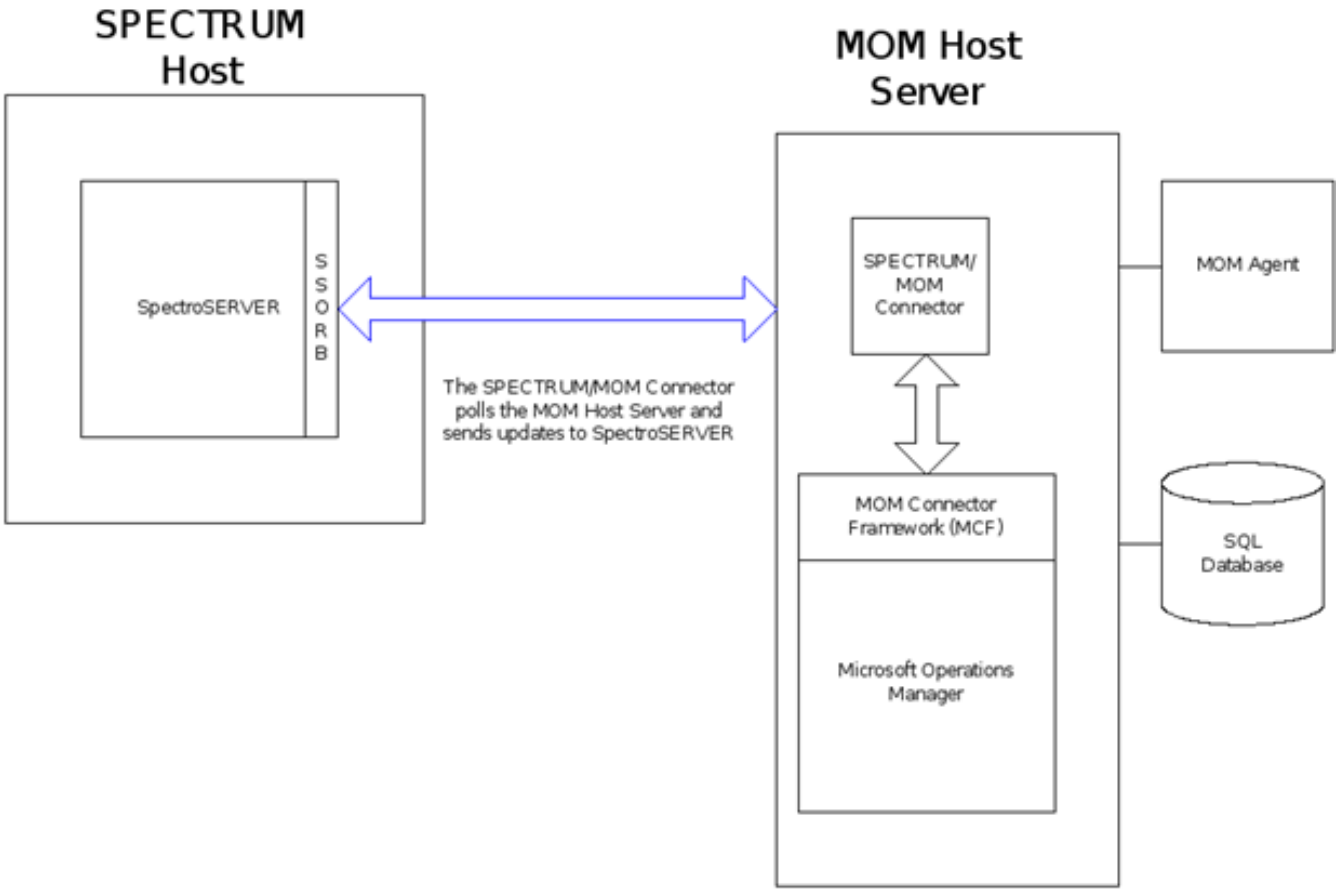
MOM Connector is an application that synchronizes alarm data between DX NetOps Spectrum and Microsoft Operations Manager. The MOM Connector uses the DX NetOps Spectrum SSORB CORBA API to interface with DX NetOps Spectrum. It uses the MOM Connector Framework (MCF) to interface with Microsoft Operations Manager.
You can monitor and respond to MOM-generated alert conditions by creating DX NetOps Spectrum events and alarms. You can also monitor the status of the MOM agents managed by the MOM application using DX NetOps Spectrum.
The MOM Connector performs the following tasks:

- Creates DX NetOps Spectrum alarms when alerts are generated in MOM.
- Clears DX NetOps Spectrum alarms when the resolution state of the corresponding MOM alert is set to "Resolved" (and the reverse).
- Acknowledges a DX NetOps Spectrum alarm when the resolution state of the corresponding MOM alert is set to "Acknowledged."

WARNING

The MOM connector is not fully supported in a DSS environment. For more information, see MOM Connector in a DSS Environment.

The following diagram illustrates the DX NetOps Spectrum/MOM architecture.



MOM Connector and Fault Tolerant Environments

If you are deploying a fault-tolerant environment with the MOM Connector running, you must restart the MOM Connector after fault tolerance has been set up. Restart the connector because the connector only checks the landscape map once to find a backup SpectroSERVER, typically during initialization or startup. After the MOM Connector has finished initializing, it does not check again to find a backup SpectroSERVER unless it is restarted.

MOM Connector in a DSS Environment

When the MOM Connector is deployed in a Distributed SpectroSERVER (DSS) environment, Microsoft Operations Manager only forwards alerts to a single connector. The connector must be configured to connect to the SpectroSERVER that is managing the same set of servers and hosts as the Microsoft Operations Manager server.
When the Microsoft server connects to a connector on the main location server (MLS), only models that are present on the MLS have corresponding DX NetOps Spectrum alarms created for Microsoft Operations Manager alerts. Any Microsoft Operations Manager alerts that are forwarded to the MLS are not subsequently forwarded to other location servers in the DSS environment. Therefore, the alarms are not raised on models in the other SpectroSERVERs in the environment.

MOM Connector Software Requirements

Verify the following prerequisites before you install the connector software:

- v9.2 or later.
- Microsoft Operations Manager 2005 (the only version that the DX NetOps Spectrum MOM Connector supports).
- The PATH environment variable on the Microsoft Operations Manager server is updated to include the path to the Microsoft .NET Framework.

Add the .NET Framework Path

You must add the path to the Microsoft .NET Framework to the PATH environment variable on the server running MOM. Verify that you have met the following prerequisites before adding the .NET path:

- The Microsoft .NET Framework, Version 1.1 is installed on the MOM Server Host.
  See the documentation provided with the Microsoft Operations Manager 2005 software or the relevant support website for the most accurate requirements.
- You know the path to the .NET Framework software.

Follow these steps:

1. Open the Windows Control Panel.
2. Double-click the System icon.
   The System Properties dialog opens.
3. Click the Advanced tab.
4. Click Environment Variables.
5. Select the Path variable in the System variables table and click Edit.
6. Add the .NET Framework path to the end of the value and click OK.
   Your changes are saved.

Install the MOM Connector

The following procedure describes how to install the MOM Connector software.

> **NOTE**
> In a Distributed SpectroSERVER (DSS) environment, the connector must be configured to connect to the SpectroSERVER that is managing the same set of servers and hosts as the Microsoft Operations Manager server. For more information, see MOM Connector in a DSS Environment.

Follow these steps:

1. Copy the <*$SPECROOT*>/MOMConnector directory to the MOM Server Host.

   > **NOTE**
   > Once you install the DX NetOps Spectrum MOM Connector on the MOM Server Host, you cannot move the directory; select a stable destination directory. For example: C:\Program Files\MOMConnector.
2. On the MOM Server Host, rename the file momrc.example to .momrc.
3. Open the .momrc file with a text editor.
4. Change the ssHost entry to the name of the SpectroSERVER Host. For example, ssHost=MOM01.
5. Execute the following command from the MOMConnector directory on the MOM Server Host:

   ```
   SpectrumMomConnector.exe --install
   ```

   This command sets up the required registry entries used by the MOM Connector and installs the MOM Connector as a Windows Service.
6. In the Windows Control Panel select Administrative Tools, Services.
   The Services dialog opens.
7. Double-click the Spectrum MOM Connector service.

8. Click the Log On tab and select the 'This account' option.
9. Choose a valid DX NetOps Spectrum user account, for example, Administrator.
10. Type and confirm the password for the account.
11. Click OK.
    The MOM Connector is now installed.

Add the MOM Host Server to the Host Security on SpectroSERVER

You must add the MOM Host Server name to the Server List on your SpectroSERVER to allow the servers to communicate.

Follow these steps:

1. Open the Spectrum Control Panel.
2. Select Configure, Host Security.
   The Host Security dialog opens.
3. Enter the MOM Host Server name in the text box under Server List.
4. Click Add.
   The MOM Host Server name is added to the Server List.
5. Click OK.
   Your changes are saved and the Host Security dialog closes.

Run the MOM Connector

Use the following procedure to run the MOM Connector.

Follow these steps:

1. Open the Windows Control Panel and select Administrative Tools, Services.
   The Windows Services dialog opens.
2. Select the Spectrum MOM Connector service.
3. Select Start from the Action menu.
   The MOM Connector service starts.

Verify that the MOM Connector is Running Properly

Verify that the MOM Connector is running from the Microsoft Operations Manager Administrator Console.

Follow these steps:

1. Open the MOM Administrator Console.
2. From the tree in the left pane, click Console Root, Microsoft Operations Manager (hostname), Administration, Product Connectors.
3. Verify that ‘SPECTRUM_Connector’ appears in the right pane.

Viewing MOM Alarms

When the MOM Connector receives alerts, it generates events based on the content of the alert. DX NetOps Spectrum then determines whether to generate an alarm.

After an alarm is generated, right-click the device model and select Alarm Details. You can view the cause of the alarm and the events that generated the alarm in the Alarm Details tab.

For information about sending MOM alerts to the MOM Connector, see Configure MOM Event Rules for Alert Forwarding.

Uninstall the MOM Connector

You can uninstall the MOM Connector if necessary.

Follow these steps:

1. Select Start, Control Panel, Administrative Tools, Services.
   The Services window opens.
2. Right-click the Spectrum MOM Connector service and select Stop.
3. Execute the following command from the MOMConnector directory on the host server:
   ```
   SpectrumMOMConnector.exe --remove
   ```
   The MOM Connector no longer appears in the Windows Services dialog. The connector is removed from the list of Product Connectors in the MOM Administrator Console.

MOM Connector Remains in Windows Services after Uninstallation

Symptom:

After I uninstalled the MOM Connector, the Connector remains in the Windows services dialog. The MOM Connector service is listed as Disabled. I refreshed the Windows services dialog, but the service remains in the list.

Solution:

When the MOM Connector service is in a 'Disabled' state, you cannot reinstall it.

Restart Windows. The Services dialog releases the handle. When you call up the Services dialog, the service is removed. You can now reinstall the MOM Connector.

Configure the MOM Connector

Configure MOM Event Rules for Alert Forwarding

An Event Rule specifies how a condition must be met for MOM to generate alerts to the DX NetOps Spectrum/MOM Connector. If the condition is met, an alert is generated. For an alert to be sent to the DX NetOps Spectrum/MOM Connector, the event resolution state must be set to SPECTRUM_Connector.

> **NOTE**
> See the Microsoft Operations Manager documentation for additional information about event processing rules that generate alerts.

Use the following procedure to create a new event rule that sends alerts that the rule generate to the SPECTRUM_Connector.

Follow these steps:

1. In Microsoft Operations Manager, select Console Root, Microsoft Operations Manager (hostname), Management Packs.
2. Select Rule Groups, Microsoft Operations Manager, Operations Manager 2005, Agent, Event Rules.
3. Right-click Event Rules and select Create Event Rule.
   The Select Event Rule Type dialog opens.
4. Select “Alert on or Respond to Event (Event)” as the rule type, and click Next.
   The Event Rules Properties - Event Provider dialog opens
5. Select “System” from the list box as the data provider and click Next.
   The Event Rules Properties - Criteria dialog opens
6. Click Next to proceed.

The Event Rules Properties - Schedule dialog opens.
7. Click Next to proceed.
   The Event Rules Properties - Event Provider dialog opens.
8. Click Next to proceed.
   The Event Rules Properties - Alert dialog opens.
9. Select Generate alert.
10. Select SPECTRUM_Connector from the Resolution state list box.
11. Click Next.
    The Event Rule Properties - Alert Suppression dialog opens.
12. Clear the option to 'Suppress duplicate alerts,' and click Next.
    The Responses dialog opens.
13. Click Next to proceed.
    The Event Rule Properties - Knowledge Base dialog opens.
14. Click Next to proceed.
    The Event Rule Properties - General dialog opens.
15. Enter a name for the event rule in the Rule Name field, and click Finish.

NOTE
If alerts are not forwarded to DX NetOps Spectrum immediately, the MOM server is updating. Wait a few minutes, and check again for alerts.

Configure Alert Rules to Modify Alerts Generated by MOM Event Rules

By default, alerts created by MOM event rules are not forwarded to the MOM Connector because their resolution state is set to something other than SPECTRUM_Connector. Instead of changing your MOM event rules, you can create an alert rule to forward alerts to DX NetOps Spectrum. You can create an alert rule that modifies alerts created by event rules so that they are forwarded to DX NetOps Spectrum.

Follow these steps:

1. In the MOM administrator console, create an Alert Rule with your specified criteria. Be sure that the Rule Group containing your Alert Rule is associated with the correct Computer Groups.
2. In the new alert rule's properties, in the Responses tab, click Add, and select Launch a script.
   The Launch Script dialog opens.
3. In the Launch a Script dialog, create a new script. The language should be set to VBScript. Use the following format:

```
Option Explicit
Sub Main()
 Dim myAlert
 'change resolution state
 Set myAlert = ScriptContext.Alert
 myAlert.ResolutionState = 211
End Sub
```

NOTE
The scripts set myAlert.ResolutionState to 211, which is the default value for the connectorID parameter. If you have modified the connectorID value in the .momrc file, you will need to change the value used to define myAlert.ResolutionState in the scripts to the connectorID value specified in the .momrc file.

4. Once you have entered the script source, click Next.
5. Click Finish.

NOTE
The script does not require any parameters.

6. In the Alert Rule Properties dialog, click OK.

7. Right-click Console Root and select Microsoft Operations Manager (<*host_name*>), Management Packs node.
8. Select Commit Configuration Change from the menu.
   It can take several minutes for the MOM system to update. When the update is complete, you see alerts that are forwarded to DX NetOps Spectrum.

Create Models for MOM Agents

Model each of the MOM agent hosts on your network so that you can view them in the Topology tab. When you model a MOM agent, DX NetOps Spectrum selects the host device model type that most accurately represents each MOM agent.

Follow these steps:

1. Click the Topology tab.
2. Click the Create a new model by IP icon in the toolbar.
   The Create Model by IP Address dialog opens.
3. Enter the Network Address, Community Name, and Agent Port for DX NetOps Spectrum to use to communicate with MOM agents.
4. Click OK.
   DX NetOps Spectrum creates a model that represents the host device with the specified IP address.

   NOTE
   If the model is not created successfully, verify that the information you entered in Step 3 is correct.
5. Select the new model in the Topology tab.
   The new model information is displayed in the Component Detail panel Information view.
6. Click 'set' next to the label displaying the IP address to the right of the model icon, and type the host name of the MOM agent host.
   This value is case-insensitive.
7. Press Enter to set the name.
   The MOM Connector uses one of the MOM alert properties to identify which model should receive the DX NetOps Spectrum event that is generated.
8. Repeat Step 1 through Step 7 for each MOM agent host in your network.

Install and Run the SCOM Connector

NOTE

10.3 supports SCOM 2016 and 2012 R2 Standard.

The SCOM connector is a Windows service that synchronizes alarm data between DX NetOps Spectrum and Microsoft System Center Operations Manager. The SCOM connector uses the DX NetOps Spectrum SSORB (CORBA) API to communicate with DX NetOps Spectrum and it uses the Operations Manager Connector Framework (OMCF) API to communicate with System Center Operations Manager.

WARNING
The SCOM connector versions are available to support multiple versions of System Center Operations Manager. With version 2012, Microsoft introduced some architectural changes that eliminated the Root Management Server (RMS). Therefore, your deployment differs fundamentally based on your version of Microsoft SCOM.

The SCOM connector synchronizes System Center Operations Manager alerts with DX NetOps Spectrum events and alarms. It provides bidirectional alert/alarm clearing and bidirectional alert/alarm acknowledgment. Thus, if you clear or acknowledge a System Center Operations Manager-related alarm in DX NetOps Spectrum, the corresponding alert is cleared or acknowledged in System Center Operations Manager, and the reverse.

WARNING
The SCOM connector is not fully supported in a DSS environment.

In DX NetOps Spectrum you can monitor and respond to System Center Operations Manager-generated alert conditions by generating DX NetOps Spectrum events and alarms. You can also monitor the status of the System Center Operations Manager agents that are managed by the Microsoft application using DX NetOps Spectrum.

The SCOM connector performs the following tasks:

- Creates DX NetOps Spectrum alarms in response to alerts that are generated in System Center Operations Manager.
- Clears DX NetOps Spectrum alarms when the resolution state of the corresponding System Center Operations Manager alert is set to "Closed" (and the reverse).
- Acknowledges a DX NetOps Spectrum alarm when the resolution state of the corresponding System Center Operations Manager alert is set to "Acknowledged."

> **NOTE**
> By default, the SCOM connector does not include a resolution state for acknowledging an alert; you must create it. For more information about creating this resolution state, see Create an Acknowledged Resolution State.

Installing and Configuring the SCOM Manager

The supported operating system is Windows 2016 and 2012R2.

> **NOTE**
> DX NetOps Spectrum 10.4.2 has not been validated on Windows Server 2012. However, Broadcom will support any DX NetOps Spectrum product issues, if found. We reserve the right to have you upgrade to Windows Server 2016 (or later) if deemed necessary.

Prerequisites

Following are the prerequisites to install SCOM Manager:

1. Install CGI and Microsoft .NET FrameWork v.4.1 or above.
2. IIS should be installed and all ISS services are to be installed.
3. Enable IIS and CGI configuration.
4. Install reporter.exe.
5. Add the machine to the domain.
6. Install SQL 2014 SP1 with the default instance.
7. Ensure whether all the SQL 2014 SP1 services have been started.

Configuring

1. Navigate to the '**Operations Manager Set up - Configuration'** window.
2. Select either one of the installation options:
   - Create the first management server in a new management group: This option is best suited for first-time installation.
   - Add a management server to an existing management group: This option is best suited for those who want to add another management group on top of an existing one.
3. To create a management group for the first time, specify the management group name as SCOM2016.
   The 'Configure Operational Database' window appears.
4. Provide the SQL Server name and the instance name.
   The 'Configure Data Warehouse Database' window appears.
5. Verify the database name and the port. Ensure that you have sufficient permissions on the database instance.
   The 'SQL Server for reporting instances' window appears. This window is to configure report services and web console.
6. Select Next.
   The 'Specify a website for use with the web console' window appears.
7. Select the default website and select Next.

The 'Authentication Mode for Web Console' window appears.

8. Select 'Use Mixed Authentication' and select Next.
   The 'Configure Operations Manager accounts' window appears.
9. Select the 'Management server action account' option for 'Local System' and then select the 'System Center Configuration' option for 'Local System'.
10. Enter the credentials for the 'Data Reader account' and the 'Data Writer account', select Next.
    The 'Microsoft Window' appears.
11. Turn off the Microsoft updates, by enabling the 'Off' radio button and select Next.
    The 'Setup is complete' window appears.
12. Once the setup is completed, refer to the Microsoft support site, to update the product key for SCOM.

Installing

To Install SCOM connector:

1. Copy the SCOM folder from the SpectroSERVER machine. By default it is at 'C:\win32app\Spectrum$'
2. Install the SCOM connector from the command prompt. Refer to the section on Installing SCOM connector for more information.
3. After installing the SCOM connector, refer to the section on SCOM Configuration for Spectrum Manager.
4. Subscribe the SCOM connector with the SCOM Manager to receive alerts from the SCOM Manager to Spectrum Manager.

SCOM Connector Versions

Changes to the SCOM connector in CA Spectrum 10.3 enable integration with Microsoft System Center Operations Manager 2016 and 2012.

Previous versions of Microsoft SCOM used a Root Management Server (RMS), which represented a single point of failure. With SCOM 2012, this server has been eliminated. As a result, CA has changed the SCOM connector architecture for deployment with SCOM 2012.

Previous versions of the connector were deployed with one SCOM connector on the RMS. Because alerts from all SCOM servers were propagated to the RMS, one DX NetOps Spectrum SCOM connector could send all SCOM alerts to DX NetOps Spectrum.

With no RMS running in the Microsoft SCOM environment, you must deploy a connector on each SCOM management server from which you want to collect alerts.

DX NetOps Spectrum architecture dictates that each connector can only be associated with a single landscape. Therefore, all DX NetOps Spectrum landscapes must reflect the devices that each management server is managing. Otherwise, the alerts are mapped under the VNM model.

The SCOM connector version for SCOM 2012 does not support new SCOM functionality, such as fault tolerance.

SCOM Connector Software Requirements

- Microsoft System Center Operations Manager 2016 or Microsoft System Center Operations Manager 2012. Comparable support is available for both versions of SCOM. New functionality that is introduced in Microsoft System Center Operations Manager 2012 is not supported. However, the SCOM connector for Microsoft System Center Operations Manager 2016 and 2012, offers better support for DSS deployments.
- Microsoft .NET Framework 3.0 or later. This software must be installed on the SCOM connector Host.

WARNING
On Windows 2012 Server, the .NET Framework v3.5 is required. If v4.0 of the .NET Framework is installed, the connector application fails. Be aware that .NET v4.0 is installed on this platform by default.

- Microsoft SCOM uses the SpectrumSCOMConnector.exe file to communicate with DX NetOps Spectrum. You must have vcredist_x86 (VS2008) for SpectrumSCOMConnector.exe to work properly.

SCOM Connector and Fault Tolerant Environments

If you are deploying the SCOM connector in a fault-tolerant environment, restart the connector after you set up fault tolerance. The SCOM connector only checks the landscape map once to find a backup SpectroSERVER. Typically this check is performed during initialization or at startup. After the SCOM connector has completed the initialization, it does not check again for a backup SpectroSERVER unless you restart it.

SCOM Connector in a DSS Environment

When the SCOM connector is deployed in a Distributed SpectroSERVER (DSS) environment, System Center Operations Manager only forwards alerts to a single connector. The connector must be configured to connect to the SpectroSERVER that is managing the same set of servers and hosts as the System Center Operations Manager server.

When the Microsoft server connects to a connector on the main location server (MLS), only models that are present on the MLS have corresponding DX NetOps Spectrum alarms that are created for System Center Operations Manager alerts. Any System Center Operations Manager alerts that are forwarded to the MLS are not subsequently forwarded to other location servers in the DSS environment. Therefore, the alarms are not raised on models in the other SpectroSERVERs in the environment.

For Microsoft System Center Operations Manager 2012R2

Microsoft System Center Operations Manager 2012 no longer uses a Root Management Server (RMS). Instead, each SCOM Management Server is a peer with other Management Servers in its own management group.

Deploy multiple SCOM connectors to support SCOM 2012 in a DSS environment. Each connector must be subscribed to single SCOM Management Server and must be configured to connect to a single SpectroSERVER in the DSS environment. The connector must be configured to connect to the SpectroSERVER that is managing the same set of servers and hosts as the SCOM Management Server. With this configuration, the connector supports a DSS configuration.

Update the .scomrc file on each connector to include the hostname of each Management Server (RMS) in the "scomHost" field.

Before Installing the SCOM Connector

The topics in this section assume the following:

- You have installed and configured a Microsoft System Center Operations Manager (SCOM) Management Server.
- You are integrating your System Center Operations Manager environment with DX NetOps Spectrum network management software.
- You are aware of basic deployment differences between versions of the connector.
  Versions of System Center Operations Manager prior to version 2012 and version 2012 and later have fundamental architectural differences. For more information, see SCOM Connector Versions.

Before installing the SCOM Connector, verify the following:

User Access

The user account under which the SCOM Connector will run must have access to both DX NetOps Spectrum and Microsoft System Center Operations Manager.

- In DX NetOps Spectrum, use the Users tab in the OneClick Console to create a DX NetOps Spectrum user model for the SCOM Connector user. The user model should be an administrator user but not necessarily a super user account.

  WARNING
  The user model must already exist in DX NetOps Spectrum before you install the SCOM Connector.
- In Microsoft System Center Operations Manager, verify that the connector user is a member of the Administrator User Role. By default, the Administrator User Role in SCOM contains the local Administrator user group as a User Role Member. Therefore, to give the user access to SCOM, you can add the user to the local Administrator user group using the Windows Computer Management dialog.

Host Access

If the SCOM Connector is running on a host other than the SpectroSERVER host, add the SCOM Connector host to the SpectroSERVER host security. You can add the host by using SCP or by editing the *<$SPECROOT>*/.hostrc file.

For more information, see Add the SCOM Host Server to the Host Security on SpectroSERVER.

Communication Ports

Firewalls must be configured to allow traffic to pass on certain ports:

- On the remote SpectroSERVER host, the connector tries to reach the Naming Service on Port 14006 by default.

  NOTE
  We recommend verifying firewall configuration to enable connections to the required SSORB communication ports. For more information, see the Distributed SpectroSERVER Administration section.
- On the remote SCOM host, the connector tries to reach the Connector Framework on Port 5724 by default. Verify that Port 5724 is open between the SCOM Connector and the Connector Framework on the RMS.
- By default, the CORBA API listens for communications from the connector on Port 14001.
- SpectroSERVER for TCP/CORBA uses port 14002.
- LocServ on SpectroSERVER for TCP/CORBA uses port 14004.
- You can set a different connector listening port in a configuration file. For more information, see Configure Connector Communication Settings.

Acknowledged Resolution State

To enable bidirectional alert/alarm acknowledgment, you must create a custom resolution state within SCOM. The new resolution state is used to represent an alert that has been “Acknowledged.”

Host Names and Model Names

The host name of the SCOM alert must match the DX NetOps Spectrum model name for the integration to operate properly. Case is not considered. If the host name that arrives in DX NetOps Spectrum in an SCOM alert does not match the DX NetOps Spectrum model name to which it applies, the alarm is not raised or cleared. Synchronization of host

names for SCOM alerts and model names in DX NetOps Spectrum must be maintained for the integration to operate properly.

When creating model names in DX NetOps Spectrum, consider establishing an internal policy such as setting the sysName for SNMP agents to the host name or fully qualified domain name (FQDN) for consistency. Use the Model Naming Order setting in the SpectroSERVER Control subview for the VNM model to control how models are named.

NOTE
Additional documentation discusses the Model Naming Order setting and the SpectroSERVER Control Subview. For more information, see the Modeling and Managing Your IT Infrastructure section.

Install the SCOM Connector

The following procedure describes how to install the SCOM Connector software.

NOTE
In a Distributed SpectroSERVER (DSS) environment, the connector must be configured to connect to the SpectroSERVER that is managing the same set of servers and hosts as the System Center Operations Manager server. For more information, see SCOM Connector in a DSS Environment.

Follow these steps:

1. Copy the <*$SPECROOT*>/SCOMConnector directory to the server where you plan to deploy the SCOM Connector.
   WARNING
   Once you install the SCOM Connector on the server host, you cannot move the directory. Select a stable destination directory. For example, select C:\Program Files\SCOMConnector.
2. On the host server, rename the file "scomrc.example" to ".scomrc".
3. Open the .scomrc file with a text editor and modify the parameters as needed.
   NOTE
   The .scomrc file contains descriptions of each of the available parameters.
   a. Specify the ssHost and scomHost parameters if the connector is remote from one or both of the management systems. For example, if the SCOM Connector is installed on the SCOM host and the SpectroSERVER is on a different host, specify a value for the ssHost parameter to enable the connector to reach DX NetOps Spectrum.
   b. Specify the scomAckResolutionState parameter to enable bidirectional alarm/alert acknowledgment. The value of this parameter is the numeric identifier of the 'Acknowledged' resolution state in System Center Operations Manager.
      NOTE
      To enable bidirectional alarm/alert acknowledgment, create an Acknowledged resolution state. For more information, see Create an Acknowledged Resolution State.
   c. Set the scomWebHost parameter to point the Alert URL in DX NetOps Spectrum events to a SCOM web server. This parameter lets you specify a different server than the SCOM server.
      This value defaults to scomHost. If either scomWebHost nor scomHost is set, this value defaults to localhost.
   d. (Optional) Specify the web URL port that is used to access the SCOM web console. For SCOM 2012, the default port is 80 and for SCOM 2016 the default port is 1433.
   e. (Optional) For SCOM 2012, specify the web console application name of the SCOM web console, where applicable. For example, type http://SERVER_NAME: Port/OperationsManager. The default is OperationsManager.
4. Execute the following command from the SCOMConnector directory on the host server:
   ```
   SpectrumSCOMConnector.exe --install
   ```
   This command sets up the required registry entries for the SCOM Connector and installs the SCOM Connector as a Windows Service.
5. In the Windows Control Panel, select Administrative Tools, Services.
   The Services dialog opens.
6. Double-click the Spectrum SCOM Connector service.

7. Select the login tab, and select ‘This account’.
8. Select a user account for the connector to run under, such as Administrator. Select a valid DX NetOps Spectrum user.
9. Type and confirm the password for the account.
10. Select OK to accept your changes.
    The SCOM Connector is now installed.

Add the SCOM Host Server to the Host Security on SpectroSERVER

Add the SCOM Connector host name to the Server List on your SpectroSERVER to enable the servers to communicate.

Follow these steps:

1. Open the DX NetOps Spectrum Control Panel.
2. Select Configure, Host Security.
   The Host Security dialog opens.
3. Type the SCOM Connector host name in the text box under Server List.
4. Click Add.
   The SCOM Connector host name is added to the Server List.
5. Click OK.
   Your changes are saved and the Host Security dialog closes.

> **NOTE**
> You can also add the SCOM Connector host to the host security on the SpectroSERVER by editing the <*$SPECROOT*>/.hostrc file.

Configure Connector Communication Settings

You can set the SCOM Connector listening port in a configuration file. Setting a static port is a recommended best practice. When the connector instead selects a random listening port, intervening firewalls can disrupt communications among components in a distributed environment. The configuration file (.corbarc) must be present in the <*SCOMCONNECTOR*> folder on the server where the SCOM Connector is installed.

Follow these steps:

1. On the host server, navigate to the directory where you have copied the SCOM Connector folder.
2. Rename the corbarc.example file to '.corbarc'.
3. Open the .corbarc file using your preferred text editor.
4. Change the variable (indicated by italics) in the following line to match the hostname in your environment:
   ```
   vbroker.se.iiop_tp.proxyHost= <hostname of the server where the Connector is installed>
   ```
5. Change the variable (indicated by italics) in the following line to match the port that you want to configure as the static listening port for the connector:
   ```
   vbroker.se.iiop_tp.scm.iiop_tp.listener.port=<port number>
   ```

   > **WARNING**
   >
   > - The default port is 14001. This port must be open between the connector and the SpectroSERVER.
   > - If the connector is installed on a machine where Spectrum is running, configure the connector listening port to the port that Spectrum is not using.
   >
   > Execute netstat –a | grep 14001 to check whether the 14001 port is in use.
6. Save the .corbarc file.
7. When you have started the connector with the above changes applied, verify the change by issuing the following commands:

```
netstat -ano | grep <SCOM Connector pid>
```

The information that is returned verifies whether the connector is using the listener port that you specified in the .corbarc file.

Start the SCOM Connector

The following procedure describes how to run the DX NetOps Spectrum/SCOM Connector.

Follow these steps:

1. Open Windows Control Panel and select Administrative Tools, Services.
   The Windows Services dialog opens.
2. Select the Spectrum SCOM Connector service.
3. Select Start from the Action menu to start the service.
   The SCOM Connector starts running.

Verify that the SCOM Connector is Running Properly

You can use SCOM Administration to verify that the SCOM Connector is running.

Follow these steps:

1. Open the SCOM Operations Console.
2. Change to the Administration context in the console.
3. From the tree in the left pane, expand Administration and select Product Connectors.
4. Verify that 'SPECTRUM Connector' appears in the right pane.

Viewing SCOM Alarms

When the SCOM Connector receives alerts, it generates events based on the content of the alert. DX NetOps Spectrum then determines whether to generate an alarm.

> **NOTE**
> The host name of the SCOM alert must match the DX NetOps Spectrum model name (except for case) for the integration to operate properly. For more information, see Host Names and Model Names.

After an alarm is generated, right-click the device model and select Alarm Details. You can view the cause of the alarm and the events that generated the alarm in the Alarm Details tab.

See Set Up SCOM Connector Subscriptions for information about sending SCOM alerts to the SCOM Connector.

Uninstall the SCOM Connector

The following procedure describes how to uninstall the SCOM Connector.

To uninstall the SCOM Connector

1. Select Start, Control Panel, Administrative Tools, Services.
   The Services window opens.
2. Right-click the Spectrum SCOM Connector service and select Stop.
3. Execute the following command from the SCOMConnector directory on the host machine:
   ```
   SpectrumSCOMConnector.exe --remove
   ```
   The SCOM Connector no longer appears in the Windows Services dialog and is removed from the list of Product Connectors in the SCOM Operations Console.

Configure the SCOM Connector

Set Up SCOM Connector Subscriptions

Set up a connector subscription in System Center Operations Manager. The subscription lets the SCOM Connector forward alerts to DX NetOps Spectrum.

Follow these steps:

1. After you have started the SCOM Connector for the first time, you will see DX NetOps Spectrum Connector that is listed as a Product Connector in the SCOM Operations Console. Right-click DX NetOps Spectrum Connector in the Operations Console and click Properties.
   The Product Connector Properties dialog opens.
2. Click Add in the Subscription section to create a new connector subscription.
   The Product Connector Subscription Wizard opens.
3. Enter a subscription name and a description in the General page and click Next.
4. Select the groups whose alerts you want forwarded to DX NetOps Spectrum in the Groups page and click Next.
5. Select the targets whose alerts you want forwarded to DX NetOps Spectrum in the Targets page and click Next.
6. Select the appropriate criteria for the alerts that are forwarded to DX NetOps Spectrum in the Criteria page.
7. Click Create.
   The SCOM Connector subscription is created.

Create Models for SCOM Agents

You can model each of the SCOM agent hosts on your network so that you can view them in the Topology tab. When you model an agent, DX NetOps Spectrum selects the host device model type that most accurately represents each agent.

Follow these steps:

1. Log in to OneClick.
2. In the Landscapes area, select Universe Node.
3. Click the Topology Tab in the Contents Panel.
4. Click the Create a New Model by IP icon in the toolbar.
   The Create Model by IP Address dialog opens.
5. Enter the Network Address, Community Name, and Agent Port for DX NetOps Spectrum to use to communicate with SCOM agents.
6. Click OK.
   DX NetOps Spectrum creates a model that represents the host device with the specified IP address.

   NOTE
   If the model is not created successfully, verify that the information that you entered previously is correct.
7. Select the new model in the Topology tab.
   The new model information is displayed in the Component Detail panel Information view.
8. Change the name of the model to the host name of the server that hosts the Operations Manager agent, as follows:
   1. Select *Set* next to the label displaying the IP address to the right of the model icon in the Information view.
   2. Enter the *host name of the SCOM agent* host. This value is case-insensitive.
   3. Select *Enter* to set the name.
9. The SCOM Connector uses one of the System Center Operations Manager alert properties to identify the model that receives the DX NetOps Spectrum event that is generated.
10. Repeat the previous steps for each agent host on your network.

For more information, see Modeling and Managing Your IT Infrastructure section.

Create an Acknowledged Resolution State

By default, the SCOM Connector does not include a resolution state for acknowledging an alert. Instead, you must create one. Once you have created the resolution state in SCOM, the Connector can be configured to use that resolution state to synchronize the acknowledgment of alerts and alarms. By default, however, the SCOM Connector does not provide acknowledgment synchronization.

Follow these steps:

1. On the SCOM host, open the SCOM Operations Console.
2. In the Operations Console, select Administration, Settings.
3. Right-click Alerts in the central table.
4. In the Alert Resolution States tab, click New.
5. Type a name for the new resolution state, and select a Unique ID.

> **NOTE**
> This Unique ID is used to configure the scomAckResolutionState parameter in the .scomrc file.

Troubleshoot the SCOM Connector

SCOM Alerts Not Synchronized with DX NetOps Spectrum Alarms

Symptom:

When a System Center Operations Manager alert arrives in DX NetOps Spectrum, the appropriate DX NetOps Spectrum alarm is not raised or cleared.

Solution:

The host name of the System Center Operations Manager alert must match the DX NetOps Spectrum model name (except for case) for the integration to operate properly. If the host name that arrives in DX NetOps Spectrum in an SCOM alert does not match the DX NetOps Spectrum model name for which it applies, the alarm is not raised or cleared. Synchronization of host names for SCOM alerts and model names in DX NetOps Spectrum must be maintained for the integration to operate properly.

> **WARNING**
> If you have applied any hotfixes to your DX NetOps Spectrum environment, make sure that you have also upgraded the SCOM Connector. The installed DX NetOps Spectrum and SCOM Connector versions must be the same for the integration to operate properly.

Alert States Not Updated

Symptom:

I have installed the SCOM Connector and I have configured the integration. I have even followed the steps in Create an Acknowledged Resolution State. The System Center Operations Manager alerts are coming through as alarms in DX NetOps Spectrum. However, when I acknowledge or unacknowledge an alarm in DX NetOps Spectrum, the resolution state does not update in SCOM. I eventually concluded that the SCOM Connector uses a random listening port each time it is restarted.

Solution:

To resolve the issue with alert states that are not updated on multiple servers, first check your firewall settings. Ports 14006 and 14001 (or the port that is set in the .corbarc file, if applicable) must be open between the SpectroSERVER and the SCOM Connector to enable bidirectional communication between these components.

The SCOM Connector uses listening port 14001 by default to receive communications from the CORBA API. If that port is in use, you can set another port in a CORBA configuration file. Edit the .corbarc file to include the listening port that you want to use.

SpectroSERVER for TCP/CORBA uses port 14002 and LocServ on SpectroSERVER for TCP/CORBA uses port 14004.

> **NOTE**
> If the connector is installed on a machine where Spectrum is running, configure the connector SCOM listening port to the port that Spectrum is not using.
>
> Execute netstat - a | grep 14001 to check whether the 14001 port is in use.

We have included an example of this file to guide you in configuring the applicable parameters. For more information, see the Configure Connector Communication Settings section.

SCOM Connector Loses Ownership of of Alerts After Reinstallation

Symptom:

If you uninstall and then reinstall the SCOM Connector, the Connector can lose ownership of SCOM Connector alerts.

Solution:

Manually forward the alerts again to the newly installed SCOM Connector.

SCOM Connector Remains in the Windows Services Dialog after Uninstallation

Symptom:

After I uninstalled the SCOM Connector, the Connector service remains in the Windows services dialog. The SCOM Connector service is listed as being in the “Disabled” state. Refreshing the Windows services dialog does not remove the SCOM Connector from the list.

Solution:

When the SCOM Connector is in this state, you cannot reinstall it. Restart the server. After Windows restarts, the Services dialog releases the handle and the service is removed. You can now reinstall the SCOM Connector.

SCOM Connector not available for Spectrum on Linux

Symptom:

How can the Spectrum SCOM Connector be installed to monitor SCOM environments when the Spectrum Server is only available on Linux.

The Spectrum SCOM Connector binaries only exists for Windows Platforms.

Solution:

1. Check the Spectrum License in $SPECROOT/Install-Tools/LOGS/<SpectrumRelease>/summary* file and make sure the entry "Microsoft MOM Connector" exists.
2. Check the $SPECROOT/Install-Tools/.history file for the current release and patch level of Spectrum.
3. A standalone SCOMConnector package is not available. So if the Spectrum Windows installer is available, the $SPECROOT/SCOMConnector can be extracted to the SCOM Windows System based on the extraction key or a Support Issue can be opened with the .history information to request a Zip Archive for the Windows System of the $SPECROOT/SCOMConnector directory.
4. The Spectrum SCOM Connector has to be run on the SCOM Server or on another Windows Server that has the .NET framework 3.0 installed.

Launch the Web Console

Launch the Web Console from the OneClick Console

The DX NetOps Spectrum integration with MOM and SCOM lets you launch the applicable Web Console from the OneClick Console.

To launch the MOM Web console from OneClick

1. Select any DX NetOps Spectrum model that has an associated MOM-based alarm.
2. Click the Alarms tab in the Contents panel.
3. Click the Alarm Details tab in the Component Detail panel.
4. Click the omAlertURL at the bottom of the window to display alarm details.

> **NOTE**
> A hyperlink is not shown for alarms that lack an associated URL.

To launch the SCOM Web console from OneClick

1. Select any DX NetOps Spectrum model that has an associated MOM- or SCOM-based alarm.
2. Click the Alarms tab in the Contents panel.
3. Click the Alarm Details tab in the Component Detail panel.
   The Web Context URL provides a link labeled, “Click here to launch.”

Configure the SCOM Connector to Provide HTTPS URLs

SCOM supports HTTPS for the web console. You can configure the SCOM Connector to provide HTTPS URLs.

Follow these steps:

1. Open the .scomrc file with a text editor.
2. Specify the scomWebPrefix parameter with a value of https, as follows:
   ```
   scomWebPrefix = https
   ```
3. Save the .scomrc file.
4. Restart the SCOM Connector.
   Your changes are applied.

Change the OneClick Web Context URL

You can change the context URL for launching the MOM Web Console from OneClick. This procedure requires a working knowledge of XML and HTML. Although this procedure applies to both MOM and SCOM, it uses MOM as an example.

> **NOTE**
> You can preserve the customized XML that is created in this procedure from being overwritten during a DX NetOps Spectrum or OneClick upgrade or a reinstallation. For more information, see the OneClick Customization section.

Follow these steps:

1. In the OneClick installation directory (<*$SPECROOT*>), navigate to the following location:
   <*$SPECROOT*>/tomcat/webapps/spectrum/WEB-INF/alarm/config
2. Open the column-alarmwebcontexturl-config.xml file for editing.
3. Modify the text between the <html> and </html> tags to display an alternate hyperlink in OneClick. For example, replace the existing text to display a hyperlink in OneClick that reads, "Launch MOM Web Console".
   Replace this text:
   ```
   "<html>Click <a href='" + (String)value() + "'>here</a> to launch</html>"
   ```

With the following text:

```
"<html><a href='" + (String)value() + "'>Launch MOM Web Console</a></html>"
```

Events

This chapter shows the DX NetOps Spectrum events that are generated based on alerts received from MOM- or SCOM-managed hosts. The tables in this chapter contain the following information:

- **DX NetOps Spectrum Event**
  The DX NetOps Spectrum event created based on a MOM or SCOM alert.
- **Event Code**
  The DX NetOps Spectrum event code for the event.
- **Event Action**
  The processing that is performed on the event based on the instructions in the event disposition file. For example, raise an alarm, clear an alarm, or check for a frequent problem.
- **Alarm Code**
  The alarm code generated or cleared by the event.
- **Alarm Severity**
  The severity of the alarm generated.

Supported MOM Connector DX NetOps Spectrum Events

The following table lists the DX NetOps Spectrum events supported by the MOM Connector.

DX NetOps Spectrum Event	Event Code	Event Action	Alarm Code	Alarm Severity
Success	0x3e0000a	N/A	N/A	N/A
Information	0x3e0000b	N/A	N/A	N/A
Warning	0x3e0000c	Alarm Generated	0x3e0000c	Minor
Error	0x3e0000d	Alarm Generated	0x3e0000d	Major
Critical Error	0x3e0000e	Alarm Generated	0x3e0000e	Critical
Security Breach	0x3e0000f	Alarm Generated	0x3e0000f	Critical
Service Not Available	0x3e00010	Alarm Generated	0x3e00010	Critical

Supported SCOM Connector DX NetOps Spectrum Events

The following table lists the DX NetOps Spectrum events supported by the SCOM Connector.

DX NetOps Spectrum Event	Event Code	Event Action	Alarm Code	Alarm Severity
Information	0x3e00012	N/A	N/A	N/A
Warning	0x3e00013	Alarm Generated	0x3e00011	Minor
Error	0x3e00014	Alarm Generated	0x3e00012	Critical

Nortel Preside MDM

Nortel Preside Multiservice Data Manager (MDM) is an element management system (EMS) for managing faults, configuration, accounting, performance, and security of Nortel's telecommunications network and devices.

NOTE
For details about Nortel MDM operation and concepts, see the Nortel product documentation.

The MDMConnector integrates Nortel Preside MDM telecommunication networks and devices with DX NetOps Spectrum Telco EMS Manager. MDMConnector runs on the same machine as the MDM application. Nortel Preside MDM uses a node to represent any managed element, including devices, ports, cards, shelves, and so on. Links, which are a physical or logical connection between two nodes, are also supported. When you discover and model a network managed by MDM, the MDMConnector queries the MDM Network Modeling server for nodes and links and forwards them to DX NetOps Spectrum. MDMConnector also registers with the MDM Alarm and Status server for events and forwards them to DX NetOps Spectrum. MDMConnector periodically checks the status of MDM and reports it to DX NetOps Spectrum along with its own status through the heartbeat mechanism.

Installing and Configuring MDMConnector

Install the MDMConnector Files

The MDMConnector executable and all associated configuration files are installed on the SpectroSERVER host machine in the TelcoEMSManager directory. The files are bundled into the MDMConnector.tar file, which you must copy to the MDM Server host machine.

To copy and install the MDMConnector files

1. Copy the MDMConnector.tar file to a new, empty directory on the MDM Server host machine.
2. From the directory on the MDM server host, run the following command:

   ```
   $ tar xvf MDMConnector.tar
   ```

   The following files appear in the directory:
   - MDMConnector
   - MDMConnector.cfg
   - MDMConnector.lst
   - MDMConnector.sh
   - libtelcohelper.so.1
   - libssorbconvert.so.1
   - libVPapi.so.1
   - libcosnm_r.so
   - libssorb.so.1
   - libssorbutil.so.1
   - libGlobl.so.1
   - libtelcocorba.so.1
   - liborb_r.so
   - libvport_r.so
   - libPort.so.1

MDMConnector Configuration

You can configure MDMConnector with the MDMConnector.cfg configuration file. This file is in the same directory as the MDMConnector binary. All parameters should be specified in the 'name = value' format. MDMConnector runs with default

values if the configuration file is not present or if parameters are missing. The following lists the configuration parameters with their default values:

- **SPECTRO_SERVER**
  Specifies the name of the machine where the SpectroSERVER is running.
  Default: localhost
- **TRACE_FILE**
  Specifies the file name of the log file.
  Default: MDMConnector.log
- **TRACE_LEVEL**
  Specifies the trace level for the log messages. The possible values are:
  0 = No messages are logged
  1 = Only error messages are logged
  2 = Detail log for debugging
  Default: 0
- **SPEC_FILE**
  Specifies the file specifying the discovery criteria. MDMConnector discovers components from MDM based on contents of this file.
  Default: MDMConnector.lst
- **EMS_ID**
  Specifies the EMS identifier. This should be unique for every EMS (MDM server) that DX NetOps Spectrum monitors.
  Default: MDM_*<local machine>*
- **EMS_VENDOR**
  Specifies the vendor name for managed components.
  Default: CA
- **TYPES_DIR**
  Specifies the name of the directory where MDM stores the .ltdf and .mtdf files which contain information about different component types. The default value is the current directory.
- **MDM_USER**
  Specifies the MDM username used for registering with MDM.
  Default: mdpadmin
- **MDM_ALARM_FORMAT**
  Specifies the alarm format used while displaying MDM alarms. The possible values are:
  – 0 = EPI_ALARM_TERSE_FORMAT
  – 1 = EPI_ALARM_NORMAL_FORMAT
  – 2 = EPI_ALARM_FULL_FORMAT
  Default: EPI_ALARM_FULL_FORMAT
- **OSAGENT_PORT**
  Specifies the ORB Agent Port to connect to the SpectroSERVER.
  Default: 14008

Discovery Specification File

You can specify a list of nodes and links to discover in the discovery specification file. The default discovery specification file is the MDMConnector.lst file. The MDMConnector.lst file has four sections:

- **node**
  Specifies the list of nodes you want to discover. 'All' discovers all the nodes.
- **nodeTypeId**
  Specifies the list of node types that you want to discover.
- **link**

Specifies the list of links to discover. 'All' discovers all the links.

- **linkTypeId**
  Specifies the list of link types to discover. All the links of the specified types will be discovered.

Start the MDMConnector

You must start the MDMConnector as the same user account that started MDM.

To start MDMConnector

1. Log in to the MDM server machine as the MDM user.
2. Change to the directory where the MDMConnector files are installed on the MDM server.
3. Enter the following command:

```
$ ./MDMConnector.sh
```

Discovery and Modeling for MDMConnector

Discover MDM Equipment

Discovery is the process in which DX NetOps Spectrum communicates with all installed Nortel Preside MDMConnectors and directs them to collect information about the Nortel MDM nodes. Each MDMConnector then communicates with its associated Nortel Preside MDM server and retrieves the desired network and device information. The MDMConnector then sends the discovery data back to DX NetOps Spectrum to be modeled in the database. DX NetOps Spectrum's Telco EMS Manager can discover any or all managed elements and links in a Nortel Preside MDM.

To perform a discovery of all desired MDM equipment

1. Navigate to the Telco EMS Manager application in OneClick by expanding the desired landscape icon in the Navigation panel, and then selecting Telco EMS Manager.
2. In the Information tab of the Component Detail panel, expand the Configuration subview to display Telco Discovery.
3. Click Discover.
   The Discovery process begins; any discovered MDM nodes are added to the landscape.

MDM Modeling

DX NetOps Spectrum stores all the Nortel Preside MDM equipment data in the SpectroSERVER's modeling database.

Nortel Preside MDM uses a node to represent any managed element, including devices, ports, cards, shelves, and so on. Links, which are a physical or logical connection between two nodes, are also supported. For this reason, DX NetOps Spectrum uses two different model types to model any node or link in a Nortel Preside MDM: TelcoEMSManagedElement (0x4fd0001) and TelcoEMSManagedLink (0x4fd0003).

> **NOTE**
> Dynamic models are created inside the Nortel Preside MDM only when a problem exists on a Static model and an alarm needs to be created. When there are no remaining alarms on a Dynamic model, it may be destroyed by the Nortel Preside MDM. Static models are persistent within the Nortel Preside MDM until the model is no longer being managed by the Nortel Preside MDM. To maintain event and alarm history when a Dynamic model is destroyed within a Nortel Preside MDM, the Dynamic model is not automatically destroyed in DX NetOps Spectrum.

A TelcoEMSManagedElement or TelcoEMSManagedLink model can be copied and pasted anywhere in the DX NetOps Spectrum network topology views (Universe, World, or TopOrg). This lets you arrange your network containment based on your organization needs. DX NetOps Spectrum associates TelcoEMSManagedElement models appropriately to reflect the hierarchical relationship of managed elements (device-> board-> port-> logical port) in MDM.

TelcoEMSManagedElement Model Type Attributes

The following list describes the TelcoEMSManagedElement model type attributes.

- **EmsID**
  The name of the Nortel Preside MDMConnector or Nortel Preside MDM station where TelcoEMSManagedElement is located.
  Attribute ID: 0x4fd0003
- **EmsComponentId**
  The unique name that identifies the TelcoEMSManagedElement in the associated Nortel Preside MDM and within DX NetOps Spectrum.
  Attribute ID: 0x4fd0000
- **EmsElementDescr**
  The description given to this element inside the Nortel Preside MDM.
  Attribute ID: 0x4fd0004
- **EmsElementType**
  The type given to this element inside the Nortel Preside MDM.
  Attribute ID: 0x4fd0005
- **EmsVendor**
  The vendor of this element inside the Nortel Preside MDM.
  Attribute ID: 0x4fd0007
- **EmsPersistence**
  The persistence type of this element, either Static or Dynamic.
  Attribute ID: 0x4fd0008

TelcoEMSManagedLink Model Type Attributes

The following list describes the TelcoEMSManagedElement model type attributes.

- **EmsId**
  The name of the Nortel Preside MDMConnector or Nortel Preside MDM station where TelcoEMSManagedLink is located.
  Attribute ID: 0x4fd0003
- **EmsComponentId**
  The unique name that identifies this TelcoEMSManagedLink in the associated Nortel Preside MDM and within DX NetOps Spectrum.
  Attribute ID: 0x4fd0000
- **EmsLinkDescr**
  The description given to this link inside the Nortel Preside MDM.
  Attribute ID: 0x4fd0004
- **EmsLinkType**
  The type given to this link inside the Nortel Preside MDM.
  Attribute ID: 0x4fd0005
- **EmsPersistence**
  The persistence type of this link, either Static or Dynamic.
  Attribute ID: 0x4fd0008

Accessing Nortel Preside MDM within OneClick

This chapter describes how to use Nortel Preside MDM with OneClick. In particular, it describes how you can access and work with Nortel Preside MDM using the various OneClick components.

Navigation Panel

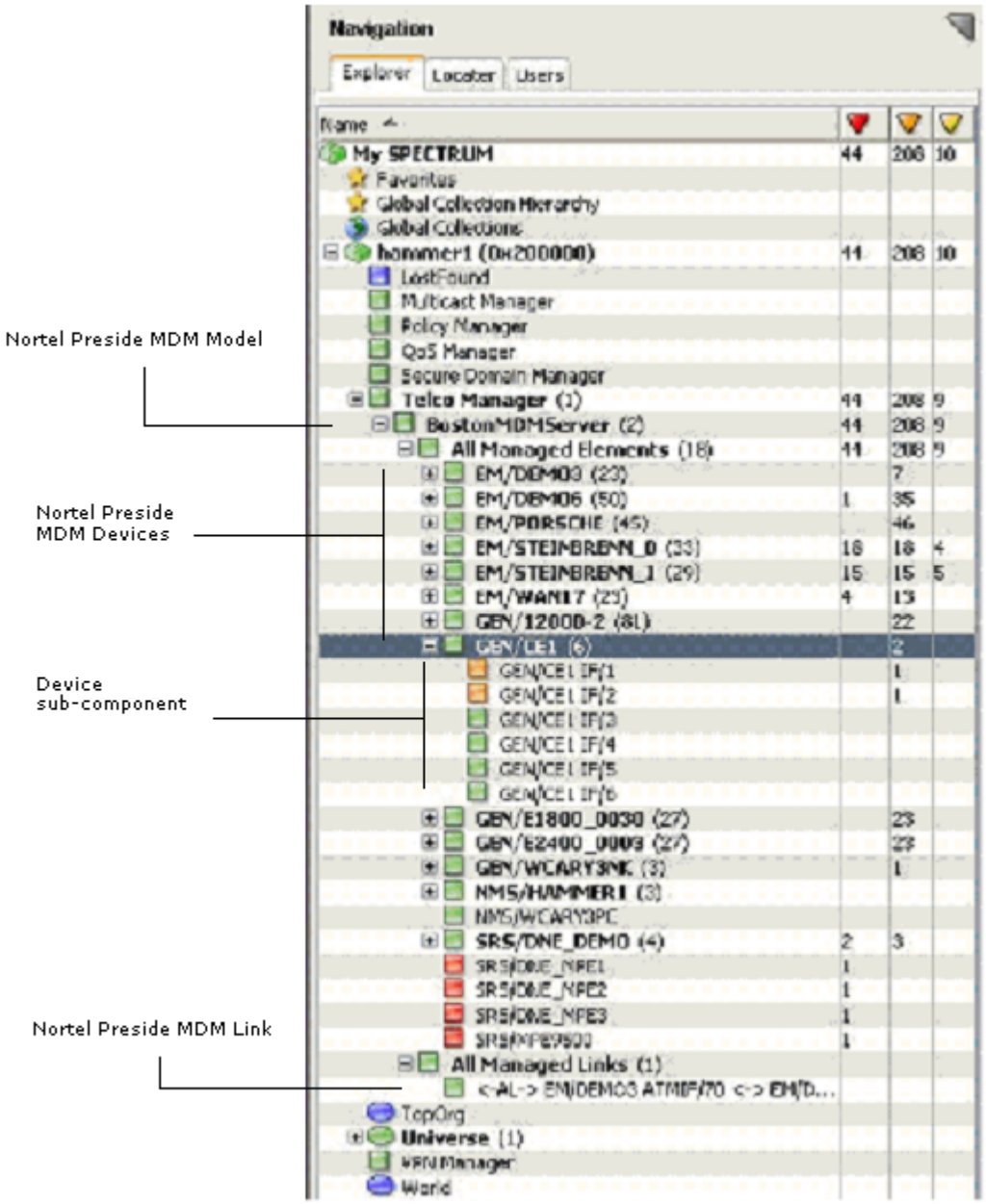
In the Explorer tab of the OneClick Navigation panel, expand the desired landscape to display the Telco EMS Manager application. Every Nortel Preside MDMConnector that connects successfully with DX NetOps Spectrum is represented by its corresponding model. Expand the Telco EMS Manager model to display all Nortel Preside MDM models connected with DX NetOps Spectrum.

All TelcoEMSManagedElement and TelcoEMSManagedLink models managed in DX NetOps Spectrum appear inside the All Managed Elements or All Managed Links model within the Nortel Preside MDM model.

The All Managed Elements model contains all device models connected to a Nortel Preside MDM model. Expand the desired device model to display its sub-components.

The All Managed Links model contains a list of all links managed by the selected Nortel Preside MDM. Expand the desired link to display its sub-links.

The following image displays an expanded view of the Telco EMS Manager and its elements.



Contents Panel

The information displayed in the Contents panel is determined by the model or element selected in the Navigation panel and the tab selected in the Contents panel. The following sections discuss the different tabs in the Contents panel.

- **Alarms tab**
  When you select any Nortel Preside MDM model in the Navigation panel, the Alarms tab displays all alarms on the selected model and its sub-components.
- **List tab**
  The List tab displays all immediate sub-components of the model selected in the Navigation panel. Depending on the type of models displayed in the List tab, different information appears in the columns.
- **Topology tab**

The Topology tab is disabled for all Nortel Preside MDM-related models because they do not have their own topology. Any TelcoEMSManagedElement or TelcoEMSManagedLink model can, however, be copied and pasted into any of the DX NetOps Spectrum network topology views (Universe, World, or TopOrg). This lets you arrange your network containment as desired.

Component Detail Panel

The information displayed in the Component Detail panel depends on which model or element is selected in the Navigation panel and in the Contents panel. When you select a model in the Alarms tab or the List tab of the Contents panel, the Component Detail panel displays information about the selected model.

> **NOTE**
> To view the Component Detail of a container model that has children (such as the Telco EMS Manager model or a Nortel Preside MDM model), you must select the List tab in the Contents panel and press the Control (Ctrl) key to deselect all models listed. This lets OneClick display the container model's information in the Component Detail panel.

Information Tab

The Information tab displays the configuration options for different elements as follows:

- **Telco EMS Manager Model**
  The Information tab for the Telco EMS Manager model contains the following sections:
  - **General Information**
    Displays the basic DX NetOps Spectrum information about the model.
  - **Configuration**
    Starts a new Telco Discovery.
  - **Telco EMS List**
    Displays a list of the Nortel Preside MDMs that are currently connected with this landscape.
- **Nortel Preside MDM Model**
  The Information tab for the Nortel Preside MDM model contains the following sections:
  - **General Information**
    Displays the basic DX NetOps Spectrum information about the model.
  - **Telco Sub-Component List**
    Shows the following Telco model grouping containers:
    - All Managed Elements
    - All Managed Links
- **All Managed Elements/All Managed Links**
  When you select the All Managed Elements or All Managed Links grouping container, the Information tab contains the following sections:
  - **General Information**
    Displays the basic DX NetOps Spectrum information about the model.
  - **Telco Sub-Component List**
    Shows all of the top-level models such as devices or links.
- **TelcoEMSManagedElement**
  The Information tab for the TelcoEMSManagedElement model contains the following sections:
  - **Telco Managed Element Information**
    Displays the EMS-specific information describing the model.
  - **General Information**
    Displays the basic DX NetOps Spectrum information about the model.
  - **Telco Sub-Component List**

Shows all the TelcoEMSManagedElement models that are sub-components of the selected model such as cards, slots, ports, and so on.

- **TelcoEMSManagedLink**
  The Information tab of the TelcoEMSManagedLink model contains the following sections:
  - **Telco Managed Link Information**
    Displays the EMS-specific information describing the model.
  - **General Information**
    Displays the basic DX NetOps Spectrum information about the model.
  - **Telco Sub-Component List**
    Shows all the TelcoEMSManagedLink models that are sub-components of the selected model such as logical links.
  - **Connected Telco Managed Element List**
    Displays the two TelcoEMSManagedElement models connected by the selected link model.

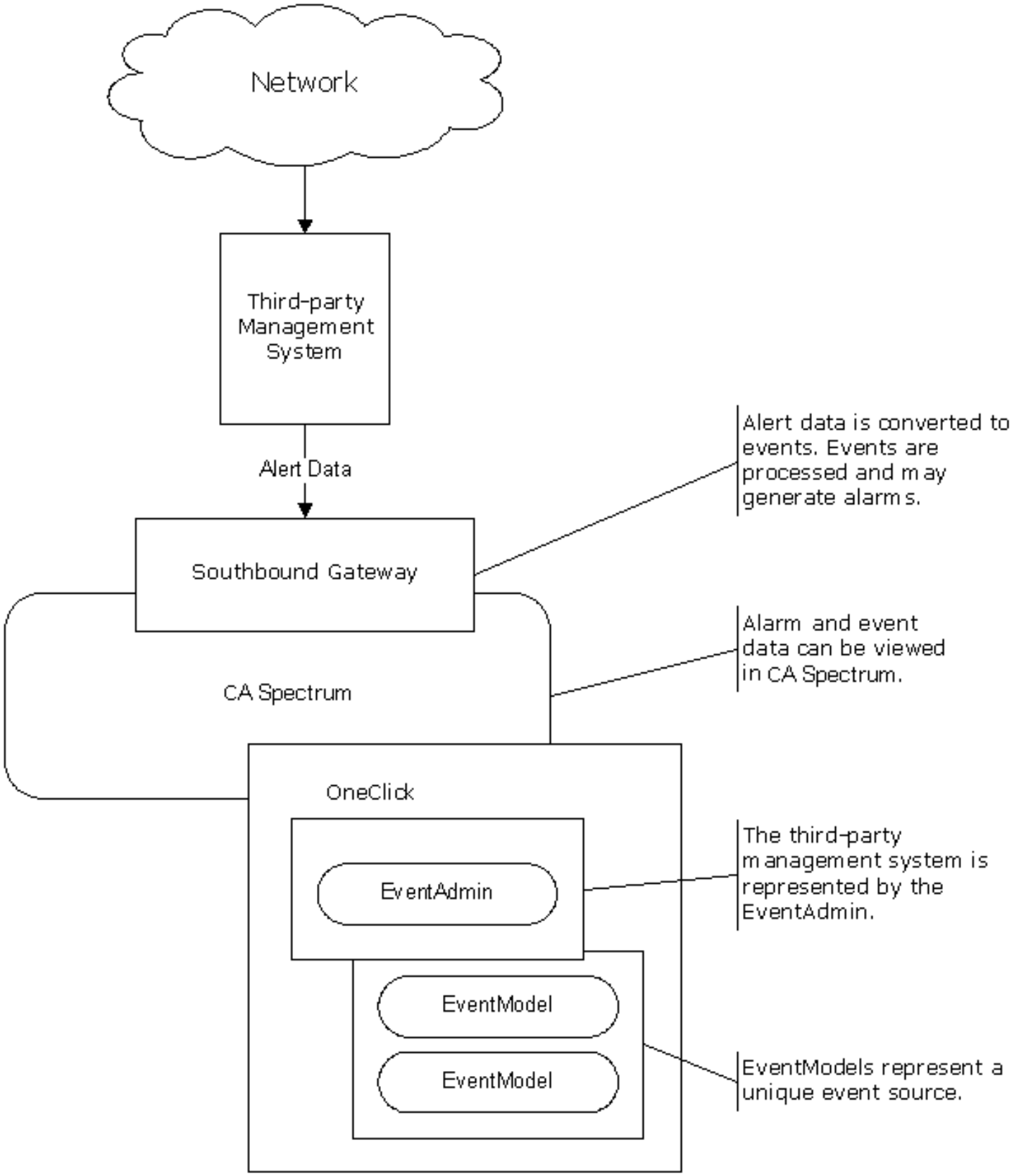
Southbound Gateway Toolkit

Centralize Network Management with Southbound Gateway

The Southbound Gateway integration point accepts alert data from third-party sources. Often these sources are other network management applications that specialize in monitoring a specific aspect of a computing environment. Using Southbound Gateway, you can centralize network management, letting DX NetOps Spectrum capture and display data from other systems. Alert data is organized into DX NetOps Spectrum event and alarm data for display in OneClick.

Southbound Gateway can be used with any incoming alert data stream format. It provides a simple, non-programmatic integration point for systems that generate SNMP traps. It is also very useful for managing non-SNMP environments. The Southbound Gateway import tool accepts XML-formatted alert data for integration with systems that cannot generate SNMP traps. A Document Type Definition (DTD) is provided to specify the XML elements used in this file. Model types that represent the external management system and its components are built into DX NetOps Spectrum, eliminating the need to generate or derive new model types. The EventAdmin model type represents a third-party system that sends alerts to DX NetOps Spectrum. The EventAdmin can contain one or more EventModels. These EventModels represent a unique event source within the network management system. To deploy the Southbound Gateway toolkit, you configure alert, event, and alarm support for these model types. When the integration is complete, you can monitor events and alarms from this third-party system using OneClick.

The following diagram illustrates the components of a Southbound Gateway integration and shows how these components work together to process third-party alert data.


Network
Third-party Management System
Alert Data
Southbound Gateway
CA Spectrum
OneClick
EventAdmin
EventModel
EventModel
Alert data is converted to events. Events are processed and may generate alarms.
Alarm and event data can be viewed in CA Spectrum.
The third-party management system is represented by the EventAdmin.
EventModels represent a unique event source.

Prerequisites for Developers

Before you use Southbound Gateway, verify that you have met the following requirements:

- Significant exposure to DX NetOps Spectrum
- Familiarity with the Getting Started section and with the underlying concepts of DX NetOps Spectrum
- A basic knowledge of MIBs and SNMP
- Detailed knowledge of the system you are integrating
- (For non-SNMP integrations) Working knowledge of XML, C++, or Java
- Ability to use UNIX or Windows operating systems to navigate the file system, copy and delete files, and create and edit text files

Southbound Gateway Architecture

Southbound Gateway Model Types

An EventAdmin model type represents a third-party system sending alerts to DX NetOps Spectrum. Each instantiated EventAdmin model represents an individual instance of a running external management application. Specific events from the third-party system are sent to the EventAdmin model. The EventAdmin model receives these events and transfers the event data to EventModels or device models depending on how the integration has been configured. Alarms can be created from this event data.

The EventModel is a model type that represents a unique source of event data on the system managed by the EventAdmin model application. A given EventAdmin model can contain one or many instantiated EventModels. Each event received through the Southbound Gateway contains information that uniquely identifies the source of that event. The EventAdmin model receives the event, finds the unique event source, and passes the event to the target model that represents the unique source. If the integration has been configured to use a unique identifier to map events exclusively to EventModels and no EventModel exists for the source, an EventModel model is automatically created to represent it. All new EventModel models are placed in the corresponding EventAdmin container model. You can cut or copy EventModel models from an EventAdmin container model and paste them into other types of container models in the Topology, Location, or Organizational view. You can cut or copy the EventAdmin model and paste it into the Topology, Location, or Organizational view. You can also place EventModels into maintenance mode as required.

Southbound Gateway Model Type Support Files

Support files supply information to the EventAdmin and EventModel models. Work with the following support files to create a Southbound Gateway integration:

- **AlertMap file:** If the alert source sends SNMP traps, create a text file to add data to the EventAdmin AlertMap file. DX NetOps Spectrum can then receive and parse the SNMP traps and convert them to DX NetOps Spectrum events.
- **XML file:** If the alert source does not send SNMP traps, create an XML file to send events to DX NetOps Spectrum. The syntax of the XML file must follow the Document Type Definition (DTD) provided with this toolkit. You can find a sample XML file named sbgw_event_sample.xml in the SS-Tools/GSADEMO directory. The Southbound Gateway import tool imports the data from the XML file into DX NetOps Spectrum.
- **EventDisp file:** Create these text files to add data to the EventDisp files that the EventAdmin and EventModel use. These additions define how the events are processed.
- **Event Format file:** Create Event Format files to provide additional, optional, or textual information and formatting for each event.
- **Probable Cause file:** Create Probable Cause files to give textual information about each alarm you create.

These support files organize the data into an event, process the event, and display event information. If the alert is SNMP-based, it is sent to the AlertMap file. If the data is non-SNMP based, it can be formatted into an XML file and processed by the Southbound Gateway import tool.

In both cases, the data is mapped from the alert to the DX NetOps Spectrum event. Variable data from the event is formatted, and a unique event source identifier is defined using the Southbound Gateway Event Data Template. The event is then passed to the EventAdmin model. The EventAdmin model checks its EventDisp file to verify that it is registered to receive this event. If the event is present in the EventDisp file, the EventAdmin examines the event to find the unique ID. The variable information sent with the event is mapped to determine a unique event source. Based on this unique identifier, the EventAdmin forwards the event to the appropriate EventModel. If this model does not yet exist, it is automatically created.

When the EventModel receives the event, it uses its EventDisp file to determine what to do with the event. The event can be logged, mapped to an alarm, used to clear an alarm, or used with other events to create an alarm. The EventModel uses the Event Format and Probable Cause files to display information about the event or alarm in the various Event and Alarm views or applications.

Host Model Types

DX NetOps Spectrum has several host model types that have the same functionality as the EventAdmin model type. These include Host_Compaq, Host_Dell, Host_IBMDirector, Host_Sun, and Host_systemEDGE. You can use any one of these host model types instead of the EventAdmin model type to represent the third-party system.

If you use one of these model types, all the integration instructions in this section are still applicable. However, modify the AlertMap and EventDisp support files for the model type that you have selected. These files are located in the following directory:

```
<$SPECROOT>/SS/CsVendor/Ctron_Gen_HOST/<model_type>
```

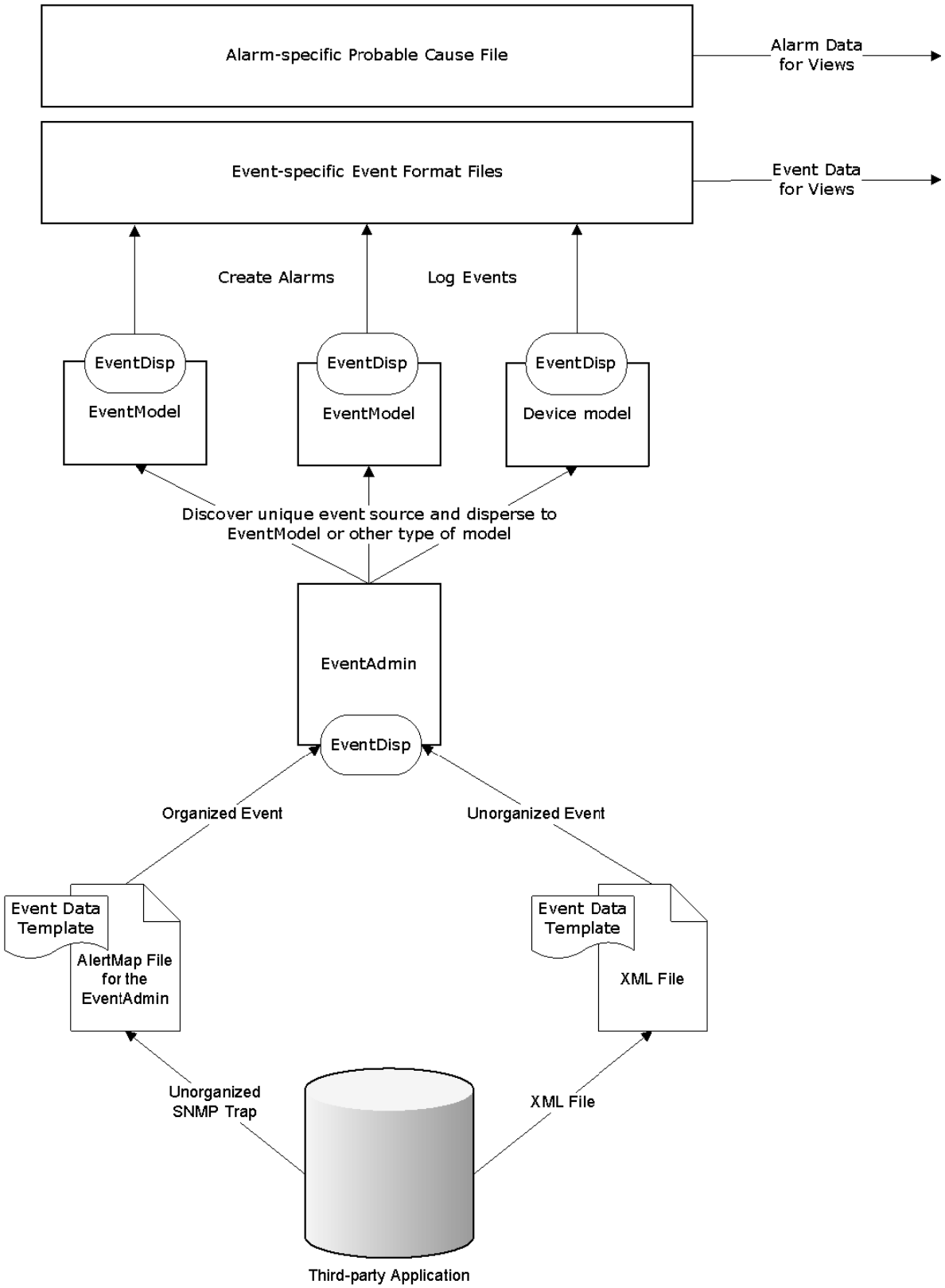
- ***<model_type>***
  Indicates the host model type that you have selected.

Once you have configured the integration, use the setup instructions in the topic titled Use a Host Model Type Instead of the EventAdmin Model Type.

> **NOTE**
> If you use a host model type instead of the EventAdmin model type, you cannot distribute the integration.

The Flow of Data through the Southbound Gateway

The following diagram illustrates how data flows from the third-party system into DX NetOps Spectrum through the Southbound Gateway:


Alarm-specific Probable Cause File
Alarm Data for Views
Event-specific Event Format Files
Event Data for Views
Create Alarms
Log Events
EventDisp
EventModel
EventDisp
EventModel
EventDisp
Device model
Discover unique event source and disperse to EventModel or other type of model
EventAdmin
EventDisp
Organized Event
Unorganized Event
Event Data Template
AlertMap File for the EventAdmin
Event Data Template
XML File
Unorganized SNMP Trap
XML File
Third-party Application

Southbound Gateway Integration

How to Create a Southbound Gateway Integration

Your primary job as the Southbound Gateway integrator is to create data for the support files that provide information to the EventAdmin and EventModel models representing the third-party system.

Integration steps include:

1. Configuring the third-party system to send alert data to DX NetOps Spectrum.
2. Mapping SNMP trap data to a DX NetOps Spectrum event if the third-party system can send SNMP traps.
3. Mapping non-SNMP alert data to a DX NetOps Spectrum event if the third-party system sends alerts that are not SNMP-based.
4. Controlling events and alarm creation by creating data for EventDisp files.
5. Defining the presentation format of events with Event Format files. These files provide event information to DX NetOps Spectrum applications.
6. Adding value to alarms by creating Probable Cause files. These files provide alarm information to DX NetOps Spectrum applications.

Configuring the Third-Party System

The methodology to configure a third-party system to send alert data to the Southbound Gateway depends on the functionality of the individual system.

If the third-party system can send SNMP traps to DX NetOps Spectrum, configure it to send them. The application must send the trap data to the hostname and port where the SpectroSERVER is running and listening. By default, DX NetOps Spectrum listens at the standard SNMP trap port 162. You can modify this default by changing the snmp_trap_port parameter in the DX NetOps Spectrum .vnmrc file located in the SS directory. Once Southbound Gateway receives the alert data, this data must be mapped to a DX NetOps Spectrum event in an AlertMap file.

If the third-party system is sending non-SNMP alerts, format the alert data using XML. Import the data into DX NetOps Spectrum using the sbgwimport tool.

In both cases, Southbound Gateway selects the appropriate EventAdmin model to receive alert data. The IP address of the host computer that sends the data determines the target model. The IP address of the host computer should match the IP address that was used to create the EventAdmin model.

About Integrating Alert Data from a Third-Party System in a Secure Domain

Consider the following before you integrate alert data from a third-party system in a secure domain:

- Set the SecureDomainAddress attribute (0x12d83) of the EventAdmin model to the IP address of the SSDC for the domain that includes the third-party system.
- When an EventModel is created, its secure domain context is set to the secure domain context of the EventAdmin that created it.
- EventAdmin forwards events only to EventModels or device models that have the same secure domain context.
- By default, Southbound Gateway models only forward events to target models when the Southbound Gateway model and the target model have the same secure domain address. Or they forward events if the Southbound Gateway model and the target model are both directly managed (the secure domain address is 0.0.0.0). Set the SBGW_Ignore_Secure_Domain_Address attribute (0x3dc001d) of the Southbound Gateway model to TRUE to forward events to all matching models, regardless of their secure domain addresses.

If you want to forward events to remote models, we recommend using a *single* Southbound Gateway model to forward events to each remote target model. If multiple Southbound Gateway models forward events,

all of the Southbound Gateway models must have the same secure domain IP address. In addition, the SBGW_Ignore_Secure_Domain_Address attribute (0x3dc001d) must be set to the same value. DX NetOps Spectrum does not store all the information about a remote model locally, nor does it search for updated information about that remote model when a new Southbound Gateway model forwards events.

Remote target models are found using the settings on the Southbound Gateway model. Each Southbound Gateway model finds a remote target model from a cache of models, by name or targets IP address. It can find a remote target even if the secure domain address does not match the address of the Southbound Gateway model.

About Host Agents and Southbound Gateway

Southbound Gateway can forward syslog messages that are received as traps from the log monitoring agents. Agents such as CA Unicenter NSM Agent, CA SystemEDGE Agent, or iAgent are supported. The messages are forwarded to the originating device if it can be determined. The forwarding algorithm inspects the trap message for an IP address or hostname to determine the actual source. If a hostname is found, DX NetOps Spectrum attempts to resolve the hostname to an IP address and locate the associated model.

For performance reasons, the Southbound Gateway caches resolved and unresolved hostnames for approximately six hours. Therefore, if you want to use newly created name service entries whose name was cached as an unsuccessful lookup in the Southbound Gateway you must either restart DX NetOps Spectrum or wait for six hours.

Southbound Gateway log matching algorithm is a set of waterfall heuristics that make a best-effort attempt to recognize and parse the inbound trap. While DX NetOps Spectrum supports over 10,000 known Cisco messages, you can set up your own.

Southbound Gateway has debugging capabilities to assist you in resolving ParseMap and translation errors. You can enable this debugging by sending the action 0x3dc0001 to the Agent model. Output appears in the DX NetOps Spectrum Control Panel and is written to the VNM.OUT file.

NOTE

For more information about working with log monitoring agents in DX NetOps Spectrum, see the Host System Resources Management section.

Map SNMP Trap Data to a DX NetOps Spectrum Event

If a third-party system is configured to send SNMP traps to DX NetOps Spectrum, map these traps to DX NetOps Spectrum events in an AlertMap file. To map traps, create data to be added to an existing AlertMap file. This data specifies the trap, the event it is mapped to, and the variable-bindings from the trap. Understand the components of an SNMP trap to map the SNMP trap data in an AlertMap file.

NOTE
DX NetOps Spectrum supports the reception and processing of SNMPv2 format traps and InformRequests as defined by RFC 2576. For more information about how to refer to SNMPv2 traps in the AlertMap file, see the Event Configuration User *section.*

For DX NetOps Spectrum to use trap information from a third-party system, associate the trap with a specific DX NetOps Spectrum event in an AlertMap file. The AlertMap file has three main functions:

- Indicates the SNMP traps that are received from the third-party system.
- Indicates the event code to be generated for a given trap.
- Maps the trap variable bindings to DX NetOps Spectrum event variable IDs.

The AlertMap file that is provided for Southbound Gateway is located in the following directory:

```
$SPECROOT/SS/CsVendor/gen_app_gw/EventAdmin
```

Do not modify the existing AlertMap file directly. The AlertMap file that is located in the specified directory can contain mapping information for pre-existing integrations. Create a text file that contains the alert map data to be added to this AlertMap file. When you distribute the Southbound Gateway integration, include the Southbound Gateway installation script that is provided with the toolkit. This script recreates the AlertMap file, which is based on the text files that are supplied by all Southbound Gateway integrations. For more information, see the Event Configuration User *section.*

Use the following convention to rename the test file that you created for AlertMap data:

```
indexfilename.amp
```

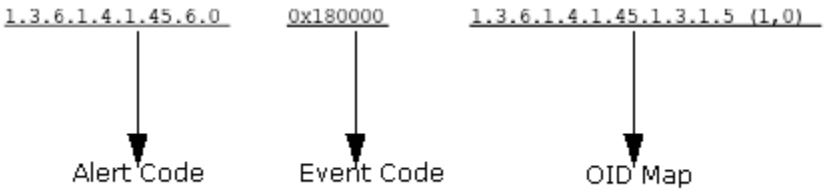
- **indexfilename**
  Specifies the name of the index file that you create when using the DX NetOps Spectrum Extension Integration toolkit to distribute this Southbound Gateway integration.

AlertMap File Syntax

Each trap mapped to a DX NetOps Spectrum event must have an entry in the AlertMap file. Each entry in the AlertMap file has three components:

- The alert code
- The event code
- The OID map

Example AlertMap file entry:



Alert Code

The alert code consists of several pieces of information from the trap: the Enterprise OID string, the generic trap identifier, and the specific trap code.

- **1.3.6.1.4.1.45:** Enterprise OID indicating, in this case, a Synoptics device.
- **6:** The Generic Trap Identifier, in this case the 6 indicates an enterprise-specific trap.
- **0:** Specific Trap Code indicating the specific enterprise trap.

Event Code

> **NOTE**
> To complete this process, a CA Developer ID is required. For further information about obtaining a Developer ID, see the Concepts *section.*

The event code is a hexadecimal number composed of the Developer ID and a number that uniquely identifies this particular event. The Developer ID makes up the first half of the event code. The second half of the event code is a unique number. Generate and manage this unique number to ensure that an event code for your Developer ID is never repeated.

If the event code is 0, this alert is ignored.

The OID Map

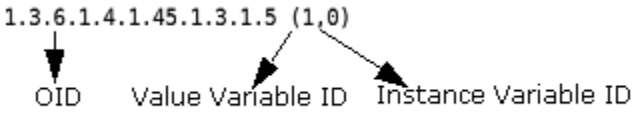
A trap can include one or more pieces of variable information, known as variable bindings. Each variable binding represents a piece of information about the trap. You can map these bindings to enable DX NetOps Spectrum to use this information.

The variable-bindings and their mappings in DX NetOps Spectrum are specified in the OID map section of the alert map entry. A single alert map entry can have multiple associated OID maps, which indicates that multiple variable bindings were sent with the SNMP trap. Separate these OID maps using the '\' character and a new line as shown in the following example:

```
1.3.6.1.4.1.52.6.271 0x1060f 1.3.6.1.4.1.52.1.2.1.9.2.1.2(3,4)\
                             1.3.6.1.4.1.52.1.2.1.1.24(5,6)\
                             1.3.6.1.4.1.52.1.2.2.3.1.1(1,2)
```

The OID map can be broken down into three parts:

- The OID
- The value variable ID
- The instance variable ID



The OID identifies the specific variable that is sent with the trap. For example, the previous OID references the s3ChassisPsStatus variable in the Synoptics Trap MIB.

The value variable ID stores the value of the variable sent in the variable binding. The instance variable ID stores the instance portion of the OID. If your variable binding identifies a particular object from a table variable within the trap MIB, it likely includes an instance ID.

Determine the integer values for the value variable ID and the instance variable ID by referring to the Southbound Gateway Event Data Template. The values that you select depend on the content from the variable binding. The following section outlines the Event Data Template and how to select the appropriate number, depending on the content of the variable binding.

Map Non-SNMP Alert Data to a DX NetOps Spectrum Event

Two approaches are available to map non-SNMP alert data to a DX NetOps Spectrum event. The first method is to use XML to create DX NetOps Spectrum events. We recommend this method, which is efficient and easy to implement.

You can also programmatically send events to DX NetOps Spectrum. This method requires a more in-depth understanding of the DX NetOps Spectrum application programming interfaces.

Use XML to Create DX NetOps Spectrum Events

Southbound Gateway provides the sbgwimport tool to create DX NetOps Spectrum events based on data from an XML file. You mark up the data in the alert using the XML elements provided in the Southbound Gateway DTD file, .sbgwimport.dtd. Once you have created the XML file, run the sbgwimport tool. The tool creates events from the data in the XML file and sends them to the EventAdmin model that represents the third-party system. Southbound Gateway identifies the EventAdmin model that represents the third-party system based on the IP address of the host running the sbgwimport tool. If an EventAdmin model to represent the third-party system does not yet exist, a new EventAdmin model is automatically created.

NOTE
Working knowledge of XML and how a DTD functions is required to use this tool.

Create an XML File to Format the Alert Data

Southbound Gateway provides the .sbgwimport.dtd file located in the SS-Tools directory. The DTD defines the elements that can be used to create the XML input file. Two elements are defined in the DTD: the Import element and the Event element.

The Import element is the root element. XML syntax rules specify that the root element is the outermost element and denotes the beginning and end of the XML file. Therefore, the Import element surrounds the rest of the XML elements in the input file. It has one child element, the Event element. Any number of Event elements can exist within the Import element.

Attributes

The Event element is used to format alert data using a series of attributes. These attributes define the event that is sent to DX NetOps Spectrum. All of these attributes correspond to specific fields in the Event Data Template, except for the eventType and eventAdminName attributes. The attributes that correspond to the Event Data Template takes on the event variable ID specified in the Event Data Template. The event variable ID can then be used in an Event Format file. The following list defines all the available attributes.

- **eventType**
  Event Data Template Variable ID: N/A
  Required attribute. The value for this attribute should be the hexadecimal event code assigned to the event. The event code is a hexadecimal number composed of the Developer ID and a number uniquely identifying this particular event. Your Developer ID makes up the first half of the event code. The second half of the event code is simply a unique number. Generate and manage this unique number to help ensure that an event code for your Developer ID is never repeated.
- **uniqueId1- uniqueId6**
  Event Data Template Variable ID: 1-6
- **targetName**
  Event Data Template Variable ID: 7

  NOTE
  You can use the attribute targetNameIgnoreCase if you do not want DX NetOps Spectrum to consider the case when matching the attribute's value to the model name.
- **targetAddress**
  Event Data Template Variable ID: 8
- **eventModelName**
  Event Data Template Variable ID: 10
- **eventAdminName**
  Event Data Template Variable ID: N/A
  Only used if an EventAdmin model representing the system sending the events does not yet exist. If this EventAdmin model does not exist, DX NetOps Spectrum creates an EventAdmin model and uses the value of this attribute as the model name.
  If this value is not specified when a new EventAdmin model is created, *EventAdmin_<HostName>* is used as the model name, where *<HostName>* is the name of the host computer running the sbgwimport tool.
  If an EventAdmin model exists, DX NetOps Spectrum ignores the value for the eventAdminName attribute. After you run the sbgwimport tool, a message indicates that the value for the eventAdminName attribute was not used because an EventAdmin model exists for this system.
- **networkAddress**
  Event Data Template Variable ID: 13
- **macAddress**
  Event Data Template Variable ID: 14
- **Manufacturer**
  Event Data Template Variable ID: 15

This attribute can be set with the manufacturer of the alert source. If defined, the value of this attribute can be referenced in the Event Format file.

- **targetNameIgnoreCase**
  Event Data Template Variable ID: 16
- **userDefined_101- userDefined_120**
  Event Data Template Variable ID: 101-120
  These attributes can be used to define user-specific data. If defined, each of these values can be referenced in the Event Format file. More user-defined attributes can be added to the DTD if they are necessary for the integration. They must use an integer value > 120.

Sample Alert Data XML File

The following is a sample XML file that creates three events:

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE Import SYSTEM ".sbgwimport.dtd">
<Import>
   <Event eventType = "0x3c80000" uniqueId1 = "xyzLocation"
    uniqueId2 = "abcPortNumber" eventAdminName = "MyTestAdmin"
    modelClass = "Router"
    userDefined_101 = "Test Application Name" />
   <Event eventType = "0x3c80004" uniqueId1 = "xyzLocation"
    uniqueId2 = "defPortNumber"
    eventAdminName = "MyTestAdmin" />
   <Event eventType = "0x3c80004" targetAddress = "10.253.29.1"
    eventAdminName = "MyTestAdmin" />
</Import>
```

The first event has an event code of 0x3c80000 defined by the eventType attribute. The attributes of uniqueId1 and uniqueId2 are used to identify the source of this event. These attributes are concatenated together to create xyzLocation_abcPortNumber as the unique identifier for the alert source. The EventAdmin model to receive the event is named MyTestAdmin. The device that sends the event is identified as a router using the modelClass attribute. The user_Defined_101 attribute is used to send data that is not described by one of the other fields.

The second event has an event code of 0x3c80004. The attributes ‘uniqueId1’ and ‘uniqueId2’ identify the source of this event. These attributes are concatenated to create ‘xyzLocation_defPortNumber_ClientID0x56FF’ as the unique alert source identifier. The EventAdmin model to receive the event is named ‘MyTestAdmin.’

The third event also has an event code of 0x3c80004. The attribute targetAddress directs the event to a pre-existing DX NetOps Spectrum device model with an IP address of 10.253.29.1. The EventAdmin model to receive the event is named ‘MyTestAdmin.’

Run the sbgwimport Tool

Once you have created the XML file, you are ready to run the sbgwimport tool.

> **NOTE**
> You can run the sbgwimport tool from the third-party host because Southbound Gateway identifies the EventAdmin model that represents the third-party system based on the IP address of the host computer that runs the import tool. To run the import tool from the third-party host, move the tool and all of its support files to that host. For instructions, see the Distributed SpectroSERVER Administrator section.

To run the import tool, use the following syntax:

```
sbgwimport  - vnm v_name inputfile
```

- **v_name**

Specifies the name of the DX NetOps Spectrum server.

- **inputfile**
  Specifies the name of the XML file.

  NOTE
  Under some circumstances, such as on Linux, the sbgwimport tool returns the localhost address (127.0.0.1) instead of the actual IP address. In this event, specify the IP address that the import tool uses to identify the desired EventAdmin model.

To run the sbgwimport tool using a specific IP address, use the following syntax:

```
sbgwimport -vnm v_name input_file -localIP IP address
```

- **IP address**
  Specifies the IP address.

Using CORBA to Create Events

You can use the CORBA API to create and send events into DX NetOps Spectrum. Use a very small portion of the functionality of this API to connect to the appropriate EventAdmin model, format the event data from the third-party system, and then send this data in an event to the EventAdmin model. The sample code to send events to an EventAdmin model is available in the SS-Tools/GSADEMO directory. Open the file named SBGWImport.java.

You can use the command-line interface (CLI) to generate these events; however, for performance reasons, we recommend that you use the CORBA API. This toolkit only provides examples using the SSORB interface.

The Event Data Template

The Southbound Gateway Event Data Template defines alert variable data. This template is used with both SNMP and non-SNMP integrations. For SNMP integrations, the Event Data Template variable IDs are used in the AlertMap file to select the appropriate number for the value variable and instance variable IDs.

Selecting the correct Event Data Template variable ID values is important because they enable essential functions of the Southbound Gateway integration. The values that you select determine how a target model for the incoming event is identified. You can control whether a new EventModel is created to handle an incoming event. You can also select variable ID values, which specify whether to use a network address or model name as the event target search criterion. You can then map events to any type of model, including devices or ports. Other values let the variable data serve as attribute values for the EventModel where the event is sent. All variables can be referenced in the Event Format files. A zero value for either the value variable or instance variable IDs indicates that the value is not stored.

The Event Data Template shows the name, type, and description of each variable ID that the Southbound Gateway mechanism supports. When selecting values for the value variable ID for an integration, choose appropriate variable IDs that describe the content of the value of the variable binding. Where applicable, the instance variable ID is stored as one of the unique identifier fields.

Specify one of the following variable ID values to identify a target model:

- One of the Unique ID variables (1-6)
- Target Name (7) or Target Name Case Insensitive (16)
- Target Address (8)

Event Data Template Fields

The following table describes the fields in the Event Data Template. Each event must contain the required vardata items to be processed by the Southbound Gateway.

NOTE
If the event received does not conform to the Event Data Template, an event with an ID 0x03dc0000 is generated against the EventAdmin model.

Variable ID	Name	Type	Description	Required/ Optional
1	Unique ID 1	String/Integer	The Unique ID variables are used to identify a target EventModel by a Unique Identifier string. The Unique Identifier is a composite of up to six variable data items (one to six). The final unique identifier string is composed as follows: <1>_<2>_<3>_<4>_ <5>_<6> in that exact order. If one of the components is not provided, it is not included within the composite unique identifier. If an existing EventModel with the Unique Identifier string cannot be found one is created. Either one unique identifier or a Target Name (field 7) or Target Address (field 8) is required.	
2	Unique ID 2	String/Integer	The Unique ID variables are used to identify a target EventModel by a Unique Identifier string. The Unique Identifier is a composite of up to six variable data items (one to six). The final unique identifier string is composed as follows: <1>_<2>_<3>_<4>_ <5>_<6> in that exact order. If one of the components is not provided, it is not included within the composite unique identifier. If an existing EventModel with the Unique Identifier string cannot be found one is created. Either one unique identifier or a Target Name (field 7) or Target Address (field 8) is required.	
3	Unique ID 3	String/Integer	The Unique ID variables are used to identify a target EventModel by a Unique Identifier string. The Unique Identifier is a composite of up to six variable data items (one to six). The final unique identifier string is composed as follows: <1>_<2>_<3>_<4>_ <5>_<6> in that exact order. If one of the components is not provided, it is not included within the composite unique identifier. If an existing EventModel with the Unique Identifier string cannot be found one is created. Either one unique identifier or a Target Name (field 7) or Target Address (field 8) is required.	
4	Unique ID 4	String/Integer	The Unique ID variables are used to identify a target EventModel by a Unique Identifier string. The Unique Identifier is a composite of up to six variable data items (one to six). The final unique identifier string is composed as follows: <1>_<2>_<3>_<4>_ <5>_<6> in that exact order. If one of the components is not provided, it is not included within the composite unique identifier. If an existing EventModel with the Unique Identifier string cannot be found one is created. Either one unique identifier or a Target Name (field 7) or Target Address (field 8) is required.	

5	Unique ID 5	String/Integer	The Unique ID variables are used to identify a target EventModel by a Unique Identifier string. The Unique Identifier is a composite of up to six variable data items (one to six). The final unique identifier string is composed as follows: `<1>_<2>_<3>_<4>_ <5>_<6>` in that exact order. If one of the components is not provided, it is not included within the composite unique identifier. If an existing EventModel with the Unique Identifier string cannot be found one is created. Either one unique identifier or a Target Name (field 7) or Target Address (field 8) is required.	
6	Unique ID 6	String/Integer	The Unique ID variables are used to identify a target EventModel by a Unique Identifier string. The Unique Identifier is a composite of up to six variable data items (one to six). The final unique identifier string is composed as follows: `<1>_<2>_<3>_<4>_ <5>_<6>` in that exact order. If one of the components is not provided, it is not included within the composite unique identifier. If an existing EventModel with the Unique Identifier string cannot be found one is created. Either one unique identifier or a Target Name (field 7) or Target Address (field 8) is required.	NA
7	Target Name	String	This field lets you specify a target model by model name. An EventModel is not created if a target model is not found using a Target Name event variable. However, if event variable 7 (Target Name) or event variable 16 (Target Name Case Insensitive) is used with the 'unique id' event variable (1-6), the unique id field is used to identify the model.	
8	Target Address	Octet/Text String	This field lets you specify a target model by IP address. Use this field if you want to send the event to a model other than an EventModel.	
9	Reserved for future CA use			
10	EventModel Name	String/Integer	This field provides the ability to assign a model name that is different than the unique identifier. If this data is not provided, the composite unique identifier becomes the model name.	Optional
11	Model Class	Integer	Populates the Model Class attribute of the target EventModel with the value specified in this event variable.	Optional
12	Reserved for future CA use			
13	Network Address	Octet/Text String	Populates the Network Address attribute of the target EventModel with the value specified in this event variable.	Optional
14	MAC Address	Octet/Text String	Populates the MAC Address attribute of the target EventModel with the value specified in this event variable.	Optional
15	Manufacturer	String	Populates the Manufacturer attribute of the target EventModel with the value specified in this event variable.	Optional
16	Target Name Case Insensitive	String	This field lets you specify a target model by model name. An EventModel is not created if a target model is not found using a Target Name Case Insensitive event variable. However, if event variable 7 (Target Name) or event variable 16 (Target Name Case Insensitive) is used with the 'unique id' event variable (1-6), the unique id field is used to identify the model.	
17-99	Reserved for future CA use			

Any other variable ID greater than or equal to 100.	Any Data	Any Type	This data is forwarded to the EventModel model unchanged. The data type and data are preserved. This data can be viewed within an event message.	Optional

Variable ID Fields 1-6

The EventModel unique identifier is composed of the first six fields within the Event Data Template. Southbound Gateway integrations often require more than one piece of data to identify the source of an alert. The Event Data Template provides a way to create an EventModel unique identifier that is composed of the elements from up to six different fields. The EventModel unique identifier is composed as follows:

```
<1>_<2>_<3>_<4>_<5>_<6>
```

For example, a location and port number can be used to represent a router as the EventModel unique identifier. The router's location would be put in Event Data Template field 1 and the port number would be put in Event Data Template field 2. The EventModel unique identifier would then appear as follows: Location_PortNumber.

Variable ID Field 7 (Target Name)

This field lets you specify the model name of the model you would like the event data to be sent to. For example, if you specify the model name *MyRouter* in the Target Name field, Southbound Gateway searches the existing models to find a model with this model name. If this model is found, Southbound Gateway sends the event data to this model for processing.

> **WARNING**
> If no model is found, an EventModel is not created.

This field has the same functionality as Variable ID Field 16 (Target Name Case Insensitive), except that it is case sensitive. If this field had the value MyRouter, and DX NetOps Spectrum had two models, one named MyRouter and one named myrouter, MyRouter would receive the event and myrouter would not. If instead you used the Target Name defined in field 16, both models (MyRouter and myrouter) would receive the event. If for some reason you used both of these fields (7 and 16) and each had the value MyRouter, the model with the model name MyRouter would receive two events (sent as a result of field 7 and 16) and the model with the model name myrouter would receive one event (sent as a result of field 16).

If you specify both a unique ID (using a combination of fields 1 through 6) and a Target Name (using fields 7 or 16), Southbound Gateway sends the event to both the EventModel specified by the unique ID and the model specified by the Target Name.

If you specify a unique ID, a Target Name using fields 7 or 16, and a Target Address, and if the Target Name and Target Address are for different models, the Southbound Gateway sends the event to all three models.

If the Target Name is not found, the EventAdmin model generates an Event that indicates that DX NetOps Spectrum has no models with the name <Target Name>. If a unique ID has also been specified, the event is sent to an EventModel.

If you specify a Target Name using fields 7 or 16 and a Target Address and one or both is not found, the appropriate error message is generated, and the event is not sent to any model. If a unique ID has also been specified, the event is sent to an EventModel.

Variable ID Field 8 (Target Address)

This field lets you specify the IP address of the model you would like the event data to be sent to. Use this field if you want to send data to a model other than the EventModel. For example, if you specify the IP address 10.253.97.2 in the Target

Address field, Southbound Gateway searches the existing models to find a model with this IP address. If Southbound Gateway finds this model, it sends the event data to this model for processing.

If you specify both a unique ID, using a combination of fields 1 through 6, and a Target Address, Southbound Gateway sends the event to both the EventModel specified by the unique ID and the model specified by the Target Address.

If you specify a unique ID, a Target Name, and a Target Address, and the Target Name and Target Address are for different models, the Southbound Gateway sends the event to all three models.

If the Target Address is not found, the EventAdmin model generates an event that indicates that DX NetOps Spectrum has no models with the network address <Target Address>. If a unique ID has also been specified, the event is sent to an EventModel.

If you specify a Target Name and a Target Address and one or both is not found, the appropriate error message is generated, and the event is not sent to any model. If a unique ID has also been specified, the event is sent to an EventModel.

Variable ID Field 10 (EventModel Name)

This is an optional name that can be given to the EventModel model. If supplied, it appears beneath the EventModel icon in the OneClick topology views. If you define an EventModel prefix when creating the EventAdmin model, the EventModel name is concatenated onto the EventModel prefix, separated by an underscore. If you do not define the EventModel prefix and the EventModel name is defined, the EventModel name becomes the model name. If the EventModel Name is not provided within Field 10, the unique identifier is used as the EventModel model name.

Variable ID Field 11 (Model Class)

If provided, this variable indicates the value with which to populate the model class attribute on the EventModel. The model class defines the type of device the model represents. The entry in Field 11 should be an integer that identifies the model class. A list of model classes and their respective integer identifiers can be found in the Getting Started section.

Variable ID Field 13 (Network Address)

If provided, this variable indicates the value with which to populate the network address attribute on the EventModel.

Variable ID Field 14 (MAC Address)

If provided, this variable indicates the value with which to populate the MAC address attribute on the EventModel.

Variable ID Field 15 (Manufacturer)

If provided, this variable indicates the value with which to populate the manufacturer attribute on the EventModel.

Variable ID Field 16 (Target Name Case Insensitive)

This field lets you specify the model name of the model where the event data is sent. This field has the same functionality as Variable ID Field 7 (Target Name), except that this field is case-insensitive.

Other Variable IDs (Any Additional Data)

Any other integration-specific data can be placed in any DX NetOps Spectrum variable ID greater than or equal to 100. This data is not set as attributes on the EventModel model, but is viewable from within the Event Format file.

The following list shows additional data that can be captured for display in an event message:

- application name: 100
- server name: 101
- event status: 102
- ticket status: 103
- ticket type: 104
- comments: 105

Control Events and Alarm Creation

EventDisp files define how events are processed. When alert information has been converted to an event, DX NetOps Spectrum identifies the model that the event has occurred in and then locates the EventDisp file that is associated with that model's model type. DX NetOps Spectrum searches the file for the appropriate event code and carries out the action specified for that event code.

Because there are two different model types that play a part in handling events for Southbound Gateway, create data for two different EventDisp files. There is one EventDisp file for the EventAdmin model type that defines how the EventAdmin models process the event and there is another EventDisp file for the EventModel model type that defines how the EventModel models process the event.

The EventAdmin EventDisp file is located in the following DX NetOps Spectrum directory:

```
<$SPECROOT>/SS/CSVendor/gen_app_gw/EventAdmin
```

The EventModel EventDisp file is located in the following DX NetOps Spectrum directory:

```
<$SPECROOT>SS/CSVendor/gen_app_gw
```

Do not modify the existing EventDisp files directly. The EventDisp files currently located in these directories can contain information for pre-existing integrations. Instead, create text files containing the EventDisp data that should be added to each of the files. When you distribute the Southbound Gateway integration, you include the Southbound Gateway installation script provided with the toolkit, which recreates each of the EventDisp files based on the text files supplied by all installed Southbound Gateway integrations.

The text files that you create for your EventDisp data must each use the following naming convention. This gives you two files with the same name. We recommended that you manage these files by keeping them in separate directories while you are working with them.

```
<indexfilename>.evd
```

- ***<indexfilename>***
  Specifies the name of the index file that you create when using the DX NetOps Spectrum Extension Integration toolkit to distribute this Southbound Gateway integration.

EventAdmin EventDisp File

The purpose of the EventAdmin EventDisp file is to define what events are received from the external alert/trap source. To do this, the EventDisp file must list the event codes of the events that are received. These event codes are defined in the AlertMap file, or in the XML import file.

NOTE
This EventDisp file must list those events that are forwarded to other DX NetOps Spectrum models through the use of Target Name or Target Address, and events that are forwarded to EventModel models.

For example, an EventDisp file expecting to receive three events with the codes 0x03dc003, 0x03dc004, and 0x03dc005 would have the following content:

```
0x03dc003
0x03dc004
0x03dc005
```

You can define more options for the event within this EventDisp file but it is not recommended. The intention of this file is to indicate what events the EventAdmin model should process. Once the EventAdmin receives the event, it forwards the event to the proper EventModel. It is at the EventModel level where further definition of the event processing takes place.

NOTE
This file is not optional. If your integration sends an event code that is not referenced in the EventAdmin EventDisp file, DX NetOps Spectrum simply drops that event, which prevents it from being forwarded to the appropriate EventModel model.

EventModel EventDisp File

Although the EventDisp for the EventModel is not required, it is highly recommended. In the EventAdmin EventDisp file, all the events that the EventAdmin could receive are listed. At the EventModel level you define how these events are processed. In addition to receiving events, EventDisp files can define a number of event behaviors and characteristics including:

- If the event should be logged
- The severity of the event
- If the event should clear an alarm
- If the event should generate an alarm
- If the alarm generated can be cleared by a user
- The severity of any alarm to be generated
- The probable cause of any alarm to be generated
- The event or series of events that should generate another event

Using the EventCondition Event Rule, you can create an event disposition entry that allows events to be created conditionally based on the value of variable bindings from a trap or based on the value of DX NetOps Spectrum attributes. This can be useful if the third-party application sending traps to DX NetOps Spectrum through the Southbound Gateway sends only one trap OID. This trap may have different meanings based on the value of one or more variable bindings. Using the EventCondition rule, you can generate different events depending on the value of these variable bindings. See the Event Configuration for complete information about the EventConditon Event Rule.

In general, if you are forwarding events to a model other than an EventModel model, you define the instructions for event processing in the EventModel's EventDisp file. This EventDisp file processes events for all appropriate model types. However, if you are using the EventCondition Event Rule as outlined in the previous paragraph, you can create this entry in the model type EventDisp file for the model to which you are forwarding events. This forces the condition parsing work to be done by that model type rather than by the EventModel.

Event Configuration provides reference information about EventDisp files. See the syntax instructions in this section when creating your EventDisp syntax.

Presentation Format of Events

Event format files determine the contents of an event message when the event is displayed in OneClick. Each event has a separate event format file. Event messages appear in the Events tab and the Alarm Details tab in OneClick.

Each event that requires a corresponding event message must have an associated event format file. Keep in mind that most of the information a user receives about an event or alarm is through the message text that is affiliated with that alarm or event. The index file that you create for distribution unpacks the event format files into the correct location (SG-Support/CsEvFormat) on the user's SpectroSERVER.

Unlike the AlertMap and EventDisp files, the data in the text files is not added to other files. Therefore, it is important to use the naming convention outlined in this section to identify your Event Format files.

The event format file must be named as Eventxxxxxxxx where xxxxxxxx equals the event code given to the event in the AlertMap file and listed in the EventDisp file. The event code must be written in a hexadecimal format with a full eight digits, including leading zeros.

The event format file can contain plain text, but can also contain variables that reference specifics about the instance of the individual event. For information about possible variable references, see the Event Configuration section.

The following is an example of the event format file:

```
{d "%w- %d %m-, %Y - %T"} - ASX DS3 LOS DETECTED - This trap indicates that the specified DS3 port has
 detected incoming LOS (Loss of Signal) Alarm. - Model Name: {m};
PortName: {S 1},
PortBoard: {I 2},
PortModule: {I 3},
PortNumber: {I 4}.
(event [{e}])
```

In this example, {d "%w- %d %m-, %Y - %T"} places the date and time into the event message, m inserts the model name, and e inserts the event code. The other variables reference values from variable bindings and their corresponding data types, S for string and I for integer.

Use the variable IDs that you have used from the Event Data Template to construct any type of message; however, verify that the variables in the message are the correct data type.

Add Value to Alarms

Alarms that are generated as a result of the EventDisp file usually contain a probable cause as part of the alarm message. The text of this probable cause is defined in the Probable Cause file associated with the alarm. Create a separate Probable Cause file for each of the alarms that you generate. Each Probable Cause file contains a textual explanation of the cause of the alarm with optional advice about the likely cause and common troubleshooting steps. The index file that you create for installation unpacks the Probable Cause files into the correct location (SG-Support/CsPCause) on the user's SpectroSERVER. Unlike the AlertMap and EventDisp files, the data in the text files are not added to other files. Therefore, it is important to use the following naming convention to identify your Probable Cause files.

Each Probable Cause file must be named Probxxxxxxxx where xxxxxxxx is the event code that generated the alarm through the EventDisp file. Write the event code in hexadecimal format with a full eight digits, including leading zeros.

The Probable Cause file contains text only. The first line states the cause of the alarm in uppercase letters. Following the cause are three separate sections, each with a title in uppercase letters: SYMPTOMS, PROBABLE CAUSES, and RECOMMENDED ACTIONS. Under each of these titles, list the appropriate data.

If there are multiple symptoms, probable causes, or recommended actions, format the data under those categories as a numbered list.

The following is an example of a Probable Cause file showing the appropriate syntax:

```
ALL DEVICE CONNECTIONS ARE UNREACHABLE
SYMPTOMS:
The VNM is unable to contact this device because all of the devices that this one is connected to are
 unreachable.
PROBABLE CAUSES:
1) A router connected to this device has gone down.
2) A network cable has become unplugged.
RECOMMENDED ACTIONS:
1) Check the status of the devices that this one is connected to for problems.
2) Check for a loose or unplugged network cable.
```

Distributing a Southbound Gateway Integration

After creating the proper files to support your Southbound Gateway integration, the next step is to package the files for distribution to other DX NetOps Spectrum users.

To do this, you use some of the tools in the DX NetOps Spectrum Extension Integration toolkit. This toolkit provides the ability to create a package of files called a VCD (Virtual CD-ROM) that the DX NetOps Spectrum installer opens and expands. This package of files can include a number of different components; at a minimum, yours includes the Southbound Gateway installation script, an index file, a Part Description file, and the support files you have created.

The Installation Script

The Southbound Gateway installation script is a pre-built script included with the Southbound Gateway toolkit. It is named MergeVendorFiles.cus and is located in the SS-Tools/GSADEMO directory. This script is responsible for taking your EventDisp and AlertMap data contained in the text files you created in the integration process, and recreating the existing AlertMap and EventDisp files to include that data. Add the following line to all Southbound Gateway index files to verify that the installation script is included with your integration.

```
file: SS cus /<$SPECROOT>/SS-Tools/GSADEMO/MergeVendorFiles.cus ? ?
```

The Part Description File

The part description file is a text file that contains a brief synopsis of the integration being installed. The user can view the text in this file during the install process. Pay close attention to the directory structure outlined in the Sample Index File. For the integration to work properly, the support files must be correctly placed on the computer.

Create an Index File

The index file maps all the files included in the VCD indicates the files needed for installation, where these files are located on your computer, and where the files should be unpacked on the customer's computer. For a description of the possible syntax for an index file, see the Extension Integration (SEI) Developer Reference section. The syntax outlined in Sample Index File describes a typical Southbound Gateway integration installation. In your index file, include references to the following files:

- Southbound Gateway Installation Script
- AlertMap file (if applicable)
- Part Description file
- EventDisp files
- Event Format files
- Probable Cause files

Create your index file using your preferred text editor and then save the index file in the SG-Tools directory. The naming convention for the index file is as follows:

```
3P-<ABCxxxx>.i
```

- **3P**
  Specifies a third-party integration.
- ***<ABC>***
  Specifies the first three letters of your developer name.
- ***<xxxx>***
  Specifies any four-digit integer that uniquely identifies this index file from any other index file you have created.

Sample Index File

The following sample index file shows the components of an integration that includes the Southbound Gateway installation script, a Part Description file, an Event Format file, a Probable Cause file, two EventDisp files, and an AlertMap file. At install time, all files are unpacked into their proper directory and the installer program runs the Southbound Gateway installation script.

```
level: 1
mm:     CA Spectrum Southbound Gateway Integration Demonstration
irev:   06.50.00.000
rev: 6.5
pprep:  Sample
vend:   gen_app_gw
file: SS cus /SpectrumRootDirectoryName/SS-Tools/GSADEMO/MergeVendorFiles.cus ? ?
file:   Install mmdesc ? Demo.mmd ? ?
file:   SG other ? ../../SG-Support/CsEvFormat/Event03e50000 ? ?
file:   SG other ? ../../SG-Support/CsPCause/Prob03e50000 ? ?
file:         SS vfile ? ../SS/CsVendor/gen_app_gw/components/3P-ABC0001.evd SS/CsVendor/gen_app_gw/components
 3P-ABC0001.evd
file:         SS vfile ? ../SS/CsVendor/gen_app_gw/EventAdmin/components/3P- ABC0001.evd SS/CsVendor/
gen_app_gw/EventAdmin/components 3P-ABC0001.evd
file:   SS vfile ? ../../SS/CsVendor/gen_app_gw/EventAdmin/AM_components/3P-ABC0001.amp ? 3P-ABC0001.amp
```

The index file can be divided into the following two portions:

- The first portion is the extension module description entries. These entries include specific information about the integration, such as name and release level. For more information, see the Extension Integration (SEI) Developer Reference section.
- The second portion of the index file contains the file distribution entries. These entries specify the files that can be included with the installation. The first line in this section has the following syntax:

  ```
  file: SS cus /SpectrumRootDirectoryName/SS-Tools/GSADEMO/MergeVendorFiles.cus ? ?
  ```

Include this line in all the index files that are created to distribute a Southbound Gateway integration. This line is responsible for activating the Southbound Gateway installation script. The rest of the index file references the Event Format file, the Probable Cause file, the two EventDisp files, and the AlertMap file for this particular installation. For more information, see Extension Integration (SEI) Developer Reference section.

mkmm and mkcd

To complete the package, run the mkmm and the mkcd tool. See the Extension Integration (SEI) Developer Reference section for more information.

Using a Southbound Gateway Integration

DX NetOps Spectrum provides different model types that represent components of a third-party system, the EventAdmin, and the EventModel. These model types work together to manage alert information sent to DX NetOps Spectrum. A unique EventAdmin exists for each instance of a third-party system managed by DX NetOps Spectrum. The EventAdmin is a container model type and includes models of the type EventModel. EventModels represent unique sources of event information that a third-party system manages. The EventAdmin receives an event from the third-party system and dispatches it to an EventModel.

When you represent this system within DX NetOps Spectrum, you are representing the management application, not the host computer that the application is running on. To manage this host computer, create a separate model of the appropriate model type.

For both SNMP and non-SNMP integrations, the EventAdmin's network address is stored in the attribute named trapIPAddress, instead of in the usual NetworkAddress attribute. This prevents the EventAdmin from interfering with another DX NetOps Spectrum model that is managing the device containing that network address, that is, the model that is managing the host computer.

Create an EventAdmin Model

Before attempting to represent your third-party system within DX NetOps Spectrum, verify that the management application is running. Once the application is running properly, the first step is to model the application in DX NetOps Spectrum using the EventAdmin model type. In most situations, you add the EventAdmin icon into the same topology view as the VNM icon.

You can define several attributes when you create this model. Most of these attributes carry over to become attributes of EventModel models created for this EventAdmin. There are several applications within DX NetOps Spectrum that display the EventModel models' attributes, events, and alarms. To make this information as useful as possible, it is a good idea to provide as much information as you can about those EventAdmin and EventModel models.

To create the EventAdmin model

1. From the Topology tab that you want to create the model in, click the Create new model by type button.
   The Select Model Type dialog appears.
2. In the Containers tab, click EventAdmin.
3. Click OK.
   The Create Model of Type EventAdmin dialog appears.
4. Configure the following parameters:
   - **Name**
     Optional. Defines the EventAdmin's model name. This model name appears in the field at the top of the EventAdmin icon.
   - **Network Address**
     Required for all integrations that are based on SNMP traps. Specifies the network address of the event source's host computer. If you are unsure if your third-party application sends SNMP traps, consult with the documentation for that product.
   - **Security String**
     (Optional) Defines who can view and edit this model.
   - **Manager Name**
     When this attribute is set on the EventAdmin, all EventModels contained within this EventAdmin also have this attribute. If the name of the third-party application does not appear on this list, select Default from the drop-down list.
   - **EventModel Prefix**

This field is prepended to the EventModel Name for all the EventModels contained by this EventAdmin. This helps ensure that there are consistent naming prefixes for all EventModels associated with a particular EventAdmin. It is also useful for sorting and filtering.

5. After you enter the appropriate information into these fields, click OK.
   This generates an EventAdmin model. A default EventModel is also created and is contained in the EventAdmin model. This model is used for fault tolerance functionality.

   NOTE
   You can configure the EventAdmin model to forward events to models on remote landscapes by setting the SBG_AlertForwardingEnabled (0x3dc000c) attribute for the EventAdmin model to TRUE.

EventAdmin Model Status

Once you have instantiated an EventAdmin model, its status becomes green (Normal) even if there are not yet any EventModel models associated with it. This status changes to reflect alarms asserted on the EventAdmin model.

NOTE
DX NetOps Spectrum does not monitor the status of the application that the EventAdmin represents, so the status of the model does not represent the state of the third-party application, nor does it represent the state of the third-party application's host computer.

Move an EventAdmin Model

An EventAdmin model can be cut or copied and then pasted into other topologies as needed.

The following is a list of containers that an EventAdmin can be placed in:

- Universe:
  - Broadband_Net
  - LAN
  - Network
  - Universe
  - WAN
  - GnChasCont
  - LAN_802_3
  - LAN_802_5
  - FDDI
- World:
  - Panel
  - Rack
  - Room
  - Site
- TopOrg:
  - Org_Owns
  - Service_Owns

You move an EventAdmin model in the same way you would move any other model. However, it is not possible for an EventAdmin model to contain another container model. The EventAdmin model can only contain EventModel models.

To move an EventAdmin model

1. Open the topology view that contains the EventAdmin model that you want to move.
2. Right-click the model that you would like to move and select Copy or Cut.

3. Go to topology view where you want to place the EventAdmin model, right-click and select Paste.
   You have now successfully moved the model.

   NOTE
   The EventAdmin model represents the third-party management system and is required for proper operation of the Southbound Gateway integration. If you delete the EventAdmin model to remove support for the third-party system, delete all EventModel models that are associated with the EventAdmin.

View EventAdmin Model Information

You can review details about the EventAdmin model from the tabs in the Contents panel and the Component Detail panel.

You can use the Information tab to review and edit the model name, network address, manager name, and the EventModel prefix of the EventAdmin model.

From the Topology tab, you can navigate to all the EventModels contained by the EventAdmin.The EventAdmin model manages the third-party application and the alerts it sends to DX NetOps Spectrum, the EventAdmin does not manage the host computer that is running the third-party application.

Use a Host Model Type Instead of the EventAdmin Model Type

You can use a DX NetOps Spectrum host model type instead of the EventAdmin model type to represent the third-party system. After making the necessary support file modifications outlined in the previous chapters, you can configure the host model.

To configure the host model

1. Click the create a new model button in the view you want to add the host model to.
   The Select Model Type dialog appears.
2. Choose the appropriate host model type:
   - Host_Dell
   - Host_Compaq
   - Host_IBMDirector
   - Host_Sun
   - Host_systemEDGE

   and click OK.
   The Create Model of Type dialog appears.
3. Fill in the fields appropriately, using the IP address of the third-party server for the Network Address field and then click OK.
   DX NetOps Spectrum creates the model.
4. For EventModels to be created, set the Enable_SouthboundGateway (0x116296e) attribute on the host model to TRUE. Do this using the Attribute Editor.

   NOTE
   For more information about using the Attribute Editor, see the Modeling and Managing Your IT Infrastructure section.
5. Specify where the EventModels created by the host model are to be placed when they are created. They can be placed in any container model available in the Topology view with the exception of any SW_Link, WA_Link, and VNM container models. To specify the appropriate container model, set the host model's EventModelContainerHandle attribute (0x3dc000a) equal to the model handle of the container where you would like the EventModels to be created. Do this using the OneClick Attribute Editor.
   If you would like the EventModels to be placed directly in the Universe view, use the model handle of the Universe view. You can retrieve the model handle of the Universe view by using the Attributes tab in the OneClick Component Detail panel tab to read the model_handle attribute on the Universe model.

The host model is now configured to receive and process traps.

Create an EventModel Model

The EventAdmin that manages the EventModels is created automatically. When an alert is sent from the third-party application into DX NetOps Spectrum, the alert is converted into an event. Each event has a unique ID string. This unique ID string is set up during the integration development process to determine a unique source of alert information within the third-party system. The EventAdmin examines this unique ID and looks for an EventModel with a matching unique ID. If this EventModel exists, the event is sent to that EventModel. If an EventModel with this unique ID string does not exist, the EventModel is automatically created.

NOTE
A default EventModel model is created when the EventAdmin model is created. This is for use in a fault-tolerant environment. EventModel models can also be created using the Modeling Gateway. For more information, see the Modeling Gateway Toolkit section.

EventModel Models

EventModels for a particular EventAdmin are automatically placed in a topology view that can be accessed by drilling down from the EventAdmin. These icons do not show any connectivity with one another because they represent an event source, not necessarily a physical device, or component.

About Moving an EventModel Model

EventModel models can be cut or copied from the EventAdmin container model and pasted into various types of container models within the Topology, Location, or Organizational views.

The following is a list of containers that an EventModel can be placed in:

- Topology view:
  - Broadband_Net
  - LAN
  - Network
  - Universe
  - WAN
  - GnChasCont
  - LAN_802_3
  - LAN_802_5
  - FDDI
- Location view:
  - Panel
  - Rack
  - Room
  - Site
- Organization view:
  - Org_Owns
  - Service_Owns

You move an EventModel model in the same way you would move any other model.

An EventAdmin model cannot contain another container model. The EventAdmin model can only contain EventModel models.

Even if you have moved all the EventModel models out of the EventAdmin model, it is important that you do not delete the EventAdmin model. This model represents the third-party management system and is required for proper operation of the Southbound Gateway integration. Any new EventModel models that are created appear in this EventAdmin model by default.

If you delete the EventAdmin model to remove support for the third-party system, delete all EventModel models associated with the EventAdmin also.

Attributes

The EventModel icon displays the model name and model type. The incoming alert information can assign the model name. If the model name is not assigned, the unique ID string that identifies the EventModel is visible within the Model Name field.

If you defined an EventModel prefix when creating the EventAdmin, this prefix prepends the EventModel name for all the EventModels contained by this EventAdmin. This provides a way to further uniquely identify the EventModel for a particular integration.

If you have chosen a specific Manager Name when creating the EventAdmin, the information is also available to the EventModel.

All these parameters can be changed from the Attributes tab in the Component Detail view of the EventAdmin. If such changes are made, existing EventModels are updated with the new attribute values.

Put an EventModel Model into Maintenance Mode

You can put EventModel models into maintenance mode from the Information tab in the Component Detail view. Putting EventModel models into maintenance mode suspends management traffic to the EventModel model and its components and prevents the generation of any events or alarms on its behalf. When an EventModel is put into maintenance mode, its icon displays a device condition color of brown.

To put an EventModel model into maintenance mode

1. Select the EventModel model you want to put into maintenance mode, click the Information tab in the Component Detail view, and then expand the General Information node.
2. Click set in the In Maintenance field, and then select Yes from the drop-down list.
   The EventModel model is now in maintenance mode; its icon color changes to brown.

Set Up a Fault-Tolerant Environment

Fault tolerance is provided by having more than one SpectroSERVER available for managing a given landscape. If the primary SpectroSERVER fails, the secondary SpectroSERVER takes over.

To set up a fault-tolerant environment with Southbound Gateway

1. Set up the fault-tolerant environment as described in the Distributed SpectroSERVER Administration section.
2. Start the primary SpectroSERVER and then start the management application being integrated through the Southbound Gateway.
   The primary SpectroSERVER creates an EventAdmin model automatically if you have used the sbgwimport tool to create your integration, or if you have used the DX NetOps Spectrum CORBA API and specified programmatically that the EventAdmin be created automatically. Otherwise, the EventAdmin model is not created automatically and creates it on the primary SpectroSERVER.
   A default EventModel model is created automatically once the EventAdmin model is activated.

3. Synchronize the primary and secondary SpectroSERVER databases using the DX NetOps Spectrum Online Backup feature.

   NOTE
   See the Distributed SpectroSERVER Administration section and the Database Management section for instructions.

If the primary SpectroSERVER goes down and the secondary SpectroSERVER takes over, the default EventModel model is used to process incoming events.

When the secondary SpectroSERVER is monitoring the network and events are received by the EventAdmin model, new EventModel models are not created. First, the EventAdmin checks to see if there is an appropriate existing EventModel to send the events to, if not, instead of creating an EventModel, the events are sent to the default EventModel model.

Allowing the secondary SpectroSERVER to create new EventModel models would cause disparate model handle information between the primary and secondary SpectroSERVERs. This would cause the Archive Manager to store the event and statistical information for a single event source in different places, as if the information is actually from different event sources. The default EventModel prevents this from happening.

NOTE
If either the EventAdmin or default EventModel is destroyed on the primary, secondary, or tertiary server, the previous steps must be repeated. Steps 1 to 3 must be repeated if the models are destroyed on the primary SpectroSERVER. Steps 2 to 3 must be repeated if the models are destroyed on the secondary or tertiary SpectroSERVER.

Archive Manager Failure

If events are being sent to the EventAdmin model through the sbgwimport tool, additional fault tolerance has been implemented in case of an Archive Manager failure. If the Archive Manager goes down, alarms triggered by incoming events are created on the default event model. When the Archive Manager is running again, the events are sent to it and can be viewed in the event log.

Southbound Gateway Case Study

This chapter walks you through an integration with typical third-party management software that can send SNMP traps to DX NetOps Spectrum.

The traps sent by this software have been directed to the SpectroSERVER and are received by Southbound Gateway. For the sake of simplicity, this chapter assumes that this application only sends one type of trap to the Southbound Gateway. The trap is an enterprise-specific trap with a number of variable bindings.

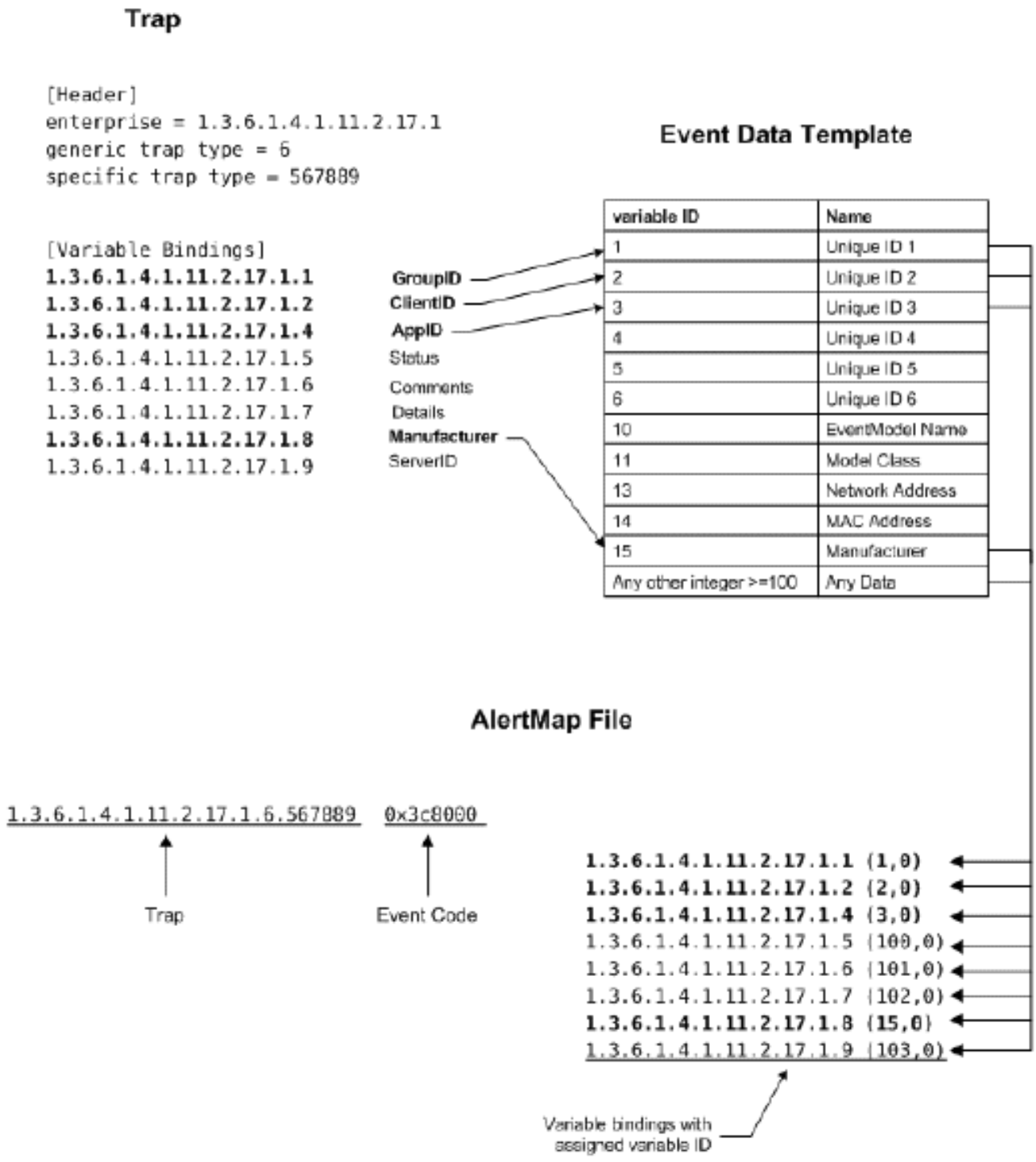
```
[Header]
enterprise = 1.3.6.1.4.1.11.2.17.1
generic trap type = 6
specific trap type = 567889
[Variable Bindings]
1.3.6.1.4.1.11.2.17.1.1
1.3.6.1.4.1.11.2.17.1.2
1.3.6.1.4.1.11.2.17.1.4
1.3.6.1.4.1.11.2.17.1.5
1.3.6.1.4.1.11.2.17.1.6
1.3.6.1.4.1.11.2.17.1.7
1.3.6.1.4.1.11.2.17.1.8
1.3.6.1.4.1.11.2.17.1.9
```

Step 1. The AlertMap File

The trap information must be mapped to an event; therefore, create an alert map entry for this trap. The most important thing to specify is the string that makes up the unique identifier for the EventModel. This unique ID can be comprised of up to six variable bindings using the variable IDs one through six. This data can then be viewed within the event message.

The following example shows the trap being mapped to an event in the AlertMap file. The variable-bindings are being assigned a variable ID from the Event Data Template. All non-bold variable bindings have been mapped to the “Any Data” category of the Event Data Template.

These variable bindings have been given a variable ID that has not been assigned to a specific category in the Event Data Template.


Trap
[Header]
enterprise = 1.3.6.1.4.1.11.2.17.1
generic trap type = 6
specific trap type = 567889
[Variable Bindings]
1.3.6.1.4.1.11.2.17.1.1
1.3.6.1.4.1.11.2.17.1.2
1.3.6.1.4.1.11.2.17.1.4
1.3.6.1.4.1.11.2.17.1.5
1.3.6.1.4.1.11.2.17.1.6
1.3.6.1.4.1.11.2.17.1.7
1.3.6.1.4.1.11.2.17.1.8
1.3.6.1.4.1.11.2.17.1.9
GroupID
ClientID
AppID
Status
Comments
Details
Manufacturer
ServerID
Event Data Template
variable ID
Name
1
Unique ID 1
2
Unique ID 2
3
Unique ID 3
4
Unique ID 4
5
Unique ID 5
6
Unique ID 6
10
EventModel Name
11
Model Class
13
Network Address
14
MAC Address
15
Manufacturer
Any other integer >=100
Any Data
AlertMap File
1.3.6.1.4.1.11.2.17.1.6.567889
0x3c8000
Trap
Event Code
1.3.6.1.4.1.11.2.17.1.1 (1,0)
1.3.6.1.4.1.11.2.17.1.2 (2,0)
1.3.6.1.4.1.11.2.17.1.4 (3,0)
1.3.6.1.4.1.11.2.17.1.5 (100,0)
1.3.6.1.4.1.11.2.17.1.6 (101,0)
1.3.6.1.4.1.11.2.17.1.7 (102,0)
1.3.6.1.4.1.11.2.17.1.8 (15,0)
1.3.6.1.4.1.11.2.17.1.9 (103,0)
Variable bindings with
assigned variable ID

Step 2. The EventAdmin EventDisp File

The following is a sample of the EventAdmin EventDisp file. This EventDisp file lists all the events generated by the third-party system. Listing them here lets them be received by the EventAdmin model.

Example: The EventAdmin EventDisp File

```
0x3c8000
```

The events should be logged at the EventModel level rather than the EventAdmin level. For this reason, the only thing listed in this file is the event code. This syntax indicates that the event should be received, but not logged. After the EventAdmin model receives the event, the EventAdmin model forwards it to the appropriate EventModel model.

Step 3. The EventModel EventDisp File

This EventDisp file gives the event received at the EventModel its properties. This file is optional, but highly recommended. These events are the same as those in the EventAdmin EventDisp file.

Example: The EventModel EventDisp File

```
0x3c8000 E 50 A 1,0x3c8000
```

Clearly this file has more data than the EventAdmin EventDisp file. The E indicates that the event is logged. The event severity is indicated by the number 50; event severity ranges from 1 to 100, 100 being the most severe. The A indicates that the event generates an alarm. The number 1 indicates the severity of the alarm. Valid values for the alarm severity are 1 through 6. The hexadecimal number 0x3c8000 is the alarm code.

There is also an optional ‘S’ flag available for use with EventDisp files which specifies whether an event should be registered for Southbound Gateway processing. It applies to modeltype-specific entries only and only those model types derived from the southbound modeltype fragment. For more information about this flag and the rest of the EventDisp file syntax, see the Event Configuration section.

Step 4. The Event Format File

The Event Format files define the event message seen in both the Events tab and the Alarm Details tab in OneClick.

Example: Event Format File

```
{d "%w- %d%m-, %Y - %T"} - Device {m} of
type {t}
The user's SMTP/POP3 mail transaction
failed with error code 554. The event
code is {e}.
Other information:
Group {S 1}
Client {S 2}
Application {S 3}
Manufacturer: {S 15}
Server: {S 103}
Status: {S 100}
Comments: {S 101}
Details: {S 102}
```

The items in parenthesis indicate that variable value is placed in the event text message. The S indicates the data type (String) and the numeric value is the variable ID assigned in the AlertMap file representing the value that is placed there.

Step 5. The Probable Cause File

The Probable Cause files define the alarm title, symptoms, probable causes, and recommended actions.

Example: Probable Cause File

```
MAIL TRANSACTION ERROR RECEIVED
SYMPTOMS:
The client's attempt to send mail failed.
PROBABLE CAUSES:
1. Network connectivity.
2. The mail server.
RECOMMENDED ACTIONS:
1. Check the network connectivity.
2. Check the mail server operation.
```

Step 6. Package for Distribution

Once you have created all the files necessary for the integration, you need to package these files so that they can be installed on a SpectroSERVER. See the Extension Integration (SEI) Developer Reference section for specific instructions and examples.

Creating a Southbound Gateway Demonstration

Introduction to Creating a Southbound Gateway Integration

This chapter helps you demonstrate the functionality of a Southbound Gateway integration without completing all the steps to create a fully functional integration.

In this demonstration, you are sending a fictitious trap to the EventAdmin model using a trap-generating software. The model receives the trap, translates the trap into a event, creates an EventModel based on the event information and forwards the event to an EventModel model. The event generates an alarm on this EventModel model and the text of the event format and probable cause files is shown in the Alarm tab and the Alarm Details tab in OneClick.

> **WARNING**
> Any changes made directly to AlertMap or EventDisp files are overwritten when a new version of is installed or when an additional Southbound Gateway integration is installed. For this reason, use the steps in this chapter for demonstration purposes only and not as a replacement for the complete integration steps outlined in the chapters of this section.

Step 1. Edit the EventAdmin AlertMap File

To allow the EventAdmin to receive an SNMP trap from a third-party system, the trap must be translated into DX NetOps Spectrum event in the EventAdmin AlertMap file.

To edit the EventAdmin AlertMap file

1. Open the AlertMap file located in the following directory:
   `<$SPECROOT>/custom/Events/gen_app_gw/EventAdmin`
2. Add the following line to the AlertMap file:
   `1.3.6.1.4.1.1850.6.1 0x5990001 1.3.6.1.4.1.1850.1.0.0.1(1,0)`

This line indicates that the trap, 1.3.6.1.4.1.1850.6.1, is translated into event 0x5990001. The value variable ID 1 (indicated in parentheses) shows that the Event Data Template variable of 1 is being used to represent the value of the variable binding, 1.3.6.1.4.1.1850.1.0.0.1. Because the Event Data Template variable 1 is one of the unique identifier values, the value of the variable binding is used as a unique identifier for the event model.

3. Save and exit the AlertMap file.

Step 2. Edit the EventAdmin EventDisp File

For the event created in the AlertMap file to be passed along to the EventModel for processing, add the event code to the EventAdmin EventDisp file.

To edit the EventAdmin EventDisp file

1. Open the EventAdmin EventDisp file located in the following directory:
   ```
   $SPECROOT/custom/Events/gen_app_gw/EventAdmin
   ```
2. Add the following line to this EventDisp file:
   ```
   0x5990001
   ```
3. Save and exit the EventDisp file.

Step 3. Edit the EventModel EventDisp File

For the EventModel model to process the event, create processing instructions in the EventModel model EventDisp file.

To edit the EventModel EventDisp file

1. Open the EventDisp file located in the following directory:
   ```
   <$SPECROOT>/custom/Events/gen_app_gw/EventModel
   ```
2. Add the following line to this file:
   ```
   0x5990001 E 50 A 1, 0x5990001
   ```
   This line indicates that the event should create a minor alarm on the EventAdmin model.

Step 4. Create an Event Format File

The event format file lets users get information about the event in the Events tab in the Component Detail view in OneClick.

To create the event format file for this event

1. Create a text file with the following content:
   ```
   {d "%w- %d %m-, %Y - %T"} - SAMPLE - This is a sample event format file for the Southbound Gateway
    demonstration.
   The event processed is event {e}.
   The variable binding data sent with the trap is: {S 1}.
   ```
2. Save this text file as *Event05990001* in the following directory:
   ```
   <$SPECROOT>/custom/Events/CsEvFormat
   ```
3. Verify that your text editor does not add a file extension to this text file and then close the text file.

Step 5. Create a Probable Cause File

The text of the probable cause file is shown in the Alarm Details tab of the Component Detail view and provides information about the causes and resolutions for the alarm.

To create the probable cause file for this alarm

1. Create a text file with the following content:

```
SAMPLE PROBABLE CAUSE FILE FOR THE SOUTHBOUND GATEWAY DEMONSTRATION
SYMPTOMS:
The symptoms of the problem are listed here
PROBABLE CAUSES:
1) Cause 1.
2) Cause 2.
RECOMMENDED ACTIONS:
1) Action 1.
2) Action 2.
```

2. Save this text file as *Prob05990001_en_US* in the following directory:

```
<$SPECROOT>/custom/Events/CsPCause
```

3. Verify that your text editor does not add a file extension to this text file and then close the text file.

Step 6. Create an EventAdmin Model

Complete this procedure to create an EventAdmin model in the Universe topology for the purposes of this demonstration.

Follow these steps:

1. Click **Create new model by type** in the Topology tab.
   The Select Model Type dialog appears.
2. In the Containers tab, click **EventAdmin**, and then click **OK**.
   The Create Model of Type EventAdmin dialog appears.
3. Configure the following parameters:
   a. Enter Test in the Name field.
   b. Enter the network address of the computer from which you launch the trap in the Network Address field.
      If you are using a trap generation program for the purposes of this demonstration, enter the network address of the computer running the trap generation program.
   c. Leave the Security String field blank.
   d. Leave the Manager Name field as Default.
   e. Leave the EventModel Prefix field blank.
4. Click OK.

The EventAdmin model is now created and appears in the Universe topology.

Step 7. Send the Trap

After you have finished Steps 1 through 6, shut down and restart the SpectroSERVER to activate the changes you have made.

After you have activated your changes, send a trap to the Southbound Gateway to show how this trap is handled. DX NetOps Spectrum accepts an individual SNMP trap packet to a maximum size of 65467 bytes.

You can use a trap generating software such as Trap Generator from Network Computing Technologies to send the trap to the SpectroSERVER. The following procedure uses Trap Generator to illustrate how to send a trap to Southbound Gateway.

> **NOTE**
> The Trap Generator shareware is being used in this example to demonstrate a process, but is not directly supported by CA. For information about downloading, licensing, and using Trap Generator, see the Network Computing Technologies web site: http://www.ncomtech.com/trapgen.html.

To send a trap to Southbound Gateway using Trap Generator

1. If you are using Trap Generator, use the following syntax in your parameter text file:

```
-d <IP Address of SpectroSERVER>:162
-c public
-o 1.3.6.1.4.1.1850
-g 6
-s 1
-v 1.3.6.1.4.1.1850.1.0.0.1 STRING "TestEventModel"
```

2. After you have created this text file, launch the trap by typing:

```
trapgen -f <name of parameter text file>
```

   This syntax directs the trap to port 162 on the SpectroSERVER and defines a variable binding, data type, and value.

Step 8. Show the Results in OneClick

The following process shows the results in OneClick.

The Alert is Received and Translated into a DX NetOps Spectrum Event

When you launch the trap from the trap generation program, the EventAdmin model receives and processes the event.

The Event is Processed by the EventAdmin

When the EventAdmin model processes the event, it uses the variable binding used in Step 1. Edit the EventAdmin AlertMap File as the unique identifier to specify which EventModel to forward the event to. Since this EventModel does not yet exist, a new one is created. To view the EventModel, from the Universe Topology, drill down into the EventAdmin container.

An Alarm is Asserted on the EventModel

In Step 3. Edit the EventModel EventDisp File, an entry was added to the EventModel's EventDisp file to indicate how the event should be processed. This entry indicated that the event should create a minor alarm.

The EventModel model has a status of yellow indicating a minor alarm.

View Data About the Alarm in OneClick

You can view data about the alarm from the Alarm Details tab in OneClick.

To view information about this alarm

1. Select the Event Model in the Explorer tab in the Navigation panel.
2. Click the Events tab in the Contents panel to view the contents of the event format file created in Step 4. Create an Event Format File.
3. Click the Alarms tab in the Contents panel, and then click the Alarm Details tab in the Component Detail view to review the contents of the probable cause file created in Step 5. Create a Probable Cause File.

How to Integrate with DX NetOps Virtual Network Assurance (DX NetOps VNA)

DX NetOps Spectrum and DX NetOps VNA integration lets you access SDN inventory via a single point of integration. The integration uses the DX NetOps VNA Client API (uses WebSocket) to connect to DX NetOps VNA and UDM (Unified Data Model) to access the data.

From 10.4.1, all the SNMP models discovered through DX NetOps VNA are retained in DX NetOps Spectrum in their respective containers after DX NetOps VNA integration is disabled.

WARNING

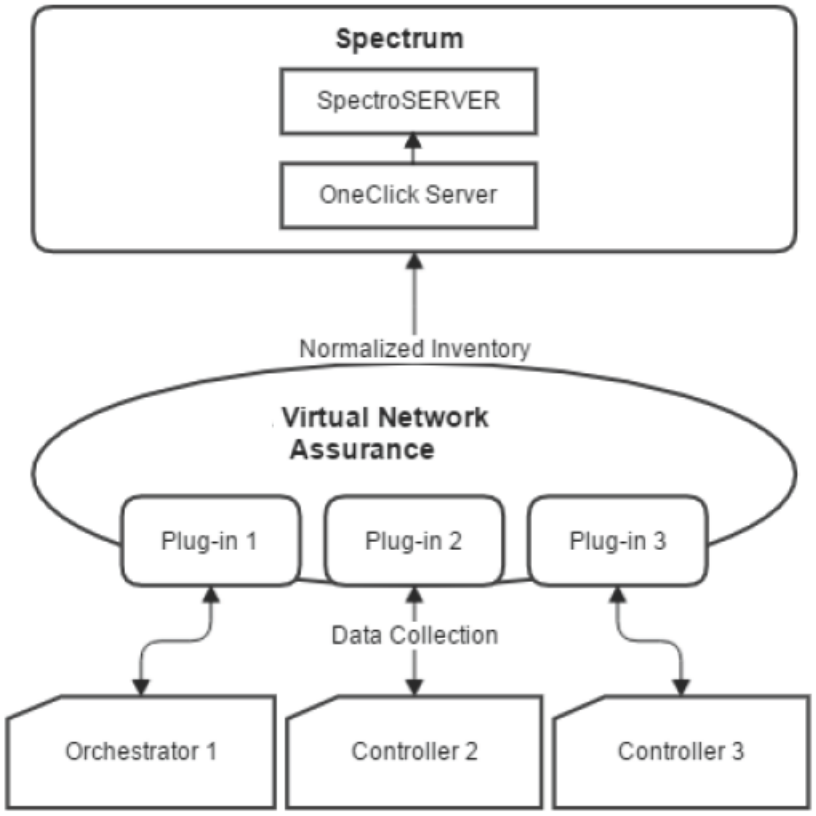
SDN Integration is not supported on Internet Explorer (IE).

The SDN Gateway data source contributes the following item types to DX NetOps Spectrum:

- Topology
- SFC View
- Inventory updates (includes SDN entities and their status information)

The following diagram illustrates DX NetOps Spectrum and DX NetOps Virtual Network Assurance Integration Architecture:

Figure 81: DX NetOps Spectrum and DX NetOps Virtual Network Assurance



Supported DX NetOps Spectrum and DX NetOps VNA version integrations:

For more information about the integration of DX NetOps Spectrum with other CA products, see Integration Compatibility.

Integrating with DX NetOps Virtual Network Assurance

NOTE

From 10.2.3 onwards, SDN Gateway Integration Configuration is now called **DX NetOps VNA Configuration** on the Administration page of the OneClick UI.

NOTE

10.2.3 introduces **Multi-tenancy Support with DX NetOps VNA**. For more information see, DX NetOps VNA Aggregator Integration section**.**

WARNING

SDN Integration is not supported on Internet Explorer (IE).

Integrating DX NetOps Spectrum with DX NetOps VNA allows you to view both virtual network data and traditional physical infrastructure data. The DX NetOps VNA collects data from all your SDN environments and delivers that information to DX NetOps Spectrum. For more information about SDN, see the DX NetOps Virtual Network Assurance documentation.

From 10.4.2.1, you can search the DX VNA models from the **Locator** search. Here is a non-exhaustive list of device models that are searchable:

- All VNA entity models Sdnmanager
  SNMP, slapath, tunnels, interfaces, devices anything, non-SNMP, anything discovered through DX VNA
- Non-SNMP entity models
  VMS, non-SNMP model-able routers/switches not interfaces, nothing granular/ep/epgs
- SNMP models
  SNMP routers, VMS, mdl_create_method : check on this to createmodelbyip, autodisc, createmodelbytype(nonsnmp for vna)
- Entity models per user domain
  Everything under the user domain. SDN_contains_entities relation

Integrating DX NetOps Spectrum with DX NetOps VNA

DX NetOps VNA Standalone and Aggregator Integration

Integration Compatibility Matrix

To know more about version compatibility of other integrated products with DX NetOps Spectrum 10.4.1, see the Integration Compatibility chart.

NOTE

For DX NetOps Spectrum-DX NetOps VNA Integration issues, see the Troubleshooting section.

WARNING

DX NetOps VNA Configuration is not supported on Internet Explorer (IE).

DX NetOps VNA Standalone Integration

To receive the DX NetOps VNA data, you must enable the integration between DX NetOps Spectrum and DX NetOps VNA, which acts as SDN Gateway. You can enable the integration through the OneClick Administration page. Specify the DX Netops VNA configuration information such as the Server Host Name, Server Port, and select a SpectroSERVER to model the inventory from DX NetOps VNA after integration.

Follow these steps:

1. Open the OneClick Administration page.
2. Click the Administration tab.
   Links to various OneClick web server configuration pages appear.

3. Click the DX NetOps VNA Configuration link in the left panel.
   The DX Netops VNA Configuration window appears with the following configuration options:
   - **VNA Server Host Name**
     Specify the IP address/hostname of the DX NetOps VNA Server.
   - **VNA Server Port**
     Specify the server port number of DX NetOps VNA.
   - **Select a SpectroSERVER**
     Specify the SpectroSERVER to be modeled.
4. For **DX NetOps VNA Configuration**s, elect the **Enable** radio button.
5. Click **Test** to verify if the connection is working.
   If the test is successful, ‘Test connection to DX Netops VNA Server was successful’ message appears.
6. Click **Save** and 'Successfully saved configuration to the database' message appears. DX NetOps VNA Configuration is now enabled.

   NOTE
   You cannot change any settings on any other OneClick server other than the integration OneClick server.
7. After saving the configurations, the **Discover** button appears, click on the **Discover** button to allow DX NetOps Spectrum to discover and models devices coming from DX Netops VNA.

   NOTE
   SDN Manager uses community strings from "VNM -> AutoDiscovery Control -> Modeling and Protocol options > SNMP Community Strings" to discover SNMP devices.

   When the DX NetOps Spectrum and DX NetOps VNA Configuration is enabled from the OneClick Administration page, DX NetOps Spectrum receives the data from DX NetOps VNA through the Rest API. Once the data is retrieved from the Rest API, DX Netops VNA hosts are modeled in DX NetOps Spectrum.

   NOTE
   To fetch and update the latest details of the userDomain from DX Netops VNA, **Sync** to remodel devices on the OneClick console by selecting SDN MAnager→ ForceGatewayfullSync→and setting 0x673002c to True. By performing this action a web socket communication towards DX Netops VNA would be restarted.

   NOTE
   Before upgrading to 10.2.3, disable the existing integration between DX NetOps VNA and CA Spectrum.

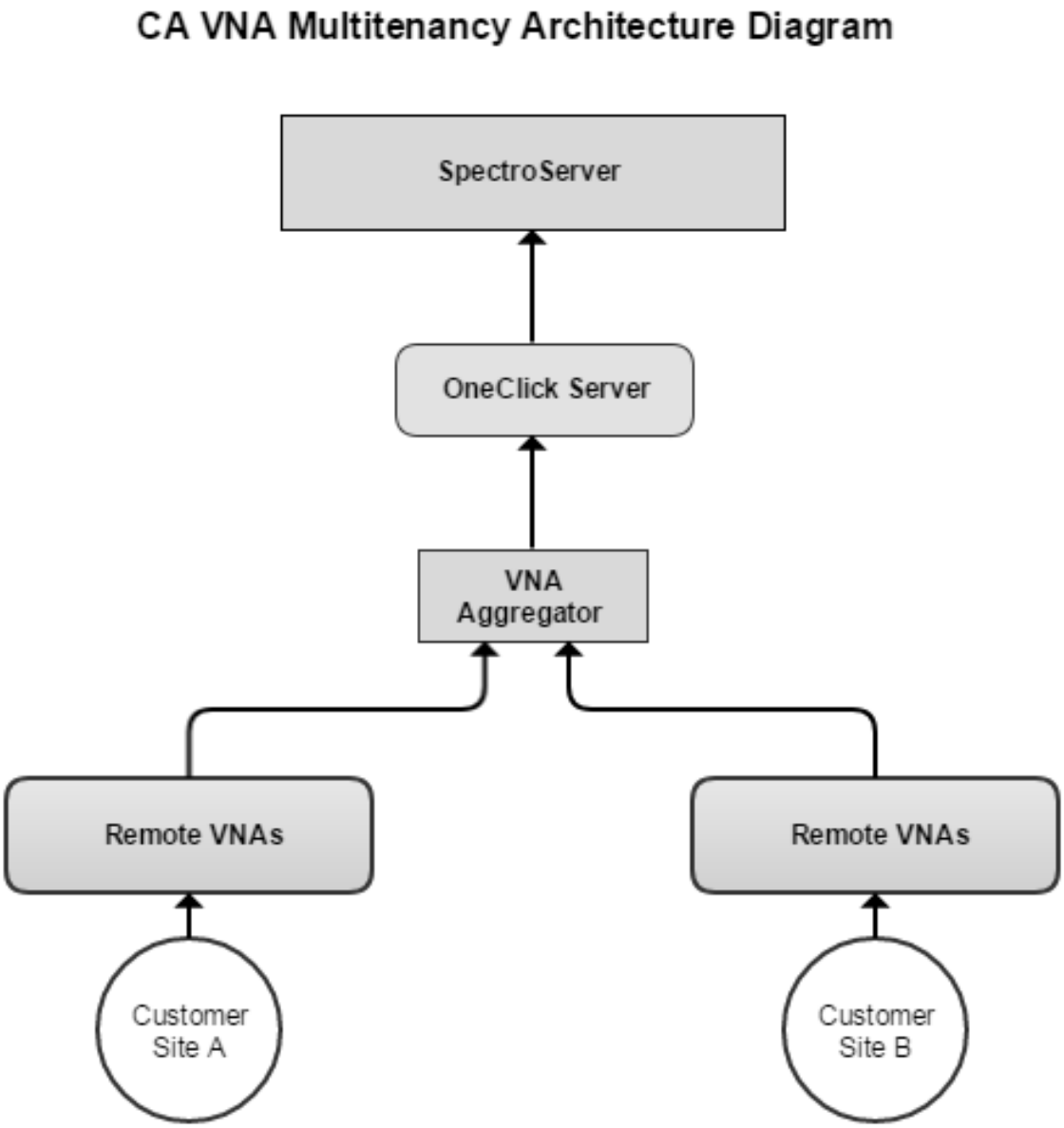
DX NetOps VNA Aggregator Integration

DX NetOps Spectrum 10.2.3 release introduces Multitenancy through VNA aggregator integration that enables modeling of multiple selected domains in any corresponding SpectroSERVER in a DSS environment to effectively manage several devices in your network. Through multitenancy, you can configure multiple domains to the corresponding landscapes and configure both SDN network elements via DX NetOps Virtual Network Assurance and the physical network elements through the Secure Domain Connector (SDC).

To aggregate data from multiple DX NetOps Virtual Network Assurance (VNA) hosts for DX NetOps Spectrum, you can configure and deploy a multi-tenant environment where multiple remote VNAs installed in a remote location/ domain can talk to one aggregator DX NetOps VNA. In this environment, a VNA Aggregator aggregates data from multiple remote VNA hosts and sends the data to DX NetOps Spectrum with unique entity IDs. Refer to the DX NetOps Virtual Network Assurance documentation for more information.

Multitenancy Architecture Diagram is as follows:

Figure 82: DX NetOps VNA Multitenancy Architecture Diagram



Prerequisites:

The prerequisite is to install and integrate all Secure Domain Connectors (from various domains) with SpectroSERVERs so that it would be listed during mapping, although this is not mandatory. This step is optional if the SDN physical elements (SNMP enabled) can be fetched by SpectroSERVER only via Secure Domain Connector.

NOTE

Proper SDC should be selected in case of physical network elements to be modeled.

- It is recommended that the VNA Aggregator and OneClick should reside on the same server, although it is not mandatory.
- The inventory/alarms update should have the DX NetOps Virtual Network Assurance plugin IP with the domain ID.
- It is recommended to not use or reuse an existing standalone VNA as Aggregator VNA. Instead, always use a fresh installed VNA as Aggregator.
- We strongly recommend having a dedicated VNA for Aggregation.
- Do not configure any plugin on Aggregator DX Netops VNA.

DX Netops VNA Configuration

To receive the VNA data, you must enable the integration between DX NetOps Spectrum and DX NetOps VNA, which acts as SDN Gateway. You can enable the integration through a web server from the OneClick Administration page. Specify the VNA configuration information such as the Server Host Name, Server Port, and select a SpectroSERVER to enable the integration.

Follow these steps:

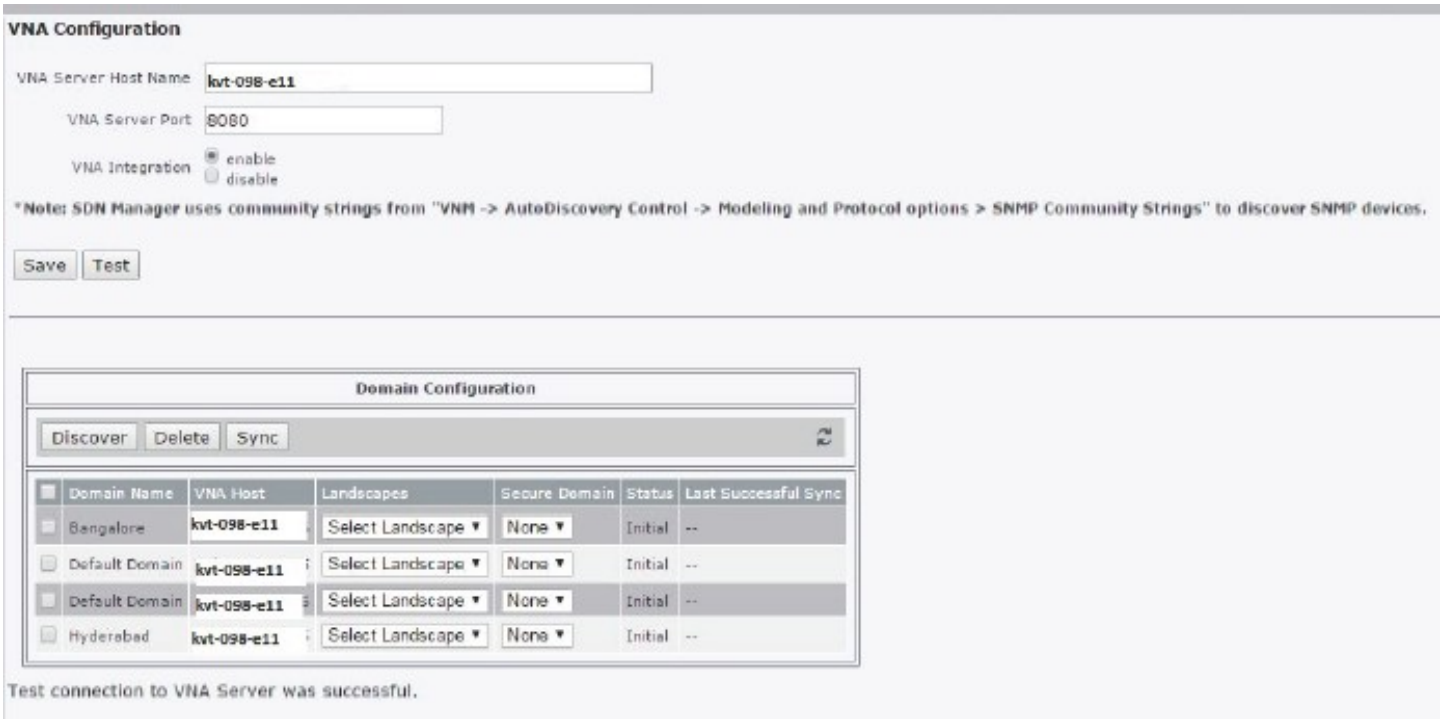
1. Open the OneClick Administration page.
2. Click the Administration tab and select the VNA Configuration from the left panel. The VNA Configuration window appears with the following configuration settings:
   - **VNA Server Host Name:** Indicates the IP address/hostname of the DX NetOps VNA Server.
   - **VNA Server Port:** Indicates the server port number of DX NetOps VNA.
   - **VNA Configuration:** Select the **Enable** radio button.
3. Click **Test** to verify if the connection is working. If test is successful, ‘Test connection to VNA Server was successful’ message appears.
4. Click **Save** and 'Successfully saved configuration to the database' message appears.
   DX NetOps VNA Configuration is now enabled and the **Domain Configuration Table** appears.

   NOTE

   SDN Manager uses community strings from "VNM -> AutoDiscovery Control -> Modeling and Protocol options > SNMP Community Strings" to discover SNMP devices.

Domain Configuration Table

The domain configuration table contains the following fields for modeling of selected domains:



The fields description is as follows:

Domain Name: Displays the selected domains fetched from the VNA server.

VNA Host: Displays the DX Netops VNA Server Hostname where userDomain is present (either DX Netops VNA name or IP address).

Landscapes: This drop-down list displays the list of SpectroSERVERs in a DSS environment. The user has to have one SpectroSERVER selected to model domain entities.

Secure Domain: By default it is None. In case the user has any private network (the network behind firewall) then the user has to have the Secure Domain Collector (SDC) installed and integrated with the SpectroSERVER. Based on SpectroSEVER selection, the SDC list will be changed. The user should ensure to have an appropriate mapping for SDC. domains will not be modeled if there is a mismatch in the SDC selection.

Status: This field displays the status of selected domains modeled in SpectroSERVER, which could be the following:

- **Initial**- Specifies the mapping which is to be done.
- **Discovery in progress** & **Sync in progress** - Specifies the waiting process for an update from VNA or that the modeling is in progress.

  NOTE

  These details are static and not dynamic in nature.
- **Discovered** - Specifies that the modeling is completed.
- **Deleted** - Specifies that the selected user domain and the corresponding models are deleted in the modeling database.

Last Successful Sync: This field displays the time and date of the last successful sync of the specified modeled domains with the VNA server.

The domain configuration table contains the following keys/buttons:

NOTE
Ensure the primary OneClick and the landscape SpectroSERVER is up and running before selecting **Discover**, **Delete** or **Sync** button.

Discover: DX NetOps Spectrum starts communicating with DX NetOps VNA and models the network elements in the corresponding mapped landscape based on the entered details under domains, landscapes, and secure domain fields and generates the status and updates the last successful sync of the existing domains which are modeled. It also starts processing faults (received from VNA) on those elements. In case of physical network elements DX NetOps Spectrum tries to identify, model the network by regular discovery methods and also resolves the connections.

Delete: To delete selected domain(s) and domain elements modeled in a corresponding SpectroSERVER.

Sync: To refresh details such as status updates and the last successful sync of selected user domain models in the modeling database. During the sync, the OneClick server sends request to the aggregator VNA to get the latest updates.

NOTE
By selecting the **Sync** button the user can remodel devices that were deleted. The **Sync** button fetches and updates the latest details of the userDomain from VNA.

WARNING
Use this option only when there are discrepancies in the inventory. Ensure the primary OneClick and the landscape SpectroSERVER is up and running.

Refresh icon: On-demand refresh when a new domain is added or removed at VNA.

Handling DX NetOps Spectrum-DX NetOps VNA On-Demand Sync of Particular Sites Issues

Due to issues in on-demand sync between DX NetOps Spectrum-DX NetOps Virtual Network Assurance of particular sites, DX NetOps Spectrum does not receive the expected data from DX NetOps Virtual Network Assurance.

From 10.4.2, the following new sync statuses are added for on-demand sync of particular sites:

- If a response is not received from DX NetOps Virtual Network Assurance, a new sync status **waiting_for_response** is displayed in DX NetOps Spectrum.
- If the web socket connection is broken between DX NetOps Spectrum and DX NetOps Virtual Network Assurance, a new status **sync_failed** is shown.

Troubleshooting with DX NetOps VNA Integration

'unable to connect to SDN Gateway through webSocket' error message found during DX NetOps Spectrum and DX NetOps VNA integration

Symptom:

The following error message is thrown 'unable to connect to SDN Gateway through webSocket' during DX NetOps Spectrum and DX NetOps VNA integration. Following is the error message log:

```
java.lang.IllegalArgumentException: no such vertex in graph: 1100
at org.jgrapht.graph.AbstractGraph.assertVertexExist(AbstractGraph.java:132)
at org.jgrapht.graph.AbstractBaseGraph.addEdge(AbstractBaseGraph.java:141)
at com.ca.em.sdn.gateway.broker.common.client.SDNClientHelper.lambda$splitUpdate
$1(SDNClientHelper.java:103)
at java.util.ArrayList.forEach(ArrayList.java:1249)
at
 com.ca.em.sdn.gateway.broker.common.client.SDNClientHelper.splitUpdate(SDNClientHelper.java:10
```

```
at
 com.ca.spectrum.app.sdn.integration.manager.SpectrumInventoryUpdater.updateReceived(SpectrumIn
at
 com.ca.em.sdn.gateway.broker.common.client.ListenerManager.notifyListenersOfInventoryUpdate(Li
at
 com.ca.em.sdn.gateway.broker.common.client.NotifierTask.notifyListeners(NotifierTask.java:65)
at com.ca.em.sdn.gateway.broker.common.client.NotifierTask.run(NotifierTask.java:36)
at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142)
at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:617)
at java.lang.Thread.run(Thread.java:745)
```

Solution:

Such exceptions occur when the DX NetOps VNA inventory is corrupted. To overcome this issue, reinstall DX NetOps VNA.

Aggregator DX NetOps VNA is sending unwanted Full updates to OneClick

Symptom:

Aggregator DX NetOps VNA is sending unwanted full updates to OneClick. Alarm occurrence count is increased when an integrated DX NetOps VNA is queried from another SDN gateway client. Aggregator DX NetOps VNA should only connect to one OneClick. In case another client connects to the Aggregator DX NetOps VNA, it sends Full updates to all the connected OneClick clients.

Solution:

Ensure only one OneClick is installed with a DX NetOps VNA Aggregator.

Inventory is not modeled or models are duplicated

Symptom:

Inventory is not modeled or models are duplicated.

Solution:

Follow these steps:

1. Run sync on the domain
2. Ensure that the SpectroSERVER and the primary OneClick is up and running.
3. Check the Tomcat logs on the primary OneClick for error messages.

Leaf and spine switches are not getting modeled

Symptom:

Leaf and spine switches are not modeled.

Solution:

DX NetOps Spectrum uses community strings from VNM>AutoDiscovery Control>Modeling and Protocol options→SNMP community string to discover SNMP devices. Follow these steps:

1. Ensure the device SNMP credentials are added.
2. Log into the APIC portal and select **Fabric** on the top bar of the page. Navigate to the **Fabric Policies**> **Pod Policies** > **Policies** > **SNMP** and check for the SNMP policies.

3. Add the SpectroSERVER IP address to the **Client Group Policies** list to discover and model ACI elements in the corresponding SpectroSERVER.

   NOTE

   Upgrade from the previous version is not supported. Users have to disable the existing integration and integrate it again with VNA 3.5.1.

'Site is missing' how to find out if the remote DX NetOps VNA is down from DX NetOps Spectrum?

Symptom:

Remote DX NetOps VNA connection is missing.

Solution:

Follow these steps:

1. You can check the connection status between DX NetOps VNA Aggregator and the remote DX NetOps VNA.
2. Check for DX NetOps VNA troubleshooting in the DX NetOps VNA documentation.

Fault data is missing, how to find out if it is a DX NetOps VNA or a DX NetOps Spectrum issue?

Symptom:

Fault data from APIC faults view is not displayed in the DX NetOps Spectrum alarm console.

Solution:

Follow these steps

1. Check if fault data is supported by DX NetOps VNA. Unsupported faults are not sent to DX NetOps Spectrum by VNA and therefore will not be displayed.
2. Check if fault data is present in the DX NetOps VNA db (MySQL). If fault is not present in the VNA db, it is either unsupported.
3. Enable SDN Gateway debug from the OneClick Admin page.

'Modeling in progress' (an attribute on SDN Manager) stuck for longer time of this?

Symptom:

DX NetOps Spectrum modeling job is stuck or not completed.

Solution:

1. Check whether any exception occurred in the OneClick server logs.
2. If 'Modeling in progress' attribute (SDN_Modelling_inProgress) is true for a longer time, enable the DX NetOps VNA configuration debug log in OneClick web page.
   To enable DX NetOps VNA configuration debug log in OneClick web page:
   a. Go to the Administration page of OneClick > Debugging > Web Server Debug Page (Runtime) > VNA Integration Information > ON radio button> Desired Level=Max > Apply
   b. Administration > Debugging > Web Server Debug Page (Runtime) > VNA Notification Sync > ON > Desired Level=Max > Apply
   c. Collect the OneClick Web Server Logs in windows and Linux
3. Disable SDN_Modelling_inProgress attribute (false) and perform **Sync** action on one of the user domains modeled in that SpectroSERVER.
4. If the problem persists, approach the DX NetOps Spectrum support team by providing the OneClick server logs.

WARNING

To enable debug logs in the DX NetOps VNA server, see DX NetOps VNA documentation.

Monitoring SDN Devices

SDN Manager

After the DX NetOps VNA Configuration is complete, you can see a folder that is named 'SDN Manager' in the Explorer tab of DX NetOps Spectrum. All SDN related entities are listed under the SDN Manager folder. When you execute a discovery in DX NetOps Spectrum, the following SDN entities are available under the SDN folder: In SDN Hierarchy the locations of Models are updated in the SdnContainsModelNameString attribute (0x133e4) If DX NetOps Spectrum need to support new faults from DX NetOps Virtual Network Assurance, those can be updated in the `EventDisp` file in `$SPECROOT/SS/CsVendor/SDN/` location.

- Compute - Contains all Hypervisors
- Fabric - Contains all switches and routers
- Technologies - Lists the technologies that are supported. For example, Cisco ACI, OpenStack, vSphere, and Juniper Contrail.
- Tenants - Contains all tenants (Applications Profiles, Virtual Machines, VRFs, and so on).

In SDN Hierarchy the locations of Models are updated in the SdnContainsModelNameString attribute (0x133e4) If the DX NetOps Spectrum need to support new faults from DX NetOps VNA, those can be updated in the EventDisp file in $SPECROOT/SS/CsVendor/SDN/ location.

Logging DX NetOps VNA Inventory Data

From 10.4.2, a new attribute (`Dump_Inventory`) is introduced to store the DX NetOps VNA inventory data.

NOTE
You must enable the `Dump_Inventory` attribute after you enable integration with DX NetOps Virtual Network Assurance.

When the `Dump_Inventory` attribute is enabled in DX NetOps Virtual Network Assurance integration on the SDN Manager node, a CSV file is stored in `%OC_Tomcat%/logs` directory. The CSV file contains the entityUUID and mh of the DX NetOps Virtual Network Assurance inventory.

Search SDN Models

The Locater tab in OneClick allows you to search the following SDN Models:

- Compute
- SDN Virtual Machines
- NFVs
- Neutron Networks / Subnet
- vSwitches
- Service Function Chains
- Tenants

NOTE

If compute is put in maintenance mode, all the VMs hosted on that compute are automatically moved into maintenance mode by DX NetOps Spectrum.

Monitor Cisco ACI Devices

Cisco Application Centric Infrastructure (ACI) is a fabric, which uses policy model provision by Application Policy Infrastructure Controller (Cisco ACI) for networks, servers, storage, security, and services. The Cisco ACI related plug-in of DX NetOps VNA collects inventory for various items such as ACI Tenants, ACI End Point Groups (EPG), and so on. For more information about how DX NetOps Virtual Network Assurance supports Cisco ACI see the Cisco ACI section in the DX NetOps Virtual Network Assurance documentation.

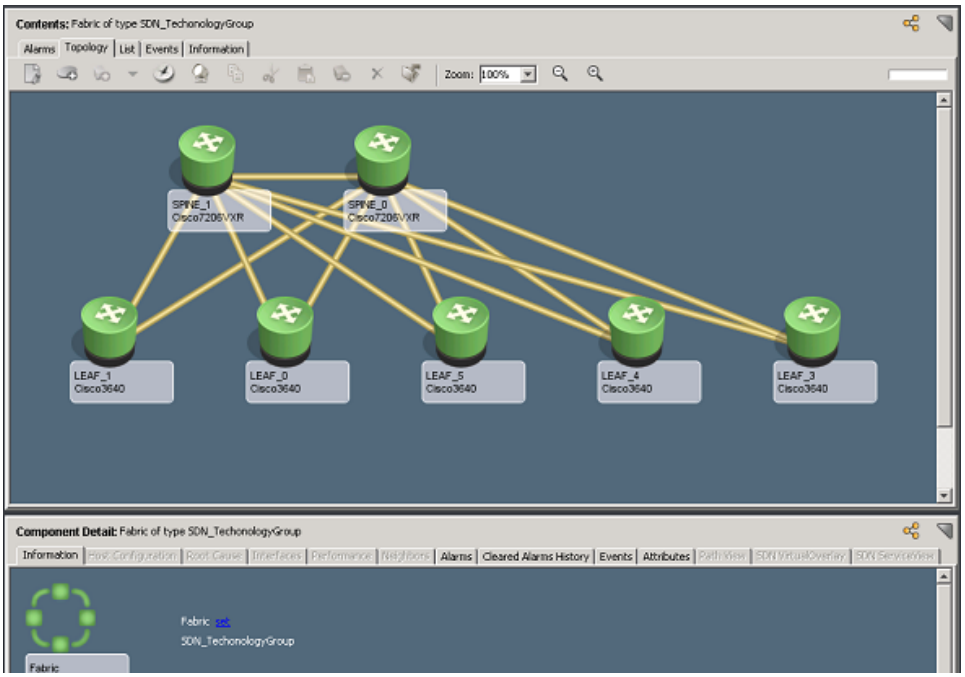
Starting from the 10.2 release, DX NetOps Spectrum supports to discover and model Cisco Application Centric Infrastructure (ACI) fabric devices. DX NetOps Spectrum displays the following Cisco ACI views in OneClick:

- Fabric View
- Contract View
- Application Profile View
- EPG View

Fabric View

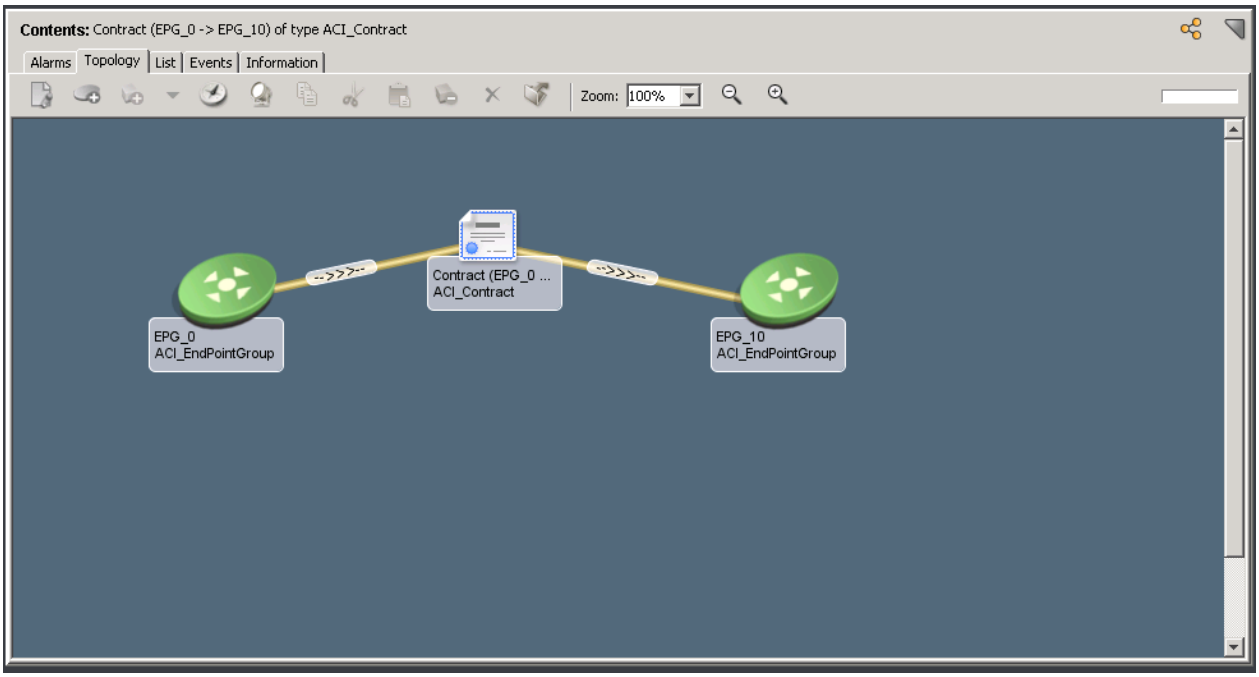
The ACI Fabric View displays the connections between Leaf and Spine switches. After the discovery in DX NetOps Spectrum, you can see all the connections from Leaf to Spine switches.

> **NOTE**
>
> To see the Fabric view of Cisco ACI devices in DX NetOps Spectrum, you must integrate DX NetOps Spectrum with DX NetOps Virtual Network Assurance.



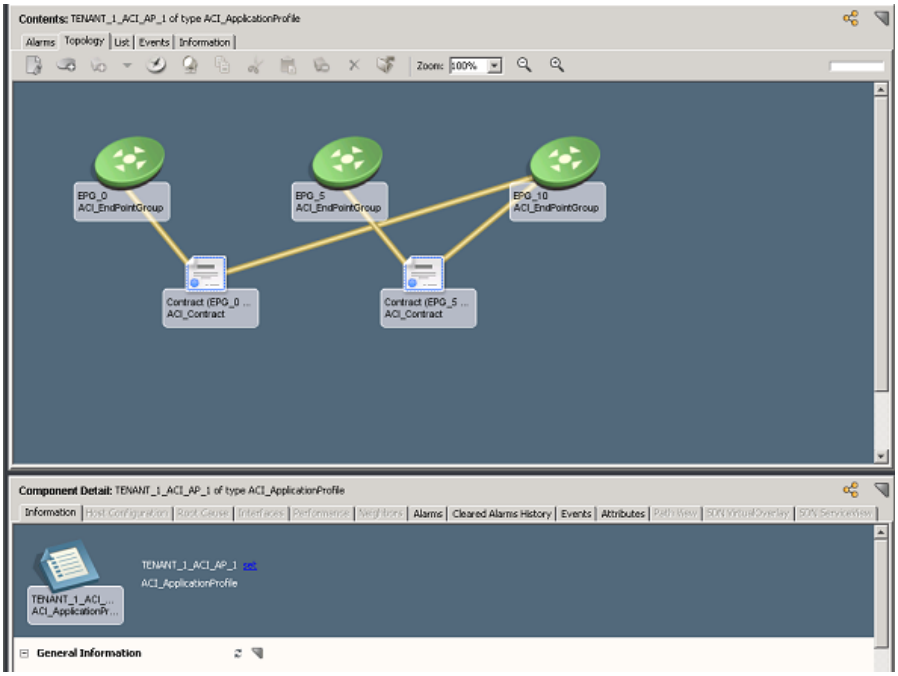
Contract View

The Contract View displays the End Point Groups that are providing and consuming the contracts.



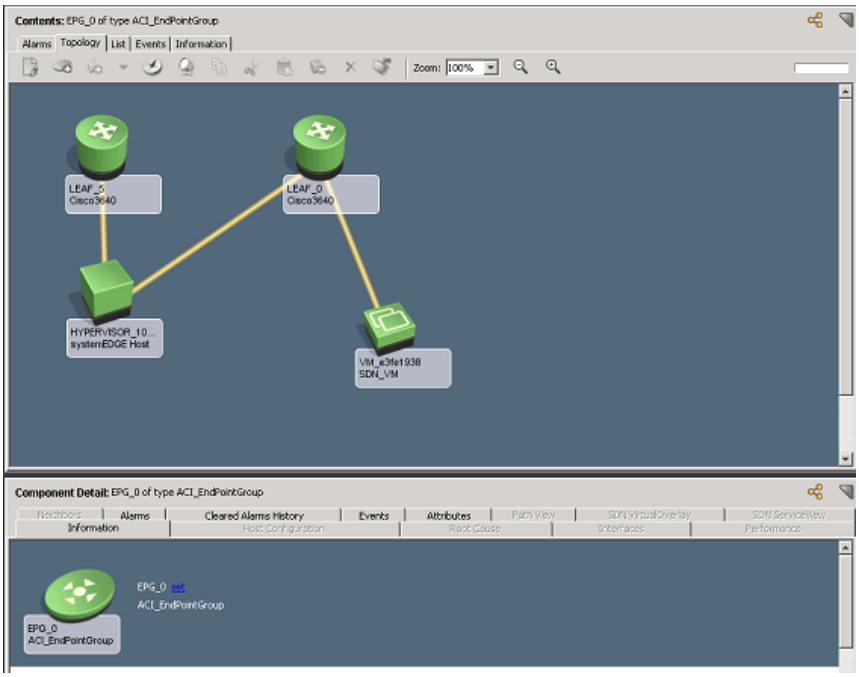
Application Profile View

The Application Profile View displays all the contracts, End Point Groups that are associated with a tenant.



EPG View

The ACI End Point Groups View displays endpoints that are associated with the group and their upstream leaf switch connectivity.



Support for Cisco ACI Faults

Cisco ACI system creates a fault based on specific conditions like component failures or an alarm. Faults are automatically created, escalated, de-escalated, and deleted by the system as specific conditions are detected. DX NetOps Spectrum receives both inventory (managed elements or devices) and Faults on the managed elements or devices from DX NetOps Virtual Network Assurance. For a detailed list of DX NetOps Spectrum supported Cisco ACI faults, refer to the Supported ACI Faults file.

DX NetOps Spectrum asserts the fault to the right entity (managed elements or devices) and displays an alarm in the OneClick alarm view.

NOTE

For entities that are not modeled in DX NetOps Spectrum using DX NetOps VNA, notifications are shown on the site model.

NOTE

Only the events which were created after the DX NetOps VNA plug-in are configured and saved in DX NetOps VNA DB and they are sent to DX NetOps Spectrum.

Correlation of SDN Fabric Devices

The following correlations are supported by Fabric (spine/leaf) Devices.

F1543 - fltFabricNodeInactive

This fault occurs when a leaf or spine node of a fabric domain is down, the ACI sends fault to DX NetOps Virtual Network Assurance and from DX NetOps Virtual Network Assurance, DX NetOps Spectrum receives the fault. Also, generates SNMP fault in DX NetOps Spectrum for that corresponding node. The SNMP faults and ACI faults are correlated and displayed in the DX NetOps Spectrum OneClick Console.

F0532 -fltEthpmIfPortDownInfraEpg

This fault occurs when an interface of leaf or spine node of a fabric domain is down, the ACI sends fault to DX NetOps Virtual Network Assurance and from DX NetOps Virtual Network Assurance, DX NetOps Spectrum receives the fault.

Also, generates SNMP fault in DX NetOps Spectrum for that corresponding node interface. The SNMP fault and ACI fault are correlated and displayed in the DX NetOps Spectrum OneClick console.

F1822: fltEqptcapacityFSPartitionFsPartitionLimitsExceededCritica

This fault occurs when a disk usage of a partition increases beyond its threshold. The ACI sends fault to DX NetOps Virtual Network Assurance and from DX NetOps Virtual Network Assurance, DX NetOps Spectrum receives the fault. Also, generates SNMP fault in DX NetOps Spectrum for that corresponding node. The SNMP fault and ACI fault are correlated and displayed in the DX NetOps Spectrum OneClick console.

In all the above scenarios the correlations are performed between SNMP fault and DX NetOps VNA Fault of Spine/Leaf. The SNMP fault is shown as the root cause and DX NetOps VNA fault is shown as a symptom.

ACI Events

DX NetOps Spectrum processes ACI events that are received from DX NetOps Virtual Network Assurance. By default, all DX NetOps VNA alarms are asserted with event code 0x673000c and event assertion 0x673000b. This event contains event conditions such as fault code and entity ID, which help to generate a corresponding Pcause code.

Steps to check if a fault code is supported by DX NetOps Spectrum:

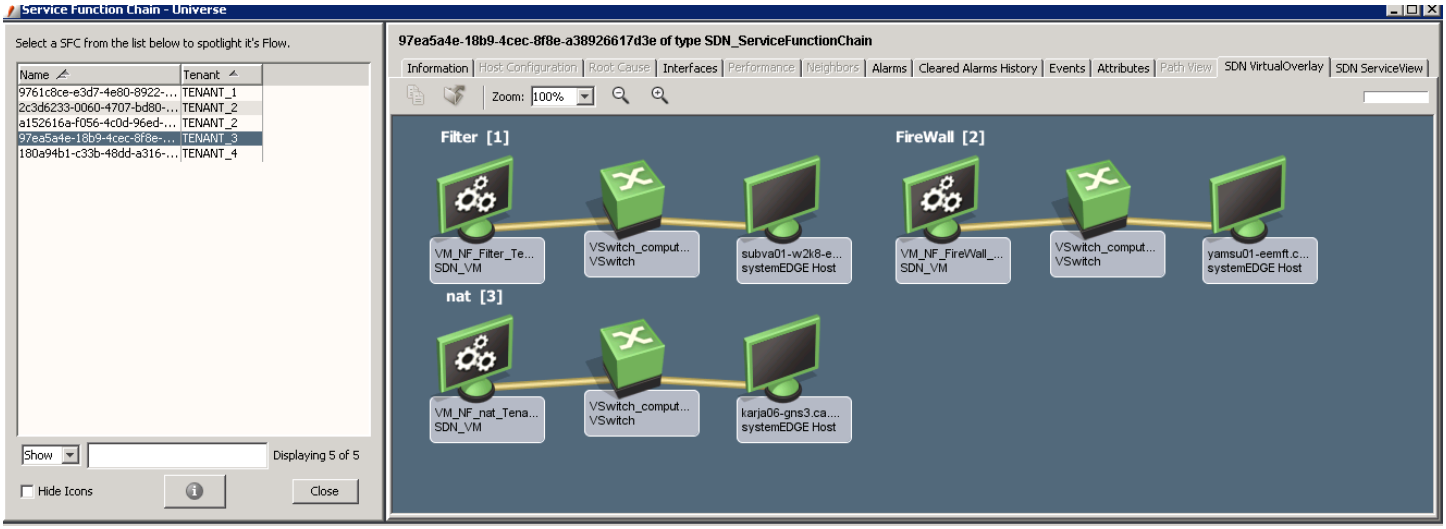
1. Go to Location <$SPECROOT>/SS/CsVendor/SDN
2. Open the SDN EventDisp file in a text editor
3. Search for the fault code in the file. For example, if you want to check fault F0532, search for 0532 in the file.
4. If you find an uncommented entry under CA EventCondition rule, then the fault is supported by DX NetOps Spectrum.
5. If you do not find any entry, then the fault is not supported by DX NetOps Spectrum.

NOTE

Due to the huge number of rules attached to this particular event code 0x673000c, you cannot modify it from the Event configuration Editor. But, you can directly open the event code file in a text editor and modify it.

SDN Virtual Overlay

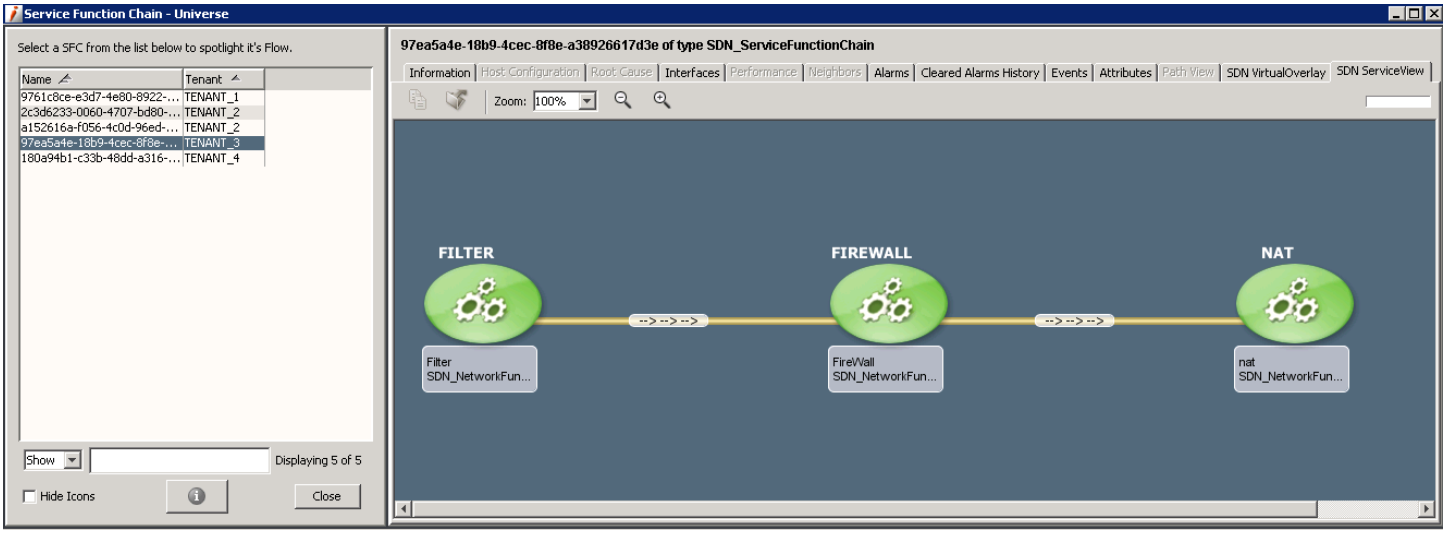
The SDN Virtual Overlay tab shows the network functions that are part of the selected SFC and Compute on which these network functions are hosted.



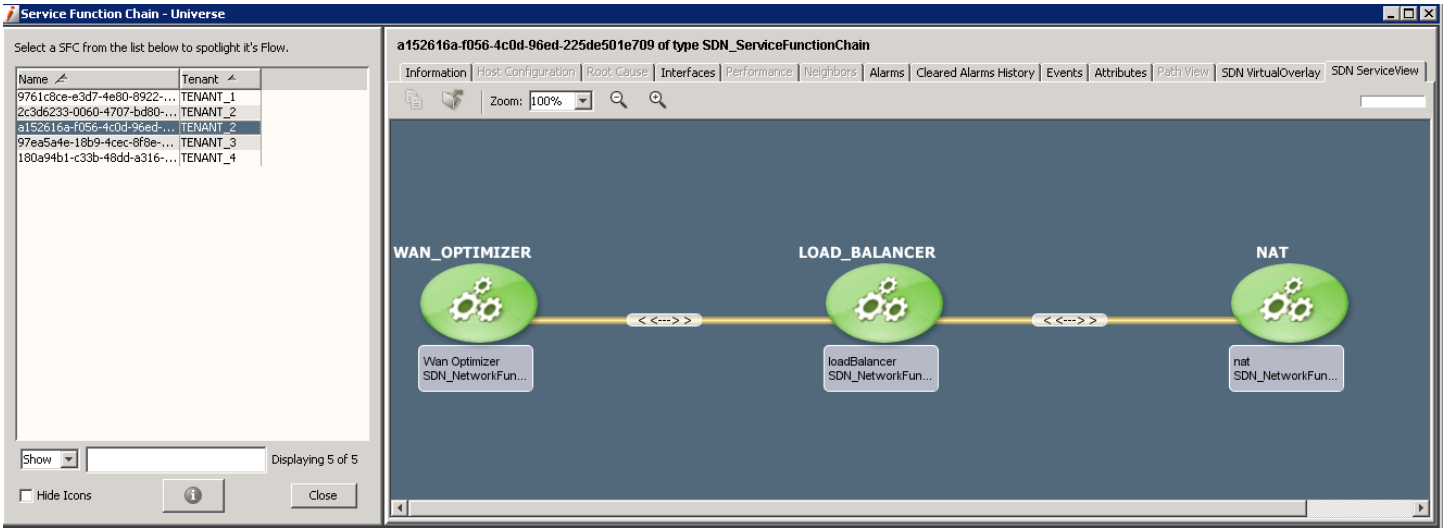
SDN Service View

The SDN Service View tab allows you to view the order of the network functions and direction of the packet flow (unidirectional or bidirectional) for the selected Service Function. The following images show how the Unidirectional and Bidirectional Service View looks like.

Unidirectional Service View:



Bidirectional Service View:



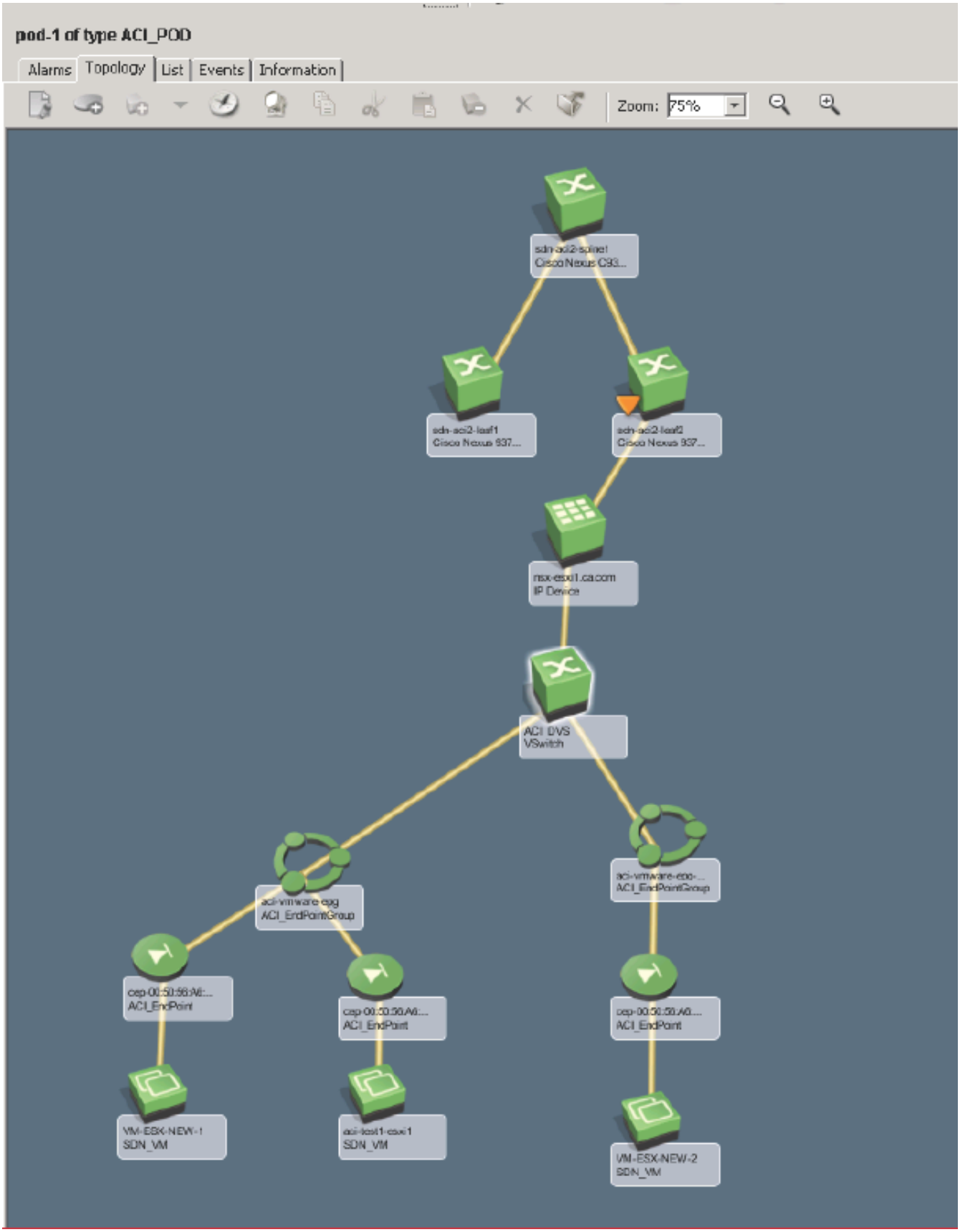
Monitoring VMware vSphere with ACI

DX NetOps Virtual Network Assurance 3.5.1 supports vSphere which communicates to vCenter and gives detailed information about data centers, ESXs and VMs. DX NetOps Spectrum then models all these entities and displays the entire topology from the leaf switches to the VMs, provided the vSphere is integrated with Cisco ACI.

The following are the prerequisites to see the topology:

1. SNMP must be enabled on ESXi Host.
2. DVS should have been created by ACI as part of ACI/VMware integration.
3. Hosts should be connected to Leaf switches.

Following is the topology view:


pod-1 of type ACI_POD
Alarms
Topology
List
Events
Information
Zoom:
75%
sdn-aci2-leaf1
Cisco Nexus 937...
sdn-aci2-leaf2
Cisco Nexus 937...
nsx-esxi1.ca.com
IP Device
ACI_DVS
VSwitch
aci-vmware-epg
ACI_EndPointGroup
ACI_EndPoint
ACI_EndPoint
ACI_EndPoint
VM-ESX-NEW-2
SDN_VM

NOTE

here is no connectivity between VMs and vswitch/DVS.

ACI and vSphere plugin correlation

Cisco ACI integrates with the VMware vCenter instances to transparently extend the Cisco ACI policy framework to VMware vSphere workloads. Cisco ACI creates a virtual distributed switch (VDS) in VMware vCenter to create a virtual network. From that point onward, Cisco ACI manages all application infrastructure components. The network administrator creates EPGs and pushes them to VMware vCenter as port groups on the DVS. Server administrators can then associate the virtual machines and provision them accordingly.

Viptela IWAN

Viptela is an SD-WAN solution. DX NetOps Virtual Network Assurance collects inventory and performance metrics from Viptela to support SD-WAN monitoring.

The plug-in collects inventory for the following items:

- Sites
- vEdge router
- vEdge interfaces
- Tunnels
- vManage
- vSmart
- Application/SLA Paths

Refer to Device Certifications section for more information about vEdge router, vSmart Controller, vManage, and vBond Orchestrator certifications.

To know more about SD-WAN support for Viptela, refer to the DX NetOps VNA Documentation for Viptela.

Service Function Chain (SFC)

The Service Function Chains that are configured in the SDN environment are displayed in DX NetOps Spectrum.

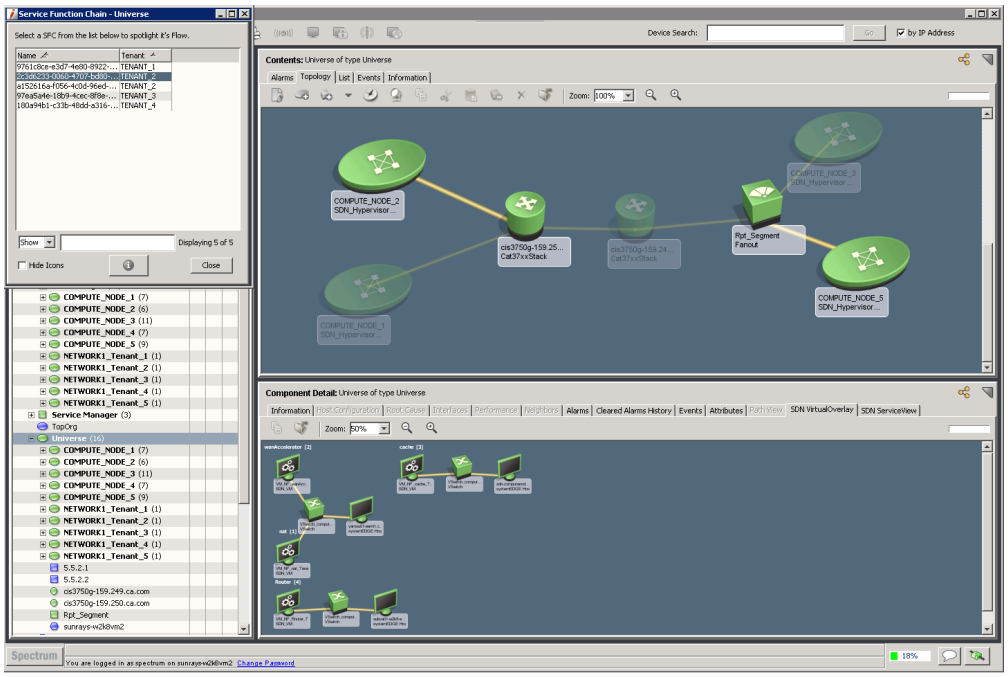
Follow these steps:

1. Open **OneClick**.
2. Expand the desired landscape on the **Explorer** tab and select **Universe**.
   Details about the selected **Universe** appear in the **Contents** panel.
3. Click the **Topology** tab.
4. Click the Spotlight View icon

   ,
   then select **Service Function Chain** from the list.
   The Service Function Chain dialog displays the list of SFCs and tenants that are associated with each SFC.
5. Select a **Service Function** from the list.
   The computer which has VMs associated with the SFC are highlighted in the Universe.

   NOTE
   The SDN Virtual Overlay and SDN Service View tabs in the Component Detail panel are enabled only when you select Universe.



NOTE

You can use the **Hide Icons** option in the **Service Function Chain List** dialog to view only the selected Service and its connections.

NOTE

To view Component details like SDN Service View and SDN Virtual Overlay details for the selected model in the Service Function list, click information the icon on the SFC dialog.

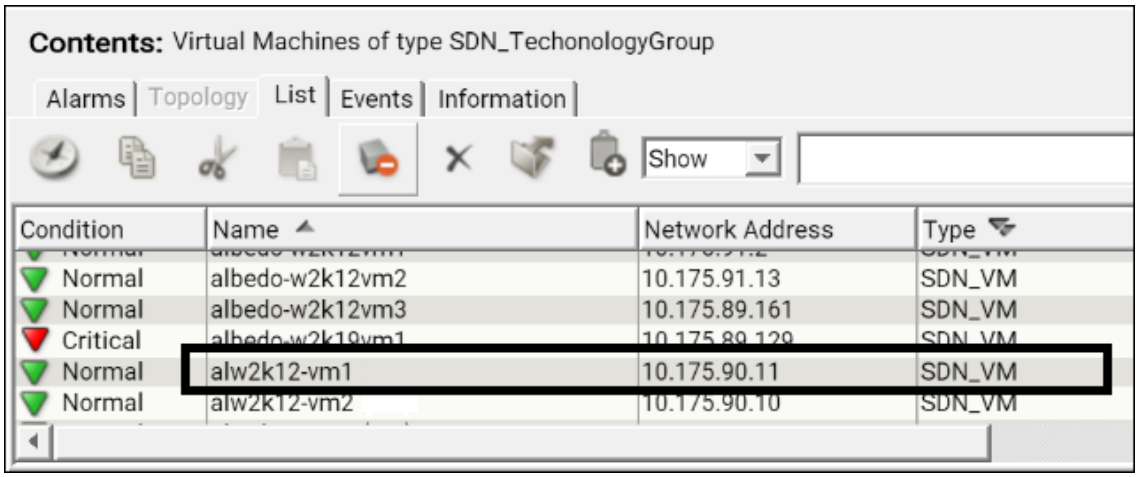
6. The Service Function Chain page integrates service chains into the traditional monitoring environment. The page shows you SDN Virtual Overlay and SDN Service View and also shows the physical and virtual connections of the Compute in the Service Chain.

Upgrade IP Devices to SNMP Devices

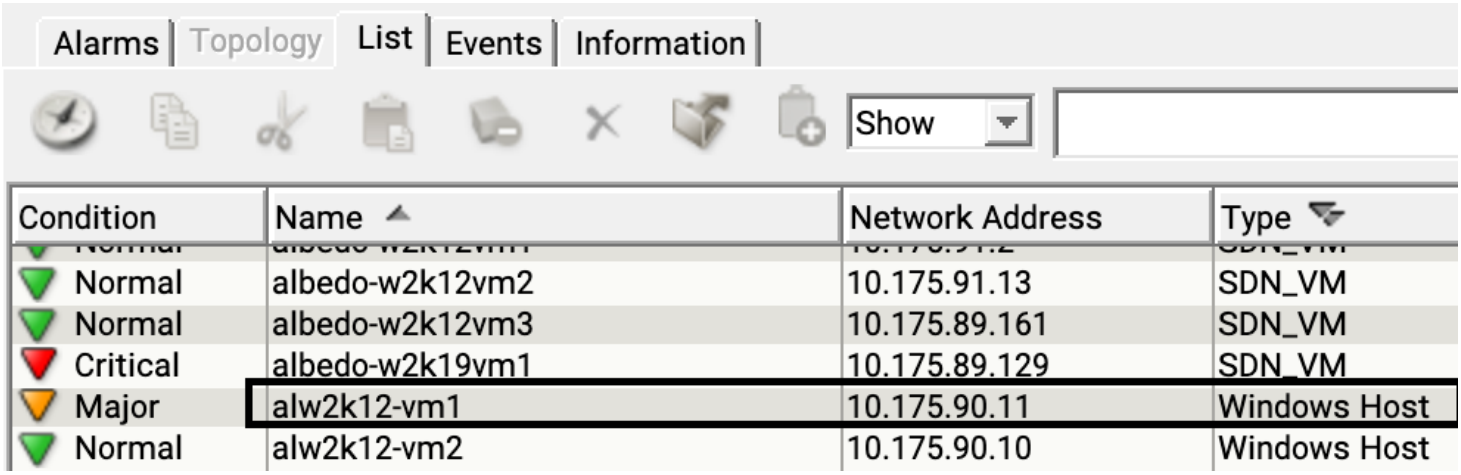
From 10.4.2, you can upgrade your existing IP pingable devices and re-model the devices to support SNMP services. You can search for these SDN IP pingable devices from the Explorer tab in OneClick. You must set the **ApplySNMPCapabilitiestoSDNVMs** attribute to **Yes** to monitor the devices as SNMP devices.

Follow these steps:

1. Select **SDN Manager** in the **Explorer** tab to view the devices.
2. Select **List** view in the **Contents** section.
   You can see a list of all devices.



3. Select **SDN Manager** in the **Explorer** tab.
4. Select **Attributes** tab, in the **Component Detail** section.
5. Search the **ApplySNMPCapabilitiestoSDNVMs** attribute.
6. Right-click the **ApplySNMPCapabilitiestoSDNVMs** attribute and select **Add**.
   The ApplySNMPCapabilitiestoSDNVMs attribute is moved to the right-hand section.
7. Right-click the **ApplySNMPCapabilitiestoSDNVMs** attribute and select **Edit**.
8. In the **Edit SDN Manager** popup, uncheck **No change** and select **Yes**.
9. Select **OK** to apply the change.
   The **Attribute Edit Results** window shows the status.
10. Close the window, on the successful assignment of the attribute.
    DX NetOps Spectrum deletes the existing SDN_VM devices which are SNMP reachable and remodel them as SNMP devices. The **List** view in the **Contents** section shows the new devices. The **Type** column shows the SDN_VM as a corresponding SNMP Model.



Disable SNMP Modeling

From 10.4.2.2, you can disable SNMP model devices from SDN Manager. By default, **Disable SNMP Modeling** option is set to **NO**. If you Enable VNA integration with **Disable SNMP Modeling** option as NO, all the devices which are SNMP reachable get modeled as SNMP model. When DX VNA integration is disabled all DX VNA models except SNMP are deleted.

If you re-enable the integration with DX VNA with the **Disable SNMP Modeling** option as NO, Since SNMP models are already existing, models are reconciled and not duplicated. Enable DX VNA integration with the **Disable SNMP Modeling**

option as YES, devices that are SNMP reachable are modeled as virtual models and duplicate models for the same IP are created (No reconciliation).

> **NOTE**
> Irrespective of **Disable SNMP Modeling** option, SPINE, and LEAF switches are modeled as SNMP only.

Dump OCS DX VNA Inventory Cache

From 10.4.2.2, inventory cache which maps DX VNA entities with DX NetOps Spectrum model handles, will be dumped in the connected OneClick server. A CSV file is generated under tomcat logs.

If No (Default), IP devices from DX VNA integration are modeled through SNMP discover. Devices which are not SNMP reachable, are modeled as a virtual device. If Yes, IP devices from DX VNA integration are modeled as virtual devicæs irrespective of their SNMP reachability. Hence, the overall discovery end modeling time is reduced. Duplicate models may be crested for SNMP devices modeled prior to the DX VNA integration (No reconciliation).

> **NOTE**
> This configuration is ignored for Cisco ACIswitches (SPINE/LEAF) as DX NetOps Spectrum always them to be SNMP enabled.

Follow these steps:

1. Navigate to Locator, SDN Manager, Information tab
2. Select SDN Manager Configuration
3. Select **Dump Now** in the **Dump Inventory** field.

Troubleshooting ACI Faults

How To Enable Debug Logs for ACI Faults in DX NetOps Spectrum

Symptom: I want to see debug logs for ACI Faults in DX NetOps VNA Integration to troubleshoot ACI inventory and faults.

Solution:

Follow these steps:

1. Go to OneClick **Administration** Page.
2. Select **Debugging** option from the menu bar.
3. Select the **Web Server Debug Page (Runtime)** The Debug Controller table appears.
4. Search for **VNA Integration Information** and select the **On** radio button.
5. Search for **VNA Notification Sync** and select the **On** radio button.
6. Go to $SPECROOT/tomcat/logs and check the debug logs for **Notification Registration**, **Historical** and **Live ACI faults** in catalina.out/stdout.log files.

When the DX NetOps VNA Integration is enabled, ACI Historical Faults are not received in OneClick Console

Symptom: When DX NetOps Spectrum is integrated with DX NetOps VNA, I cannot see Historical Faults for ACI in DX NetOps Spectrum OneClick Console.

Solution:

Follow these steps:

1. Enable Debug logs (see "How to enable debug logs" steps mentioned in the above topic)
2. Verify if the DX NetOps VNA database contains Faults.
   To verify follow these steps:

a. Log in to DX NetOps VNA host.
b. Navigate to cd /opt/CA/MySql/bin.
c. Run the following commands:
./mysql -u user -p (user should be Installed user name)
use vna
select * from notifications where notificationType='Alarm';

3. If the VNA database contains faults and those faults are not received in OneClick Console, then verify whether the Web Socket Connection is established between DX NetOps Spectrum and CA VNA.
To verify follow these steps:
a. Go to OneClick **Administration** Page.
b. Go to **VNA Configuration**.
c. Click the **Test** button to check the Web Socket Connection status. You can see 'Test connection to VNA Server was successful' message.
d. Go to $SPECROOT/tomcat/logs and check for connection time out log in catalina.out/stdout.log files.

ACI Live Faults are not received in OneClick Console

Symptom: When DX NetOps Spectrum is integrated with DX NetOps VNA, I cannot see ACI Live Faults in DX NetOps Spectrum OneClick Console.

Solution:

Follow these steps:

1. Verify the VNA database if the faults are received from APIC and also check the fault status.
To verify the DX NetOps VNA database, follow these steps:
a. Log in to DX NetOps VNA host.
b. Navigate to cd /opt/CA/MySql/bin.
c. Run the following commands:
./mysql -u user -p (user should be Installed user name)
use vna
select * from notifications where notificationType='Alarm';
2. If the faults state is shown as 'cleared' then those faults not received in DX NetOps Spectrum OneClick Console.
If the faults state is other than cleared, then verify the catalina.out/stdout.log files in $SPECROOT/tomcat/logs.

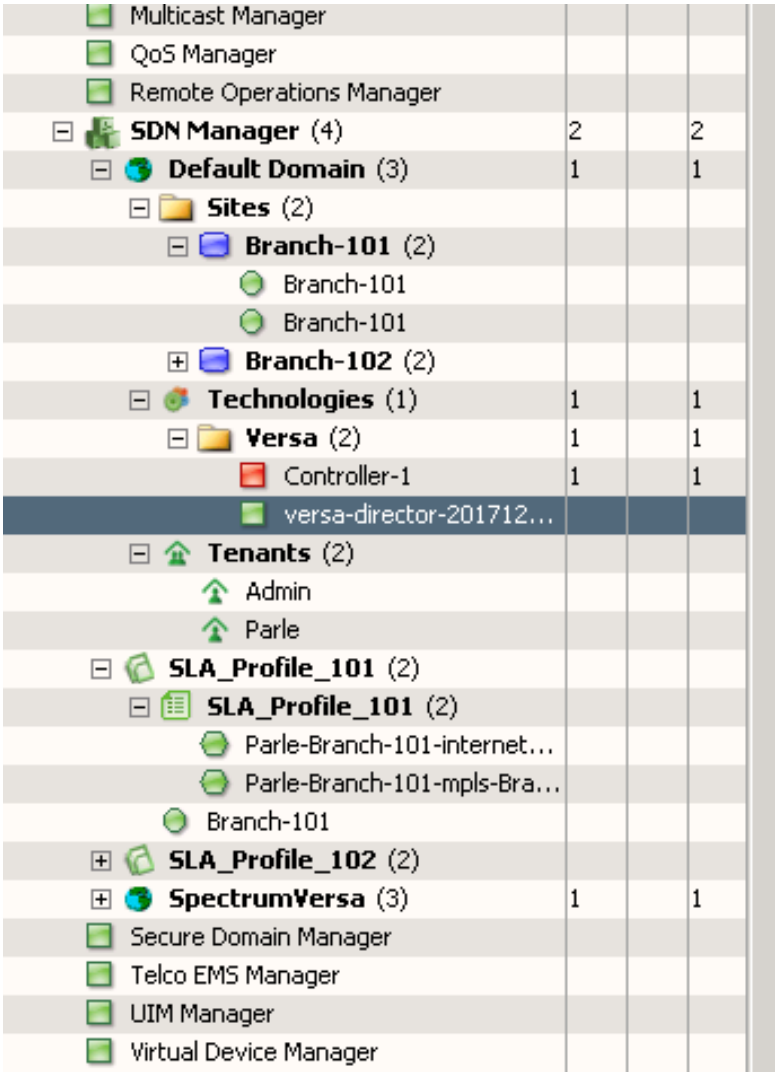
Monitoring SD-WAN for Versa

10.3.1 supports the monitoring of Versa devices through DX NetOps VNA integration. When DX NetOps Spectrum is integrated with DX NetOps VNA (configured with Versa plug-in), DX NetOps Spectrum receives the inventory information of the Versa devices through DX NetOps VNA.

> **NOTE**
> VNA configured with Versa plug-in. acts as an SDN Gateway to collect Versa inventory information and forwards information to DX NetOps Spectrum. Ensure that the DX NetOps VNA must be configured with Versa plug-in.

The DX NetOps Spectrum and DX NetOps VNA integration fetches the following Versa entities inventory information and displays under the SDN Manager hierarchy in OneClick.

- Sites
- Branch Router
- Director
- Controller
- Policy Group



This integration supports Versa entities synchronization. When the DX NetOps Spectrum and DX NetOps VNA integration is enabled, synchronization happens automatically at the scheduled time interval. Additions, deletions, and modifications of Versa entities in a Versa environment are reflected in DX NetOps Spectrum.

Versa Topology

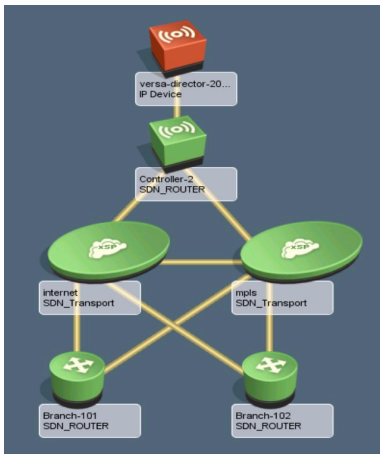
DX NetOps Spectrum displays topology for Versa devices.

Topology for Director

The overlay topology is seen by clicking the director, it displays the devices and how they are connected through the various transports.

Hub-Spoke Topology:

- The Director model connects to the controller models directly.
- Branch routers (BRs) are inter-connected to each other through transports.
  For example, MPLS transport, Internet transport and so on.
- The controller also has a connection towards branch routers through transports - as shown in the following image:



IMPORTANT
Topology is supported only for non-SNMP Versa models created by DX NetOps VNA Integration.

Versa Inventory

You can view the Versa entities such as Branch Router, vBond, Controller, Director in the OneClick console. Logical entities such as Sites and Policy Groups can also be viewed on the OneClick Navigation page. A new container '**VNA Inventory**' is created under the Universe view, this container has the Versa entities. In the Explorer View, **SDN Manager** provides a more detailed hierarchy i.e. ***Domain > Sites > Policy Groups > Policy > branch routers*** , compared to the **VNA Inventory** view, which only displays the hierarchy of branch routers and associated Tunnels.

Reconciling Versa Entity Data in DX NetOps Spectrum

During DX NetOps VNA data synchronization, when a new Versa entity is created in versa, it is reported to DX NetOps Spectrum. DX NetOps Spectrum performs a search to identify if this entity was modeled during DX NetOps Spectrum discovery and modeling. If such an existing model is found, DX NetOps Spectrum reconciles the CA Versa entity information with the existing model, instead of creating a model.

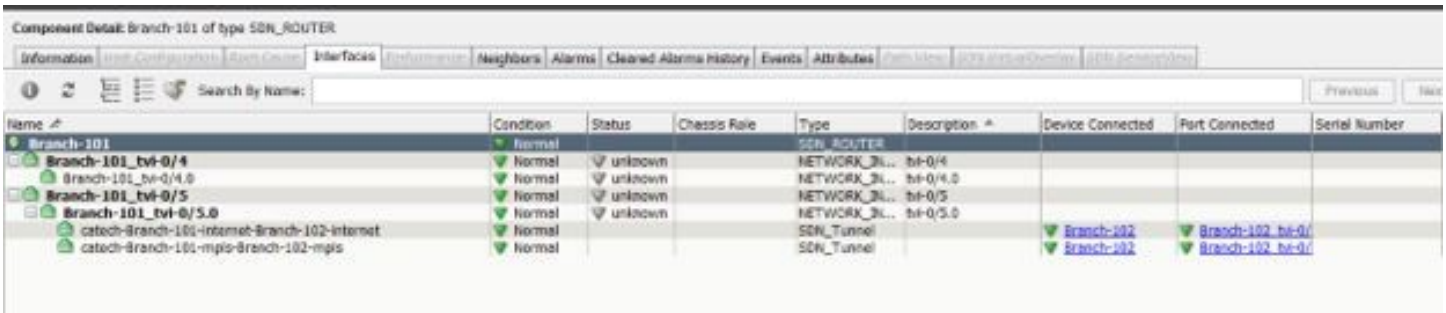
Interfaces Information

As a user, you can see the tunnels that are associated with the interfaces under the **Interfaces** tab.

- Tunnels
- Physical Interfaces
- Transport

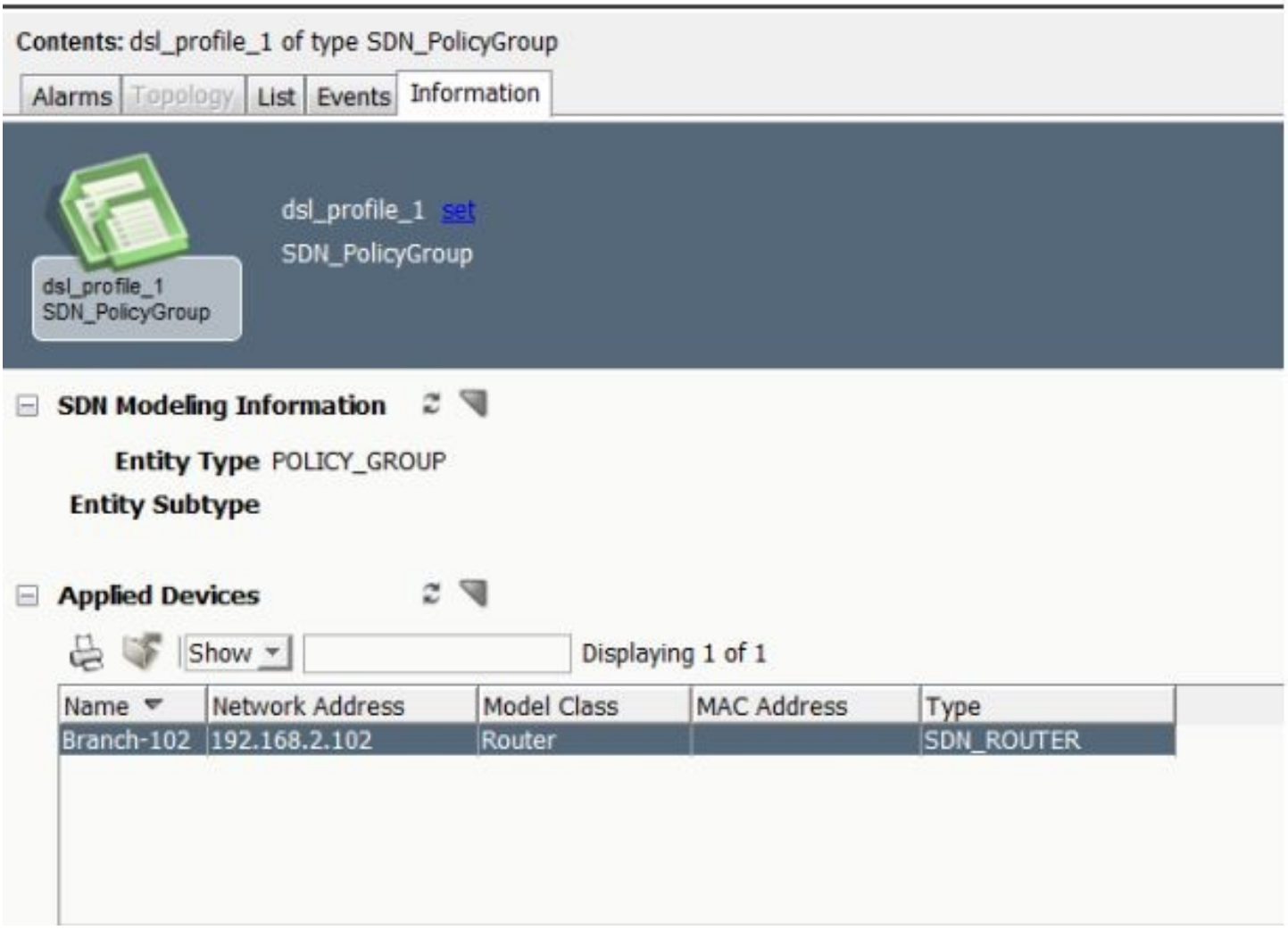
SDN Tunnels associated to Branch Router Interfaces

The Interfaces tab in the Component Details panel shows all the tunnels information which is associated with the selected branch router.



Policy Group

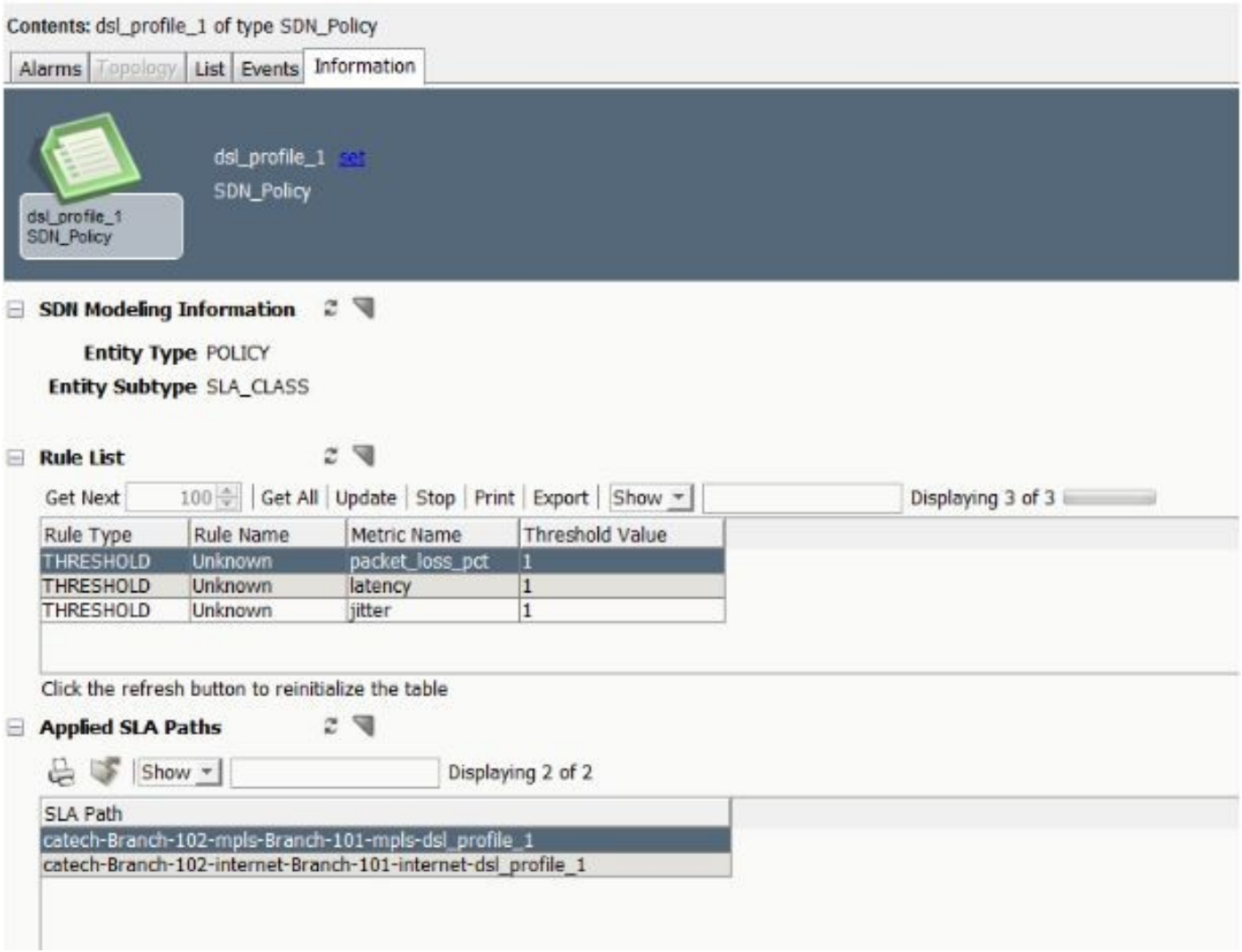
As a user you can see the edge router to which the policy group is applied:



Policies Information

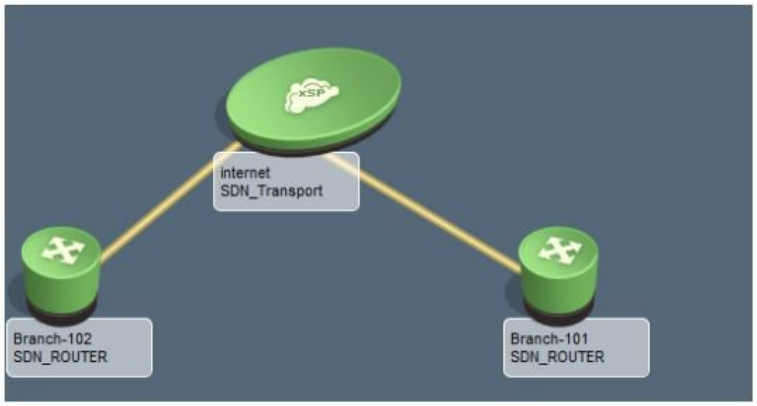
On clicking the policy, as a user, you can see the metrics that are associated with the policy and the SLA paths applied. For branch routers, associated policy and policy group information is displayed in the **Information** tab of the contents panel.

Branch router Policy Information



SLA Path

As a user you can see the edge devices and their transport associated with the SLA path:



Prior to the 10.4.2.2 release, SLA Path used to have the "Topology tab" which displayed the Branch routers connected through Transports. From the 10.4.2.2 release, same connections are shown under the SDN Overlay tab. SLA Path models are not displayed under the SDN Manager hierarchy. You can use the locater search to find SLA Path.

Events Generated in Versa Analytics

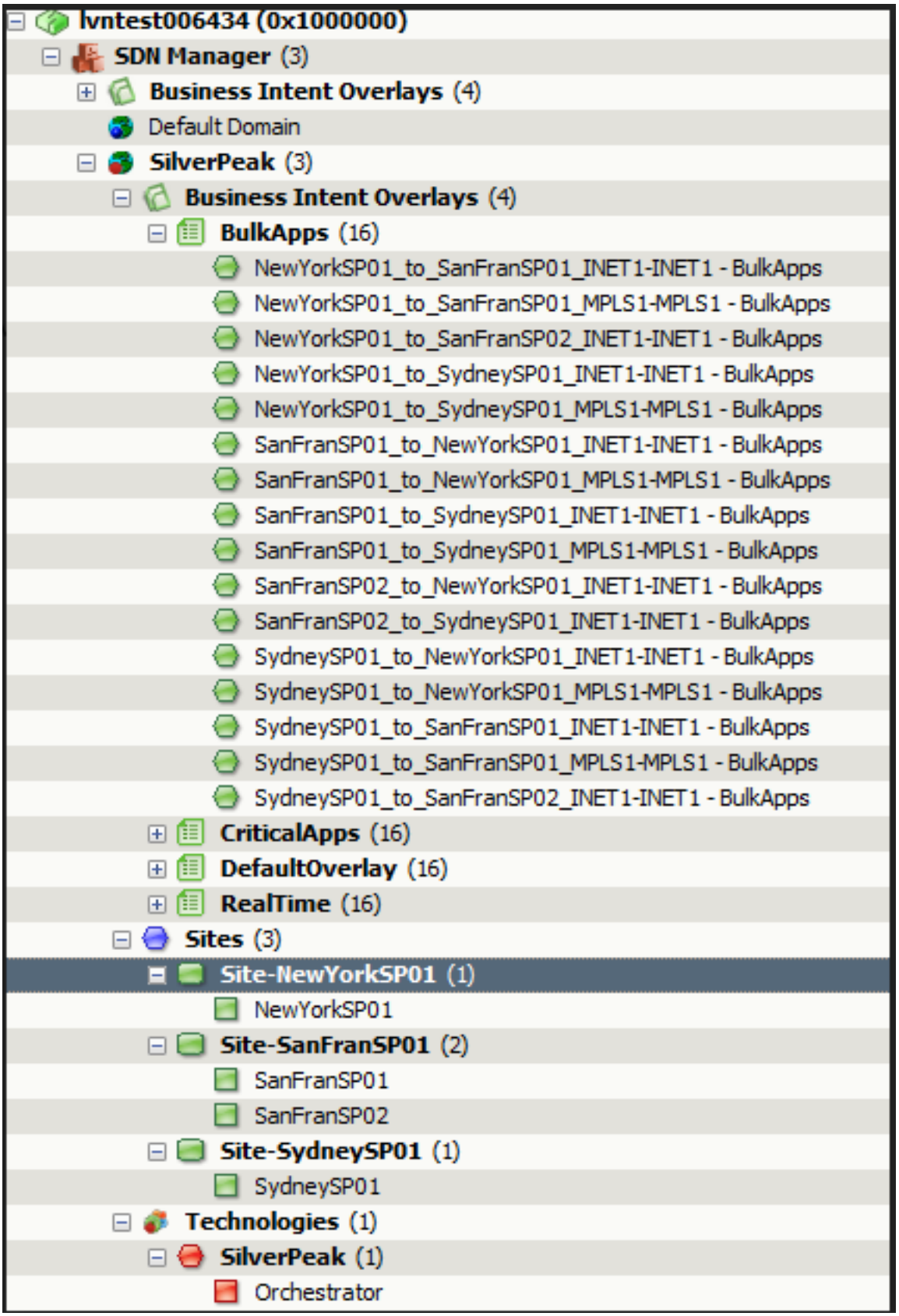
The following table provides information about the supported alarm types, threshold values, and the default destination to which the alarms are exported. Following are the events that are generated under Versa analytics and pulled into DX NetOps Spectrum:

Alarm Type	Description	Default Destination
cpu-utilization	Generated when datapath CPU utilization exceeds the configured threshold value.	SNMP, Syslog, analytics
mem-utilization	Generated when datapath memory utilization exceeds the configured threshold value.	SNMP, Syslog, analytics
disk-utilization	Generated when disk utilization exceeds the configured threshold value.	SNMP, Syslog, analytics
org-session-utilization	Generated when the number of sessions of an org exceeds configured number of sessions.	SNMP, Syslog, analytics
device-session-utilization	Generated when the number of sessions exceeds the configured number of sessions for a device/appliance.	SNMP, Syslog, analytics
Interface-down	Generated when an interface (or sub-interface) goes down.	SNMP, Syslog, analytics
uplink-bw-threshold	Generated when current uplink bandwidth exceeds the configured uplink bandwidth of an interface.	SNMP, Syslog, analytics
dnlink-bw-threshold	Generated when current downlink bandwidth exceeds configured uplink bandwidth of an interface.	SNMP, Syslog, analytics
adc-server-down	Generated when the backend server does not respond to ADC monitors for a specified amount of time. Once the server is marked down, it is not considered for load balancing.	SNMP, Syslog, analytics
adc-vservice-down	Generated when all backend servers attached to virtual service are declared down because of monitor health failure. No traffic is served by this virtual service (VIP).	SNMP, Syslog, analytics
cgnat-pool-utilization	Generated when the CGNAT pool exceeds configured threshold value or when the pool is exhausted.	SNMP, Syslog, analytics
snat-pool-utilization	Generated when the SNAT pool exceeds configured threshold value or when the pool is exhausted.	SNMP, Syslog, analytics
ipsec-tunneldown	Generated when IPSEC tunnel with a peer goes down.	SNMP, Syslog, analytics
ipsec-ike-down	Generated when IKE connection that is established with a peer goes down.	SNMP, Syslog, analytics
bgp-nbr-state-change	Generated when BGP between peers goes down or comes back up.	SNMP, Syslog, analytics
vrrp-v3-new-master	Generated when VRRP router transitions to the MASTER state.	SNMP, Syslog, analytics
vrrp-v3-new-backup	Generated when VRRP router transitions to the backup state.	SNMP, Syslog, analytics
vrrp-v3-proto-error	This notification indicates that the VRRP router has encountered protocol error like version mismatch, checksum error, or VRRP group id mismatch.	SNMP, Syslog, analytics
ddos-threshold	Generated when DDOS traffic exceeds the configured aggregate/classified DDOS threshold.	SNMP, Syslog, analytics

zone-protection-flood	Generated when flood traffic exceeds the configured zone protection threshold value.	SNMP, Syslog, analytics
port-scan-flood	Generated when PORT-SCAN from a source to destination exceeds the configured zone protection profile value.	SNMP, Syslog, analytics
sdwan-branch-disconnect	Generated a branch gets disconnected from Controller.	SNMP, Syslog, analytics
sdwan-datapath-down	Generated when all paths between two branches go down.	analytics (From Controller)
dhcp-pool-utilization	Generated when DHCP addresses are exhausted and no more addresses can be allocated from DHCP address pools.	SNMP, Syslog, analytics
software-trial-expired	Versa appliance trial period expired.	SNMP, Syslog, analytics
software-trial-error	Versa appliance trial key tampered.	SNMP, Syslog, analytics
interface-half-duplex	Generated when an interface is detected to be in Half Duplex mode.	SNMP, Syslog, analytics
nexthop-down	Generated when nexthop gateway does not respond to monitors for a specified amount of time. Once nexthop is marked down, routes are withdrawn.	SNMP, Syslog, analytics
monitor-down	Generated when IP destinations that are part of the monitor does not respond to the given type of probe packets for a specified amount of time. Once the monitor is marked down, dependent routes are withdrawn and redistribution policies are recomputed.	SNMP, Syslog, analytics
software-key-about-to-expire	Versa FlexVNF key expires soon. Contact Versa Support to replace it with a new key. For unrestricted usage, ensure Versa FlexVNF is subjugated to Versa Director and there is connectivity between Versa FlexVNF and Versa Director.	SNMP, Syslog, analytics
ha-state-change	Generated when HA state changes from master to slave or vice versa.	SNMP, Syslog, analytics
ha-sync-status	Generated after configuration sync happens between active and standby. Either sync error (or) sync ok are reported.	SNMP, Syslog, analytics

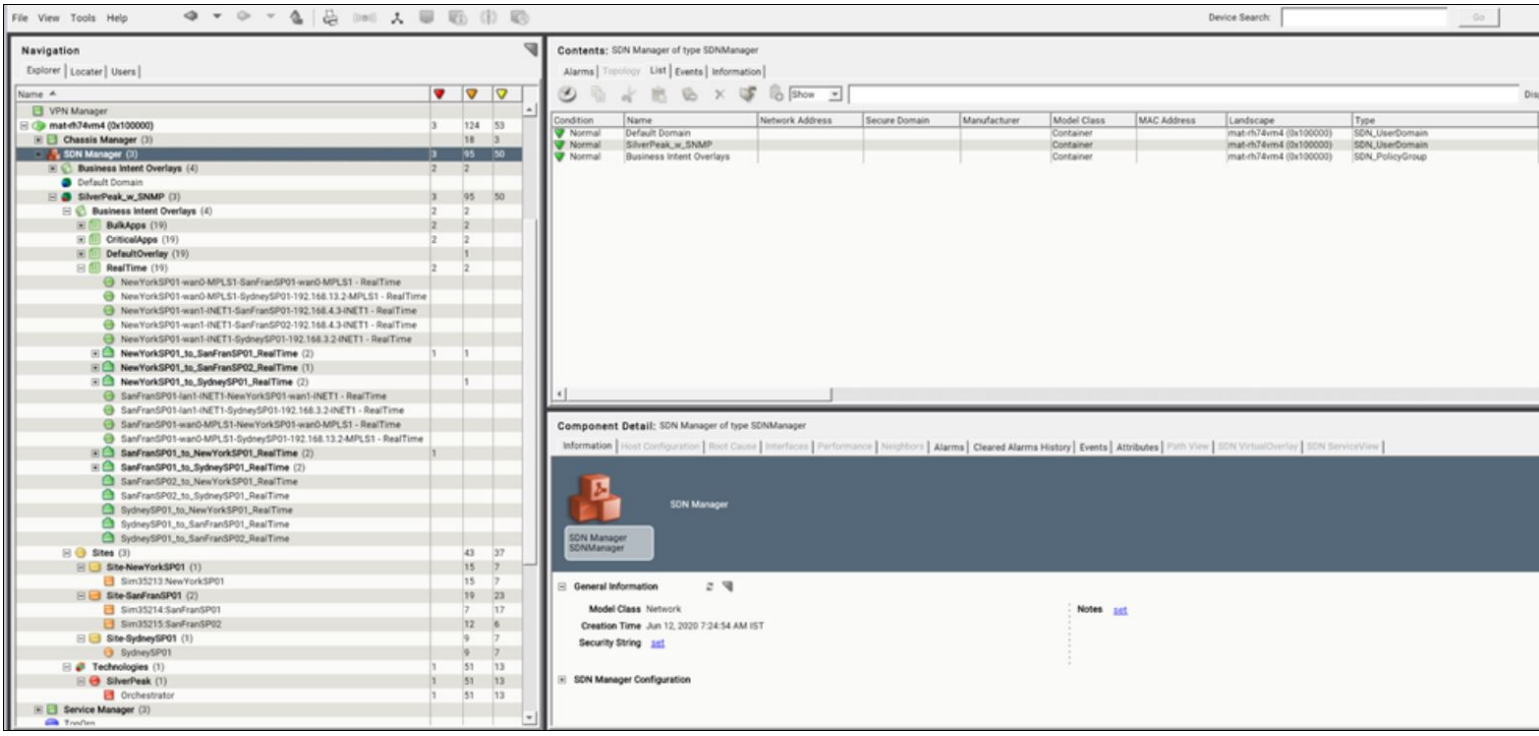
Monitoring SD-WAN for Silver Peak

DX NetOps Virtual Network Assurance (DX NetOps VNA) enables existing infrastructure management solutions to monitor software-defined networking (SDN) and network functions virtualization (NFV). DX VNA reduces the challenge and risk of SDN/NFV deployments by providing extended visibility and sustained reliability for self-service, automated networks. DX NetOps VNA collects data from SDN/NFV controllers and orchestrators and provides that data to subscribers. Each orchestrator or controller requires a plug-in to configure the connection.


lvntest006434 (0x1000000)
SDN Manager (3)
Business Intent Overlays (4)
Default Domain
SilverPeak (3)
Business Intent Overlays (4)
BulkApps (16)
NewYorkSP01_to_SanFranSP01_INET1-INET1 - BulkApps
NewYorkSP01_to_SanFranSP01_MPLS1-MPLS1 - BulkApps
NewYorkSP01_to_SanFranSP02_INET1-INET1 - BulkApps
NewYorkSP01_to_SydneySP01_INET1-INET1 - BulkApps
NewYorkSP01_to_SydneySP01_MPLS1-MPLS1 - BulkApps
SanFranSP01_to_NewYorkSP01_INET1-INET1 - BulkApps
SanFranSP01_to_NewYorkSP01_MPLS1-MPLS1 - BulkApps
SanFranSP01_to_SydneySP01_INET1-INET1 - BulkApps
SanFranSP01_to_SydneySP01_MPLS1-MPLS1 - BulkApps
SanFranSP02_to_NewYorkSP01_INET1-INET1 - BulkApps
SanFranSP02_to_SydneySP01_INET1-INET1 - BulkApps
SydneySP01_to_NewYorkSP01_INET1-INET1 - BulkApps
SydneySP01_to_NewYorkSP01_MPLS1-MPLS1 - BulkApps
SydneySP01_to_SanFranSP01_INET1-INET1 - BulkApps
SydneySP01_to_SanFranSP01_MPLS1-MPLS1 - BulkApps
SydneySP01_to_SanFranSP02_INET1-INET1 - BulkApps
CriticalApps (16)
DefaultOverlay (16)
RealTime (16)
Sites (3)
Site-NewYorkSP01 (1)
NewYorkSP01
Site-SanFranSP01 (2)
SanFranSP01
SanFranSP02
Site-SydneySP01 (1)
SydneySP01
Technologies (1)
SilverPeak (1)
Orchestrator

DX NetOps Spectrum offers the following monitoring capabilities for the Silver Peak SD-WAN:

Inventory:
Orchestrator, Sites, Routers (Edges), SLA Class/Profile, Interfaces, Tunnels, Application/SLA Paths.



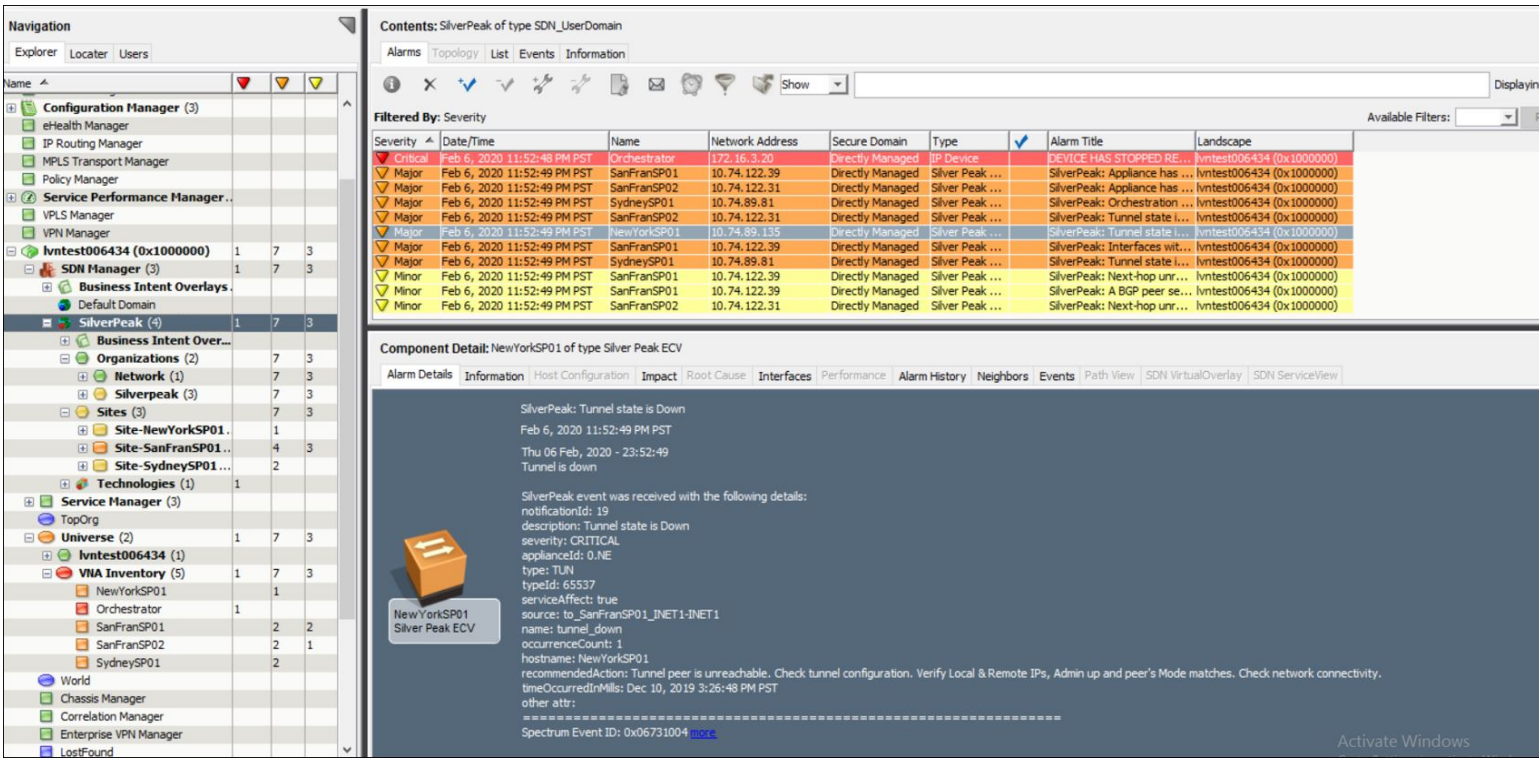
Performance

- Tunnels, SLA Path (Jitter, Latency, Packet Loss).
- SNMP statistics: interfaces, device health, UDP, TCP, SNMP, IP routing, IP/MAC address count, and IPv4 statistics.

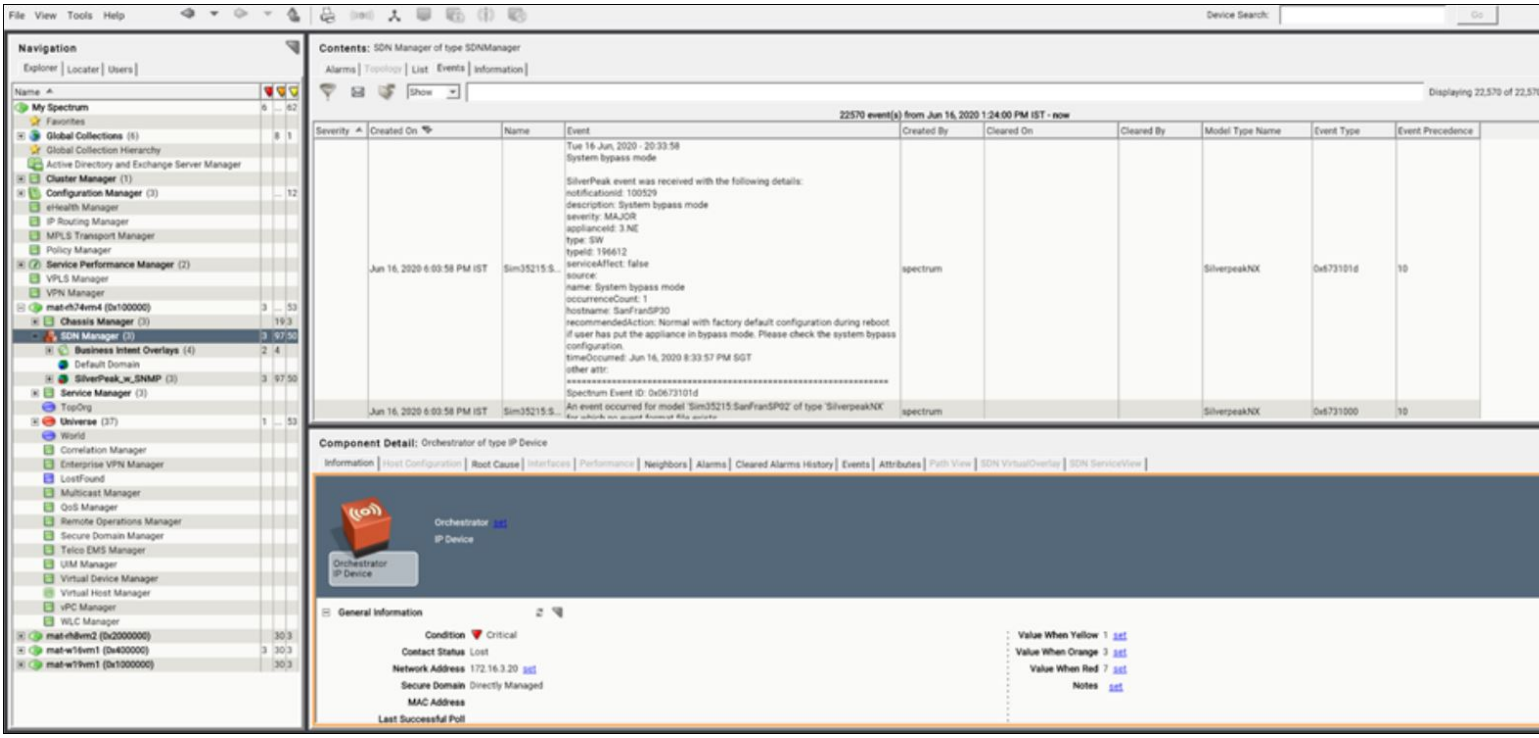
Alarms and Events

Supports over 200+ events for faster resolution of network issues.

Alarm View:



Events View:



Silver Peak SD-WAN Components

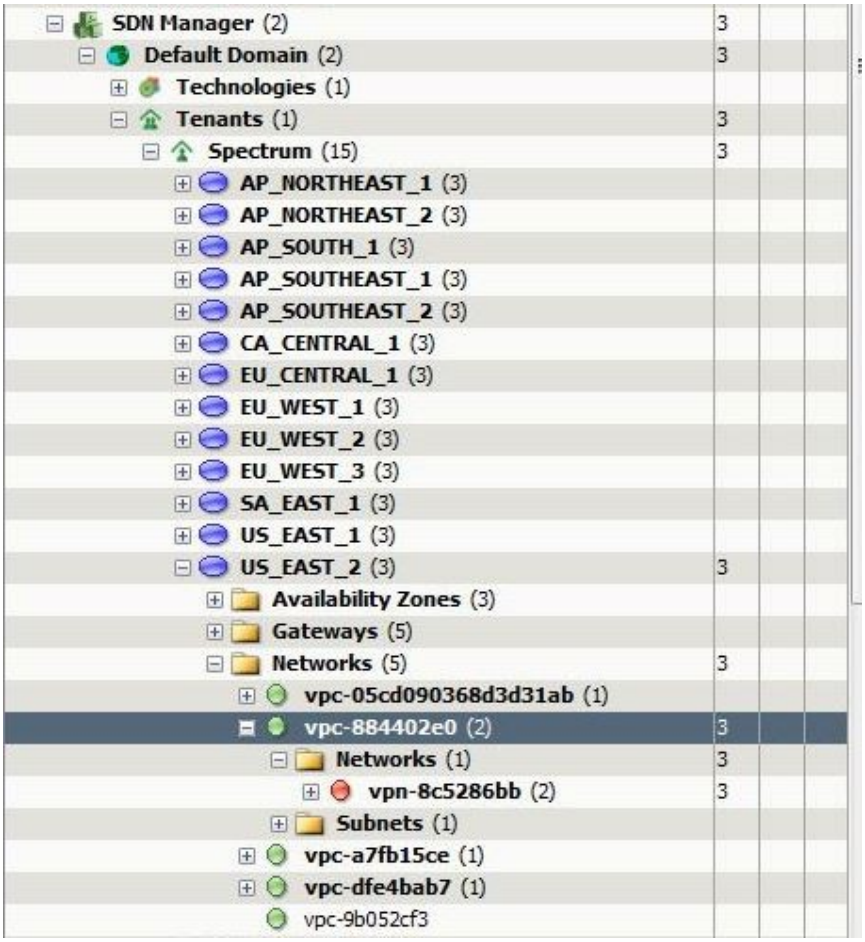
The Silver Peak SD-WAN has the following components.

- **Orchestrator:**
  Orchestrator was formerly called GMS (Global Management Software). An orchestrator is responsible for the management and configuration of Silver Peak EdgeConnect devices and NX/VX appliances. It centrally assigns policies to secure and control all WAN traffic. Orchestrator must connect to Cloud Portal for it to function.
- **EdgeConnect:**
  It can be either a physical or virtual appliances deployed in branch offices to create a secure, virtual network overlay. It is responsible for creating connections and must connect to Cloud Portal.

Support for AWS Network Monitoring

10.3.2 supports AWS Network Monitoring through the AWS VPN tunnel creation and direct connections. DX NetOps Spectrum monitors the tunnels on the VPN connection in the AWS Monitoring Network between the Virtual Private Gateway and the Customer Gateway through DX NetOps VNA notifications. The DX NetOps VNA and DX NetOps VNA integration fetches the following AWS entities inventory information in the given hierarchy and displays the same under the SDN Manager hierarchy in OneClick as shown in the screenshot below.

- Tenants
  - Regions
    - Availability Zones
    - Gateways
      - Internet Gateway
      - VPN Gatetway
      - Customer Gateway
      - NAT Gateway
    - Networks
      - VPC
      - Networks and VPN connection
    - Subnets
      - VMs



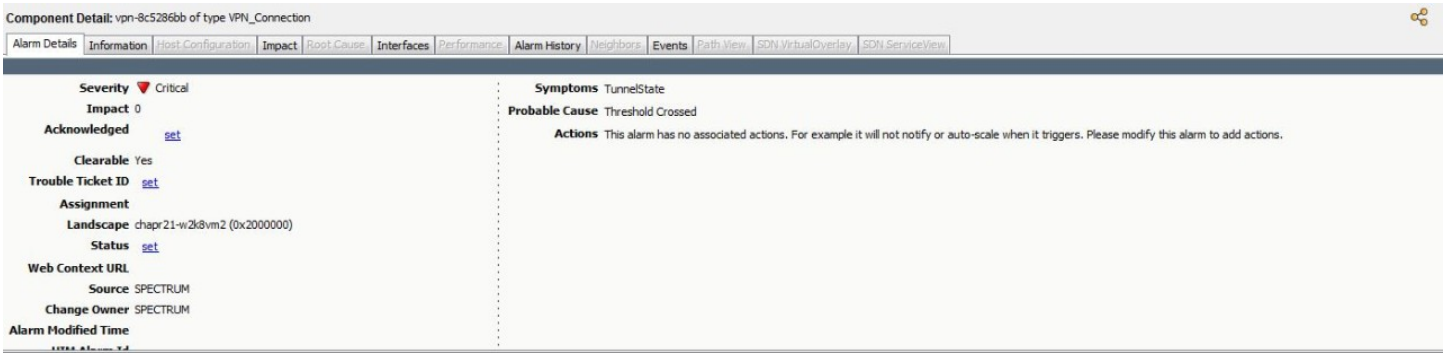
This integration supports AWS entities synchronization. When the DX NetOps Spectrum and DX NetOps VNA integration is enabled, synchronization happens automatically at the scheduled time interval. Additions, deletions, and modifications of AWS entities in a AWS environment are reflected in DX NetOps Spectrum.

Alarms Metrics

Following is the list of Alarm Metrics fetched from the AWS CloudWatch:

Alarm Metrics	Description	Value
TunnelState	Describes the state of the tunnel, for example, as shown below.	0 = down 1= up
TunnelDataIn	This is the bytes received through the VPN tunnel. Each metric data point represents the number of bytes received after the previous data point. The sum statistic to show the total number of bytes received during the period. The metric counts the data after decryption.	Bytes
TunnelDataOut	This is the bytes sent through the VPN tunnel. Each metric data point represents the number of bytes sent after the previous data point. The sum statistic to show the total number of bytes sent during the period. The metric counts the data before decryption.	Bytes

Here is an example of the TunnelState Alarm Metric details appearing in Spectrum, where the value is 0 signifying it is down.



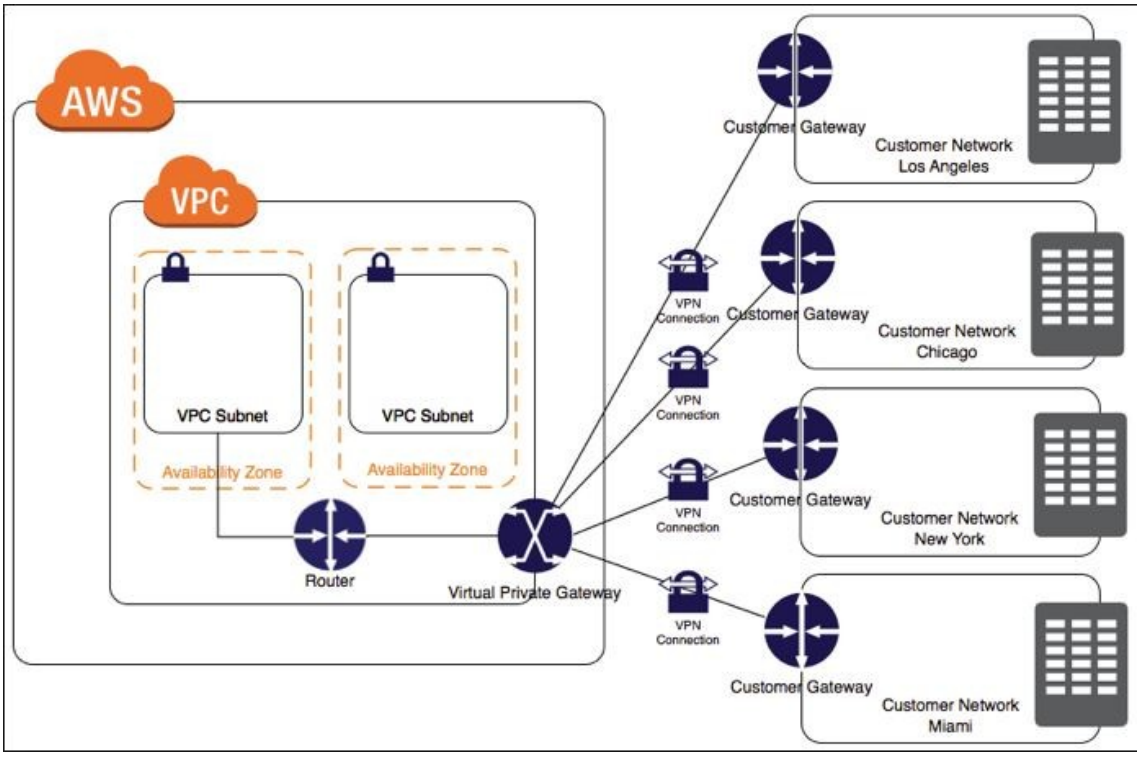
Fault Correlation

Fault correlation in the AWS Network Monitoring occurs between the VPN connection and the tunnels. If the VPN connection is down and the VPN tunnels are down, then the VPN connection is the root cause and the tunnel is the symptom (as shown in the screenshot above).

You can filter the AWS VPC VPN data using the following dimensions:

Dimension	Description
VpnId	The dimension filters the data by the VPN connection.
TunnelIPaddress	The dimension filters the data by the IP address of the tunnel for the virtual private gateway.

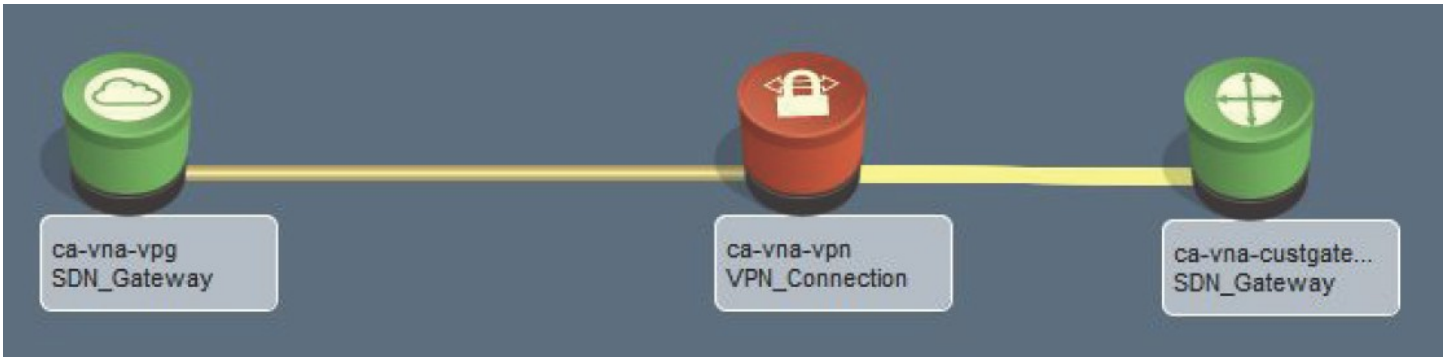
This is the AWS Network Monitoring diagram*



**This image is an intellectual property belonging to Amazon Web Services.*

AWS Monitoring Topology

The following screenshot displays the topology supported by DX NetOps Spectrum to monitor AWS network:



Icons

Following are the icons supported with this release:

Availability Zone

Availability zones (AZs) are isolated locations within data center regions from which public cloud services originate and operate.

VPN Gateway

Technology that provides network connectivity to AWS, is capable of acting as a router that intelligently forwards packets, manages availability between different network paths, and provides Virtual Private Network (VPN) or Software-Defined Wide Area Networking (SD WAN) services.

VPC

A virtual private cloud (VPC) is a virtual network dedicated to your AWS account. It is logically isolated from other virtual networks in the AWS Cloud. You can launch your AWS resources, such as Amazon EC2 instances, into your VPC. A VPC spans all the Availability Zones in the region. After creating a VPC, you can add one or more Subnets in each Availability Zone.

Customer Gateway

A customer gateway is an anchor on the user's side of the connection.

VPN Connection (down)

VPN is a managed client-based VPN service that enables you to securely access your AWS resources in your on-premises network.

Subnet

Subnet is a logical subdivision of an IP network. When you create a subnet, you specify the CIDR block for the subnet, which is a subset of the VPC CIDR block.

VM

Virtual Machine or 'instance'.

Reconciliation of models between VNA and UIM

During DX NetOps VNA data synchronization, when a new entity is created in AWS, it is reported to DX NetOps Spectrum. Spectrum performs a search to identify if this entity was modeled during DX NetOps Spectrum discovery and modeling or not, if it was, then the AWS data is reconciled through UIM, with the existing model, instead of creating a model.

Support for Cisco Meraki

Virtual Network Assurance and CA Spectrum 10.3.2 (or higher) support modeling of Cisco Meraki devices (Cloud Controller and Access Points) and proactive monitoring and managing of wireless networks. The monitoring information is provided by DX NetOps VNA, whereas, DX NetOps Spectrum relies on the inventory data from VNA to monitor and manage Meraki devices. The Access Points are modeled as certified SNMP devices, if they are SNMP- reachable, if not, then they are modeled as Access Point pingable models. To model the Access Points using SNMP, ensure the SNMP community string of the access points is added under the following:

- VNM→AutoDiscovery
- Control→Modeling
- Protocol Options-→SNMP Community Strings

Discovering connections between controller, switches and access points is based on the SNMP data available on the corresponding models, without which DX NetOps Spectrum cannot create any connections, and VNA cannot provide this connectivity information.

Properties of Cloud Controller and Access Points are displayed under the OneClick information tab. The operation status of Cloud Controller and Access Points is monitored based on the VNA inventory updates. Access Point availability is monitored through DX NetOps Spectrum and VNA inventory updates. If Access Point is unreachable from DX NetOps Spectrum (by Poll/Ping) and from VNA (OperationStatus is down), then a device unreachable alarm is asserted on the AP model.

Meraki Cloud controller dashboard exposes APIs to monitor its elements (Cloud Controller, Access Points, Service Appliances, Switches, Routers, and so on). VNA uses APIs to get monitoring data of DX NetOps Spectrum Controller and Access Points. DX NetOps Spectrum is integrated with VNA (using a Meraki plugin) that consume inventory information from VNA to model Meraki devices in the SDN/VNA manager hierarchy.

Following are the entities which are newly supported for Meraki under VNA:

- POLICY
- POLICY_GROUP
- SSID
- WIFI_CONTROLLER
- WIRELESS_NETWORK
- ACCESS_POINT
- MX_ROUTER

These entities are modeled in DX NetOps Spectrum with special model types. Meraki Cloud Controller is auto-modeled as a Wifi_Controller, which is a virtual entity.

The Cloud Controller properties are displayed under OneClick view, which contains:

- General Information, displaying the serial number, MAC, IP address, uptime, system name, location, condition/contact status, etc.
- Wireless Information, displaying the Access Points hardware details.

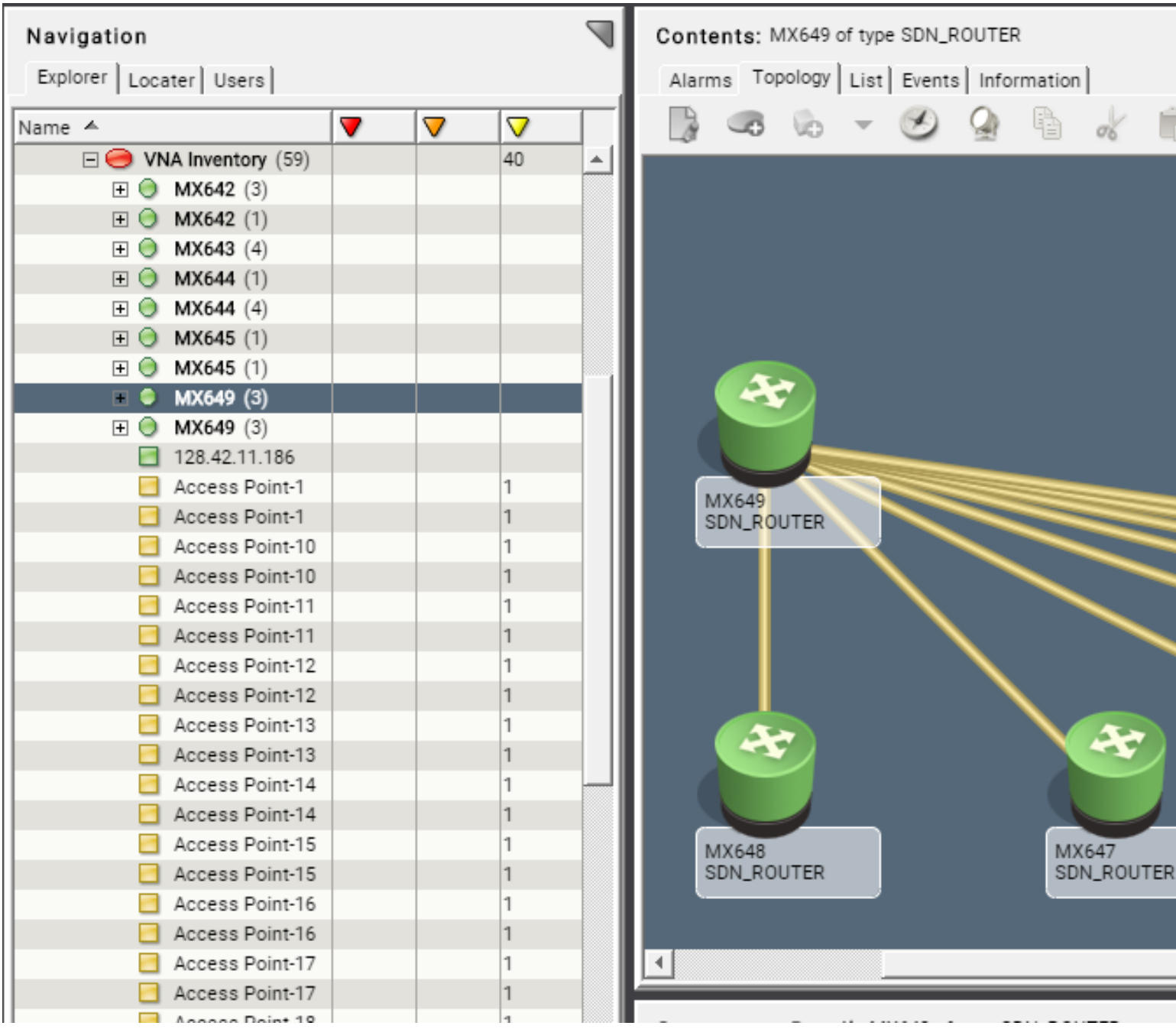
The Access Point properties that are displayed under the OneClick view are:

- General
- Asset Information

Hierarchy and Icons

Following is the hierarchy and icons that are supported with this release:

Organizations>Wireless Networks>Access Points, Appliances (MX Routers), and SSIDs



Organization

A collection of networks that are all part of a single organizational entity.

Access Points

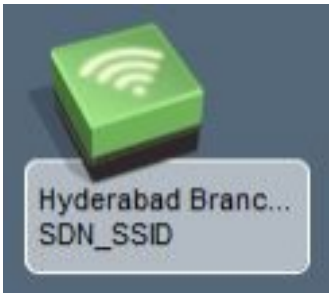
Access points provide deep network insight enabling smarter network management.

MX Router

MX Routers provide powerful routing, switching, security, and services features.

SSIDs

Service Set Identifier (SSID) is a unique identifier that is applied to the Access Point (AP) and the wireless client, allowing them to associate.



Cloud Controller

Cloud Controller is the control framework that allows various Cisco Meraki products to work together seamlessly, including indoor and outdoor wireless access points, switches, security appliances.

Events and Alarm

Events and alarms are raised on Meraki entities using the data that is provided by VNA. Following is the list of Events and Alarms identified:

MX Routers

Event Type	Description	Alarm Severity
events	VPN connectivity change	minor VPN_Connectivity=false
events	VPN connectivity change	Clear VPN_Connectivity=false
events	uplink connectivity change	critical Cellular Connection Down
events	uplink connectivity change	Major
events	uplink connectivity change	Major
events	uplink connectivity change	Clear Cellular Connection Down
events	client DHCP lease	NA
ids-alerts	ids signature matched	NA
ids-alerts	ids signature matched	NA
security_event ids_alerted	ids signature matched	NA
security_event security_filtering_file_scanned	Malicious file blocked by amp	Minor action="block"
security_event security_filtering_disposition_change	File issued retrospective malicious disposition	Clear Minor action="block"

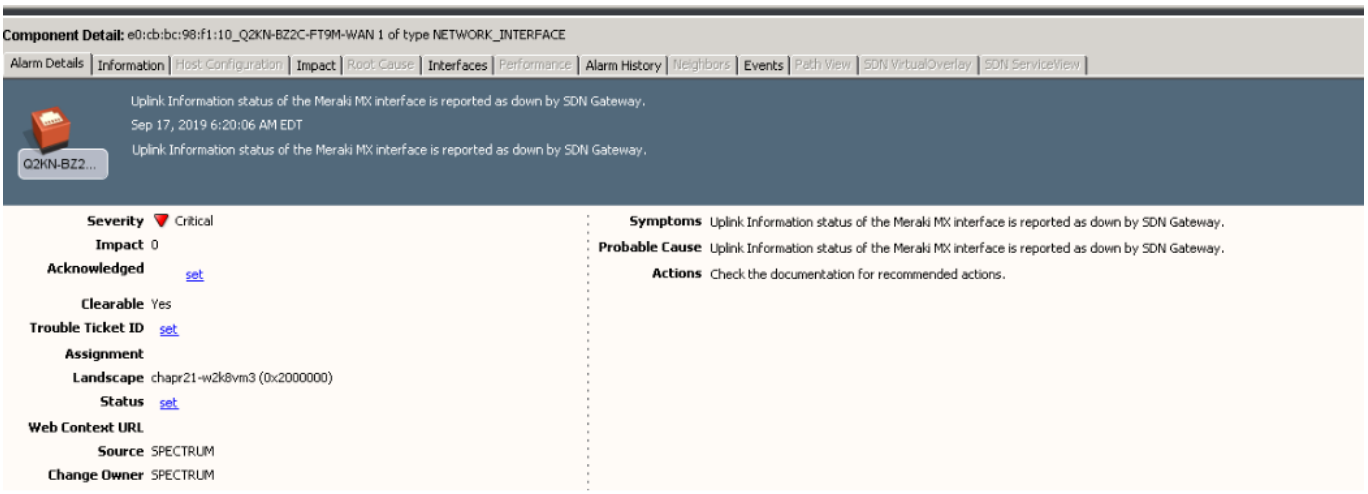
Meraki Dashboard API

The DX NetOps Virtual Network Assurance polls the Meraki Dashboard API. The following REST API returns mappings of above DX NetOps Spectrum alarm model fields to incident fields:

URL	GET /networks/{networkId}/devices/ {serial}/uplink

Response	<pre>[
	{ "interface": "WAN 1",
	"status": "Active",
	"ip": "1.2.3.4",
	"gateway": "1.2.3.5",
	"publicIp": "123.123.123.1",
	"dns": "8.8.8.8, 8.8.4.4",
	"usingStaticIp": false
	}
	]</pre>

In case the status of the uplink interface is down, the Virtual Network Assurance sends an update to DX NetOps Spectrum. Once DX NetOps Spectrum receives an update, it raises an alarm.



Access Points

Event Type	Event Description	Alarms
events	802.11 association	N/A
events	802.11 disassociation	N/A
events	WPA authentication	N/A
events	WPA deauthentication	N/A
events	WPA failed authentication attempt	N/A
events	802.1x failed authentication attempt	N/A
events	802.1x deauthentication	N/A
events	802.1x authentication	N/A
events	splash authentication	N/A
events	wireless packet flood detected	Minor
events	wireless packet flood end	Clear packet flood detected alarm

events	rogue SSID detected	Major

NOTE

There is no overlay topology for Meraki devices. The topology is formed based on the connections that are discovered as part of the regular/legacy SNMP discover connections action in DX NetOps Spectrum.

WARNING
Refer to SNMP Support for Cisco Meraki Solutions for a list of supported Meraki solutions, including wireless appliances and switches.

Monitoring SD-WAN for Viptela

Starting from the 10.3.1 release, DX NetOps Spectrum supports monitoring of Cisco Viptela devices through DX NetOps VNA integration. This functionality allows you to use the SD-WAN solution that is provided by Viptela. SD-WAN stands for Software-Defined Wide Area Networking. It is a combination of Software Defined Networking (SDN) and Wide Area Networking (WAN).

When DX NetOps Spectrum is integrated with DX NetOps VNA (configured with Viptela plug-in), DX NetOps Spectrum receives the inventory information of the Viptela devices through DX NetOps VNA.

NOTE

DX NetOps VNA configured with Viptela plug-in. acts as an SDN Gateway to collect Viptela inventory information and forwards information to DX NetOps Spectrum. Ensure that the DX NetOps VNA must be configured with Viptela plug-in.

The DX NetOps Spectrum and DX NetOps VNA integration fetches the following Viptela entity inventory information and displays under the SDN Manager hierarchy in OneClick.

Supported in 10.4.2 and later releases

- Sites

  NOTE
  Viptela sites map to site groups in the Performance Centre. If desired, you can manage your site groups in the Performance Centre to update a site name.
- vEdge router
- vEdge interfaces
- vManage
- vSmart
- Policy Group

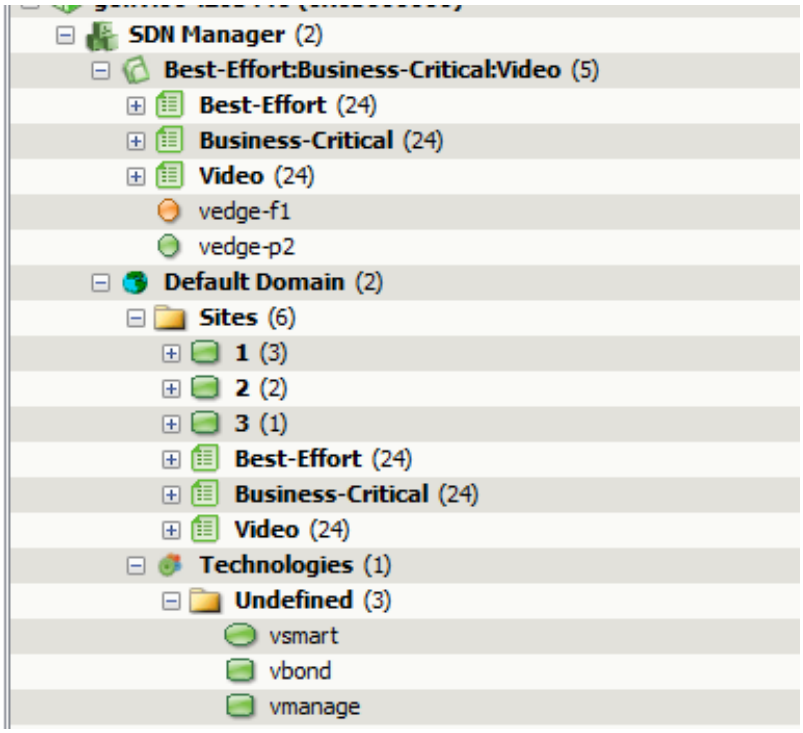
Supported in 10.4.2.2

- Tunnels
- Alarms and events raised on vEdge routers

  NOTE
  DX NetOps Spectrum consumes alarms raised on vEdge routers. For more information, see the Viptela documentation
- Application/SLA Paths
- cEdge router (16.x and 17.x)
- cEdge interfaces (16.x and 17.x)

  NOTE

  - cEdge Interface is available for NetOps 20.2.4 or higher versions.
  - Performance Stats are not supported for cEdge



This integration supports Viptela entity synchronization. When the DX NetOps Spectrum and DX NetOps VNA integration is enabled, synchronization happens automatically at the scheduled time interval that is displayed in the OneClick view. Additions, deletions, and modifications of Viptela entities in DX NetOps VNA are reflected in DX NetOps Spectrum after full synchronization.

Icons for Viptela Devices in Spectrum

The following are the new icons that are created for Viptela devices in DX NetOps Spectrum.

SDN Site

A site is a particular physical location within the Viptela overlay network. Each site is identified by a unique integer, called a site ID. Each Viptela device at a site is identified by the same site ID.

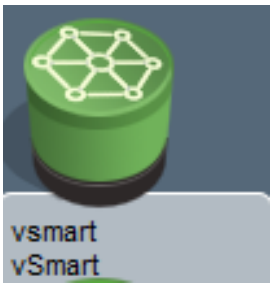
vManage

The vManage is a centralized network management system to maintain all Viptela devices and links in the overlay network. It is a fully manageable centralized portal to run and operate software-defined network (SD-WAN).

vBond

The vBond orchestrator is a software module that authenticates the vSmart controllers and the vEdge routers in the overlay network and coordinates connectivity between them. A Viptela overlay network can have one or more vBond orchestrators. It initiates the bring-up process of every vEdge device, at the first step it creates a secure tunnel with vEdge and informs vSmart and vManage about its parameters like for instance IP address. It has to be fully connected with every device.

vSmart



The vSmart controller establishes a secure DTLS connection to each vEdge router in the network and runs an Overlay Management Protocol (OMP) to share routes, security, and policy information.

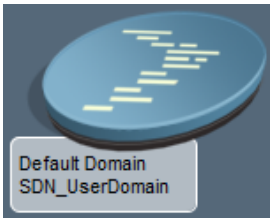
vEdge

vEdge is a router that receives complete control and data policies from the vSmart. It establishes secure IPSec tunnels with others vEdges depending on selected topology.

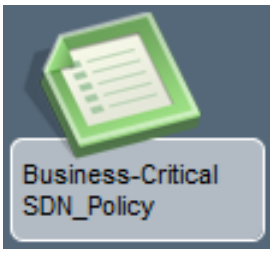
SLA Path

Prior to the 10.4.2.2 release, SLA Path used to have the "Topology tab" which displayed the Branch routers connected through Transports. From the 10.4.2.2 release, the same connections are shown under the SDN Overlay tab. SLA Path models are not displayed under the SDN Manager hierarchy. You can use the locater search to find SLA Path.

Default Domain



SDN Policy and Policy Group



SDN Transport

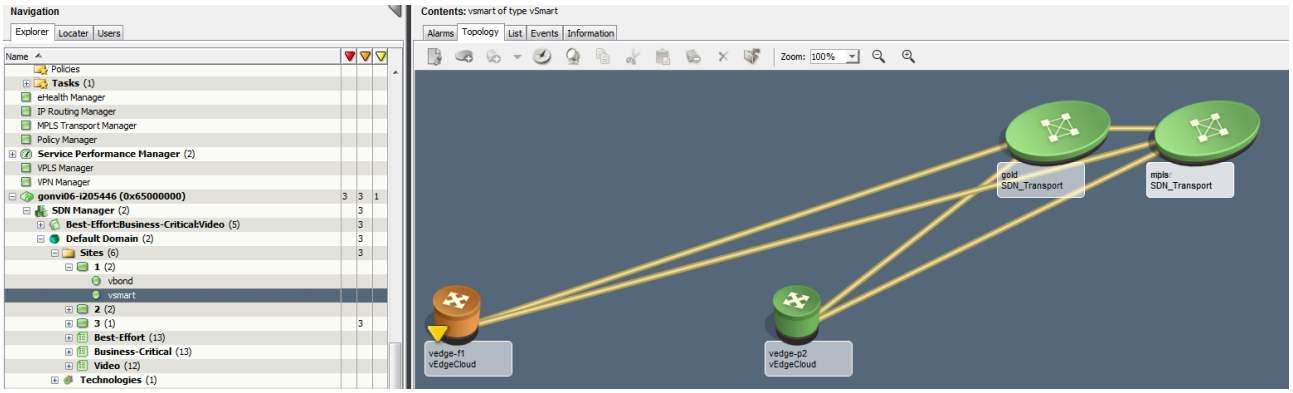
SDN Tunnel

Viptela Topology

DX NetOps Spectrum displays topology for Viptela devices. vSmart, vEdge, and SLA Paths.

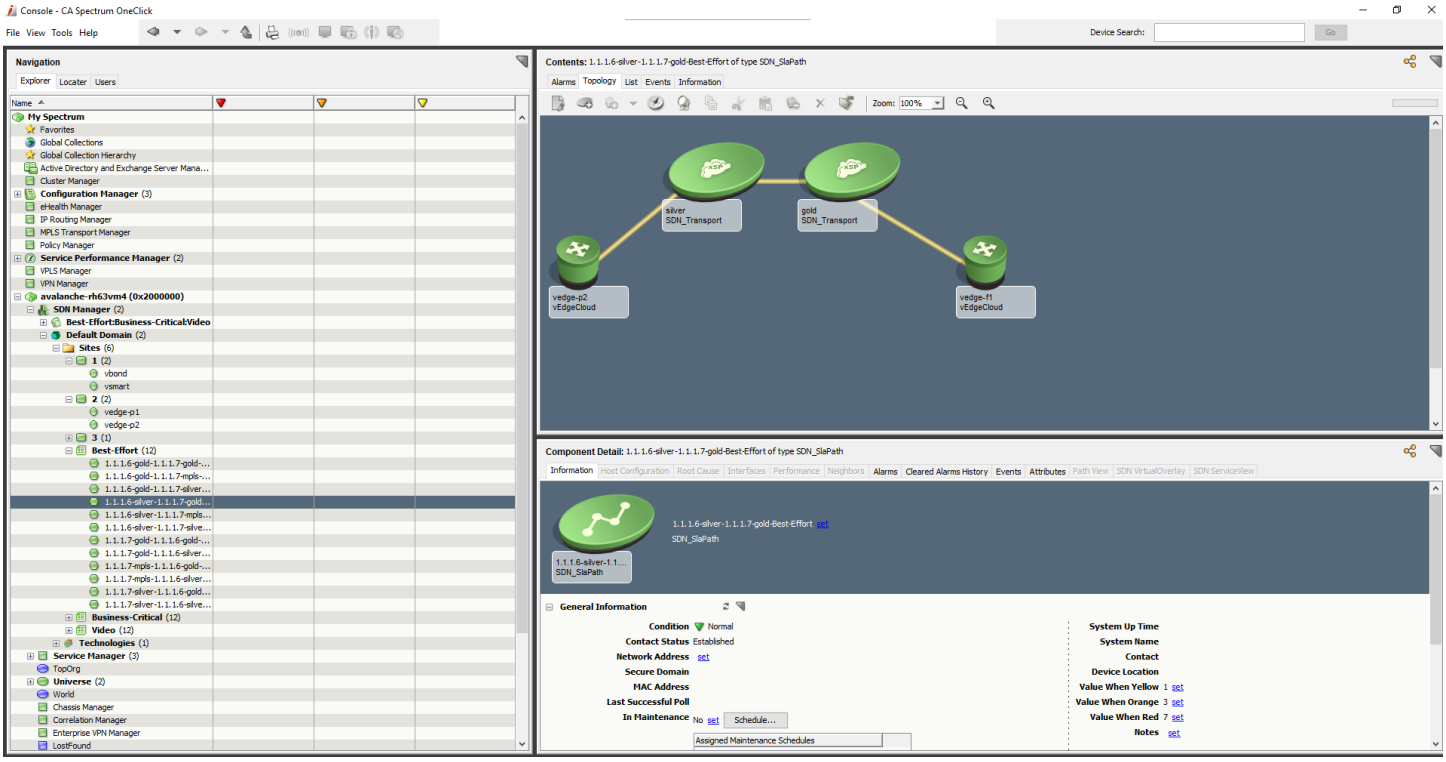
Topology for vSmart

DX NetOps Spectrum displays an Overlay Topology for vSmart, which shows vEdges associated with SDN Transports.



Topology for SLA Paths

The SLA Path topology that is displayed when an SLA path is selected under Policy Group in the Explorer tab. This topology shows the connectivity between vEdge router models, Transport models.



Topology for vEdges

The vEdge topology is displayed when a Policy is selected in the Policy Group under SDN Manager.

Support for Alarms Traps and Events

DX NetOps Spectrum can perform automatic discovery and mapping of a device interface and connections based on the following events and conditions:

- A change in the number of configured interfaces on a device
- When a device sends a LINK up a trap
- When DX NetOps VNA reconfigures a modeled device

Viptela Inventory

You can view the Viptela entities such as vEdge, vBond, vSmart, vManage in the OneClick console. Logical entities such as Sites and Policy Groups can also be viewed on the OneClick Navigation page. A new container '**VNA Inventory**' is created under the Universe view, this container has the Viptela entities. In the Explorer View, **SDN Manager** provides a more detailed hierarchy i.e. ***Domain > Sites > Policy Groups > Policy > vEdges*** , compared to the **VNA Inventory** view, which only displays the hierarchy of vEdges and associated Tunnels.

The vEdge devices are discovered and placed in the VNA Inventory folder under Universe in OneClick. A vEdge device must have a valid security certificate to participate in a Viptela network. You can configure the certificate state from the vManage Certificates administration page.

Viptela Alarms

The Alarms tab in OneClick displays the alarms for Viptela devices. The status of the alarms is fetched from VNA.

DX NetOps Spectrum and DX NetOps VNA integration for Viptela devices management support the alarms that are received from VNA. Alarms with following statuses are supported in DX NetOps Spectrum:

- Alarms with state as 'CREATED'
  When an alarm notification is received from VNA, an event and alarm is generated in DX NetOps Spectrum for the respective alarm with the event code 0x673000c
- Alarms with state as 'CLEARED'
  When an alarm notification is received from VNA, an event and alarm is generated in DX NetOps Spectrum for the respective alarm with the event code 0x673000d

The following Viptela alarms are supported in DX NetOps Spectrum:

Alarm	Description
BFD Between Sites Up	At least one BFD session on a vEdge router between two sites is in the Up state.
BFD Node Up	At least one BFD session for a vEdge router is in the Up state.
BFD Site Up	At least one BFD session on a vEdge router in a site is in the Up state.
BFD TLOC Up	At least one BFD session for a TLOC is in the Up state.
Control Node Up	At least one control connection for a vEdge router is in the Up state.
Control Site Up	At least one control connection from the vManage NMS and the vBond orchestrator in the site is in the Up state.
Control vSmart Up	At least one control connection from a vSmart controller in the overlay network is in the Up state.
Control TLOC Up	At least one control connection for a TLOC is in the Up state.
OMP vSmarts Up	At least one OMP connection from all vSmart controllers in the overlay network is in the Up state.
OMP Node Up	At least one OMP connection for a vEdge router is in the Up state.
OMP Site Up	At least one OMP connection to vSmart controllers from all nodes in the site is in the Up state.
Control vManage Up	At least one control connection from a vManage controller in the overlay network is in the Up state.

OSPF Router Up	All OSPF peering sessions from a particular OSPF router to all its OSPF peers on other vEdge routers are up.
Admin Password Change	The password for the admin that is changed on a router or controller.
Clear Installed Certificate	All certificates on a controller or device, including the public and private keys and the root certificate, have been cleared, and the device has returned to the factory-default state.
Cloned vEdge Detected	A duplicate router that has the same chassis and serial numbers and the same system IP address has been detected.
Cloud onRamp	The Cloud onRamp service was started on a router.
Control vManage Down	All control connections from a vManage NMS are in the Down state.
Default App List Update	The default application and application family lists, which are used in application-aware routing policy, have changed.
Interface Admin State Change	The administrative status of an interface in a controller or router that is changed from up to down (Critical) or down to up (Medium).
Memory Usage	The memory usage on a controller or device has reached a critical level that could impair or shut down functionality, or a medium level that could impair functionality.
New CSR Generated	A controller or router generated a certificate signing request (CSR).

Correlation of DX NetOps Spectrum Alarms to Viptela Alarms

DX NetOps Spectrum tries to correlate alarms that are received from Viptela and creates a root cause alarm in Spectrum. For example, if there is a physical interface down in Viptela, Spectrum receives multiple alarms for different events. DX NetOps Spectrum correlates these alarms from Viptela and creates a Bad Link Detected alarm in Spectrum. Alarms that are received from Viptela are shown as symptomatic alarms and the Spectrum alarm is considered as the root cause.

DX NetOps Spectrum keeps updating/replacing the symptomatic alarms with the latest alarms that are received from Viptela. The root cause alarm remains the same (bad link detected) in Spectrum until all the alarms are cleared from Viptela.

Alarms for SLA Violation

Whenever there is a violation of the defined SLA, DX NetOps Spectrum generates a critical alarm on the vEdge routers for the SLA. The SLA violation alarms in Spectrum helps you to identify the bad policies (which resulted in too many policy violations) and take corrective actions.

Reconciling Viptela Entity Data in DX NetOps Spectrum

During DX NetOps VNA data synchronization, when a new Viptela entity is created in DX NetOps VNA is reported to DX NetOps Spectrum. Spectrum performs a search to identify if this entity was modeled during DX NetOps Spectrum discovery and modeling. If such an existing model is found, DX NetOps Spectrum reconciles the CA Viptela entity information with the existing model, instead of creating a model.

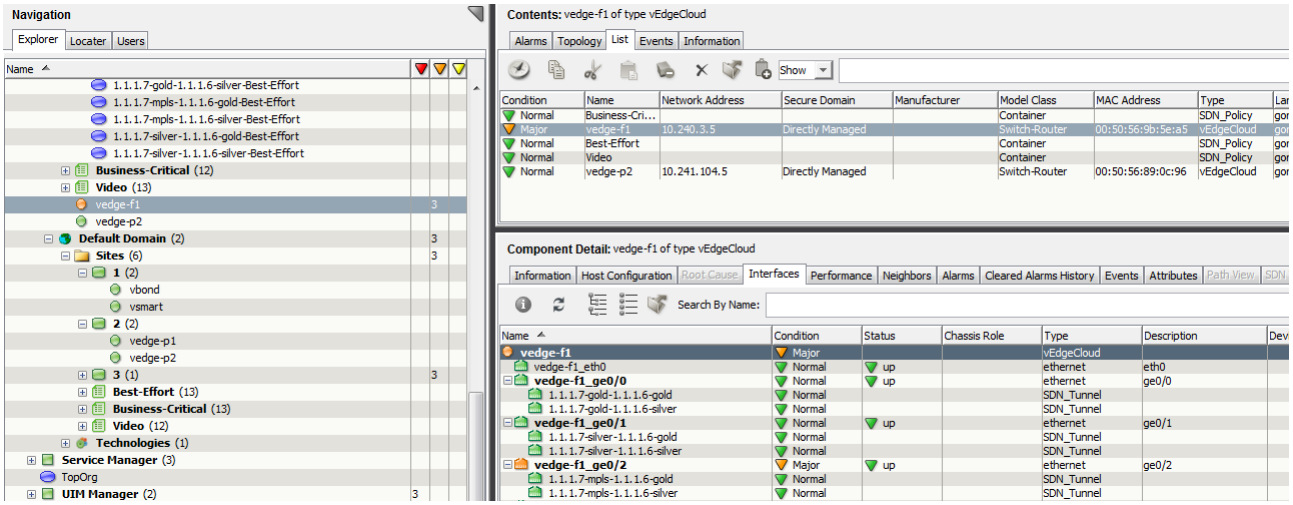
Interfaces Information

DX NetOps Spectrum reads the information of the following Viptela devices from VNA and displays in the OneClick Interfaces tab:

- Tunnels
- Physical Interfaces
- Transport

SDN Tunnels associated with vEdge Interfaces

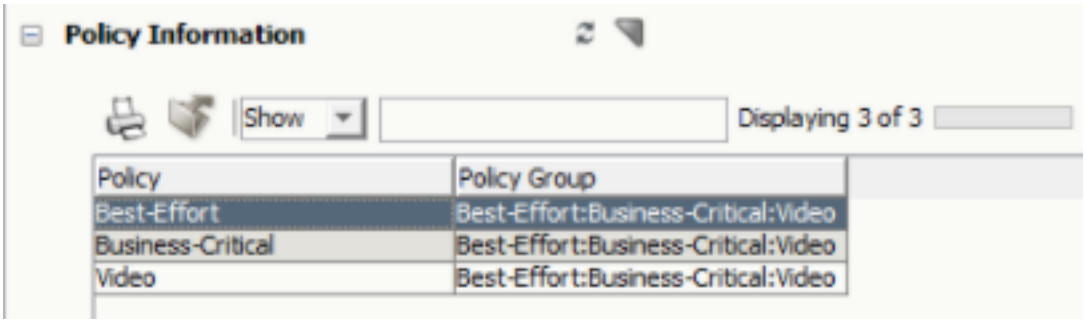
The Interfaces tab in the Component Details panel shows all the tunnels information which is associated with the selected vEdge.



Policies Information

For vEdges, associated policy and policy group information is displayed in the Information tab of the contents panel.

vEdge Policy Information



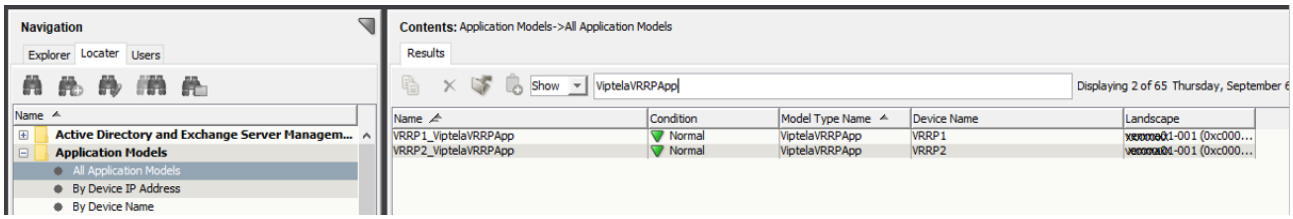
Viptela VRRP Models

Using the search functionality in the Locater tab to find Virtual Router Redundancy Protocol (VRRP) enabled Viptela devices that are available in the DX NetOps Spectrum environment. You can access the Locater search from the Locater tab of the Navigation Panel. The Search results appear in the Results tab of the Contents panel.

Change the State of the ViptelaVRRPMode Attribute

To search VRRP models, follow these steps:

1. Open the DX NetOps Spectrum OneClick Console.
2. From the Navigation Panel, select the Locater tab.
   The Search Options window opens.
3. Expand Application Models, All Application Models.
   The Locater Search results are displayed in the Contents pane.
4. In the Results tab, filter for the ViptelaVRRPApp model.
   The VRRP enabled Viptela devices are displayed.



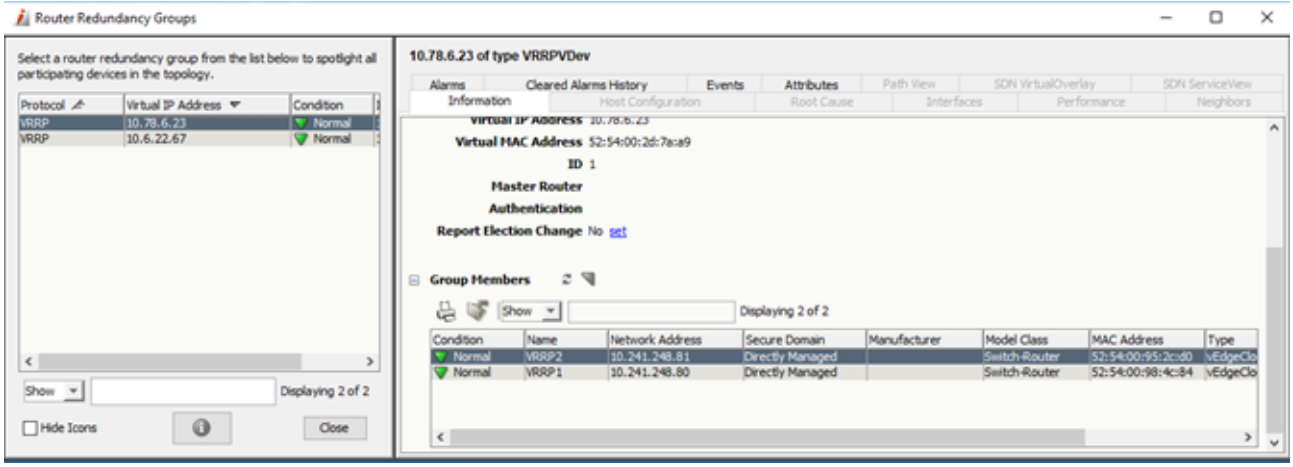
5. Select an App model from the list
6. In the Component Detail panel, select the Attributes tab then search for the ViptelaVRRPMode attribute.
7. Set the attribute value to Active. The default value is Off.
   Repeat the same for remaining similar App models.

Spotlighting Viptela Devices

Spotlighting Viptela Devices in the Topology view helps you isolate and visualize the Viptela model relationships within your network. Use the OneClick Spotlight feature to see all Access Points that are related to a Viptela in the Topology view.

To spotlight Viptela devices, follow these steps:

1. Open OneClick.
2. Expand the desired landscape on the Explorer tab and select Universe.
   Details about the selected Universe appear in the Contents panel.
3. Select the Topology tab.
   The topology of the Universe is displayed.
4. Select the Spotlight View and select Router Redundancy from the list of options.
   The list of Router Redundancy Groups appears.
5. Select a Router Redundancy group from the list to spotlight all participating devices in the topology.



The Group Members view shows the list of devices for the selected Router Redundancy Group.

Network Configuration Manager (NCM) Support for Viptela Devices

After the DX NetOps Spectrum and DX NetOps VNA integration is enabled, during the DX NetOps Spectrum discovery a folder for Viptela is created under the Device Families folder of Configuration Manager. Network Configuration Manager automatically assigns all Viptela devices to the Viptela family.

The Configuration Management process helps in identifying and monitoring configurations of single devices and device families that comprise a network. Using the Network Configuration Manager, you can perform the following tasks for Viptela devices:

- Manage configurations for Viptela devices that are modeled in DX NetOps Spectrum.
- Capture device configurations and store them in the DX NetOps Spectrum database.

Configure Device Family General Settings

The General Configuration subview contains the Configuration Manager settings. Configuration Manager lets you disable tasks for an entire device family. When Configuration Manager is disabled, Network Configuration Manager operations are not performed on any of the devices that are contained by this device family.

Configure Device Family Communication Mode

The Primary Communication Mode determines how Network Configuration Manager interacts with the associated devices. By default, Viptela device family supports 'SSH' communication mode. You need to specify the device username, password, and enable password to contact the devices.

Manually Capture the Viptela Device Configuration

You can manually capture the Viptela device configurations in OneClick. Capturing the device configurations helps to see that any changes occurred in device configurations and helps to store the updated configuration in the database.

Follow these steps:

1. Select the Viptela device in the Explorer tab.
   The device appears in the List tab of the Contents panel.
2. Select the Host Configuration tab in the Component Detail panel.
3. Select the Capture Configuration for Selected Host icon.
   The results of the capture appear. Either a new configuration appears in the list or the last verified time is updated for the current configuration.

Upload Configuration

You can manually upload a configuration file to a Viptela device on your network. When you upload a configuration file, you merge it into the existing configuration file.

Follow these steps:

1. Select the Viptela device in the Explorer tab.
   The device appears in the List tab of the Contents panel.
2. Select a configuration from the Host Configuration tab in the Component Detail panel.
3. Select the Upload Configuration to the Selected Host icon.
   The Upload Configuration window opens.
4. Modify the configuration and select update
5. Perform any of the following optional steps:
   – Edit configuration content as desired.
   – Enter the criteria in the Search field to locate specific lines in the configuration file to change content or to verify content prior to upload.
   – Select Open to import a previously exported configuration file that is saved locally on your system.
   – Select Save As if you want to save and export this configuration file in txt or HTML format.
6. Select Upload to upload the configuration file to the selected device.
   The message “The configuration upload succeeded” appears when the procedure is complete.

   NOTE

   To compare configuration differences, capture the newly uploaded configuration.

Known Anomalies

DX NetOps Spectrum has following known anomalies for supporting Viptela devices:

1. SDN Tunnels do not show the connected device and port unless the vEdge model is reconfigured.
2. SDN Tunnels getting deleted when the physical interface goes down.
3. Overlay topology shows the overlapped transports.

Monitoring SD-WAN for VeloCloud

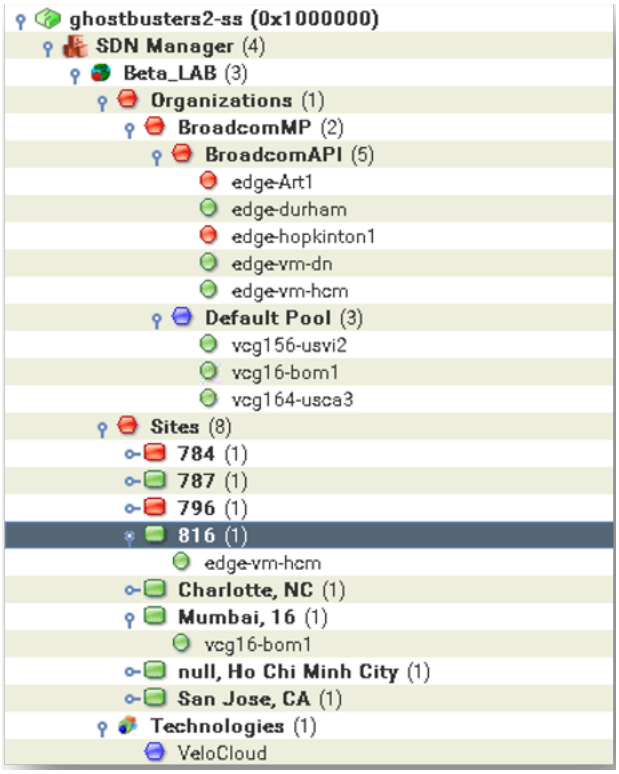
From the 10.4.2.2 / 20.2.5 release, DX NetOps Spectrum supports the monitoring of VeloCloud devices through DX NetOps VNA integration. This functionality allows you to use the SD-WAN solution that is provided by VeloCloud. SD-WAN stands for Software-Defined Wide Area Networking. It is a combination of Software Defined Networking (SDN) and Wide Area Networking (WAN).

When DX NetOps Spectrum is integrated with DX NetOps VNA (configured with VeloCloud plug-in), DX NetOps Spectrum receives the inventory information of the VeloCloud devices through DX NetOps VNA.

> **NOTE**
> DX NetOps VNA configured with VeloCloud plug-in. acts as an SDN Gateway to collect VeloCloud inventory information and forwards information to DX NetOps Spectrum. Ensure that the DX NetOps VNA must be configured with VeloCloud plug-in.

The DX NetOps Spectrum and DX NetOps VNA integration fetches the following VeloCloud entity inventory information and displays under the SDN Manager hierarchy in OneClick:

- Enterprise Proxy
- Enterprise
- Gateway Pool
- Gateway (VCG)
- Site
- Router (VCE)

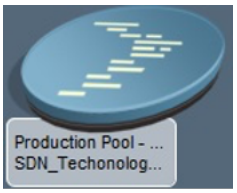


This integration supports VeloCloud entity synchronization. When the DX NetOps Spectrum and DX NetOps VNA integration is enabled, synchronization happens automatically at the scheduled time interval that is displayed in the OneClick view. Additions, deletions, and modifications of VeloCloud entities in DX NetOps VNA are reflected in DX NetOps Spectrum after full synchronization.

Icons for VeloCloud Devices in DX NetOps Spectrum

The following are the icons that are created for VeloCloud devices in DX NetOps Spectrum.

SDN Technology Group



SDN Site

SDN Gateway

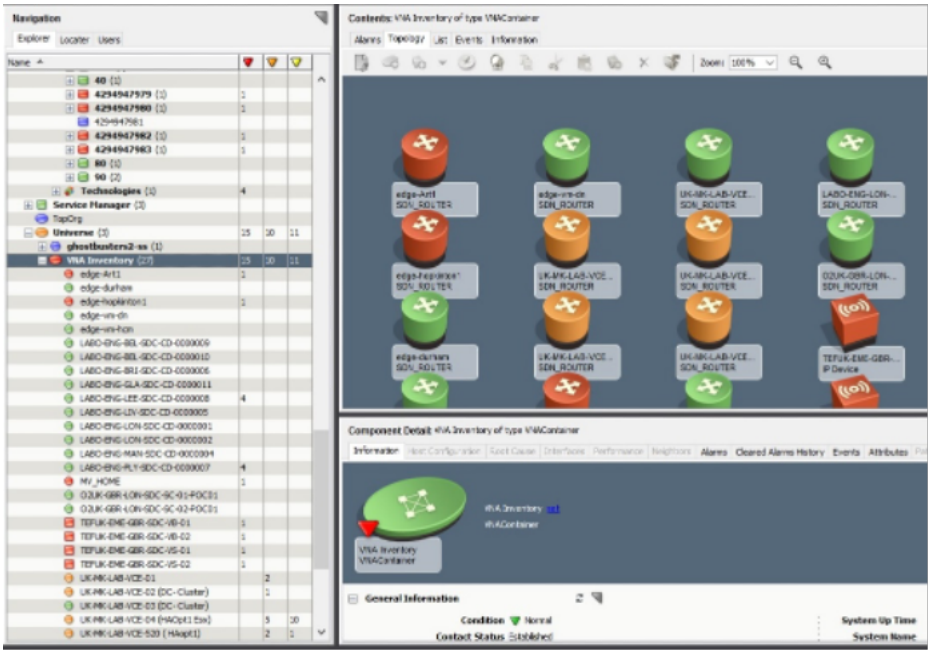
SDN Router

VeloCloud Edge 5X0

VeloCloud Inventory

You can view the VeloCloud entities such as Router, Gateway in the OneClick console. The other entities - Sites, Enterprise Proxy, Enterprise, and Gateway Pool can also be viewed on the OneClick Navigation page.

The Router devices are discovered and placed in the VNA Inventory folder under Universe in OneClick.



VeloCloud Alarms

The Alarms tab in OneClick displays the alarms for VeloCloud devices. The status of the alarms is fetched from DX NetOps VNA. DX NetOps Spectrum and DX NetOps VNA integration for VeloCloud devices management support the alarms that are received from DX NetOps VNA. Alarms with the following statuses are supported in DX NetOps Spectrum:

Alarm	Description
EDGE_DOWN	Edge down
EDGE_UP	Edge up
LINK_DOWN	Link down
LINK_UP	Link up
VPN_TUNNEL_DOWN	VPN tunnel down
EDGE_HA_FAILOVER	HA failed
EDGE_SERVICE_DOWN	Edge service down
GATEWAY_SERVICE_DOWN	Gateway service down
VNF_VM_EVENT	VNF VM event
VNF_VM_DEPLOYED	VNF VM deployed
VNF_VM_POWERED_ON	VNF VM powered on
VNF_VM_POWERED_OFF	VNF VM powered off
VNF_VM_DEPLOYED_AND_POWERED_OFF	VNF VM deployed and powered off
VNF_VM_DELETED	VNF VM deleted
VNF_VM_ERROR	VNF VM error
VNF_INSERTION_EVENT	VNF Insertion event
VNF_INSERTION_ENABLED	VNF Insertion enabled
VNF_INSERTION_DISABLED	VNF Insertion disabled
TEST_ALERT	Test alert
EDGE_CSS_TUNNEL_DOWN	Edge CSS Tunnel down
EDGE_CSS_TUNNEL_UP	Edge CSS Tunnel up
VNF_IMAGE_DOWNLOAD_EVENT	VNF Image download event
VNF_IMAGE_DOWNLOAD_IN_PROGRESS	VNF Image download in progress
VNF_IMAGE_DOWNLOAD_COMPLETED	VNF Image download completed
VNF_IMAGE_DOWNLOAD_FAILED	VNF Image download failed

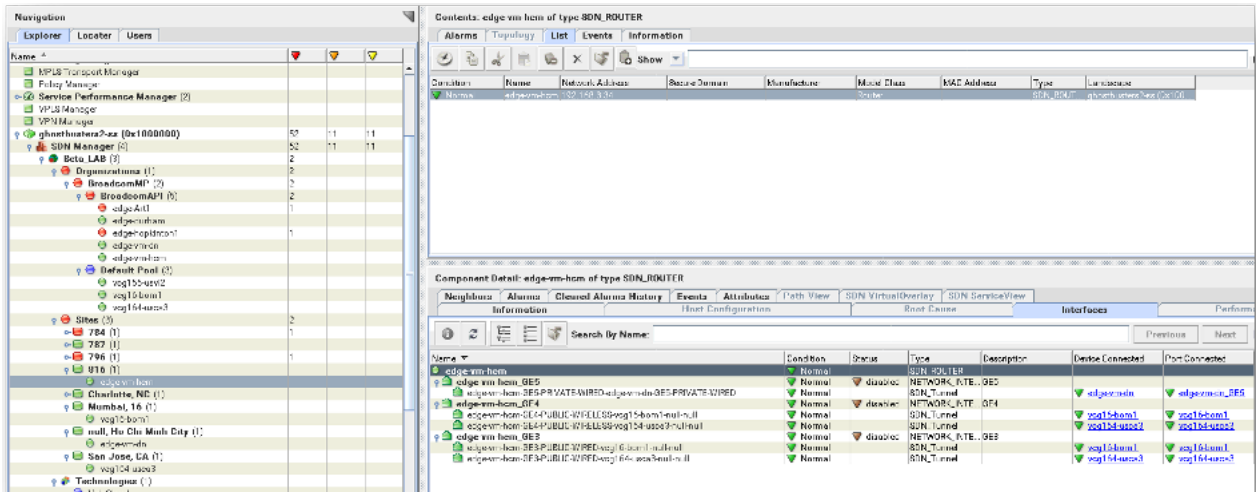
VeloCloud Interface Information

DX NetOps Spectrum reads the information of the following VeloCloud devices from VNA and displays in the OneClick Interfaces tab:

- Physical Interfaces
- Tunnels
- Transport

SDN Tunnels Associated with Router Interfaces

The Interfaces tab in the Component Details panel shows all the tunnels information which is associated with the selected Router.



VMware vSphere Support

DX NetOps Spectrum introduces support for vSphere virtualization through CA VNA 3.5.1, which provides better insights resulting in improved performance. Starting from the 10.2.3 release, the Vmware vSphere plugin collects inventory for the following from the vCenter server:

- Data centers
- Clusters
- Hosts
- Virtual machines
- Host interfaces
- Virtual machine interfaces
- vSphere distributed switches (Created by the Cisco ACI Virtual Machine Manager Domain integration with VMware vSphere)

NOTE

There is no connectivity between VMs and vSwitch/DVS.

The vSphere events in and their details are as follows:

Event	Event Description	Event Code
DrsVmMigratedEvent	This event records a virtual machine migration that was recommended by Distributed Resource Scheduler (DRS)	0x6730162
VmMigratedEvent	This event records a virtual machine migration.	0x6730161
VmFailedMigrateEvent	This event records a failure to migrate a virtual machine.	0x6730160
VmBeingHotMigratedEvent	This event records that a virtual machine is being hot-migrated.	0x673015f
VmBeingMigratedEvent	This event records that a virtual machine is being migrated.	0x673015e

NOTE

The default event is 0x6730011.

Disable the DX NetOps VNA Integration

You can disable the DX NetOps Spectrum and DX NetOps VNA Integration from the OneClick Administration page.

NOTE

You must disable DX NetOps VNA integration from the web server on which the integration is enabled.

Follow these steps:

1. Open the OneClick Administration page.
   The OneClick Administration page opens.
2. Click the Administration tab.
   Links to various OneClick web server configuration pages are displayed.
3. Click the VNA Integration Configuration link in the left panel.
4. To disable the DX NetOps VNA integration, select Disable and click Save.
   The Integration disabled message appears.

NOTE

Wait for the all the DX NetOps VNA Host server models and the folder hierarchy to be cleared from the DX NetOps VNA in the OneClick view after disabling the integration. To validate, search for any DX NetOps VNA Server models, using the search option.

Co-Existence of DX NetOps VNA and Unified Infrastructure Management (UIM) Integration

Overview

DX NetOps Spectrum supports the co-existence of UIM and DX NetOps VNA integration together for the VMware inventory. This enhancement allows you to reconcile vCenter, ESX Host, and Virtual machines as SNMP and Pingable models.

Benefits for this co-existence include:

- Reconciliation of UIM and DX NetOps VNA Host Servers
- Fault Correlation between UIM and DX NetOps VNA alarms
- OneClick Views for reconciled Events & Alarms

Previously, when you integrated DX NetOps VNA to monitor VMware entities in your VMware environment, and if you also integrated UIM Virtualization for the same environment, duplicate entries were created in DX NetOps Spectrum. Starting Spectrum 10.3, you can reconcile the same set of data from different sources. The UIM models are reconciled with VNA models and these results are shown in the Explorer tab.

NOTE

To synchronize UIM alarms with DX NetOps Spectrum, the VM names of UIM should match with the VM names of DX NetOps Spectrum.

VMware Entities Reconciliation

After DX NetOps Spectrum is integrated with UIM and DX NetOps VNA, then the reconciliation of the models occurs, which are part of both UIM and DX NetOps VNA. The reconciled models are moved from Universe to DX NetOps VNA Inventory Container.

For ESX Hosts and VMs, new icons are created during the reconciliation from UIM to VNA.

NOTE

When DX NetOps VNA Integration is disabled, the models which were discovered and reconciled, get deleted as part of the inventory clean-up.

Based on the following scenarios, the reconciliation process has different results.

Integrate UIM first and DX NetOps VNA next

When you integrate UIM first then integrate DX NetOps VNA, you observe the following results:

- ESX Hosts are reconciled and the attributes are auto-filled.
  When you integrate UIM, then ESX Host models get created in DX NetOps Spectrum. After you integrate DX NetOps VNA, reconciliation happens and these ESX Host models will get deleted and recreated under the VNA Inventory container. The VMs under ESX Hosts are moved from the UIM Inventory container to the VNA Inventory container.
- The DataCenter is created in both UIM and VNA inventories and Model Handle is the same in both the inventories.
- In the UIM Manager hierarchy, the UIM vCenter name is changed, to the name of the VNA vCenter.
- Reconciliation of existing UIM logical entities with DX NetOps VNA.
  The DataCenter and Cluster should get reconciled and the UIM models get VNA attributes.
  The following attributes are filled during reconciliation of existed UIM Data Center and Cluster to VNA Cluster and Data Center, ESX Host Server:
  – UIMOrgin
  – VirtulaEntityUniqueId

The following attributes are filled for SDN_VM's when VNA Integration is enabled.
SDNUIM_DATA_SOURCE_CONTEXT
SDNUIM_DESCRIPTION
SDNUIM_INSTANCE_ID
SDNUIM_IP_ADDRESS_LIST
SDNUIM_MAC_ADDRESS_LIST
SDNUIM_MEMORYINGB
SDNUIM_NUMBEROFCORES
SDNUIM_OSTYPE
SDNUIM_OSVERSION
SDNUIM_SOURCEPRODUCTNAME
SDNUIM_BIT_RATE
SDNUIM_STORAGEINGB
SDNUIM_TOPOUPDATE_ACTIVE
SDNUIM_VIRTUALIZATION_ENV
SDNUIM_VM_ID
SDNUIM_VM_NAME
SDNUIM_VNF_NAME

When DX NetOps VNA integration is disabled

When UIM first and DX NetOps VNA next integrated and if you disable the VNA Integration, you observe the following results:

- The inventory gets deleted including the ‘VNA Inventory’ container and the inventory under the ‘SDN Manager’.
- Deletes all reconciled models under the ‘UIM Manager’ and the ‘UIM Inventory’ containers.

Integrate DX NetOps VNA first and UIM next

When you integrate DX NetOps VNA first then integrate UIM, you observe the following results:

- Virtual Entity Unique ID is the as same under DX NetOps VNA and UIM
- UIM inventory of type ‘Cluster’ shows Modeltype_Name as ‘SDN_Cluster’. This entity_type matched with the same in DX NetOps VNA. The DataCenter under UIM and DX NetOps VNA shows Modeltype_Name as SDN_Cluster.
- vCenter name is changed according to the vCenter name in UIM inventory
- The ESX Hosts moves out of UIM Inventory (Universe -> UIM inventory) and displayed directly under the Universe.

When UIM integration is disabled

When both UIM and DX NetOps VNA are integrated, and if you disable the UIM integration, you observe the following results:

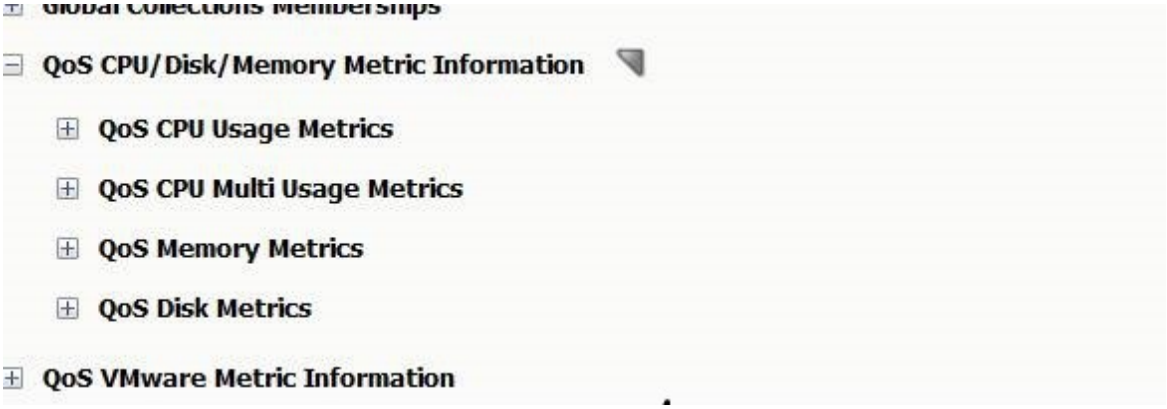
- The UIM inventory is cleaned up from the UIM Inventory folder.
- The UIMEventAdmin folder gets deleted.

OneClick Views

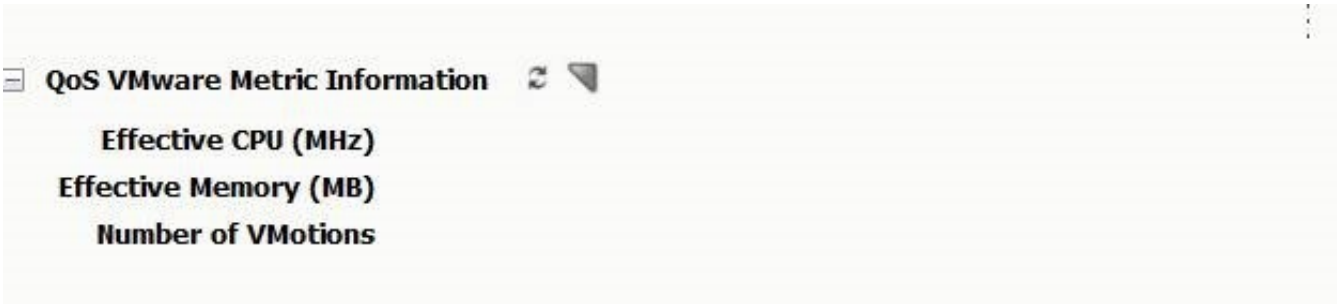
When DX NetOps VNA integrated first and UIM next, OneClick views get reconciled from UIM to DX NetOps VNA on VNA models.

> **NOTE**
>
> No views are reconciled for a Data Center.

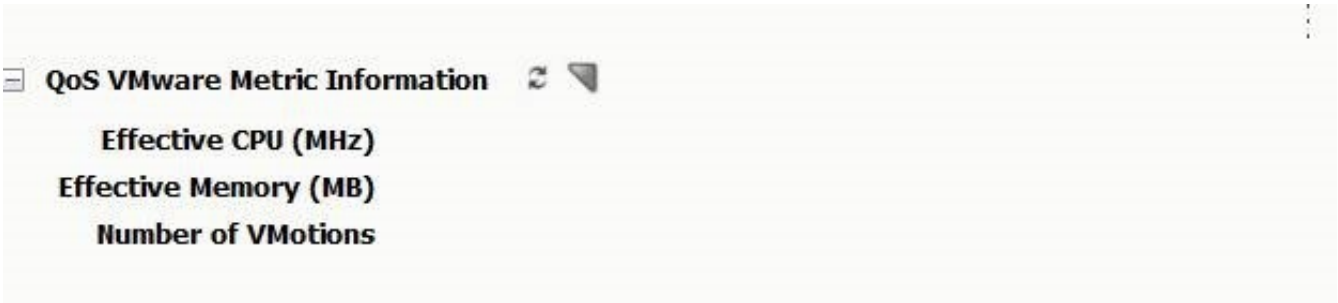
The following OneClick view gets reconciled for ESX Hosts and VMs:



The following OneClick view gets reconciled for a vCenter:



The following OneClick view gets reconciled for a Cluster:



When both the integrations are disabled

When you disable both the UIM and DX NetOps VNA integrations, you observe the following results:

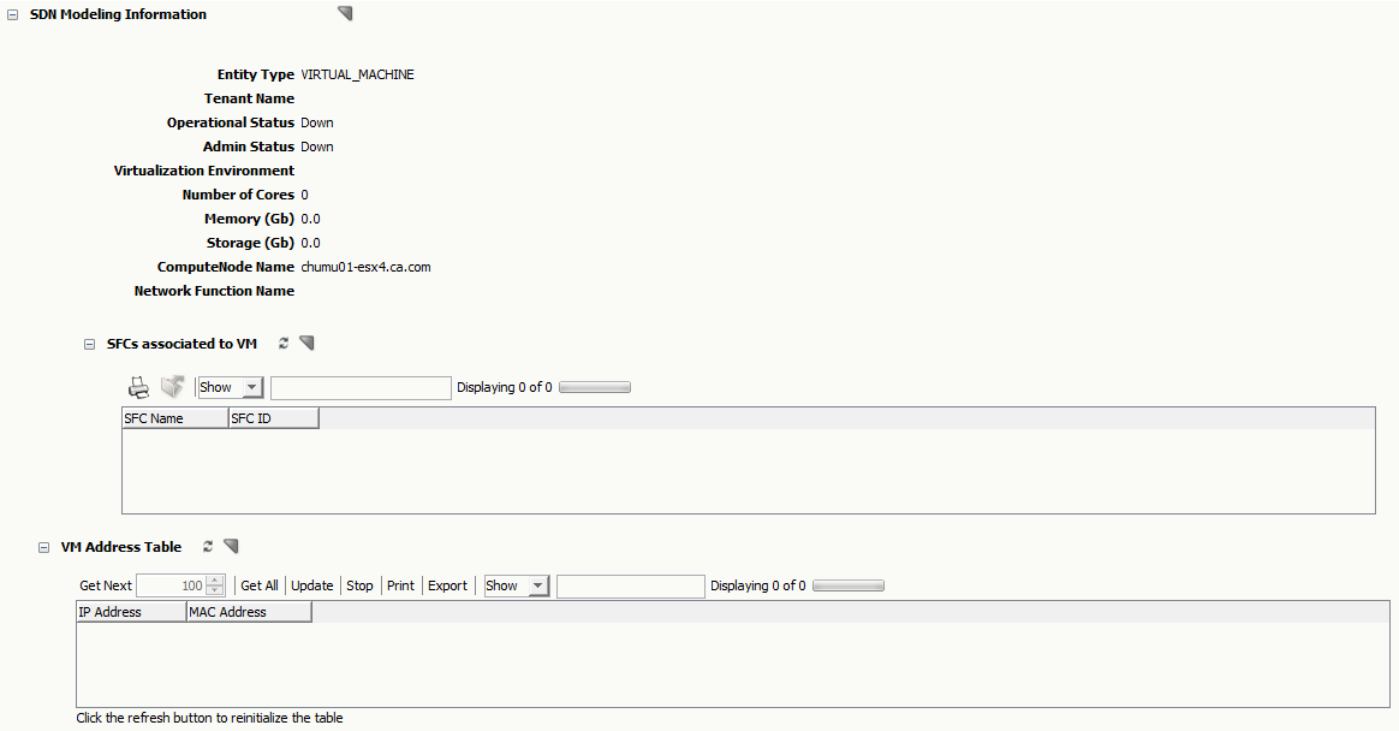
- The VNA Inventory is cleaned-up from SDN Manager.
- The UIM Inventory is cleaned-up from UIM Manager.
- Both VNA Inventory and UIM Inventory containers get deleted from the Universe.
- The existing SNMP Models (such as ACI Leaf, Spine, vCenter or Controller), and Pingable Models (VMs), which were modeled before the integration, are not deleted.

SDN Modeling Information

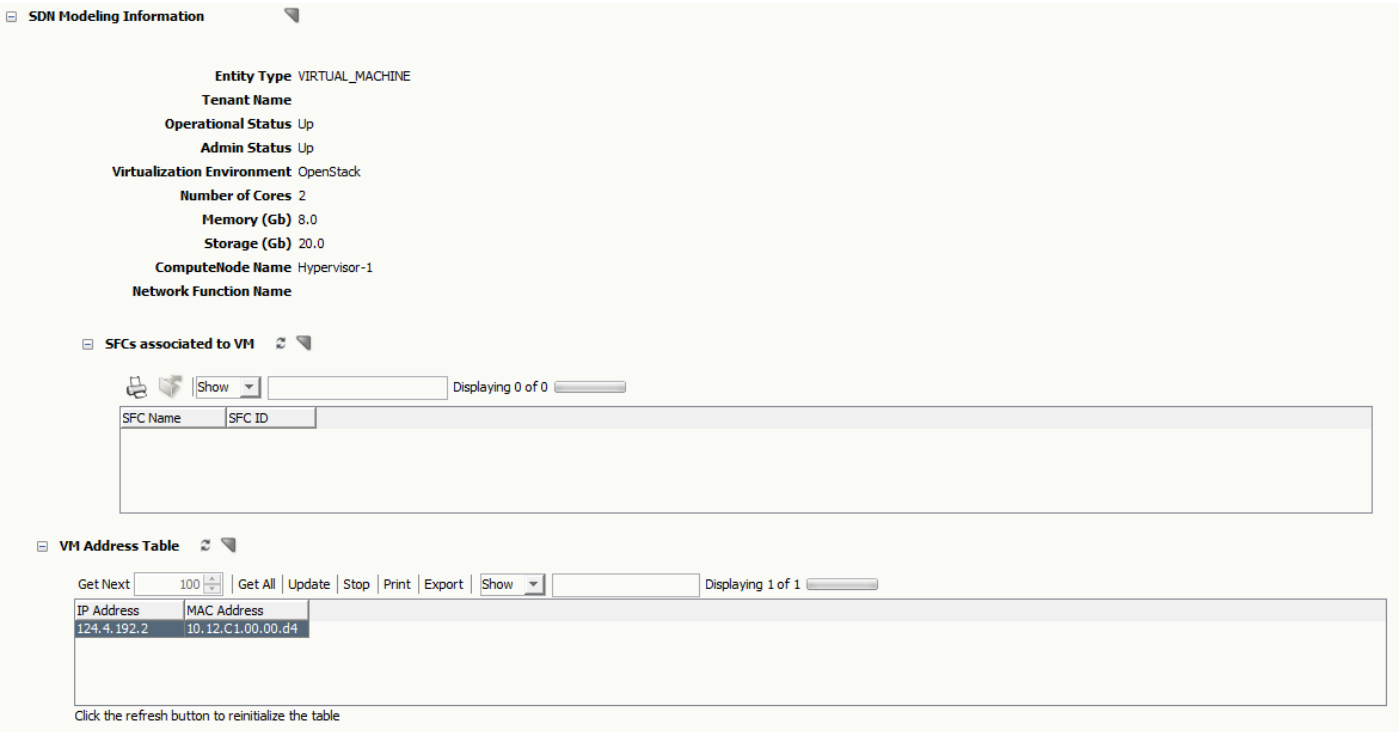
From the OneClick console, you can view the SDN Modeling Information of the Virtual Machine (VM) which is reconciled.

Follow these steps:

1. Open the DX NetOps Spectrum OneClick console.
2. Select **UIM Manager**, **Virtualization, VMware, Cluster, and** select **Resources** from the Navigation panel.
   The **Contents** pane opens.
3. Select the **Information** tab.
4. Expand **SDN Modeling Information**.
   SDN Modeling Information for the selected virtual machine is displayed.
   The following image displays the SDN Modeling Information of a virtual machine, when you integrate UIM first and DX NetOps VNA next.



The following image displays the SDN Modeling Information of a virtual machine, when you integrate DX NetOps VNA first and UIM next.



Launch-in-Context

The Launch-in-Context feature allows you to launch a UIM-UMP device/alarm view, from DX NetOps Spectrum for performance views of the device/alarm, in the given context. The UIM-UMP device view provides detailed information

about the IP elements such as Virtual Center (VC), ESX Hosts, and Virtual Machines (VM). The information (such as disk usage, CPU usage, processor queue length, paging, and memory usage) about the UIM virtual entities is displayed graphically.

If you are launching the UMP view for the first time in a browser, a dialog for user credentials appears. The user credentials dialog does not appear if you are launching the UMP view using the same browser instance.

NOTE

The Launch-in-Context feature is available only for the IP elements. This feature is disabled for logical entities such as Data Centers, ResourcePools, and Clusters.

Follow these steps:

1. In the DX NetOps Spectrum OneClick console, select **UIM Manager**, **Virtualization > VMWare**
2. Navigate to the device details that you want to view.
   A list of all entities under the sub-section that you have selected is displayed.
3. Right-click on one of the IP elements and select **Launch UIM UMP View**.
   The UIM UMP login page opens.
4. Enter the UIM UMP credentials and select 'Login'.
   The UMP view opens with the detailed information about the IP elements.

Analytics

The integration between DX NetOps Spectrum and CA Digital Operational Intelligence (App Experience Analytics) provides you with digital experience insights and helps you quickly determine if an issue with an app is with the infrastructure. You can triage the issue before it impacts the customer experience. This solution provides data-driven, actionable insights that are based on information that is received from different domains, types, and processes through the analytics platform. The DX NetOps Spectrum integration with the Analytics platform leverages the following information from DX NetOps Spectrum to help the user in a proactive resolution of issues:

- Alarms
- Network Configuration Manager - Change events (Disabled by default, enable if required)
- Topology

When you integrate DX NetOps Spectrum with DX Operational Intelligence, your enterprise gains the following benefits:

- Provide a holistic view of the availability of host servers and VMware environment across the network and their performance data for fault management in a single pane of the application.
- End-to-end root cause and impact analysis across network and server elements, extending the DX NetOps Spectrum core capabilities to other infrastructure domains.
- Advance condition correlation between CA UIM and DX NetOps Spectrum helps build robust fault management.
- Alarm management for both fault and performance alarms using either solution

With this integration, IT operators can do the following:

- Know the trends about device availability by groups. The groups can be created for different criteria using global collections.
- View the alarm information that is presented using multiple filters and are available out of the box or with custom filters.
- Use custom attributes to customize the out of the box dashboards.
- Drill down through the dashboards to view detailed information.

Integrate With DX Operation Intelligence

Overview

The integration between DX NetOps Spectrum and DX Operational Intelligence allows you to leverage data from DX NetOps Spectrum and helps to analyze, correlate, and proactively resolve the network issues. The SpectrumDataPublisher is a utility/service in DX NetOps Spectrum that publishes the DX NetOps Spectrum data to the analytics platform.

The 10.4.2 release supports sending Topology inventory and its relations from DX NetOps Spectrum to Topology Analytics Service (TAS) using DX NetOps Spectrum Data Publisher (Spub).

- SpectrumDataPublisher synchronizes DX NetOps Spectrum inventory to the TAS database used by DX OI.
- SpectrumDataPublisher pushes all the DX NetOps Spectrum devices and their relations among the adjacent devices to the TAS database.
- SpectrumDataPublisher converts DX NetOps Spectrum devices into Vertices, and devices relations to Edges before sending them to TAS.

The following data is synchronized from DX NetOps Spectrum to DX Operational Intelligence:

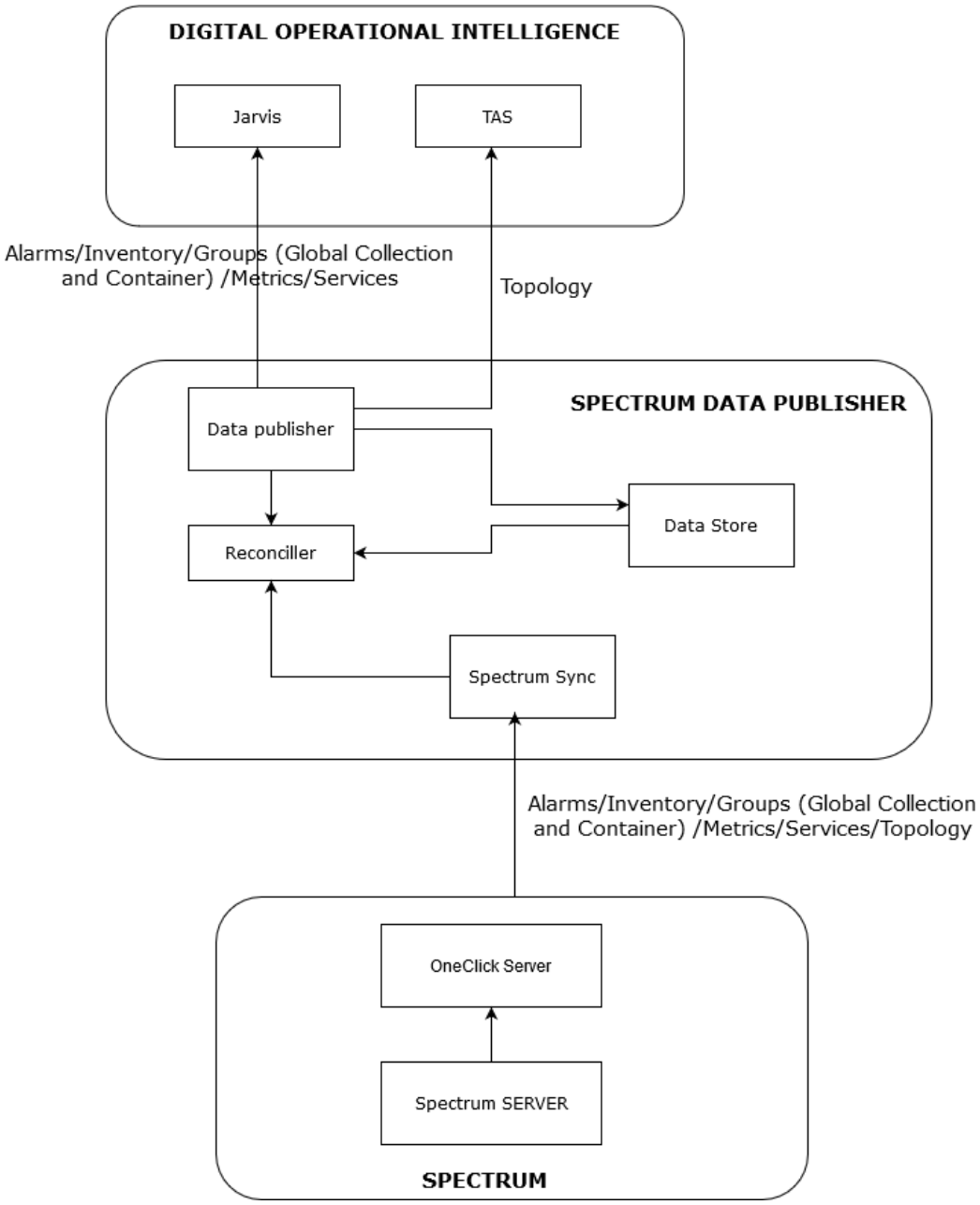
- Alarms

  IMPORTANT
  Reconciliation does not happen when elastic has a large number of alarms. There is a limitation of 10000 alarms from DX OI.
- Network Configuration Manager- Change events (Disabled by default, enable if required)
- Topology

  NOTE
  The following data are not supported from 10.4.2 release as the related data is pushed as part of Topology (TAS Integration):

  - Inventory
  - Groups (Global Collection and Container)
  - VNAInventory (Disabled by default, enable if required)
  - Metrics: Device count and Device availability (based on Model Type and Group)

The following diagram explains how the DX NetOps Spectrum-DX Operational Intelligence integration works:



Compatibility Matrix

For more information about the integration of DX NetOps Spectrum with other CA products, see Integration Compatibility.

Installing the SpectrumDataPublisher

The DX NetOps Spectrum OneClick page allows you to download and install the SpectrumDataPublisher JAR file. The DX NetOps Spectrum Data Publisher version should be the same as the version of the DX NetOps Spectrum.

Follow these steps:

1. Log in to the OneClick WebApp.
2. Navigate to the **OneClick Administration** page.

3. Select the **Analytics Configuration** option from the panel on the left.
   The **Analytics Configuration** page opens in a separate window.
4. Under the **DX NetOps Spectrum Data Publisher** section, select the SpectrumDataPublisher JAR file link to download the installer.
   The SpectrumDataPublisher.jar file is downloaded to the Downloads folder on your computer.
5. Perform one of the following steps:
   – In **Windows:** Double-click on the jar to open the install wizard and enter the **Install Directory**. By default, the path is:
     ```
     C:\win32app\
     ```
     The **DX NetOps Spectrum Data Publisher Install** wizard opens.
   – In **Linux** perform one of the following tasks to start the installation:
     • Execute the following command for silent installation:
       ```
       java -jar SpectrumDataPublisher.jar -i silent -DUSER_INSTALL_DIR="<install directory>"
       ```
       For example:
       ```
       java -jar SpectrumDataPublisher.jar -i silent -DUSER_INSTALL_DIR="/opt"
       ```
     • Execute the following command to export DISPLAY to a Windows Server and start GUI based installation:
       Export DISPLAY=<machine-name>:0 and then run the following command:
       ```
       java -jar SpectrumDataPublisher.jar
       ```
6. Select **Next**.
7. Select Install for the **DX NetOps Spectrum Data Publisher** to be installed.
8. Select **Done** to close the Install wizard.
   After the installation is complete, the SpectrumDataPublisher folder is created in the install folder location.

Configuring the SpectrumDataPublisher

To synchronize data from DX NetOps Spectrum to DX Operational Intelligence, configure the SpectrumDataPublisher.

Follow these steps:

1. Navigate to the DX NetOps Spectrum Data Publisher install location.
2. Open the 'config' folder.
3. Open the ConnectorConfig.xml file, and configure the following:
   a. Enter the DX NetOps Spectrum OneClick host server details in the **<SpectrumConfiguration>** section.
      For example:
      ```
      <SpectrumConfiguration>
             <OneClickServerUrl></OneClickServerUrl> <!-- give OneClickServerUrl. example: http://
      spectrum-123.net:8080/ -->
             <ConfigFile>SpectrumConfig.xml</ConfigFile>
             <WebappLaunchUrl></WebappLaunchUrl> <!-- give webapp url. example: http://
      spectrum-123.net:8080/spectrum/webapp/ -->
      </SpectrumConfiguration>
      ```

      If you want to use the Secure Sockets Layer (SSL) protocol to encrypt communications between OneClick and the SpectrumDataPublisher, you must import the SSL (https) Certificate of DX NetOps Spectrum into SpectrumDataPublisher.

      Launch OneClick Console or WebApp from DX OI Alarms (Introduced in 10.4.2.1)
      When `<WebappLaunchUrl></WebappLaunchUrl>` is empty OneClick console is launched and when you provide the WebApp URL the WebApp is launched.
   b. Enter the DX Operational Intelligence host Jarvis server details in the **<DestinationConfiguration>** section.
      For example:
      ```
      <!-- For CA Digital Operational Intelligence details -->
      ```

```
        <DestinationConfiguration>
            <DestinationType>DOI</DestinationType>
            <DestinationDefaultTenant></DestinationDefaultTenant>
            <DestinationUrl></DestinationUrl> <!-- give Jarvis host url. example: http://
jarvis.doi-123.net:8080/ -->
            <DestinationEntitiesPerPayload>200</DestinationEntitiesPerPayload>
            <ProxyHostForDestination></ProxyHostForDestination>
            <ProxyPortForDestination></ProxyPortForDestination>
            <ConfigFile>DOIConfig.xml</ConfigFile>
            <Alarms>true</Alarms>
            <Ncm>false</Ncm>
            <AlarmReconcileConfiguration> <!-- Clears the stale alarms in DOI during service startup.
 -->
                <Enable>false</Enable>
                <OIUrl></OIUrl> <!-- give OI host url. example:  https://doi.dxi-na1.saas.broadcom.com
 -->
                <BearerToken></BearerToken>
           </AlarmReconcileConfiguration>
        </DestinationConfiguration
```

NOTE

- Set the `<Enable></Enable>` parameter to **true** to clear the stale alarms.
- Use the **AlarmReconcileConfiguration** parameter to clear the stale alarms from the DX OI. Provide the DX OI URL and the bearer token. See Get DX OI Bearer Token Used in Alarm Reconcile Configuration to get the DX OI bearer token.

IMPORTANT

Reconciliation does not happen when elastic has a large number of alarms. There is a limitation of 10000 alarms from DX OI.

If you want to use the Secure Sockets Layer (SSL) protocol to encrypt communications between Jarvis and the SpectrumDataPublisher, you must import SSL (https) Certificate CA Jarvis into SpectrumDataPublisher.

c. Enter the TAS configuration in the **<TasConfiguration>** section:

```
 <TasConfiguration>
        <EnableIntegration>false</EnableIntegration>
        <TasUrl></TasUrl> <!-- give TAS host url. example: http://ngtas.doi-123.net:8080/ -->
        <EnableIncrementalSync>false</EnableIncrementalSync>
        <FullSyncInterval>720</FullSyncInterval> <!-- value in hours -->
        <IncrementalSyncInterval>30</IncrementalSyncInterval> <!-- value in minutes -->
        <FailedPayloadRetry>3</FailedPayloadRetry> <!-- default is 3 times, Set max up to 10 times -->
        <RetryTimeout>10</RetryTimeout> <!-- value in Seconds, each retry will wait for 10 seconds by
default, Set max up to 60 seconds -->
        <StorePath>tas/graph/storeAsync</StorePath>
    </TasConfiguration>
```

NOTE

Enter the **TasUrl** value and set the **EnableIntegration** to true to sync the data.

NOTE

If you want to use the Secure Sockets Layer (SSL) protocol to encrypt communications between TAS and the SpectrumDataPublisher, you must import SSL (https) Certificate to TAS endpoint.

After the configuration is complete, start the SpectrumDataPublisher service.

Start the SpectrumDataPublisher Service

To start the SpectrumDataPublisher service, follow these steps:

1. Perform one of the following step to start the SpectrumDataPublisher service:
   - In Windows, from the command line, run the `run.bat start` command.
   - In Unix, from the console, run the `run.sh start` command.
2. When you run the 'run.bat start' or 'run.sh' for the first time, it asks to create a log in password for SpectrumDataPublisher.

   NOTE
   You need this login password for next time when you run 'run.sh' or 'run.bat start' commands.
3. After creating the login password for SpectrumDataPublisher, it asks for the following login details that must be encrypted/decrypted with the password.

   1. **Enter OneClick username:** DX NetOps Spectrum username
   2. **Enter OneClick password:** DX NetOps Spectrum user password
   3. **Enter TAS Bearer Token:** TAS bearer token
   4. **Enter username for destination proxy -- if applicable:** Enter a proxy server user name
   5. **Enter password for destination proxy -- if applicable:** Enter the password of the proxy server

   The SpectrumDataPublisher service starts.

SpectrumDataPublisher Validation at Service Start

1. The SpectrumDataPublisher validates the user inputs and all the connections at the service startup using the following criteria before starting the service:
2. <SpectrumConfiguration> URL is empty.
3. Both Jarvis Url is empty and TAS integration is disabled, either of one is required to send the data.
4. It runs the test connection validations to all given URLs, and checks for the following errors:
   - SSL Error - This error comes when https configured, and certificates not imported.
   - HTTP 401 - if the authentication fails.
   - HTTP 407 - if proxy details are incorrect.
   - UnknownHost exception - if the provided URL is not valid.

   You can check the SpectrumDataPublisher.log for the exact error description.

Post-installation

- To avoid manual restart of the SpectrumDataPublisher service, change the 'Startup Type' from 'Manual' to 'Automatic'in Windows Services.

Stop the SpectrumDataPublisher service

If you want to stop the SpectrumDataPublisher-service, run the following command
Windows: `run.bat stop (Windows)`
Linux: `run.sh stop (Linux)`

Restart the SpectrumDataPublisher service

If you want to restart the SpectrumDataPublisher-service, run the following command:
Windows: `run.bat restart (Windows)`
Linux: `run.sh restart (Linux)`

NOTE
On Windows machines, the SpectrumDataPublisher-service can be stopped and restarted through the Windows Services Console.

Import SSL (https) Certificate into SpectrumDataPublisher

If you want to use the encrypted communication (https protocol) between DX NetOps SpectrumDataPublisher and integrated products (DX NetOps Spectrum and CA Jarvis server), you must import the SSL/https certificate from the respective products into the SpectrumDataPublisher.

Follow these steps to import the certificate:

1. Ensure that the 'https' certificate is exported and copied to the server where the SpectrumDataPublisher is installed. For export instructions, refer to the notes provided at the end of this section.
2. Perform one of the following steps to download the CA Jarvis Server SSL/HTTPS certificate:
   - Run the following command:
     ```
     openssl s_client -connect Jarvis_Hostname:443  < /dev/null | sed -ne '/-BEGIN CERTIFICATE-/,/-END
      CERTIFICATE-/p' > jarvisServer.cer
     openssl s_client -connect TAS_Endpoint:443  < /dev/null | sed -ne '/-BEGIN CERTIFICATE-/,/-END
      CERTIFICATE-/p' > tas.cer
     ```
     OR
   - Access the Jarvis URL using HTTPS in a browser and save the certificate from site information.
3. Run the following command to import the certificate into the Keystore of SpectrumDataPublisher.
   ```
   keytool -importcert -alias <certificate_alias> -file /<PATH>/<FILENAME.cer> -keystore /<PATH>
   ```
   For example:
   ```
   keytool -importcert -alias tomcatssl -file /SpectrumDataPublisher/OCServer.cer -keystore /
   SpectrumDataPublisher/Security/cacerts
   keytool -importcert -alias jarvisssl -file /SpectrumDataPublisher/jarvisServer.cer -keystore /
   SpectrumDataPublisher/Security/cacerts
   keytool -importcert -alias tas -file /SpectrumDataPublisher/tas.cer -keystore /SpectrumDataPublisher/
   Security/cacerts
   ```

   NOTE
   Ignore the warning about migrating to PKCS12 format.
4. When prompts, provide the Keystore 'changeit' as a password.
5. After the certificate is imported, stop the SpectrumDataPublisher service.
6. Make sure the hostname, port, and protocol details are proper in the /SpectrumDataPublisher/config/ConnectorConfig.xml
7. Start the SpectrumDataPublisher service.

   NOTE
   To export the ssl/https certificate of DX NetOps Spectrum, run the following command from DX NetOps Spectrum Oneclick server:
   ```
   keytool -export -keystore /usr/<PATH> -alias <certificate_alias> -file <FILENAME.cer>
   ```
   For example:
   keytool -export -keystore /usr/Spectrum/custom/keystore/cacerts -alias tomcatssl -file OCServer.cer

Disable SSL Certificate Validation

You can disable the verification to check the validity of certificates. You can disable SSL Certificate Validation in the following scenarios:

- When you do not have the SSL certificates.
- When you get an SSL exception, even after you import the SSL certificates.

Add the following tags in the `ConnectorConfig.xml` file:

```
<DisableSSLHostnameVerifier>true</DisableSSLHostnameVerifier>
<TrustAllX509Certificates>true</TrustAllX509Certificates>
```

Reset the Login Details of SpectrumDataPublisher service

In case you forgot the login password for the SpectrumDataPublisher service, you can reset the same by using the following commands. Note that, when you reset the login details, the stored configuration details are deleted and you should provide the details again in the ConnectorConfig.xml file.
Windows: `run.bat reset (Windows)`
Linux: `run.sh reset (Linux)`

Logging Configuration for SpectrumDataPublisher

You can find the following log files in the SpectrumDataPublisher/logs/ folder:

- The `SpectrumDataPublisher.log` shows the information and error messages of all the syncs.
- The `topology.log` shows the inventory data sent from DX NetOps Spectrum to TAS.
- The `alarm.log` shows the alarm data sent from DX NetOps Spectrum to Jarvis..
- The `ncm.log` shows the NCM data sent from DX NetOps Spectrum to Jarvis.

Note the following points about the logs:

- To set the `SpectrumDataPublisher.log` level change the value of `rootLogger.level` to info, debug, error, or fatal in the log4j2.properties file.
- Set the `logger.topoLogger.level` parameter to **trace** in log4j2.properties to log the topology data. **logger.topoLogger.level=trace**
- Set the `logger.alarmLogger.level` parameter to **trace** in log4j2.properties to log the alarm data. **logger.alarmLogger.level=trace**
- Set the `logger.ncmLogger.level` parameter to **trace** in log4j2.properties to log the NCM data. **logger.ncmLogger.level=trace**
- The maximum size of the log file is 100 MB. When the file reaches the maximum size, a backup file is created. The name of the backup file is SpectrumDataPublisher.log prefixed with a timestamp. The normal logging is continued in the SpectrumDataPublisher.log file.

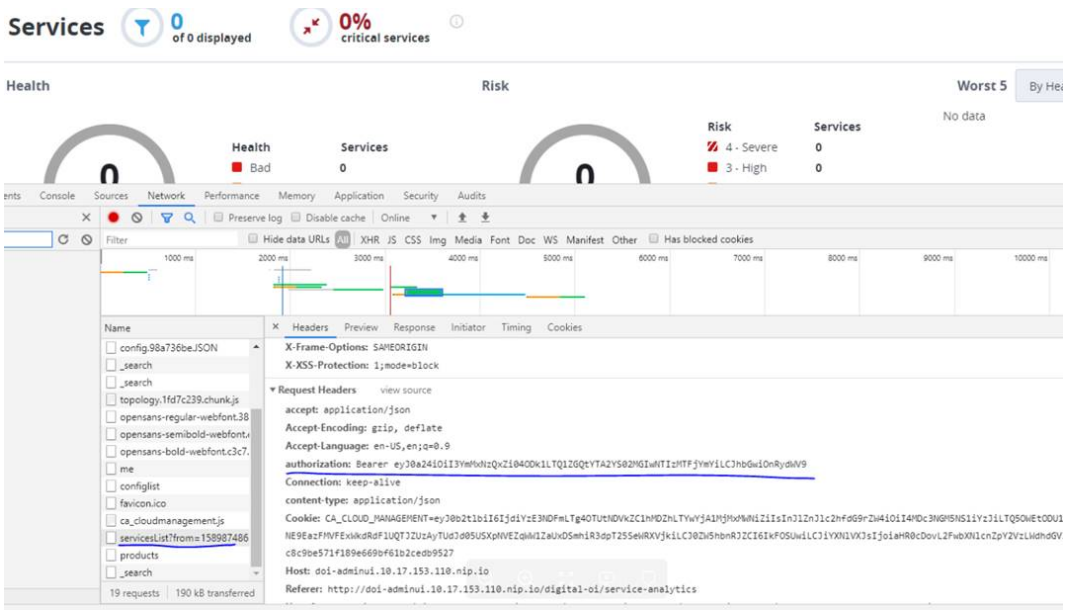
Get DX OI Bearer Token Used in Alarm Reconcile Configuration

You need DX OI bearer token when you configure the SpectrumDataPublisher. Note this bearer token for future use.

> **NOTE**
> The token expires periodically, hence you must fetch the token afresh while starting the service.

Follow these steps:

1. Refresh the DOI page after login.
2. Press CTRL+SHIFT+R on the keyboard.
3. Navigate to the Network.
4. Refresh the page again.
5. Click on any API in the **Network** tab.
6. Scroll down, you can see the authorization: Bearer.



Troubleshooting SpectrumDataPublisher

Troubleshooting information for the problems that are encountered with SpectrumDataPublisher.

DX NetOps Spectrum Data does not sync to DX Operational Intelligence

Symptom:
SpectrumDataPublisher.log shows the exception: com.ca.spectrum.spub.common.ConnectorException. DX NetOps Spectrum Data does not sync to DX Operational Intelligence.

Resolution:
Provide correct details of the Jarvis server under the DestinationConfiguration section of ConnectorConfig.xml and make sure the DestinationHostname is resolved to a valid IP address.

Configure Events Synchronization

After the DX NetOps Spectrum-CA Digital Operational Intelligence integration, configure the following UIM probes to view the analytics dashboards for DX NetOps Spectrum events in the CA Digital Operational Intelligence.

1. Configure the **Log Forwarder (log_forwarder) probe.**
   The log_forwarder probe reads the DX NetOps Spectrum log files and publishes the log file content to a specific UIM queue with subject LOG_ANALYTICS_LOGS. For log_forwarder probe configuration instructions, see the log_forwarder (Log Forwarder) section in the CA Unified Infrastructure Management Probes documentation. The Log Forwarder probe should be present in the SpectroSERVER.
2. Configure the **AXA Log Gateway (axa_log_gateway) probe.** The AXA Log Gateway (axa_log_gateway) probe transfers the log data from the UIM Log Analytics Queue (subject: LOG_ANALYTICS_LOGS) and writes the data to the CA Digital Operational Intelligence (Default: logAnalyticsLogs). For axa_log_gateway probe configuration instructions, see the axa_log_gateway (AXA Log Gateway) section in the DX Unified Infrastructure Management Probes documentation.

Set up the Data Source

You must configure the log data collection source to send log data from DX NetOps Spectrum to CA Digital Operational Intelligence. The Log Forwarder probe collects the log data and sends it to the UIM queue.

In DX NetOps Spectrum, the Archive Manager resource settings allow you to control how historical records are processed. These resources also enable the Archive Manager to communicate with the SpectroSERVER and DDM databases.
Archive Manager resources are defined in the .configrc file, which is located in the $SPECROOT/SS/DDM directory. The resources are listed in the form “resource = resource_value”.

Follow these steps to update the .configrc file:

1. Edit $SPECROOT/SS/DDM/.configrc to add the following attributes:
   - **FLATFILE_EVENTS_FILENAME** - Specifies the file name and location where events log file should be saved.
     Example: FLATFILE_EVENTS_FILENAME=C:\win32app\Spectrum\SS\AXAEventsFile.log
     If you are saving the log file to a different computer or mapped network drive, you need to provide the full path of the computer. Providing only the drive name such as 'z:' is not accepted in the path.
     Example: \\<host name>\shared\AXAEventsFile.lo

     NOTE
     The Log file can be saved only on the local computer and cannot be saved on any SAN/NAS storage.
   - **FLATFILE_EVENTS_MAX_FILESIZE** - Specifies the file size in MB.
     Example: FLATFILE_EVENTS_MAX_FILESIZE=1 (the value 1 represents the file size in MB, each log file should be of 1-MB size and anything more than 1 MB is rotated)
     If the attribute is not added in the .configrc file, then the default max size considered is 100 MB. The maximum size accepted is 2048 MB.
   - **FLATFILE_EVENTS_BACKED_UP_FILECOUNT** - Specifies the count of the number of rotated log files to retain, rest all would be deleted.
     Example: FLATFILE_EVENTS_BACKED_UP_FILECOUNT=5 (the value 5 represents the count of the number of rotated log files to retain, it retain only 5 rotated log files)
     If the attribute value is not configured, then the default number of files that are considered is 10.
2. Edit $SPECROOT/SS/.vnmrc to set the **AXA_EVENT_INTEGRATION_ENABLED** attribute value to true. This attribute populates IPs and Groups of the devices or servers that you want to collect log data from. If the attribute value is not configured, then the IPs and Groups show no data.
   Example: AXA_EVENT_INTEGRATION_ENABLED=tru

   NOTE
   To read the custom events, the events and PCause files should be present in OneClick and SpectroSERVER.
3. Restart the archive manager and SpectroSERVER to start the logging of events.
   a. Stop the SpectroSERVER and the Archive Manager by clicking Stop SpectroSERVER in the DX NetOps Spectrum Control Panel
      Or, you can stop the SpectroSERVER and Archive Manager from the command line by running the "< $SPECROOT>/bin/stopSS.pl" as Spectrum Owner at the command prompt.
   b. Start the SpectroSERVER and the Archive Manager by clicking Start SpectroSERVER in the DX NetOps Spectrum Control Panel
      Or, you can start the SpectroSERVER and Archive Manager from the command line by running the "< $SPECROOT>/bin/startSS.pl" as Spectrum Owner at the command prompt.

Create Profile and Add Filters for Spectrum Events

In the Log Forwarder (log_forwarder) probe, you need to create a profile and define filters (Log Exclude Rules and Log Format Rules) for Spectrum events.

Create Spectrum Events Profile

To monitor the Spectrum events data in CA Digital Operational Intelligence, you need to create a configuration profile in the Log Forwarder (log_forwarder) probe. For the steps to create a profile, see **Create a Profile** section in the log_forwarder AC Configuration page.

Define Filters

In the Log Forwarder (log_forwarder) probe, you can define the following filters:

- Log Exclude Rules - To identify a line or block of input which you want the probe should exclude from monitoring. For the steps to define Log Exclude Rules, see Create Log Excludessection.
- Log Format Rules - To specify a line or block of input which starts with a defined start and end text or expression. For the steps to define Log Exclude Rules see Create Log Format Rules section.

> **NOTE**
> The Log Format Rule for DX NetOps Spectrum events must contain proper Start and End Expressions. For details see Create Log Format Rules section in the Log Forwarder (log_forwarder) probe documentation.

View DX NetOps Spectrum Events Analytics Dashboards in CA Digital Operational Intelligence

After configuring the probes, setting up the data collection source, creating profile and defining filters you can view the Spectrum events log analytics dashboards in CA App Experience Analytics. The Data Studio page in the AXA is a primary user interface for Log Analytics. Data Studio provides out-of-the-box dashboards for the supported log types, full-text search, and ad-hoc data exploration. For more information, see Data Studio.

Programming

This section contains information about using DX NetOps Spectrum to create a custom solution and contains reference material for the developers.

Development API Reference

The Development API is the Common Object Request Broker Architecture (CORBA)-based application program interface (API) to SpectroSERVER. You can also refer to it as the SpectroSERVER Object Request Broker (SSORB) interface or SSORB, because it depends on an object request broker (ORB). This section explains the purpose of the Development API, what it consists of, and what you can do with it.

> **NOTE**
> See the Release Information section for notifications of release-specific changes (if any) to the Development API interface.

Architecture

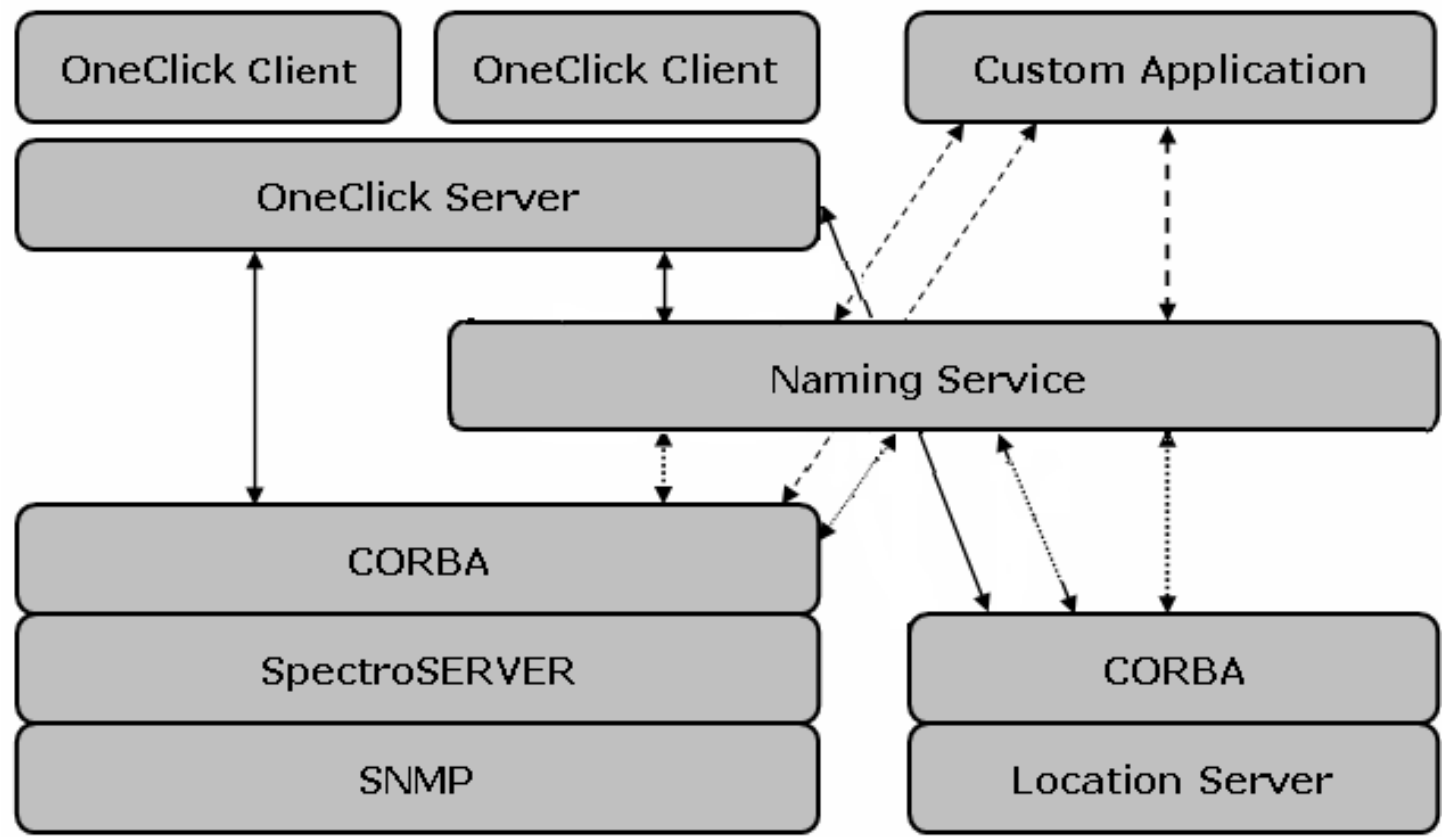
The Development API advertises through the CORBA Naming Service. Advertising through the CORBA naming service facilitates compatibility with other ORBs such as Orbacus and Java that support the standard Naming Service.

Client applications contact the Naming Service to retrieve the list of available services. You can view the list by using the *osfind* tool or the *nsutil* tool for the Naming Service list. Once the client application contacts an advertiser and gets the location of the required service, the client application can contact that service directly. Calls for data retrieval are possible after security checks are cleared.

Native threads handle incoming data requests at the VisiBroker CORBA layer. The requests then queue for SpectroSERVER processing.

The SpectroSERVER takes the first request off the queue and retrieves the required data. The data is attached to the outgoing queue to return to the client application. In the case of asynchronous data or callbacks, the SpectroSERVER maintains an active list of changes and notifies the client application about them. On multi-CPU systems where client applications implement callbacks, this workload is redistributed among native threads, freeing up processing cycles.

The Development API has the following architecture:



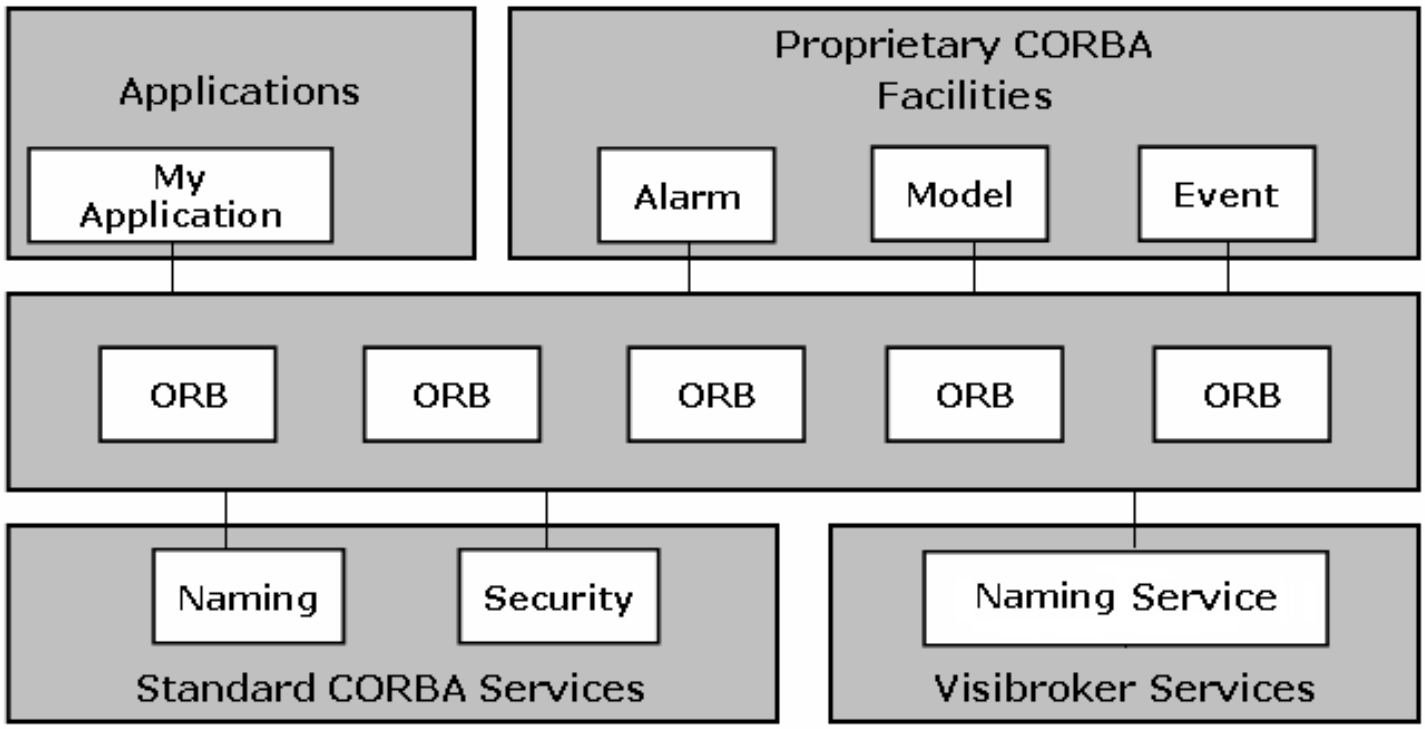
In this architecture, the Location Server, SpectroSERVER, and Archive Manager (not pictured) each have a CORBA interface for data retrieval. The Location Server provides the list of SpectroSERVERs available in the distributed SpectroSERVER domain. SpectroSERVER handles the interfaces to the model domain, alarm domain and type catalog. The Archive Manager handles the event domain.

When the SpectroSERVER, Location Server, or Archive Manager starts up, it advertises its services to the Naming Service.

The CORBA Standard

CORBA is a standard developed by the Object Management Group (OMG) consortium. The CORBA standard is a middleware, a framework that specifies how distributed applications can communicate using an ORB.

The following illustration shows client-server communication through an ORB:



The ORB facilitates communication between software components coded in different languages for different operating systems with different hardware architectures. It enables disparate objects on the network to recognize each other and invoke each other's methods.

The ORB is not a product; it is an implementation of OMG's CORBA standard. The CORBA standard defines the Interface Definition Language (IDL) used to describe the following:

- Interfaces to CORBA objects
- Programming language mappings
- Transport protocols
- High level services like naming and security

The CORBA Mechanism

The CORBA component that facilitates communication between remote applications is the IDL that defines an API. The SPECTRUM Development API consists of an IDL file for each object type. This set of IDL files is usually referred to as the DX NetOps Spectrum IDL.

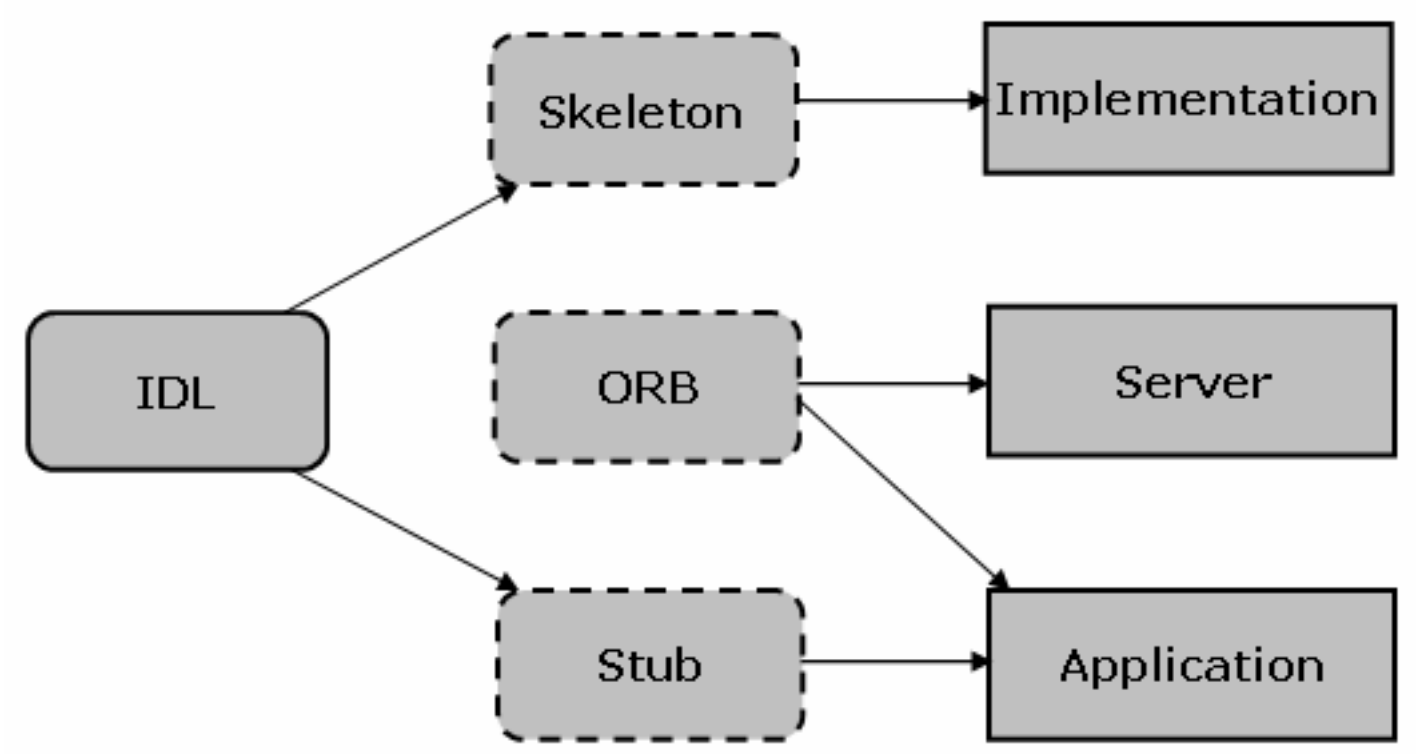
NOTE
The term IDL can also describe an interface.

CORBA defines a language mapping for every programming language. The language mapping describes how an IDL converts into the language-specific API. An IDL compiler implements this mapping by compiling an IDL file into a set of stubs and skeletons. *Stubs* are the methods and objects that an application can use; *skeletons* are empty methods and objects that you fill in to implement a server. The IDL generates stubs and skeletons while a vendor provides the ORB.

A programmer who creates a CORBA service object defines the interface in a CORBA-standard language, namely the IDL. If the IDL is run through the appropriate compilers, it produces both server and client code in standard programming languages. The server code is the skeleton because it represents a basic object as an abstract class from which the programmer derives a class to implement a service. The client code consists of a set of object reference stubs, which need no further coding.

For example, for a C++ mapping, the stubs include the header files used to compile the application and the object library linked to the application.

The application and server should link to the ORB library as shown in the following diagram:



To develop a server, you need to write the IDL and the implementation inside the skeletons. To develop an application, you need to call the stubs.

A program participating in the CORBA framework relies on its local ORB to overcome the communication barriers of language and process location. The program sends requests and gets responses to and from a data pipeline composed of all the ORBs in the system. This network of ORBs is often referred to as the ORB.

Each CORBA service resides on some computer and is implemented as an object coded in a programming language. To make itself available to clients, the object advertises its services through the ORB. A naming service (a standard CORBA service) picks up the advertisements and records the service name and object reference. If a client wants to use the service, it specifies the service name and contacts the naming service through the ORB. It receives the object reference and uses it to access the object's services.

The object reference and its corresponding server object are instances of different classes with the same interface. An object reference, obtained by the client from the naming service, executes on the client. The server object resides on the server. If a client invokes an object reference method, the method uses the ORB to call the corresponding method on the server. This server method fulfills the service request and returns the required data. For the client, the local object reference simulates the server object, but actually, the reference only does remote method invocations to the server.

Why DX NetOps Spectrum Uses CORBA

The Development API uses CORBA to focus on Java application development.

Use of Java in web-based applications is increasing. The SPECTRUM Development API focuses more on Java though it provides the same set of core services to both C++ and Java developers.

CORBA, an open standard, is best suited for Java application development. CORBA helps DX NetOps Spectrum customers write CORBA applications that integrate DX NetOps Spectrum with other CORBA services.

CORBA Sources

OMG publishes CORBA standard sections that cover the architecture and the language-specific mappings of CORBA. These standards are available at http://www.omg.org.

You can also get CORBA information from Borland Software Corporation, the vendor of VisiBroker ORB that DX NetOps Spectrum uses. The documentation is available at http://www.borland.com.

Data Structure Classes

Data structure classes are of two types: parameter and helper.

- **Parameter Class**
  Transfers data. Your client programs instantiate these classes and pass them to interface methods. The methods also return instances of these classes to your programs. Examples include CsCModelPropList, CsCMTypePropList, and CsCAttrValList.
- **Helper Class**
  Builds and unwraps complex parameter class objects.

You can find the Java docs for these classes by clicking the link com.aprisma.spectrum.core.util in the file <*$SPECROOT*>/SDK/docs/SSORB/index.html. Examples include CsCorbaValueHelper and CsCAttrFilterHelper.

Error Reporting

CORBA reports errors through exceptions. Error reporting through exceptions is an all-or-nothing approach because no other result is possible. This approach is not helpful for partial failures, when only a few items of a request fail. Also, it is not optimal to raise an exception for every failure. The CORBA approach is to use exceptions for complete failures and report partial failures with error codes in result lists. Because it is not optimal to search an entire list to see if any item failed, result lists also provide an overall error code to indicate partial failures.

C++ Development on Windows

The following table lists the library and dynamic link libraries necessary for C++ development on Windows. The corresponding directory location is <*$SPECROOT/lib*>.

Library Files	SDK
libGlobl (dll), libssorb (dll), ibssorbutil (dll)	SPECTRUM
orb_r_70 (dll), cosnm_r_70 (dll), vport_r_70 (dll), vdlog_r_70 (dll)	Borland VisiBroker

The following table shows the corresponding header file locations:

Library	Header File Location
libGlobl.dll	<*$SPECROOT*>\SDK\include\GLOBL
libssorb.dll	<*$SPECROOT*>\SDK\include\SSORB\idl
libssorbutil.dll	<*$SPECROOT*>\SDK\include\SSORB\util
cosnm_r_70.dll, orb_r_70.dll, vbsec.dll, vdlog_r_70.dll, vport_r_70.dll	Provided by Borland VisiBroker SDK (Not included in SPECTRUM CORBA toolkit.)

Prerequisites for programming with the Development API

Verify that you have the following in place before you start programming with the Development API:

- Development requirements.
- DX NetOps Spectrum version -- You need r9.*x* or above, with SpectroSERVER and Archive Manager (ArchMgr) running.
- DX NetOps Spectrum access -- If your DX NetOps Spectrum server and client computer are different, the .hostrc file in the DX NetOps Spectrum directory on your DX NetOps Spectrum server should allow the programs on your client to interact with the SpectroSERVER. To facilitate this, open the .hostrc file with a text editor, remove the individual host name if it exists, and add either a plus (+) sign or the name of your client computer, and close the file. You need not shut down and restart any DX NetOps Spectrum processes; they automatically re-read the configuration file within a minute.
- User model -- Verify that DX NetOps Spectrum has a user model with a name that matches the user name you use on your client computer to run the example programs.
- Devices -- Many of the exercises require you to create models or read model attributes. It is helpful to have a few SNMP devices (switches, routers, and so on) running on your network so that you can model them. You can model your own client computer if it is running a supported SNMP agent.
- Work directory -- Create a work directory for the example exercises in the following section. You need to set the CLASSPATH variable pointing to the CORBA JAR files, which are located in <*$SPECROOT*>/lib.

  NOTE
  For CLASSPATH examples, see Java Development Specifics.

If your development environment meets all these criteria, you can compile a program by executing javac <*ProgramName*>.java and run a program by executing java <*ProgramName*>.

Environment Verification

You can perform a test of your development environment once you have verified the following:

- SpectroSERVER and ArchMgr are running on your DX NetOps Spectrum server.
- JDK and DX NetOps Spectrum SDK are present on your client computer.
- DX NetOps Spectrum SDK contains the necessary example programs, jar files, and javadoc files.

The test involves verifying the client-server communication and whether you can compile and run the example programs from the cmdline directory.

To verify that the example programs can communicate with your SpectroSERVER, enter the following:

```
cd SDK/examples/SSORB/DevelopmentAPI
javac GetDomainID.java
java GetDomainID <domainName>
```

Replace <*domainName*> with the Domain Naming System (DNS) or network name of your SpectroSERVER host system. The GetDomainID example should display the landscape handle of your SpectroSERVER.

Version Requirements

The following table lists the version requirements for the various software tools used with the Development API:

Development Tools	Version
Borland VisiBroker SDK	8.5sp1

Windows	Windows Server 2016 Windows Server 2012
Java SDK	1.7.0
Red Hat Enterprise Linux	6, 7
GNU C++	4.1 or 4.4
Java SDK	1.7.0
Java SDK	1.7.0

Java Development Specifics

The following JAR files are necessary for Java development with DX NetOps Spectrum SDK, where *xx* is the version of DX NetOps Spectrum that you are running. For example, if you are running DX NetOps Spectrum r9.3, global*xx*.jar would be global93.jar.

- global*xx*.jar
- ssorb*xx*.jar
- ssorbutil*xx*.jar
- util*xx*.jar
- utilapp*xx*.jar
- utilnet*xx*.jar
- utilsrv*xx*.jar
- vbhelper*xx*.jar
- vbjorb.jar
- bsec.jar
- lm.jar
- cryptojFIPS.jar
- sanct6.jar (Linux only)
- sanctuary.jar (Linux only)

The corresponding directory location is <*$SPECROOT*>/lib, and the JAR file location should be set by the CLASSPATH environment variable.

Example: Setting the CLASSPATH environment variable

The following are examples of setting the CLASSPATH variable for different platforms. Replace the following variables accordingly:

- ***<$SPECROOT>***
  Specify the complete path to your DX NetOps Spectrum installation directory.
- ***xx***
  Specify the version of DX NetOps Spectrum that you are running. For example, if you are running DX NetOps Spectrum r9.3, global*xx*.jar would be global93.jar.
- You can set the CLASSPATH on Windows as follows:

```
CLASSPATH=<$SPECROOT>\lib\globalxx.jar;<$SPECROOT>\lib\ssorbxx.jar;<$SPECROOT>\lib\ssorbutilxx.jar;<
$SPECROOT>\lib\vbhelperxx.jar;<$SPECROOT>\lib\vbjorb.jar;<$SPECROOT>\lib\vbsec.jar;<$SPECROOT>\lib
\utilxx.jar;<$SPECROOT>\lib\utilnetxx.jar;<$SPECROOT>\lib\utilsrvxx.jar;<$SPECROOT>\lib\lm.jar;<$SPECROOT>
\lib\cryptojFIPS.jar;<$SPECROOT>\lib\utilappxx.jar;.;
```

- You can set the CLASSPATH on Linux as follows:

```
CLASSPATH=<$SPECROOT>/lib/globalxx.jar:<$SPECROOT>/lib/ssorbxx.jar:<$SPECROOT>/lib/ssorbutilxx.jar:<
$SPECROOT>/lib/vbhelperxx.jar:<$SPECROOT>/lib/vbjorb.jar:<$SPECROOT>/lib/vbsec.jar:<$SPECROOT>/
lib/utilxx.jar:<$SPECROOT>/lib/utilnetxx.jar:<$SPECROOT>/lib/utilsrvxx.jar:<$SPECROOT>/lib/lm.jar:<
$SPECROOT>\lib\cryptojFIPS.jar:<$SPECROOT>/lib/sanct6.jar:<$SPECROOT>/lib/sanctuary.jar:<$SPECROOT>/lib/
utilappxx.jar:.:
```

C++ Development on Windows

The following table lists the library and dynamic link libraries necessary for C++ development on Windows. The corresponding directory location is <*$SPECROOT/lib*>.

Library Files	SDK
libGlobl.lib libssorb.lib ibssorbutil.lib	DX NetOps Spectrum
orbcore_r.lib cosnm_r_80.lib vport_r_80.lib vdlog_r_80.lib	Borland VisiBroker

The following table shows the corresponding header file locations:

Library	Header File Location
libGlobl.dll	<*$SPECROOT*>\SDK\include\GLOBL
libssorb.dll	<*$SPECROOT*>\SDK\include\SSORB\idl
libssorbutil.dll	<*$SPECROOT*>\SDK\include\SSORB\util
orbcore_r_80.dll cosnm_r_80.dll vport_r_80.dll vdlog_r_80.dll	Provided by Borland VisiBroker SDK (Not included in DX NetOps Spectrum CORBA toolkit.)

C++ Development on Linux

The following table lists the shared objects and libraries necessary for C++ development on Linux. The corresponding directory location is <*$SPECROOT/lib*>.

Library File	SDK
libGlobl (so, so.1) libssorb (so, so.1) libssorbutil (so, so.1) libPort (so, so.1) libVPapi (so, so.1)	DX NetOps Spectrum
liborbcore_r.so libcosnm_r.so libvport_r.so libvdlog_r.so	Borland VisiBroker

The following table shows the corresponding header file locations:

Shared Object/Library File	Header File Location
libGlobl.so	<*$SPECROOT*>/SDK/include/GLOBL
libssorb.so	<*$SPECROOT*>/SDK/include/SSORB/idl
libssorbutil.so	<*$SPECROOT*>/SDK/include/SSORB/util
libPort.so	<*$SPECROOT*>/SDK/include/PORT
libVPapi.so	<*$SPECROOT*>/SDK/include/VPAPI
liborbcore_r (so, so.8.0) libcosnm_r (so, so.8.0) libvport_r (so, so.8.0) libvdlog_r (so, so.8.0)	Provided by Borland VisiBroker SDK (Not included in DX NetOps Spectrum CORBA toolkit.)

Standard Naming Service and VisiBroker ORB

Client applications use the Standard Naming Service, resulting in ORB independence. With ORB independence, the only requirement is that the ORB implement the Standard Naming Service.

ORB Examples

The DX NetOps Spectrum Developer Kit includes CORBA client applications built with ORBs from other vendors that use the standard naming service.

The following table lists the ORBs and directory locations for different development platforms:

ORB	Platform and Development	Directory Location
Mico	Linux, C++	<*$SPECROOT*>/SDK/examples/SSORB/nameserv/cc/mico
Java	Windows Java	<*$SPECROOT*>/SDK/examples/SSORB/nameserv/java/

The DX NetOps Spectrum IDL

The IDL files are the essential part of the SPECTRUM Development API (which also includes the helper classes, examples, and documentation). The IDL files define the interfaces to various DX NetOps Spectrum services (model domain, event domain, and so on) and are available in the following directory:

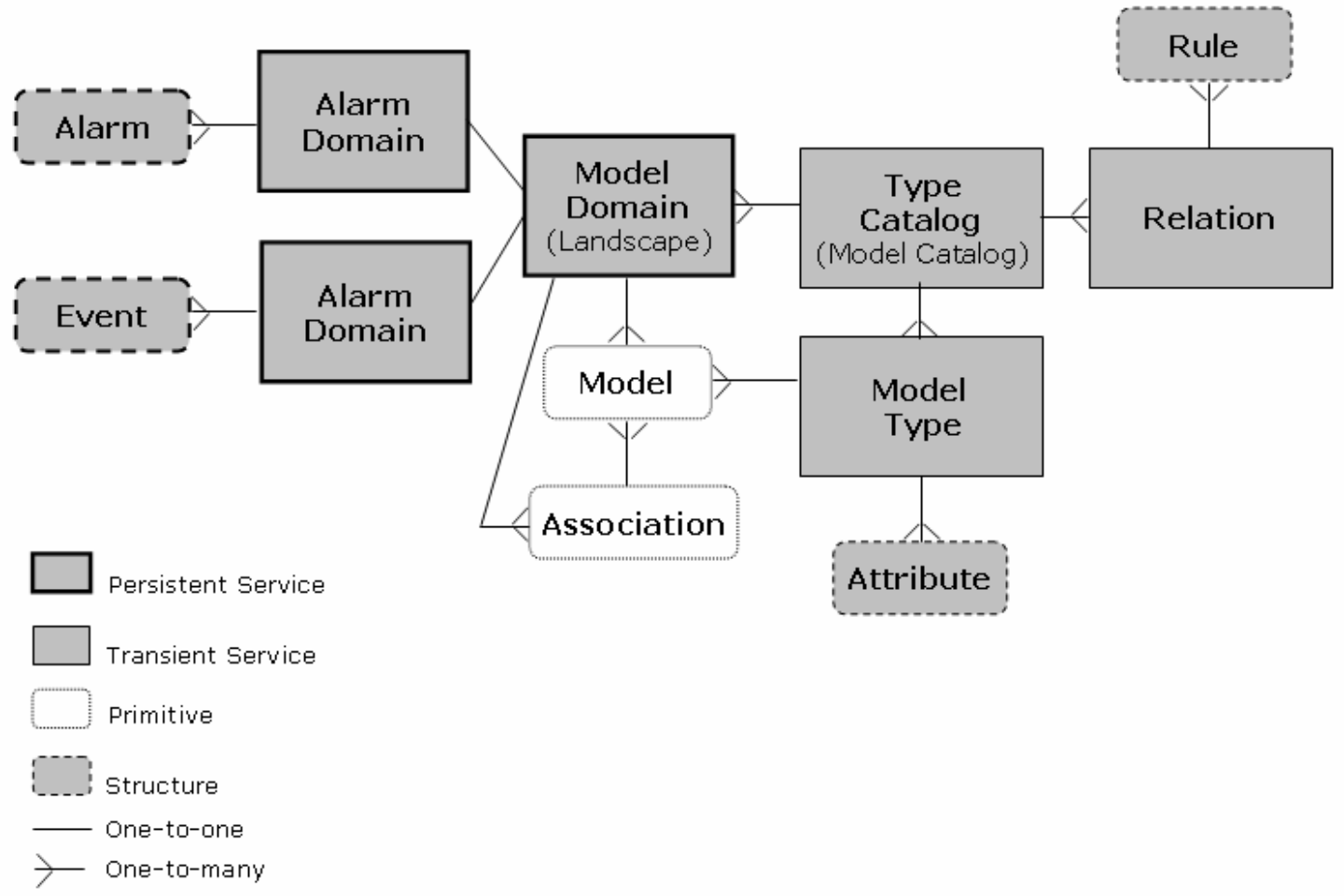
```
<$SPECROOT>/SDK/examples/SSORB/idl
```

WARNING
The DX NetOps Spectrum IDL is proprietary; even two CORBA-compliant platforms cannot collaborate unless they use the same IDL.

The DX NetOps Spectrum IDL is based on SSAPI and IHAPI to keep it close to the concepts familiar to DX NetOps Spectrum legacy application developers. The DX NetOps Spectrum IDL data model is similar to that of earlier toolkits and all of the following continue to exist:

- Models with attributes
- Model types
- Associations between models
- Relations
- Alarms
- Events
- Statistics

The Development API data model is as follows:



The Development API Classes

The SPECTRUM Development API includes interface and data structure classes. Interface classes implement the DX NetOps Spectrum services and data structure classes include parameter, helper, and global classes.

Interface Classes

SpectroSERVER and ArchMgr contain the following classes. You need object references to access these classes.

- **CsCModelDomain**
  Defines the model domain. Each instance of this class contains models and associations of a model domain. Each SpectroSERVER implements a model domain.

This is a persistent CORBA service and the supporting IDL file is cscmodeldomain.idl.

- **CsCTypeCatalog**
  Provides access to a type catalog, which contains model types and relations. All the SpectroSERVERs of a release at a site should have the same type catalog, but differences may occur. Each type catalog has its own instance of this service, with each SpectroSERVER implementing a type catalog.
  This is a transient CORBA service and the supporting IDL file is csctypecatalog.idl.
- **CsCModelType**
  Provides access to a model type and its attributes. A model type is a template for creating and classifying models. Each model type has its own instance of this service.
  This is a transient CORBA service and the supporting IDL file is cscmodeltype.idl.
- **CsCRelation**
  Provides access to a relation and its rules. A relation is a template for creating and classifying associations. Each relation has its own instance of this service.
  This is a transient CORBA service and the supporting IDL file is cscrelation.idl.
- **CsCAlarmDomain**
  Provides access to an alarm domain. An alarm domain contains alarms for models of a model domain. Each alarm domain has its own instance of this service, with each SpectroSERVER implementing one alarm domain.
  This is a persistent CORBA service and the supporting IDL file is cscalarmdomain.idl.
- **CsCEventDomain**
  Provides access to an event domain. An event domain contains events for the models of a model domain. Each event domain has its own instance of this service with each ArchMgr implementing one event domain.
  This is a persistent CORBA service and the supporting IDL file is csceventdomain.idl.

You can find the Java Docs for these classes by opening the file <*$SPECROOT*>/SDK/docs/SSORB/index.html and clicking com.aprisma.spectrum.core.idl in the Packages list. The IDL files for these classes are available in <*$SPECROOT*>/SDK/examples/SSORB/idl.

Additional IDLs

The <*$SPECROOT*>/SDK/examples/SSORB/idl includes the following IDL files but the services they support are not considered CORBA services:

- **cscattribute.idl**
  Defines attribute types and attribute values that other IDLs reuse.
- **cscdeveloper.idl**
  Defines the structures that management module developers use in the type catalog.
- **cscerror.idl**
  Defines the error codes that other IDLs reuse.
- **cscexception.idl**
  Defines the DX NetOps Spectrum exception that other IDLs reuse because most methods raise this exception.

Data Structure Classes

Data structure classes are of two types: parameter and helper.

- **Parameter Class**
  Transfers data. Your client programs instantiate these classes and pass them to interface methods. The methods also return instances of these classes to your programs. Examples include CsCModelPropList, CsCMTypePropList, and CsCAttrValList.
- **Helper Class**
  Builds and unwraps complex parameter class objects.

You can find the Java docs for these classes by clicking the link com.aprisma.spectrum.core.util in the file <*$SPECROOT*>/SDK/docs/SSORB/index.html. Examples include CsCorbaValueHelper and CsCAttrFilterHelper.

Global Classes

Global classes contain standard security and login objects that DX NetOps Spectrum uses. You can find the Java docs for these classes in <*$SPECROOT*>/SDK/examples/SSORB/idl.

Java Development Specifics

The following JAR files are necessary for Java development with DX NetOps Spectrum SDK, where *xx* is the version of DX NetOps Spectrum that you are running. For example, if you are running 10.3, global*xx*.jar would be global103.jar.

- global*xx*.jar
- ssorb*xx*.jar
- ssorbutil*xx*.jar
- util*xx*.jar
- utilapp*xx*.jar
- utilnet*xx*.jar
- utilsrv*xx*.jar
- vbhelper*xx*.jar
- vbjorb.jar
- bsec.jar
- lm.jar
- cryptojFIPS.jar
- sanct6.jar (Linux only)
- sanctuary.jar (Linux only)

The corresponding directory location is <*$SPECROOT*>/lib, and the JAR file location should be set by the CLASSPATH environment variable.

Example: Setting the CLASSPATH environment variable

The following are examples of setting the CLASSPATH variable for different platforms. Replace the following variables accordingly:

<*$SPECROOT*>

Specify the complete path to your DX NetOps Spectrum installation directory.

xx

Specify the version of DX NetOps Spectrum that you are running. For example, if you are running 10.3, global*xx*.jar would be global103.jar.

- You can set the CLASSPATH on Windows as follows:

CLASSPATH=<*$SPECROOT*>\lib\global*xx*.jar;<*$SPECROOT*>\lib\ssorb*xx*.jar;<*$SPECROOT*>\lib
\ssorbutil*xx*.jar;<*$SPECROOT*>\lib\vbhelper*xx*.jar;<*$SPECROOT*>\lib\vbjorb.jar;<*$SPECROOT*>\lib
\vbsec.jar;<*$SPECROOT*>\lib\util*xx*.jar;<*$SPECROOT*>\lib\utilnet*xx*.jar;<*$SPECROOT*>\lib\utilsrv*xx*.jar;<*$SPECROOT*>
\lib\lm.jar;<*$SPECROOT*>\lib\cryptojFIPS.jar;<*$SPECROO*T>\lib\utilapp*xx*.jar;.;

- You can set the CLASSPATH on Linux as follows:

CLASSPATH=<*$SPECROOT*>/lib/global*xx*.jar:<*$SPECROOT*>/lib/ssorb*xx*.jar:<*$SPECROOT*>/lib/
ssorbutil*xx*.jar:<*$SPECROOT*>/lib/vbhelper*xx*.jar:<$SPECROOT>/lib/vbjorb.jar:<*$SPECROOT*>/lib/
vbsec.jar:<*$SPECROOT*>/lib/util*xx*.jar:<*$SPECROOT*>/lib/utilnet*xx*.jar:<*$SPECROOT*>/lib/utilsrv*xx*.jar:<*$SPECROOT*>/
lib/lm.jar:<*$SPECROOT*>\lib\cryptojFIPS.jar:<*$SPECROOT*>/lib/sanct6.jar:<*$SPECROOT*>/lib/
sanctuary.jar:<*$SPECROOT*>/lib/utilapp*xx*.jar:.:

C++ Development on Windows

The following table lists the library and dynamic link libraries necessary for C++ development on Windows. The corresponding directory location is <*$SPECROOT/lib*>.

Library Files	SDK
libGlobl.lib libssorb.lib ibssorbutil.lib	DX NetOps Spectrum
orbcore_r.lib cosnm_r_80.lib vport_r_80.lib vdlog_r_80.lib	Borland VisiBroker

The following table shows the corresponding header file locations:

Library	Header File Location
libGlobl.dll	<*$SPECROOT*>\SDK\include\GLOBL
libssorb.dll	<*$SPECROOT*>\SDK\include\SSORB\idl
libssorbutil.dll	<*$SPECROOT*>\SDK\include\SSORB\util
orbcore_r_80.dll cosnm_r_80.dll vport_r_80.dll vdlog_r_80.dll	Provided by Borland VisiBroker SDK (Not included in DX NetOps Spectrum CORBA toolkit.)

C++ Development on Linux

The following table lists the shared objects and libraries necessary for C++ development on Linux. The corresponding directory location is <*$SPECROOT/lib*>.

Library File	SDK
libGlobl (so, so.1) libssorb (so, so.1) libssorbutil (so, so.1) libPort (so, so.1) libVPapi (so, so.1) libWrappers (so, so.1)	CA Spectrum
liborbcore_r.so libcosnm_r.so libvport_r.so libvdlog_r.so	Borland VisiBroker

The following table shows the corresponding header file locations:

Shared Object/Library File	Header File Location
libGlobl.so	<*$SPECROOT*>/SDK/include/GLOBL
libssorb.so	<*$SPECROOT*>/SDK/include/SSORB/idl
libssorbutil.so	<*$SPECROOT*>/SDK/include/SSORB/util
libPort.so	<*$SPECROOT*>/SDK/include/PORT
libVPapi.so	<*$SPECROOT*>/SDK/include/VPAPI
liborbcore_r (so, so.8.0) libcosnm_r (so, so.8.0) libvport_r (so, so.8.0) libvdlog_r (so, so.8.0)	Provided by Borland VisiBroker SDK (Not included in DX NetOps Spectrum CORBA toolkit.)

Standard Naming Service and VisiBroker ORB

Client applications use the Standard Naming Service, resulting in ORB independence. With ORB independence, the only requirement is that the ORB implement the Standard Naming Service.

ORB Examples

The DX NetOps Spectrum Developer Kit includes CORBA client applications built with ORBs from other vendors that use the standard naming service. The following table lists the ORBs and directory locations for different development platforms:

ORB	Platform and Development	Directory Location
Mico	Linux, C++	<*$SPECROOT*>/SDK/examples/SSORB/nameserv/cc/mico
Java	Windows, Java	<*$SPECROOT*>/SDK/examples/SSORB/nameserv/java/sun

DX NetOps Spectrum IDL

The IDL files are an essential part of SPECTRUM Development API (which also includes the helper classes, examples, and documentation). The IDL files define the interfaces to various DX NetOps Spectrum services (model domain, event domain, and so on) and are available in the following directory:

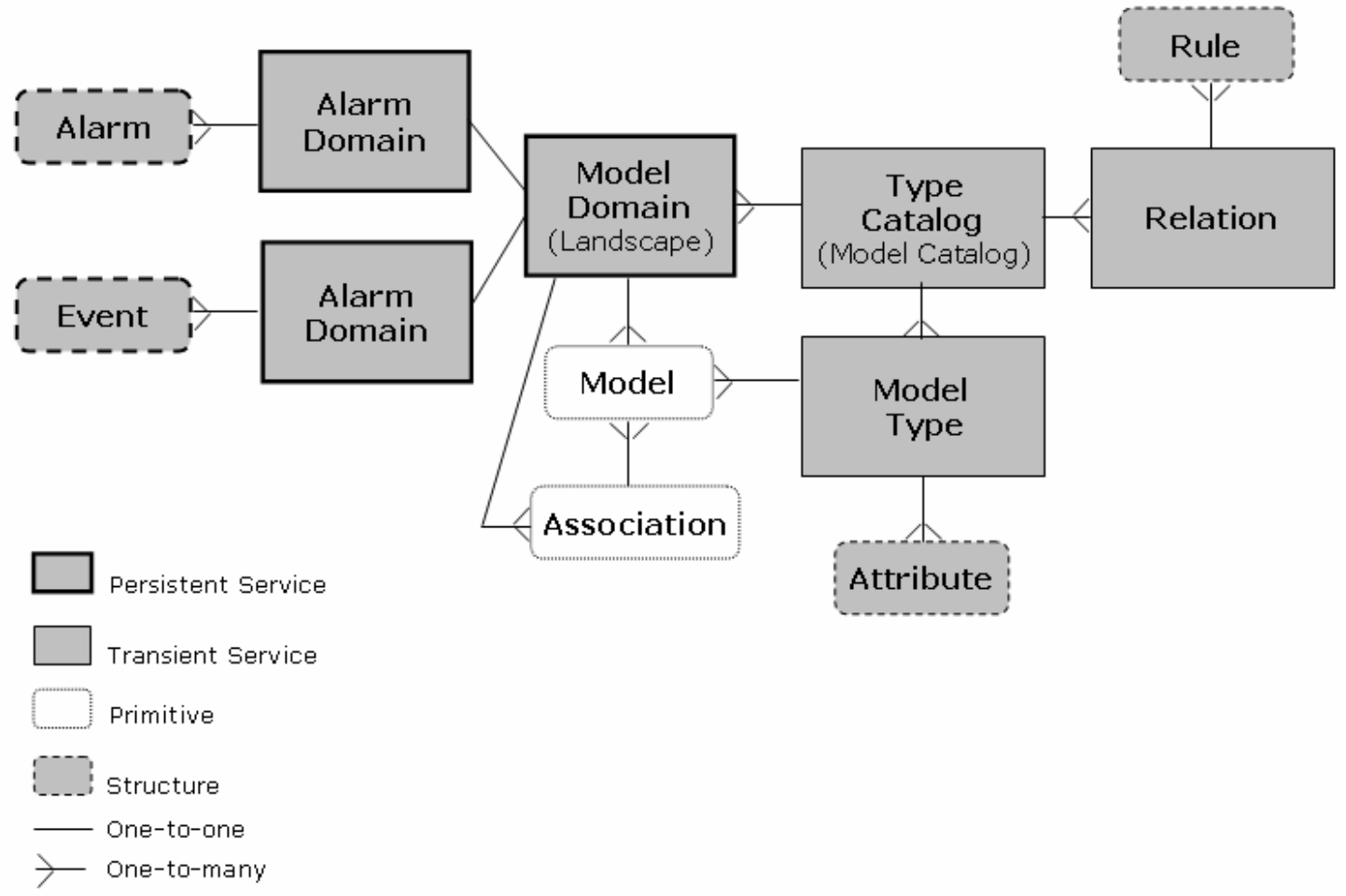
```
<$SPECROOT>/SDK/examples/SSORB/idl
```

> **WARNING**
> DX NetOps Spectrum IDL is proprietary; even two CORBA-compliant platforms cannot collaborate unless they use the same IDL.

DX NetOps Spectrum IDL is based on SSAPI and IHAPI to keep it close to the concepts familiar to DX NetOps Spectrum legacy application developers. The DX NetOps Spectrum IDL data model is similar to that of earlier toolkits and all of the following continue to exist:

- Models with attributes
- Model types
- Associations between models
- Relations
- Alarms
- Events
- Statistics

The Development API data model is as follows:



Development API Classes

The SPECTRUM Development API includes interface and data structure classes. Interface classes implement the DX NetOps Spectrum services and data structure classes include parameter, helper, and global classes.

Interface Classes

SpectroSERVER and ArchMgr contain the following classes. You need object references to access these classes.

- **CsCModelDomain**
  Defines the model domain. Each instance of this class contains models and associations of a model domain. Each SpectroSERVER implements a model domain.

This is a persistent CORBA service and the supporting IDL file is cscmodeldomain.idl.

- **CsCTypeCatalog**
  Provides access to a type catalog, which contains model types and relations. All the SpectroSERVERs of a release at a site should have the same type catalog, but differences may occur. Each type catalog has its own instance of this service, with each SpectroSERVER implementing a type catalog.
  This is a transient CORBA service and the supporting IDL file is csctypecatalog.idl.
- **CsCModelType**
  Provides access to a model type and its attributes. A model type is a template for creating and classifying models. Each model type has its own instance of this service.
  This is a transient CORBA service and the supporting IDL file is cscmodeltype.idl.
- **CsCRelation**
  Provides access to a relation and its rules. A relation is a template for creating and classifying associations. Each relation has its own instance of this service.
  This is a transient CORBA service and the supporting IDL file is cscrelation.idl.
- **CsCAlarmDomain**
  Provides access to an alarm domain. An alarm domain contains alarms for models of a model domain. Each alarm domain has its own instance of this service, with each SpectroSERVER implementing one alarm domain.
  This is a persistent CORBA service and the supporting IDL file is cscalarmdomain.idl.
- **CsCEventDomain**
  Provides access to an event domain. An event domain contains events for the models of a model domain. Each event domain has its own instance of this service with each ArchMgr implementing one event domain.
  This is a persistent CORBA service and the supporting IDL file is csceventdomain.idl.

You can find the Java Docs for these classes by opening the file <*$SPECROOT*>/SDK/docs/SSORB/index.html and clicking com.aprisma.spectrum.core.idl in the Packages list. The IDL files for these classes are available in <*$SPECROOT*>/SDK/examples/SSORB/idl.

Additional IDLs

The <*$SPECROOT*>/SDK/examples/SSORB/idl includes the following IDL files but the services they support are not considered CORBA services:

- **cscattribute.idl**
  Defines attribute types and attribute values that other IDLs reuse.
- **cscdeveloper.idl**
  Defines the structures that management module developers use in the type catalog.
- **cscerror.idl**
  Defines the error codes that other IDLs reuse.
- **cscexception.idl**
  Defines the DX NetOps Spectrum exception that other IDLs reuse because most methods raise this exception.

Data Structure Classes

Data structure classes are of two types: parameter and helper.

- **Parameter Class**
  Transfers data. Your client programs instantiate these classes and pass them to interface methods. The methods also return instances of these classes to your programs. Examples include CsCModelPropList, CsCMTypePropList, and CsCAttrValList.
- **Helper Class**
  Builds and unwraps complex parameter class objects.

You can find the Java docs for these classes by clicking the link com.aprisma.spectrum.core.util in the file <*$SPECROOT*>/SDK/docs/SSORB/index.html. Examples include CsCorbaValueHelper and CsCAttrFilterHelper.

Global Classes

Global classes contain standard security and login objects that DX NetOps Spectrum uses. You can find the Java docs for these classes in <*$SPECROOT*>/SDK/examples/SSORB/idl.

Product-Related Services

The following sections explain how application developers can use DX NetOps Spectrum services.

Security

Security is a major concern because the development API accepts Java applet calls over the web. To continue to provide the same level of security as earlier DX NetOps Spectrum versions, the API includes user passwords. The API also includes helper classes to set the login and maintain a secure connection.

Error Reporting

CORBA reports errors through exceptions. Error reporting through exceptions is an all-or-nothing approach because no other result is possible. This approach is not helpful for partial failures, when only a few items of a request fail. Also, it is not optimal to raise an exception for every failure. The CORBA approach is to use exceptions for complete failures and report partial failures with error codes in result lists. Because it is not optimal to search an entire list to see if any item failed, result lists also provide an overall error code to indicate partial failures.

Documentation and Examples

The SPECTRUM Development API provides a well-commented IDL as the primary source for application developers. You can also use the IDL to create Java doc web pages through Java-generated stubs.

The toolkit also includes working examples and test programs for which Java versions are available in <*$SPECROOT*>/SDK/examples/SSORB/DevelopmentAPI.

Additional VisiBroker Arguments

The *VisiBroker Programmer's Reference* includes additional VisiBroker arguments that the SPECTRUM Development API provides.

> **NOTE**
> For more information, access the following web address: http://info.borland.com/techpubs.

Developer Kit

Access DX NetOps Spectrum Java Documentation

The Development API Java documentation is an important part of the SPECTRUM Development API.

To access the Development API Java documentation

1. Open <*$SPECROOT*>/SDK/docs/SSORB/index.html in your browser and click com.aprisma.spectrum.core.idl under Packages.
   The Interface Summary and Class Summary appear.
2. Click a link under the summaries.
   The interface or class description appears.
3. Scroll past the description.
   The field, constructor, and method summaries appear. Method descriptions appear under Method Summary.

4. Click a method link.
   The Method Detail section appears.

   NOTE
   The code samples are available in <*$SPECROOT*>/SDK/examples/SSORB/DevelopmentAPI.

Java Permissions

You need to set permissions as follows:

- The java.util.PropertyPermission: Set both read and write permissions.
- The java.lang.RuntimePermission: Use setContextClassLoader.

You need to set additional permissions depending on the application you develop. For example, to read a properties file or write to a log file, you need to set the file permission.

Model Domain ID

Each SpectroSERVER in a DX NetOps Spectrum environment has a model domain with a unique ID and name as follows:

- The unique ID is the entity named *landscape handle* in DX NetOps Spectrum terminology.
- The unique name is the DNS or network name of the SpectroSERVER host system.
- The GetDomainID program uses a model domain name to get and display the model domain ID. In doing so, GetDomainID also performs the following functions:
- Bind to a DX NetOps Spectrum CORBA service
- (Optional) Set security

All programs that you write should perform these functions.

Service Binding

CORBA provides a helper object to simplify the binding process for DX NetOps Spectrum. The CORBAHelper class encapsulates a CORBA ORB and provides helper methods for locating and advertising CORBA objects. The purpose of this class is to hide ORB vendor-specific methods of advertising CORBA objects. CORBAHelper is an abstract class; you can obtain a vendor-specific implementation through one of the static methods.

NOTE
You need to import the CORBAHelper class from the com.aprisma.util.corba package.

The CORBAHelper class defines the following properties:

- **LOGIN_USERNAME (user.name)**
  Defines the user name.

  NOTE
  user.name is set in the System Properties at startup, but you can override it in Supplied Properties.
- **LOGIN_APPNAME (CORBAHelper.AppName)**
  Defines the application name.
- **LOGIN_SUBAPPNAME (CORBAHelper.SubAppName)**
  Defines the sub-application name.
- **LOGIN_APPVERSION (CORBAHelper.AppVersion)**
  Defines the version of the application.
- **LOGIN_PASSWORD (CORBAHelper.password)**
  Defines the user password.

NOTE
For all the previous properties, for SSORB communication, the corresponding CsApplicationInfo field is set.

- **LOGIN_CONNECTIONTYPE (CORBAHelper.ConnectionType)**
  Defines the connection type as enumerated in CsApplicationInfo.
- **NAMING_SERVICE_PORT**
  Set this property only if your Naming Service is configured to run on a port other than 14006 (the SPECTRUM default).

An application can set these properties through *System.setProperty* or a Properties object supplied to the init method. The defined values are in parentheses.

Binding Implementation

The following example shows binding implementation. Because a user is not specified, the helper defaults to the user running the program. It also shows the majority of code enclosed in a try catch statement. The methods getObjectImplementation and getModelDomainID throw exceptions if you refer to the Java Docs.

NOTE
Code samples in the later sections of this section leave out the *try* and *catch* statements for brevity, but you must include them in your programs.

```
import com.aprisma.spectrum.core.idl.*;
import com.aprisma.spectrum.core.util.*;
import com.aprisma.util.corba.* ;
import java.io.*

public class ExampleBind
{
    public static void main(String[] args)
    {
        try
        {
            //*************************************************
            // Initialize.
            //*************************************************
            String domainName = new String("mySpectroServer");
            // Construct helper
            CORBAHelper helper = CORBAHelper.getHelperImpl () ;
            if (helper.init(null, null))
            {
                CsCModelDomain md = (CsCModelDomain)
                helper.getObjectImplementation (
                    CsCModelDomain.class, domainName ) ;
                //*********************************************
                //Get the Model Domain ID
                //*********************************************
                int ID = md.getModelDomainID();
                System.out.println("0x" + Integer.toHexString(ID) );
            }
        }
        catch(Throwable e)
        {
            System.out.println(e);
        }
    }
```

```
}
```

Binding Implementation with User Specification

The following example shows how to implement binding with user specification.

```
import com.aprisma.spectrum.core.idl.*;
import com.aprisma.spectrum.core.util.*;
import com.aprisma.util.corba.* ;
import java.io.*;
public class ExampleBind
{
    public static void main(String[] args)
    {
        try
        {
            //*************************************************
            // Initialize.
            //*************************************************
            String domainName = new String("mySpectroServer");
            // Construct helper
            CORBAHelper helper = CORBAHelper.getHelperImpl () ;
            Properties props = new Properties () ;
            props.setProperty ( "user.name", "JonDoe" ) ;
            if (helper.init(null, props))
            {
                CsCModelDomain md = (CsCModelDomain)
                helper.getObjectImplementation (
                    CsCModelDomain.class, domainName ) ;
                //*************************************************
                //Get the Model Domain ID
                //*************************************************
                int ID = md.getModelDomainID();
                System.out.println( "0x" + Integer.toHexString(ID) );
            }
        }
        catch(Throwable e)
        {
            System.out.println(e);
        }
    }
}
```

How To Disable Automatic Reconnect

If connection to a service is lost, reconnecting starts automatically until the connection is regained. Because reconnecting is unnecessary at times, you can turn off the function by passing FALSE as the third parameter to the getObjectImplementation method, as follows:

```
String domainName = new String("mySpectroServer");
// Construct helper
CORBAHelper helper = CORBAHelper.getHelperImpl () ;
Properties props = new Properties () ;
props.setProperty ( "user.name", "JonDoe" ) ;
```

```
if (helper.init(null, props))
{
      CsCModelDomain md = (CsCModelDomain)
        helper.getObjectImplementation (
           CsCModelDomain.class, domainName, FALSE ) ;
}
```

NOTE
You can also disable automatic reconnect by setting java -Dvbroker.orb.cacheDSQuery=false ClientApp.

Security Levels

Each DX NetOps Spectrum CORBA service requires a different level of security. For example, a financial database object might require an encrypted user name and password. A DX NetOps Spectrum model domain requires an unencrypted user name. Security is preset for the user and the host in which the Java application is implemented, but you can change the settings.

If this ORB needs to be set to a user and host other than the default values, the Properties class can be set and passed to *init* as shown in the example Binding Implementation with User Specification.

Type Catalog Interface

The type catalog interface, CsCTypeCatalog, contains information about the Developer IDs, Model Types, Relations, Rules, and Attribute properties. The CsCModelDomain can create models only against the model types that exist in this type catalog. This non-persistent interface does not allow the creation or destruction of new model types, relations, rules, or attributes. But it allows reading of properties for developer information, model types, relations, rules, and attributes and the modification of default values of attributes.

Retrieve the Developer ID Example

You can retrieve the developer ID and prefix from the interface. The getDevProperties method call returns a CsCDevProperties data structure.

NOTE
The Model Type Editor section explains the concept of Developer ID and how to use it with Model Type Editor.

Example: Retrieve a developer ID

The following example shows how to retrieve a developer ID:

```
import com.aprisma.spectrum.core.idl.*;
import com.aprisma.spectrum.core.util.*;
import com.aprisma.util.corba.* ;
import com.aprisma.spectrum.core.idl.CsCDeveloper.*;
import java.io.*;
public class GetTypeCatalogInfo
{
    public static void main(String[] args)
    {
        try
        {
            //*************************************************
            // Initialize.
            //*************************************************
            String domainName = new String("mySpectroServer");
            // Construct helper
```

```
            CORBAHelper helper = CORBAHelper.getHelperImpl () ;
            if (helper.init(null, null))
            {
                CsCModelDomain md = (CsCModelDomain)
                helper.getObjectImplementation (
                    CsCModelDomain.class, domainName ) ;
                //*********************************************
                //Get the Retrieve the other Domains
                //*********************************************
                CsCTypeCatalog tc = md.getTypeCatalog();
                CsCDevProperties devprop =
                              tc.getDevProperties(0x210000);
                System.out.println("Name:" + devprop.name );
                System.out.println("Prefix:" + devprop.prefix );
            }
        }
        catch(Throwable e)
        {
            System.out.println(e);
        }
    }
}
```

This example requires the following objects and methods:

- Package com.aprisma.spectrum.core.idl
  - Interfaces
    - CsCModelDomain
    - CsCTypeCatalog
  - Classes
    - CsCModelDomainHelper
- Package com.aprisma.spectrum.core.idl.CsCDeveloper
  - Classes
    - CsCDevProperties

Model Domain Interface

The CsCModelDomain class provides several methods for creating DX NetOps Spectrum models. These methods are inherited from CsCModelDomainOperations but need to be called from the CsCModelDomain class interface.

User Security

Most client applications you develop using the SPECTRUM Development API use the services of one or more of the following classes:

- **CsCTypeCatalog**
  Provides access to the Modeling Catalog.
- **CsCModelDomain**
  Provides access to the models, associations and attributes.
- **CsCEventDomain**
  Provides access to events.
- **CsCAlarmDomain**
  Provides access to alarms.

CsCModelDomain, CsCEventDomain, and CsCAlarmDomain are persistent interfaces, that is, interfaces that persist across a SpectroSERVER restart. Once the application binds to the ORB and retrieves the interface reference, the reference is valid across SpectroSERVER and ArchMgr restarts.

Obtain interface domain references to these services by first binding to the model domain and invoking the appropriate CsCModelDomain method as follows:

```
import com.aprisma.spectrum.core.idl.*;
import com.aprisma.spectrum.core.util.*;
import com.aprisma.util.corba.* ;
import java.io.*

public class ExampleBind
{
    public static void main(String[] args)
    {
        try
        {
            //*************************************************
            // Initialize.
            //*************************************************
            String domainName = new String("mySpectroServer");
            // Construct helper
            CORBAHelper helper = CORBAHelper.getHelperImpl () ;
            if (helper.init(null, null))
            {
                CsCModelDomain md = (CsCModelDomain)
                    helper.getObjectImplementation (
                        CsCModelDomain.class, domainName ) ;
                //*********************************************
                //Get the Retrieve the other Domains
                //*********************************************
                CsCTypeCatalog tc = md.getTypeCatalog();
                CsCEventDomain ed = (CsCEventDomain)
                    helper.getObjectImplementation(
                        CsCEventDomain.class, domainName );
                CsCAlarmDomain ad = md.getAlarmDomain();
            }
        }
        catch(Throwable e)
        {
            System.out.println(e);
        }
    }
}
```

Model IDs and Model Handles

Model IDs are different from model handles. A model handle is a 32-bit number in which the high order 12 bits make up the landscape and the rest of the 20 bits make up the model ID.

Method calls frequently require model handles. You can make up the handles by joining the landscape and the model ID together.

Device Modeling

Creating a model is different from modeling a device. Creating involves telling DX NetOps Spectrum to create a model that usually represents a simple, unaddressable entity like a container or user. When you model a device, you describe the device (by IP address) to DX NetOps Spectrum, let the server examine the device, and create a set of interrelated models that represent the components and complexities. The previous section explained how to create a network model. The following section explains how to model a device and place it in a network container model.

You can model a device by invoking a single method as follows:

```
CsCModelProperties mp = md. createModelByIP(
ip, // IP address of device.
community, // CsCValue containing the SNMP community string.
5000, // time to wait for ping reply.
3, // retry count
161 // agent SNMP Port
```

The args[i] variables represent command line arguments passed to a dummy program that contains all the code snippets. The parameters in lines 2 through 6 are as follows:

- **ip**
  Defines the IP address of the target device. You can use a helper class to convert an IP address from string format (for example, 192.168.1.81) to int[] form as follows:
  ```
  CsCorbaValueHelper helper = new CsCorbaValueHelper();
  CsCValue ipValue = valueHelper.parseInternetAddressValue(argv[0]);
  byte[] ip = ipValue.internetAddress();
  ```
  NOTE
  The IP address can either be IPv4 or IPv6.
- **community**
  Defines a CsCValue containing the SNMP community string that DX NetOps Spectrum uses to satisfy security requirements when it contacts the device.
- **5000**
  Defines the time (in milliseconds) that DX NetOps Spectrum waits between each attempt to contact the device.
- **3**
  Defines the number of times DX NetOps Spectrum attempts to ping the device before giving up.
- **161**
  Defines the SNMP Agent Port.

This example requires the following objects and methods:

- Package com.aprisma.spectrum.core.idl
  - Interfaces
    - CsCModelDomain
    - CsCTypeCatalog
    - CsCRelation
    - CsCModelType
  - Classes
    - CsCModelDomainHelper
- Package com.aprisma.spectrum.core.idl.CsCModelPackage
  - Classes
    - CsCModelProperties
- Package com.aprisma.spectrum.core.util
  - Classes

- CsCorbaValueHelper

Model Types

DX NetOps Spectrum model types are related to models in the same way that classes are related to objects in object-oriented programming. Model types are the templates, while models are instances. The following sections show how to explore a model type, that is, retrieve its object reference, its position in the type hierarchy, its attributes, and properties.

Retrieve Model Type Object Reference Example

The following example shows how to retrieve the object reference for a given model type ID:

```
Integer mtypeID = Integer.decode(args[0]);
CsCTypeCatalog tc = md.getTypeCatalog();
CsCModelType mt = tc.getModelType(mtypeID.intValue());
```

The first line gets an mtype ID from the command line. The second line gets an object reference to the type catalog. The third line gets an object reference to the particular model type.

Use the following code to retrieve the object reference for a given model type name:

```
String mtypeName = args[0];
CsCTypeCatalog tc = md.getTypeCatalog();
CsCMTypePropList mtpl = tc.getMTypePropListByName(mtypeName);
```

The first line gets a model type name from the command line. The second line gets an object reference to the type catalog. The third line gets a list of all model types that have the specified name. In most cases, the list contains only one member.

Model Type Hierarchy

Once you have a CsCModelType object reference (mt in the following example), you can use the following methods to determine the model type's base and derived relatives.

```
CsCMTypePropList bmtpl = mt.getBaseMTypePropList();
CsCMTypePropList dmtpl = mt.getDerivedMTypePropList();
```

Model Type Properties

A model type has the following Boolean properties:

- **instantiable**
  Indicates that you can create instances from this model type.
- **underivable**
  Indicates that model types can not be derived from this model type.
- **noDestroy**
  Indicates that you cannot destroy models of this type.
- **unique**
  Indicates that at most one model of this model type can exist.
- **required**
  Indicates that at least one model of this type is required.

How To Read Model Type Properties

Use the following process to read a model type's properties:

1. Obtain a CsCMTypeProperties object to represent a model type. (CsCTypeCatalog and CsCModelType classes have methods to return such an object.)
2. Access the CsCMTypeProperties member variable flags (of type CsCMTypeFlags) that contains a Boolean variable for each property.

Example: Read a model type's properties:

```
Integer mtypeID = Integer.decode(args[0]);
CsCTypeCatalog tc = md.getTypeCatalog();
CsCMTypeProperties mtp = tc.getMTypeProperties
(mtypeID.intValue());
CsCMTypeFlags flags = mtp.flags;
```

The first line gets a model type ID from the command line. The second line gets an object reference to the type catalog. The third and fourth lines get a model type properties object. The fifth line extracts a model type flags object.

Read Model Type Properties Example

The following example shows how to retrieve model type information:

```
import com.aprisma.spectrum.core.idl.*;
import com.aprisma.spectrum.core.util.*;
import com.aprisma.util.corba.* ;
import com.aprisma.spectrum.core.idl.CsCModelTypePackage.*;
import java.io.*;
public class GetTypeCatalogInfo
{
    public static void main(String[] args)
    {
        Integer mtid = Integer.decode( args[0] );

        try
        {
            //*************************************************
            // Initialize.
            //*************************************************
            String domainName = new String("mySpectroServer");
            // Construct helper
            CORBAHelper helper = CORBAHelper.getHelperImpl () ;
            if (helper.init(null, null))
            {
                CsCModelDomain md = (CsCModelDomain)
                helper.getObjectImplementation (
                    CsCModelDomain.class, domainName ) ;
                //*********************************************
                //Get the Retrieve the other Domains
                //*********************************************
                CsCTypeCatalog tc = md.getTypeCatalog();
                CsCModelType mt = tc.getModelType( mtid.intValue() );
                CsCMTypeProperties mtprop = mt.getMTypeProperties();
                CsCMTypeFlags mtflags = mtprop.flags;
                System.out.println("Name:" + mtprop.name );
```

```
                System.out.println("Underivable:"
                                   + mtflags.underivable );
            }
        }
        catch(Throwable e)
        {
            System.out.println(e);
        }
    }
}
```

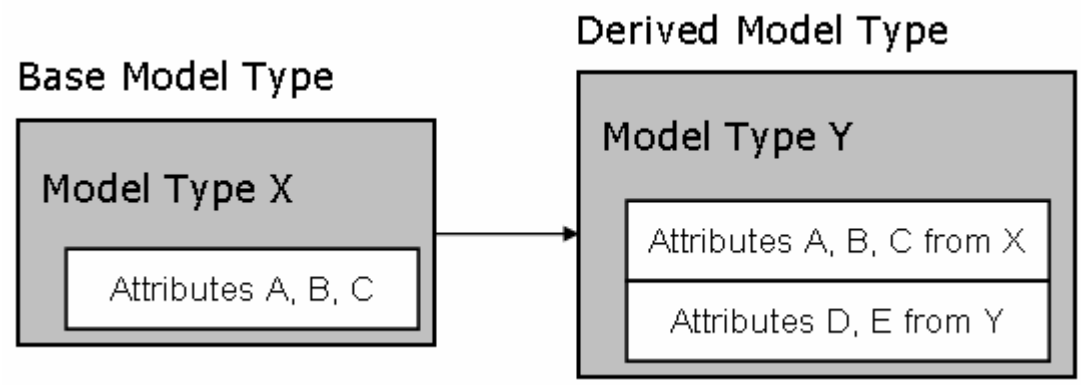
The previous example requires the following objects and methods:

- Package com.aprisma.spectrum.core.idl
  - Interfaces
    - CsCModelDomain
    - CsCTypeCatalog
    - CsCModelType
  - Classes
    - CsCModelDomainHelper
- Package com.aprisma.spectrum.core.idl.CsCModelTypePackage
  - Classes
    - CsCMTypeProperties
    - CsCMTypeFlags

Model Type Attributes

DX NetOps Spectrum model types have attributes associated with them. Attributes describe the model capabilities and reflect internal storage of some data type and external MIB object information. Both basic and derived model types have attributes.

The following illustration shows a base model type and its derived model type:

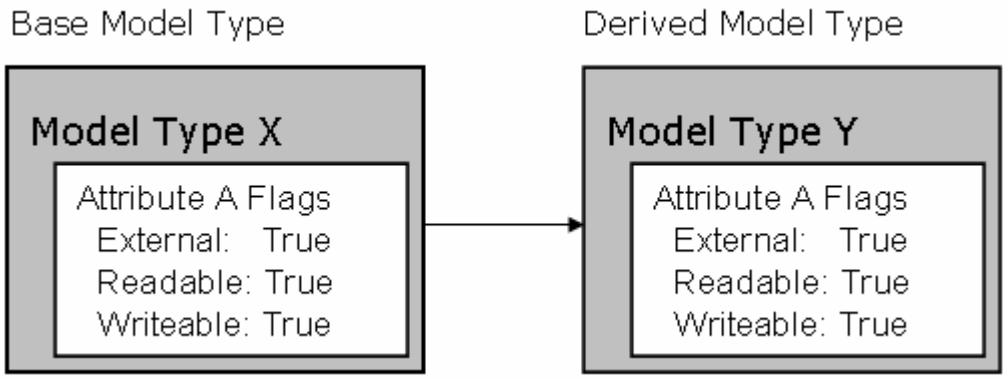


Model type attributes have their own metadata, attribute modifiable properties and attribute non-modifiable properties.

Non-Modifiable Properties

Model type attribute properties remain the same across base and derived model types. That is, the properties cannot be modified for derived model types.

The following illustration shows an example of non-modifiable properties in a base model type and its derived model type:



The non-modifiable properties include the following:

- **external**
  Indicates that the attribute is external to DX NetOps Spectrum and must be obtained through a protocol such as SNMP.
- **readable**
  Indicates that the attribute is readable through an application.
- **writable**
  Indicates that the attribute is writable through an application.
- **shared**
  Indicates that the attribute has only one shared value for all models of a model type.
- **list**
  Indicates that the attribute is a list. The list attributes usually form a column of an attribute table.
- **guaranteed**
  Indicates that the attribute is guaranteed not to be removed by the management module developer.
- **global**
  Reserved for internal use.
- **preserveLegacyValue**
  Indicates that the attribute value in the database is not overwritten by imported database catalog (.e or .xml) files. If you customize model types for your system requirements and set this flag, the changed model type attribute values are preserved if model types are updated by subsequent release versions.

How To Read Attribute Properties

You can read attribute properties as follows:

1. Obtain a CsCAttrProperties object representing a single attribute.
2. Access the Flags member variable (of type CsCAttrFlags) of the CsCAttrProperties class, which contains one Boolean variable for each of these eight characteristics.

Example: Read Attribute Properties

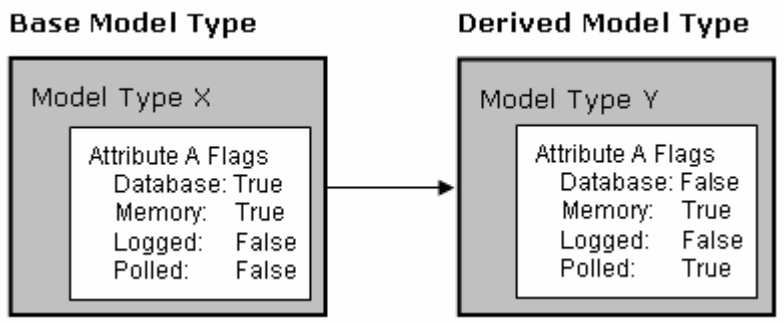
```
Integer attrID = Integer.decode(args[0]);
CsCTypeCatalog tc = md.getTypeCatalog();
CsCAttrProperties ap =
tc.getAttrProperties(attrID.intValue());
CsCAttrFlags flags = ap.flags;
```

The first line gets an attrID from the command line; the second line gets an object reference to the type catalog; the third and fourth lines get an attribute properties object; the fifth line extracts an attribute flags object.

Modifiable Properties

An attribute can have modifiable properties that can be modified at the derived type level. You can also refer to these properties as extended flags.

The following illustration shows an example of modifiable properties in a base model type and its derived model type:



The modifiable properties include the following:

- **polled**
  Indicates that the attribute value is read periodically. The polling interval sets the frequency.
- **logged**
  Indicates that the attribute value is read periodically and saved in the DX NetOps Spectrum statistics log. The logging interval sets the frequency.
- **memory**
  Indicates that, for frequently-read attributes, the attribute value is stored in memory.
- **database**
  Indicates that the attribute value is stored in the database. This is used for attributes that must be preserved between SpectroSERVER restarts.
- **oidPrefix**
  Indicates the first part of the object identifier (OID) in the MIB of an agent that supports a network entity. It is the branch of the MIB tree where a table attribute begins.
- **oidReference**
  Indicates an attribute ID whose value is an OID suffix. This object identifier supplements the OID prefix so as to fully identify the object in the MIB tree. The rows of a table attribute are represented using this value as the suffix. A zero value indicates that the OID reference is not specified for this model type attribute.
- **pollingGroup**
  Indicates the polling group identifier. Attributes of a polling group are polled and logged together. Each model type can have up to 256 polling groups. Each polling group is identified by a number from 0 to 255.
- **pollingInterval**
  Indicates the time, in seconds, after which an attribute value is updated to the latest value.

Model type attribute properties differ from attribute properties as follows:

- Model type attribute properties are not limited to Boolean values.
- Second, they apply to an attribute for a particular model type.

How To Read Model Type Attribute Properties

You can read model type attribute properties as follows:

1. First obtain a CsCModelType object reference representing a single model type.
2. Obtain a CsCMTypeAttrProperties object and access several data members of the object, which comprise the model type attribute properties.

Example: Read Model Type Attribute Properties

```
Integer mtypeID = Integer.decode(args[0]);
Integer attrID = Integer.decode(args[1]);
CsCModelType mt = tc.getModelType(mtypeID.intValue());
CsCMTypeAttrProperties mtap = mt.getMTypeAttrProperties(attrID.intValue());
CsCMTypeAttrFlags flags = mtap.flags;
int[] oidPrefix = mtap.oidPrefix;
int oidReference = mtap.oidReference;
int pollingInterval = mtap.pollingInterval;
int loggingInterval = mtap.loggingInterval;
int pollingGroup = mtap.pollingGroup;
```

The first and second lines get the model type ID and attribute ID from the command line. The third line gets the type catalog object reference. The fifth line gets the model type attribute properties object. The remaining lines access all the fields that comprise the attribute properties.

Retrieve Attribute Properties Example

The following example shows how to retrieve attribute properties:

```
import java.io.*;
import com.aprisma.spectrum.core.idl.*;
import com.aprisma.spectrum.core.util.*;
import com.aprisma.spectrum.core.idl.CsCException.*;
import com.aprisma.spectrum.core.idl.CsCAttribute.*;
import com.aprisma.util.corba.*;
public class GetAttributeInfo
{
    public static void main( String[] args )
    {
        CsCorbaValueHelper help = new CsCorbaValueHelper();
        Integer attrid = Integer.decode( args[0] );
        Integer mtid = Integer.decode( args[1] );
        CsCModelDomain md = null;
        try
        {
            String domainName = new String( "mySpectroServer" );

            CORBAHelper helper = CORBAHelper.getHelperImpl();
            helper.init( null, null );

            md = (CsCModelDomain) helper.getObjectImplementation(
                                CsCModelDomain.class, domainName );

            CsCTypeCatalog tc = md.getTypeCatalog();
            CsCModelType mt = tc.getModelType( mtid.intValue() );
            CsCMTypeAttrProperties mtap = mt.getMTypeAttrProperties(
                  attrid.intValue() );
            CsCMTypeAttrFlags mtaf = mtap.flags;
            System.out.println( "Memory: " + mtaf.memory );
            System.out.println( "Logged: " + mtaf.logged );

            CsCAttrProperties ap = mtap.commonProperties;
            System.out.println( "Name: " + ap.name );
            System.out.println( "Group ID: " + ap.attrGroupID );
```

```
            System.out.println( "Type: " + ap.type.toString() );
            System.out.println( "OID: " +
                  CsCorbaObjectIDHelper.toString( mtap.oidPrefix ) );
            CsCAttrFlags af = ap.flags;
            System.out.println( "Global: " + af.global );
            System.out.println( "Shared: " + af.shared );
            System.out.println( "External: " + af.external );
            System.out.println( "Readable: " + af.readable );
            System.out.println( "Writable: " + af.writable );
        }
        catch (Throwable e)
        {
            System.out.println( e );
        }
    }
}
```

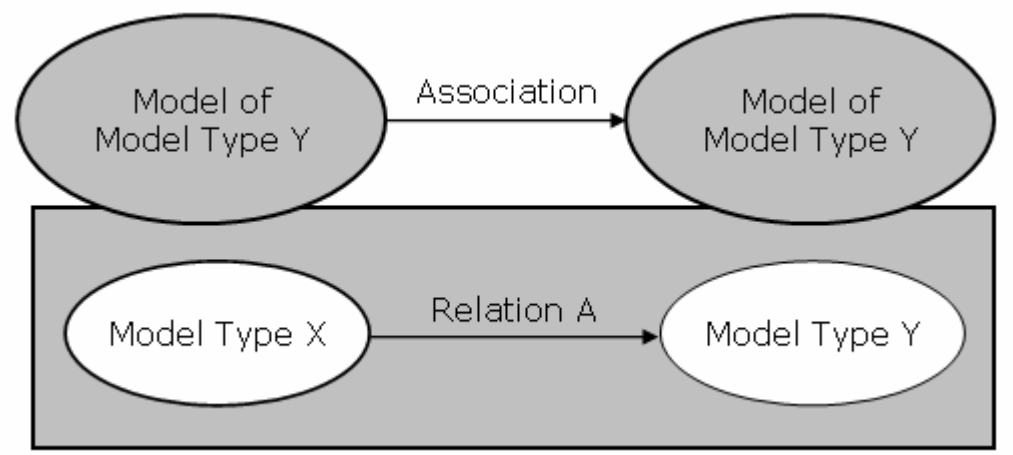
This example requires the following objects and methods:

- Package com.aprisma.spectrum.core.idl
  - Interfaces
    - CsCModelDomain
    - CsCTypeCatalog
    - CsCModelType
  - Classes
    - CsCModelDomainHelper
- Package com.aprisma.spectrum.core.idl.CsCAttribute
  - Classes
    - CsCMTypeAttrProperties
    - CsCMTypeAttrFlags
    - CsCAttrProperties
    - CsCAttrFlags
    - CsCValueType_e
- Package com.aprisma.spectrum.core.util
  - Classes
    - CsCorbaObjectIDHelper

Model Creation

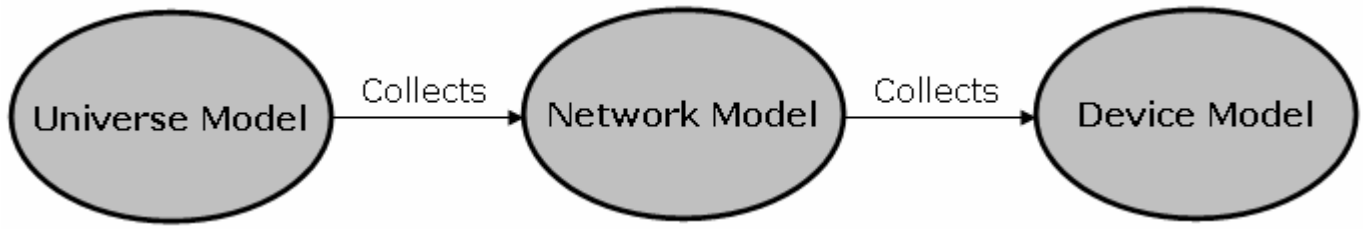
Some methods perform a single task while other methods perform multiple tasks. For example, createModelByID only creates a model. It does not name the model or set attribute values or associate it with another model. However, createModelByIP discovers a network device at the specified IP address and creates a set of models to represent the device. Both creation and association methods require model type IDs and relation IDs as building blocks for the model domain.

The following illustration shows models and associations as instantiations of model types and relations.



To create a container model and collect the model into the Universe view, use the following CsCModelDomain methods: createModelByID and createAssociationByIDs. The first method creates a model of the specified model type and initializes one of the new model's attribute values. Because you want to create a container model, you need to specify a network model type (0x1002e). The second method creates an association between two models. For our purpose, we will specify Collects as the relation (0x10002), the Universe model as the left-hand model and the container as the right-hand model.

The following illustration shows the Collects relation:



NOTE
The new container does not appear in the OneClick topology until the client program invokes the second method.

The following example shows how to invoke CsCModelDomain's Create Model method:

```
Integer mtypeID = Integer.decode(args[0]);
CsCModelProperties mp = md.createModelByID(
mtypeID.intValue());
```

The first and second lines get the model type ID as a primitive data type int. The second line creates the model, specifying the model type ID and returning a CsCModelProperties data class. The createModelByID method does not return the CORBA object, CsCModelType, with the CsCModelProperties. These fields contain null references.

Model Association

The following example shows how to invoke CsCModelDomain's associate model method:

```
int universeID = ...
int myContainerID = ...
int relationID = 0x10002;
md.createAssociationByIDs(relationID,universeID,myContainerID);
```

Lines 1-3 specify the model IDs and relation ID as primitive data types. Line 4 invokes the createAssociationByIDs. This method creates an association between two models in the model domain using a relation. It is important to have container and device models associated in a *collects* association, or else, the model belongs to LostAndFound.

Create and Associate Models Example

The following example shows how to create and associate models:

```
import java.io.*;
import com.aprisma.spectrum.core.idl.*;
import com.aprisma.spectrum.core.util.*;
import com.aprisma.spectrum.core.idl.CsCModelPackage.*;
import com.aprisma.spectrum.core.idl.CsCAttribute.*;
import com.aprisma.spectrum.core.idl.CsCError.*;
import com.aprisma.util.corba.* ;
public class CreateModelAndAssocByIP
{
    public static void main(String[] args)
    {
        CsCorbaValueHelper valhelper = new CsCorbaValueHelper();
        byte[] ipAddress = null;
        try
        {
            CsCValue ipValue = valhelper.parseInternetAddress( argv[0] )
            ipAddress = ipValue.internetAddress();
        }
        catch( java.lang.NumberFormatException e)
        { System.out.println(e); }
        Integer relID = Integer.decode(args[1]);
        Integer modeltypeID = Integer.decode(args[2]);
        Integer universeID = Integer.decode(args[3]);
        try
        {
            String domainName = new String("mySpectroSERVER");
            // Bind
            CORBAHelper helper = CORBAHelper.getHelperImpl () ;
            helper.init(null, null);
            CsCModelDomain md =
                   (CsCModelDomain) helper.getObjectImplementation (
                                CsCModelDomain.class,
                                domainName ) ;
           // Create the Network Models
           CsCModelProperties mpNetwork =
                md.createModelByID(modeltypeID.intValue());
           System.out.println( mpNetwork.modelID );
           // Create the association: Universe "Collects" Network
           md.createAssociationByIDs( relID.intValue(),
                                      universeID.intValue(),
                                      mpNetwork.modelDomainID +
                                      mpNetwork.modelID);
           // Calling this method will return a CsCModelProperties with
           // only the handle value set in the data object
           CsCValue commStrVal = new CsCValue() ;
```

```
            commStrVal.textString( EncodeUtils.convertToBytes( "public" )) ;
                      CsCModelProperties mpDevice = md.createModelByIP( ipAddress,
                                                                        commStrVal,
                                                                        300, 2, 161,
                                                                        avl);
           // Create the association: Network "Collects" Device
            md.createAssociationByIDs( relID.intValue(),
                   mpNetwork.modelDomainID + mpNetwork.modelID,
                   mpNetwork.modelDomainID + mpDevice.modelID );
        }
        catch(Throwable e)
        {
            System.out.println(e);
        }
    }
}
```

Retrieve a Model Through Association Example

You need to retrieve models through associations for the following reasons:

- To know the port models that exist for a device
- To know the models that exist in a particular container model
- To know how devices are connected to each other.

You can get this information by calling getAssocModelIDList or getAssocModelIDListOfMType. You can make both calls against the model domain.

Example: Retrieve all models collected by the LAN container model

The following example shows how to retrieve all models collected by the LAN container model:

```
int myLANContainer = ...
int relationID = 0x10002; //Collects relation
int[] list = md.getAssocModelIDList(relationID,
myLANContainer,
CsCSide_e.CSC_LEFT_SIDE);
```

NOTE
If you want to narrow down the result set to a particular model type, use getAssocModelIDListOfMType.

The parameters are as follows:

- **myLANContainer**
  Defines the model handle of the container.
- **relation**
  Collects the relation ID.
- **list**
  Defines the int [] of model IDs.
- **CsCSide_e.CSC_LEFT_SIDE**
  Defines the side on which the myLANContainer model handle resides.

This example requires the following objects and methods:

- Package com.aprisma.spectrum.core.idl
  – Interfaces

    - CsCModelDomain
  - Classes
    - CsCModelDomainHelper
- Package com.aprisma.spectrum.core.idl.CsCRelationPackage
  - Classes
    - CsCSide_e

The following example shows how to retrieve a model through associations:

```
import com.aprisma.spectrum.core.idl.*;
import com.aprisma.spectrum.core.util.*;
import com.aprisma.spectrum.core.idl.CsCRelationPackage.*;
import com.aprisma.spectrum.core.idl.CsCError.*;
import com.aprisma.util.corba.* ;
public class GetModelAssocInfo
{
    public static void main(String[] args)
    {
        Integer relID = Integer.decode(args[0]);
        Integer modelID = Integer.decode(args[1]);
        try
        {
            String domainName = new String("mySpectroSERVER");
            // Bind
            CORBAHelper helper = CORBAHelper.getHelperImpl () ;
            helper.init(null, null);
            CsCModelDomain md =
                   (CsCModelDomain) helper.getObjectImplementation (
                                  CsCModelDomain.class,
                                  domainName ) ;
            int[] modelIDList = md.getAssocModelIDList(
                   relID.intValue(), modelID.intValue(),
                   CsCSide_e.CSC_LEFT_SIDE);
        }
        catch(Throwable e)
        {
            System.out.println(e);
        }
    }
}
```

Rules and Relations

You can define rules and relations for DX NetOps Spectrum models.

A *relation* describes the interaction between two models and is expressed as a verb or verb phrase, for example, Contains, Manages, Is_Adjacent_to, and so on.

A *rule* is a statement that applies a relation to two model types. The rule lets models of the two types to be associated through the relation. For example, if there is a relation named Contains, a rule can be Model Type A Contains Model Type B. The left side of the rule statement is referred to as the subject of the rule (Model Type A) and the right side as the object (Model Type B). The role of a model type depends on the side it appears on.

Many relations in the DX NetOps Spectrum knowledge base have one or more metarules that specify which model types can participate in that relation. Model types derived from another model type that is specified in a metarule inherit

the eligibility to participate in that relation. For example, one of the metarules for the Contains relation is LAN Contains Device. If a model type named myLAN is derived from the model type LAN, SpectroSERVER automatically follows the rule myLAN Contains Device.

DX NetOps Spectrum supports the following relation types:

- **One-to-Many:** For example, One LAN Collects Many Devices. Only one LAN can be the parent owner of a device in the topology.
- **Many-to-Many:** For example, Many Ports Connects_To Many Devices.

The cardinality may differ from relation to relation.

NOTE
For a detailed discussion on relations, rules, and metarules, see the Getting Started section.

Relation Handles

Every relation in the DX NetOps Spectrum database has a handle. The following table shows examples of relations and handles:

Relation	Handle
Collects	0x10002
HASPARTS	0x10004
Manages	0x1001f

Retrieve Relation Information Example

The following example shows how to retrieve relation information:

```
import java.io.*;
import com.aprisma.spectrum.core.idl.* ;
import com.aprisma.spectrum.core.util.* ;
import com.aprisma.spectrum.core.idl.CsCException.* ;
import com.aprisma.spectrum.core.idl.CsCError.* ;
import com.aprisma.spectrum.core.idl.CsCRelationPackage.* ;
import com.aprisma.util.corba.* ;
public class GetRelationInfo
{
    public static void main(String[] args)
    {
        CsCModelDomain md = null;
        CsCTypeCatalog tc = null;
        CsCRelProperties relprop = null;
        try
        {
            String domainName = new String("mySpectroServer");
            // Construct helper
            CORBAHelper helper = CORBAHelper.getHelperImpl () ;
            helper.init(null,null);

            md = (CsCModelDomain) helper.getObjectImplementation (
                        CsCModelDomain.class, domainName);
            tc = md.getTypeCatalog();
```

```
            CsCRelPropList rpl = tc.getAllRelPropList();
            for( int i=0; i < rpl.list.length; i ++)
            {
                relprop = rpl.list[i];
                if(relprop.error == CsCError_e.SUCCESS){
                    System.out.println ("Name        : " +
                                        relprop.name);
                    System.out.println ("Cardinality : " +
                            relprop.cardinality.toString());
                }
            }
        }
        catch (Throwable e) {
            System.out.println(e);
        }
    }
}
```

The previous example requires the following objects and methods:

- Package com.aprisma.spectrum.core.idl
  - Interfaces
    - CsCModelDomain
    - CsCTypeCatalog
    - CsCRelation
  - Classes
    - CsCModelDomainHelper
- Package com.aprisma.spectrum.core.idl.CsCRelationPackage
  - Classes
    - CsCRelProperties

Significance of Rules and Metarules

Rules and metarules prevent illogical associations that may cause SpectroSERVER to malfunction. A rule specifies two model types that can participate in a particular relation. The rule also specifies the relative role of each model type by its position (that is, left or right) in the statement. Usually, the model type on the left is the parent in parent-child relationships. A metarule defines that derived model types also take part in that rule.

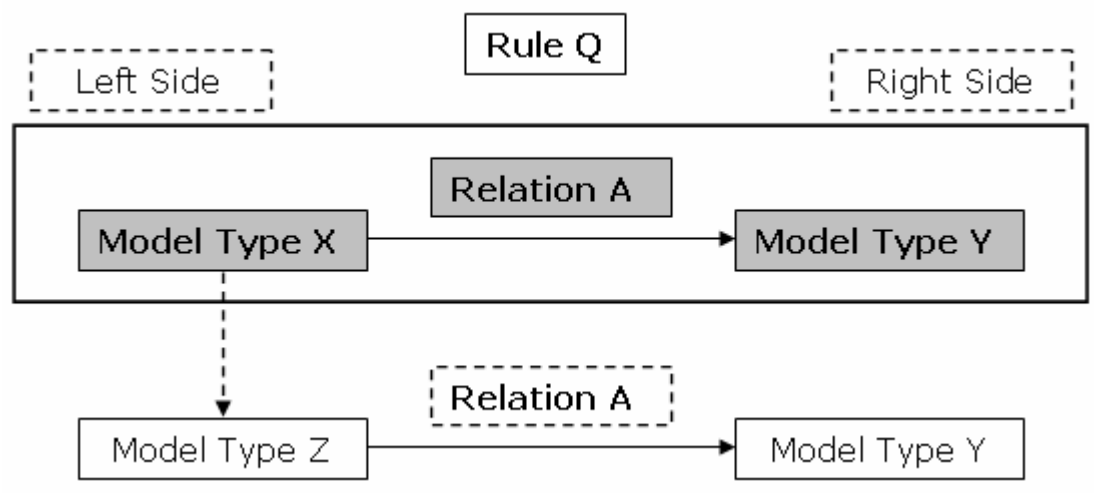
The following table shows the left and right sides of a relation:

Left Side	Relation	Right Side
Universe	Collects	LAN
Port	Connects to	Device
GnSNMPDev	Manages	Manages Apps

For programs that create model associations, check the existing rules to determine the model types that can be associated through a relation.

The getRelMTypePropList method call in the topic Method to Retrieve Rule Information passes the model type and the side (that is, left or right). The *side* parameter is with respect to the model type.

The following illustration shows a rule and its left and right sides:



If the developer is looking for all models of type Y that are on the right side in the rule, model type X is passed. Model type X is on the left.

Retrieve Rule Information Example

The following example shows how to retrieve rule information:

```
import java.io.*;
import com.aprisma.spectrum.core.idl.*;
import com.aprisma.spectrum.core.util.*;
import com.aprisma.spectrum.core.idl.CsCException.*;
import com.aprisma.spectrum.core.idl.CsCModelType.*;
import com.aprisma.spectrum.core.idl.CsCModelTypePackage.*;
import com.aprisma.spectrum.core.idl.CsCRelationPackage.*;
import com.aprisma.util.corba.*;
public class GetRuleInfo
{
    public static void main( String[] args )
    {
        Integer relid = Integer.decode( args[0] );
        Integer mtid = Integer.decode( args[1] );
        CsCModelDomain md = null;
        try
        {
            String domainName = new String("mySpectroServer");
            CORBAHelper helper = CORBAHelper.getHelperImpl();
            helper.init( null, null );
            md = (CsCModelDomain) helper.getObjectImplementation(
                CsCModelDomain.class, domainName );
            CsCTypeCatalog tc = md.getTypeCatalog();
            CsCModelType mt = tc.getModelType( mtid.intValue() );
            CsCRelation rel = tc.getRelation( relid.intValue() );
            CsCMTypePropList mtpl = rel.getRelMTypePropList(
                mt, CsCSide_e.CSC_LEFT_SIDE );
            System.out.println( "Read: " + mtpl.error.toString() );

            for ( int i=0; i<mtpl.list.length; i++ ) {
```

```
                CsCMTypeProperties mtp = mtpl.list[i];
                System.out.println( "ModelTypeName "
                    + new Integer(i).toString() + ": " + mtp.name );
            }

        } catch ( Throwable e ) {
            System.out.println( e );
        }
    }
}
```

This example requires the following objects and methods:

- Package com.aprisma.spectrum.core.idl
  - Interfaces
    - CsCModelDomain
    - CsCTypeCatalog
    - CsCRelation
    - CsCModelType
  - Classes
    - CsCModelDomainHelper
- Package com.aprisma.spectrum.core.idl.CsCModelTypePackage
  - Classes
    - CsCMTypePropList
    - CsCMTypeProperties
- Package com.aprisma.spectrum.core.idl.CsCRelationPackage
  - Classes
    - CsCSide_e

Filters

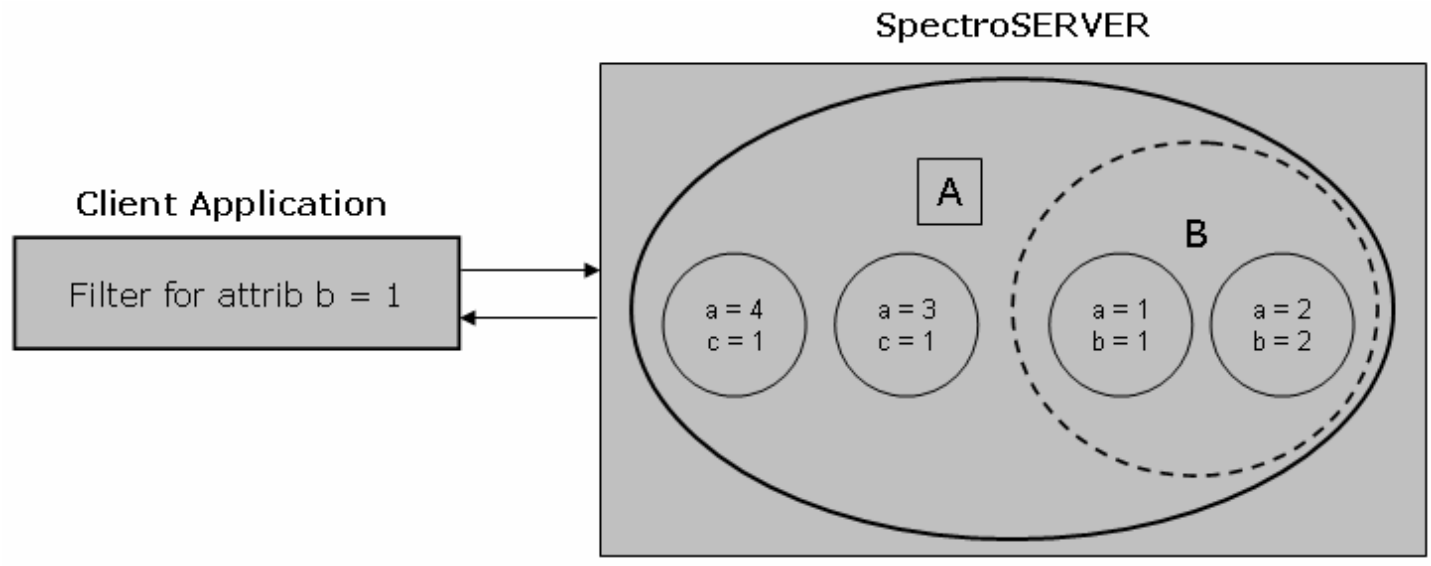
The CsCModelDomain interface provides several techniques to specify the information to be returned in a list. Server-side filtering gives the following benefits:

- Reducing network traffic.
- Removing extraneous data and providing only the necessary information to the client application.

You should ask DX NetOps Spectrum to filter information before sending it to your client application. Using the right attribute as the basis of your filter improves the search performance.

The following illustration shows two sets, A and B. A represents all models, and B represents the subset of models with attribute b. Assume that most customers use attribute a as a filter. The search will work but because attribute A exists for all models; it compares all the models of the database. If attribute B was used, only two models are compared. That is, the comparison time is halved because only half the models are compared.

The following illustration shows the client communicating with the SpectroSERVER using filters:



A DX NetOps Spectrum attribute filter (CsCValue[]) lets you define database search instructions. For example, to obtain a list of all models whose model types derive from the model type Device, you can use a filter as follows:

```
CsCorbaFilterAttrNode node =
CsCorbaAttrFilterHelper.createModelTypeIDNode(
0x10001, // Attribute ID of Model Type Handle
CsCOperator_e.CSC_IS_DERIVED_FROM,
0x1004b ); // Model Type Handle for Device
CsCValue[] filter = node.getFilter();
```

The class CsCorbaAttrFilterHelper is central to this example. You can use methods of this class to build different types of filters, including the createModelTypeIDNode method used in the example. This method takes the following parameters: attribute ID, operator, and model type ID. Once you create the node (lines 1-5), you call the node's getFilter method to create the filter (line 6), which is an array of the following CsCValue objects:

- CSC_OPERATOR = IS_DERIVED_FROM
- CSC_ATTR_VALUE = 0x10001
- CSC_MODEL_TYPE_ID = 0x1004b

An attribute filter contains an operator followed by an attribute ID and a value.

Interfaces that accept attribute filters as parameters require an array of CsCValue objects. Constructing this array is complex because of the order. CsCorbaAttrFilterHelper provides an alternate way of creating attribute filters for CORBA interfaces. This class provides an easy way of creating elementary pieces of the filter and assembling them together. After assembling the filter, you can use the getFilter method to get the final array.

For example, if you want a simple filter for all models with the Model_Name attribute MyName, you can write a filter as follows:

```
CsCorbaFilterNode nameNode =
CsCorbaAttrFilterHelper.createTextStringNode (
0x1006e,
CsCOperator_e.CSC_EQUALS,
"MyName" ) ;
CsCValue [] filter = nameNode.getFilter () ;
```

Here, the method createTextStringNode is called because the attribute of type TextString is compared. The attribute type dictates the call made to the CsCorbaAttrFilterHelper. CsCOperator houses all the logical and text operators that are

used. Because you can use other operators such as CSC_EQUALS_IGNORE_CASE and CSC_HAS_SUBSTRING for Text String attributes, you should consider the available operators before comparing the attribute types.

Complex Attribute Filter

To get all models with model types derived from the Device model type, except models of type Indirect RMON, you can write a filter as follows:

```
CsCorbaFilterAttrNode node1 =
CsCorbaAttrFilterHelper.createModelTypeIDNode(
0x10001, // Attribute ID of Model Type Handle
CsCOperator_e.CSC_IS_DERIVED_FROM,
0x1004b ); // Model Type Handle for Device
CsCorbaFilterAttrNode node2 =
CsCorbaAttrFilterHelper.createModelTypeIDNode(
0x10001, // Attribute ID of Model Type Handle
CsCOperator_e.CSC_DOES_NOT_EQUAL,
0x59001b); // model type ID IndirectRMON
CsCorbaFilterBinaryNode bnode =
CsCorbaAttrFilterHelper.createBinaryNode(
node1,CsCOperator_e.CSC_AND,node2);
CsCValue[] filter = bnode.getFilter();
```

This example creates three nodes as follows:

- Node1 represents all models with model types derived from Device.
- Node2 specifies that model type should not be Indirect RMON.
- The third node, bnode (binary node) is the logical AND of node1 and node2.

The filter array in the example contains the following information:

```
CSC_OPERATOR = AND
CSC_OPERATOR = IS_DERIVED_FROM
CSC_ATTR_VALUE = 0x10001
CSC_MODEL_TYPE_ID = 0x1004b
CSC_OPERATOR = DOES_NOT_EQUAL
CSC_ATTR_VALUE = 0x10001
CSC_MODEL_TYPE_ID = 0x59001b
```

The logical AND appears in the first CsCValue object followed by two triplets representing node1 and node2, respectively.

If you want to augment this filter with another restriction namely, all models of the list must have a Model Class attribute value of 2, you can write this filter as follows:

```
CsCorbaFilterAttrNode node1 =
CsCorbaAttrFilterHelper.createModelTypeIDNode(
SSORBHelper.attrIDMType,
CsCOperator_e.CSC_IS_DERIVED_FROM,
SSORBHelper.mtypeIDDevice);
CsCorbaFilterAttrNode node2 =
CsCorbaAttrFilterHelper.createModelTypeIDNode(
0x10001, // Attribute ID of Model Type Handle
CsCOperator_e.CSC_DOES_NOT_EQUAL,
0x59001b); // model type ID IndirectRMON
CsCorbaFilterBinaryNode bnode =
CsCorbaAttrFilterHelper.createBinaryNode(
```

```
node1,CsCOperator_e.CSC_AND,node2);
CsCValue modelClass = new CsCValue();
modelClass.intValue(2);
node2 = CsCorbaAttrFilterHelper.createAttrValNode(
0x11ee8,
CsCOperator_e.CSC_EQUALS, 2);
bnode = CsCorbaAttrFilterHelper.createBinaryNode(
bnode,CsCOperator_e.CSC_AND,node2);
CsCValue[] filter = bnode.getFilter();
```

NOTE
The attribute specifies whether the model is a bridge, switch, router, brouter, and so on, and a value of 2 specifies switch.

Lines 1-15 are identical to the previous example. The binary node bnode contains information from node1 and node2 and the logical AND operator. Lines 17-23 create a new node specifying Model Class equals 2, reusing the available node2. Lines 25-26 combine bnode and node2 and store the result in bnode again, overwriting the previous information. Line 28 extracts the CsCValue[] array from bnode. The content of the array is as follows:

```
CSC_OPERATOR AND
CSC_OPERATOR AND
CSC_OPERATOR IS_DERIVED_FROM
CSC_ATTR_VALUE 0x10001
CSC_MODEL_TYPE_ID 0x1004b
CSC_OPERATOR DOES_NOT_EQUAL
CSC_ATTR_VALUE 0x10001
CSC_MODEL_TYPE_ID 0x59001b
CSC_OPERATOR EQUALS
CSC_ATTR_VALUE 0x11ee8
CSC_INTEGER 2
```

Lines 3-5 represent one node, and lines 6-8 represent the second. Line 2 represents the logical AND between the two nodes. That is, line 2 makes lines 3-8 into a single node. Lines 9-12 represent the third node. Line 1 represents the logical AND between this node and the single node of lines 3-8.

The previous example requires the following objects and methods:

- Package com.aprisma.spectrum.core.idl
  - Interfaces
    - CsCModelDomain
    - CsCModelPropList
    - CsCModeProperties
  - Classes
    - CsCModelDomainHelper
- Package com.aprisma.spectrum.core.util
  - Interfaces
    - CsCorbaFilterAttrNode
    - CsCorbaFilterBinaryNode
  - Classes
    - CsCorbaAttrFilterHelper

Search Criteria XML

Use getModelIDListByXmlSearchCriteria() to invoke a search in DX NetOps Spectrum. Any valid search criteria xml in DX NetOps Spectrum can be passed to getModelIDListByXmlSearchCriteria().

NOTE
getModelIDListByXmlSearchCriteria() provides more capabilities than the legacy getModelIDListByAttrFilter(), which only supports a subset of the possible searches.

Use the following procedure to create search criteria xml that can be used with getModelIDListByXmlSearchCriteria().

Follow these steps:

1. Use the OneClick Locater tab to create and save the new search. Custom searches are saved as xml files in the <*SPECROOT$*>/custom/console/config directory.

   NOTE
   For more information about creating searches using the Locater tab, see the Administration section.
2. Use the content of the new search as the xml input to getModelIDListByXmlSearchCriteria(). Make the following minor code modifications:
   - Turn the search criteria into a string or char * data type that is appropriate to the language that you are coding in.
   - Escape any quotes in the code with a backslash (\).

Search Criteria XML

Use getModelIDListByXmlSearchCriteria() to invoke a search in DX NetOps Spectrum. Any valid search criteria xml in DX NetOps Spectrum can be passed to getModelIDListByXmlSearchCriteria().

NOTE
getModelIDListByXmlSearchCriteria() provides more capabilities than the legacy getModelIDListByAttrFilter(), which only supports a subset of the possible searches.

Use the following procedure to create search criteria xml that can be used with getModelIDListByXmlSearchCriteria().

Follow these steps:

1. Use the OneClick Locater tab to create and save the new search. Custom searches are saved as xml files in the <*SPECROOT$*>/custom/console/config directory.

   NOTE
   For more information about creating searches using the Locater tab, see the Administration section.
2. Use the content of the new search as the xml input to getModelIDListByXmlSearchCriteria(). Make the following minor code modifications:
   - Turn the search criteria into a string or char * data type that is appropriate to the language that you are coding in.
   - Escape any quotes in the code with a backslash (\).

Retrieve Models Using Search Criteria Example

The following is a simple search that uses getModelIDListByXmlSearchCriteria:

```
// %SOURCEHEADER%

import java.net.*;

import com.aprisma.spectrum.core.idl.*;
import com.aprisma.spectrum.core.util.*;
```

```
import com.aprisma.spectrum.core.idl.CsCRelationPackage.*;
import com.aprisma.spectrum.core.idl.CsCAttribute.*;
import com.aprisma.spectrum.core.idl.CsCError.*;
import com.aprisma.util.corba.* ;

public class GetDevices
{
    public static void main(String[] args)
    {
        String usage = "Usage:\n    GetDevices modelClassId [domain name]";
        Integer modelclass;
        String domainName = null;
        try
        {
            if ( args.length < 1 ||
                 args[0].equals( "?" ) ||
                 args[0].equalsIgnoreCase( "-help" ) )
            {
                System.out.println( usage );
                return;
            }
            else
            {
                modelclass = Integer.decode( args[0] );
                if ( args.length < 2 )
                {
                    InetAddress address = InetAddress.getLocalHost();
                    String domainNameStr = address.getHostName();
                    String[] names = domainNameStr.split( "\\." );
                    domainName = names[0].toLowerCase();
                }
                else
                {
                    domainName = args[1];
                }
            }

            // Bind
            CORBAHelper helper = CORBAHelper.getHelperImpl () ;
            helper.init(null, null);
            CsCModelDomain md =
                   (CsCModelDomain) helper.getObjectImplementation (
                                   CsCModelDomain.class,
                                   domainName ) ;

            int domainID = md.getModelDomainID();

            String xmlSearchCriteria = "<search-criteria>\n" +
                                       "  <filtered-models>\n" +
                                       "    <equals>\n" +
                                       "      <attribute id=\"0x11ee8\">\n" +
                                       "        <value>" +
                                       modelclass.intValue() + "</value>\n" +
```

```
                                    "      </attribute>\n" +
                                    "    </equals>\n" +
                                    "  </filtered-models>\n" +
                                    "</search-criteria>\n" ;
            CsCValue val = new CsCValue() ;
            val.textString( EncodeUtils.convertToBytes( xmlSearchCriteria )) ;
            int[] modelIDs =
                   md.getModelIDListByXmlSearchCriteria ( val ) ;
            int[] modelIDs =
                    md.getModelIDListByXmlSearchCriteria( xmlSearchCriteria ) ;
            for( int i = 0; i<modelIDs.length; i++ )
            {
                System.out.println( "Model ID: " + modelIDs[i] +
                                   ", mh = 0x" + Integer.toHexString(
domainID + modelIDs[i]) );
            }
        }
        catch(Throwable e)
        {
            System.out.println(e);
        }
    }
}
```

Attribute Values

DX NetOps Spectrum models are instances of model types. Models and objects contain values of attributes defined in model types.

To read or write a model attribute value, you need to specify a model ID and an attribute ID. After getting a model ID, you need to determine the attributes available for that model.

You can get a model ID as explained in the topics about creating and associating models, retrieving models though associations, and retrieving models though filter searches. You can get the available attributes from one of the following: OneClick Attribute Editor, Model Type Editor, Command Line Interface, the API method to retrieve attribute information. After determining the attributes that can be read, you should know the flags set on them.

Read an Internal Attribute Value Example

The following example shows how to read and display an attribute value for a given model ID:

```
Integer attrID = Integer.decode(args[0]);
int[] attrIDList = new int[1];
attrIDList[0] = attrID.intValue();

int[]modelIDList = md.getModelIDListByXmlSearchCriteria( xmlSearchCriteria ) ;
CsCAttrReadMode_e[] readMode =
   {CsCAttrReadMode_e.CSC_MOST_AVAILABLE};
CsCAttrValListOfModels avlom = md.readAttrValListOfModelsByIDs(
                                    modelIDList,
                                    attrIDList,
                                    readMode );
if( avlom.error == CsCError_e.SUCCESS )
 {
```

```
    for( int i = 0; i < avlom.list.length; i ++ )
    {
       CsCModelAttrValList mavl = avlom.list[i];
       CsCAttrValList avl = mavl.attrValList;
       if( avl.error == CsCError_e.SUCCESS )
       {
         for( int j = 0; j < avl.length; j ++ )
         {
           CsCAttrValue aval = avl.list[j];
           CsCError_e err = aval.error;
           if( err == CsCError_e.SUCCESS )
           {
             CsCValue val = aval.value;
             String name = val.textString();
             System.out.println("Model Name: " + name );
           }
        }
    }
}
```

Lines 1-3 set up an array of attribute IDs. Line 5 retrieves an int array of model IDs from a filter search as explained previously. Line 6 sets up an array of CsCAttrReadMode_e for each attribute requested. Lines 9-12 make a call to the model domain interfaces to read one attribute in MOST_AVAILABLE for an attribute for the list of model IDs found by the search. The method call readAttrValListOfModelsByIDs reads attributes on models and returns the value CsCAttrValListofModels.

The first check is for the overall error code. A check should always look for SUCCESS and not FAILURE because there are many variations of FAILURE. Line 15 constructs a FOR loop to iterate through the array of CsCModelAttrValList classes. Line 19 again checks for SUCCESS before iterating through the CsCAttrValList on line 21. The error code is checked for each CsCAttrValue, and finally the attribute is extracted as a string by calling the method textString on the CsCValue.

Error Codes and CsCValue

You should check for error codes when reading attributes internally or externally. Data classes that pass data contain information about more than one model. If 1 out of 100 models has an error, it permits 99 models to return successfully. Design the code so you can know if only some of the models are successful.

Many classes use CsCValue as a member variable and as an argument in several member methods. A CsCValue object is like an enum because it can hold a variety of objects including the following:

- actionID
- agentID
- alarmID
- attributeID
- attrGroupID
- attrValID
- boolValue
- counterValue
- dateTime
- actionID
- developerID
- enumValue
- eventID
- eventIDList
- gaugeValue
- hiddenValue
- intValue
- ipAddress
- model
- modelID
- modelDomain
- modelDomainID
- eventIDList
- modelType
- modelTypeID
- nullValue
- objectID
- octetString
- operatorValue
- realValue
- relation
- relationID
- taggedOctetString
- textString
- timeTicks

How the Void Constructor Works

To use the class CsCValue, you need to instantiate it through the void constructor.

You can use the void constructor as follows:

1. Create an object with no type or value using the void constructor as follows:
   ```
   CsCValue v = new CsCValue();
   ```
2. Set the object to a particular value type (for example, textString) and value using the appropriate set method as follows:
   ```
   v.textString("Container III");
   ```
3. Use the appropriate get method as follows to extract the value from the object:
   ```
   String name = v.textString();
   ```

Set and *get* are overloaded methods. If you want to determine the type of a particular CsCValue object, enter the following:

```
CsCValueType_e type = v.discriminator();
```

4. To display the type and value of a CsCValue object, use the CsCorbaValueHelper class as follows:

```
CsCorbaValueHelper vh = new CsCorbaValueHelper();
System.out.println(
v.discriminator().toString()
+ " "
+ vh.toStringValue(v));
```

The CsCorbaValueHelper class has a number of methods to deal with CsCValue objects. To learn more about this helper class, click the link com.aprisma.spectrum.core.util on the Java doc page <*$SPECROOT*>/SDK/docs/SSORB/index.html.

Read Modes

CsCAttrReadMode_e provides the following read modes:

- **CSC_MOST_AVAILABLE**
  Reads internally if the memory or db flags are set. If the flags are not set, go external if the attribute has the external flag set.
- **CSC_MOST_CURRENT**
  Reads externally if the external flag is set; otherwise reads internally.
- **CSC_TRY_MOST_CURRENT**
  Reads externally if the external flag is set, else reads internally. If the external read fails, go internal if memory or database flag is set.
- **CSC_SYNC_CURRENT**
  Not used because try latest facilitates the same functionality. If the latest value differs from the memory or database value, it updates the internal values to the latest.

If you are reading an internal attribute, use CSC_MOST_AVAILABLE, and if you are reading externally, read with CSC_MOST_CURRENT. You need to make a decision only if the attribute is external but has memory or database. The direction of read is dictated by whether you need the latest value or the current value is acceptable.

Rules for External Reading

Rules for external reading are as follows:

- Verify that the attribute Dev_Contact_Status is set to 1, that is, Established.
- Verify that the attribute isManaged is set to TRUE.

If these attributes do not have the correct values, your programs may hang because you try to read while the contact was lost. This situation causes the read operation to wait for timeouts and retries. Another reason for the wait can be that the device or port is under maintenance.

Read External Attributes Example

The following example shows how to read external attributes:

```
import com.aprisma.spectrum.core.idl.*;
import com.aprisma.spectrum.core.util.*;
import com.aprisma.spectrum.core.idl.CsCModelPackage.*;
import com.aprisma.spectrum.core.idl.CsCAttribute.*;
import com.aprisma.spectrum.core.idl.CsCError.*;
import com.aprisma.util.corba.* ;
public class ReadExternalAttribute
```

```
{
    static CsCModelDomain md = null;

    public static void main(String[] args)
    {
        Integer extAttrID = Integer.decode(args[1]);
        Integer modelID = Integer.decode(args[0]);
        int[] modelIDList = new int[1];
        int[] extAttrIDList = new int[1];
        modelIDList[0] = modelID.intValue();
        extAttrIDList[0] = extAttrID.intValue();
        try
        {
            String domainName = new String("mySpectroSERVER");
            // Bind
            CORBAHelper helper = CORBAHelper.getHelperImpl () ;
            helper.init(null, null);
            md = (CsCModelDomain) helper.getObjectImplementation (
                                  CsCModelDomain.class,
                                  domainName ) ;
            if( okToReadExternal( modelIDList ))
            {
                CsCAttrReadMode_e [] readMode =
                     { CsCAttrReadMode_e.CSC_MOST_CURRENT};
                CsCAttrValListOfModels avlom =
                     md.readAttrValListOfModelsByIDs(
                                        modelIDList,
                                        extAttrIDList,
                                        readMode );
                if( avlom.error == CsCError_e.SUCCESS )
                {
                    for( int i = 0; i < avlom.list.length; i ++ )
                    {
                        CsCModelAttrValList mavl = avlom.list[i];
                        CsCAttrValList avl = mavl.attrValList;
                        if( avl.error == CsCError_e.SUCCESS )
                        {
                            CsCAttrValue aval = avl.list[0];
                            CsCError_e err = aval.error;
                            if( err == CsCError_e.SUCCESS )
                            {
                                CsCValue val = aval.value;
                                System.out.println("Text String: " +
                                    val.textString() );
                            }
                        }
                    }
                }
                else
                {
                    System.out.println("Read was not overall
                       successful");
                }
```

```
            }
        }
        catch(Throwable e)
        {
            System.out.println(e);
        }
    }
    public static boolean okToReadExternal(int[] modelIDList)
    {
        boolean ok = false;
        // Dev_Contact_Status 0x110ed
        // isManaged 0x1295d
        int[] okToReadExternAttrs = { 0x110ed, 0x1295d };
        CsCAttrReadMode_e[] readMode = {
            CsCAttrReadMode_e.CSC_MOST_AVAILABLE,
            CsCAttrReadMode_e.CSC_MOST_AVAILABLE};
        try
        {
            CsCAttrValListOfModels avlom =
                 md.readAttrValListOfModelsByIDs(
                     modelIDList,
                     okToReadExternAttrs,
                     readMode);
            if (avlom.error == CsCError_e.SUCCESS)
            {
                if (avlom.list[0].attrValList.list[0].value.intValue()
                     == 1 &&
                    avlom.list[0].attrValList.list[1].value.boolValue()
                     == true)
                {
                    ok = true;
                }
            }
        }
        catch (Throwable e)
        {
            System.out.println(e);
        }
        return ok;
    }
}
```

This example requires the following objects and methods:

- Package com.aprisma.spectrum.core.idl
  - Interfaces
    - CsCModelDomain
  - Classes
    - CsCModelDomainHelper
- Package com.aprisma.spectrum.core.idl.CsCAttribute
  - Classes

- CsCAttrReadMode_e
- CsCValueType_e
- CsCValue

Reading Values from an Attribute Table

Reading values from an attribute table is similar to reading a value from a single attribute. The techniques of selecting the right read mode and how to read externally are still valid. But if the tables are large, these techniques can have effects other than increased processing time, such as device performance degradation and increased network traffic.

The API table-read ability is robust and lets you specify ranges (by defining high and low instances) and cap limits on the number of entries that can be retrieved. This improves performance if you are reading only a little data from a table, as shown in the following example:

```
int [] attr = {attrhandle.intValue()};
CsCOIDSpec low = new CsCOIDSpec();
CsCOIDSpec high = new CsCOIDSpec();
int [] lowValue = { 0 };
int [] highValue = { 100 };
low.objectID( lowValue );
high.objectID( highValue );
int length = 23;
CsCModel model = md.getModel( ());
CsCAttrValTable tablevalues = md.readAttrValTable(
modelhandle.intValue(),attr,low, high, length,
CsCAttrReadMode_e.CSC_MOST_CURRENT );
for( int j = 0; j < tablevalues.table.length; j++ )
{
System.out.println(
tablevalues.table[j][0].value.toString());
}
```

The first parameter is an array of attributes that is read. The array should specify attributes that belong to the table, because the instance array bounds may not be the same between different tables.

The second and third parameters specify the low and high ends of the instance range. The range is specified using CsCOIDSpec data parameters for the following reasons:

- Instances can comprise more than one value.
- Usually tables are instanced by a series of values.

The fourth parameter specifies the maximum number of table entries to collect. If the range exceeds the maximum limit, then only the maximum is specified. If the maximum is not met, it returns only the entries found in the table. The fifth parameter follows the single read convention.

In the following example, attribute IDs make up the columns and instances make up the rows. CsCValues make up the body of the array. Error data structures exist for the columns, rows, and values.

> **NOTE**
> Low, high, and limit parameters have wild card ability.

```
CsCModelDomain md = (CsCModelDomain)
helper.getObjectImplementation
 ( CsCModelDomain.class, domainName );
CsCOIDSpec low = new CsCOIDSpec();
CsCOIDSpec high = new CsCOIDSpec();
int length = 0;
```

```
low.noLimit(true);
high.noLimit(true);
int [] attr = {attrid.intValue()};
CsCAttrValTable tablevalues = md.readAttrValTable(
 modelid.intValue(),attr, low, high, length,
 CsCAttrReadMode_e.CSC_MOST_CURRENT);
```

The noLimit method is called on the CsCOIDSpec objects. That sets up the object for wild card capability. For the limit parameter, the wild card capability is set up by setting the limit to zero.

This example requires the following objects and methods:

- Package com.aprisma.spectrum.core.idl
  - Interfaces
    - CsCModelDomain
  - Classes
    - CsCModelDomainHelper
- Package com.aprisma.spectrum.core.idl.CsCAttribute
  - Classes
    - CsCAttrReadMode_e
    - CsCValueType_e
    - CsCValue
    - CsCOIDSpec
    - CsCAttrValTable
    - CsCValueError
    - CsCAttrErrorList
    - CsCError_e

Read Table Attributes Example

The following example shows how to read table attributes:

```
import java.io.*;
import com.aprisma.spectrum.core.idl.*;
import com.aprisma.spectrum.core.util.*;
import com.aprisma.spectrum.core.idl.CsCException.*;
import com.aprisma.spectrum.core.idl.CsCAttribute.*;
import com.aprisma.util.corba.*;
public class ReadTableAttr {
    public static void main( String[] args ) {

        CsCorbaValueHelper help = new CsCorbaValueHelper();
        Integer attrid = Integer.decode( args[0] );
        Integer mid = Integer.decode( args[1] );
        CsCModelDomain md = null;
        try
        {
            String domainName = new String("mySpectroSERVER");

            CORBAHelper helper = CORBAHelper.getHelperImpl();
            helper.init( null, null );
            md = (CsCModelDomain) helper.getObjectImplementation(
                 CsCModelDomain.class, domainName );
```

```
            int [] attr = { attrid.intValue() };
            int [] lowValue = { 0 };
            int [] highValue = { 100 };
            CsCOIDSpec low = new CsCOIDSpec();
            CsCOIDSpec high = new CsCOIDSpec();
            int length = 10;
            low.objectID( lowValue );
            high.objectID( highValue );
            CsCAttrValTable tableValues = md.readAttrValTable(
                mid.intValue(), attr, low, high, length,
                CsCAttrReadMode_e.CSC_MOST_CURRENT );
            CsCAttrErrorList ael = tableValues.columnIndexList;
            for ( int i=0; i < ael.list.length; i++ )
            {
                System.out.println( "AEL len: " + ael.list.length );
                CsCOIDError[] vea = tableValues.rowIndexList;
                System.out.println( "VEA len: " + vea.length );
                for ( int j=0; j < vea.length; j++ )
                {
                    CsCValue attrval = tableValues.table[j][i].value;
                    System.out.println( i + ":" + j + " " +
                       attrval.textString() );
                }
            }
        }
        catch (Throwable e)
        {
            System.out.println( e );
        }
    }
}
```

Writing an Attribute Value and Table Values

The principles of reading an attribute also apply to writing attributes, except for the read modes. If an attribute is both external and memory or database, you need to write to both. Verify the attribute Dev_Contact_Status and isManaged before writing externally. Verify if the SNMP community string of the model has write privileges or writing externally fails.

The following example shows how to write a model's attribute value:

```
CsCValue val = new CsCValue();
val.textString(args[2] );
CsCAttrValue attrVal = new CsCAttrValue( attrID.intValue(),
     val, CsCError_e.SUCCESS );
CsCAttrValue[] attrValArray = new CsCAttrValue[1];
attrValArray[0] = attrVal;
CsCAttrValList writeValList = new CsCAttrValList( attrValArray,
     CsCError_e.SUCCESS );
CsCAttrErrorListOfModels aelom = md.writeAttrValListOfModelsByIDs(
     modelIDList,
     writeValList );
```

To write an attribute, you need to create a CsCAttrValList containing the attributes and values. Line 1 begins the construction of CsCAttrValList by creating a CsCValue. Line 2 assigns a test string value to CsCValue. Lines 3-5 create a CsCAttrValue passing the attribute ID, the CsCValue, and the error code SUCCESS. Line 6 is an array of one for CsCAttrValue, and line 7 assigns the CsCAttrValue to the array. Lines 8 and 9 construct the CsCAttrValList with the array and an error code of SUCCESS.

You can also construct CsCAttrValList as follows:

Read the value returned in a CsCAttrValList.

1. Set a value to CsCValue and pass it to the method writeAttrValListOfModelsByIDs.

The previous example requires the following objects and methods:

- Package com.aprisma.spectrum.core.idl
  - Interfaces
    - CsCModelDomain
  - Classes
    - CsCModelDomainHelper
- Package com.aprisma.spectrum.core.idl.CsCModelPackage
  - Classes
    - CsCAttrErrorListOfModels
- Package com.aprisma.spectrum.core.idl.CsCAttribute
  - Classes
    - CsCAttrReadMode_e
    - CsCValueType_e
    - CsCValue
- Package com.aprisma.spectrum.core.util
  - Classes
    - CsCorbaValueHelper

Write Attributes Example

The following example shows how to write attributes:

```
import com.aprisma.spectrum.core.idl.*;
import com.aprisma.spectrum.core.util.*;
import com.aprisma.spectrum.core.idl.CsCModelPackage.*;
import com.aprisma.spectrum.core.idl.CsCAttribute.*;
import com.aprisma.spectrum.core.idl.CsCError.*;
import com.aprisma.util.corba.* ;
public class WriteModelInfo
{
    public static void main(String[] args)
    {
        Integer attrID = Integer.decode(args[1]);
        Integer modelID = Integer.decode(args[0]);
        int[] modelIDList = new int[1];
        int[] attrIDList = new int[1];
        modelIDList[0] = modelID.intValue();
        try
        {
            String domainName = new String("mySpectroSERVER");
            // Bind
```

```
            CORBAHelper helper = CORBAHelper.getHelperImpl () ;
            helper.init(null, null);
            CsCModelDomain md =
                   (CsCModelDomain) helper.getObjectImplementation (
                                  CsCModelDomain.class,
                                  domainName ) ;
            // Create CsCValue
            CsCValue val = new CsCValue();
            val.textString(args[2] );
            // Create Attr value
            CsCAttrValue attrVal = new CsCAttrValue( attrID.intValue(),
                                                     val,
                                                     CsCError_e.SUCCESS );
            // Construct a CsCAttrValue array
            CsCAttrValue[] attrValArray = new CsCAttrValue[1];
            attrValArray[0] = attrVal;
            CsCAttrValList writeValList =
                 new CsCAttrValList( attrValArray,
                                     CsCError_e.SUCCESS );
            CsCAttrErrorListOfModels aelom =
                md.writeAttrValListOfModelsByIDs( modelIDList,
                                                  writeValList );
            if( aelom.error == CsCError_e.SUCCESS )
            {
                System.out.println("Read was overall successful");
            }
            else
            {
                System.out.println("Read was not overall successful");
            }
        }
        catch(Throwable e)
        {
            System.out.println(e);
        }
    }
}
```

The previous example requires the following objects and methods:

- Package com.aprisma.spectrum.core.idl
  - Interfaces
    - CsCModelDomain
  - Classes
    - CsCModelDomainHelper
- Package com.aprisma.spectrum.core.idl.CsCAttribute
  - Classes

- CsCAttrReadMode_e
- CsCValueType_e
- CsCValue
- CsCOIDSpec
- CsCAttrValTable
- CsCValueError
- CsCAttrErrorList
- CsCError_e

Build a CsC Attribute Value Table Example

Building a CsCAttrValTable for writing table attributes is similar to building a single value.

Example: Build a CsC attribute value table

The following example shows how to build a CsC attribute value table:

```
int size = 5;
CsCValueError[][] table = new CsCValueError[size][1];
CsCOIDError[] rowindexlist = new CsCOIDError[size];
CsCValue value;
for(int i = 0; i < size; ++i )
{
 int[] oidsuffix = new int[1];
 oidsuffix[0] = i;
 rowindexlist[i] = new CsCOIDError( oidsuffix,CsCError_e.SUCCESS);
 value = new CsCValue();
 value.textString("Hello");
 table[i][0] = new CsCValueError( value, CsCError_e.SUCCESS );
}
CsCAttrError[] attrlist = new CsCAttrError[1];
attrlist[0] = new CsCAttrError( attrid.intValue(),
    CsCError_e.SUCCESS);
CsCAttrErrorList columnindexlist = new CsCAttrErrorList( attrlist,
    CsCError_e.SUCCESS);
CsCAttrValTable attrvaltable = new CsCAttrValTable( table,
    rowindexlist, columnindexlist, CsCError_e.SUCCESS);
```

Line 1 sets the table size. Only one table attribute is set. Line 2 constructs a two-dimensional array of CsCValueError that forms the body of the CsCAttrValTable. Line 3 constructs a CsCOIDError array of size 5 for the five entries that are updated in the table. Line 5 sets up a For loop to iterate through the creation of the individual CsCValues. Lines 7, 8, and 9 set the rowindexlist array with the OID suffix and error code SUCCESS. Lines 10, 11, and 12 construct a CsCValue, assign the text string value of Hello, and assign it to the table array as a CsCValueError.

Lines 14-18 set up the attribute ID to write to. First, a CsCAttrError array of size one is constructed. Next, the first element is assigned a CsCAttrError with attribute and error code SUCCESS. Lines 19 and 20 read the constructed values to construct the CsCAttrValTable.

Write Table Attributes Example

The following example shows how to write table attributes:

```
import java.io.*;
import com.aprisma.spectrum.core.idl.*;
import com.aprisma.spectrum.core.util.*;
```

```
import com.aprisma.spectrum.core.idl.CsCException.*;
import com.aprisma.spectrum.core.idl.CsCAttribute.*;
import com.aprisma.spectrum.core.idl.CsCError.*;
import com.aprisma.util.corba.*;
public class WriteTableAttr
{
    public static void main( String[] args )
    {
        Integer attrid = Integer.decode( args[0] );
        Integer mid = Integer.decode( args[1] );
        CsCModelDomain md = null;
        try
        {
            String domainName = new String( "mySpectroSERVER" );
            CORBAHelper helper = CORBAHelper.getHelperImpl();
            helper.init( null, null );

            md = (CsCModelDomain) helper.getObjectImplementation(
                CsCModelDomain.class, domainName );
            int size = 3;
            CsCValueError[][] table = new CsCValueError[size][1];
            CsCOIDError[] rowindexlist = new CsCOIDError[size];
            CsCValue value;
            for ( int i=0; i < size; i++ )
            {
                int [] oidsuffix = new int[1];
                value = new CsCValue();
                value.textString( "Hello" );
                table[i][0] = new CsCValueError( value,
                                    CsCError_e.SUCCESS );
                oidsuffix[0] = i + 1;
                rowindexlist[i] = new CsCOIDError( oidsuffix,
                                    CsCError_e.SUCCESS );
            }
            CsCAttrError[] attrlist = new CsCAttrError[1];
            attrlist[0] = new CsCAttrError( attrid.intValue(),
                                   CsCError_e.SUCCESS );
            CsCAttrErrorList columnindexlist = new CsCAttrErrorList(
                  attrlist, CsCError_e.SUCCESS );
            CsCAttrValTable attrvaltable = new CsCAttrValTable( table,
                  rowindexlist, columnindexlist, CsCError_e.SUCCESS );
            CsCAttrErrorTable aet = md.writeAttrValTable(
                  mid.intValue(), attrvaltable );
        } catch ( Throwable e  )
        {
            System.out.println( e );
        }
    }
}
```

If you want to clear all entries of an internal attribute list, you cannot write an empty CsCAttrValTable because the writeAttrValTable method adds supplemental entries. Existing entries are updated and new entries added.

One of the methods to clear a list is as follows:

```
int[] attr = { 0xffff0021 };
md.clearListAttrs(mid.intValue(),attr);
```

Event Management

In addition to DX NetOps Spectrum, network managers often rely on third-party products to manage special functions. For example, a third-party product monitors firewalls and another watches the network traffic. These products gather information, filter data, and send events to DX NetOps Spectrum, which correlates, stores, and displays the events, and promotes them to alarms.

The following example shows how to use the API interface to create and manage events:

```
Integer modelID = Integer.decode( args[0] );
Integer eventCode = Integer.decode( args[1] );
int severity = 3;
CsCValue svalue = new CsCValue();
svalue.gaugeValue(severity);
CsCAttrValue sav = new CsCAttrValue(
    CsCorbaEventHelper.SEVERITY,
    svalue,
    CsCError_e.SUCCESS);
CsCAttrValue[] ava = new CsCAttrValue[]{sav};
CsCAttrValList avl = new CsCAttrValList( ava,
    CsCError_e.SUCCESS);
byte[] eventid = md.createEvent( modelID.intValue(),
    eventCode.intValue(), avl );
System.out.println("Event ID:" +
    CsCorbaEventHelper.toString( eventide ));
```

In the preceding example, an event is generated for a model through the CsCModelDomain interface. Lines 1 and 2 decode the model ID and event code. Lines 4-12 create a CsCValue of value 3 to assign as the severity of the event being created. On lines 13-14, the CsCorbaEventHelper is called to print the event ID to text string format. The toString method on the next line displays the event ID, which uniquely identifies the event among all the event domains in the installation. An example of an event ID is 3a63543c-0026-1000-00d9-0080108d4051.

This class contains all attributes from which you can retrieve the following values:

- Creation_Date
- Creator
- EventID
- Event_Type
- Model_Handle
- Model_Name
- Model_Type
- Model_Type_Name
- Severity and Type_Attributes

The class also contains helper methods for converting data such as the event ID into text format. CsCEventDomain provides access to events of a model domain. An event is a record of a situation in a monitored entity such as a router or an application. You can create a history of the entire network by storing a series of events from multiple entities. You can use the history to track down a problem in a monitored entity or a problem caused by the interaction of multiple monitored entities.

A client or DX NetOps Spectrum creates events but DX NetOps Spectrum receives them. Most events are distributed to clients and stored in a database. Events that DX NetOps Spectrum uses internally are not available to clients. CsCEventDomain provides a client interface to retrieve events from the database and receive events.

Every event contains a type and an identifier. Each new situation that occurs in a network creates a new event type. Two examples are *link up* and *link down* event types. An event identifier (CsCEventID) uniquely identifies events independent of the model domain in which it is created. By default, only this basic information is passed to a client.

Each event contains additional information stored in an attribute value error list. An attribute ID identifies the additional information which is sent to a client upon request.

The following section lists the event attributes and their descriptions:

- **Event ID**
  Uniquely identifies an event independent of the model domain in which the event is created.
  Type: CsCEventID
  Attribute ID: 0x11fbc
- **Event Type**
  Indicates the nature of the situation for which an event is created. Event types are 32-bit identifiers. The top 16 bits identify the developer, and the bottom 16 bits are allocated by the management module developer to distinguish each event type the developer creates. Event types are defined as part of management modules and are always contained in each event.
  Type: Integer
  Attribute ID: 0x11fb8
- **Creator**
  The name of the user that created the event. DX NetOps Spectrum creates this attribute when creating an event.
  Type: Text String
  Attribute ID: 0x11fb9
- **Creation Date/Time**
  The date and time of recording the event.
  Type: CsCDateTime
  Attribute ID: 0x11f4e
- **Severity**
  Defines the event severity. A higher severity event needs more attention by the network administrator or represents a bigger failure. Allowed values are 0 through 100, with 100 being the highest severity. Event severity is assigned by a management module and can be changed by users. Every event has a severity associated with it.
  Type: Integer
  Attribute ID: 0x11fb5
- **Model ID**
  The model ID for which the event was created.
  Type: Model ID
  Attribute ID: 0x11f53
- **Model Name**
  The name of the model.
  Type: Text String
  Attribute ID: 0x1006e
- **Model Type**
  The type of the model for which the event was created.
  Type: CsCMType
  Attribute ID: 0x10001
- **Model Type Name**
  Type: Text String
  Attribute ID: 0x10000

The name of the model type.

- **Type Attributes**
  Type: Boolean
  Attribute ID: 0x11fba
  Events may have event-specific attributes. You can use event-specific attributes to construct event messages from event format files. You can use this attribute to specify that the client wants DX NetOps Spectrum to send all event-specific attributes with each event.
- **Model ID**
  The ID of the model for which the event is created.
  Type: Integer
  Attribute ID: 0x129aa
- **Domain ID**
  The ID of the domain for which the event is created.
  Type: Integer
  Attribute ID: 0x129ac

Create Events Example

The following example shows how to create events:

```
import java.io.*;
import com.aprisma.spectrum.core.idl.*;
import com.aprisma.spectrum.core.util.*;
import com.aprisma.spectrum.core.idl.CsCException.*;
import com.aprisma.spectrum.core.idl.CsCAttribute.*;
import com.aprisma.spectrum.core.idl.CsCError.*;
import com.aprisma.util.corba.*;
public class CreateEvent
{
    public static void main( String[] args )
    {
        Integer modelID = Integer.decode( args[0] );
        Integer eventCode = Integer.decode( args[1] );
        CsCModelDomain md = null;

        try
        {
            String domainName = new String( "mySpectroSERVER" );

            CORBAHelper helper = CORBAHelper.getHelperImpl();
            helper.init( null, null );

            md = (CsCModelDomain) helper.getObjectImplementation(
                 CsCModelDomain.class, domainName );
            int severity = 7;
            CsCValue svalue = new CsCValue();
            svalue.gaugeValue( severity );
            CsCAttrValue sav = new CsCAttrValue(
                 CsCorbaEventHelper.SEVERITY,
                 svalue, CsCError_e.SUCCESS );
            CsCAttrValue[] ava = new CsCAttrValue[]{sav};
            CsCAttrValList avl = new CsCAttrValList( ava,
                 CsCError_e.SUCCESS );
```

```
            byte[] eventid = md.createEvent( modelID.intValue(),
                 eventCode.intValue(), avl );
        } catch ( Throwable e ) {
            System.out.println( e );
        }
    }
}
```

The previous example requires the following objects and methods:

- Package com.aprisma.spectrum.core.idl
  - Interfaces
    - CsCModelDomain
    - CsCEventDomain
  - Classes
    - CsCModelDomainHelper
- Package com.aprisma.spectrum.core.idl.CsCEventDomainPackage
  - Classes
    - CsCEvent
- Package com.aprisma.spectrum.core.util
  - Classes
    - CsCorbaEventHelper
- Package com.aprisma.spectrum.core.idl.CsCAttribute
  - Classes
    - CsCAttrValList
    - CsCAttrValue
    - CsCValue
- Package com.aprisma.spectrum.core.idl.CsCError
  - Classes
    - CsCError_e

Event Retrieval

You can retrieve events by defining search filters. First, retrieve the CsCEventDomain interface and then construct a filter type to define the event result set returned. Filters can specify a time range, particular model, event types, or a combination of criteria.

Calling the method getEventListByAttrFilter returns events to a client. The first parameter is the filter. The second parameter specifies the attributes to be returned for every event found by the filter. If a response to a request cannot be returned as one result set, CsCError::LIMIT_REACHED is returned as the error. You should call getNextEventListByAttrFilter to get the remaining events. The search limit is 10,000 records. You should pass the requestID to getNextEventListByAttrFilter to request the next group of events associated with this request. This requestID is invalid five minutes after it is returned or when all events have been returned to the client.

Retrieve Events by Type Example

The following example shows how to retrieve events by type:

```
import java.io.*;
import com.aprisma.spectrum.core.idl.*;
import com.aprisma.spectrum.core.util.*;
import com.aprisma.spectrum.core.idl.CsCEventDomainPackage.*;
import com.aprisma.spectrum.core.idl.CsCException.*;
```

```
import com.aprisma.spectrum.core.idl.CsCAttribute.*;
import com.aprisma.spectrum.core.idl.CsCError.*;
import com.aprisma.util.corba.*;
public class RetrieveEvent
{
    public static void main( String[] args )
    {
        Integer modelID = Integer.decode( args[0] );
        Integer eventCode = Integer.decode( args[1] );
        CsCModelDomain md = null;
        try
        {
            String domainName = new String( "mySpectroSERVER" );
            CORBAHelper helper = CORBAHelper.getHelperImpl();
            helper.init( null, null );

            md = (CsCModelDomain) helper.getObjectImplementation(
                CsCModelDomain.class, domainName );
            CsCEventDomain ed = (CsCEventDomain)
                helper.getObjectImplementation(
                    CsCEventDomain.class, domainName );
            CsCorbaFilterAttrNode node =
                CsCorbaAttrFilterHelper.createGaugeValueNode(
                    CsCorbaEventHelper.EVENT_TYPE,
                    CsCOperator_e.CSC_EQUALS, eventCode.intValue());
            CsCValue[] filter = node.getFilter();
            int[] attrIDs = new int[4];
            attrIDs[0] = CsCorbaEventHelper.CREATOR;
            attrIDs[1] = CsCorbaEventHelper.MODEL_NAME;
            attrIDs[2] = CsCorbaEventHelper.MODEL_TYPE_NAME;
            attrIDs[3] = CsCorbaEventHelper.TYPE_ATTRIBUTES;
            org.omg.CORBA.IntHolder requestID =
                new org.omg.CORBA.IntHolder();
            CsCEventList el = ed.getEventListByAttrFilter(filter,
                attrIDs, requestID);
            System.out.println("Overall:" + el.error.toString());
            if (el.error == CsCError_e.SUCCESS ||
                el.error == CsCError_e.LIMIT_REACHED )
            {
                printEventInfo( el );
                if (el.error == CsCError_e.LIMIT_REACHED)
                {
                    el = ed.getNextEventListByAttrFilter(
                        requestID.value);
                    printEventInfo(el);
                }
            }
        } catch ( Throwable e ) {
            System.out.println( e );
        }
    }
    public static void printEventInfo( CsCEventList el )
    {
```

```
        for (int i = 0; i < el.list.length; i++)
        {
            CsCEvent event = el.list[i];
            CsCAttrValList avl = event.attrValList;
            System.out.println("EventCode:" + event.type);
            for (int j = 0; j < avl.list.length; j++)
            {
                CsCAttrValue av = avl.list[j];
                System.out.println("AttrID:" + av.attributeID);
                CsCValue val = av.value;
                System.out.println("AttrVal:" + val.toString());
            }
        }
    }
}
```

The previous example requires the following objects and methods:

- Package com.aprisma.spectrum.core.idl
  - Interfaces
    - CsCModelDomain
    - CsCEventDomain
  - Classes
    - CsCModelDomainHelper
- Package com.aprisma.spectrum.core.idl.CsCEventDomainPackage
  - Classes
    - CsCEvent
- Package com.aprisma.spectrum.core.util
  - Classes
    - CsCorbaEventHelper
- Package com.aprisma.spectrum.core.idl.CsCAttribute
  - Classes
    - CsCAttrValList
    - CsCAttrValue
    - CsCValue
- Package com.aprisma.spectrum.core.idl.CsCError
  - Classes
    - CsCError_e

Alarms in Development API

The SPECTRUM Development API creates alarms through the event creation process. When an event is created, it is mapped to an entry in the EventDisp files found in a subdirectory of *$SPECROOT*/SS/CsVendor.

For example, the EventDisp file for the mtype 0x1c8000a resides in the *$SPECROOT*/SS/CsVendor/Ctron_SSAH directory and resembles the following file:

```
0x00010306 E 30
0x00010307 E 30
0x00010308 E 50 A 1,0x010308
0x00010309 E C 0x010308
0x0001030a E 50 A 1,0x01030a
```

```
0x0001030b E 50 A 1,0x01030b
0x021b070a E 50
0x021b070b E 50 A 1,0x021b070b
0x021b070c E 50
0x021b070d E 50
0x000d01a0 E 50 A 1,0x0d01a0
0x000d01a1 E 50 A 1,0x0d01a1
0x000d01a2 E 50 A 1,0x0d01a2
0x000d01a3 E 50 A 1,0x0d01a3
0x000d0351 E 50 A 1,0x0d0351
0x000d0001 E 50 A 1,0x0d0001
0x000d0002 E 50 A 1,0x0d0002
0x00830000 E 30
0x00830001 E 30
0x00830002 E 30
0x003c0002 E 50 A 1,0x003c0002
0x000d2afb E 30
```

The numbers in the left column are event codes. Code E tells DX NetOps Spectrum to log the event; Code C tells DX NetOps Spectrum to clear the event; Code A tells DX NetOps Spectrum to generate an alarm; 1 specifies the severity of the alarm. The numbers in the right column indicate the probable cause to display with the alarm.

CsCAlarmDomain provides access to alarms of a model domain. Alarms indicate a situation, in a monitored entity such as a router or application, that requires the attention of product operators. The alarm priority suggests an order in which to address the situations.

The DX NetOps Spectrum alarm system delivers only the most important alarms to clients. You can do this task in two ways such as, deliver only what a client requests, or prioritize the alarm delivery. Only the basic information like the alarm ID, cause, priority, and severity is shipped with alarms by default. If desired, clients must specify additional characteristics, which are defined by attribute IDs that are sent with each alarm.

To prioritize alarm delivery, clients specify prioritized alarm filters. The filters define which alarms are shipped to the client. Prioritizing the filters lets DX NetOps Spectrum generate only high-priority alarms. For example, prioritization is useful in situations where some alarms are delayed due to a high volume of alarm updates. The priority that is assigned to an alarm depends on the priority alarm filters of all the clients that receive the alarm.

The client should use the interface in a way that minimizes the amount of data that is sent to the client, while clarifying the priority of client alarm requests.

Alarm Attributes

DX NetOps Spectrum offers two categories of attributes, Alarm and Model. Alarm attributes exist only within the context of an alarm. Model attributes are stored with models. Clients can select the attributes to include with alarms. For example, you can configure alarms to include the model name and troubleshooter name. Model name is a model attribute and troubleshooter name is an alarm attribute. Letting clients select model attributes for each alarm minimizes the need for clients to query DX NetOps Spectrum for additional data when a new alarm is received.

Verify the following alarm attributes and their descriptions:

- **Acknowledged**
  (Read/Write) Indicates the state of the alarm acknowledgment, that is, whether the user has seen the alarm.
  Type: Boolean
  Attribute ID: 0x11f4d
- **Alarm ID**
  (Read Only) Uniquely identifies the alarm independent of the model domain where the alarm was created. This attribute is automatically sent with every alarm.

Type: CsCAlarmID
Attribute ID: 0x11f9c

- **Alarm Status**
  (Read/Write) Conveys information about the alarm. Status always indicates whether repair is complete.
  Type: string
  Attribute ID: 0x11f4f
- **Alarm Cause**
  (Read Only) Identifies the alarm cause and the nature of the situation for which an alarm is created. This attribute is automatically sent with every alarm so applications need not explicitly request for it.
  Type: CsCCauseID
  Attribute ID: 0x11f50
- **Alarm Source**
  (Read Only) An enumerated value that indicates the alarm source. Currently, the following values are supported.
  – **0**
  Specifies that the alarm is current.
  – **1**
  Specifies that the alarm is residual; it exists from a previous run of the SpectroSERVER.
  Type: enum
  Attribute ID: 0x11fc4
- **Cleared by Name**
  (Read Only) Specifies the user name that cleared the alarm through the API. This attribute is not present when an alarm is created but is added prior to removal.
  Type: string
  Attribute ID: 0x11f51
- **Creation Date**
  (Read Only) The time and date of the alarm creation.
  Type: CsCDateTime
  Attribute ID: 0x11f4e
- **Event ID List**
  (Read/Write) Specifies the list of events that are associated with the alarm. Include the events that caused the alarm to be generated.
  Type: CsCEventIDList
  Attribute ID: 0x11f52
- **Model ID**
  (Read Only) Identifies the model with the situation reported by the alarm.
  Type: Model ID
  Attribute ID: 0x11f53
- **Occurrences**
  (Read Only) Contains the number of occurrences of the alarm. The count is updated when the situation that asserted the alarm recurs through a trap, event, or IH. This attribute is initialized to 1 as the initial alarm is the first occurrence.
  Type: unsigned long (gauge)
  Attribute ID: 0x11fc5
- **Last Occurrence Date**
  (Read Only) Indicates the time and date when the alarm last occurred.
  Type: CsCDateTime
  Attribute ID: 0x1321a
- **Primary Alarm**
  (Read Only) Indicates whether the alarm is a primary alarm. Primary alarms are the highest priority for a model. A model can have more than one primary alarm at a time.
  Type: Boolean

Attribute ID: 0x11f54

- **Resolved**
  (Read Only) Indicates whether the alarm is resolved internally by DX NetOps Spectrum.
  Type: Boolean
  Attribute ID: 0x11f55
- **Severity**
  (Read Only) Indicates the alarm severity. This attribute is automatically sent with every alarm so applications need not explicitly request for it.
  Type: CsCAlarmSeverity
  Attribute ID: 0x11f56
- **Troubleshooter**
  (Read/Write) Specifies the name of the troubleshooter that is assigned to repair the alarm.
  Type: string
  Attribute ID: 0x11f57

  NOTE
  When this attribute is written, the troubleshooter model attribute is updated accordingly, if the name corresponds to a valid troubleshooter model for the user.
- **Troubleshooter Model**
  (Read/Write) Identifies the model of the troubleshooter that is assigned to repair the alarm.
  Type: CsCModel
  Attribute ID: 0x11fc6

  NOTE
  When this attribute is written, the troubleshooter name attribute is updated accordingly. However, if the model is not valid or is not associated with a user, the write fails and returns the error ATTR_BAD_VALUE.
- **Trouble Ticket ID**
  (Read/Write) Contains the ID of the trouble ticket that is associated with the alarm.
  Type: string
  Attribute ID: 0x12022
- **User Clearable**
  (Read Only) Indicates whether a client can clear the alarm. This attribute is provided to let client applications indicate to users that the alarm is not clearable. A client cannot control whether an alarm can be cleared.
  Type: Boolean
  Attribute ID: 0x11f9b
- **Model ID**
  (Read Only) Identifies the ID of the model with the situation reported by the alarm.
  Type: integer
  Attribute ID: 0x129aa
- **Model Type ID**
  (Read Only) Specifies the ID of the model type.
  Type: integer
  Attribute ID: 0x129ab

Retrieving an alarm is similar to retrieving an event. Retrieve events by defining search filters for alarms. First, retrieve the CsCAlarmDomain interface and then construct a filter to define the alarm result set. Filters can specify a time range, a particular model, alarm type, or a combination of criteria. You can use all attributes that are listed here for the filter. The CsCorbaAlarmHelper class defines all alarm attributes. In addition, the helper class can convert alarm ID from byte array to text string and text string to byte array. Alarm IDs are unique like event IDs and have long formats.

Retrieve Alarms by Model ID Example

The following example shows how to retrieve alarms by model ID:

```
import java.io.*;
import com.aprisma.spectrum.core.idl.*;
import com.aprisma.spectrum.core.util.*;
import com.aprisma.spectrum.core.idl.CsCException.*;
import com.aprisma.spectrum.core.idl.CsCAttribute.*;
import com.aprisma.spectrum.core.idl.CsCAlarmDomainPackage.*;
import com.aprisma.spectrum.core.idl.CsCError.*;
import com.aprisma.util.corba.*;
public class RetrieveAlarms
{
    public static void main( String[] args )
    {
        Integer modelID = Integer.decode( args[0] );
        CsCModelDomain md = null;

        try
        {
            String domainName = new String( "train124" );

            CORBAHelper helper = CORBAHelper.getHelperImpl();
            helper.init( null, null );
            md = (CsCModelDomain) helper.getObjectImplementation(
                CsCModelDomain.class, domainName );
            CsCorbaFilterAttrNode node =
                CsCorbaAttrFilterHelper.createModelIDNode(
                        CsCorbaAlarmHelper.MODEL_HANDLE,
                        CsCOperator_e.CSC_EQUALS, modelID.intValue() );
            CsCValue[] filter = node.getFilter();
            int[] attrIDs = new int[2];
            attrIDs[0] = CsCorbaAlarmHelper.CREATION_DATE;
            attrIDs[1] = CsCorbaAlarmHelper.MODEL_HANDLE;

            CsCAlarmDomain ad = md.getAlarmDomain();
            CsCAlarmList al = ad.getAlarmListByAttrFilter( filter,
                attrIDs );
            System.out.println( "Status:" + al.error.toString() );
            for ( int i=0; i < al.list.length; i++ ) {
                CsCAlarm alarm = al.list[i];
                System.out.println( "Priority:" + alarm.priority );
                System.out.println( "Severity:" + alarm.severity );

                CsCAttrValList avl = alarm.attrValList;
                System.out.println( "CreationDate: " +
                    avl.list[0].value.toString() );
                System.out.println( "ModelHandle: " +
                    avl.list[1].value.toString() );
            }
        } catch ( Throwable e ) {
            System.out.println( e );
        }
    }
}
```

This example requires the following objects and methods:

- Package com.aprisma.spectrum.core.idl
  - Interfaces
    - CsCModelDomain
    - CsCAlarmDomain
  - Classes
    - CsCModelDomainHelper
- Package com.aprisma.spectrum.core.idl.CsCAlarmDomainPackage
  - Classes
    - CsCAlarm
    - CsCAlarmList
- Package com.aprisma.spectrum.core.util
  - Interfaces
    - CsCorbaFilterAttrNode
  - Classes
    - CsCorbaAlarmHelper
    - CsCorbaAttrFilterHelper
- Package com.aprisma.spectrum.core.idl.CsCAttribute
  - Classes
    - CsCAttrValList
    - CsCAttrValue
    - CsCValue
- Package com.aprisma.spectrum.core.idl.CsCError
  - Classes
    - CsCError_e

Updating and Clearing Alarms

You can update alarm attributes as you update model attributes. The following example shows how to update the trouble ticket ID through a trouble ticketing system. You can also use other attributes such as Trouble_Shooter and Resolved for automation of alarms.

```
byte[][] alarms = new byte[][]{alarm.alarmID};

CsCValue val = new CsCValue();
val.textString("123123");
CsCAttrValue attrVal = new
    CsCAttrValue(CsCorbaAlarmHelper.TROUBLE_TICKET_ID, val,
    CsCError_e.SUCCESS);
CsCAttrValue[] attrValArray = new CsCAttrValue[1];
attrValArray[0] = attrVal;
CsCAttrValList writeValList = new CsCAttrValList(attrValArray,
    CsCError_e.SUCCESS);
try
{
    ad.writeAttrValListOfAlarms(alarms, writeValList);
}
catch (Throwable e)
{
    System.out.println(e);
```

```
}
```

Line 1 takes the Alarm ID byte array and assigns the value to a two-dimensional array if there are multiple alarms to be updated at once. Lines 3-10 construct a CsCAttrValList for setting TROUBLE_TICKET_ID. Line 14 makes the call to write the attribute value.

You can clear alarms automatically through events and EventDisp files. You can also use the following API to clear alarms:

```
try
{
ad.clearAlarm(alarm.alarmID);
}
catch (Throwable e)
{
System.out.println(e);
}
```

Watches using the Dev API

You can register with DX NetOps Spectrum for asynchronous notification of events, such as the following:

- Model creation and deletion
- Attribute value changes
- Event and statistic creation
- Alarm updates

A client application obtains references to objects on the server and uses these stubs to make remote method invocations to the objects. To register for asynchronous notification, the client itself instantiates a watch object and passes a reference to the server instructing it to call a method in the object if a specified condition occurs. If the specified condition occurs, the server calls the method in the client's watch object and notifies the client.

If you are writing a client application that is notified when the value of a particular attribute in the DX NetOps Spectrum knowledge base changes, you should first design an attribute value change class, for example, AttrValCallback. The client program instantiates the class and passes an object reference (and the attribute ID to watch) to DX NetOps Spectrum. If the attribute value changes, DX NetOps Spectrum calls the attrValsChanged method in your AttrValCallback class.

The important factors in changing attribute values include the following:

- For DX NetOps Spectrum to detect a change in the attribute, the memory flag must be set. If not set, the attribute change mechanism does not work because DX NetOps Spectrum does not retain the last known value for comparison.
- For external attributes, you should set the polled flag. Without the mechanism to poll every given period externally, DX NetOps Spectrum never detects a change until some other program reads the attribute.

How To Set Up a Watch

Setting up a watch involves the following steps:

1. Create a class file for the callback and extend it from the desired callback class.
2. Create a main class that instantiates the callback class and passes it to the respective domain.

The following table lists the available callback classes and their domains:

Monitor	Extend Callback Class	Domain
Attribute Value Changes	CsCAttrValWatchCBPOA	Model Domain
Model Creation/Destroy	CsCModelWatchCBPOA	Model Domain
Model Association Changes	CsCAssocModelWatchCBPOA	Model Domain
Event Creation	CsCEventWatchCBPOA	Event Domain
Alarm Creation/Change/ Clear	CsCAlarmWatchCBPOA	Alarm Domain

Watch Attribute Changes for a Model ID Example

The following example shows how to watch attribute changes for a given model ID:

```
import java.io.*;
import com.aprisma.spectrum.core.idl.* ;
import com.aprisma.spectrum.core.util.* ;
import com.aprisma.util.corba.* ;
import com.aprisma.spectrum.core.idl.CsCAttribute.* ;
import com.aprisma.spectrum.core.idl.CsCModelPackage.* ;
public class WatchAttrValue
{
    public static void main(String[] args)
    {
        try
        {
            String domainName =  new String("mySpectroSERVER");
            CORBAHelper hlp = CORBAHelper.getHelperImpl();
            hlp.init(null, null);
            CsCModelDomain md = (CsCModelDomain)
               hlp.getObjectImplementation
                   (CsCModelDomain.class, domainName);

            AttrValCallback callback = new AttrValCallback();
            //decode model ID
            int[] modelIDs = {Integer.decode(args[0]).intValue()};
            int[] attrs = {0x1006e}; //model name
            CsCAttrReadMode_e[] readModes =
                {CsCAttrReadMode_e.CSC_MOST_AVAILABLE};
            CsCAttrValWatchCB cb =
                CsCAttrValWatchCBHelper.narrow(
                    hlp.servant_to_reference( callback ) );
            CsCAttrValListOfModels avlm =
                md.startWatchAttrValsOfModelsByIDs(modelIDs,
                    attrs, readModes, cb);
            System.out.println("Watching started.." +
              " Press Enter to exit");
            System.in.read();
            md.stopWatchAttrValsOfModelsByIDs(cb, attrs, modelIDs);
        }
        catch (Throwable e)
        {
            System.out.println(e);
```

```
        }
    }
}
import java.io.*;
import com.aprisma.spectrum.core.idl.* ;
import com.aprisma.spectrum.core.idl.CsCAttribute.* ;
import com.aprisma.spectrum.core.idl.CsCException.*;
import com.aprisma.spectrum.core.idl.CsCModelPackage.*;
import com.aprisma.spectrum.core.idl.CsCError.*;
public class AttrValCallback extends CsCAttrValWatchCBPOA
{
    public void attrValsChanged ( CsCAttrValListOfModels avlom)
    {
        System.out.println("Overall:" + avlom.error.toString());
        for (int i = 0; i < avlom.list.length; i++)
        {
            CsCModelAttrValList mavl = avlom.list[i];
            System.out.println("Model ID:" + mavl.modelID);
            CsCAttrValList avl = mavl.attrValList;
            for (int j = 0; j < avl.list.length; j++)
            {
                CsCAttrValue av = avl.list[j];
                CsCValue v = av.value;
                System.out.println("Attr ID:" + av.attributeID +
                    " --> Value:" + v.textString());
            }
        }
    }

}
```

This example requires the following objects and methods:

- Package com.aprisma.spectrum.core.idl
  - Interfaces
    - CsCModelDomain
    - CsCAttrValWatchCBPOA
  - Classes
    - CsCModelDomainHelper
- Package com.aprisma.spectrum.core.idl.CsCModelPackage
  - Classes
    - CsCAttrValListOfModels
    - CsCModelAttrValList
- Package com.aprisma.spectrum.core.idl.CsCAttribute
  - Classes
    - CsCAttrValList
    - CsCAttrValue
    - CsCValue
- Package com.aprisma.spectrum.core.idl.CsCError
  - Classes
    - CsCError_e

Watch Model Creation and Deletion Example

If a model is deleted, you can retrieve only the model ID. If you need more information, you should make a repository keyed on model ID.

The following example shows how to watch model creation and deletion:

```
import java.io.*;
import com.aprisma.spectrum.core.idl.*;
import com.aprisma.spectrum.core.util.*;
import com.aprisma.spectrum.core.idl.CsCException.*;
import com.aprisma.spectrum.core.idl.CsCModelPackage.*;
import com.aprisma.spectrum.core.idl.CsCError.*;
import com.aprisma.util.corba.*;
public class WatchModel
{
    public static void main( String[] args )
    {
        Integer mtypeID = Integer.decode( args[0] );
        int[] mtypeIDs = { mtypeID };
        CsCModelDomain md = null;
        try
        {
            String domainName = new String( "mySpectroSERVER" );
            CORBAHelper helper = CORBAHelper.getHelperImpl();
            helper.init( null, null );
            md = (CsCModelDomain) helper.getObjectImplementation(
                CsCModelDomain.class, domainName );

            ModelCallback callback = new ModelCallback();
            CsCModelWatchCB cb =
            CsCModelWatchCBHelper.narrow(
                helper.servant_to_reference(callback));
            md.startWatchModelsByTypeIDs(mtypeIDs, cb);
            System.out.println( "Press Enter to exit" );
            System.in.read();
            md.stopWatchModelsByTypeIDs(mtypeIDs, cb);
        }
        catch (Throwable e)
        {
            System.out.println( e );
        }
    }
}
import java.io.*;
import com.aprisma.spectrum.core.idl.*;
import com.aprisma.spectrum.core.util.*;
import com.aprisma.spectrum.core.idl.CsCException.*;
import com.aprisma.spectrum.core.idl.CsCModelPackage.*;
import com.aprisma.spectrum.core.idl.CsCError.*;
import com.aprisma.util.corba.*;
public class ModelCallback extends CsCModelWatchCBPOA
{
    public void modelsAdded( CsCModelPropList mpl )
```

```
    {
        System.out.println( "Overall: " + mpl.error.toString() );
        for ( int i=0; i < mpl.list.length; i++ )
        {
            CsCModelProperties mp = mpl.list[i];
            System.out.println("ModelID:" + mp.modelID);
        }
    }
    public void modelsRemoved( CsCModelPropList mpl )
    {
        System.out.println( "Overall: " + mpl.error.toString() );
        for ( int i=0; i < mpl.list.length; i++ )
        {
            CsCModelProperties mp = mpl.list[i];
            System.out.println( "ModelID:" + mp.modelID );
        }
    }
}
```

This example requires the following objects and methods:

- Package com.aprisma.spectrum.core.idl
  - Interfaces
    - CsCModelDomain
    - CsCModelWatchCBPOA
  - Classes
    - CsCModelDomainHelper
- Package com.aprisma.spectrum.core.idl.CsCModelPackage
  - Classes
    - CsCModelPropList
    - CsCModelProperties
- Package com.aprisma.spectrum.core.idl.CsCError
  - Classes
    - CsCError_e

Watch Model Association Changes Example

The following example shows how to watch model association changes:

```
import java.io.*;
import com.aprisma.spectrum.core.idl.*;
import com.aprisma.spectrum.core.util.*;
import com.aprisma.spectrum.core.idl.CsCException.*;
import com.aprisma.spectrum.core.idl.CsCModelPackage.*;
import com.aprisma.spectrum.core.idl.CsCRelationPackage.*;
import com.aprisma.spectrum.core.idl.CsCError.*;
import com.aprisma.util.corba.*;
public class WatchAssocModel
{
    public static void main( String[] args )
    {
        Integer modelID = Integer.decode( args[0] );
        int[] modelIDs = { modelID };
```

```
        int relationID = 0x10002;
        CsCModelDomain md = null;
        try
        {
            String domainName = new String( "mySpectroSERVER" );
            CORBAHelper helper = CORBAHelper.getHelperImpl();
            helper.init( null, null );
            md = (CsCModelDomain) helper.getObjectImplementation(
                CsCModelDomain.class, domainName );

            AssocModelCallback callback = new AssocModelCallback();
            CsCAssocModelWatchCB cb =
            CsCAssocModelWatchCBHelper.narrow(
                helper.servant_to_reference(callback));
            md.startWatchAssocModelsOfModels(relationID,modelIDs,
                CsCSide_e.CSC_RIGHT_SIDE, cb);
            System.out.println( "Press Enter to exit" );
            System.in.read();
            md.stopWatchAssocModelsOfModelsByIDs(cb, modelIDs);
        }
        catch (Throwable e)
        {
            System.out.println( e );
        }
    }
}

import java.io.*;
import com.aprisma.spectrum.core.idl.*;
import com.aprisma.spectrum.core.util.*;
import com.aprisma.spectrum.core.idl.CsCException.*;
import com.aprisma.spectrum.core.idl.CsCModelPackage.*;
import com.aprisma.spectrum.core.idl.CsCError.*;
import com.aprisma.util.corba.*;
public class AssocModelCallback extends CsCAssocModelWatchCBPOA
{
    public void associatedModelsAdded(CsCAssocModelPropList ampl)
    {
        System.out.println( "Overall: " + ampl.error.toString() );
        System.out.println( "Added Assoc: " + ampl.relationID);
        System.out.println( " For Model:  " + ampl.withModel.modelID);
        System.out.println( " On Side:    " +
                                          ampl.withSide.toString());
        CsCModelPropList mpl = ampl.assocModelList;
        for ( int i=0; i < mpl.list.length; i++ )
        {
            CsCModelProperties mp = mpl.list[i];
            System.out.println("   ModelID:" + mp.modelID);
        }
    }
    public void associatedModelsRemoved(CsCAssocModelPropList ampl)
    {
        System.out.println("Overall: " + ampl.error.toString());
```

```
        System.out.println("Remove Assoc: " + ampl.relationID);
        System.out.println(" For Model:   " + ampl.withModel.modelID);
        System.out.println(" On Side:     " +
                                            ampl.withSide.toString());
        CsCModelPropList mpl = ampl.assocModelList;
        for ( int i=0; i < mpl.list.length; i++ )
        {
            CsCModelProperties mp = mpl.list[i];
            System.out.println( "   ModelID:" + mp.modelID );
        }
    }
}
```

The previous example passes the model type ID and adds or removes the device model type through OneClick. The previous example requires the following objects and methods:

- Package com.aprisma.spectrum.core.idl
  - Interfaces
    - CsCModelDomain
    - CsCModelAssocWatchCBPOA
    - AssocModelCallback
- Package com.aprisma.spectrum.core.idl.CsCModelPackage
  - Classes
    - CsCModelPropList
    - CsCModelProperties
- Package com.aprisma.spectrum.core.idl.CsCRelationPackage
  - Classes
    - CsCSide_e
- Package com.aprisma.spectrum.core.idl.CsCError
  - Classes
    - CsCError_e

Watch Events Created on a Model Example

The following example shows how to watch events created on a model:

```
import java.io.*;
import com.aprisma.spectrum.core.idl.*;
import com.aprisma.spectrum.core.util.*;
import com.aprisma.spectrum.core.idl.CsCException.*;
import com.aprisma.spectrum.core.idl.CsCAttribute.*;
import com.aprisma.spectrum.core.idl.CsCError.*;
import com.aprisma.util.corba.*;
public class WatchEvent
{
    public static void main( String[] args )
    {
        Integer modelID = Integer.decode( args[0] );
        CsCModelDomain md = null;
        try
        {
            String domainName = new String( "mySpectroSERVER" );
            CORBAHelper helper = CORBAHelper.getHelperImpl();
```

```
            helper.init( null, null );
            md = (CsCModelDomain) helper.getObjectImplementation(
                CsCModelDomain.class, domainName );
            CsCorbaFilterAttrNode node =
                CsCorbaAttrFilterHelper.createModelIDNode(
                        CsCorbaEventHelper.MODEL_HANDLE,
                        CsCOperator_e.CSC_EQUALS, modelID.intValue() );
            CsCValue[] filter = node.getFilter();
            int[] attrIDs = new int[5];
            attrIDs[0] = CsCorbaEventHelper.CREATOR;
            attrIDs[1] = CsCorbaEventHelper.EVENT_TYPE;
            attrIDs[2] = CsCorbaEventHelper.MODEL_NAME;
            attrIDs[3] = CsCorbaEventHelper.MODEL_TYPE_NAME;
            attrIDs[4] = CsCorbaEventHelper.TYPE_ATTRIBUTES;
            CsCEventDomain ed = (CsCEventDomain)
                helper.getObjectImplementation(
                    CsCEventDomain.class, domainName );
            EventCallback callback = new EventCallback();
            CsCEventWatchCB cb = CsCEventWatchCBHelper.narrow(
                helper.servant_to_reference(callback));
            ed.startWatchEvents( filter, attrIDs, cb );
            System.out.println( "Press Enter to exit" );
            System.in.read();
            ed.stopWatchEvents( cb );
        }
        catch (Throwable e)
        {
            System.out.println( e );
        }
    }
}

import java.io.*;
import com.aprisma.spectrum.core.idl.*;
import com.aprisma.spectrum.core.util.*;
import com.aprisma.spectrum.core.idl.CsCException.*;
import com.aprisma.spectrum.core.idl.CsCAttribute.*;
import com.aprisma.spectrum.core.idl.CsCEventDomainPackage.*;
import com.aprisma.spectrum.core.idl.CsCError.*;
import com.aprisma.util.corba.*;
public class EventCallback extends CsCEventWatchCBPOA
{
    public void eventsCreated( CsCEventList el )
    {
        System.out.println( "Overall: " + el.error.toString() );
        for ( int i=0; i < el.list.length; i++ ) {
            CsCEvent event = el.list[i];
            CsCAttrValList avl = event.attrValList;
            System.out.println( "EventCode: " + event.type );
            for ( int j=0; j < avl.list.length; j++ )
            {
                CsCAttrValue av = avl.list[j];
                System.out.println( "AttrID: " + av.attributeID );
```

```
                CsCValue val = av.value;
                System.out.println( "AttrVal: " + val.toString() );
            }
        }
    }
    public void eventAttrValsUpdated( CsCEventList el )
    {
        System.out.println( "Overall: " + el.error.toString() );
        for ( int i=0; i < el.list.length; i++ )
        {
            CsCEvent event = el.list[i];
            CsCAttrValList avl = event.attrValList;
            System.out.println( "EventCode: " + event.type );
            for ( int j=0; j < avl.list.length; j++ )
            {
                CsCAttrValue av = avl.list[j];
                System.out.println( "AttrID: " + av.attributeID );
                CsCValue val = av.value;
                System.out.println( "AttrVal: " + val.toString() );
            }
        }
    }
}
```

The previous example passes a model filter and an attribute list to be returned by the callback. The previous example requires the following objects and methods:

- Package com.aprisma.spectrum.core.idl
  - Interfaces
    - CsCModelDomain
    - CsCEventWatchCBPOA
  - Classes
    - CsCModelDomainHelper
- Package com.aprisma.spectrum.core.idl.CsCEventDomainPackage
  - Classes
    - CsCEvent
- Package com.aprisma.spectrum.core.util
  - Interface
    - CsCorbaFilterAttrNode
  - Classes
    - CsCorbaAttrFilterHelper
    - CsCorbaEventHelper
- Package com.aprisma.spectrum.core.idl.CsCAttribute
  - Classes
    - CsCAttrValList
    - CsCAttrValue
    - CsCValue
- Package com.aprisma.spectrum.core.idl.CsCError
  - Classes
    - CsCError_e

Watch Alarm Creation, Clearing, and Attribute Updates Example

The following example shows how to watch alarm creation, attribute updates, and clearing:

```
import java.io.*;
import com.aprisma.spectrum.core.idl.*;
import com.aprisma.spectrum.core.util.*;
import com.aprisma.spectrum.core.idl.CsCException.*;
import com.aprisma.spectrum.core.idl.CsCAttribute.*;
import com.aprisma.spectrum.core.idl.CsCAlarmDomainPackage.*;
import com.aprisma.spectrum.core.idl.CsCError.*;
import com.aprisma.util.corba.*;
public class WatchAlarms
{
    public static void main( String[] args )
    {
        Integer modelID = Integer.decode( args[0] );
        CsCModelDomain md = null;
        try
        {
            String domainName = new String( "mySpectroSERVER" );
            CORBAHelper helper = CORBAHelper.getHelperImpl();
            helper.init( null, null );
            md = (CsCModelDomain) helper.getObjectImplementation(
                CsCModelDomain.class, domainName );
            CsCAlarmDomain ad = md.getAlarmDomain();
            CsCorbaFilterAttrNode node =
                CsCorbaAttrFilterHelper.createModelIDNode(
                        CsCorbaAlarmHelper.MODEL_HANDLE,
                        CsCOperator_e.CSC_EQUALS, modelID.intValue() );
            CsCValue[] filter = node.getFilter();
            int[] attrIDs = new int[1];
            attrIDs[0] = CsCorbaAlarmHelper.CREATION_DATE;
            AlarmCallback callback = new AlarmCallback();
            CsCAlarmWatchCB cb =
                   CsCAlarmWatchCBHelper.narrow(
                        helper.servant_to_reference(callback));
            ad.startWatchAlarms( filter, attrIDs, cb );
            System.out.println( "Press Enter to exit" );
            System.in.read();
            ad.stopWatchAlarms( cb );
        }
        catch (Throwable e)
        {
            System.out.println( e );
        }
    }
}
import java.io.*;
import com.aprisma.spectrum.core.idl.*;
import com.aprisma.spectrum.core.util.*;
import com.aprisma.spectrum.core.idl.CsCException.*;
import com.aprisma.spectrum.core.idl.CsCAttribute.*;
import com.aprisma.spectrum.core.idl.CsCAlarmDomainPackage.*;
```

```
import com.aprisma.spectrum.core.idl.CsCError.*;
import com.aprisma.util.corba.*;
public class AlarmCallback extends CsCAlarmWatchCBPOA
{
    public void alarmsUpdated( CsCAlarmUpdate au )
    {
        CsCAlarmList addlist = au.addList;
        CsCAttrValListOfAlarms changelist = au.changeList;

        System.out.println( "addList len:" + addlist.list.length +
                    " changeList len:" +
                    changelist.list.length );

        for ( int i=0; i < addlist.list.length; i++ ) {
            CsCAlarm alarm = addlist.list[i];
            System.out.println( "Added AlarmID:" +
                    CsCorbaAlarmHelper.toString( alarm.alarmID ) );
                        System.out.println("Value: " +
                           alarm.attrValList.list[0].value );
                        System.out.println("Value: " +
                           alarm.attrValList.list[1].value );
        }
        for ( int j=0; j < changelist.list.length; j++ ) {
            System.out.println( "Changed AlarmID:" +
                    CsCorbaAlarmHelper.toString(
                       changelist.list[j].alarmID ) );
        }
        for (int k = 0; k < au.removeList.length; k++)
        {
            System.out.println("Removed AlarmID:" +
                    CsCorbaAlarmHelper.toString(au.removeList[k]));
        }
    }
}
```

The previous example passes a model filter and an attribute list to be returned by the callback to the startWatch method. In the AlarmCallback class where DX NetOps Spectrum notifies this class, alarms are of three types: newly created, updated, and removed.

The previous example requires the following objects and methods:

- Package com.aprisma.spectrum.core.idl
  - Interfaces
    - CsCModelDomain
    - CsCAlarmDomain
  - Classes
    - CsCModelDomainHelper
- Package com.aprisma.spectrum.core.idl.CsCAlarmDomainPackage
  - Classes

    - CsCAlarm
    - CsCAlarmList
    - CsCAlarmUpdate
    - CsCAttrValListOfAlarms
- Package com.aprisma.spectrum.core.util
  – Interfaces
    - CsCorbaFilterAttrNode
  – Classes
    - CsCorbaAlarmHelper
    - CsCorbaAttrFilterHelper
- Package com.aprisma.spectrum.core.idl.CsCAttribute
  – Classes
    - CsCAttrValList
    - CsCAttrValue
    - CsCValue
- Package com.aprisma.spectrum.core.idl.CsCError
  – Classes
    - CsCError_e

Miscellaneous Considerations

Distributed Environments

Use the following checklist to verify that the required files and configuration are present when running a client from another workstation:

- User creation in the SpectroSERVER
- Appropriate .hostrc permissions
- Hostname resolution from either DNS or /etc/hosts
- Access control lists on any router between the client and server
- Firewall-blocking ports

> **NOTE**
> For more information about setting up a distributed SpectroSERVER environment, see Distributed SpectroSERVER Administration .

Location Service Interface

To get a list of SpectroSERVERs in a distributed environment, you can use the interface CsCLocServMapInt. The Location Service advertises on this interface and only one call is available to retrieve the SpectroSERVER list. Once the CsCModelDomain interface reference is retrieved, the isPrimary method is called to determine if the interface is primary or secondary.

> **NOTE**
> Calling the isPrimary method is one way of knowing whether the SpectroSERVER is the primary or secondary in a fault-tolerant environment.

Binding to Multiple Domains Example

The following example shows the method for multiple domain binding:

```
import java.io.*;
import com.aprisma.spectrum.core.idl.*;
import com.aprisma.spectrum.core.util.*;
import com.aprisma.spectrum.core.idl.CsCException.*;
import com.aprisma.spectrum.core.idl.CsCModelDomainPackage.*;
import com.aprisma.spectrum.core.idl.CsCError.*;
import com.aprisma.util.corba.*;
public class GetDomainList
{
    public static void main(String[] args)
    {
        CsCorbaValueHelper help = new CsCorbaValueHelper();
        CsCLocServMapInt locserv;
        CORBAHelper helper = null;
        String[] domainnames = { "" };
        CsCModelDomain[] md;
        helper = CORBAHelper.getHelperImpl();
        try
        {
            String domainName = new String("mySpectroSERVER");
            helper.init(null, null);
            locserv = (CsCLocServMapInt)helper.getObjectImplementation(
                CsCLocServMapInt.class, domainName);
            domainnames = locserv.getModelDomainNameList();
        }
        catch (Throwable e)
        {
            System.out.println(e);
        }
        md = new CsCModelDomain[domainnames.length];
        for (int i = 0; i < domainnames.length; i++)
        {
            try
            {
                md[i] =
                   (CsCModelDomain)helper.getObjectImplementation(
                   CsCModelDomain.class, domainnames[i]);
                System.out.println("Name:" + domainnames[i]);
                System.out.println("Is Primary: " + md[i].isPrimary());
            }
            catch (Throwable e)
            {
                System.out.println(e);
            }
        }
    }
}
```

This example requires the following objects and methods:

- Package com.aprisma.spectrum.core.idl

  - Interfaces
    - CsCLocServMapInt
    - CsCModelDomain
  - Classes
    - CsCLocServMapIntHelper
    - CsCModelDomainHelper
- Package com.aprisma.spectrum.core.idl.CsCError
  - Classes
    - CsCError_e

Status Monitoring of CORBA Interface

Server bouncing may cause non-persistent interfaces and watch reference to be lost except for persistent interfaces like ModelDomain, AlarmDomain and EventDomain. CORBAObjectMonitor provides the facility to know if a domain is lost and contact is reestablished. When the CORBAObjectMonitor, located in utilsrvXX.jar, is instantiated, it binds to the domain through the application user.

Monitoring the CORBA interface status involves the following components:

- CORBAObjectMonitor to poll the domain.
- CORBAObjectMonitorListener to react to a contact change that the CORBAObjectMonitor detected.

CORBAObjectMonitor adds the listener to the CORBAObjectMonitorListener methods to get invoked. CORBAObjectMonitorListener has the following methods:

- connectionError
- connectionEstablished
- connectionLost
- successfullyPolled

This example requires the following objects and methods:

- Package com.aprisma.spectrum.core.idl
  - Interfaces
    - CsCModelDomain
  - Classes
    - CsCModelWatchCBImplBase
    - CsCModelDomainHelper
- Package com.aprisma.spectrum.core.idl.CsCError
  - Classes
    - CsCError_e
- Package com.aprisma.util.corba
  - Interfaces
    - CORBAObjectMonitorListener
  - Classes
    - CORBAObjectMonitor

How CORBA Interface Status Is Monitored

Monitoring the CORBA interface involves the following process:

1. The PersistentWatcher class implements the CORBAObjectMonitorListener.

2. The constructor makes a call to get the required domain (in this case, CsCModelDomain) and sets that reference to a variable. The CORBAObjectMonitor is created. The arguments passed are the service name, the service type, and the polling interval in milliseconds.
3. A call is made to connect on the monitor that obtains the object implementation. If the connection is already established, the object is returned. If not, the connection is attempted again if tryToConnect is set to True.
4. If the connection is good, add the listener and start polling.
5. If you invoke startPolling(), a check is made against the CORBA object to see if it still exists. If it exists, the successfullyPolled method is called. If not, the connection is retried.
6. If the connection is successful, connectionEstablished method is called. If the connection attempt fails, the connectionLost or connectionError is called depending on the problem.

   NOTE
   For fault-tolerant environments, if the primary fails, the CORBAObjectMonitor attempts to monitor the secondary. If successful, the connectionEstablished method is called. If the monitor object cannot connect to the secondary, connectionLost or connectionError is called.

   The following CORBAObjectMonitorListener methods require notification:
   - **successfullyPolled**
     Indicates the listener method called when connection is successfully polled.
   - **connectionEstablished**
     Indicates the listener method called when connection is successful.
   - **connectionLost**
     Indicates the listener method called when connection is not successful and the problem lies in the CORBA framework.
   - **connectionError**
     Indicates the listener method called when connection is not successful and the problem lies outside the CORBA framework.
7. The final call in the constructor sets up the model watch. For each trigger on the listener, an action is performed. For the connection lost and connection error, a message prints informing the user that connection is down. For the connection established, the model watch is reestablished.

Monitor CORBA Interface Status Example

The following example shows how to monitor an interface:

```
import java.io.*;
public class PersistentWatcherApp
{
    public static void main( String[] args )
    {
        String serverName = new String( "mySpectroSERVER" );
        String userName = new String( "user" );
        Integer mtID = Integer.decode( args[0] );
        PersistentWatcher pw = new PersistentWatcher( serverName,
            userName, mtID.intValue() );
    }
}
import java.util.*;
import com.aprisma.spectrum.core.idl.CsCError.*;
import com.aprisma.spectrum.core.idl.*;
import com.aprisma.spectrum.core.util.*;
import com.aprisma.spectrum.core.idl.CsCException.*;
import com.aprisma.util.corba.*;
```

```
/**
 *  This class implements the CORBAObjectMonitorListener but is also
 * responsible for connecting to the server and establishing the
 * watching of the Model Domain.
 */
public class PersistentWatcher implements CORBAObjectMonitorListener
{
    int mt_id;
    // Objects needed throughout the program
    CsCModelDomain modelDomain = null;
    CORBAObjectMonitor monitor = null;

    /**
        Constructs the PersistentWatcher
     */
    public PersistentWatcher( String servername, String username,
                              int modelTypeid)

    {
        this.mt_id = modelTypeid;
        getServerModelDomain( servername, username );

        // Setup CORBA Monitor
        monitor = new CORBAObjectMonitor ( servername,
                                           CsCModelDomain.class,
                                           20000); // 20 seconds
        // Check to see if the monitor has a connection if start the
        // setup the model watch and start to monitor
        if ( monitor.connect ( true ) != null )
        {
            monitor.addListener( this );
            monitor.startPolling();
            setupModelWatch();
            System.out.println("Press Enter to exit");
            try
            {
                System.in.read();
            }
            catch ( Exception e )
            {
                System.out.println( e );
            }
        }
        else
        {
            System.out.println("Monitor could not connect." +
                               "Not Watching");
        }
    }

    /**
        This method is responsible obtaining the SERVER's model domain.
```

```
 */
private void getServerModelDomain( String server, String username )
{
    try
    {
        CORBAHelper helper = CORBAHelper.getHelperImpl() ;
        helper.init(null,null);
        modelDomain =
          (CsCModelDomain) helper.getObjectImplementation (
               CsCModelDomain.class, server);
    }
    catch ( Throwable e )
    {
        System.out.println ( "Could not find server");
        System.out.println ( e ) ;
    }
}
/**
    This method is responsible creating the model watch on
    the model domain.
 */
 private void setupModelWatch()
 {
    try
    {
        int[] mtypeIDs = { mt_id };
        CORBAHelper helper = CORBAHelper.getHelperImpl();
        ModelCallback callback = new ModelCallback();
        CsCModelWatchCB cb =
        CsCModelWatchCBHelper.narrow(
            helper.servant_to_reference(callback));
        modelDomain.startWatchModelsByTypeIDs(mtypeIDs, cb);
    }
    catch ( Exception e )
    {
        System.out.println ( "Could not find server");
        System.out.println ( e ) ;
    }
 }
/**
    Interface CORBAObjectMonitorListener
    This method is Invoked when the connection has been lost.

 */
 public void connectionLost ( String objectName,
                              Class idlInterfaceClass )
 {
     System.out.println ("Watch and Connection Lost");
 }
/**
    Interface CORBAObjectMonitorListener
    This method is Invoked when a connection fails for some reason
    other than object
```

```
 */
 public void connectionError ( String objectName, Class
        idlInterfaceClass, Throwable e )
 {
      System.out.println ("Error");
 }

/**
    Interface CORBAObjectMonitorListener
    This method is Invoked when the connection has been
    established.
 */
 public boolean connectionEstablished (
                                  org.omg.CORBA.Object object,
                                    String objectName,
                                    Class idlInterfaceClass )
 {
     setupModelWatch();
     System.out.println ( "Re-established Watch and Connection");
     return true;
 }
/**
    Interface CORBAObjectMonitorListener
    This method is Invoked when the object is polled successfully.
 */
 public void successfullyPolled( org.omg.CORBA.Object object,
                                    String objectName,
                                    Class idlInterfaceClass )
 {
      System.out.println ( "Successful Poll" );
 }
```

Fault-Tolerant Environments

Do not write or update secondary servers in a fault-tolerant environment. You can pull data from the servers to know the client application status. We recommend that all data be updated from primary to secondary through online backup.

When switching from primary to secondary, the Naming Service should receive new advertisements from the secondary seamlessly. To check whether you are on the primary or secondary, call the isPrimary method from CSCModelDomain. You can also determine a switch through the CORBA Object Monitor.

In a fault-tolerant environment, the best way to get a service object from SpectroSERVER is using following API from CORBAHelper class:

```
/**
 * Obtains the implementation for the given object name.
 *
 * @param IDLClass the class of the object as defined in the IDL.
 * @param objectName the name of the object's implementation
 * @return the implementation object or an exception on failure.
 */
public org.omg.CORBA.Object getObjectImplementation(
                      Class IDLClass, String objectName )
                                    throws Throwable
```

The method gets the object from the correct server (primary or secondary). Internally, it gets the domain hierarchy map from the Location Server first, then try to get the object from domain hierarchy hosts, and return the object found.

You can also use following methods to get service object but need to specify the primary/secondary host name of the SpectroSERVER:

```
/**
 * Obtains the implementation for the given object name.
 *
 * @param IDLClass the class of the object as defined in the IDL.
 * @param objectName the name of the object's implementation
 * @param host the host to connect to.
 * @param enableReconnect determines whether to automatically
 *                        re-connect if the current connection is broken.
 *                        The implementation may not support this
 *                        option.
 * @return the implementation object or an exception on failure.
 */
public org.omg.CORBA.Object getObjectImplementation(
                      Class IDLClass, String objectName, String host,
                      boolean enableReconnect )
                                throws Throwable ;
```

For example, to get service object from secondary, the host should be the host name of the secondary.

There is another method that you can specify host names of both primary and secondary, and it will return the object from the correct server:

```
public org.omg.CORBA.Object getObjectImplementation(
                      Class IDLClass, String objectName, String[] hostList,
                      boolean enableReconnect )
                                throws Throwable ;
```

Here is an example to get domain ID without worrying about if the service is from primary or secondary:

```
try
{
     String domainName = "myLandscape";

    // Construct helper
    CORBAHelper helper = CORBAHelper.getHelperImpl();
    if ( helper.init(null, null) )
    {
        CsCModelDomain md = (CsCModelDomain)
        helper.getObjectImplementation(
                   CsCModelDomain.class, domainName );

        int id = md.getModelDomainID();
        System.out.println( "0x" + Integer.toHexString(id) );
    }
}
catch(Throwable e)
{
    e.printStackTrace();
}
```

Performance Issues

You can improve application performance by updating old method calls to the new method calls. All examples in this document use the new calls. Old method calls use large data objects that increase the interface load when they pass through IP packets. The new method calls minimize the data size by using the basic types and improve the speed.

The two main factors in API performance are speed and memory usage. To improve the speed, consider the following:

- How many models are involved?
- What calls are you making against the models? Are the calls identical?
- If you are reading or writing data, how many attributes are involved?
- Are you repeating any methods? If so, can you replace them with a single call using a list of model IDs?
- Is it necessary to read or write all the data?
- Can you group all the attribute reads and writes?
- If you are searching, is the search too broad?

If you have a list of model IDs that need to have an attribute written to, pass the complete list instead of iterating through each ID with a separate call.

To improve the memory usage, do not group too much work together. A large model group request can result in a large memory increase on the SpectroSERVER side. For example, if you are reading a route table, do not read the entire table at one time. Break up the read requests by using the CsCOIDSpec class and set a range. You cannot pass more than 500 models at a time.

> **NOTE**
> If you continue to experience performance issues after following these recommendations, contact CA Support.

Client Application Debugging

Use the following code to enable detailed debugging of a client application:

```
Properties props = new Properties () ;
props.put("ORBwarn", "2");
CORBAHelper helper = CORBAHelper.getHelperImpl();
helper.init(null, props);
CsCModelDomain md = (CsCModelDomain)
helper.getObjectImplementation
(CsCModelDomain.class, "myWorkstation" );
```

Extension Integration (SEI) Developer Reference

About the DX NetOps Spectrum Extension Integration Developer Toolkit (SEI Toolkit)

The DX NetOps Spectrum Extension Integration Developer Toolkit (SEI Toolkit) lets you organize and package DX NetOps Spectrum extension modules for installation and integration into other DX NetOps Spectrum hosts. An *extension module* is any set of software that extends the capabilities of the DX NetOps Spectrum product. There are three basic types of extension modules:

- **Core Components**
  Provide standard DX NetOps Spectrum functionality. CA is the exclusive developer of core components.
- **Management Modules**
  Extend DX NetOps Spectrum to manage various network devices. Although CA provides management modules for a large number of common network devices, it cannot possibly provide management modules for all of the new network devices that appear on the market almost daily. For this reason, DX NetOps Spectrum value-added resellers

(VARs) and device hardware manufacturers often develop and sell management modules that provide DX NetOps Spectrum management for network devices not included in CA's standard management module suite.

- **External Applications**
  Work in conjunction with DX NetOps Spectrum to enhance its usability. Though it is possible for you to develop external DX NetOps Spectrum applications, it is rarely done.

This section focuses primarily on the applications of the SEI Toolkit in the development of management modules. The SEI Toolkit is a Level 1 toolkit and a part of a suite of developer toolkits provided by CA to facilitate the development and distribution of DX NetOps Spectrum management modules. In the context of management module development, you can use the SEI Toolkit to do the following:

- Test the packaging and installation of newly developed management modules.
- Propagate management modules from one DX NetOps Spectrum host to others.
- Package management modules for installation on outside DX NetOps Spectrum hosts. For example, you develop management modules for new network devices, and then you use the SEI toolkit to package these management modules for sale and distribution to your customers. The customers can then install the packaged management modules onto existing DX NetOps Spectrum hosts to allow DX NetOps Spectrum to manage the new network devices.

  NOTE
  CA does not certify or support management modules unless they are distributed using the SEI Toolkit.

SEI Toolkit Goals

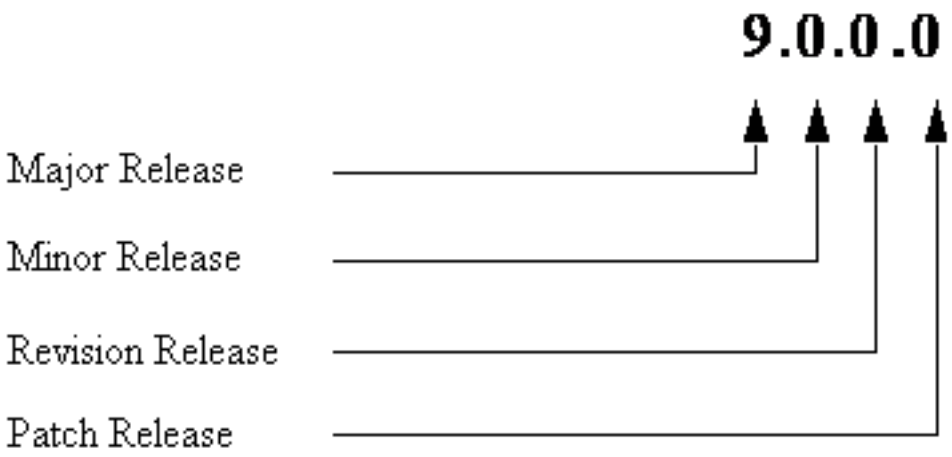
The primary goal of the SEI Toolkit is to provide you with the tools to distribute management modules for integration with DX NetOps Spectrum products. The SEI Toolkit tools and interfaces are standardized to ensure smooth integration of management modules from several different developers onto a single DX NetOps Spectrum host. By providing customers with a standard mechanism for installing and integrating DX NetOps Spectrum management modules, the SEI Toolkit accomplishes the following:

- **Reduces time to market:** Less time is needed to develop and test management module software and test its compatibility with software from CA and other developers.
- **Facilitates integration of components:** Since all CA development partners use the same tools to produce DX NetOps Spectrum management modules, these management modules are less likely to cause installation or integration compatibility problems for customers when used individually or together.
- **Reduces the investment required for developing DX NetOps Spectrum extension modules:** This follows not only from the reduction in time-to-market and a decrease in compatibility issues, but also because the software and expertise used to produce an extension module can be applied to multiple products, across multiple platforms and multiple releases.
- **Increases the value of DX NetOps Spectrum extension modules to customers:** Customers benefit from interoperable products from different developers. Whether they buy extension modules off the shelf or develop their own, customers know that these extension modules will work together.

In general, all of the DX NetOps Spectrum developer toolkits are designed to place minimal constraints on development environments. They are intended to integrate into existing environments, thereby providing maximum advantage in the development of DX NetOps Spectrum-compatible products.

Version Control

The DX NetOps Spectrum numbering scheme for version control consists of four numeric fields:



- **Major Release:** Increments to the number in this field represent major changes in the product's design, functionality, or user interface.
- **Minor Release:** Minor releases are scheduled product upgrades that provide new or enhanced features but do not represent functionality changes as significant as those involved in a major release.
- **Revision Release:** This field is incremented for subsequent revisions to a release and is reset to zero for each new major or minor release.
- **Patch Release:** This field -- previously used for internal tracking of individual builds under a particular revision and not usually visible to the customer -- assumes special importance to the extension developer. The value of this field must be zero for the first release of a management module or application. In addition, it must be incremented to some higher value (not necessarily sequentially) for each successive release of the same module or application.

SEI Toolkit Versioning

The versioning of the SEI Toolkit is tied to the versioning of the DX NetOps Spectrum enterprise management product. Consequently, management modules and applications produced using the SEI Toolkit must follow a similar convention to that described in the preceding section. The actual numbers used can and will be different, but the numbering format must use the same approach in order for the products to be integrated with DX NetOps Spectrum.

SEI Toolkit Architecture

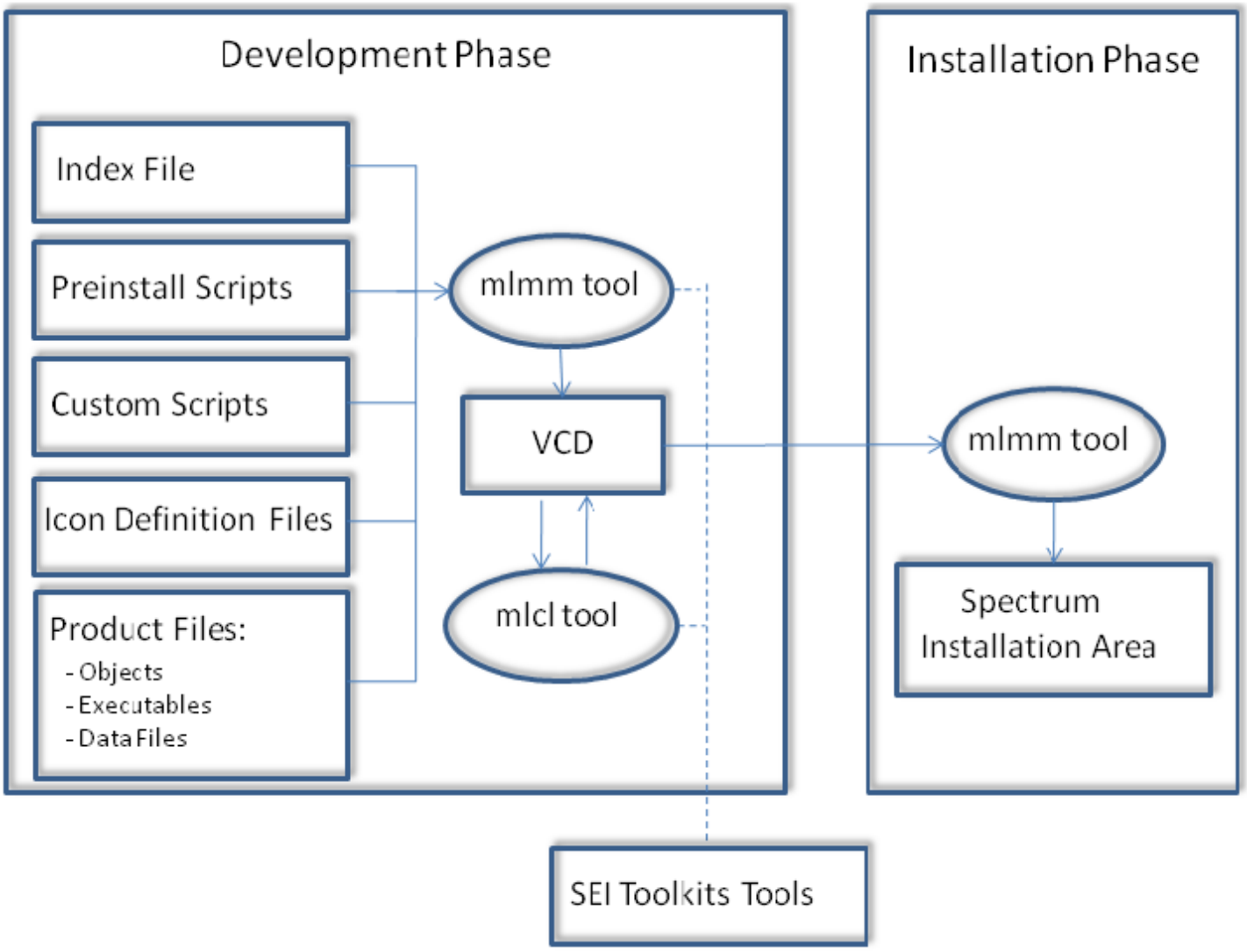
About Extension Modules

The overall process of creating a new DX NetOps Spectrum extension module involves three phases:

- **Development/Customization:** Development and testing of a new extension module is done using the Model Type Editor. The mmship utility can be used to create the index file necessary for the manufacturing process.
- **Manufacturing:** The SEI Toolkit tools mkmm and mkcd are used to manufacture the final product of the development phase.
- **Installation:** Installation and integration of the finished extension module into the DX NetOps Spectrum product is accomplished by the Install program, residing on the DX NetOps Spectrum host.

The tools covered in this document are mmship, mkmm and mkcd, which assemble the applicable files for a given module or application into a VCD.

The following diagram provides an overview of this manufacturing process and the follow-on activity of actually installing the VCD contents onto a host system.



Overview of the Development Phase

The development phase involves the following general steps:

1. Design and develop the files that are to be shipped and installed as part of the extension module. CA provides a Level 1 Toolkit and a CORBA Toolkit to aid in developing new extension modules.

   NOTE
   For more information about management module development, see the Certifications section. For more information about general DX NetOps Spectrum functionality and integration points, see the DX NetOps Spectrum Modeling Concepts section and the SpectroSERVER and DX NetOps Spectrum Databases Overview section.
2. Prepare the module for integration by creating an index file and auxiliary files that describe the module. The index file includes descriptive information about the extension module and lists the files to be shipped and installed as part of the module. For Level 1 management modules, the mmship utility does much of the work of creating an index file.

Additional auxiliary files include part description files, preinstall scripts, and custom scripts, which may be used to customize the installation of the extension module. They also include icon definition files, which simplify creation of icons during installation.

3. Edit the setup initialization files to insert your product name into the installation instructions.
4. Use the mkmm tool and the index files to create a virtual CD-ROM (VCD) or to add the extension module to the VCD. A single invocation of mkmm may be used to add several extension modules to the VCD.
5. Use the mkcd tool to add version identification to the VCD, making it possible to install the extension module software from the VCD into an existing DX NetOps Spectrum installation area.
   The VCD can then be installed from a disk, downloaded over the Internet and installed, or written to a physical CD-ROM and installed from that medium.

Installation Phase

Integration of an extension module on a VCD into an existing DX NetOps Spectrum installation host is accomplished by installing the VCD on that host. This is done by executing one of the following installation tools placed on the VCD when it is created:

- Windows setupnt.exe
- Linux installer

Both of these tools can be invoked either by a command line or using icons.

Beyond these tools, the VCD contains very little of its own installation support software. Instead, the installation process reuses the installation support software that already exists in the host system's DX NetOps Spectrum installation area, placed there when DX NetOps Spectrum was originally installed.

Use Cases for the SEI Toolkit

Common Customizations

You can use the SEI Toolkit in the context of developing the following:

- A DX NetOps Spectrum management module developed using the Level 1 Toolkit and the CORBA Toolkit. This is the most common scenario where VARs use the SEI Toolkit.
- A customized OneClick application.
- A DX NetOps Spectrum external application.

The following is the high-level process to follow:

1. Do one or more of the following depending on your requirements:
   – **Create the management module:** Use the Model Type Editor to create a model type that represents the new device in the SpectroSERVER database. You can also use the Model Type Editor to create a database export file, which you can then use to import the new model type into other DX NetOps Spectrum databases.

     NOTE
     For information about using the Model Type Editor, see the Model Type Editor section.

     Customize the various files in your new management module until the module represents and manages the new network device to your satisfaction. Two common customizations are creating application model types for the device to support the functionality of a MIB and adding support for alerts, events, and alarms.
   – **Create the customized OneClick application:** Using mmship and applicable toolkits, customize OneClick by creating custom views, searches, tables, and menus.

NOTE
For more information about customizing OneClick, see the OneClick Customization section.

– **Create the external application:** Using mmship and applicable toolkits, develop the shippable files for the external application.

NOTE
For more information about developing shippable files, see the Certifications section.

2. Once your new management module, customized OneClick application, or external application performs as desired, create the following auxiliary files needed for the distribution process:
   – An index file
   – A part description file

   While you can create these files entirely by hand using a text editor, doing so using the mmship tool provided with the SEI Toolkit simplifies the process. If you use mmship to generate the files, you should use a text editor to check the generated files for accuracy and completeness, and to make any necessary changes or customizations.
   If you are creating a new management module, be aware that mmship does not include file: entries in the index file for any object files you may have added using the CORBA Toolkit. If you are creating a customized OneClick application, be aware that mmship does not include file: entries in the index file for any XML files you may have added to define your OneClick customizations. In either case, you must use a text editor to add the necessary file: entries to the index file.
   If you are creating a new management module or customized OneClick application, specify an index file level of 1 in the index file. If you are creating an external application, specify an index file level of 2.
3. Edit the setup initialization files to insert your product name into the installation instructions.
4. Use the mkmm tool to add the management module, customized OneClick application, or external application to a VCD.
   If the VCD does not exist, mkmm will create it. If the VCD already exists, mkmm lets you add any number of additional modules or applications to it. Several for Linux and/or Windows can be added to a single VCD.
5. Use the mkcd tool to do the following:
   – Check the integrity of the VCD.
   – Add a version number to the VCD so that it can be installed.
6. Perform final testing of the management module, customized OneClick application, or external application by installing it into an existing DX NetOps Spectrum installation area from the finished VCD. If testing fails, modify the source files as necessary and then rebuild the VCD.
7. Distribute the VCD to customers or local developers. The VCD can be:
   – Installed directly from disk.
   – Installed from a mounted file system.
   – Downloaded from the Web and installed.
   – Committed to physical CD-ROM and installed from that medium

   CA does not provide hardware or software for distributing VCDs using the Web or physical CD-ROM, but these features are available from various third parties.

Types of VCDs

Shippable files fall into two categories:

- **Platform-specific shippable files.** A file is platform-specific if it cannot be installed and used successfully on all supported platforms (Windows and Linux). An example of a platform-specific file is an object file. Since Windows

object files cannot be linked on Linux or vice versa, separate versions of the same object file must be shipped for each platform.

- **Platform-neutral shippable files.** A file is platform-neutral if it can be installed successfully on all supported platforms. An example of a platform-neutral shippable file is a text file. Since text files have the same format on Windows and Linux, the same text file can be shipped once and installed and used successfully on any platform.

VCDs fall into the following categories:

- **Multi-platform VCD.** A multi-platform VCD contains platform-specific shippable files for one or more platforms. As an example, a multi-platform VCD can contain platform-specific shippable files for Windows and Linux. Platform-specific shippable files are kept in the following folders on the VCD:
  - (Windows) nt folder
  - (Linux) linux folder

  When the VCD is installed on a specific platform, files from the appropriate platform-specific folder are installed.
- **Single-platform VCD.** A single-platform VCD is just a multi-platform VCD containing platform-specific shippable files for only one platform. For example, a single-platform VCD for Windows is a multi-platform VCD having an nt area but no `sunos5` or `linux` areas. When the VCD is installed on Windows, therefore, files from the nt area are installed, but the VCD cannot be installed on Linux.
- **Platform-neutral VCD.** A platform-neutral VCD is a VCD that contains only platform-neutral shippable files. These platform-neutral files are kept in any area of the VCD. When installed on any platform (Windows and Linux), files from any area are installed.

The sections that follow describe the scenarios that represent recommended methods for creating multi-platform, single-platform, and platform-neutral VCDs. The scenarios assume the following:

- The necessary shippable files and index files have been created for each platform type that you intend to include on the VCD.
- The shippable files and index files have been made available on the machines on which you intend to create the VCD. This is typically done by copying the shippable files or creating a remote file system.
- The SEI Toolkit has been installed and made available on the machine on which you intend to create the VCD.

Creating a Single-Platform VCD

In this scenario, assume that you have created the shippable files and an index file within a DX NetOps Spectrum directory on a Windows host, and you want to create a single-platform VCD by running mkmm on that host.

1. Change to the directory containing the index file.
2. Run the following command as a single-line entry:

```
<$SPECROOT>/INSDK/mkmm <indexFile> vcd=<vcdDirectory>
```

Because mkmm is running on a Windows host in this case, the program responds to this environment by adding Windows shippable files to the VCD. This results in the creation of a single-platform Windows VCD, and the VCD will not be installable on a Linux platform.

NOTE
mkcd must be invoked on the VCD to make it installable. This is done by running the following command on any platform:

```
<$SPECROOT>/INSDK/mkcd -f <vcdDirectory> <vcdVersion>
```

Example: Create a Multi-Platform VCD (Method 1)

In the following scenario, assume that you have created the shippable files and an index file within a DX NetOps Spectrum directory on a Windows host and on a Linux host, and you want to create a multi-platform VCD that will install on any of these platforms. To accomplish this, you must run mkmm once on each platform-specific host.

On the Windows host, create a single-platform Windows VCD by doing the following:

1. Navigate to the directory containing the Windows index file.
2. Run the following command as a single-line entry:
   ```
   <$SPECROOT>/INSDK/mkmm <WindowsIndexFile> vcd=<vcdDirectory>
   ```

On the Linux host, copy or mount the multi-platform VCD.

1. Still on the Linux host, add the Linux shippable files to the VCD:
   1. Change to the directory containing the Linux index file.
   2. Run the following command as a single-line entry:
      ```
      <$SPECROOT>/INSDK/mkmm <LinuxIndexFile> vcd=<vcdDirectory>
      ```

   The resulting VCD is now capable of being installed on any of the platforms.

   NOTE
   mkcd must be invoked on the VCD to make it installable. This is done by running the following command on any platform:

```
<$SPECROOT>/INSDK/mkcd -f <vcdDirectory> <vcdVersion>
```

Example: Create a Multi-Platform VCD (Method 2)

In the following scenario, assume that you have created the shippable files and an index file within a DX NetOps Spectrum directory on a Windows host and on a Linux host. You want to make a multi-platform VCD that will install on any of these platforms, and you want to use a single host -- in this scenario, the Windows host -- for the production of the multi-platform VCD.

1. On the Windows host, create a single-platform Windows VCD by doing the following:
   a. Navigate to the directory containing the Windows index file.
   b. Run the following command as a single-line entry:
      ```
      <$SPECROOT>/INSDK/mkmm <WindowsIndexFile> vcd=<vcdDirectory>
      ```

      NOTE
      Verify that any directories specified in the index file are accessible by the Windows host.
2. While still on the Windows host, mount the Linux DX NetOps Spectrum area.
3. Add the Linux shippable files to the VCD:
   a. Change to the directory containing the Linux index file.
   b. Run the following command as a single-line entry:
      ```
      <$SPECROOT>/INSDK/mkmm plat=linux <LinuxIndexFile> vcd=<vcdDirectory>
      ```

   The resulting VCD is a multi-platform VCD that is installable on any of the platforms.

   NOTE
   mkcd must be invoked on the VCD to make it installable. This is done by running the following command on any platform:

```
<$SPECROOT>/INSDK/mkcd -f <vcdDirectory> <vcdVersion>
```

Example: Platform-Neutral VCD

In this scenario, assume you have created shippable files and an index file within a DX NetOps Spectrum directory on a Windows host. The shippable files are all platform-neutral, so they can be safely installed on Windows or Linux. Similarly, the index file contains no platform-specific entries (plat=<platform> modifiers), so the same set of files can be shipped for any platform. You want to run mkmm on a single host to create a platform-neutral VCD.

Assume that you are creating the VCD on a Windows host.

1. On the Windows host, navigate to the directory containing the Windows index file.
2. Run the following command as a single-line entry:

   ```
   <$SPECROOT>/INSDK/mkmm any=yes <WindowsIndexFile> vcd=<vcdDirectory>
   ```

   The any=yes flag causes mkmm to create a platform-neutral VCD. Bear in mind that mkmm will fail if you subsequently attempt to add platform-specific shippable files to this VCD.

   NOTE
   mkcd must be invoked on the VCD to make it installable. This is done by running the following command on any platform:

```
<$SPECROOT>/INSDK/mkcd -f <vcdDirectory> <vcdVersion>
```

Creating Index Files

About Index Files

After using the Level 1 Toolkit or the CORBA Toolkit to create a new extension module, a developer will typically proceed to write an index file for the extension module. The index file describes the extension module, describes its relationships to other extension modules, and lists the files that are installed as part of the extension module.

The purpose of writing the index file is to allow the extension module to be added to a VCD. Once on the VCD, the extension module can then be installed from the VCD and integrated into any compatible DX NetOps Spectrum installation area. You use the mkmm and mkcd tools to create the VCD, to add the extension module to the VCD, and to prepare the VCD for installation, but you must create the index file before you can use these tools.

If your extension module is a management module developed using the Level 1 Toolkit or the CORBA Toolkit, you can run the mmship tool to do much of the work of creating the index file for the extension module.

Create an Index File Using mmship

You can use the mmship tool to create index files and auxiliary files for newly developed management modules. mmship uses the following syntax:

```
mmship -s <scriptfilename>
```

- ***<scriptfilename>***
  Is the name of the script file that mmship generates. This script file records your progress using mmship. If you need to stop the tool for any reason, you can use this file to pick up where you left off.

To create an index file using the mmship tool

1. Navigate to the <*$SPECROOT*>/SS-Tools directory.
2. Enter the mmship command using the syntax described at the beginning of this section.
3. You are asked a series of questions concerning the management module. Answer each question and press Enter to continue. A basic overview of each question:
   a. *Please enter the management module version number:*
      Enter a valid version number and press Enter.
   b. *Please enter your vendor name as assigned by CA.*
      Enter your vendor name, and press Enter.
   c. *Please enter your 4-character vendor ID as assigned by CA.*
      Enter your developer ID, and press Enter. The alpha characters in your developer ID is entered in capitals.

NOTE
For more information on developer IDs, see the SpectroSERVER and DX NetOps Spectrum Databases Overview section.

d. *You are now creating a new part <partnumber>. Please enter a short paragraph which will identify the function of part <partnumber> to the user during installation. End the description with a blank line.*
Enter your description. Be sure that the last line in your description is a blank line. Press Enter twice.

e. *You are now adding the representative MM to part <partnumber>. This MM contains the descriptive information for part <partnumber>. What model type are you shipping in the representative MM?*
Enter the model type name, and press Enter.

f. *Does this Management Module include database import files?*
If you have used the Model Type Editor to export files that are used with this model type, enter Yes; otherwise, enter No. Press Enter. For more information about exporting model types to a catalog file (a .e file that can also be imported), see the Model Type Editor section.

g. *Please enter the name of a database import file relative to the DX NetOps Spectrum installation directory.*
Enter the name and path of the .e file that you have exported using the Model Type Editor.

h. *Do you want to include more database import files.*
Enter Yes or No, and press Enter.

i. *Please enter the Base Mode type for <modeltypename>.*
Enter the appropriate base model type name, and press Enter.

j. *Do you want to add another MM to part <partnumber>?*
Answer Yes if you would like to include more model types, or enter No if you are finished.

k. *Do you want to add another part number?*
Enter Yes if you want to create another installable part number, or enter No if you are finished.

mmship has now created the appropriate index files and part description files. They can be found in the *<$SPECROOT>*/SS-Tools directory.

NOTE
If mmship is not applicable to your situation, you can use a text editor to construct an index file for your extension module.

General Index File Syntax

The following general syntax rules apply to DX NetOps Spectrum index files:

- Each index file entry consists of zero or more fields of visible characters, which may be group-separated by one or more spaces or tabs. Spaces and tabs may also appear at the beginning or end of an index file entry.
- Index file entries are newline-delimited (that is, one entry per line, terminated either by a newline character or end of file). There is no mechanism for continuing an index file entry across more than one line. There is no preset limit on the length of an index file entry.
- The following characters are legal in an index file:

Character	ASCII Code
Visible character	33 (!) through 126 (~)
Space	32
Tab	9
Newline (Linefeed)	10
Carriage Return	13 (^M)
Formfeed	12 (^L)

- Each line of an index file consists of an index file entry and can conclude with an optional comment. If a comment is included, the comment begins at the first pound sign (#) in the line and extends to the next newline or end of file. If an index file entry has no fields, the entire line is considered a comment line. Otherwise, the first field of an index file entry must be a valid index file label, as identified in Index File Entry Definitions. Any other index file entry is illegal and will cause mkmm to fail while processing the index file.

Environment Variables

Environment variables of the form $VAR, ${VAR}, or $(VAR) may appear in index files, where VAR is a standard shell variable name (one or more alphanumerics and underscores, not beginning with a digit).

When mkmm processes an index file, embedded environment variables are expanded to their values, as exported in the invoking shell. Therefore, it is necessary to set and export any environment variables used in an index file prior to invoking mkmm to process that index file. For example, if the environment variable $VNUM is set to the value “1.1” and exported before mkmm is invoked, the index file line:

```
mm: Acme Router Version $VNUM
```

is expanded to:

```
mm: Acme Router Version 1.1
```

Environment variables having special meaning to a particular shell (for example, $_, $PATH, $CWD, $HOME, and so on) are not guaranteed to expand to expected values. Environment variables not of the form $VAR, ${VAR}, or $(VAR), (for example, $$, $!, $?, $0, $1, and so on) are not expanded. Avoid using these sorts of environment variables in index files.

Escaping Special Characters

The pound sign (#) and the dollar sign ($) have special meaning within an index file. These characters must be escaped to take on their literal value within an index file entry.

Pound Sign (#)

The occurrence of a pound sign (#) in an index file always denotes the beginning of a comment. The pound sign and any subsequent characters on the line are ignored when mkmm processes the index file. To prevent interpretation of the pound sign as the beginning of a comment, replace the pound sign with the environment variable ${PS}, which mkmm will expand to a literal pound sign.

For example, the index file line:

```
mm: Acme Router #2 Management Module
```

is interpreted by mkmm as the entry “mm: Acme Router” followed by the comment “#2 Management Module”, whereas the line:

```
mm: Acme Router ${PS}2 Management Module
```

is expanded to:

```
mm: Acme Router #2 Management Module
```

Dollar Sign ($)

The dollar sign in the context of $VAR, ${VAR}, or $(VAR) denotes the beginning of an environment variable. When mkmm processes the index file, environment variables are expanded, and the delimiting dollar sign is lost. To prevent interpretation of a dollar sign as part of an environment variable, replace the sign with the environment variable ${DS}, which mkmm expands to a literal dollar sign.

If the environment variable $VNUM is set to the value “1.1” and exported before mkmm is invoked, the index file line:

```
mm: Acme Router Version $VNUM
```

expands to:

```
mm: Acme Router Version 1.1
```

whereas the entry:

```
mm: Acme Router Version ${DS}VNUM
```

expands to:

```
mm: Acme Router Version $VNUM
```

The above example also illustrates the use of the ${VAR} or $(VAR) forms of an environment variable to separate environment variable names from text that immediately follows. For instance, in the index file line:

```
mm: Acme Router Version $VNUM
```

the environment variable $VNUM is expanded, while in the index file line:

```
mm: Acme Router Version ${V}NUM
```

the environment variable ${V} is expanded and NUM keeps its literal value.

Index File Platform Names

Distribution and installation of CA Spectrum 10.3 extension modules are supported on the following platforms (operating systems). Whenever referring to these platforms within index files, use the platform names shown in the following table:

DX NetOps Spectrum Platform Name	Release Level	Platform Name in Index Files	Platform Name in VCD Area
Windows	Microsoft Windows Server 2012 and 2016	intel	nt
Linux	Red Hat Enterprise Linux 6.x and 7.x	linux	linux

The index file platform names refer to the operating systems, not to any specific hardware architectures.

Index File Entry Definitions

Each entry in an index file must contain an index file label followed by an appropriate number of data fields. The label generally indicates the meaning and purpose of the entry, and each entry must conform to the syntax associated with its label.

The index file entries within each category are identified in the shaded block at the beginning of the discussion. For each entry, this discussion gives the entry syntax, a brief example, the entry description, and applicable conventions.

Extension Module Description Entries

Extension module description entries describe the extension module associated with the index file. Each extension module description entry should appear exactly once in each index file. If an extension module index file does not contain

an extension module description entry or contains more than one such entry, mkmm will generate an error and will not add the extension module to the VCD.

irev Entry

Syntax:

irev: <*version*>

Example:

irev: 09.00.00.000

Description:

A single irev: entry is required in every index file.

The irev: entry provides <*version*>, the machine-friendly version number of the extension module. The <*version*> identification is a replica of the DX NetOps Spectrum version control number except that the number value is translated to the format “dd.dd.dd.ddd” where each “d” is a decimal digit. For example, the 09.00.00.000 value shown in the example in this section is what mmship would produce if the user typed in 9.0.0.0 when mmship asked for the version number of the module being developed.

The main use of the irev: entry is to identify any instance of extension module version downgrading during installation. Extension module version downgrading occurs when an older version of an extension module (that is, one with a lower-valued patch release number as the final numerical value) is installed into an area that already contains a newer version of that same extension module. Normally, version downgrading is undesirable, and the DX NetOps Spectrum installation program will flag that situation, preventing you from installing the older version on top of the newer one.

Install detects a downgraded extension module by comparing the irev: version numbers of the previously installed extension module with the one being installed. If the previously installed extension module has a larger irev: version number, an older version of the extension module is being installed over a newer version, and a downgrade is reported for that extension module.

Another important consideration is that the value of the final number (the ddd field, referenced elsewhere as the Patch Release number) must be zero (000) for the initial version of any module. A Patch Release value of zero thus identifies the module as being the original installation. Each successive revision of the same module must have a higher number value, but the increase does not have to be an incremental value with each successive release.

level Entry

Syntax:

level: <*level*>

- **Example:**
  level: 1
- **Description:**
  The level: entry specifies the level of the extension module. The extension module level indicates how software in the extension module is related to DX NetOps Spectrum (SpectroSERVER and/or OneClick). The <level> designation must be one of the following three number values:

0

The software in the extension module is a required part of DX NetOps Spectrum, either for SpectroSERVER and/or OneClick. A level 0 extension module is often referred to as a DX NetOps Spectrum core module. The right to develop and ship level 0 extension modules is reserved to CA alone.

1
The software in the extension module is an optional part of DX NetOps Spectrum. A level 1 extension module is often referred to as a management module, typically containing software that helps DX NetOps Spectrum manage a particular device or class of devices. While CA develops and ships a wide variety of level 1 extension modules, many others are developed by DX NetOps Spectrum users for their own management needs, or by VARs for resale to other DX NetOps Spectrum users.

2
The software in the extension module is not part of DX NetOps Spectrum (although it may work in conjunction with DX NetOps Spectrum). A level 2 extension module is often referred to as an external application or toolkit, and it typically contains a standalone tool or demo for use in the DX NetOps Spectrum environment. Most level 2 extension modules are developed at CA.

Install uses the extension module level value to select extension modules for installation. In addition, the extension module level also affects the way in which files in an extension module are processed during installation. Since level 0 and level 1 extension modules are considered part of DX NetOps Spectrum (SpectroSERVER or OneClick), files shipped in level 0 and level 1 extension modules undergo special processing during installation to integrate them into DX NetOps Spectrum. In contrast, level 2 extension modules are not considered part of DX NetOps Spectrum, so files contained in level 2 extension modules are not integrated into DX NetOps Spectrum during installation.

Do not confuse extension module levels with toolkit levels. CA provides two toolkits for creating and customizing extension modules: the Level 1 Toolkit and the CORBA Toolkit. The Level 1 toolkit can be used by any DX NetOps Spectrum user, but the CORBA Toolkit requires advanced programming knowledge and familiarity with C++. The extension modules that are developed using either of these toolkits are always optional extension modules (level 1 management modules) regardless of whether they were developed using the Level 1 Toolkit or the CORBA Toolkit. Therefore, when shipping extension modules developed using either of these toolkits, always use level: 1 in the index file.

mm Entry

Syntax:

mm: <*name*>

Example:

mm: Adco Controller

Description:

The mm: entry specifies a descriptive name for the extension module.

- ***<name>***
  May consist of a single word or several words but should be kept to a maximum of 30 characters, including spaces, in order to be visible on the screen when displayed as part of a message. <*name*> should be descriptive -- that is, “UPS_Monitor” is preferable to “M4423R-XXX-01.”
  Install displays the <*name*> identification on the installation summary.

rev Entry

Syntax:

rev: <*version*>

Example:

rev: 9.0.2.0

Description:

This value must comply with the following rules.

- <*version*> must be a single alphanumeric field with no spaces allowed (for example, 9.0.0.000).
- Maximum length of <version> is 15 characters.
- With the exception of the fact that mkmm requires this entry, rev: is not used in the distribution and installation utilities. Its presence in the index file serves to identify the extension module version for software maintenance purposes.

vend Entry

Syntax:

vend: <*vendor*>

Example:

vend: CA

Description:

The vend: entry specifies the CA-assigned developer name of the vendor creating the extension module.

- <*vendor*> must be a single-word text string.
- With the exception of the fact that mkmm requires this entry, vend: is not used in the distribution and installation utilities. Its presence in the index file serves to identify the extension module developer for software maintenance purposes.

Extension Module Relation Entries

Extension module relation entries describe relationships between the extension module associated with the index file and other extension modules.

pprep Entry

Syntax:

pprep: <*ppname*> <*mmname*> <*mmname*> …

Example:

pprep: SA-CSI1000

Description:

The pprep: entry specifies that the index file contains the representative extension module of a purchasable part named <*ppname*>, which will inherit its attributes (name, version numbers, and so on) from its representative extension module.

A *purchasable part* can be a single extension module or a group of extension modules that are purchased and installed together as a unit, with the unit having a single purchasable part name (<*ppname*>) for the entire package. If the purchasable part is a group of extension modules, <*ppname*> is the name of the primary module and the subsequent <*mmname*> fields list the additional extension modules that are included in the purchasable part. All names must be unique.

As an example, suppose the index file JOE.i contains the following entries:

```
mm:   JoeCorp Router Management Module
rev:   6.0rev3
pprep:  SM-103701 JIM JILL
```

The pprep: entry defines purchasable part SM-103701. Purchasable part SM-103701 contains the representative extension module JOE and the additional extension modules JIM and JILL, which are defined in index files JIM.i and JILL.i, respectively.

An extension module may represent at most one purchasable part. Therefore, a given index file can contain at most one pprep: entry. The mkmm tool will flag multiple pprep: entries in an index file.

A purchasable part must have a unique representative extension module. Therefore, each purchasable part must have a pprep: entry in exactly one index file. To enforce this, the mkcd tool will flag multiple pprep: entries for a purchasable part, establishing an error condition.

Each purchasable part must have a part description file, which is distributed via a file: Install mmdesc entry. Therefore, if an index file contains a pprep: entry, it must also contain a file: Install mmdesc entry.

Extension modules that do not include either a pprep: or a pp: entry in the index file for that module are known as orphan parts. You can work with these extension modules for testing purposes if you do not have all the applicable parts, but you must provide one or the other of these index file entries before distributing the final VCD for installation.

pp Entry

Syntax:

pp: <*ppname*> <*mmname*> <*mmname*> …

Example:

pp: SA-CSI1000

Description:

The pp: entry specifies that this index file contains an extension module which belongs to the purchasable part named <*ppname*>. The subsequent <*mmname*> fields (if any) list additional extension modules that are included in purchasable part <*ppname*>. The <*ppname*> and <*mmname*> values can be anything that the developer wishes except that they must be unique. They can include standard alphanumeric characters, underscore characters, and hyphen characters.

The pp: entry is similar to the pprep: entry except that it does not establish the current extension module as the representative of the purchasable part <*ppname*>. The pp: entry merely adds the current extension module and the extension modules given by <*mmname*> arguments to the purchasable part <*ppname*>. The representative extension module of the purchasable part is established by the pprep: entry in the current index file or in some other index file.

Extension modules that do not include either a pp: or a pprep: entry in the index file for that module are known as orphan parts. You can work with these extension modules for testing purposes if you do not have all the applicable parts, but you must provide one or the other of these entries before distributing the final VCD for installation.

comp Entry

Contents

Syntax:

comp: <*mmname*>

Example:

comp: CHASSIS-MIMS

Description:

The comp: entry lets developers identify distribution dependencies between two or more extension modules. A distribution dependency exists when the successful distribution and installation of an extension module requires distribution and

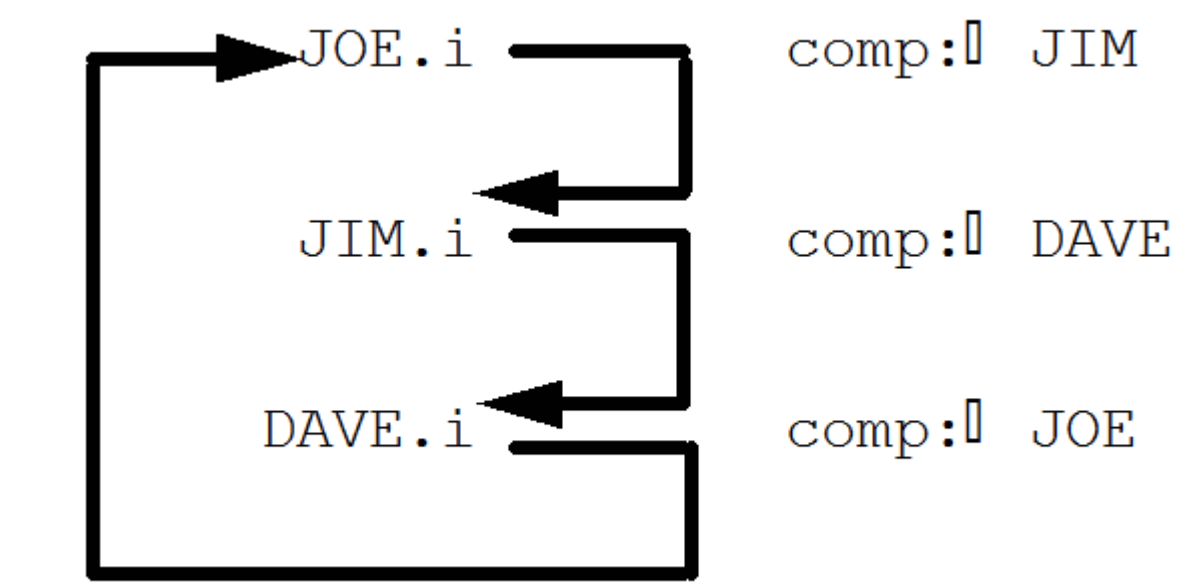
installation of some other extension module. For example, a comp: dependency is commonly used to ensure that if a management module for a derived model type is distributed on a VCD, then the management module for its base model type is also distributed on the same VCD. The dependency also ensures installation of the management module for the derived model type from the VCD will trigger installation of the management module for the base model type.

The comp: entry indicates that the extension module associated with the index file has a distribution dependency on the extension module identified by <*mmname*>. For example, a comp: JIM entry in the index file JOE.i indicates that extension module JOE has a distribution dependency on extension module JIM.

comp: allows only one dependency to be defined at a time. You can establish multiple dependencies by providing a sequence of comp: statements, each on its own line.

The mkcd tool enforces comp: dependencies. If extension module JOE has a comp: dependency on extension module JIM, and JOE is included on a VCD without JIM, by default mkcd will report the unresolved dependency and will not finish the VCD.

comp: dependencies may be circular. For example, assume the following lines in the index files for extension modules JOE, JIM, and DAVE:



These dependencies guarantee that JIM, DAVE, and JOE will be distributed and installed together.

comp: cannot be used to establish a dependency between extension modules of different levels. If this is done, mkcd will generate a warning message and ignore the dependency. (The distribution dependencies between extension modules of different levels is inferred from the levels, and explicit comp: dependencies are unnecessary.)

File Distribution Entries

File distribution entries are used to specify files to be included in an extension module, or to modify the way these extension module files are distributed or installed.

attr Entry

Syntax:

attr: <*file*> [mode=<*mode*>] [uid=root]

Example:

attr: SS/SpectroSERVER mode=4777 uid=root

Description:

The attr: entry lets you change the accessibility of a file, effective following completion of installation.

Normally, after installation, installed files have the mode with which they were extracted from the VCD or created in the installation area, and they are “owned” by the installation owner as specified using the DX NetOps Spectrum installation. However, there are times when it is desirable to override these default permissions or ownerships for specific files. The attr: entry can be used to accomplish this.

In the attr: entry, <*file*> is a file name specified relative to the installation root directory, and <mode> is a UNIX chmod-style mode modifier. At the end of the installation, the mode modifier is applied to the specified file, as if a “chmod <*mode*> <*file*>” UNIX command had been executed. If the “uid=root” flag is also present in the attr: entry, <*file*> is given root ownership.

> **NOTE**
> You can use file: and attr: entries to enforce mode and ownership of installed files.

cus_dep Entry

Syntax:

cus_dep: <*mmname*>

Example:

cus_dep: CHASSIS-MIMS

Description:

The cus_dep: entry lets developers identify custom script dependencies between two or more extension modules. A custom script dependency exists when the custom script of an extension module requires another custom script to be run first.

For example, a cus_dep: dependency is used by the DX NetOps Spectrum installation program to ensure that the mysql custom script runs before the Report Manager or Archive Manager custom scripts run. This sets up mysql before the other extension module.

deflt Entry

The following syntax layout folds the line in order to show the entire contents of the simulated text within the limits of the available page width. However, in an actual index file, your deflt: entry must be entered as a single line, and you cannot use a backslash to continue an entry across more than one line.

- **Syntax:**
  deflt: <*prod*> <*type*> <*defname*> <*value*> [plat=<*plat*>][prot=yes]
  [req=yes]
- **Example:**
  deflt: SS xtn sdir /usr/swd/bin plat=sun4c
- **Description:**
  The deflt: entry associates a default value with a product and file type. These default values then become expanded within subsequent file: or head: entries that have matching product and file types.
  deflt: entries are the only entries that may appear in the data files mkmm.sys and mkmm.loc. These deflt: entries establish the standard default values applied by mkmm to file: and head: entries in all index files.

In index files, deflt: entries are used to override the standard default values defined in mkmm.sys and mkmm.loc. The defaults set by a “deflt: <*prod*> <*type*> …” entry in an index file apply to all subsequent file: and head: entries with the same <*prod*> and <*type*> fields as the deflt: entry, until the end of the index file or until another deflt: entry with the same <*prod*> and <*type*> fields is encountered.
<*prod*> File Product
<*type*> File type
<*prod*> and <*type*> take on any combination of values legal for the <*prod*> and <*type*> fields of the file: entry.
The entry “deflt: <*prod*> <*type*>…” specifies a default source or target directory value to be applied to subsequent file: and head: entries having the same <*prod*> and <*type*> fields.

- ***<defname>***
  Specifies the name of the default being assigned and may take on one or the other of the following two values:
- **sdir**
  Set the <*sdir*> default source directory.
- **tdir**
  Set the <*tdir*> default target directory.
- ***<value>***
  If <defname> = sdir, <value> may be an absolute or relative path name. If <*defname*> = tdir, <*value*> must be a path name that is relative to the current directory when installation is invoked. You can set <*value*> to the “?” variable to recover the system default value.
- **plat=<*plat*>**
  This optional flag must occur after the <*value*> field. It specifies that the deflt: entry is to be processed only on platform <*plat*>. Multiple plat=<*name*> flags may be given to specify that the deflt: entry can be processed on those platforms.
- **plat!=<*plat*>**
  This optional field must occur after the <*value*> field. It specifies that the deflt: entry is not to be processed on platform <*plat*>. For example, to set the default only for Linux, you would specify plat!=intel. Multiple plat=<*name*> flags may be given to specify that the deflt: entry will not be processed on those platforms.

The plat=plat and plat!=plat flags are mutually exclusive; they cannot be used together in the same deflt: entry. If neither plat=<*plat*> nor plat!=<*plat*> is used, the deflt: entry is processed on all platforms.
The prot=yes and req=yes options shown in the syntax example, both of which must occur after the <*value*> field if used, are intended for use only in the mkmm.sys data file. The prot=yes field specifies that files of prod <*prod*> and type <*type*> shipped using subsequent file: entries should be checksum-protected at installation time.
The req=yes field specifies that the deflt: entry cannot be overridden by subsequent deflt: entries. VAR extension module developers should not use these options in their index files.

The “?” fields in the following file: example (and elsewhere in these entry descriptions) will be replaced by system default values for these identifications during processing by mkmm.

file Entry

The following syntax layout (and examples) folds the line in order to show the entire contents of the simulated text within the limits of the available page width. In an actual index file, however, a file: entry must be entered as a single line, and you cannot use a backslash to continue an entry across more than one line.

- **Syntax:**
  file: <*prod*> <*type*> <*sdir*> <*sname*> <*tdir*> <*tname*> [mode=<*mode*>]
  [uid=root] [prot=yes] [plat=<*plat*>]
- **Examples:**
  file: OC file ? company.png tomcat/webapps/CA Spectrum/images/
  Background ?
  file: SS,OC evformat ? CsEvFormat/Event00821000 ? ?

file: OC file source hitv-app-config.xml tomcat/webapps/CA Spectrum/WEB-INF/devman/config/contrib ?

- **Description:**
  The file: entry specifies that a shippable file is to be included in an extension module. When mkmm adds the extension module to a VCD, it attempts to locate the file and add it to a VCD record for the management module (except for a few special types of files which are added to the installation support files of the VCD). mkmm looks for the shippable file <*sdir*>/<*sname*>. If mkmm finds that file, it adds the file to the VCD record so that it will later be installed as <*tdir*>/<*tname*> relative to the DX NetOps Spectrum installation directory; otherwise mkmm fails.
  The file: entry gives the extension module developer precise control over the locations of installed files without having to relocate source files.On the Windows platform, if mkmm is unable to locate the source file <*sdir*>/<*sname*>, it attempts to locate <*sdir*>/<*sname*> with various common extensions (by default, .exe, .cmd, and .bat). If mkmm finds such a file, it adds the file to the VCD so that it will be installed as <*tdir*>/<*tname*> with the same extension. This reduces the need to include multiple file: entries to ship executable files on both Windows and UNIX platforms.
  In the file: entry, the <*prod*>, <*type*>, <*sdir*>, <*sname*>, <*tdir*>, and <*tname*> fields are required, while the remaining fields are optional:
  - ***<prod>***
    File product
  - ***<type>***
    File type

  The <*prod*> and <*type*> fields, respectively, identify the product and file type associated with the file being distributed. The mkmm tool uses <*prod*> and <*type*> to help determine default directories for locating and installing the file (see the associated <*sdir*> and <*tdir*> discussions that follow). Install uses <*prod*> and <*type*> to decide what standard processing to apply to the file during installation.
  The developer must exercise some care in selecting the <*prod*> and <*type*> designations, as this combination in effect determines how Install will process the information in the <*sdir*><*sname*> file to create the <*tdir*><*tname*> file on the target system.
- ***<sdir>***
  <*sdir*> is the source directory of the file to be distributed. <*sdir*> may be an absolute path name, a relative path name (relative to the directory where mkmm is invoked), or the “?” variable. If the “?” variable is used, this causes mkmm to supply a system default value, which either is a standard default value depending on <*prod*> and <*type*>, or else is a default value defined in a previous deflt: command. When the file is in the same working area, the usual practice is to enter a period (.) as the source directory designation.
- ***<sname>***
  <*sname*> is the source name of the file to be distributed; this file will have all of the entries that need to go into the target file (<*tname*>) that will be created by Install in accordance with the process applicable to the <*prod*><*type*> combination, as previously noted. <*sname*> must be an explicitly relative path name or file base name (relative to <*sdir*>) and should be unique; developers should individualize such files by appending the developer name or the model type name either as a prefix or an affix. If <*sname*> is expressed as the “?” variable, however, mkmm will fail.
- ***<tdir>***
  <*tdir*> is the target directory of the file when it is installed. <*tdir*> may be either a relative path name (relative to the installation root directory) or the “?” variable. If <*tdir*> is the “?” variable, mkmm supplies a default value. This value is either a standard default value, depending on <*prod*> and <*type*>, or a default value defined in a previous deflt: command. For detailed information on system defaults for <*tdir*>, examine the mkmm.sys entries. In general, you should not ship to specific directories, but instead use the deflt: directories as defined by the SEI toolkit.
- ***<tname>***
  <*tname*> is the target name of the file when it is installed. <*tname*> may be any of a relative path name, or a file basename (relative to <*tdir*>), or the “?” variable. If <*tdir*> is the “?” variable, mkmm defaults that field to the value of the applicable <*type*> for the given <*prod*>, as determined by <*sname*>.
- **mode=<*mode*>**
  This optional field must occur after the <*tname*> field. <*mode*> must be a legal argument to the POSIX “chmod” command (for example, “mode=644,” “mode=a+x,” or “mode=-w”). This option is used to alter the access permissions of the installed target file. The determination of the installed file permissions is done as follows:

  - On Linux, the target file is added to the VCD with the same permissions as the source file. At installation time, when the target file is extracted, a umask of 022 is applied, turning off write and execute permissions for group and other. Then, if the file: entry for the file includes a mode=<*mode*> option, “chmod <*mode*>” is performed on the file.
  - On Windows, the target file is added to the VCD with full control access for the DX NetOps Spectrum Users group and with its readonly permission matching that of the source file. At installation time, the target file is extracted with these permissions. Then, if the file: entry for the file includes a mode=<*mode*> option, it is applied as follows: If “chmod <*mode*>” specifies to turn on (off) UNIX user write permission, readonly permission of the target file is turned off (on).
- **uid=root**
  This optional field must occur after the <*tname*> field. This option is used to alter the ownership of the installed target file. On Windows the ownership of the installed file is unaffected (all installed target files belong to the DX NetOps Spectrum Users group).
- **prot=yes**
  This optional field must occur after the <*tname*> field. It specifies that at installation time, checksum protection should be used to save the previously installed version of this file if it contains user edits.
  The prot=yes option is intended for use only in CA-produced index files. VAR extension module developers should not use this option.
- **plat=<*plat*>**
  This optional field must occur after the <*tname*> field. This option specifies that the file: entry is to be processed only on the platform designated by the <*plat*> field.
  Since the code setup allows multiple plat=<*name*> flags to be given in sequence on the same line, the presence of multiple entries allows the file: entry to be processed on multiple, specified platforms. However, since the same capability exists if no plat=<*name*> is given, the preferred approach would be to not include this optional field when the file: entry is to be applicable for all platforms (Windows and Linux).
- **plat!=<*plat*>**
  If used, this optional field must occur after the <*tname*> field. This flag specifies that the file: entry is to be processed only on platforms other than the one designated by the <*plat*> field.
  plat=plat and plat!=plat flags cannot be used together in the same file: entry. If neither plat=plat nor plat!=plat is used, the file: entry is processed on every platform.

fdir Entry

The following syntax layout folds the line in order to show the entire contents of the simulated text within the limits of the available page width. However, in an actual index file, your fdir: entry must be entered as a single line, and you cannot use a backslash to continue an entry across more than one line.

- **Syntax:**
  fdir: <*prod*> <*type*> <*sdir*> <*snamepat*> <*tdir*> <*tname*>
  [mode=<mode>] [uid=root] [prot=yes] [plat=<plat>]
- **Example:**
  fdir: OC file source/devman \.xml$ tomcat/webapps/spectrum/WEB-INF/devman ?
- **Description:**
  The fdir: entry specifies that mkmm should traverse a directory, locating all files having base names (filenames less any extension suffixes) that match a specified pattern, and generate file: entries to ship all matching files found.
  The fdir: entry is useful for shipping all files of a given type (identifiable by base name) within a specified directory, where the directory structure of the shipped files should correspond to the directory structure of the files in the development area.
  In the fdir: entry, the <*prod*>, <*type*>, <*sdir*>, <*snamepat*>, <*tdir*>, and <*tname*> fields are required, while the remaining fields are optional.
  When mkmm processes an fdir: entry, it traverses the <*sdir*> directory looking for files to include in the extension module. Whenever it finds a file having a base name that matches the pattern <*snamepat*>, it effectively generates a file: entry to ship the matching file. The <*sname*> and field of each generated file: entry is the name of a matched file relative to <*sdir*>. The <*tname*> field of each file: entry is ? regardless of the <*tname*> specified in the fdir: entry. All

other fields of the generated file: entries -- including the <*prod*>, <*type*>, <*sdir*>, <*tdir*>, and flag fields -- are the same as the corresponding fields of the file: entry.
The <*snamepat*> pattern field must be a regular expression, as used by the Perl programming language (similar to the regular expressions used by the UNIX grep command). <*snamepat*> is used to match the base names of files encountered during traversal of the <*sdir*> directory, not their path names relative to <*sdir*>.
While this document does not describe the complete syntax of Perl-style regular expressions, the following table shows how files are matched by specific <*snamepat*> patterns, thereby indicating how to compose most fdir: entries:

<snamepat>	**Matches files whose base name …**
\.xml$	Ends with extension .xml
^[a-z]	Begins with a lower case letter
^Makefile$	Is “Makefile”
file	Contains the substring “file”
.	Includes all files in the directory

head Entry

The following syntax layout folds the line in order to show the entire contents of the simulated text within the limits of the available page width. However, in an actual index file, your head: entry must be entered as a single line, and you cannot use a backslash to continue an entry across more than one line.

- **Syntax:**
  head: <*prod*> <*type*> <*sdir*> <sname> <*tdir*> <*tname*> [plat=<plat>]
  [mode=<mode>]
- **Description:**
  The head: entry should be avoided in VAR-developed index files.
  The head: entry specifies that a file should be added to the installation support files on the VCD. Except for its label, the syntax and usage of the head: entry is precisely the same as that of the file: entry.
  Any file: entry with <*prod*> = Install or <*type*> = mmdesc, preinst, or cus is interpreted as a head: entry, and mkmm adds the file to the installation support files on the VCD.

obsbackup Entry

Syntax:

obsbackup: <*prod*> <*type*> <*sdir*> <*sname*> <*tdir*> <*tname*>
[mode=<*mode*>] [uid=root] [prot=yes] [plat=<*plat*>]

> **NOTE**
> The syntax parameters for this entry are the same as those for the file entry.

Example:

obsbackup: SS vfile ? ? SS/CsVendor/Ctron_MMAC_Pls/9G426_02 AlertMap

Description:

The obsbackup entry indicates that the specified file is to be obsoleted by the installation. A backup copy of the file will be created in the <*$SPECROOT*>/Install-Tools/LOGS/<version_date>/SavedFiles directory. The backup will include the path to the original file.

For example, if the <*$SPECROOT*>/SS/CsVendor/Ctron_MMAC_Pls/9G426_02 AlertMap file were to be obsoleted, the following backup file would be created: <*$SPECROOT*>/Install-Tools/LOGS/<version_date>/SavedFiles/SS/CsVendor/Ctron_MMAC_Pls/9G426_02 AlertMap.

obschkbackup Entry

The syntax parameters for this entry are the same as those for the file entry.

- **Syntax:**
  obschkbackup: <*prod*> <type> <*sdir*> <*sname*> <*tdir*> <*tname*>
  [mode=<*mode*>] [uid=root] [prot=yes] [plat=<*plat*>]
- **Example:**
  obschkbackup: SS vfile ? ? SS/CsVendor/Ctron_MMAC_Pls/9G426_02 AlertMap
- **Description:**
  The obschkbackup entry indicates that the specified file is to be obsoleted by the installation. If the file has been previously customized in some way, a backup copy of the file will be created in the <*$SPECROOT*>/Install-Tools/LOGS/<version_date>/SavedFiles directory. The backup will include the path to the original file.
  For example, if a DX NetOps Spectrum upgrade were to obsolete the <*$SPECROOT*>/SS/CsVendor/Ctron_MMAC_Pls/9G426_02 AlertMap file, and the obschkbackup entry had been used to specify that the file must be backed up if it had ever been customized, the DX NetOps Spectrum installation program would use a checksum value to determine if the file had ever been customized. If the file had been customized, the following backup file would be created: <*$SPECROOT*>/Install-Tools/LOGS/<version_date>/SavedFiles/SS/CsVendor/Ctron_MMAC_Pls/9G426_02 AlertMap.

obsdir Entry

Syntax:

obsdir: <*prod*> <*type*> ? ? <*dest dir*> ?

Example:

obsdir: SS vfile ? ? SS/CsVendor/Nortel_Carrier/Passport74xx ?

Description:

The obsdir directive will recursively remove all contained files and subdirectories within the <*dest dir*> directory.

obsnobackup Entry

Syntax:

obsnobackup: <*prod*> <*type*> <*sdir*> <*sname*> <*tdir*> <*tname*>
[mode=<*mode*>] [uid=root] [prot=yes] [plat=<*plat*>]

> **NOTE**
> The syntax parameters for this entry are the same as those for the file entry.

Example:

obsnobackup: SS vfile ? ? SS/CsVendor/Ctron_MMAC_Pls/9G426_02 AlertMap

Description:

The obsnobackup entry indicates that the specfied file is to be obsoleted by the installation, and a backup copy should not be made.

CA-Reserved Entries

The entry labels info: and mtype: are reserved for use by CA and should not be used in VAR-developed index files.

CA-reserved entries are restricted for use by CA and are subject to undocumented changes. Unauthorized use of these CA reserved entries in a VAR-developed index file can jeopardize the extension module installation or the behavior of the installed product.

Orphan Extension Modules

At installation time, all purchasable parts are selected for installation. If an extension module on the VCD does not belong to any purchasable part on the VCD, the module will not be installed.

For example, the extension module JOE would be an orphan under either of the following conditions:

- JOE is not included as a part of itself, that is, the index file JOE.i does not include any pp: or pprep: entries
- None of the other extension modules on the VCD include JOE as a part, that is, none of them have index files that include either a pp: entry or a pprep: entry mentioning JOE

The mkcd tool detects orphan extension modules on a VCD. By default, mkcd will fail if orphan extension modules are detected.

Before distributing a VCD to a customer, make sure that the VCD does not contain any orphan parts. You can do this by adding pp: and/or pprep: entries to the applicable index files to ensure that all extension modules on the VCD belong to some part on that VCD.

Example Index File

The following example presents an index file for a fictitious DX NetOps Spectrum management module. The intent of this example is to clarify the use of index file entries that are commonly used in development of the management module index files.

The scenario for this example is that the Jill Corporation sells a line of network hub devices, for which it provides DX NetOps Spectrum management support. As a DX NetOps Spectrum VAR, the Jill Corporation received the software developer name of JillCorp and hexadecimal Developer ID 2f7 from CA.

Jill Corporation has already developed the model type Hub_Jill, generic management of its line of hub devices, which is distributed and sold as part of the Hub_Jill management module. Jill Corporation recently added the new Jill 700 hub to its line, and it wants to develop the new DX NetOps Spectrum management module Hub_Jill700 for the Jill 700 hub, to be packaged and sold with the hub hardware.

To accomplish this task, a developer at Jill Corporation used CA Model Type Editor to derive a new model type Hub_Jill700 for the Jill 700 hub from the base model type Hub_Jill and created the database import file Jill700.e for the new Hub_Jill700 model type. This developer also used the Level 1 Toolkit and the CORBA Toolkit to create the other DX NetOps Spectrum support files that are required to manage the Jill 700 hub. On the development host, the DX NetOps Spectrum support files for the Jill 700 hub reside in a DX NetOps Spectrum installation area in the /usr/devarea/spectrum directory.

The complete set of DX NetOps Spectrum support files for the Jill 700 hub relative to /usr/devarea/DX NetOps Spectrum is:

- A part description file
  Hub_Jill700.mmd
- A SpectroSERVER database import file
  SS/db/Jill700.e
- A SpectroSERVER Event Disposition file
  SS/CsVendor/JillCorp/EventDisp
- A SpectroSERVER Alert Map file
  SS/CsVendor/JillCorp/Hub_Jill700/AlertMap
- A OneClick app-config file
  tomcat/webapps/spectrum/WEB-INF/devman/config/contrib/

jill700-app-config.xml

- Several OneClick customization xml files
  tomcat/webapps/spectrum/WEB-INF/devman/config/
  view-jill700-devicedetails-config.xml
  tomcat/webapps/spectrum/WEB-INF/devman/config/
  privileges-jill700-config.xml
  tomcat/webapps/spectrum/WEB-INF/devman/config/
  table-jill700-sessions-config.xml
  tomcat/webapps/spectrum/WEB-INF/devman/config/
  search-jill700-criteria.xml

The developer must now package these files into the new Hub_Jill700 management module. The ultimate object is to add this management module to a VCD so that it can be distributed to customers and installed into their DX NetOps Spectrum areas.

The first task is to create an index file for Hub_Jill700. The index file must have the same name as the management module, and, therefore, it is named Hub_Jill700.i. We assume Hub_Jill700.i is created in the DX NetOps Spectrum installation directory /usr/devarea/spectrum. The developer has the option of writing the index file manually or using the mmship tool to generate the index file and then editing the resulting file. For the purposes of this example, assume that the developer has chosen to write the index file from scratch.

When the developer has finished editing Hub_Jill700.i, its contents are as follows (with line numbers included to allow for easy reference):

```
1. mm: Jill 700 Hub Management Module
2. rev: 1.0
3. irev: 01.00.00.000
4. level: 1
5.
6. vend: JillCorp
7. pprep: 3M-02F7-1001 Hub_Jill
8. file: Install mmdesc ? Hub_Jill700.mmd ? ?
9.
10. deflt: SS db sdir $SPECROOT/SS/db
11. file:  SS db ? Jill700.e ? ?
12.
13. deflt: SS vfile sdir $SPECROOT/SS
14. file:  SS vfile ? CsVendor/JillCorp/EventDisp ? ?
15. file:  SS vfile ? CsVendor/JillCorp/Hub_Jill700/AlertMap ? ?
16.
17. deflt: SS,OC evformat sdir $SPECROOT/custom/Events
18. file:  SS,OC evformat ? CsEvFormat/Event00821000 ? ?
19. file:  SS,OC evformat ? CsEvFormat/Event00821001 ? ?
20.
21. deflt: SS,OC pcause sdir $SPECROOT/SG-Support
22. file:  SS,OC pcause ? CsPCause/Prob00821000 ? ?
23. file:  SS,OC pcause ? CsPCause/Prob00821001 ? ?
24.
25. deflt:  OC file sdir $SPECROOT/tomcat/webapps/spectrum/WEB-INF/devman
26. deflt:  OC file tdir tomcat/webapps/spectrum/WEB-INF/devman
27. file:   OC file ? config/contrib/jill700-app-config.xml ? ?
28. file:   OC file ? config/view-jill700-devicedetails-config.xml ? ?
29. file:   OC file ? config/privileges-jill700-config.xml ? ?
30. file:   OC file ? config/table-jill700-sessions-config.xml ? ?
```

```
31. file:   OC file ? config/search-jill700-criteria.xml ? ?
```

Let us analyze this index file.

The name of the index file is Hub_Jill700.i, indicating that it is the index file for the Hub_Jill700 management module.

The first five lines of Hub_Jill700.i describe the Hub_Jill700 management module.

- Line 1 says that the descriptive name of the Hub_Jill700 management module is “Jill 700 Hub Management Module.” The Install program displays this name on the installation summary. Line 2 says that the descriptive version of the Hub_Jill700 management module is 1.0.
- Line 3 says that the internal version of the Hub_Jill700 management module is 01.00.00.000. (The internal version must have the format dd.dd.dd.ddd where each d is a decimal digit.) During installation, this number is used to detect installation of an older version Hub_Jill700 over a newer version (such an occurrence can be named as downgrading, and it should only be done with the consent of the installing user). If Jill Corporation elects to update and re-release the Hub_Jill700 management module in the future, the “patch release” segment (the final ddd value) of this number must have a higher value than zero (for example, 01.00.00.001 or 01.01.00.123). This indicates that the re-released management module is newer than the original release.

  NOTE
  The “Patch Release” value (the final ddd segment of the number) must be higher than any preceding version of a given module to avoid being flagged as downgrading with this value starting over again at zero whenever any of the preceding number segments become incremented.
- Line 4 says that the extension module level of Hub_Jill700 is 1. This indicates that Hub_Jill700 is an optional extension of the DX NetOps Spectrum product that is, a management module (as opposed to a required DX NetOps Spectrum core component or an independent external application).
- Line 6 says that the developer of Hub_Jill700 is JillCorp. This is the CA-assigned developer name for Jill Corporation.

Lines 7 and 8 are both part description entries:

- Line 7 says that the management module Hub_Jill700 represents a part that is named 3M-02F7-1001. The 3M-02F7-1001 part includes the physical Hub_Jill700 and the additional management module Hub_Jill (which includes the base model type for Hub_Jill700).
- Line 8 ships a part description file for part 3M-02F7-1001. This file is required but not used during install.
  Line 8 gives us a first chance to see how the file: entry works. The line reads as follows:

  ```
  file:   Install mmdesc ? Hub_Jill700.mmd ? ?
  ```

  The file: label indicates that a shippable file is being included in the management module.
  The next two fields (Install mmdesc) give the product and the type of the shippable file. The file product Install indicates that the file is used during installation of the management module. The file type mmdesc says that the file is a management module part description file.
  The next two fields (? Hub_Jill700.mmd) specify the location of the source for the part description file (the file we eventually intend to place on the VCD). The source directory field is ?, which indicates that the default value can be used. The default source directory for part description files is “.” (as specified by the following deflt: entry in the mkmm.sys file that is shipped with the SEI Toolkit):

  ```
  deflt:   Install mmdesc sdir .
  ```

  The source name of the part description file is given explicitly as Hub_Jill700.mmd. Keep the source directory and name together, so that the location of the source for the part description file is ./Hub_Jill700.mmd. Since this is a relative path, it is taken relative to the directory containing the index file, which in this case is /usr/devarea/spectrum. Putting it all together, the source for the part description file is /usr/devarea/spectrum/Hub_Jill700.mmd.
  The last two fields (? ?) specify where the part description file should be installed from the VCD. Since the target directory field is ?, the Install program uses the default target directory for part description files. This default directory is Install-Tools/MMD, as specified by the following deflt: entry in the mkmm.sys file:

  ```
  deflt:   Install mmdesc tdir Install-Tools/MMD req=yes
  ```

The req=yes option that is shown in this example is intended for use only in the mkmm.sys data file. Since Install-Tools/MMD is the only acceptable target directory for installing part description files, the req=y flag prevents Install-Tools/MMD from being overridden by other deflt: entries in mkmm.loc or the index file. VAR extension module developers should not use this req=yes option in their index or mkmm.loc files.

Since the target name is also ?, we use the default target file name, which is always the same as the source file name -- in this case, Hub_Jill700.mmd. So, the part description file installs as Install-Tools/MMD/Hub_Jill700.mmd relative to the DX NetOps Spectrum installation directory.

- The deflt: entry on line 10 and the other deflt: entries that follow it set up the default source directories and target directories for the various other types of files that are being included in the management module. Look at Line 10 as a representative case with the following syntax:

  ```
  deflt:  SS db sdir $SPECROOT/SS/db
  ```

  $SPECROOT is an environment variable, which is expanded first to its exported value. In a Level 1 DX NetOps Spectrum development environment, $SPECROOT is set to the DX NetOps Spectrum installation directory, in this case, /usr/devarea/spectrum. So the deflt: entry becomes:

  ```
  deflt:  SS db sdir /usr/devarea/spectrum/SS/db
  ```

  The deflt: label itself indicates that we are setting a default attribute for shippable files.
  The next two fields (SS db) indicate that we are setting the default attribute for shippable files with product SS and file type db. These are SpectroSERVER database import files.
  The next field (sdir) indicates that the default attribute we are setting is the default source directory.
  The last field (<*$SPECROOT*>/SS/db) is the default value being set. This entry thus sets the default source directory for SpectroSERVER database files to /usr/devarea/spectrum/SS/db. This default source directory applies to any subsequent “file: SS db” entries in the index.
- The file: entry on line 11 and the other file: entries that follow it describe the shippable files in the management module. Look at line 11 as a representative case, which reads with the following syntax:

  ```
  file:   SS db ? Jill700.e ? ?
  ```

  The file: label indicates that a shippable file is being included in the management module.
  The next two fields (SS db) give the product and the type of the shippable file. The file product SS indicates that the file is part of the SpectroSERVER product. The file type db says that the file is a SpectroSERVER database import file.
  The next two fields (? Jill700.e) specify the location of the source for the SpectroSERVER database import file. The source directory field is ?, indicating that we can use the default source directory for import files.
  As previously discussed, on line 10, the default source directory is set to <*$SPECROOT*>/SS/db, which becomes /usr/devarea/spectrum/SS/db. The source name of the part description file is given explicitly as Jill700.e. Putting the source directory and name together, the source location of the part description file is /usr/devarea/spectrum/SS/db/Jill700.e.
  The last two fields (? ?) specify where the database import file should be installed from the VCD. The target directory ? indicates that we can use the default target directory for database import files; that default target directory is SS/db (as set in mkmm.sys). Since the target file name is also ?, we use the default target file name, which is always the same as the source file name, in this case, Jill700.e. So, the part description file can be installed as SS/db/Jill700.e relative to the spectrum installation directory.

  NOTE
  You must place all the custom defined events, Pcause, and event table files of the Index file under the $*SPECROOT*\custom directory.

Creating VCDs

Special Requirements on Windows

When using the SEI Toolkit tools on the Windows platform, some special considerations apply:

- On the Windows platform, the proper behavior of tools in the SEI Toolkit requires prior installation of CYGWIN32 UNIX support software. This will not be a problem, normally, as the required software is shipped with DX NetOps Spectrum

and is installed prior to installing DX NetOps Spectrum. However, be aware that interim removal or modification of the installed CYGWIN32 software can result in failure of SEI Toolkit tools (as well as other DX NetOps Spectrum software).
- On the Windows platform, it is necessary to run SEI Toolkit tools within a CYGWIN32 Bash Shell (bash) environment. You must also keep this requirement in mind when invoking SEI Toolkit tools from Makefiles or other build management tools.

Editing Setup Initialization Files

In order for your product name to appear during the installation of the VCD, you must edit the following setup initialization files in the <*$SPECROOT*>/INSDK/CD directory:

- (Windows) setupnt.ini
- (Linux) setuplin.ini

In each file, replace the words YOUR_PRODUCT _NAME_HERE with the name of your product, and then save and close the file.

If you did not make the edits described above and have already created a VCD, to correct this problem, you can replace the words YOUR_PRODUCT_NAME_HERE in the following files:

- <*VCD*>/setupnt.ini
- <*VCD*>/setuplin.ini

Create a VCD Using mkmm

You can use the mkmm tool to create a VCD and/or add one or more extension modules to a VCD.

System Environment

The mkmm utility must reside in its originally installed location in the <*$SPECROOT*>/INSDK directory, and it must be invoked from the directory containing the index files to be processed.

Syntax

The following syntax layout folds the line in order to show the entire contents of the sample text within the limits of the available page width. In actual use, however, your mkmm command-line entry must be entered as a single line:

```
mkmm [exts=<extlist>] [any=y/n] [vcd=<vcddir>] <indexFile>
<indexFile> ...
```

Description

The mkmm tool uses an extension module index file to add an extension module to a VCD. This involves transferring the descriptive information and shippable files for the extension module to the VCD. If the VCD does not yet exist, mkmm creates the VCD. The VCD may be used to install the VCD into an existing DX NetOps Spectrum installation area. You can use a single invocation of mkmm to add several extension modules to a single VCD.

The mkmm tool is located in the <*$SPECROOT*>/INSDK directory. The straight-forward use of mkmm is simply to run it as a command, using as its arguments the names of the index files to be added to the VCD. Use of the optional arguments allows great flexibility.

If mkmm is invoked with no options, it prints a usage summary identifying the required syntax and inputs, as follows:

```
Error: No arguments specified.
Usage: mkmm [options] [indexFile] [indexFile] ...
options:
    exts=<extlist>      Specify filename extensions   default: ".exe,,.cmd,.bat"
```

```
    any=(y|n)           Create platform-neutral VCD   default: n
    vcd=<vcdDirectory>  VCD directory                 default: ./vcd

required:
    indexFile           Index file
```

The VCD created by the mkmm tool cannot be installed immediately. Instead, you must first use the mkcd tool to add a version number to the VCD.

Environment Variables

Any environment variable included explicitly in an index file must be exported before mkmm can be invoked. Also, if an index file uses a default value that references an environment variable, the referenced environment variable must be exported before invoking mkmm.

mkmm sometimes uses the $VCD environment variable to determine the location of the VCD. For details, see the discussion of the vcd=<*vcddir*> entry in the next section on command line options.

Input Files

mkmm uses the following input files:

- **index files**
  For each extension module specified by its index file, mkmm reads the index file, <moduleName>.i.
- **shippable files**
  The mkmm tool reads each shippable file mentioned in the file: and head: entries of each index file it processes.
- **mkmm.sys**
- **mkmm.loc**
  The mkmm tool reads the mkmm.sys and mkmm.loc data files to determine the set of file types legal in DX NetOps Spectrum extension module index files. The tool also reads the default characteristics (including source and target directories, and so on) associated with those file types.

Command Line Options

The following options are available for the mkmm tool; note that some can be used in combination, while others are mutually exclusive.

- **any=n**
  (Default) Allows the extension modules being added to the VCD to be installed from the VCD only on the current DX NetOps Spectrum platform.
  This option should be used with non-portable extension modules, that is, extension modules that contain shippable files that are not portable across all supported DX NetOps Spectrum platforms (such as executables or object files). To distribute a non-portable extension module on multiple platforms, it is necessary to install the SEI Toolkit and run mkmm on each platform.
- **any=y**
  Allows platform-neutral extension modules being added to the VCD to be installed from the VCD on any supported DX NetOps Spectrum platform.
  This option should be used with portable extension modules, that is, extension modules containing only shippable files which are portable across all supported DX NetOps Spectrum platforms.
  If this option is used to ship a non-portable extension module, installation failure or product failure will result.
- **exts=<*extlist*>**
  Use the <extlist> list of filename extensions when looking for shippable files. By default, <extlist> is “,.exe,.cmd,.bat.” Using filename extension lists helps make index files more portable between platforms.
- ***<indexFile>***

Adds extension module <indexFile> to the VCD. <indexFile> may be the name of an extension module (for example, JOE) or its index file (for example, JOE.i) Either of these would cause mkmm to read the index file <indexFile>.i and add extension module <indexFile> to the VCD.
You can add several extension modules to the VCD by including several <indexFile> options on the mkmm command line.

- **vcd=<*vcddir*>**
  Adds the extension modules to the VCD in directory <vcddir>.
  If this option is omitted, mkmm obtains the VCD directory from the export environment variable, $VCD. If $VCD is not set and exported, mkmm uses the VCD directory, ./vcd. If there is no VCD in the VCD directory, mkmm creates a new VCD.

Output Files

The mkmm tool creates the following files:

- **error files**
  Any errors generated by mkmm while adding extension module <*mmname*> to the VCD are saved in the files <*mmname*>.err files.
- **warning files**
  Any warnings generated by mkmm while adding extension module <*mmname*> to the VCD are saved in the <*mmname*>.warn files.
- **VCD**
  The mkmm tool creates a VCD area if there is none and adds extension module files to the VCD area if it already exists.

Exit Status

If mkmm is able to successfully add the extension modules to the VCD for all index files specified on its command line, it returns with exit status 0; otherwise it returns with exit status 1.

Create a VCD for a Distributed Installation

To allow the VCD to be installed in a distributed SpectroSERVER environment, copy the sdic directory in <*$SPECROOT*>/Install-Tools and paste it into the <*$VCD*> directory:

```
cp -r <$SPECROOT>/Install-Tools/sdic <$VCD>/sdic
```

Perform the copy on each platform on which you want to run a distributed installation.

If you are working in the Windows environment, complete these additional steps

1. Create a dummy text file using a text editor. This text file should not contain any information.
2. Replace the cygdist.tgz in the <*$VCD*>/sdic directory with a new gzipped tar file of the same name. This new file should only contain the dummy text file.
   ```
   <$SPECROOT>/Install-Tools/newtar czvf cygdist.tgz <dummy_text_filename>
   ```
   WARNING
   Do not ship CA's cygdist.tgz file. It will cause installation problems.
3. Place cygdist.tgz in the <*$VCD*>/sdic directory. When this file is unpacked during the installation process, it will be placed in the <*$SPECROOT*>/NT-Tools/SRE directory and be used instead of the archive of the distribution of Cygwin shipped with DX NetOps Spectrum.
4. Copy the userconf.exe file from a Windows DX NetOps Spectrum installation area running the same version of DX NetOps Spectrum for which you are creating the VCD. Put this file in the <*$VCD*>/nt directory.
   ```
   cp <$SPECROOT>/Install-Tools/userconf.exe $VCD/nt
   ```

Platform-Neutral Extension Modules

A *platform-neutral extension module* is an extension module that can be installed without change on any supported DX NetOps Spectrum platform. You can add such platform-neutral extension modules to a VCD by using the mkmm “any=y” option. Management modules created by the use of the Level 1 Toolkit typically are platform-neutral extension modules.

If mkmm is running on Windows, for example, the mkmm any=y option causes mkmm to process the index file as a platform-neutral index file. This means that the extension module will be added to the any (platform-neutral) area of the VCD and hence that the extension module can later be installed from the VCD onto any supported DX NetOps Spectrum platform. If the index file contains any platform-specific entries (entries with plat=<*platform*> modifiers), mkmm will fail.

You can use this mkmm any=y option only if every shippable file in the extension module is portable with respect to all supported DX NetOps Spectrum platforms -- that is, the extension module cannot contain any files with formats specific to a given platform (such as object files or completed executables). Furthermore, a VCD cannot contain both platform-neutral and platform-dependent extension modules at the same time, so you should not add a platform-neutral extension module to a VCD unless you intend all extension modules on that VCD to be platform-neutral.

mkmm.sys and mkmm.loc

The mkmm tool uses two data files, mkmm.sys and mkmm.loc, to define the set of file types that may legally be used in file: and head: entries in index files, as well as the standard default characteristics (including source directory, target directory, and so on) associated with each file type. Through proper use of these files, extension module developers can customize the default behavior of mkmm when processing file: and head: entries. Specifically, extension module developers can change the standard default source directories associated with any given file type.

CA ships the mkmm.sys file with the SEI Toolkit, providing default values consistent with proper installation of the DX NetOps Spectrum product. The extension module developer should never remove or modify this file to change the default settings. Removal or improper modification of mkmm.sys can cause mkmm failure or result in production of VCDs that install shippable files to incorrect locations. Moreover, future upgrades of the SEI Toolkit will overwrite mkmm.sys, undoing any interim modifications.

Customization of these default values can be accomplished by modifying the mkmm.loc file, which does not ship with the SEI Toolkit. Instead, the extension module developer can create mkmm.loc (either directly or initially as a copy of mkmm.sys) and then edit it manually to customize the standard default values used by mkmm. Upgrades of the SEI Toolkit will not affect the contents of mkmm.loc, letting those changed values remain in effect.

Both mkmm.sys and mkmm.loc should reside in the same directory in which the mkmm tool resides, namely, the <*$SPECROOT*>/INSDK directory.

When processing an index file, mkmm goes through the following steps:

1. Process mkmm.sys.
   The mkmm.sys file is in index file format and contains deflt: entries for each file type that may legally be used in an index file. It should contain no other entries. These deflt: entries establish the standard default values (source directory, target directory, and checksum protection status) associated with each file type that may legally be used in an index file.
2. Process mkmm.loc (if present).
   The mkmm.loc file also is in index file format and contains only deflt: entries. Since mkmm processes the mkmm.loc entries after the mkmm.sys entries, any same-application deflt: entries in mkmm.loc override the standard default values set by the deflt: entries in mkmm.sys. This arrangement thus lets the developer customize standard default values associated with file types by adding the desired override commands to the mkmm.loc file.
3. Process the index file.
   The standard default values for the various file types defined in mkmm.sys and mkmm.loc now apply to the index file. The deflt: entries in the index file may be used to override these default values, however. The deflt: entries in mkmm.sys and mkmm.loc apply to all index files processed by mkmm, whereas deflt: entries in a given index file apply only to that index file.

The following example illustrates how you can use mkmm.loc to customize standard default values processed by mkmm.

As shipped, the mkmm.sys file contains the following deflt: entries:

```
deflt: Install mmdesc sdir .
deflt: Install mmdesc tdir Install-Tools/MMD req=yes
```

Note that designations of product “Install” and file type “mmdesc” (that is, “Install mmdesc” files, pertaining to Install description files for management modules), as shown in the preceding entries, are legal in DX NetOps Spectrum extension modules.

The standard default source directory (sdir) of “Install mmdesc” files is “.” -- meaning that mkmm, by default, looks for “Install mmdesc” files in the “.” directory (that is, in the same directory in which mkmm was invoked).

The standard default target directory (tdir) is “Install-Tools/MMD” -- meaning that mkmm, by default, adds “Install mmdesc” to the VCD so that Install subsequently can extract the included files from the VCD into the Install-Tools/MMD directory (relative to the DX NetOps Spectrum installation directory).

> **WARNING**
> The req=yes option shown in this example is intended for use only in the mkmm.sys data file. The req=yes flag prevents the standard default target directory (tdir) of “Install mmdesc” files from being overridden by later deflt: entries in mkmm.loc or the index file. VAR extension module developers should not use this req=yes option in their index or mkmm.loc files.

Now, suppose that the source for “Install mmdesc” files at a particular development site were kept in the /user/devarea/install directory. At that site, it would be convenient to arrange for mkmm to look for “Install mmdesc” files in that location. To get this to happen, the developer could add the following entry to mkmm.loc:deflt: SG stdmenu sdir /user/devarea/install.

This entry then would override the “deflt: Install mmdesc sdir .” entry in mkmm.sys, redefining the standard default source directory for “Install mmdesc” files to /user/devarea/install. This new default identification would now apply whenever mkmm is used to process any index file.

Source File Extension Handling in mkmm

This section applies primarily to developers of extension modules for the Windows platform.

There are several file naming conventions that differ between UNIX and Windows file systems. The most notable is that certain types of files on Windows platforms, such as executable files and batch files, require specific filename extensions (for example, .bat, .com, .cmd, .exe, .ksh, and so on), whereas this typically is not the case on UNIX systems.

These differences can cause inconveniences in the index files for multi-platform extension modules that ship on both UNIX-based and Windows platforms. For instance, a compiled executable called mytool on UNIX-based would be called mytool.exe on Windows. In an extension module index file prepared for shipping such an executable, therefore, there would apparently need to be two separate file: entries, one for UNIX platforms and one for Windows platforms, as follows:

```
file: SS exe ? mytool ? ? plat!=intel
file: SS exe ? mytool.exe ? ? plat=intel
```

However, this solution is unsatisfactory, because it potentially requires a lot of duplicated entries in the index files. An alternative solution would be to use a ${EXE} environment variable, set to the null string on UNIX and to “.exe” on Windows. To illustrate, consider the following single index file entry:

```
file: SS exe ? mytool${EXE} ? ? plat=intel
```

This entry would suffice to ship the file on Linux and Windows platforms. Deciding exactly which files need an extension would still be a chore when many files were involved, however.

To alleviate this burden for the extension module developer, the mkmm tool automatically checks for standard file name extensions when processing file: and head: entries.

For example, suppose you were using mkmm to add an extension module to a VCD. Suppose further that the extension module index file contained the following entry:

```
file: SS exe ? mytool ? ?
```

When mkmm processed this entry, it would look for the source file mytool with a number of standard extensions. The default list of extensions is:

- No extension
- .exe
- .cmd
- .bat

Accordingly, mkmm would look first for source file mytool, then mytool.exe, then mytool.cmd, and finally mytool.bat. on a Linux platform then, it would find and ship mytool, whereas on a Windows platform, where mytool does not exist, mkmm would find and ship mytool.exe. Therefore, this single file: entry works without change on either UNIX or Windows.

In the rare case that it may be necessary, the list of extensions used by mkmm when processing file: or head: entries can be changed by using the exts=<*extlist*> option, where <*extlist*> is a comma-separated list of extensions to be checked. For instance, consider the following command:

```
mkmm exts=,.exe,.com DCNET.i
```

This command instructs mkmm to check first for no extension (because there is nothing in <*extlist*> before the first comma), then for an .exe extension, and finally for a .com extension when processing file: and head: entries in index file DCNET.i.

If mkmm exhausts the extension list without finding the source file, it will use the source file name with no extension. This means that mkmm would not find the source file and therefore would fail.

Finishing and Versioning a VCD Using mkcd

You can use the mkcd tool to add version identification to the VCD, making it possible to install the extension module software from the VCD into an existing DX NetOps Spectrum installation area.

System Environment

The mkcd utility must reside in its originally installed location (the INSDK directory in the DX NetOps Spectrum installation area) but is invoked from the same directory in which the mkmm-output files are kept.

Syntax

```
mkcd [-f] [-t complete] <vcddir> <version>
```

or:

```
mkcd -h
```

Description

A VCD created by the mkmm tool cannot be installed immediately, Instead, the mkcd tool must first be used to add a version number to the VCD. The mkcd tool performs the following functions:

- Checks dependencies between purchasable parts and extension modules on the VCD and reports inconsistencies and errors.
- Adds a version number to the VCD, which makes the VCD installable.
- “Finishes” the VCD, locking the VCD against addition of further extension modules via mkmm, as well as preparing the VCD for customer installation.

If mkcd is invoked with no options, it prints a usage summary identifying the required syntax and inputs, as follows:

```
Usage: mkcd [-fh] <vcdDirectory> <vcdVersion>
-f: Finish VCD (prevents future MM additions)
-h: Help (display this message)
```

Command Line Options

The following options are available for the mkcd tool:

- **-h**
  Displays a help message that illustrates usage syntax requirements. This same result occurs if you enter mkcd with no options.

  NOTE
  You cannot use the -h option with other options listed below, as it takes precedence and overrides all other options.
- **-t**
  (Default) Specifies that the VCD contains a complete DX NetOps Spectrum distribution.
  When this option is in effect, mkcd assumes that unresolved extension module dependencies will not be satisfied at installation time, and it fails if it detects unresolved dependencies.Developers using the SEI Toolkit to distribute extension modules should never use the -t complete option when creating a VCD via mkcd.
- **-f**
  Instructs mkcd to “finish” the VCD. This locks the VCD so that no further extension modules may be added via mkmm. It also performs VCD compression and other operations to prepare the VCD for distribution and installation by customers.
  You should always “finish” a VCD before making it available to customers for installation.
- ***<vcddir>***
  Instructs mkcd to process VCD in the directory <vcddir>.
- ***<version>***
  Instructs mkcd to give VCD the version number <version>. The addition of the version number -- which is a required option -- makes it possible to install the VCD. The version number must contain no spaces.

Integration Tools Specification

Actual integration of DX NetOps Spectrum-compatible extensions with an installed DX NetOps Spectrum system is accomplished with the installation program, which is not a part of the SEI Toolkit and is mentioned here only for completeness.

When the installation program is given the VCD output from the SEI Toolkit, it can extract the DX NetOps Spectrum extension module files from the VCD and perform any necessary processing of the extracted files.

For more detailed information on the installation program, see the Installing DX NetOps Spectrum section.

Shippable Files

This section explains how shippable files are handled within the SEI Toolkit’s shippable software model. Extension module levels, products, and file types are explained, along with their effect on distribution and installation of shippable files. This chapter also includes tables that describe the following:

- How several common types of shippable files are written
- How various standard types of shippable files are distributed and installed

About Shippable Files

DX NetOps Spectrum extension modules support the distribution and installation of many kinds of shippable files. Files of different kinds are associated with different DX NetOps Spectrum products, and they must be developed, distributed, and installed in different ways. This section covers the basic concepts (levels, products, and file types) used within the SEI Toolkit to control distribution and installation of shippable files.

During installation, files of different types are integrated into the DX NetOps Spectrum product in different ways. This section lists the various file types and describes the processing applied to files of each type.

Among the many shippable file types supported by the SEI Toolkit are several types that either are used to customize the installation of extension modules or are commonly included in extension modules. These file types include the following:

- **Part description files** that contain part descriptions
- **Preinstall scripts** used primarily to remove or save obsolete files just prior to installing a new extension module
- **Custom install scripts** that perform customized installation functions for extension modules
- **Icon definition files** that describe icons for applications

Levels, Products, and File Types

The way in which a shippable file in an extension module is packaged, distributed, and installed is determined largely by the following factors:

- **The level of the extension module containing the file.** The extension module level is specified via the “level: <*level*>” entry in the extension module index file. <*level*> may have the value 0, 1, or 2.
- **DX NetOps Spectrum associated with the file.** A file’s product is given by the <*prod*> field of the file: or head: index file entry for the file. <*prod*> may have any one of these values: Install, SS, or OC. You can specify multiple values separated by commas.
- **The file type associated with the file.** A file’s type is given by the <*type*> field of the file: or head: index file entry for the file. There are many possible file types, all of which are documented later in this section.

Except in a few cases, the level and product of a file determine its relationship to the installed DX NetOps Spectrum product, as outlined in the following table:

<*level*>	**<*prod*>**	**Identification and Relation to Installed Product**
0, 1, 2	Install	Installation support file. Includes general DX NetOps Spectrum installation support files and installation support files for individual extension modules (for example, index files, part description files, preinstall scripts, and custom install scripts). Files of certain types (for example, mmdesc, preinst, and cus) are classified as installation support files regardless of their associated product.
0	SS	SpectroSERVER core file. Required for installation of the SpectroSERVER product. Level 0 extension modules are produced only by CA.
1	SS	SpectroSERVER management module file. May be optionally integrated into the SpectroSERVER product (typically, to help SpectroSERVER manage a particular device or device group).
2	SS	SpectroSERVER external application file. Part of a separate application or tool used in conjunction with SpectroSERVER but not integrated into SpectroSERVER itself. A level 2 extension module can be considered as belonging to its own standalone product.
0	OC	OneClick core file. Required for installation of the OneClick product. Level 0 extension modules are produced only by CA.
1	OC	OneClick management module file. Maybe optionally integrated into the OneClick product (typically to help OneClick display a particular device or device group).

2	OC	OneClick the external application file. Part of a separate application or tool used in conjunction with OneClick, but not integrated into OneClick itself. A Level 2 module can be thought of as belonging to its own standalone product.

Development of Shippable Files

The following table outlines the preferred development methods for files that are to be shipped in DX NetOps Spectrum extension modules. Each row of the table lists:

- The product of the shippable file as specified in the <*prod*> field of the file: or head: entry in the extension module index file.
- The file type of the shippable file as specified in the <*type*> field of the file: or head: entry in the extension module index file.
- A short description of the shippable file of the given product and type identifying the applicable nomenclature as a descriptive name.
- A short description of the format of the shippable file indicating the preferred method of development for the file where possible. Where special editors are recommended, the table gives references.

<*prod*>	**<*type*>**	**Description**	**File Format**
Install	doc	Documentation file	Text format
Install	file	General Install file	Any format.
Install	mmdesc	Part description file	Text file.
Install	tool	Installation tool	Compiled program, script, or data file.
SS	convtab	SS database partition conversion table	Text file. Undocumented format, internal to CA.
SS	cus	SS custom install script	Bourne shell script fragment.
SS	db	SS database import file	Data file. Format created by dbtool's export function or the Model Type Editor's export function.
SS	doc	SS documentation file	Text file.
SS	file	SS general file	Any format.
SS	icondef	icon definition for shortcuts	Used by Install to create icons on the user's desktop.
SS	iconimg	icon image for shortcuts	Used to ship icon images for use by icons created with icondef files.
SS	lib	SS library file	Compiled object library.
SS	preinst	SS preinstallation script	Text file containing Perl 5 code fragment.
SS	tool	SS related tool	Compiled program, script, or data file.
SS	vfile	SS vendor file	Text file in Alertmap file format or EventDisp file format.
OC,SS	evformat	event format file	Event format file in custom/Events/CsEvFormat.
OC,SS	pcause	probable cause file	Probable cause file in custom/Events/CsPCause.
OC	file	OneClick general file	Any format.
OC	web_xml	file to contribute to web.xml	An XML file that will be incorporated into the tomcat/webapps/spectrum/WEB-INF/web.xml file.
OC	cus	OneClick custom install script	Bourne shell script fragment.
OC	exe	OneClick executable file	A file that needs to be executed and, therefore, have the execute mode set on the file.
OC	preinst	OneClick preinstallation script	Text file containing Perl 5 code fragment.

Installation of Shippable Files

To install a usable DX NetOps Spectrum extension module, it is seldom enough to extract the files from the VCD to the correct location in the installation area. Additional processing of the installed files typically needs to be done, for example, importing SpectroSERVER database import files into the SpectroSERVER database.

The Install program performs much of this additional processing as standard installation services. Install decides what processing to perform on a shippable file by looking at the file's product and file type. For example, files of product "SS" and file type "db" are identified as database import files and imported into the SpectroSERVER database.

If a file is erroneously shipped in an extension module of an inappropriate level, Install may not correctly process the file.

Similarly, if a file is erroneously shipped with a base name extension incompatible with its file type, Install may not correctly process the file.

The following table outlines the standard installation processing performed on files of each supported product and file type. Each row of the table lists the following:

- The file type of a shippable file, as specified in the <*type*> field of the file: or head: entry in the extension module index file.
- The appropriate levels for extension modules used to ship the file, as specified via the level: entry of the extension module index file.
- The appropriate base name extensions for the shipped file (if applicable), as specified in the <*tname*> field of the file: or head: entry.
- A summary of the standard processing that Install performs on the file.

Type	Levels	Base Name	Standard Installation Processing Performed
convtab	0, 1	*.v	Installation of database partition conversion table will cause SpectroSERVER and Database partition converter executables to be linked.
cus	0, 1, 2	*.cus	Install invokes custom install scripts after extraction of all VCD records. An extension module should use such a custom install script for special install-time processing of files that it distributes to its associated VCD record.
db	0, 1	*.e	Install imports database import files into the SpectroSERVER database. Install archives and deletes database import files prior to terminating.
doc	0, 1, 2	N/A	No special installation processing.
file	0, 1, 2	N/A	No special processing is performed on general files. Any file that does not need special Install processing should be shipped as this file type.
icondef	0, 1, 2	*.icd	Install uses icondef files to create icons on the user's desktop.
iconimg	0, 1, 2	*.1.pm *.t.pm *.m.p *.ico	Used to ship icon images for use by icons created with icondef files. On Windows, icon image files must be in Windows icon file format and have the .ico extension. Windows icon files are binary files, which can be created with Microsoft Visual Studio.
mmdesc	0, 1, 2	*.mmd	Not currently used in the install but is required by mkmm.
preinst	0,1,2	*.pl	Install executes preinstall modules after start of VCD installation and prior to extraction of files from the VCD.
tool	0, 1, 2	N/A	No special installation processing.
vfile	0, 1	N/A	No special installation processing.

Extension Module Support Files

Extension module support files include shippable files which may be needed in any extension module. They are used mainly to customize the installation of extension modules. Among these files are the following:

- **Part description files:** Provide descriptions for purchasable parts.
- **Preinstall scripts:** Perform preparatory tasks for installing extension modules.
- **Custom install scripts:** Perform customized installation functions for extension modules.

Part Description Files

The mkmm utility requires each shippable extension module that defines a purchasable part to include a part description file (.mmd file). Recall that an extension module defines a purchasable part if it includes a pprep: entry. Therefore, if an index file contains a pprep: entry, a part description file is required. If it is required but not provided, mkmm will not add the extension module to the VCD.

> **NOTE**
> Since the part description file is not used by the installation program, you can create an empty file.

To include a part description file in a extension module, include in your extension module index file a line similar to the following:

```
file: Install mmdesc ? FOO.mmd ? ?
```

> **NOTE**
> In your line entry, replace FOO.mmd with the actual name of the part description file you intend to ship. You should ship a part description file only if the index file contains a pprep: entry.

Preinstall Scripts

Preinstall scripts give an extension module the ability to perform installation functions prior to the extraction of its shippable files.

To include a preinstall script in an extension module, include a line in the extension module index file similar to the following:

```
file: Install preinst ? script.pl ?
```

In your line entry, replace script.pl with the actual name of the preinstall scripts you intend to ship. If your equivalent of script.pl is not in the same directory as the index file, change the target directory field as applicable.

Preinstall Scripts

A preinstall script is a code fragment written in the Perl 5 programming language. As such, preinstall scripts can be used for very complex preinstallation tasks. However, the most common functions performed using preinstallation scripts are fairly simple and include:

- Deletion of obsolete files installed with previous versions of an extension module
- Backup of files that might be overwritten or modified when the extension module is installed
- Suspending running programs that would be adversely affected by simultaneous installation of an extension module

This section shows how to write preinstall modules to delete and back up files. More complex preinstallation functions can be easily programmed by developers skilled in the Perl 5 programming language.

The following example shows a preinstall module for deleting obsolete files installed with previous releases of an extension module. Suppose that in an older version of the fictitious FOO extension module, the following files were installed and that these files are no longer used by the current FOO extension module:

```
SG-Support/CsGIb/FooCom/CsModCnfg.30
SG-Support/CsGIb/FooCom/CsPortCfg.30
SG-Support/CsGIb/FooCom/CsPortCfg.XNT
SG-Support/CsGIb/FooCom/CsPortCfgXNT.30
```

Before installing its own files, the new version of the FOO extension module should clean up these obsolete files.

Suppose further that the old version of FOO extension module included the following configuration file which is still shipped in the current version, and that the user might have edited footool.cfg in the installation area:

```
SS-Tools/footool.cfg
```

If we do not back up footool.cfg before installing the new FOO extension module, the new version of footool.cfg would overwrite the older version, and any user edits to the older version would be lost.

To address these issues, we add preinstall scripts to the FOO extension module. The first preinstall script is associated with the SG product, and is executed prior to extracting SG files for the FOO extension module. This preinstall script would be used to delete the obsolete files:

```
removeFiles(
    "SG-Support/CsGIb/FooCom/CsModCnfg.30",
    "SG-Support/CsGIb/FooCom/CsPortCfg.30",
    "SG-Support/CsGIb/FooCom/CsPortCfg.XNT",
    "SG-Support/CsGIb/FooCom/CsPortCfgXNT.30",
);
```

The removeFiles() function, which is not a standard part of Perl, is provided by CA for use in preinstall scripts. It takes any number of relative file path names as arguments, and it deletes the path name relative to the DX NetOps Spectrum installation directory. If a path specifies a file, the file is deleted. If a path specifies a directory, the directory is recursively deleted. If a path does not exist or is absolute, it is silently ignored.

removeFiles() is designed to minimize the possibility of mistaken file deletion. In preinstall scripts, use removeFiles() in preference to Perl's unlink() function. Be careful not to delete necessary files.

The second preinstall script is associated with the SS product and is executed prior to extracting SS files for the FOO extension module. This preinstall script would be used to save footool.cfg:

```
# SS preinstall script for FOO extension module
# Save the footool configuration file
saveFiles("SS-Tools/footool.cfg");
```

Again, saveFiles() is not native Perl, and is supported for use only in preinstall scripts. saveFiles() takes any number of relative file path names as arguments, and recursively copies the file from the DX NetOps Spectrum installation directory to a save area within the Install-Tools/SAVES directory. The save areas are unique per installation, so that files saved during one installation will not be overwritten by files saved during subsequent installations. Since the save areas will accumulate over many installations, consuming disk space (which may eventually lead to installation or product failure), you should try to avoid saving very large files.

Custom Installation Scripts

The set of installation services provided by the Install utility is fairly complete, handling most extension module installation needs. A few extension modules may need to perform special installation processing not provided by the Install utility, however. To perform such processing, use custom install scripts.

Each extension module has up to three components, called the Install, SpectroSERVER, and OneClick components. The Install component includes shippable files used to install the extension module, and the SpectroSERVER and OneClick components include shippable files associated with the SpectroSERVER and OneClick products, respectively. In some cases, an extension module may not have an SS or OneClick component.

An extension module may include custom install scripts for its SpectroSERVER or OneClick component or for both (Custom scripts should not be created for the Install component). During VCD installation, Install invokes the SpectroSERVER custom scripts when the SpectroSERVER component of the extension module is installed, after the shippable files in the SpectroSERVER component have been extracted from the VCD. Similarly, Install invokes the

OneClick custom scripts when the OneClick component of the extension module is installed, after the shippable files in the OneClick component have been extracted from the VCD. It is rare that an extension module includes two or more custom scripts for a single component; if this is done, the custom scripts for the component are executed in the order in which they are specified in the index file. The custom scripts for a given component perform the special installation processing for that component.

The section that follows provides pointers on when to use custom install scripts and why to do so, gives details on writing these custom install scripts, and explains how to install these scripts. The discussion then concludes with a discussion of predefined Bourne shell variables and functions that you can use in these custom install scripts.

If an index file has a cus_dep: entry, the custom script in the specified management module runs before the custom script in the management module that has the cus_dep: entry. The index file for the specified management module should be included on the same VCD.

Uses of Custom Install Scripts

The Install utility provides standard services for the proper installation of many different types of files. Most extension modules can be properly installed using only these standard services, and these standard services should be used in preference to custom install scripts when possible. However, Install cannot provide standard services for all of the possible specialized installation functions that could conceivably be required by extension modules. When Install does not provide all the services needed to install a given extension module, custom install scripts should be used to help install the extension module. Since custom install scripts have all the capabilities of Bourne shell scripts, they can handle any special installation needs.

The following are examples of installation functions that should be handled using custom scripts:

- **Special installation methods** -- The utility installation services of the Install program are sufficient for most installation needs. However, an extension module will sometimes need to perform special installation tasks unique to itself, such as interacting with third-party software. You can use custom install scripts to perform such specialized installation functions.
- **Modifying contents of installed files to contain information specific to the installation site** -- Most of the time, the contents of a file, as extracted from the VCD, do not need to be modified during installation. There are times, however, when installed files need to contain information specific to the installation site -- such as the DX NetOps Spectrum installation directory or the installation host name -- that cannot be known until installation time. It then becomes necessary to embed that information in the installed files.
  If you must embed installation-specific information in a shippable file, you should use custom install scripts to embed the necessary information.
- **Installing files into system locations** -- In order to protect the integrity of the installation host, Install never extracts files from the VCD to locations outside of the DX NetOps Spectrum installation directory.
  If possible, you should avoid the modification of files in system areas during extension module installation. In rare cases, however, modification to files in system locations may be necessary for proper installation of an extension module. In those cases, you should use custom scripts to modify files in system locations. The custom script should be designed such as to notify the user when files in system locations are modified.
- **Embedding host-specific information into installed files** -- Custom install scripts can be used to embed host-specific information into installed files. Bourne shell commands can be used to embed host-specific information in an installed file.

When Not To Use Custom Install Scripts
When Not To Use Custom Install Scripts

Avoid using custom scripts to move installed files from their originally extracted location to some other location within the DX NetOps Spectrum installation area. The file: and head: index file entries can be modified so that any shippable

file can be extracted from the VCD to any desired location in the DX NetOps Spectrum installation area. DX NetOps Spectrum does not support individually relocatable applications.

Custom Install Scripts

The following example provides information on how to write custom install scripts. In this example, Joe Corporation ships a graphical tool called JoeTool that is used in conjunction with the OneClick product. JoeTool is to be shipped as the OC component of the JoeTool extension module. JoeTool requires a configuration file which, when installed, must include some embedded values (the installation host and the DX NetOps Spectrum installation directory) that cannot be determined until the installation is performed.

The JoeTool developer decides to write a custom install script that will dynamically create the configuration file with the embedded values at installation time. The custom script will be called joetool.cus (all custom scripts must end with the .cus extension). Here are the contents of joetool.cus:

```
# Custom install script for Joe Corporation's JoeTool
# This script creates the JoeTool configuration file
# Obtain the installation host name
# Use the UNIX hostname command
HOSTNAME=`hostname`
# Set JoeTool installation directory
# This uses $IROOT, which is the SPECTRUM installation directory
TOOLDIR="$IROOT/JoeTool"
# If the tool directory doesn't exist, create it
[ -d "$TOOLDIR" ] || mkdir  - p $TOOLDIR
# Set the configuration file name
CONFIGFILE="$TOOLDIR/.joetoolrc"
# If the configuration file already exists, save it
if [ -f "$CONFIGFILE" ] ; then
rm  - f $CONFIGFILE.save
mv $CONFIGFILE $CONFIGFILE.save
echo "JoeTool configurations saved in $CONFIGFILE.save"
fi
# Create the configuration file
# $HOSTNAME and $TOOLDIR will be expanded to their values
cat <<EOF >$CONFIGFILE
##### Start JoeTool configuration file
DISPLAY=$HOSTNAME:0.0
export DISPLAY
ND_PATH=$TOOLDIR/lib
export ND_PATH
PRF=$TOOLDIR/preferences/default.PRF
GIF_BASE=$TOOLDIR/gifs
DATA_TYPE=ATTRIBUTES
GENERATE_HTML=true
DEFAULT_FONT_SIZE=4
##### End JoeTool configuration file
EOF
# A zero exit status prevents custom script failure
exit 0
```

joetool.cus provides an example of a simple custom install script. Custom install scripts to handle more complex installation functions can easily be created by experienced Bourne shell developers.

Shipping Custom Install Scripts

Since custom install scripts perform special processing functions associated with the installation of a particular extension module, as previously noted, they must be present at the installation site in order to perform those functions. To help ensure this, a file: entry must be included in the extension module index file for each custom install script to be included in the extension module.

In the example in the preceding section, joetool.cus is the custom install script for the OneClick component of the JoeTool extension module. We assume that, in the developer's source area, joetool.cus resides in the same directory with the JoeTool extension module index file JoeTool.i. (It is recommended to keep source custom scripts in the same directory with the index file.) In order to have joetool.cus executed when the OneClick component of JoeTool is installed, joetool.cus must be shipped as a OneClick custom install script.

To do this, you would include the following entry in the index file, JoeTool.i:

```
file: OC cus ? joetool.cus ? ?
```

Custom Install Script Predefined Environment Variables

The Install utility predefines a number of environment variables, as listed and explained below, for use in custom scripts. These variables are helpful or necessary for writing proper custom install scripts. In order to maximize the portability of custom install scripts, use these variables rather than other techniques of obtaining information whenever possible.

For instance, use the value of $TARGET_OS to determine the installation platform, rather than the UNIX uname command, which is not available on all platforms.

- **$AWKUTIL**
  Should contain the name of a reliable version of the UNIX awk program on the current platform. Since the native awk utility is unreliable on some supported DX NetOps Spectrum platforms, use of $AWKUTIL is safer.
  For example, use the command:
  ```
  $AWKUTIL '$1 = "file:" { print $5 }' $INDEX
  ```
  rather then:
  ```
  awk '$1 = "file:" { print $5 }' $INDEX
  ```
- **$DEV_NULL**
  Identifies the output sink device for the platform. The $DEV_NULL variable evaluates to /dev/null on platforms and evaluates to nul: on Windows platforms.
  Make absolutely sure that all literal references to /dev/null and nul: are replaced with $DEV_NULL in your custom scripts. Failure to do this can result in both subtle and severe problems if the custom script is invoked on the wrong platform. (This is especially important for experienced UNIX shell programmers, who tend to use /dev/null in scripts without thinking).
- **$IROOT**
  Identifies the DX NetOps Spectrum installation directory. $IROOT is the current directory at the beginning of custom script execution.
  ```
  CUS_DOCS=$IROOT/OnlineDocs/helpdoc/helpdoc
  ```
- **$OC_PORT**
  The $OC_PORT variable is the Tomcat port number that can be used to create the URL for DX NetOps Spectrum.
  ```
  http://localhost:$OC_PORT/spectrum
  ```
- **$PATH**
  Is the Bourne shell command search path. Install sets $PATH to allow access to the following tools:
  - Standard installation tools ($IROOT/Install-Tools)
  - SpectroSERVER tools ($IROOT/SS-Tools)
  - Standard UNIX commands on the installation host
- **$TARGET_OS**
  Contains information about the platform on which the installation is running. Possible values are:

  - Linux
  - NT
- For example:

```
if [ "$TARGET_OS" = "NT" ]
then
    ....
fi
```

- **$TOMCAT_ROOT**
  Identifies the DX NetOps Spectrum Tomcat installation directory.

```
CUS_WEB=$TOMCAT_ROOT/webapps/spectrum
```

Non-Web-Based, Graphical Application Implementation

When writing non-web-based graphical applications designed to run within the DX NetOps Spectrum environment, you should adhere to the following recommendations:

- On Linux, DX NetOps Spectrum graphical applications should use the X windowing system, Motif window manager, and K Desktop Environment (KDE).
- On Windows, DX NetOps Spectrum graphical applications should use the Windows windowing system (included in the Windows operating system).

Icon Definition and Icon Image Files

For some external applications, (usually graphical applications), it is convenient to be able to launch the application from a desktop icon. For these applications, it is necessary to install a desktop icon and to define how the application is to be launched when the icon is opened. Icon definition files and icon image files are used to ship and install icons for desktop-launchable applications.

Each line in an icon definition file is of the form:

```
<label> <value>
```

where <*label*> is an icon definition file label from the table below, and <value> is the associated value. A label may carry the suffix .NT to indicate that the label applies only on Windows. For example, an imagefile.NT: entry is an imagefile: entry that applies on the Windows platform only.

Use of icon definition files and icon image files for desktop-launchable DX NetOps Spectrum applications currently is not supported on Linux.

The following table lists valid labels in icon definition files, as well as the interpretation of their associated values.

<*label*>	**Windows**	**Interpretation of <*value*>**
program:	Mandatory	The absolute or relative path name of the application program to be launched when the icon is opened. If the path is relative, it is taken relative to the DX NetOps Spectrum installation directory.
args:	Optional	The arguments to pass to the program when it is launched. If <*value*> is omitted, no arguments are passed.
workingdir:	Optional	The working directory when program is launched. If <*value*> is omitted, working directory defaults to “.” (current directory).

imagefile:	Optional	The icon image file. On Windows, <*value*> must be a path name, taken relative to the DX NetOps Spectrum installation directory if the path is not an absolute path. The image file may be a Windows icon file (.ico file) or a DLL containing an icon library (see the imageordinal: entry in this table). <*value*> may be omitted if the icon is included in the application program.
imageordinal:	Optional	On Windows, if the specified image file is a DLL containing an icon library, the image ordinal gives the index of the image file within the DLL.
iconfolder:	Mandatory	The desktop menu folder that will contain the icon.
iconname:	Mandatory	The descriptive name of the icon. This name appears in the menu pick for the icon and/or immediately beneath the icon on the desktop
description:	Optional	The description of the icon. On Windows, this name appears when the cursor hovers over the icon.
windowstate:	Optional	The initial state of the window launching the application. On Windows, <*value*> may be any one of the following: SW_SHOWNORMAL SW_SHOWMINIMIZED SW_SHOWMAXIMIZED SW_SHOWMINNOACTIVE SW_HIDE If <*value*> is omitted, the initial state of the launch window is SW_SHOWNORMAL.

The following is a icon definition file named appl.icd, for a hypothetical DX NetOps Spectrum application named Application. Assume that this application is associated with the OneClick application:

```
program: %IROOT%/bin/Application
args:
workingdir: .
imagefile.NT: %IROOT%/Install-Tools/icons/appl.ico
imagefile.windows: appl
iconfolder: CA
iconname: My Application
description: My <sp> Application
```

The program: entry specifies that the icon will invoke the application <*$SPECROOT*>/bin/Application when opened (clicked) on either Windows. The args: entry specifies that Application will be invoked with no arguments. The workingdir: entry specifies that Application will be invoked from the current directory (“.”) at the time the icon is opened.

On Windows, the imagefile.NT: entry specifies that the icon image file used to display the icon is to be

<*$SPECROOT*>/Install-Tools/icons/appl.ico (Note that on Windows, DX NetOps Spectrum icon image files are normally installed into the <*$SPECROOT*>/Install-Tools/icons directory). This file must be in the Windows icon file format and have an .ico extension. Windows icon files are binary files, which can be created by use of the Microsoft Visual Studio program.

If the icon had been compiled into the application, it would have been unnecessary to ship the icon image file on Windows, and the imagefile.NT: entry could have been omitted. If the icon had part of a Windows icon library, and imageordinal: entry would have been included to specify the icon's position within the library.

The iconfolder: entry specifies that, on Windows, the Application icon will appear in the CA folder under Start menu > Programs.

Shipping Icon Definition and Icon Image Files

The following index file entries might be used to ship the icon definition file and icon image files described in the scenario outlined in the previous section. These entries assume that all icon definition files and icon image files are in the same directory as the index file.

```
file: OC icondef ? appl.icd ? ?
file: OC iconimg ? appl.ico ? ? plat=intel
file: OC iconimg ? appl.l.pm ? ? plat=sun4c
file: OC iconimg ? appl.m.pm ? ? plat=sun4c
file: OC iconimg ? appl.t.pm ? ? plat=sun4c
```

DX NetOps Spectrum Extension Integration Developer Toolkit (SEI Toolkit) Troubleshooting

This section describes problems that can occur while creating the integration module. You can follow the solutions that are provided to resolve the integration errors.

Unable to Archive the Event045a0025.txt file

Symptom:

mkmm script failed with the following error message:

```
Unable to archive file /opt/spectrum/SG-Support/CsEvFormat/Event045a0025.txt
```

Solution:

You must place all the customer defined events, Pcause, and the event table files under the $*SPECROOT*\custom folder.

Web Services API Reference

About the DX NetOps Spectrum Web Services API

The DX NetOps Spectrum Web Services API provides a RESTful HTTP Web Service interface in DX NetOps Spectrum. This distributed API provides an HTTP-based integration point to the DX NetOps Spectrum data model, allowing web-centric read/write access to devices, models, relationships, attributes, actions and alarms. The DX NetOps Spectrum Web Services API is a core DX NetOps Spectrum component and is installed along with the core DX NetOps Spectrum product.

Using the DX NetOps Spectrum Web Services API, DX NetOps Spectrum data can be accessed directly from a browser or integrated into your own application. Because of the inherent standards in the REST architecture, the DX NetOps Spectrum Web Services API makes the DX NetOps Spectrum data model accessible to many different external

development environments and methods. The DX NetOps Spectrum Web Services API can be used with any language that knows how to manage HTTP integration and provides a lightweight alternative to SOAP services.

By using the DX NetOps Spectrum Web Services API, you can take advantage of functionality provided by the OneClick server, such as using its search infrastructure to more easily find groups of models. You can also perform many standard DX NetOps Spectrum functions for a single SpectroSERVER or in a distributed SpectroSERVER (DSS) environment, such as the following:

- Access devices, models, relationships, attributes, actions and alarms
- Manage devices, ports, containers, services, and links
- Read, update, and clear alarms
- Manage subscriptions and notifications

You can use the DX NetOps Spectrum Web Services API to perform these functions and more, similar to those provided by other DX NetOps Spectrum API tools such as the Command Line Interface (CLI), Modeling Gateway, AlarmNotifier, Southbound Gateway, or the SpectroSERVER Object Request Broker (SSORB) interface.

REST Architecture

Representational State Transfer (REST) is a software architectural style that promotes standardized behaviors between interacting elements, or clients and servers. Conforming to the REST constraints is referred to as being RESTful.

The REST architecture is a lightweight HTTP/HTTPS-based approach for SOAPless Web Services using create (POST), read (GET), update (PUT), and delete (DELETE) operations, or verbs.

RESTful architecture and applications are stateless, which means that no client context information is stored between requests. Each request contains all the information necessary to service the request.

The DX NetOps Spectrum Web Services API supports the REST architecture.

Supported Technologies

The DX NetOps Spectrum Web Services API provides a flexible, lightweight RESTful approach for integration of DX NetOps Spectrum data into your application. As an alternative to CORBA integration, the DX NetOps Spectrum Web Services API is language-independent and can be integrated into any development platform that supports HTTP integration, such as:

- Java
- Perl
- Ruby on Rails
- .net

The DX NetOps Spectrum Web Services API provides an HTTP/HTTPS interface, is both HTTP and XML/JSON-based, and provides Java Beans for client development.

Performance Considerations

The DX NetOps Spectrum Web Services API is a client, very similar to OneClick. Because the DX NetOps Spectrum Web Services API reuses much of the OneClick server code, performance of a single API client should generally mimic the performance of the comparable OneClick operation. The DX NetOps Spectrum Web Services API should not alter SpectroSERVER performance except through normal request load.

WARNING

The following can cause performance issues:

- Because data is cached in the OneClick server, large data requests across multiple clients can cause performance degradation.
- You may experience performance degradation and even SpectroSERVER crashes, if you create and execute new and poorly performing searches through the DX NetOps Spectrum Web Services API.
  For more information, See Troubleshooting the Web Services API Issues.

Limitations

The following should be considered when using the DX NetOps Spectrum Web Services API:

- The DX NetOps Spectrum Web Services API does not provide interfaces for the modeling catalog where data is generally static.
- While an HTML or browser application can be built on top of the DX NetOps Spectrum Web Services API, the DX NetOps Spectrum Web Services API itself does not provide HTML.
- The DX NetOps Spectrum Web Services API accepts XML only for input and produces XML or JSON for output. If you choose JSON as output, you may need to parse it out and produce XML for the next request.

NOTE
The DX NetOps Spectrum Web Services API can be used for simple CLI ad hoc commands and as an alternative to complex CLI scripting.

WARNING
The DX NetOps Spectrum Web Services API is a powerful tool. It does not provide the safeguards that OneClick does, especially related to modeling. It should be used only by DX NetOps Spectrum administrators who understand the potentially harmful effects on a network modeling scheme of haphazardly creating and destroying models and modifying model attributes.

Improve DX NetOps Spectrum REST API Capability (Swagger)

From 10.4.1, you can access DX NetOps Spectrum API documentation using Swagger. The Swagger UI lets you visualize and interact with DX NetOps Spectrum REST API by providing visual documentation.

You now have Swagger documentation and endpoints to work with DX NetOps Spectrum RESTful web services. The REST API endpoints are provided within a self-documenting framework that lets you try the methods and see the generated responses. All methods are grouped by resource type and can be displayed in a single window. Use the **Show/Hide**, **List Operations**, and **Expand Operations** options to expand and collapse the resources and methods.

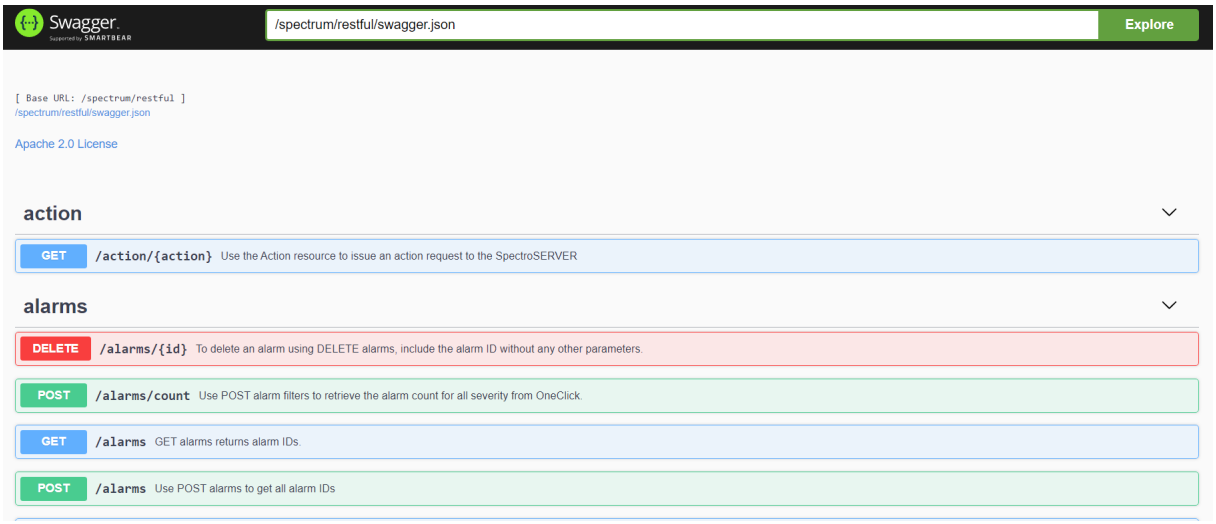
The methods are color-coded to make it easier to distinguish between the different methods. To the right of each method is a brief description of the action that is performed by the method. Documentation is provided for request parameters. Sample model schema is available for methods requiring payload in the request. Use the provided model schema to populate the body field, and then replace the default values with desired settings. Use the **Try it out!** Button to send the request and see the generated response in real-time.

The **Response Messages** section for each method provides a documented list of response codes. Use this information to help you determine a corrective action if an error condition was encountered. The response that is returned for each request appears in the method drop-down window.

Accessing the API Documentation

Click the **API Documentation** menu link in the OneClick Web interface to access the REST API.

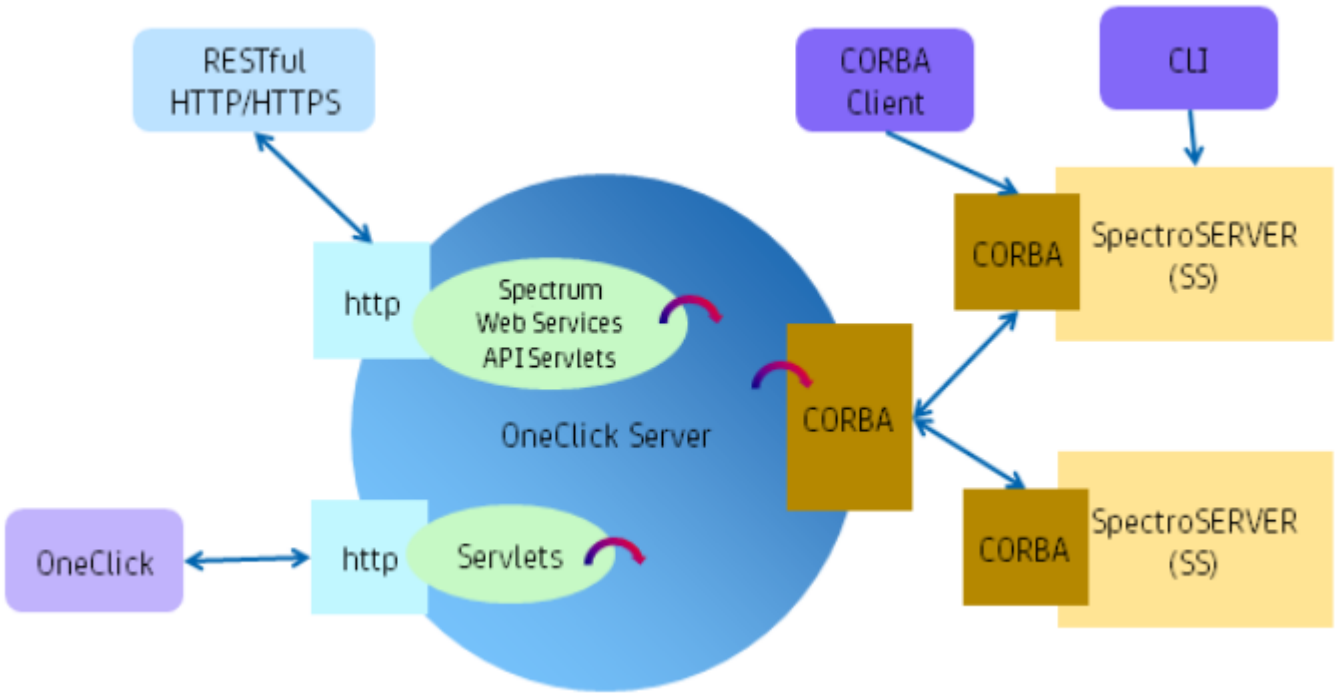
The following screenshot displays the Swagger interface. This interface displays some of the RESTful web services that DX NetOps Spectrum provides:



For more information on using Swagger for REST API, see Swagger documentation.

Web Services API and OneClick

The DX NetOps Spectrum Web Services API is hosted by and sits on top of the OneClick server. As shown in the following diagram, the DX NetOps Spectrum Web Services API servlets provide the entry point to the OneClick server:



This approach allows full access to OneClick server functions, essentially using the same code as the OneClick server. You can take advantage of OneClick features such as user authentication, search capability, and defined alarm filters. Throttled reads provide bulk query efficiency, and load-balanced OneClick servers provide high availability. The DX NetOps Spectrum Web Services API is fully supported in a distributed SpectroSERVER environment.

As with all RESTful services, the DX NetOps Spectrum Web Services API requests are stateless, requiring each request to include all necessary information to perform the operation.

How to Use the Web Services API

The following describes how to use the DX NetOps Spectrum Web Services API:

1. Generate a request to the OneClick server. The request can be in the form of a URL or imbedded in XML content. The OneClick server processes the request and returns a response.
2. Review the response, which contains the results of the request in the form of an XML or JSON document, depending on the client configuration.
3. Using information returned by the response, such as throttled results or other request-specific information, you can create follow-up requests.

Both requests and responses are regulated by defined schemas.

Web Services API Files

DX NetOps Spectrum Web Services API files are located in <*$SPECROOT*>/RestfulExamples, which includes the following folders:

- **lib**
  Contains all .jar files necessary for Java client development.
- **src**
  Contains the following folders:
  - **test**
    Contains example Java code that demonstrates basic functionality, JAXB beans, and hard-coded XML. An Eclipse .project and .classpath are also provided. For more information about provided examples, see Java Code Examples.
  - **xsd**
    Contains xsd files. For more information, see XML Schema Definitions.
- **xml**
  Contains sample XML for GET Tunneling requests. For more information about provided examples, see XML Examples.

  NOTE
  For more about GET Tunneling, see Simple and Complex Requests.

Requests

To use the DX NetOps Spectrum Web Services API, you generate a request to the OneClick server for which you receive results in response. The request can be in the form of a URL or imbedded in XML content.

RESTful Web Service URLs are noun-oriented, and the HTTP operation itself represents the verb. The following is the base URL when generating a request using the DX NetOps Spectrum Web Services API:

```
http://<hostname><:portnumber>/spectrum/restful/<request>
```

- ***request***
  Specifies the DX NetOps Spectrum Web Services API RESTful resource. Any request that requires a body will use the types defined in the Request.xsd.

NOTE
Because RESTful architecture and applications are stateless, no client context is stored between requests. Each request contains all the information necessary to service the request in the URL or XML content.

Request Types

There are two types of requests used in the DX NetOps Spectrum Web Services API. Both request types offer the same performance.

- **URL Requests**
  Used when a resource targets a specific model. For example, DELETE <*mh*> deletes a specific model. This type of request works well for simple requests with few parameters, does not require body text, and can be used as a browser URL.

  NOTE
  All URL requests have GET Tunneling equivalents.
- **GET Tunneling Requests**
  Used when a resource targets multiple models using complex parameters. An HTTP GET command can only support a limited number of parameters in a URL and does not support a body. The GET Tunneling operation tunnels a GET request through a POST body, allowing complex requests to be created. The XML body can contain rich filters by using new and existing search criteria and pre-defined alarm filters. Semantics of GET Tunneling requests are defined by Request.xsd and Filter.xsd schemas and can be created manually or generated from request beans.

General URL Parameters

The following are general parameters that are supported for some of the resources, depending on application. In general, the order of URL parameters in a list does not matter. Any exceptions are noted.

WARNING
Parameters are preceded by either ? or &. In accordance with standard HTTP syntax, the first parameter specified must be preceded by ?, and any subsequent parameters must be preceded by &. All of the following parameters are shown preceded by &. Use the appropriate prefix depending on the location of the parameter in the list.

- **&attr=<*attr_ID*>&val=<*num*>**
  Specifies attribute value pairs, which are used to obtain specific attribute values for the resource being queried. For example, if you GET alarms, you get the alarm IDs only by default. However, if you retrieve alarms with specified attributes you get the alarm IDs with their specific attribute values.
  When attribute value pairs are specified, especially when updating attribute values, attribute ID and values must occur in sequence for proper binding of the ID and value, as in the following:
  ```
  ?attr=0x10062&val=text&attr=0x10001&val=12334
  ```
  This pairing tells the RESTful servlet that an attr ID will be followed by the desired value for that ID when updating the value of the attribute.
  Multiple attribute-value pairs can be specified.
- **&landscape=<*landscape_handle*>**
  Specifies which landscapes to include in query. Multiple landscape parameters can be specified.
- **&mh=<*model_handle*>**
  Specifies the model handle.
- **&throttlesize=<*num*>**
  Specifies the number of items from the result set to retrieve per response. If the throttlesize value is less than the total number of items in the result set, the results contain a "next" link. This link can be used to retrieve more results. See Using Throttle and Next.

NOTE
Not all resources support the throttlesize parameter.

Security

User Security in OneClick

All requests using the DX NetOps Spectrum Web Services API respect DX NetOps Spectrum user model-based security.

WARNING
The DX NetOps Spectrum Web Services API is a powerful tool. It does not provide the safeguards that OneClick does, especially related to modeling. It should be used only by DX NetOps Spectrum administrators with strong knowledge of the DX NetOps Spectrum data model.

Secure Connections

The DX NetOps Spectrum Web Services API supports secure connections to the OneClick server. To use secure connections, the OneClick server must be configured for Secure Sockets Layer (SSL).

NOTE
For information about configuring OneClick for SSL, see the Administration section.

Responses

Responses are in the form of an XML or JSON document, as determined by the client's accept header value/MIME type. This flexible output makes the output data easy to integrate into client development. Responses are defined by Response.xsd schema.

NOTE
The DX NetOps Spectrum Web Services API supports both application/xml and application/json MIME types in the accept header. If you choose JSON as output, you may need to parse it out and produce XML for the next request.

The following describes elements of a response output:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<resource-response-list xmlns="http://www.ca.com/spectrum/restful/schema/response" error="<msg>"
 throttle="<num>" total-resource="<num>">
  <resource-responses>
...
  </resource-responses>
<link rel="next" href="http://<hostname><:portnumber>/spectrum/restful/alarms?
id=<result_set_ID>&start=3&throttlesize=<num>" type="application/xml" />
</resource-response-list>
```

- **error="<*msg*>"**
  Indicates the overall success or failure of the operation. If there is a failure, additional information is provided. "EndOfResults" indicates that all items in the result set have been returned.
- **throttle="<*num*>"**
  Indicates the number of items returned in this response, as specified by the throttlesize parameter on the request. If the throttle number is less than the total results found by the server, a "next" link is provided. If the complete set of data is returned, there is no "next" link.
- **total-*resource*="<*num*>"**

Indicates the total number of items in the result set. *Resource* varies depending on the type of request (for example, alarms, landscape, model, models).

- **link rel="next" href="<*URL*>"**

  NOTE
  The "next" link appears only if additional items remain in the result set.

  Indicates that additional items remain in the result set and provides the URL to obtain them according to the throttle value. This link conforms to the HATEOAS (Hypermedia as the Engine of Application State) constraints and can be used in a client or a browser to issue the "next" request. By using this link, the full cost of the original call is avoided. The location is cached on the OneClick server and expires after ten minutes of inactivity. After expiration, the URL returns the string "The cursor is no longer valid".

  NOTE
  Each RESTful call has its own cache.

  The following parameters appear in the generated URL:

  – **id=<*result_set_ID*>**
  Specifies the location of the result set on the OneClick server. The data expires after ten minutes of inactivity.
  – **start=<*element_num*>**
  Indicates the start position within the result set.
  – **throttlesize**
  Indicates the number of items to return in the next response.

  NOTE
  The format of the parameters within the generated link may vary depending on your environment. For example, "&" may be generated as "&" This is due to differences between XML, HTTP, and JSON formatting standards.

RESTful Resources (Nouns)

The DX NetOps Spectrum Web Services API provides the URLs for the appropriate RESTful HTTP operations against the DX NetOps Spectrum data model nouns, as displayed in the following table. All nouns are lowercase.

Resource	Description	POST	GET	PUT	DELETE
action	Used to send actions		yes		
alarms	Read, update, and delete alarms	GET Tunneling	yes	yes	yes
associations	Create, read, update, and delete associations	yes	yes		yes
attribute	Reads values for enumerated attribute values		yes		
connectivity	Reads connectivity information for a specified IP address		yes		
count	Retrieves total alarm counts of critical, major, and minor severity		yes		
devices	Reads devices and attributes		yes		
filters	Retrieves names of all the alarm filters, and also retrieves the alarms based on alarm filters	yes	yes		
landscapes	Reads landscape information		yes		
model	Creates, reads, and updates attributes and deletes models	yes	yes	yes	yes
models	Reads and writes models attributes	GET Tunneling	yes*	yes	

Events	Creates one or more events on single or multiple models	yes			
subscription	Creates a request for change notification	yes	yes		yes

* GET can only be performed against the result set created by POST (GET Tunneling).

action

Use the Action resource to issue an action request to the SpectroSERVER. An internal action code determines which operation is performed by the action.

- **Base URL**
  `http://<hostname><:portnumber>/spectrum/restful/action`

GET action

GET action issues an action request to the SpectroSERVER. Initiated as an HTTP GET, the action performed on the SpectroSERVER, including modifying data, is specified by an action code. All actions return results, which differ by the requested action.

- **URL**
  `http://<hostname><:portnumber>/spectrum/restful/action/<action_code>[?mh=<model_handle>]`
  `[&attr=<attr_ID>&val=<num>][&throttlesize=<num>]`
- **HTTP Method**
  GET
- **Body**
  None
- **Body Content**
  Not used
- **Header**
  application/xml, application/json
- **Output**
  XML or JSON listing. Content differs depending on requested action.

URL Parameters

- ***action_code***
  Specifies the action to perform on the SpectroSERVER. The following table contains some common actions.

Action	Target Model	Description
0x1000e	device	Reconfigures a model (reconfig)
0x480003	watch	Activates a watch (activate)
0x480004	watch	Deactivates a watch (deactivate)
0x210008	device	Reconfigures application models on Cisco and Wellfleet devices (reconfigure_apps)
0x100a2	VNM	Updates the SpectroSERVER with changes to EventDisp and AlertMap files (reload_event_disp)
0x100f6	global collection	Updates membership for periodic search-based global collections
0x10274	VNM	Turns on EMS debugging

0x10275	VNM	Turns off EMS debugging
0x10301	VNM or global collection	Turns on Search Debug for all searches if directed to the VNM model or for a single global collection if directed to a global collection
0x10300	VNM or global collection	Turns off Search Debug for all searches if directed to the VNM model or for a single global collection if directed to a global collection
0xbc614d	RFC2790App	Toggles debug for File System Monitoring
0xbd6146	RFC2790App	Toggles debug for Process Monitoring

- **&mh=<*model_handle*>**
  (Optional) Specifies the target of the action.
- **&attr=<*attr_ID*>&val=<*num*>**
  (Optional) Specifies attribute values as input to the action. Multiple attribute-value pairs can be specified.
- **&throttlesize=<*num*>**
  (Optional) Specifies a throttle size.

Example

The following URL initiates the 'FIND_DEV_MODELS_BY_IP' action (0x10093) against the SearchManager model (0x400018):

```
http://<hostname><:portnumber>/spectrum/restful/action/0x10093?
mh=0x400018&attr=0&val=172.22.96.11&attr=1&val=172.22.96.5
```

NOTE

- The SearchManager model handle will differ from SpectroSERVER to SpectroSERVER.
- You can also find device models by IP by embedding the correct locater search in the POST body of a GET models.

Using Alarm Resources

Use the Alarms resource to read, modify, and delete DX NetOps alarms. Asynchronous responses are supported.

- **Base URL**
  ```
  http://<hostname><:portnumber>/spectrum/restful/alarms
  ```

POST alarms (GET Tunneling)

Use POST alarms (GET Tunneling) to get all alarm IDs. POSTing an alarms request allows you to provide an XML document that specifies the alarms to be retrieved. The document can specify specific alarms by ID or by a model specification. A model specification can be a list of model handles, XML search criteria or a reference to existing search criteria on the OneClick server. It is also possible to GET alarms that satisfy a specific alarm attribute filter. An attribute filter is a subset of the search criteria.

WARNING
This POST is a form of GET Tunneling. You cannot create an alarm with POST alarms. DX NetOps alarms are created in response to events, as defined by the Event Disposition files.

- **URL**
  ```
  http://<hostname><:portnumber>/spectrum/restful/alarms[?symptoms=no][?symptoms=yes]
  ```
- **HTTP Method**

POST

- **Body**
  Examples are provided in <*$SPECROOT*>/RestfulExamples/xml/Alarms. These XML files provide examples of properly formed XML for different purposes. Use these examples as a basis for your own XML:
  – GetAlarmsByModelHandles.xml
  – GetAlarmsBySearchCriteria.xml
  – GetAlarmsForAllDevices.xml
  – GetAlarmsByAlarmIDs.xml
  – GetAlarmsByAttributeFilter.xml
  The following example is a simple xml criteria to retrieve alarm ids and their associated details like Model Type, Model Type of Alarmed Model, Security String, Condition, Model Name, Model Class:

```
<?xml version="1.0" encoding="UTF-8"?>
 <rs:alarm-request throttlesize="0"-->
 xmlns:rs="http://www.ca.com/spectrum/restful/schema/request"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.ca.com/spectrum/restful/schema/request ../../../
xsd/Request.xsd ">
 <rs:requested-attribute id="0x10000"/> <!--Model Type Name-->
 <rs:requested-attribute id="0x10001"/> <!--Model Type of Alarmed Model-->
 <rs:requested-attribute id="0x10009"/> <!--Security String-->
 <rs:requested-attribute id="0x1000a"/> <!--Condition-->
 <rs:requested-attribute id="0x1006e"/> <!--Model Name-->
 <rs:requested-attribute id="0x11ee8"/> <!--Model Class-->
<rs:landscape id="0x100000"/>
</rs:alarm-request>

<!--If you include landscape attribute(s) in the xml criteria, alarms of all the
 landscapes that are included are retrieved.-->
```

- **Body Content**
  application/xml
- **Header**
  application/xml, application/json
- **Output**
  XML or JSON listing all alarms in the SpectroSERVER or distributed SpectroSERVER
  Example
  The following is an example call:

```
Java GenericPoster noun=Alarms file=resources/xml/Alarms/GetAlarmsBySearch.xml server=localhost username=jdoe
 password=spectrum port=8080
```

URL Parameters

- **<?symptoms=no>**
  Use this parameter to retrieve all the alarm ids excluding the symptom alarm ids.
  Based on the example xml body, the following output is returned for "http://<hostname><:portnumber>/spectrum/restful/alarms?symptoms=no":

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<alarm-response-list error="EndOfResults" throttle="18" total-alarms="18">
 <alarm-responses>
```

```
 <alarm id="55d30306-7d7b-1000-02a9-6c8814e75b4c">
<attribute id="0x10000">WA_Link</attribute>
<attribute id="0x10001">0x102e2</attribute>
<attribute id="0x10009" />
<attribute id="0x1000a">3</attribute>
<attribute id="0x1006e">10.10.10.0</attribute>
<attribute id="0x11ee8">7</attribute>
 </alarm>
  </alarm>
 </alarm-responses>
</alarm-response-list>
```

- **<?symptoms=yes>**
  Use this parameter to retrieve all the alarm ids including the symptom alarm ids.
  Based on the example xml body, the following output is returned for "http://<hostname><:portnumber>/spectrum/restful/alarms?symptoms=yes":

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<alarm-response-list error="EndOfResults" throttle="20" total-alarms="20">
 <alarm-responses>
 <alarm id="55d30306-7d7b-1000-02a9-6c8814e75b4c">
<attribute id="0x10000">WA_Link</attribute>
<attribute id="0x10001">0x102e2</attribute>
<attribute id="0x10009" />
<attribute id="0x1000a">3</attribute>
<attribute id="0x1006e">10.10.10.0</attribute>
<attribute id="0x11ee8">7</attribute>
 </alarm>
 </alarm-responses>
</alarm-response-list>
```

- **?lasthour=<*num*>**
  (optional) Use this parameter to retrieve alarm ids generated in the last <*num*> of hours, where <*num*> of hours. For example, to retrieve the alarm ids generated in the last two hours, use the following url:

```
http://<hostname><portnumber>/spectrum/restful/alarm?lasthour=2
<?xml version="1.0" encoding="UTF-8"?>
<rs:alarm-request xmlns:rs="http://www.ca.com/spectrum/restful/schema/request" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance" throttlesize="100" xsi:schemaLocation="http://www.ca.com/spectrum/
restful/schema/request ../../../xsd/Request.xsd ">
   <rs:requested-attribute id="0x11f56" />
   <rs:requested-attribute id="0x12b4c" />
   <rs:requested-attribute id="0x1006e" />
</rs:alarm-request
```

GET alarms

GET alarms returns alarm IDs. You can retrieve specific alarm attributes by using the &attr=<*attr_id*> parameter.

- **URL**

```
http://<hostname><:portnumber>/spectrum/restful/alarms[?attr=<attr_ID>][&landscape=<landscape_handle>]
[&throttlesize=<num>]
```

- **HTTP Method**
  GET
- **Body**
  None
- **Body Content**
  Not used
- **Header**
  application/xml, application/json
- **Output**
  XML or JSON listing alarms in the SpectroSERVER or distributed SpectroSERVER
  The following is an example taken from our environment for retrieving only alarm ids:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<alarm-response-list error="EndOfResults" throttle="20" total-alarms="20">
 <alarm-responses>
<alarm id="55d30306-7d7b-1000-02a9-6c8814e75b4c" />
<alarm id="55d2d38b-47a6-1000-02a9-6c8814e75b4c" /
>
 </alarm-responses>
</alarm-response-list>
```

This is an example based on our environment.

URL Parameters

- **&attr=<*attr_ID*>**
  (Optional) Specifies the requested attributes. Multiple attribute parameters can be specified.

  > **NOTE**
  > There are many attributes on a model; for best performance you should limit attribute selection to attributes of interest.
- **&landscape=<*landscape_handle*>**
  (Optional) Filters which landscapes are queried. Multiple landscape parameters can be specified.
- **&throttlesize=<*num*>**
  (Optional) Specifies a throttle size.

```
http://<hostname><portnumber>/spectrum/restful/alarm
```

The following output is generated:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<alarm-response-list error="EndOfResults" throttle="6" total-alarms="6">
 <alarm-responses>
<alarm id="55d30306-7d7b-1000-02a9-6c8814e75b4c" />
<alarm id="55d2d38b-47a6-1000-02a9-6c8814e75b4c" /
>
 </alarm-responses>
</alarm-response-list>
```

This is an example based on our environment.

GET alarm filters

Use GET alarm filters to retrieve the alarm filters defined in OneClick.

- **URL**
  `http://<hostname><:portnumber>/spectrum/restful/alarms/filters`
- **HTTP Method**
  GET
- **Body**
  None
- **Body Content**
  Not Used
- **Header**
  application/xml, application/json
- **Output**
  The following xml output containing the alarm filter names is returned:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
- <alarmfilters-response xmlns="http://www.ca.com/spectrum/restful/schema/response">
<alarmfilter>LAN_alarms</alarmfilter>
<alarmfilter>Chassis_Alarms</alarmfilter>
<alarmfilter>core_router_alarms</alarmfilter>
</alarmfilters-response>
```

This is an example based on our environment.

POST alarm count

Use POST alarm filters to retrieve the alarm count for all severity from OneClick.

- **URL**
  `http://<hostname><portnumber>/spectrum/restful/alarms/count`
- **HTTP Method**
  POST
- **Body**
  You can use an extensive xml criteria to retrieve the required alarm count. For example, use the following body:

```
<?xml version="1.0" encoding="UTF-8"?>
<rs:alarms-count-request xmlns:rs="http://www.ca.com/spectrum/restful/schema/
request"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.ca.com/spectrum/restful/schema/request
../../../xsd/Request.xsd ">

<rs:alarm-filter name="def"/>

</rs:alarms-count-request>
```

- **Body Content**
  application/xml
- **Header**
  application/xml, application/json
- **Output**

The following xml output containing the alarm count for all severity is returned:

```
<alarmcount-response>
    <critical>4</critical>
    <major>21</major>
    <minor>4</minor>
</alarmcount-response>
```

PUT alarms

Use PUT alarms to update alarm attributes.

- **URL**
  `http://<hostname><:portnumber>/spectrum/restful/alarms/<alarm_id>?attr=<attr_ID>&val=<num>`
- **HTTP Method**
  PUT
- **Body**
  None
- **Body Content**
  Not Used
- **Header**
  application/xml, application/json
- **Output**
  XML or JSON listing indicating success or failure

URL Parameters

- ***alarm_ID***
  Specifies the alarm ID.
- **&attr=<*attr_ID*>&val=<*num*>**
  Specifies the attributes and values. Multiple attribute-value pairs can be specified.

DELETE alarms

To delete an alarm using DELETE alarms, include the alarm ID without any other parameters.

- **URL**
  `http://<hostname><:portnumber>/spectrum/restful/alarms/<alarm_id>`
- **HTTP Method**
  DELETE
- **Body**
  None
- **Body Content**
  Not Used
- **Header**
  application/xml, application/json
- **Output**
  XML or JSON listing indicating success or failure

URL Parameters

- ***alarm_ID***
  Specifies the alarm to delete.

associations

Use the Associations/Relation resource to create, read, or delete associations.

> **NOTE**
> Associations cannot be updated. To modify an association, delete the existing association and create a new one. Association operations are restricted to one SpectroSERVER at a time.

- **Base URL**

```
http://<hostname><:portnumber>/spectrum/restful/associations/relation
```

> **NOTE**
> The Associations noun is a pass-through noun. You must also specify the Relation noun to create, read, or delete associations.

POST associations

POST associations creates an association. A specific relation is created between two models. Only models on the same SpectroSERVER can be associated.

- **URL**

```
http://<hostname><:portnumber>/spectrum/restful/associations/relation/<rel_handle>/leftmodel/<LMhandle>/
rightmodel/<RMhandle>
```

- **HTTP Method**
  POST
- **Body**
  None
- **Body Content**
  Not Used
- **Header**
  application/xml, application/json
- **Output**
  XML or JSON listing association information

URL Parameters

- ***rel_handle***
  Specifies the relation handle.
- ***LMhandle***
  Specifies the handle of the left model in the relation.
- ***RMhandle***
  Specifies the handle of the right model in the relation.

> **NOTE**
> To get the left or right model handle, you can use the CLI show associations command, as follows:

```
show associations mh=<model_handle_of_device/port>
```

> **NOTE**
> For more information about this command, see the Command Line Interface section.

GET associations

GET associations returns associations for a specific relation and model. Associations are read based on the side of the association that the model handle is on.

- **URL**

```
http://<hostname><:portnumber>/spectrum/restful/associations/relation/<rel_handle>/model/<model_handle>?
side=[left|right]
```

- **HTTP Method**
  GET
- **Body**
  None
- **Body Content**
  Not Used
- **Header**
  application/xml, application/json
- **Output**
  XML or JSON listing association information

URL Parameters

- ***rel_handle***
  Specifies the relation handle.
- ***model_handle***
  Specifies the handle of the model in the relation.
- **side=[left|right]**
  Specifies the side of the association that the model handle is on.

DELETE associations

DELETE associations removes associations by deleting a specific relation between two models.

- **URL**
  ```
  http://<hostname><:portnumber>/spectrum/restful/associations/relation/<rel_handle>/leftmodel/<LMhandle>/
  rightmodel/<RMhandle>
  ```
- **HTTP Method**
  DELETE
- **Body**
  None
- **Body Content**
  Not Used
- **Header**
  application/xml, application/json
- **Output**
  XML or JSON listing indicating success or failure

URL Parameters

- ***rel_handle***
  Specifies the relation handle.
- ***LMhandle***
  Specifies the handle of the left model in the relation.
- ***RMhandle***
  Specifies the handle of the right model in the relation.

attribute

Use the Attribute resource to retrieve strings for enumerated attributes.

- **Base URL**

```
http://<hostname><:portnumber>/spectrum/restful/attribute
```

GET attribute

GET attribute returns XML describing the enumerations for the specified attribute.

- **URL**
  ```
  http://<hostname><:portnumber>/spectrum/restful/attribute/<attr_ID>/enums
  ```
- **HTTP Method**
  GET
- **Body**
  None
- **Body Content**
  application/xml, application
- **Header**
  application/xml, application/json
- **Output**
  XML describing the enumerations for the specified attribute

URL Parameters

- ***attr_ID***
  Specifies the attribute ID.

  NOTE
  Do not use external attributes.

Example

The following URL requests the enumerations for the model class attribute (0x11ee8):

```
http://localhost:8080/spectrum/restful/attribute/0x11ee8/enums
```

The following is returned for the request:

```
<?xml version="1.0" encoding="UTF-8"?>
<attrtribute-enum-response xmlns="http://www.ca.com/spectrum/restful/schema/response">
    <attribute id="0x11ee8"/>
    <enum id="0" value="Unknown"/>
    <enum id="1" value="Other"/>
    <enum id="2" value="Switch"/>
    <enum id="3" value="Router"/>
. . .
```

connectivity

Use the Connectivity resource to retrieve connectivity, and device and port condition information for a specified device model based on its IP address.

- **Base URL**
  http://<*hostname*><:*portnumber*>/spectrum/restful/connectivity

GET connectivity

GET connectivity returns XML describing all the connectivity from a requested end point.

- **URL**

```
http://<hostname><:portnumber>/spectrum/restful/connectivity/[ip_address][model_handle]
```

- **HTTP Method**
  GET
- **Body**
  None
- **Body Content**
  application/xml, application
- **Header**
  application/xml, application/json
- **Output**
  XML describing the connectivity

URL Parameters

- ***ip_address***
  Specifies the IP address to retrieve connectivity information for.

Example

The following URL requests connectivity, and device and port condition information for the specified IP address:

```
http://chave10-w7:80/spectrum/restful/connectivity/138.42.94.149
```

The following is returned for the request:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
- <connection-response-list xmlns="http://www.ca.com/spectrum/restful/schema/response">
- <connection-response>
- <connection-element-left>
  <ipaddress>138.42.94.149</ipaddress>
  <mh>0x100259</mh>
  <name>junM7i-96.20</name>
  <type>JuniperJUNOSRtr</type>
  <class>Router</class>
  <condition>0</condition>
  </connection-element-left>
- <connection-element-left>
  <ipaddress>138.42.94.149</ipaddress>
  <mh>0x100d73</mh>
  <name>junM7i-96.20_fxp1.0</name>
  <type>Gen_IF_Port</type>
  <class>Port</class>
  <condition>0</condition>
  </connection-element-left>
- <connection-element-right>
  <mh>0x100f59</mh>
  <name>10.0.0.0</name>
  <type>Unplaced</type>
  <class>Link</class>
  <condition>0</condition>
  </connection-element-right>
- <connection-element-right>
  <mh>0x0</mh>
  <class />
  </connection-element-right>
```

```
  </connection-response>
- <connection-response>
- <connection-element-left>
  <ipaddress>138.42.94.149</ipaddress>
  <mh>0x100259</mh>
  <name>junM7i-96.20</name>
  <type>JuniperJUNOSRtr</type>
  <class>Router</class>
  <condition>0</condition>
  </connection-element-left>
- <connection-element-left>
  <ipaddress>138.42.94.149</ipaddress>
  <mh>0x100d84</mh>
  <name>junM7i-96.20_fe-0/0/0.0</name>
  <type>Gen_IF_Port</type>
  <class>Port</class>
  <condition>0</condition>
  </connection-element-left>
- <connection-element-right>
  <mh>0x100f5b</mh>
  <name>192.168.95.96</name>
  <type>Unplaced</type>
  <class>Link</class>
  <condition>0</condition>
  </connection-element-right>
- <connection-element-right>
  <mh>0x0</mh>
  <class />
  </connection-element-right>
  </connection-response>
- <connection-response>
- <connection-element-left>
  <ipaddress>138.42.94.149</ipaddress>
  <mh>0x100259</mh>
  <name>junM7i-96.20</name>
  <type>JuniperJUNOSRtr</type>
  <class>Router</class>
  <condition>0</condition>
  </connection-element-left>
- <connection-element-left>
  <ipaddress>138.42.94.149</ipaddress>
  <mh>0x100d89</mh>
  <name>junM7i-96.20_fe-0/0/2.0</name>
  <type>Gen_IF_Port</type>
  <class>Port</class>
  <condition>0</condition>
  </connection-element-left>
- <connection-element-right>
  <ipaddress>138.42.94.150</ipaddress>
  <mh>0x100252</mh>
  <name>cis3640-96.44</name>
  <type>Rtr_Cisco</type>
  <class>Switch-Router</class>
```

```
  <condition>0</condition>
  </connection-element-right>
- <connection-element-right>
  <ipaddress>138.42.94.150</ipaddress>
  <mh>0x100667</mh>
  <name>cis3640-96.44_Fa0/0</name>
  <type>Gen_IF_Port</type>
  <class>Port</class>
  <condition>0</condition>
  </connection-element-right>
  </connection-response>
- <connection-response>
- <connection-element-left>
  <ipaddress>138.42.94.149</ipaddress>
  <mh>0x100259</mh>
  <name>junM7i-96.20</name>
  <type>JuniperJUNOSRtr</type>
  <class>Router</class>
  <condition>0</condition>
  </connection-element-left>
- <connection-element-left>
  <ipaddress>138.42.94.149</ipaddress>
  <mh>0x100d8b</mh>
  <name>junM7i-96.20_fe-0/0/3.0</name>
  <type>VLAN_If</type>
  <class>Port</class>
  <condition>0</condition>
  </connection-element-left>
- <connection-element-right>
  <ipaddress>138.42.94.158</ipaddress>
  <mh>0x100248</mh>
  <name>junM20-96.18</name>
  <type>JuniperJUNOSRtr</type>
  <class>Switch-Router</class>
  <condition>0</condition>
  </connection-element-right>
- <connection-element-right>
  <ipaddress>138.42.94.158</ipaddress>
  <mh>0x10077b</mh>
  <name>junM20-96.18_em0.0</name>
  <type>Gen_IF_Port</type>
  <class>Port</class>
  <condition>0</condition>
  </connection-element-right>
  </connection-response>
- <connection-response>
- <connection-element-left>
  <ipaddress>138.42.94.149</ipaddress>
  <mh>0x100259</mh>
  <name>junM7i-96.20</name>
  <type>JuniperJUNOSRtr</type>
  <class>Router</class>
  <condition>0</condition>
```

```
  </connection-element-left>
- <connection-element-left>
  <ipaddress>138.42.94.149</ipaddress>
  <mh>0x100d92</mh>
  <name>junM7i-96.20_t1-0/1/0.0</name>
  <type>Serial_IF_Port</type>
  <class>Port</class>
  <condition>0</condition>
  </connection-element-left>
- <connection-element-right>
  <mh>0x100f4b</mh>
  <name>138.42.94.212</name>
  <type>WA_Segment</type>
  <class>Link</class>
  <condition>0</condition>
  </connection-element-right>
- <connection-element-right>
  <mh>0x0</mh>
  <class />
  </connection-element-right>
  </connection-response>
  </connection-response-list>
```

devices

Use the Devices resource to retrieve DX NetOps Spectrum devices and attribute information.

> **NOTE**
> A *device* is defined as all models returned by the SpectroSERVER action FIND_ALL_DEVICE_MODELS (0x1023a).

- **Base URL**

  ```
  http://<hostname><:portnumber>/spectrum/restful/devices
  ```

GET devices

GET devices returns all device model handles. You can retrieve specific model attributes by using the &attr=<*attr_id*> parameter.

- **URL**

  ```
  http://<hostname><:portnumber>/spectrum/restful/devices[?attr=<attr_ID>][&landscape=<landscape_handle>]
  [&throttlesize=<num>]
  ```

- **HTTP Method**
  GET
- **Body**
  None
- **Body Content**
  Not Used
- **Header**
  application/xml, application/json
- **Output**
  XML or JSON listing all device model handles and requested attributes in the SpectroSERVER or distributed SpectroSERVER

URL Parameters

- **&attr=<*attr_ID*>**
  (Optional) Specifies the requested attributes. Multiple attribute parameters can be specified.

  NOTE
  There are many attributes on a model; for best performance you should limit attribute selection to attributes of interest.
- **&landscape=<*landscape_handle*>**
  (Optional) Filters which landscapes are queried. Multiple landscape parameters can be specified.
- **&throttlesize=<*num*>**
  (Optional) Specifies a throttle size.

Example

The following URL requests the name and model type of all devices:

```
http://<hostname><:portnumber>/spectrum/restful/devices?attr=0x1006e&attr=0x10000
```

The following is returned for the request:

```
   <?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
 - <model-response-list xmlns="http://www.ca.com/spectrum/restful/schema/response" error="EndOfResults"
  throttle="5" total-models="5">
   - <model-responses>
     - <model mh="0x1800178">
        <attribute id="0x1006e">ciscoRPM-9.18.ca.com</attribute>
        <attribute id="0x10000">Rtr_Cisco</attribute>
       </model>
     - <model mh="0x1800351">
        <attribute id="0x1006e">Rtr7301IPT_248</attribute>
        <attribute id="0x10000">Rtr_Cisco</attribute>
       </model>
     - <model mh="0x1800082">
        <attribute id="0x1006e">jun2300-96.17</attribute>
        <attribute id="0x10000">JuniperJUNOSRtr</attribute>
       </model>
     - <model mh="0x1800062">
        <attribute id="0x1006e">junM7i-96.20</attribute>
        <attribute id="0x10000">JuniperJUNOSRtr</attribute>
       </model>
     - <model mh="0x18002fb">
        <attribute id="0x1006e">AdminActive</attribute>
        <attribute id="0x10000">HPBladeOnboardAdmin</attribute>
       </model>
     </model-responses>
   </model-response-list>
```

Events

Use the events resource to create events.

NOTE
The events resource operates on a single model or groups of models.

- **Base URL**
  `http://<hostname><:portnumber>/spectrum/restful/events`

POST event

POST event creates a new event.

- **URL**
  `http://<hostname><:portnumber>/spectrum/restful/events/<eventType>/model/<model_handle>[?varbind=<varbind id>][&val=<varbind value>]`
- **HTTP Method**
  POST
- **Body**
  None
- **Body Content**
  Not Used
- **Header**
  application/xml, application/json
- **Encoding**
  UTF-8
- **Output**
  XML or JSON listing the model handle of the created model or the error if the model cannot be created.

URL Parameters

- **<eventType>**
  Specifies the event type to be created.
- **<model_handle>**
  Specifies the model handle on which the event will be created.
- **<varbind id>**
  Specifies the varbind id.
- **<varbind value>**
  Specifies the varbind value.

POST events

Use POST events to create one or more events on one or more models. POSTing an event request lets you provide an XML document that specifies the events to create and the models on which to create them. The document contains a list of model handles, XML search criteria, or a reference to the existing search criteria on the OneClick server.

- **URL**
  `http://<hostname><:portnumber>/spectrum/restful/events`
- **HTTP Method**
  POST
- **Body**
  Examples are provided in <$SPECROOT>/RestfulExamples/xml/Events. These XML files provide examples of properly formed XML for different purposes. Use these examples as a basis for your own XML:
  – CreateEventByModelHandleList.xml
  – CreateEventByModelSearch.xml
  – CreateMultipleEventsByModelHandle.xml
- **Body Content**
  application/xml
- **Header**

application/xml, application/json

- **Encoding**
  UTF-8
- **Output**
  XML or JSON listing of models satisfying the request input in the SpectroSERVER or distributed SpectroSERVER

Example: Create an event 0x10f06 on all RTR_Cisco Model Types, with 4 varbinds

The following is an example of creating an event 0x10f06 on all RTR_Cisco Model Types, with 4 varbinds

- **URL**

Post the following XML body to:

```
http://<hostname><:portnumber>/spectrum/restful/events
```

- **Body**

```
<?xml version="1.0" encoding="UTF-8"?>
<!--This sample event request will create an event of type 0x10f06 (generates a High Memory Utilization alarm)
 on all models found by the specified locater search (all models derived from RTR_Cisco) -->
<rs:event-request throttlesize="10" xmlns:rs="http://www.ca.com/spectrum/restful/schema/request"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.ca.com/spectrum/restful/
schema/request ../../../xsd/Request.xsd">
  <rs:event>
    <rs:target-models>
      <rs:models-search>
        <rs:search-criteria
             xmlns="http://www.ca.com/spectrum/restful/schema/filter">
          <filtered-models>
            <is-derived-from>
              <attribute id="AttributeID.MTYPE_HANDLE">
                <value>0x21000c</value> <!-- RTR_Cisco -->
              </attribute>
            </is-derived-from>
          </filtered-models>
        </rs:search-criteria>
      </rs:models-search>
    </rs:target-models>
   <!-- event ID -->
    <rs:event-type id="0x10f06"/>
    <!-- attributes/varbinds -->
    <rs:varbind id="0">75</rs:varbind>
    <rs:varbind id="1">99</rs:varbind>
    <rs:varbind id="3">mem_instance</rs:varbind>
    <rs:varbind id="5">name</rs:varbind>
  </rs:event>
</rs:event-request>
```

POST events (GET Tunneling)

This API is used to retrieve events of a model handle.

NOTE

This API is supported only on device models.

Attributes	Description
`model handle`	Specifies the device model handles to retrieve events
`start time and end time`	Specifies the start and end time in the epoch format.
`subcomponents events`	Specifies the events for subcomponents. Set to True if sub-components events are needed.
`exclude events`	Specifies the events which are excluded. These events if more than one are separated by coma.
`tags`	Specifies the requested attributes can be customized as per the user's preference, either fewer or more attributes can be requested.
`landscape`	Only one landscape can be supported.
`null`	Null value attributes will be returned if there is no existing value for the attribute.

- **URL**
  `http://ocserver:portnumber/spectrum/restful/events/getEvents`
- **HTTP Method**
  POST
- **Sample Body or Payload**

```
<?xml version="1.0" encoding="UTF-8"?>
<rs:get-event-request throttlesize="10"
xmlns:rs="http://www.ca.com/spectrum/restful/schema/request"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.ca.com/spectrum/restful/schema/request ../../../xsd/
Request.xsd ">
<rs:get-events-filter
start-time="1522640184000"
end-time="1522644324000"
subcomponents-events="false"
exclude-events="0x10005,0x10219,0x10706"
/
>
<rs:requested-attribute id="0x11f56"/> <!--Alarm Severity -->
<rs:requested-attribute id="0x11f4e"/> <!-- Created On -->
<rs:requested-attribute id="0x129fa"/> <!-- MODEL_HANDLE -->
<rs:requested-attribute id="0x1006e"/> <!-- Name -->
<rs:requested-attribute id="0x4820007"/> <!-- Event -->
<rs:requested-attribute id="0x11fb9"/> <!-- Created By -->
<rs:requested-attribute id="0x482001b"/> <!--Alarm Cleared On -->
<rs:requested-attribute id="0x482001c"/> <!--Alarm Cleared By -->
<rs:requested-attribute id="0x10000"/> <!-- Model type Name -->
<rs:requested-attribute id="0x11fb8"/> <!-- Event Type -->
```

```
<rs:requested-attribute id="0x12c0a"/> <!-- Event Precedence --
>
<!-- Models of Interest -->
<rs:target-models>
<rs:model mh="0x800258"/>
<!--<rs:model mh="0x100934"/>-->
<!-- <rs:model mh="0x1000089"/>-->
</rs:target-models>
<rs:landscape id="0x800000" />
</rs:get-event-request>
```

- **Sample Response**

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<get-event-response-list xmlns="http://www.ca.com/spectrum/restful/schema/response"
 total-events="30" throttle="2">
<get-event-responses>
<event id="5abe765b-e8a1-100c-0437-0050568c73e5">
<attribute id="0x11f56">3</attribute>
<attribute id="0x11f4e">1522431579</attribute>
<attribute id="0x129fa">0x1009ae</attribute>
<attribute id="0x1006e">Cisco7606-96.37.37.ca.com</attribute>
<attribute id="0x4820007">Device Cisco7606-96.37.37.ca.com of type Rtr_Cisco has
 stopped responding to polls and/or external requests. An alarm will be generated.</
attribute>
<attribute id="0x11fb9">System</attribute>
<attribute id="0x482001b">1522432925</attribute>
<attribute id="0x482001c">System</attribute>
<attribute id="0x10000">Rtr_Cisco</attribute>
<attribute id="0x11fb8">0x10d35</attribute>
<attribute id="0x12c0a">10</attribute>
</event>
<event id="5abe765b-e8a3-100c-0437-0050568c73e5">
<attribute id="0x11f56">null</attribute> , null will be retuned if there is no value
 exists for the attribute
<attribute id="0x11f4e">1522431579</attribute>
<attribute id="0x129fa">0x1009ae</attribute>
<attribute id="0x1006e">Cisco7606-96.37.37.ca.com</attribute>
<attribute id="0x4820007">Alarm number 845986 with probable cause id 0x10009
 generated for device Cisco7606-96.37.37.ca.com of type Rtr_Cisco. The severity of
 this alarm is CRITICAL.</attribute>
<attribute id="0x11fb9">System</attribute>
<attribute id="0x482001b">null</attribute>
<attribute id="0x482001c">null</attribute>
<attribute id="0x10000">Rtr_Cisco</attribute>
<attribute id="0x11fb8">0x10701</attribute>
<attribute id="0x12c0a">10</attribute>
</event>
```

```
</get-event-responses>
<link type="application/xml" href="http://ocserver-portnumber/spectrum/restful/
events/getEvents?
id=cdf7caef-9c4a-4f28-9270-6b4b8e87feb6&amp;start=2&amp;throttlesize=2" rel="next"/>
</get-event-response-list>
```

- **Body Content**
  application/xml, application
- Header
  application/xml, application/json
- **Output**
  XML or JSON listing of model-response indicating the success or failure of each update.

landscapes

Use the Landscapes resource to retrieve landscape information.

- **Base URL**

```
http://<hostname><:portnumber>/spectrum/restful/landscapes
```

GET landscapes

GET landscapes returns information about all landscapes.

- **URL**
  http://<*hostname*><:*portnumber*>/spectrum/restful/landscapes
- **HTTP Method**
  GET
- **Body**
  None
- **Body Content**
  Not Used
- **Header**
  application/xml, application/json
- **Output**
  XML or JSON listing all landscapes in SpectroSERVER or distributed SpectroSERVER

Example

The following URL requests all landscapes:

```
http://localhost/spectrum/restful/landscapes
```

The following is returned for the request:

```
  <?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
- <landscape-response xmlns="http://www.ca.com/spectrum/restful/schema/response" total-landscapes="1">
  - <landscape>
      <id>0x1800000</id>
      <name>comp001</name>
      <isPrimary>true</isPrimary>
      <spectrumVersion>9.2.1.005</spectrumVersion>
    </landscape>
  </landscape-response>
```

model

Contents

Use the model resource to create or delete a model and to read or modify model attributes. An example that illustrates model creation is included in this section. For more information, see Example: Create a Model Based on Model Type.

NOTE
The model resource operates on a single model at a time. For groups of models, use the models resource.

- **Base URL**
  `http://<hostname><:portnumber>/spectrum/restful/model`

POST model

The POST model lets you creates a new model and update list attribute on the Specified model.

Create a New Model

The following POST model creates a new model and returns the model handle.

- **URL**
  `http://<hostname><:portnumber>/spectrum/restful/model[?landscapeid=<landscape_handle>]`
  `[&mtypeid=<mtype_handle>][&agentport=<snmp_port>][&commstring=<comm_str>][&retry=<retry_cnt>]`
  `[&timeout=<timeout_val>][&ipaddress=<ip_address>][&parentmh=<model_handle>][&relationid=<rel_handle>]`
  `[&attr=<attr_id>&val=<num>]`
- **HTTP Method**
  POST
- **Body**
  None
- **Body Content**
  Not Used
- **Header**
  application/xml, application/json
- **Output**
  XML or JSON listing the model handle of the created model or the error if the model cannot be created

URL Parameters

- **&landscapeid=<*landscape_handle*>**
  (Optional) Specifies the landscape where the model will be created.
- **&mtypeid=<*mtype_handle*>**
  (Optional) Specifies the model type handle of the model to be created. Required if the ipaddress parameter is not specified (see below).
- **&agentport=<*snmp_port*>**
  (Optional) Specifies the SNMP management port on the device.
  Default: 161
- **&commstring=<*comm_str*>**
  (Optional) Specifies the community string.
  Default: public
- **&retry=<*retry_cnt*>**
  (Optional) Specifies the retry count.
  Default: 3
- **&timeout=<*timeout_val*>**
  (Optional) Specifies the timeout value.

Default: 3000

- **&ipaddress=<*ip_address*>**
  (Optional) Specifies the IP address.
- **&parentmh=<*model_handle*>**
  (Optional) Specifies the parent (left Association) model.
- **&relationid=<*rel_handle*>**
  (Optional) Specifies the association from the parent to this model.
  Default: Collects
- **&attr=<*attr_id*>&val=<*num*>**
  (Optional) Specifies attribute ID and value.

Update List Attribute on the Specified Model

The following POST model updates the list attribute on the specified model.

- **URL**
  `http://<hostname><:portnumber>/spectrum/restful/model/attr/update`
- **HTTP Method**
  POST
- **Body**
  XML
- **Body Content**

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<update-model-attrlist-request xmlns="http://www.ca.com/spectrum/restful/schema/request">
  <model mh="<model_handle>"/>
  <attribute-list id="<attribute_id>">
    <instance oid="<instance_id_0>" value="<value>"/>
    <instance oid="<instance_id_1>" value="<value>"/>
  </attribute-list>
</update-model-attrlist-request>
```

- **Body Parameters**
  <model mh="<model_handle>"/>
  Specifies the model to update.
  <attribute-list id="<attribute_id>">
  Specifies the attribute ID to update.
  <instance oid="<instance_id_0>" value="<value>"/>
  Specifies the instance id and values to update.

  NOTE
  Repeat the instance id tag to update multiple instances in the list attribute.
- **Header**
  application/xml, application/json
- **Output**
  XML listing the model handle of the updated model or an error if the model cannot be updated.

GET model

GET model reads attributes from the specified model.

- **URL**
  `http://<hostname><:portnumber>/spectrum/restful/model/<model_handle>[?attr=<attr_ID>]`
- **HTTP Method**

GET

- **Body**
  None
- **Body Content**
  Not Used
- **Header**
  application/xml, application/json
- **Output**
  XML or JSON listing of the requested attributes

URL Parameters

- ***model_handle***
  Specifies the model.
- **&attr=<*attr_ID*>**
  (Optional) Specifies the requested attributes. Multiple attribute parameters can be specified.

PUT model

PUT model updates attributes on the specified model.

- **URL**
  `http://<hostname><:portnumber>/spectrum/restful/model/<model_handle>?attr=<attr_ID>&val=<num>`
- **HTTP Method**
  PUT
- **Body**
  None
- **Body Content**
  Not Used
- **Header**
  application/xml, application/json
- **Output**
  XML or JSON listing of the requested attributes

URL Parameters

- ***model_handle***
  Specifies the model to update.
- **&attr=<*attr_ID*>&val=<*num*>**
  Specifies the attributes and values to update. Multiple attribute-value pairs can be specified.

DELETE model

DELETE model deletes the specified model.

- **URL**
  `http://<hostname><:portnumber>/spectrum/restful/model/<model_handle>`
- **HTTP Method**
  DELETE
- **Body**
  None
- **Body Content**
  Not Used
- **Header**

application/xml, application/json

- **Output**
  XML or JSON listing indicating success or failure

URL Parameters

- ***model_handle***
  Specifies the model to delete.

Example Create a Model Based on Model Type

You can use the POST method to create a model based on a model type. Use the model noun and the model type ID for the model type that you want to create. You can deploy this basic method to create pingable models, user models, maintenance schedules, containers, and any other type of model that you can create using DX NetOps Spectrum web services.

The basic format to use is shown in the following URL. Variables are indicated with *italics*:

http*://hostname*:*portnumber*/spectrum/restful/model[?mtypeid=*modeltype_of_a_device*][&parentmh=*model_handle*][&relationid=*rel_handle*][&attr=*attr_id*&val=<num>]

Example

Use any REST client to post the following URL:

http://*host IP address*:*portnumber*/spectrum/restful/model?mtypeid=0x00010290&parentmh=0x40000005&attr=0x12d7f&val=10.13.12.111

In this example, we supplied the following parameters:

- **val**
  The IP address of a pingable device (in this example, 10.13.12.111).
- **mtypeid**
  The model type of a device that you are modeling. In this example, it is a pingable device, 0x00010290. To find the model type ID for the model type that you want to create, use the Search feature in the Model Type Editor.
- **parentmh**
  The model_handle for a universe.
- **relationid**
  (Optional) The relation ID to identify an association between two models in the model domain (a relation).
- **attr**
  The attribute ID of ipaddress. The value is the device IP address.

models

Use the Models resource to retrieve and update models and attributes.

NOTE
The models resource operates on groups of models. For a single model, use the model resource.

Base URL

```
http://<hostname><:portnumber>/spectrum/restful/models
```

POST models (GET Tunneling)

Use POST models to get all model handles and requested attributes. POSTing a models request lets you provide an XML document that specifies the models to retrieve. The document contains a list of model handles, XML search criteria, or a reference to the existing search criteria on the OneClick server.

WARNING
POST models is a form of GET Tunneling used to retrieve multiple models. You cannot create a model with POST models. To create a model, use POST model.

- **URL**
  `http://<hostname><:portnumber>/spectrum/restful/models`
- **HTTP Method**
  POST
- **Body**
  Examples are provided in <*$SPECROOT*>/RestfulExamples/xml/Models. These XML files provide examples of properly formed XML for different purposes. Use these examples as a basis for your own XML:
  – GetCiscoRouterModels.xml
  – GetModelsByModelHandles.xml
  – GetModelsFromExistingSearch.xml
- **Body Content**
  application/xml
- **Header**
  application/xml, application/json
- **Output**
  XML or JSON listing of models satisfying the request input in the SpectroSERVER or distributed SpectroSERVER

Example 1: Java CALL

The following is an example call:

```
Java GenericPoster noun=Models file=resources/xml/Models/GetCiscoRouterModels.xml server=localhost
 username=jdoe password=spectrum port=8080
```

Example: Request XML to find device models by IP address

The following is an example of how to find device models by IP address by embedding the correct locater search in the POST body of a models GET. This operation searches all landscapes in a distributed SpectroSERVER environment as well as allowing you to specify the attributes of interest.

- **URL**
  POST the following XML body to:
  `http://<hostname><:portnumber>/spectrum/restful/models`
- **Body**

```
<?xml version="1.0" encoding="UTF-8"?>
<rs:model-request throttlesize="5"
 xmlns:rs="http://www.ca.com/spectrum/restful/schema/request"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.ca.com/spectrum/restful/schema/request ../../../xsd/
Request.xsd">
<rs:target-models>
 <rs:models-search>
 <rs:search-criteria xmlns="http://www.ca.com/spectrum/restful/schema/filter">
 <action-models>
```

```
 <filtered-models>
 <equals>
 <model-type>SearchManager</model-type>
 </equals>
 </filtered-models>
 <action>FIND_DEV_MODELS_BY_IP</action>
 <attribute id="AttributeID.NETWORK_ADDRESS">
 <value>172.22.96.6</value>
 </attribute>
 </action-models>
 </rs:search-criteria>
 </rs:models-search>
 </rs:target-models>
<rs:requested-attribute id="0x1006e" />
 <rs:requested-attribute id="0x10000" />
 <rs:requested-attribute id="0x10032" />
 <rs:requested-attribute id="0x12de2" />
</rs:model-request>
```

Example 2: Query list of device names and ip addresses that belong to a Global Collection

Method 1: The following is an example of how to query device names and IP addresses that belong to a Global Collection.

- **URL**
  POST the following XML body to:
  `http://<hostname><:portnumber>/spectrum/restful/models`
- **Body**

```
<?xml version="1.0" encoding="UTF-8"?>
<rs:model-request throttlesize="5"
 xmlns:rs="http://www.ca.com/spectrum/restful/schema/request"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.ca.com/spectrum/restful/schema/request ../../../xsd/
Request.xsd">
<rs:target-models>
 <rs:models-search>
 <rs:search-criteria
 xmlns="http://www.ca.com/spectrum/restful/schema/filter">
 <filtered-models>
 <has-substring>
 <attribute id="0x12adb"> <!-- This attribute stores the list of global collections
 to which a model belongs -->
 <value>gc1</value> <!-- Name of the Global Collection -->
 </attribute>
 </has-substring>
 </filtered-models>
 </rs:search-criteria>
 </rs:models-search>
```

```
    </rs:target-models>
   <rs:requested-attribute id="0x1006e" /> <!-- Model Name -->
    <rs:requested-attribute id="0x12d7f" /> <!-- Network Address -->
   </rs:model-request>
```

Method 2: You can also create a custom search criteria file (xml file) at "<SPECROOT>\tomcat\webapps\spectrum\WEB-INF\topo\config", and use it in the following xml file:

The following is the content of the search-devices-criteria.xml:

```
<?xml version="1.0" encoding="utf-8"?
>
 <search-criteria id="search-devices-criteria"
   xmlns="http://www.aprisma.com"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.aprisma.com../../common/schema/search-criteria-
config.xsd">
   <action-models>
     <filtered-models>
       <equals>
         <model-type>SearchManager</model-type>
       </equals>
     </filtered-models>
     <action>0x10474</action>
     <attribute id="AttributeID.MODEL_NAME">
       <value>gc1</value>
     </attribute>
   </action-models>

 </search-criteria>
```

Example 3: The following is an example of how to get JuniperJUNOSRtr model type from one landscape in a DSS environment.

- **URL**

  ```
  http://OCServer/spectrum/restful/models
  ```

- **Body**

  ```
  <rs:model-request throttlesize="500"
   xmlns:rs="http://www.ca.com/spectrum/restful/schema/request"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.ca.com/spectrum/restful/schema/request ../../../xsd/
  Request.xsd
   ">
   <rs:landscape id="0x100000" />
   <rs:target-models>
   <rs:models-search>
   <rs:search-criteria xmlns="http://www.ca.com/spectrum/restful/schema/filter">
   <filtered-models>
   <and>
  ```

```
 <equals>
 <attribute id="0x10001">
 <value>0x3b10002</value>
 </attribute>
 </equals>
 </and>
 </filtered-models>
 </rs:search-criteria>
 </rs:models-search>
 </rs:target-models>
 <rs:requested-attribute id="0x1006e" />
</rs:model-request>
```

- **Response**

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<model-response-list xmlns="http://www.ca.com/spectrum/restful/schema/response"
 total-models="1" throttle="1" error="EndOfResults">
 <model-responses>
 <model mh="0x1001d4">
 <attribute id="0x1006e">junM7i-96.19</attribute>
 </model>
 </model-responses>
</model-response-list>
```

NOTE

More than one landscape can be mentioned as shown:

<rs:landscape id="0x100000" />
<rs:landscape id="0x600000" />

GET models

GET models returns model handles and requested attributes. To retrieve specific model attributes, use the &attr=<*attr_id*> parameter.

WARNING
GET models can be used only after a POST (GET Tunneling) models has been performed. POST models generates the result set, and GET models is used to retrieve model handles and attribute information for the items within that result set.

- **URL**

```
http://<hostname><:portnumber>/spectrum/restful/models?id=<result_set_pointer>[&attr=<attr_ID>]
[&landscape=<landscape_handle>][&throttlesize=<num>]
```

- **HTTP Method**
  GET
- **Body**
  None
- **Body Content**
  Not used
- **Header**
  application/xml, application/json
- **Output**

XML or JSON listing of models satisfying the request input in the SpectroSERVER or distributed SpectroSERVER

URL Parameters

- **?id=<*result_set_pointer*>**
  Specifies the location of the result set on the OneClick server. This value is generated by using POST (GET Tunneling) models and appears on the "next" relative link. The location expires on the OneClick server after ten minutes of inactivity.
- **&attr=<*attr_ID*>**
  (Optional) Specifies the requested attributes. Multiple attribute parameters can be specified.

  NOTE
  There are many attributes on a model; for best performance you should limit attribute selection to attributes of interest. Not all models support the same set of attributes. Unsupported attributes will return 'NoSuchAttribute'
- **&landscape=<*landscape_handle*>**
  (Optional) Filters which landscapes are queried. Multiple landscape parameters can be specified.
- **&throttlesize=<*num*>**
  (Optional) Specifies a throttle size.

PUT models

PUT models updates attributes across multiple models. The models can be selected by a variety of techniques including search criteria. Not all selected models need to support the attributes being modified.

- **URL**
  `http://<hostname><:portnumber>/spectrum/restful/models`
- **HTTP Method**
  PUT
- **Body**
  Uses Request.xsd:update-models-request. The Request.xsd is located in <*$SPECROOT*>/RestfulExamples/src/xsd
- **Body Content**
  application/xml, application
- **Header**
  application/xml, application/json
- **Output**
  XML or JSON listing of model-response indicating the success or failure of each update.

GET (/models)

This API can be used to retrieve the models of a Global Collection. This rest call will accept the Global Collection name and retrieve those devices which are a part of the Global Collection.

- **URL**
  `http://ocserver:port/spectrum/restful/models`
- **HTTP Method**
  GET
- **Body**

```
<?xml version="1.0" encoding="UTF-8"?>
<rs:model-request throttlesize="100"
 xmlns:rs="http://www.ca.com/spectrum/restful/schema/request"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
 xsi:schemaLocation="http://www.ca.com/spectrum/restful/schema/request ../../../xsd/
Request.xsd
 ">
 <rs:target-models>
 <rs:gc name="GC1"/>
 </rs:target-
models>
 <rs:requested-attribute id="0x1006e" />
 <rs:requested-attribute id="0x10000" />
 <rs:requested-attribute id="0x10032" /
>
 </rs:model-request>
```

- **Body Content**
  application/xml, application
- **Header**
  application/xml, application/json
- **Output**
  XML or JSON listing of model-response indicating the success or failure of each update.

subscription

Use the Subscription resource to create or retrieve subscriptions. A *subscription* is a request to be notified of activity on any of the following:

- Model types/attributes. Registers watches for model creation/deletion of type and attribute changes on those types.

  NOTE
  ModelType subscriptions watch for new models.
- Models/attributes. Registers watches on specified models and related attribute changes.
- Alarms/attributes. Registers watches for alarm creation/clearing and attribute changes.
- **Base URL**

```
http://<hostname><:portnumber>/spectrum/restful/subscription
```

POST Subscription

POST subscription creates a subscription. Subscriptions are *pull* or *push*. A *pull* subscription requires that the client poll the subscription ID whereas a *push* subscription requires that the client provide a URL to which notifications can be POSTed. Notifications contain change information in XML or JSON format.

NOTE
Push subscriptions are useful when the integration client is another OneClick server.

- **URL**

```
http://<hostname><:portnumber>/spectrum/restful/subscription
```

- **HTTP Method**
  POST
- **Body**
  Uses Request.xsd:subscription-request. The Request.xsd is located in <*$SPECROOT*>/RestfulExamples/src/xsd
  For specific examples of Subscription XML, see the following:

  - In <*$SPECROOT*>/RestfulExamples/xml/Alarms:
    - PullAlarmsSubscription.xml
  - In <*$SPECROOT*>/RestfulExamples/xml/Models
    - PullAttrWatchForAllDevices.xml
  - In <*$SPECROOT*>/RestfulExamples/xml/MTypes:
    - PullWatchForNewMTypes.xml
    - PushWatchForNewMTypes.xml
    - PushWatchForNewMTypesBatchMode.xml
    - PushWatchForNewMTypesNoExpiration.xml
- **Body Content**
  application/xml, application
- **Header**
  application/xml, application/json
- **Output**
  XML or JSON listing the subscription-id
  - **subscription-id**
    Identifies the subscription. This value is used with GET subscription and DELETE subscription and expires if not used.

Body Parameters

The following parameters are used in the subscription request body XML.

- For pull subscriptions:
  - **max-notifications**
    Specifies the number of notification items.
    Default: 100
  - **max-queue-size**
    Specifies the number of notifications to queue.
    Default: 10000
  - **pull-interval**
    Specifies the frequency of polling in milliseconds.

    NOTE
    Subscriptions expire if not polled.

    Minimum: 5000

    NOTE
    If you expect a high frequency of notifications, you can accommodate plenty of notifications in the queue by increasing the **max-queue-size**. Simultaneously, to avoid a full queue, increase the **max-notifications** to pull more notifications in every poll. To empty the queue at a faster rate, reduce the **pull-interval**.

    For a POST Subscription request containing multiple InstanceRequests in it (for example, multiple GC Models in the body), allocate ThreadPoolSize, so that these individual InstanceRequests can be processed in parallel by individual threads. The default value is 20 and it should be sufficient. You can set the ThreadPoolSize in OneClickService.conf (Windows) or catalina.sh (Non-Windows) by adding this property under jvm_opts:

    ```
    restful.subscription.ThreadPoolSize=<int-value>
    ```

- For push subscriptions:
  - **destination-url**
    Specifies the URL where notifications are POSTed.
    A test notification servlet has been if prints results to the DX NetOps Spectrum tomcat log. This servlet is installed with DX NetOps Spectrum, and no additional configuration is required. To use the servlet, set the destination-url to:

    ```
    http://<hostname><:portnumber>/spectrum/restful/TestNotifications
    ```

Each time a notification is received it is printed to the DX NetOps Spectrum tomcat log, as follows:
- For Windows: <*$SPECROOT*>/tomcat/logs
- For Linux: <*$SPECROOT*>/tomcat/logs/catalina.out

– **username**
Specifies the user name required by the destination URL.
– **password**
Specifies the password required by the destination URL.
– batch-notifications
- **max-notifications**
Specifies the number of notifications in a batch. If this number is reached, notification is sent immediately.
Default: 100
- **max-time**
Specifies the time in milliseconds to wait before sending notifications. If this number is reached, notifications are sent immediately.
Default: 1000

– **heartbeat-interval**
Specifies the interval in milliseconds that the subscription is validated. The subscription servlet sends a message every hearbeat-interval to the receiving servlet (designated by the destination-url) to indicate that the subscription is active. If the receiving servlet stops receiving the heartbeat message, it knows that the subscription has stopped (for example, if the OneClick server stops or restarts). If the receiving servlet becomes unavailable, notifications stop.

NOTE
The TestNotifications servlet does not print the heartbeat message.

Example

The following is an example subscription registration snippet:

```
<rs:destination-url>http://<hostname><:portnumber>/spectrum/restful/TestNotifications</rs:destination-url>
<rs:username><user></rs:username>
<rs:password><password></rs:password>
```

The following is an example Java call:

```
Java GenericPoster noun=Models file= RestfulExamples/xml/Models/PullAttrWatchForAllDevices.xml
 server=localhost username=jdoe password=spectrum port=8080
```

GET Subscription

Use GET subscription to return XML describing all the changes of interest based on the initial subscription request.

NOTE
GET subscription applies to the pull model only.

- **URL**
  `http://<hostnumber><:portnumber>/spectrum/restful/subscription/<subscription_ID>`
- **HTTP Method**
  GET
- **Body**
  None
- **Body Content**
  application/xml, application
- **Header**
  application/xml, application/json

URL Parameters

- ***subscription_ID***
  Specifies the subscription ID. This value is returned from the original subscription request (POST subscription) or GET subscription/requests. It is used to obtain the set of changes since the last request.

GET subscription/requests

Use GET subscription/requests to return a string of the current, active subscriptions. This request is used primarily for debugging or verification purposes to see if a particular model, type or attribute is registered for notification.

> **NOTE**
> To pull the actual model, model type, and attribute changes, see GET subscription.

- **URL**
  `http://<hostname><:portnumber>/spectrum/restful/subscription/requests`
- **HTTP Method**
  GET
- **Body**
  None
- **Body Content**
  application/xml, application
- **Header**
  application/xml, application/json
- **Output**
  HTML page listing the subscription-id and other subscription-related information for each current, active subscription
  - **subscription-id**
    Identifies a subscription. This value is used with GET subscription and DELETE subscription.

DELETE subscription

DELETE subscription deletes the specified subscription.

- **URL**
  `http://<hostname><:portnumber>/spectrum/restful/subscription/<subscription_ID>`
- **HTTP Method**
  DELETE
- **Body**
  None
- **Body Content**
  application/xml, application
- **Header**
  application/xml, application/json

URL Parameters

- ***subscription_ID***
  Specifies the subscription to delete. This value is returned from the original subscription request (POST subscription) or GET subscription/requests.

Alarm and Model Attributes

The follow table contains alarm and model attributes that are commonly used when using the DX NetOps Spectrum Web Services API. *Italicized* items are model attributes. Non-italicized items are alarm-specific attributes.

Attribute Specification	Attribute Name	Data Type
0x10000	*Model Type Name*	*Text String*
0x10001	*Model Type of Alarmed Model*	*Model Type Handle*
0x10009	*Security String*	*Text String*
0x1000a	*Condition*	*Integer*
0x1006e	*Model Name*	*Text String*
0x11ee8	*Model Class*	*Integer*
0x11f4d	Acknowledged	Boolean
0x11f4e	Creation Date	Date
0x11f4f	Alarm Status	Text String
0x11f50	Cause Code	Integer
0x11f52	Event ID List	Octet String
0x11f53	Model Handle of Alarmed Model	Model Handle
0x11f54	Primary Alarm	Boolean
0x11f56	Severity	Integer
0x11f57	Troubleshooter	Text String
0x11f9b	User Clearable	Boolean
0x11f9c	Alarm ID	Octet String
0x11fc4	Alarm Source	Enumeration (0=Current, 1=Residual)
0x11fc5	Occurrences	Gauge
0x11fc6	Troubleshooter Model Handle	Model Handle
0x12022	Trouble Ticket ID	Text String
0x1296e	Originating Event	Octet String
0x12a04	Symptom List	Octet String
0x12a05	Cause List	Octet String
0x12a06	Symptom Count	Integer
0x12a07	Cause Count	Integer
0x12a56	Significant Model ID	Integer
0x12a63	Web Context URL	Text String
0x12a6f	Event Symptom List	Octet String
0x12a70	Event Symptom Count	Integer
0x12a82	IP to Domain Map	Octet String
0x12b4c	Alarm Title	Text String
0x12c05	Secure Domain Display	Text String
0x12d7f	Network Address	Internet Address
0x12d83	Secure Domain Address	Internet Address
0x1321a	Last Occurrence Date	Date

REST Operations (Verbs)

The following REST verbs are supported in the DX NetOps Spectrum Web Services API. All operations respect standard DX NetOps Spectrum user security.

- POST (Create)
- GET (Read)
- PUT (Update)
- DELETE (Delete)

> **NOTE**
> The HEAD, OPTIONS, and TRACE verbs are not supported in the DX NetOps Spectrum Web Services API.

Web Application Description Language (WADL) URL

The following is the WADL URL for the DX NetOps Spectrum Web Services API:

```
http://<hostname><:portnumber>/spectrum/restful/services
```

> **NOTE**
> The WADL is provided for advanced users and describes the simple URLs but does not describe GET Tunneling. Use the schema files for GET Tunneling.

XML Schema Definitions

Input and output data transfer is done in XML, conforming to a strict schema that enables standard access to data from different applications. The schemas provided in the DX NetOps Spectrum Web Services API defines the correct data layout for all valid requests and responses.

Used to define the required syntax for GET Tunneling, the schema files are also more useful than WADL for advanced usage. The schema files are used to generate the JAXB beans that can be used for java and javascript development. These beans are available in <*$SPECROOT*>/RestfulExamples/lib/spectrumrest.jar.

DX NetOps Spectrum provides the following XML Schema Definitions (.xsd) files, which are located in <*$SPECROOT*>/RestfulExamples/src/xsd:

- **Filter.xsd**
  Specifies the XML for attribute filters and search criteria.

  > **NOTE**
  > This is the same XML that is used by the DX NetOps Spectrum Locater; however, the Filter.xsd is only used by the DX NetOps Spectrum Web Services API.
- **Request.xsd**
  Defines the XML format used in GET Tunneling. All POST body XML must have a tag like the following:
  ```
  xsi:schemaLocation="http://www.ca.com/spectrum/restful/schema/request ../../../xsd/Request.xsd">
  ```
- **Response.xsd**
  Defines the XML or JSON format for all synchronous responses and XML format for all asynchronous responses.

Java Code and XML Examples

Java Code Examples

The DX NetOps Spectrum Web Services API provides example Java code that demonstrates basic functionality, JAXB beans, and hard-coded XML. An Eclipse .project and .classpath are also provided. The files are located in <*$SPECROOT*>/RestfulExamples/src/test.

NOTE
Each of the example programs requires certain parameter values to be provided.

The following examples are provided in the /client subfolder.

- **AlarmPoller**
  Subscribe for and pull alarms.
- **CreateSimilarModels**
  Create many models of the same type.
- **GenericPoster**
  A generic way to POST XML.
- **GetAllAlarms**
  A simple alarm reader.
- **GetAllDevices**
  A simple example that reads attributes on devices.
- **GetAllLandscapes**
  The simplest example.
- **GlobalCollectionExample**
  Creates a new Global Collection Model.

For more information, see the readme.txt files and internal documentation.

Program Arguments

The following program arguments are used by the provided Java code examples.

- **server={*server*}**
  Specifies the host name of the OneClick server.
- **username={*username*}**
  Specifies a user name for the OneClick server.
- **password={*password*}**
  Specifies the username password for the OneClick server.
- **method={*method*}**
  Specifies the REST verb, or HTTP method.
  Values: GET, POST, PUT, DELETE
- **port={*port*}**
  (Optional) Specifies the port number of the OneClick server.
  Default: 80
- **accept={*acceptType*}**
  (Optional) Specifies the accept header value/MIME type.
  Values: application/xml, application/json
  Default: application/xml
- **inputdata={*inputdataType*}**
  (Optional) Specifies the body content type.
  Value: application/xml
- **secure=true|false**
  (Optional) Specifies whether to secure the connection to the OneClick server.
  Default: False

If set to True, the following parameters are required:

- **keystore_file={*fqfn for keystore file*}**
  Specifies the fully-qualified keystore file name.
- **keystore_password={*keystore password*}**
  Specifies the password for the keystore.
- **alias={*certificate alias*}**
  Specifies the certificate alias name.

> **NOTE**
> To use a secure connection, the OneClick server must be configured for SSL. For information about configuring OneClick for SSL, see the OneClick Administration section.

XML Examples

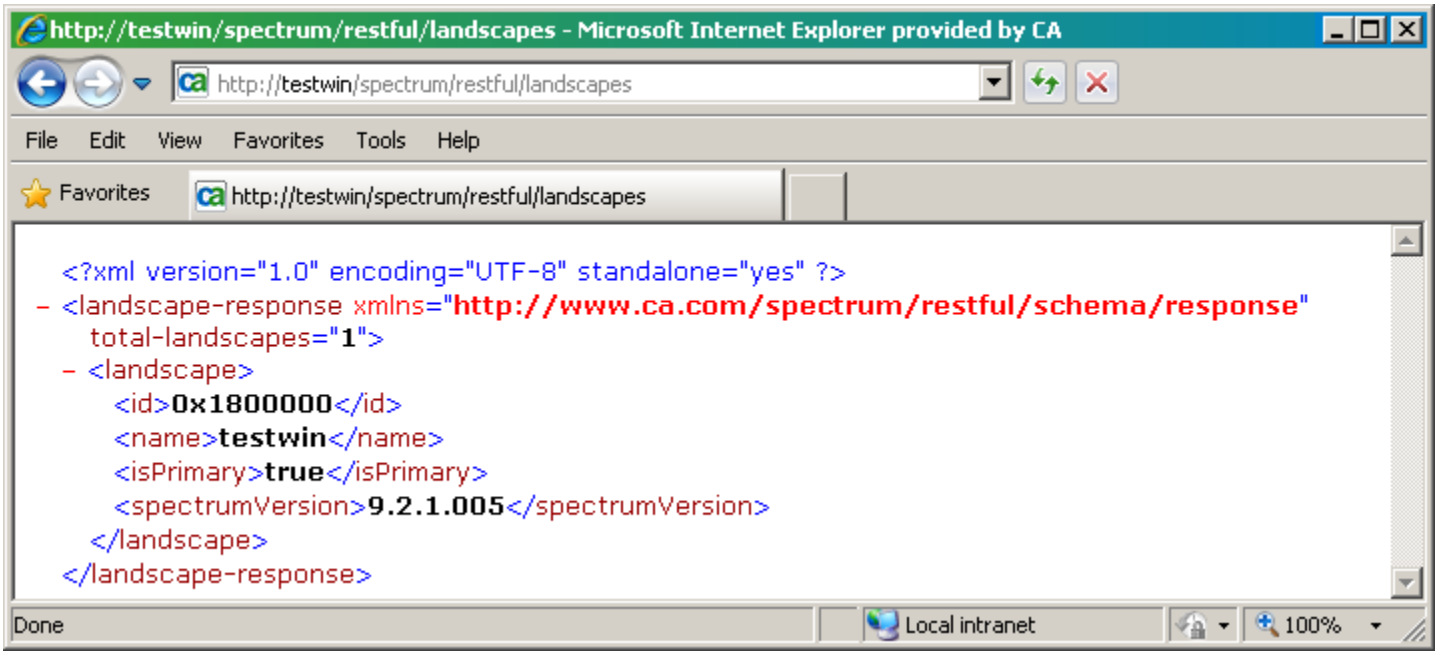
The DX NetOps Spectrum Web Services API provides example XML for GET Tunneling requests. The files are located in <*$SPECROOT*>/RestfulExamples/xml.

The following examples are provided:

- Alarms
  - GetAlarmsByAlarmIDs.xml
  - GetAlarmsByAttributeFilter.xml
  - GetAlarmsByModelHandles.xml
  - GetAlarmsBySearchCriteria.xml
  - GetAlarmsForAllDevices.xml
  - PullAlarmsSubscription.xml
  - FilterSecondaryAlarmsWhenInMaintenance.xml
- Events
  - CreateEventByModelHandleList.xml
  - CreateEventByModelSearch.xml
  - CreateMultipleEventsByModelHandle.xml
- Models
  - GetCiscoRouterModels.xml
  - GetModelsByModelHandles.xml
  - GetModelsFromExistingSearch.xml
  - PullAttrWatchForAllDevices.xml
  - PutModelsByModelHandles.xml
- ModelTypes (MTypes)
  - PullWatchForNewMTypes.xml
  - PushWatchForNewMTypes.xml
  - PushWatchForNewMTypesBatchMode.xml
  - PushWatchForNewMTypesNoExpiration.xml

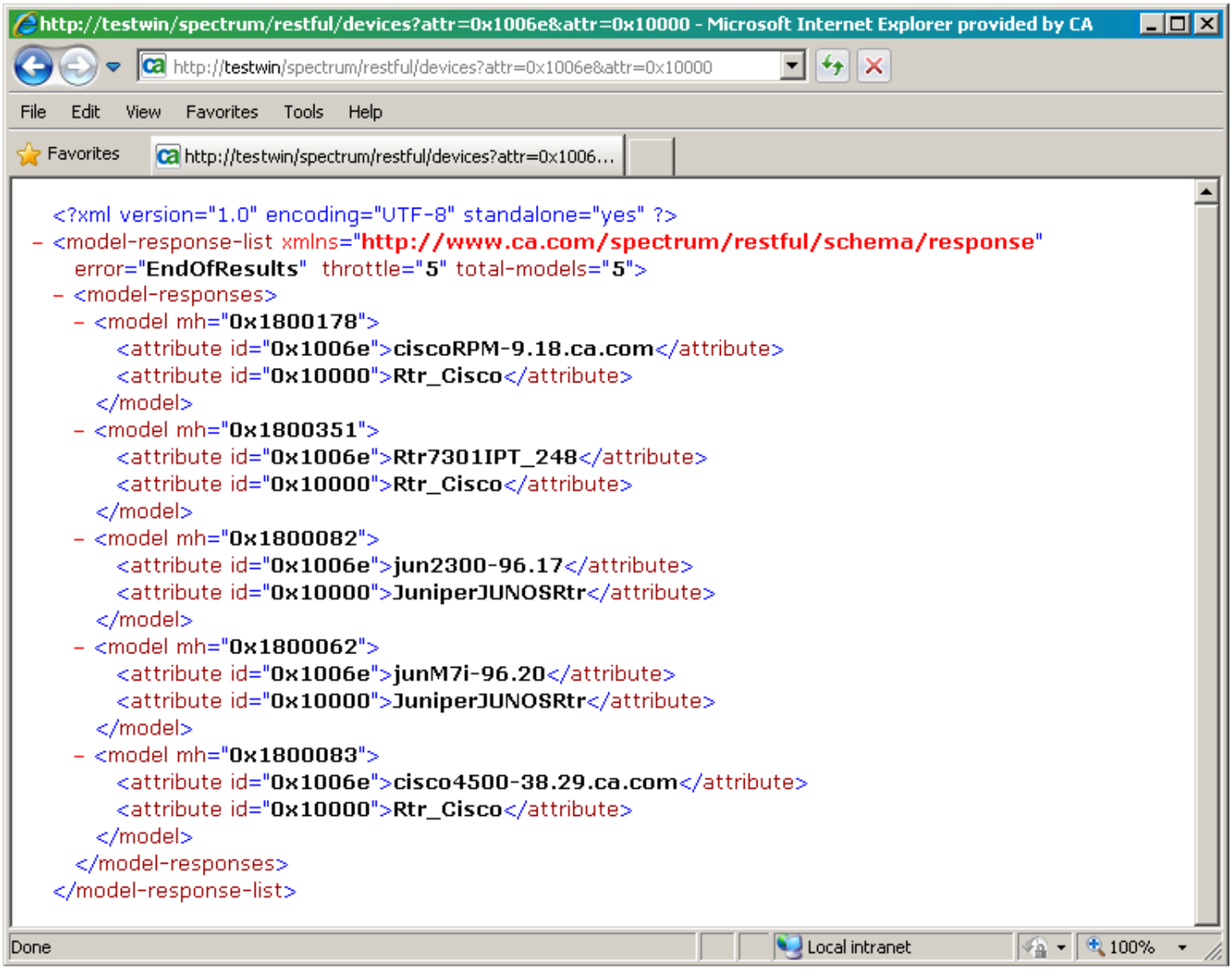
Simple URL Request in Browser

This example shows a GET landscapes request and results in a browser window. The GET verb is implied in the browser environment. Parameters are not necessary on the URL to return a list of all landscapes in the distributed SpectroSERVER environment.



Simple URL Request with Parameters in a Browser

This example uses the devices noun with URL parameters to request the name (0x1006e) and model type (0x10000) of all devices. The GET verb is implied in the browser environment.



Using Throttle and Next

This example uses GET alarms to retrieve all alarms and shows how the throttle and "next" link are used.

Follow these steps:

1. Use the following request, which asks for all alarms to be returned but only three at a time:

   ```
   http://comp001/spectrum/restful/alarms?throttlesize=3
   ```

   Notice the following in this request:

   – **throttlesize**
   Indicates the number of items to return at a time. This example requests three alarms to be returned at a time.

   The following response returns from the SpectroSERVER. This response indicates that a total of five alarms exist and only three are returned. A relative link is provided that points to the next items in the list.

   ```
   <?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
   - <alarm-response-list xmlns="http://www.ca.com/spectrum/restful/schema/response" throttle="3" total-alarms="5">
   ```

```
- <alarm-responses>
    <alarm id="4e77f4fd-1c9b-1000-0336-000874f00c29" />
    <alarm id="4e77f4fe-1ca2-1000-0336-000874f00c29" />
    <alarm id="4e80da2b-a48e-1001-0336-000874f00c29" />
  </alarm-responses>
  <link rel="next" href="http://comp001/spectrum/restful/alarms?id=c6a367df-0b3e-4461-aa8d-
aad451a45bf7&start=3&throttlesize=3" type="application/xml" />
  </alarm-response-list>
```

Notice the following in this response:

- **throttle**
  Indicates the number of items returned in this response. In this example, three alarms have been returned in this response.
- **total-alarms**
  Indicates the total number of items in the result set. In this example, there a five total alarms in the result set.
- **link rel="next"**
  Indicates the URL to retrieve the next items in the result set. This link can be used in a browser or a program to issue the next request.
  - **id=<*result_set_ID*>**
    Specifies the location of the result set on the OneClick server. The data expires after ten minutes of inactivity.
  - **start=<*element_num*>**
    Indicates the position within the result set.

2. Use the "next" relative link from the previous response to request the next three items in the result set:

```
http://comp001/spectrum/restful/alarms?id=c6a367df-0b3e-4461-aa8d-
aad451a45bf7&start=3&throttlesize=3
```

The following response returns from the SpectroSERVER:

```
  <?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
- <alarm-response-list xmlns="http://www.ca.com/spectrum/restful/schema/response" error="EndOfResults"
 throttle="2" total-alarms="5">
- <alarm-responses>
    <alarm id="4e83d694-18e1-1002-0336-000874f00c29" />
     <alarm id="4e77f4fe-1c9e-1000-0336-000874f00c29" />
  </alarm-responses>
</alarm-response-list>
```

Notice the following in this response:
error="EndOfResults"
Indicates that all items in the list have been retrieved.

Using URL and GET Tunneling for Same Request

This example shows the same request made by a URL request and a GET Tunneling request.

- URL Request
- GET Tunneling Request

URL Request

This example uses GET devices, in a simple URL request format, to retrieve attributes for all devices, throttling at 100.

Initial request

This request specifies the attributes to read:

```
http://localhost/spectrum/restful/devices?attr=0x1006e&attr=0x10000&attr=0x10032&attr=0x12de2&throttlesize=100
```

All matching devices are found and cached. The first 100 devices are returned, and, if there are more than 100, a link to get the next 100.

Subsequent request

Using the link from the initial request-response, the following request pulls the next set of results from the cache:

```
http://localhost/spectrum/restful/devices?id=0120a63a-e2d3-4175-9563-c17c90c783a7&start=100&throttlesize=100
```

GET Tunneling Request

This example uses GET devices, in embedded XML request format, to retrieve attributes for all devices, throttling at 100.

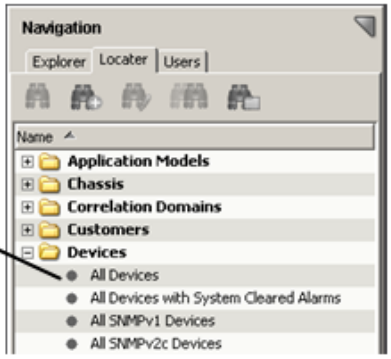
```
<?xml version="1.0" encoding="UTF-8"?>

<rs:model-request throttlesize="100"
  xmlns:rs="http://www.ca.com/spectrum/restful/schema/request"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.ca.com/spectrum/restful/schema/request ../../../xsd/Request.xsd ">


    <rs:target-models>
      <rs:models-search>
        <rs:search-criteria-file>
          topo/config/search-devices-criteria.xml
        </rs:search-criteria-file>
      </rs:models-search>
    </rs:target-models>

    <rs:requested-attribute id="0x1006e" /> <!-- Model name -->
    <rs:requested-attribute id="0x10000" /> <!-- Model Type Name -->
    <rs:requested-attribute id="0x10032" /> <!-- Manufacturer -->
    <rs:requested-attribute id="0x12de2" /> <!-- NRM_RunningFirmwareFilters -->

</rs:model-request>
```

Throttlesize set to 100

Schema

Uses existing search in OneClick Server

Attributes to read



The following portions of Request.xsd are of interest for this example. This request uses the *model-request* type. The type definition determines the parameters that can be specified, in this case, "target-models", "requested-attribute", and "throttlesize".

> **NOTE**
> See the complete contents in <*$SPECROOT*>/RestfulExamples/src/xsd/Request.xsd.

```
<xs:complexType name="model-request">
  <xs:sequence>
    <xs:element name="target-models" type="rs:target-models" minOccurs="1"
      maxOccurs="1" />
    <xs:element name="requested-attribute" type="rs:requested-attribute"
      minOccurs="0" maxOccurs="unbounded" />
  </xs:sequence>
  <xs:attribute name="throttlesize" type="xs:int" />
</xs:complexType>
   .
   .
   .
```

```
<xs:complexType name="target-models">
  <xs:choice>
    <xs:element name="model" type="rs:model" minOccurs="0" maxOccurs="unbounded" />
    <xs:element name="models-search" type="rs:models-search" minOccurs="0"
      maxOccurs="1"/>
  </xs:choice>
</xs:complexType>
   .
   .
   .
<!-- Elements -->

<xs:element name="alarm-request" type="rs:alarm-request" />
<xs:element name="model-request" type="rs:model-request" />
<xs:element name="subscription-request" type="rs:subscription-request" />
<xs:element name="update-models-request" type="rs:update-models-request" />
```

Troubleshooting the Web Services API Issues

Common Issues

If you are experiencing problems when using the DX NetOps Spectrum Web Services API, check the following:

- Is the URL constructed correctly?
- Is the correct verb used with the noun?
- Is the Request XML for GET Tunneling constructed correctly?
- Have the proper credentials been entered?
- Has the "accept" header and "Content-type" been set correctly for XML or JSON?
- Do any DX NetOps Spectrum Web Services API errors appear in the tomcat log?

"Next" Link Does not Work

Symptom:

The "next" link that was generated in a response does not work correctly when I try to retrieve the next throttled set of results.

Solution:

The format of the parameters within the generated link may vary depending on your environment. For example, "&" may be generated as "&" This is due to differences between XML, HTTP, and JSON formatting standards. Notice the difference in the prefixes for the start and throttlesize parameters in the following URLs:

```
http://localhost/spectrum/restful/models?id=5b03b5ba-64ed-4603-
b3e1-8e71919fccd8&#38;start=2&#38;throttlesize=2
http://localhost/spectrum/restful/models?id=5b03b5ba-64ed-4603-b3e1-8e71919fccd8&start=2&throttlesize=2
```

To correct, change '&amp' to '&' in your URL.

> **NOTE**
> This applies to XML only (not JSON).

Web Services API Search Crashes the SpectroSERVER

Symptom:
The DX NetOps Spectrum SpectroSERVER crashes shortly after running a webservices api search.
The stack output will look similar to this (with the key beingCsModelDomainSrvc::getModelIDListByXmlSearchCriteria):

```
#0 0x00002b2f85bc9ca0 in CsBuffer::makestr(int, char) const () from /opt/SPECTRUM/lib/libGlobl.so.1
#0 0x00002b2f85bc9ca0 in CsBuffer::makestr(int, char) const () from /opt/SPECTRUM/lib/libGlobl.so.1
#1 0x00002b2f7e8d46cb in CsLandscape::search_keyed_models(CsFindSpec const*, CsULHashTable*,
 CsFindSpec::LogicalOp_e, CsULHashTable*, int) () from /opt/SPECTRUM/lib/../SS/libsskrnl.so.1
#2 0x00002b2f7e8d4ad7 in CsLandscape::terminal_find_model_handles(CsFindSpec const*, CsULHashTable*,
 CsFindSpec::LogicalOp_e, CsError::CsError_e*, int) () from /opt/SPECTRUM/lib/../SS/libsskrnl.so.1
#3 0x00002b2f7e8d4ee6 in CsLandscape::find_model_handles(CsFindSpec const*, CsError::CsError_e*, CsSecurityIf
 const*, int) () from /opt/SPECTRUM/lib/../SS/libsskrnl.so.1
#4 0x00002b2f7e8d5786 in CsLandscape::find_models(CsFindSpec const*, CsSecurityIf const*, int) () from /opt/
SPECTRUM/lib/../SS/libsskrnl.so.1
#5 0x00002b2f7e8d5871 in CsLandscape::find_models(CsFindSpec const*, CsSecurityIf const*) () from /opt/
SPECTRUM/lib/../SS/libsskrnl.so.1
#6 0x00002b2f7e8f8d04 in CsModelDomainSrvc::get_model_desc_list(CsSecurityIf const&, char const*, char const*,
 CsULHashTable*) () from /opt/SPECTRUM/lib/../SS/libsskrnl.so.1
#7 0x00002b2f7e8f9b4c in CsModelDomainSrvc::getModelIDListByXmlSearchCriteria(CsCAttribute::CsCValue const&,
 CsSecurityIf const&) () from /opt/SPECTRUM/lib/../SS/libsskrnl.so.1
#8 0x00002b2f7e858217 in CModelDomainItcM::processItcRequest(ITC_Request_Parms*) () from /opt/SPECTRUM/lib/../
SS/libsskrnl.so.1
#9 0x00002b2f80fb30d4 in ITC_Request_Parms_Corba::processRequest_TransferExceptions() () from /opt/SPECTRUM/
lib/libitc.so.1
#10 0x00002b2f80fb28e1 in ItcWorkQueue::process_work_item(ItcQdItem*) () from /opt/SPECTRUM/lib/libitc.so.1
#11 0x00002b2f80fb26a2 in ItcWorkQueue::process_work_node(CsWorkNode*) () from /opt/SPECTRUM/lib/libitc.so.1
#12 0x00002b2f8340d3aa in CsWorkScheduler::do_work() () from /opt/SPECTRUM/lib/libwkmgr.so.1
#13 0x00002b2f7ebde7a3 in moot_thread_start () from /opt/SPECTRUM/lib/libmoot.so.1
#14 0x00000031a16419e0 in ?? ()
```

Cause:
The cause is due to an incompatible search type in the web service query.
For example, the isManaged attribute (0x1295d) only supports and "equal" value.
It will not support "has-substring" or "has-substring-ignore-case"

Solution:
Check your web services search. If you are using an "incorrect" search type for the attribute you are searching on you will need to correct it. If you are unsure that you are using an incorrect search type, you can create the same search in the OneClick Locater and save the search. Then review the xml associated to the search on the OneClick web server in the <SPECROOT>/custom/console/config directory.

For example, this search for the isManaged attribute (0x1295d) will crash your SpectroSERVER because the isManaged attribute will not work with a substring search:

<has-substring-ignore-case>
<attribute id="0x1295d">
<value>no</value>
</attribute>
</has-substring-ignore-case>
This needs to be updated to be:
<equals>
<attribute id="0x1295d">
<value>no</value>
</attribute>

</equals>

DX NetOps Spectrum Integrator

The following section provides an overview of the DX NetOps Spectrum Integrator features and capabilities, as well as in-depth procedures to perform integration activities.

Integrator Overview

The following section provides a brief overview of the DX NetOps Spectrum Integrator features and capabilities.

Integrating Third-Party Applications with DX NetOps Spectrum

IT organizations often use multiple applications to manage various aspects of their infrastructure. Integrating other management applications with DX NetOps Spectrum can be beneficial. You can create a powerful architecture that lets you combine the automation features and analysis from multiple management systems.

When designing a distributed automation architecture, evaluate how each tool contributes to the overall solution. Consider how the integrated tools can exchange data. This section can help you decide how to integrate third-party applications with DX NetOps Spectrum. The topics in this space demonstrate where and how you can exchange data with DX NetOps Spectrum.

Typically, an integration plan for DX NetOps Spectrum seeks to accomplish one or more of the following tasks:

- Send alert information to DX NetOps Spectrum
- Send network topology information to DX NetOps Spectrum
- Retrieve alarm information from DX NetOps Spectrum
- Create new SNMP device management modules within DX NetOps Spectrum
- Launch an external application from within a managed object context from one or more DX NetOps Spectrum application user interfaces
- Package and distribute an integrated system

This section discusses your options for accomplishing those tasks.

DX NetOps Spectrum Toolkits

DX NetOps Spectrum has several toolkits to help you complete integration tasks. This section provides an overview of each of the available toolkits and refers you to sources of more information.

- The Southbound Gateway lets you send alert data from third-party systems to DX NetOps Spectrum. The third-party system and all the sources for the appropriate alerts are automatically modeled in DX NetOps Spectrum. As a result, the network administrator can monitor this portion of the environment from OneClick.
- The Modeling Gateway imports network topology information into the SpectroSERVER database. An XML specification standardizes the information that is required to create the topological information. You can configure DX NetOps Spectrum to check for updated information at specified intervals.
- The AlarmNotifier and DX NetOps Spectrum Alarm Notification Manager (SANM) extract alarm data from DX NetOps Spectrum and pass it along to a user-defined application.
- The Model Type Editor and the GnSNMPDev model type let you derive new model types that are based on third-party MIBs. You can use these types to create new SNMP device management modules.
- You can edit DX NetOps Spectrum support files to enable the launching of third-party applications from OneClick. Context data can pass to the application as it is launched.
- The DX NetOps Spectrum Extension Integration (SEI) toolkit provides a series of tools and files that let you package all the components for an integration. You can then install these components on a DX NetOps Spectrum host system.

All of these tools and features provide safe, stable mechanisms for integration.

You can also use a programmatic interface with DX NetOps Spectrum. However, the programmatic interface requires an in-depth understanding of the DX NetOps Spectrum knowledge base (information model). This knowledge base includes many attributes, relations, and model types. A full understanding of component interactions is required to avoid risky changes that affect DX NetOps Spectrum operation. Also, a tight programmatic integration requires frequent updates to the integration code as DX NetOps Spectrum technology evolves.

Therefore, we recommend that both developers and end users rely on the tools that are detailed in this section whenever possible. These integration points reduce risk and maintain a flexible, powerful integration interface. DX NetOps Spectrum integration tools not only minimize the complexity of the integration tasks, but also enhance the functioning of the solution.

CORBA API Toolkit

The DX NetOps Spectrum CORBA API is an advanced toolkit that lets you develop C++ or Java based advanced extensions to DX NetOps Spectrum. Use this toolkit to create programs that integrate with the SpectroSERVER through an object-oriented remote method invocation interface.

Programming with this API requires an in-depth knowledge of DX NetOps Spectrum. Use the API only when necessary. The other toolkits that are discussed in detail in this section have been designed for use by most integrators. We recommend using them in most situations.

Command Line Interface

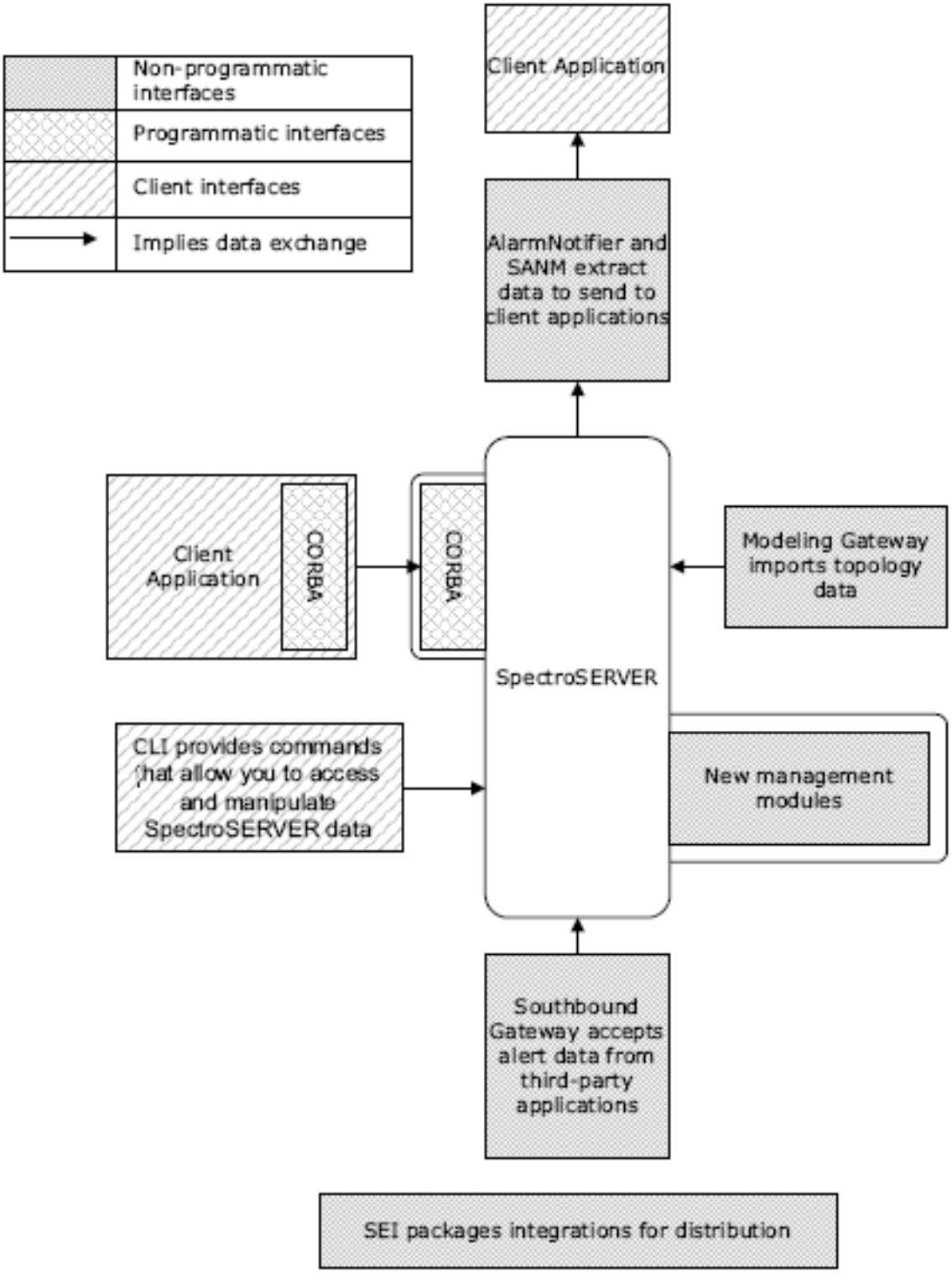
The DX NetOps Spectrum Command Line Interface (CLI) lets you access the data in the DX NetOps Spectrum knowledge base from a Unix shell environment or Windows command prompt. These commands can also be used in script files. Use the CLI to view, create, modify, and delete DX NetOps Spectrum data about models, alarms, events, and associations.

The CLI does not provide the safeguards that are available from the OneClick interface. We therefore do not recommend using it if you are unfamiliar with DX NetOps Spectrum and its modeling scheme. We also recommend testing a CLI-based integration with each release of DX NetOps Spectrum to verify the preservation of the intended behavior.

The CLI can also be an effective tool to use before working directly with the CORBA API. Use the CLI to test a prototype of your code as it interacts with the DX NetOps Spectrum knowledge base. If the data is as expected, you can then be more confident in your approach toward the CORBA API, which carries some risks.

The following diagram illustrates the integration points available in DX NetOps Spectrum:

SPECTRUM Integration Points



Benefits of Southbound Gateway and Modeling Gateway

Both the Southbound Gateway and the Modeling Gateway provide well-defined integration points for data exchange. These gateways were designed for the most common areas where third-party developers want to exchange data with DX NetOps Spectrum.

The Southbound Gateway integration point has built-in model types and inference handlers that are designed to handle integration with a third-party alert source. You can further customize the Southbound Gateway integration by using the EventAdmin model type or the EventModel model type as a derivation point. Use these types as you would use the GnSNMPDev model type to derive a new model type.

The Modeling Gateway provides a set of XML elements that lets you port data from a provisioning database or other network topology database. In addition, you can expand and customize the elements and attributes in the XML syntax to accommodate most integrations. You can also use the Modeling Gateway to export DX NetOps Spectrum topology data to an XML file.

Prerequisites for Developers

Significant experience using DX NetOps Spectrum is required before you attempt to use any of the DX NetOps Spectrum integration points.

In addition, a Developer ID is required to distribute your integrations.

NOTE
For more information, see *DX NetOps Spectrum* Getting Started .

Required Components

Many of the toolkits described in this section are packaged and sold as separate components. The following toolkits and the associated components are needed for each integration:

- **Southbound Gateway Toolkit**
  Available as an add-on component for some DX NetOps Spectrum packages that you can purchase.
  - **DX NetOps Spectrum Extension Integration (SEI)**
    Available as part of the Level 1 Toolkit.

    NOTE
    The Level 1 Toolkit includes the Model Type Editor and the SEI Toolkit.
  - **[assign the value for corba in your book]**
    (Optional) Required only if you want a programmatic Southbound Gateway integration. Available as an add-on component for some DX NetOps Spectrum packages that you can purchase.
- **Modeling Gateway Toolkit**
  Available as an add-on component for some DX NetOps Spectrum packages that you can purchase.
- **AlarmNotifier Application**
  Included with all DX NetOps Spectrum packages.
- **DX NetOps Spectrum Alarm Notification Manager (SANM)**
  Available as an add-on component for some DX NetOps Spectrum packages that you can purchase.
- **Model Type Editor**
  Available as part of the Level 1 Toolkit.
- **SPECTRUM Extension Integration (SEI)**
  Available as part of the Level 1 Toolkit.
- **Launch Point Integration**
  None
- **CLI Application**
  Included with all DX NetOps Spectrum packages.
- **[assign the value for corba in your book]**

Available as an add-on component for some DX NetOps Spectrum packages that you can purchase.

Sending Alert Data to the Product

The following topics cover in detail the various ways in which incoming alert data is managed and processed by ,

Southbound Gateway Model Types

Two model types are available for managing incoming alert data from a Southbound Gateway integration:

- EventAdmin
- EventModel

The EventAdmin model type is designed to represent a third-party system sending alerts to DX NetOps Spectrum. Each instantiated EventAdmin model represents an individual instance of a running external management application. Specific events that have come from the third-party application are sent to the EventAdmin. The EventAdmin models have built-in functionality that receives the events and appropriately transfers the event data to another model, with optional alarm creation. The event data is typically transferred to EventModels, but it is also possible to transfer to device models. EventAdmin models are also containers that are used to group one or more instantiated EventModel models.

The EventModel model type represents a unique source of event data within the system that the EventAdmin application manages. Each event that has been received through the Southbound Gateway contains information that uniquely identifies its source. The EventAdmin receives the event, finds the unique event source, and passes the event to the EventModel for the source.

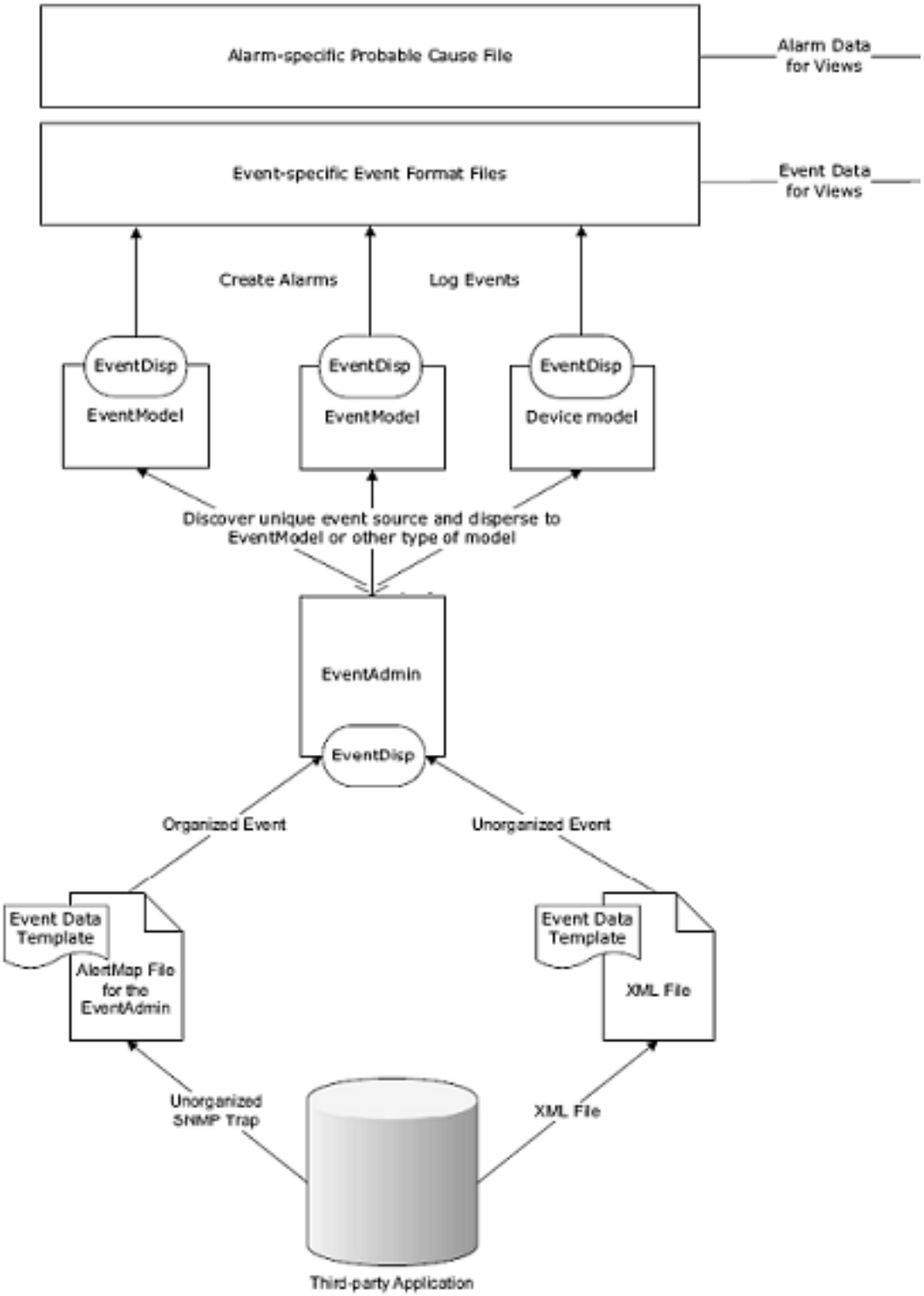
If no EventModel exists for the source, DX NetOps Spectrum creates an EventModel model to represent it. All new EventModel models are placed in the corresponding EventAdmin container model. You can cut or copy EventModel models from an EventAdmin container model. You can then paste them into other types of container models in the Topology, Location, or Organizational view. You can also cut or copy the EventAdmin model and paste it into those same views.

Southbound Gateway Support Files

As the Southbound Gateway integrator, you create or add data to support files that supply information to the EventAdmin and EventModel models. You can work with the following files:

- **AlertMap file**
  If the alert source sends SNMP traps, you create a text file that adds data to the EventAdmin AlertMap file. These additions let the AlertMap file receive the SNMP traps and convert them to DX NetOps Spectrum events.
- **XML file**
  If the alert source does not send SNMP traps, you create an XML file to send events to DX NetOps Spectrum. The syntax of this XML file follows the document type definition (DTD) provided with this toolkit.
- **EventDisp file**
  You create text files that add data to the EventDisp files at the EventAdmin and EventModel level. These additions define how the events are processed.
- **Event Format file**
  You create Event Format files to give supporting textual information about each event.
- **Probable Cause file**
  You create Probable Cause files to give supporting textual information about each alarm.

The following diagram illustrates how data flows from the third-party system into DX NetOps Spectrum through the Southbound Gateway. It shows the model types and support files working together to process the data.


Alarm-specific Probable Cause File
Alarm Data for Views
Event-specific Event Format Files
Event Data for Views
Create Alarms
Log Events
EventDisp
EventModel
EventDisp
EventModel
EventDisp
Device model
Discover unique event source and disperse to EventModel or other type of model
EventAdmin
EventDisp
Organized Event
Unorganized Event
Event Data Template
AlertMap File for the EventAdmin
Event Data Template
XML File
Unorganized SNMP Trap
XML File
Third-party Application

Integration Steps

To set up a Southbound Gateway integration, take the following three required steps:

- Direct Alert Information from the Third-Party System to DX NetOps Spectrum.
- Map the Alert Information to a DX NetOps Spectrum Event.
- Define Support Files for the Event.

Direct Alert Data to the Product

The method to set up and configure a third-party system to send alert data to the Southbound Gateway depends on the individual system. The most important decision to make is the type of alert that the third-party system sends.

If the system can send SNMP traps to DX NetOps Spectrum, perform the required configuration. The application must send the trap data to the host name and TCP port where the SpectroSERVER is listening. By default, the SpectroSERVER is configured to receive SNMP traps on port 162. DX NetOps Spectrum processes the traps that it receives from the third-party application in accordance with the AlertMap file that is associated with the EventAdmin model type. The AlertMap file translates the trap data into a DX NetOps Spectrum event. Create a text file that adds the appropriate data to the AlertMap file so that it can properly process the trap information.

If your application cannot send SNMP alert data, you can import your data using the Southbound Gateway import tool. This tool requires data formatting using the XML elements from the document type definition in the Southbound Gateway toolkit.

Map Alert Information to a DX NetOps Spectrum Event

Mapping the Alert Information to a DX NetOps Spectrum Event

Regardless of whether the third-party application can send SNMP alerts, use the Southbound Gateway Event Data Template to map the data from the trap or alert to the DX NetOps Spectrum event. This template, which is described in the following table, indicates how to organize the alert variables. The template also lets you specify variables that constitute an identifier for the unique alert source. An event variable ID is assigned to the alert variables. Identification of each unique alert source lets the integration route the event to the proper EventModel model. Event Variables 1 - 6 designate the variable alert data that uniquely identifies the alert source. Other variable alert data is mapped to the Event Data Template variable that matches its content.

Variable ID	Name	Type	Description	Required/Optional
1	Unique ID 1	String/Integer	The Unique ID variables are used to identify a target EventModel by a Unique Identifier string. The Unique Identifier is a composite of up to 6 variable data items (1-6). The final unique identifier string is composed as follows: <1>_<2>_<3>_<4>_ <5>_<6> Use that exact order. If one of the unique identifier components is not provided, it is not included within the composite unique identifier. If an existing EventModel with the Unique Identifier string cannot be found, one is created.	Either one unique identifier or a Target Name (field 7) or Target Address (field 8) is required.

2	Unique ID 2	String/Integer	The Unique ID variables are used to identify a target EventModel by a Unique Identifier string. The Unique Identifier is a composite of up to 6 variable data items (1-6). The final unique identifier string is composed as follows: <1>_<2>_<3>_<4>_ <5>_<6> Use that exact order. If one of the unique identifier components is not provided, it is not included within the composite unique identifier. If an existing EventModel with the Unique Identifier string cannot be found, one is created.	Either one unique identifier or a Target Name (field 7) or Target Address (field 8) is required.
3	Unique ID 3	String/Integer	The Unique ID variables are used to identify a target EventModel by a Unique Identifier string. The Unique Identifier is a composite of up to 6 variable data items (1-6). The final unique identifier string is composed as follows: <1>_<2>_<3>_<4>_ <5>_<6> Use that exact order. If one of the unique identifier components is not provided, it is not included within the composite unique identifier. If an existing EventModel with the Unique Identifier string cannot be found, one is created.	Either one unique identifier or a Target Name (field 7) or Target Address (field 8) is required.
4	Unique ID 4	String/Integer	The Unique ID variables are used to identify a target EventModel by a Unique Identifier string. The Unique Identifier is a composite of up to 6 variable data items (1-6). The final unique identifier string is composed as follows: <1>_<2>_<3>_<4>_ <5>_<6> Use that exact order. If one of the unique identifier components is not provided, it is not included within the composite unique identifier. If an existing EventModel with the Unique Identifier string cannot be found, one is created.	Either one unique identifier or a Target Name (field 7) or Target Address (field 8) is required.
5	Unique ID 5	String/Integer	The Unique ID variables are used to identify a target EventModel by a Unique Identifier string. The Unique Identifier is a composite of up to 6 variable data items (1-6). The final unique identifier string is composed as follows: <1>_<2>_<3>_<4>_ <5>_<6> Use that exact order. If one of the unique identifier components is not provided, it is not included within the composite unique identifier. If an existing EventModel with the Unique Identifier string cannot be found, one is created.	Either one unique identifier or a Target Name (field 7) or Target Address (field 8) is required.
6	Unique ID 6	String/Integer	The Unique ID variables are used to identify a target EventModel by a Unique Identifier string. The Unique Identifier is a composite of up to 6 variable data items (1-6). The final unique identifier string is composed as follows: <1>_<2>_<3>_<4>_ <5>_<6> Use that exact order. If one of the unique identifier components is not provided, it is not included within the composite unique identifier. If an existing EventModel with the Unique Identifier string cannot be found, one is created.	Either one unique identifier or a Target Name (field 7) or Target Address (field 8) is required.

7	Target Name	String	This field lets you specify a target model by model name. An EventModel is not created when a target model cannot be found while using the Target Name event variable. This field is case-sensitive. The 'Target Name Case Insensitive' field that is defined by Variable ID Field 16 instructs Southbound Gateway not to consider case when identifying the model by model name.	
8	Target Address	Octet/Text String	This field lets you specify a target model by IP address. Use this field if you want to send the event to a model other than an EventModel.	
9	Reserved			
10	Event Model Name	String/Intege	This field lets the user give a model name that is different than the unique identifier. If this data is not provided, the composite unique identifier becomes the model name.	Optional
11	Model Class	Integer	Populates the Model Class attribute of the target EventModel with the value specified in this event variable	Optional
13	Network Address	Octet/Text String	Populates the Network Address attribute of the target EventModel with the value specified in this event variable	Optional
14	MAC Address	Octet/Text String	Populates the MAC Address attribute of the target EventModel with the value specified in this event variable.	
15	Manufacturer	String	Populates the Manufacturer attribute of the target EventModel with the value specified in this event variable.	Optional
16	Target Name Case Insensitive	String	This field lets you specify a target model by model name. An EventModel is not created if a target model cannot be found when using the Target Name Case Insensitive event variable. This field is case-insensitive. If you want Southbound Gateway to consider the case when identifying the model by model name, use the Target Name field defined by Variable ID Field 7 (Target Name), which is case sensitive.	
17-99	Reserved			
Any other variable ID greater than or equal to 100.	Any Data	Any Type	This data is forwarded to the EventModel model unchanged. The data type and data are preserved. This data can be viewed within an event message.	Optional

Define Support Files

The following topics are covered in this section:

Defining Support Files

Once the alert has been translated into an event, define information for the support files. The EventDisp files, Event Format files, and Probable Cause files enable the correct processing and display of the alert data.

EventDisp Files

EventDisp files define DX NetOps Spectrum handling for each event. Two different EventDisp files support the EventAdmin model type and the EventModel model type.

The EventDisp file for the EventAdmin model type defines the events that are received from the external alert or trap source. The EventDisp file for the EventModel gives the desired properties to the events. This support file indicates whether the event is logged, the event severity, and whether the event becomes an alarm.

Event Format Files

Event Format files determine the contents of an event message when the event is displayed in OneClick. The These files can contain variable data from a specific occurrence of an event. The assigned event variable ID from the Event Data Template identifies this variable data within the Event Format file.

Probable Cause Files

Alarms that are generated as a result of the EventDisp file usually contain a probable cause as part of the alarm message. The text of this probable cause information is defined in the Probable Cause file.

References about Working with the Southbound Gateway Toolkit

Further References

For further information about working with the Southbound Gateway toolkit, see the following:

- Southbound Gateway Toolkit provides in-depth information about Southbound Gateway usage.
- Getting Started provides an overview of the DX NetOps Spectrum technology and the terminology that is used throughout the DX NetOps Spectrum documentation.
- Extension Integration Developer provides details about using SEI to package an integration for distribution.
- Event Configuration provides an in-depth look at AlertMap EventDisp, Event Format, and Probable Cause files.

Extracting Alarm Data

This section includes information about how to extract alarm data from DX NetOps Spectrum.

AlarmNotifier

DX NetOps Spectrum AlarmNotifier is a SpectroSERVER client application that provides a way to extract alarm information from DX NetOps Spectrum. The AlarmNotifier tool is launched from the command line and connects to a single SpectroSERVER. Once connected, it monitors alarm information from that server.

AlarmNotifier uses three scripts to get data from alarms:

- SetScript
- ClearScript
- UpdateScript

AlarmNotifier receives information from the SpectroSERVER when an alarm is sent, cleared, or updated. AlarmNotifier invokes a relevant script in response to alarm data. These scripts, which are located in the DX NetOps Spectrum Notifier directory, enable the display of alarm data. You can customize each script to send alarm data in an email notification or to another third-party application, such as a trouble ticketing system. The scripts contain parameters to reduce the amount of information that is displayed or is sent to other applications.

DX NetOps Spectrum Alarm Notification Manager (SANM)

The following topics are covered in this section:

About the DX NetOps Spectrum Alarm Notification Manager (SANM)

DX NetOps Spectrum Alarm Notification Manager (SANM) supports and enhances the AlarmNotifier application. SANM lets the AlarmNotifier monitor alarm information from all landscapes in the SpectroSERVER landscape map. SANM also provides more alarm data for the AlarmNotifier. When SANM has been installed on the SpectroSERVER, SetScript, ClearScript, and UpdateScript have additional parameters to provide detailed alarm data. SANM lets you acknowledge and clear alarms from the terminal shell where the alarm data is displayed. You can control the scope and volume of alarm data using the SANM Policy Administrator tool.

The Policy Administrator tool is the only way to work with SANM directly. SANM works directly with the AlarmNotifier to extend its functionality. You continue to work with the AlarmNotifier using the added SANM capability. For more information, see Alarm Notification Manager .

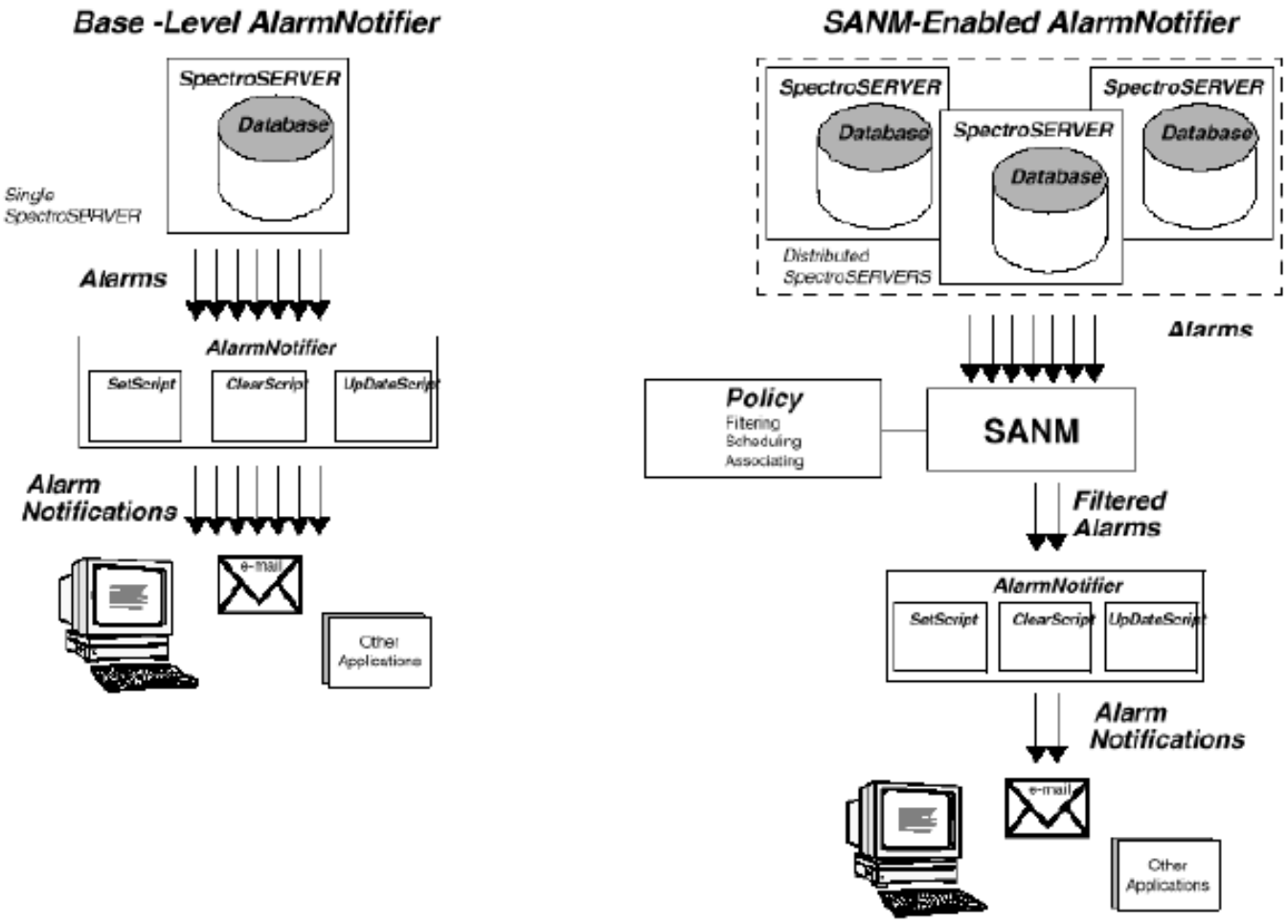
Policy Administrator Tool

The Policy Administrator configuration tool lets you create, save, and implement alarm notification filtering policies. You can create policies that specify the types of alarms that are sent to an alarm-processing application. Or you can create policies that exclude irrelevant or unimportant alarms or alarm data. You can also schedule associations between policies and applications so that alarm data filtered through that policy is sent to the particular application during the specified time period.

Policies can be very simple or quite complex. A simple policy can instruct SANM to pass information about critical alarms for all routers to an application that is associated with the policy. A more complex policy can instruct SANM how to handle information about critical and major alarms that remain unresolved for 10 minutes for a device with specific characteristics.

SANM Architecture

The following diagram shows the architecture of AlarmNotifier and SANM.



Further References

Further References about AlarmNotifier and SANM

For more information about working with AlarmNotifier and SANM, see the following sections:

- Alarm Notification Manager gives in-depth instructions about how to use SANM to enhance the capabilities of the AlarmNotifier.
- AlarmNotifier provides detailed instructions about how to use the AlarmNotifier and how to customize the AlarmNotifier scripts.
- Getting Started provides an overview of the DX NetOps Spectrum technology and the terminology that is used throughout the DX NetOps Spectrum documentation.

Sending Topology Data to the Product

Modeling Gateway Architecture

The following types of input files can be used with the Modeling Gateway:

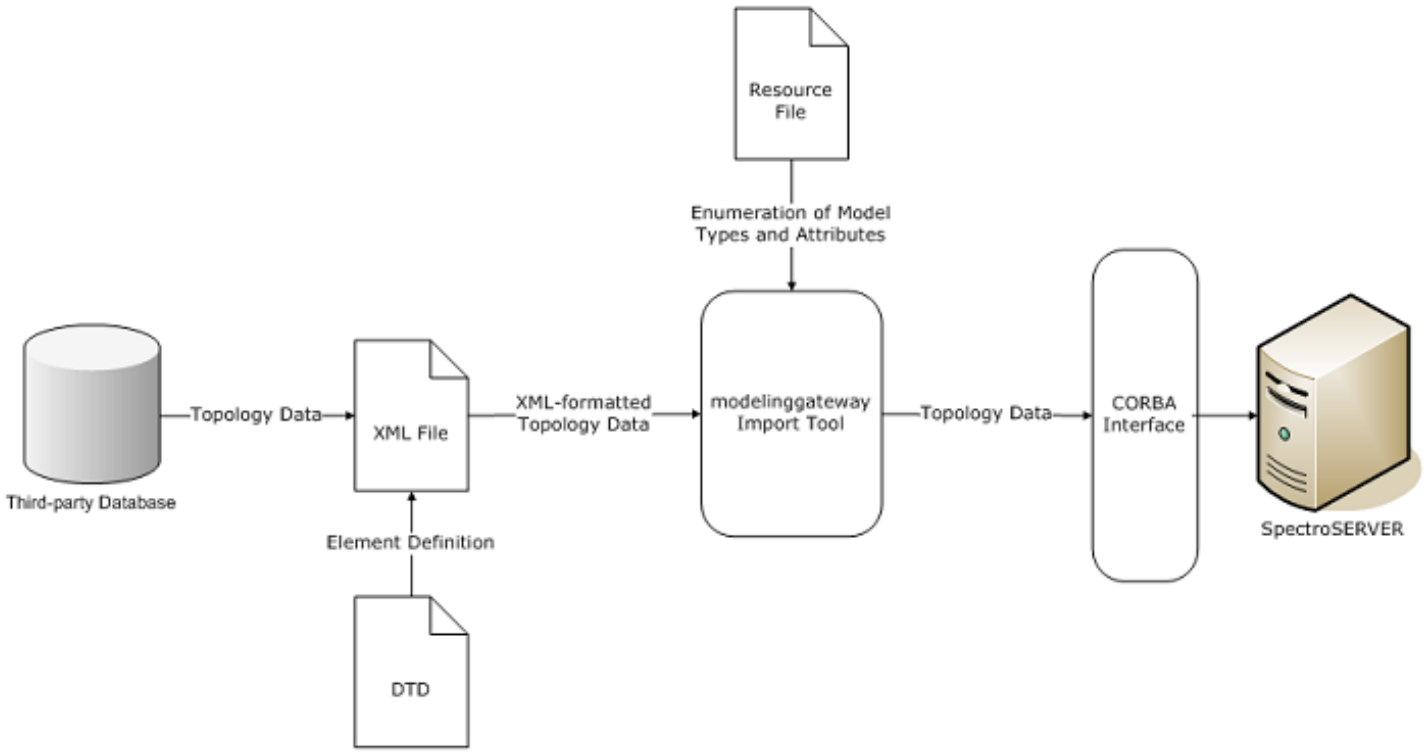
- XML file
- comma-delimited ASCII text file

The comma-delimited input file is used for creating connections for Frame Relay and ATM circuits. The XML input file provides a wider range of functionality. In addition to specifying Frame Relay and ATM circuit connections, you can also create, update, and destroy models and other types of connections.

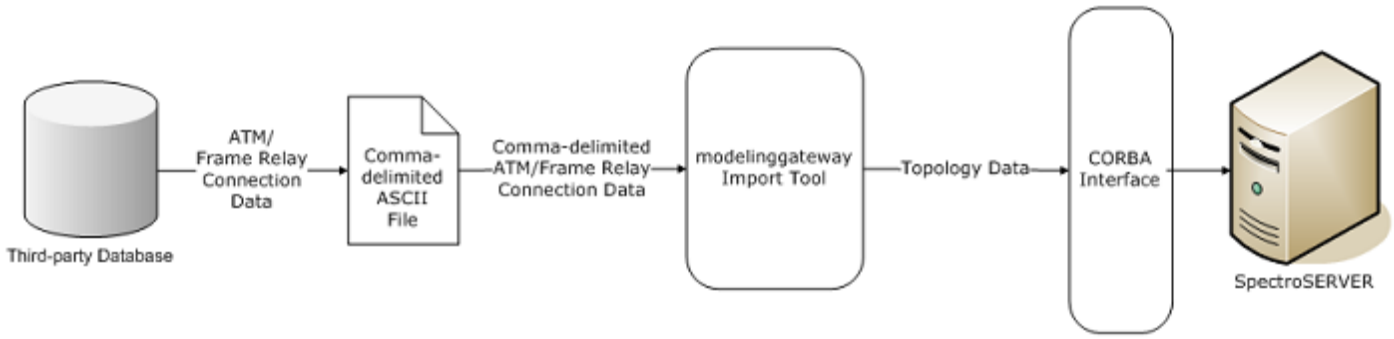
The integrator creates the XML input file that structures the network data for import. If you have a thorough understanding of the network topology, you can more easily represent network data in XML format. DX NetOps Spectrum provides a Document Type Definition (DTD) file that defines the XML elements and their associated syntax rules. A second file, the .modelinggatewayresource.xml file, shows you the DX NetOps Spectrum model types and attributes that are available. Create the XML input file, and then use the import tool (modelinggateway) to import the data into the SpectroSERVER database.

NOTE
For more information, see the Modeling Gateway Toolkit *section.*

The following diagram shows how the Modeling Gateway uses an XML file for importing data. Data that flows from the third-party database is formatted in the XML file using the syntax from the DTD and .modelinggatewayresource.xml. The import tool interprets the XML file and sends data into DX NetOps Spectrum through the DX NetOps Spectrum CORBA interface.



The following diagram illustrates the process for using a comma-delimited ASCII text file to import Frame Relay and ATM connection information.



NOTE
For more information about the comma-delimited file, see the Modeling Gateway Toolkit *section.*

Modeling Gateway Integration Mechanics

Integration requires information about how network data is formatted when it is imported. The modeling gateway uses input files to format the data that it imports. Create input files that specify data formatting rules.

NOTE
The following sections assume that you have a basic knowledge of XML and Document Type Definition files (DTDs). If you lack experience working in XML, we recommend becoming familiar with XML syntax before working with these files.

Gateway Modeling Tool for Input

During the integration process, you take your network data from a third-party database and create an input file. Depending on the content, this input file can be an XML file or a comma-delimited ASCII file.

If you only want to import data from Frame Relay or ATM connections, use the comma-delimited file.

Use an XML file to import more complex topology information. Use the DX NetOps Spectrum DTD file and the modelinggatewayresource.xml file to format the data properly. The following section contains an overview of each of these files.

NOTE
Note: For more information, see the Modeling Gateway Toolkit *section.*

Document Type Definition (DTD) File

The DTD file (.modelinggateway.dtd) outlines all of the elements and attributes that can be contained in the XML file. This file shows you the structural hierarchy of elements, indicating which elements can act as child elements for any given element and which attributes can be specified for each element.

The root element in the DTD is the Import element. This required element specifies the type of import to perform. The Import element has several possible child elements such as Topology, Location, GenericView, Update, Destroy, and Connection. To import new models, use the Topology, Location, or GenericView elements. The element that you select determines the OneClick view where the imported data appears.

Use the Update element to change an existing collection of models. Use the Destroy element to remove existing models, and use the Connection element to establish connections between ports or devices. These elements all have child elements, which define the components of the network.

You can customize the DTD file to meet the needs of your specific integration. For example, the GenericView element lets you create your own view in which to place container and device models. You can then modify and add to the child elements of the GenericView element to define the contents of your customized view.

The .modelinggatewayresource.xml File

All of the possible model types and attributes that can be used in the XML input file are defined in the .modelinggatewayresource.xml file. This file lists many DX NetOps Spectrum model types and the model type handles that uniquely identify them. It also lists many DX NetOps Spectrum attributes and their attribute IDs. This file lets you use model type and attribute names in the XML file rather than their hexadecimal identifiers. As a result, the XML file is more intuitive to create and read. The .modelinggatewayresource.xml file can be modified to include any instantiable, visible model type that exists in DX NetOps Spectrum and any attribute.

Gateway Modeling Tool for Import

Once you have set up the input file, use the Modeling Gateway import tool to import the network data into the SpectroSERVER database. The import tool is a command-line utility that is located in the SS-Tools directory.

> **NOTE**
> Copy the modelinggateway tool and all of its supporting files to run it on another server. For more information, see the *Distributed SpectroSERVER* Administrator *section.*

The following example command shows the modelinggateway tool syntax for import.

```
modelinggateway -vnm <vnm_name> -i <import_file> [-o <outputfile>] [-debug]
```

- **-vnm *vnm_name***
  Is the name of the SpectroSERVER host.
- **-i *import_file***
  Is the filename of the XML file with the input information that is compiled with .modelinggateway.dtd.
- **-o *outputfile***
  (Optional) Logs the error information to the file named in the outputfile parameter. If no file is specified, error information is logged to a file named import_file.log, where 'import_file' is the name of the XML input file.
- **debug**
  (Optional) Creates a debugging output file during the import process. The debug file is named ModelingGatewayDebug.txt in the running directory.

View Import Status and Import Errors

The DX NetOps Spectrum Modeling Gateway provides mechanisms to verify the safety and accuracy of each database import operation. DX NetOps Spectrum Modeling Gateway maintains an audit trail that includes a record of each creation, deletion, association, and update. You can view information about the import from within OneClick. You can also track import problems in the error and debug logs.

Verify the results of the Modeling Gateway import from the Information Tab for the VNM model in OneClick.

Follow these steps:

1. Select the applicable VNM model in the OneClick Console.
2. Click the Information tab in the Component Detail view.
3. Expand the Modeling Gateway node.
4. Review the table in the Modeling Gateway section for information about recent imports.

All errors and their possible causes are logged in an error log file. By default, the import tool creates an error log named <*nameofimportfile*>.log, where <*nameofimportfile*> is the name of your import file. You can also specify a name for your log file. For more information, see Import Tool.

When the import is complete, the log file appears in the SS-Tools directory. The log file records the number of successful creations, deletions, and updates of models and connections and each failure that occurred during the import.

The import tool also lets you enable a debug log, which is also created in the SS-Tools directory. This file, ModelingGatewayDebug.txt, explains each step in the import process.

NOTE
For more information see the Modeling Gateway Toolkit *section.*

Further References - Using the Modeling Gateway

For more information about importing topology data using the Modeling Gateway, see the following spaces:

- Modeling Gateway Toolkit provides information about the toolkit and the syntax used in its support files.
- Getting Started provides an overview of the DX NetOps Spectrum technology and the terminology that is used throughout the DX NetOps Spectrum documentation.
- Distributed SpectroSERVER Administration provides information about moving the modeling gateway tool to another server.

Exporting Topology Data from the Product

Configure Topology Data for Export

The .modelinggatewayresource.xml file controls the content of the exported information. All types of data are exported by default. To limit the data that is exported, use the ExportConfiguration element. This element contains the following attributes:

- **export_devices**
  Exports device models.
- **export_containers**
  Exports container models.
- **export_port_attributes**
  Exports port attributes.
- **export_links**
  Exports device links.
- **export_topology_layout**
  Exports device and container models x,y coordinates.
- **export_annotation**
  Exports annotations and model group information.
- **export_WA_Link_models**
  Exports WA_Link models. If you do not export WA_Link models, they are treated as transparent. Wide area links between two device models are exported as a direct link.
- **export_spectrum_settings**
  Exports DX NetOps Spectrum settings, such as the settings for Fault Isolation, Discovery, and VNM Control.
- **export user models**
  Exports user models, licenses, privileges, preferences, and all other user-related relations attributes and models.
- **export service modeling**
  Exports service management schemes and attributes.

NOTE
For more information, see the Service Manager User section.

- **export schedules**
  Exports schedules.
- **export global collections**
  Exports static and dynamic global collections including all models in each global collection, all dynamic collection criteria, zoomed list, grouped list, and topology layout.
- **export discovery configs**
  Exports Discovery configurations.
- **export from primary SpectroSERVER only**
  Specifies whether Modeling Gateway connects to the secondary SpectroSERVER when the primary SpectroSERVER is down.

```
<ExportConfiguration
  export_devices = "true"
  export_containers = "true"
  export_port_attributes = "true"
  export_links = "true"
  export_topology_layout = "true"
  export_annotation = "true"
  export_WA_Link_models = "true"
  export_spectrum_settings = "true"
  export_user_models = "true"
  export_service_modeling  = "true"
  export_schedules = "true"
  export_global_collections = "true"
  export_discovery_configs = "true"
  export_from_primary_ss_only = "true"
/>
```

For example, to exclude port attribute information, set export_port_attributes to false.

The attributes that are exported for devices, containers, and ports are defined in the following elements:

- DeviceExportAttributes
- ContainerExportAttributes
- PortExportAttributes

Add and subtract attributes from these elements as needed.

The SpectrumConfigurationExport element controls what types of DX NetOps Spectrum configuration data are exported if the export_spectrum_settings flag in the ExportConfiguration element is set to true. For example, the following element controls the attributes that are exported for the LostFound model:

```
<SpectrumConfigurationExport model_type="LostFound" >
        <Automatic_Model_Destruction attribute_id="0x11de1" />
        <Model_Destruction_Interval_Hours attribute_id="0x11de3" />
        <Model_Destruction_Interval_Minutes attribute_id="0x11de4" />
</SpectrumConfigurationExport>
```

By default, all topology and modeling information under the Universe container is exported. Modify the RootContainerToExport element to specify a different root container from which to export. All of the contents of the root container and each of its subcontainers are exported.

NOTE
For more information, see the embedded comments in the *.modelinggatewayresource.xml* file.

Gateway Modeling Tool for Export

The Modeling Gateway command-line tool (modelinggateway, or modelinggateway.bat on Windows) is located in the SS-Tools directory. The syntax for export is:

```
modelinggateway -vnm <vnm_name> -e <export_file> [-o <outputfile>] [-debug]
```

For more information, see Gateway Modeling Tool for Import.

> **NOTE**
> Copy the modelinggateway tool and all of its supporting files to run it on another server. For more information, see the *Distributed SpectroSERVER* Administrator *section.*

Export DX NetOps Spectrum Topology Data

You can export DX NetOps Spectrum topology data using the modelinggateway tool. Use the -e flag to export DX NetOps Spectrum topology data.

For example, running the following command exports the data from the SpectroSERVER on NOC1_DX NetOps Spectrum into a Modeling Gateway formatted xml file named NOC1_data.xml:

```
modelinggateway -vnm NOC1_Spectrum -e NOC1_data.xml
```

Import Modeling Gateway XML file

You can import the data from a Modeling Gateway formatted XML file into DX NetOps Spectrum. Use the -i flag to import from a Modeling Gateway formatted xml file.

For example, running the following command imports the data from NOC1_data.xml into the SpectroSERVER at NOC2_Spectrum.

```
modelinggateway -vnm NOC2_Spectrum -i NOC1_data.xml
```

Launching Applications from OneClick

The following section details how you can launch application from the DX NetOps Spectrum OneClick.

Launch Applications from the OneClick Interface

A configuration file contains examples for adding custom menus and menu items to your OneClick Console. Find this file in the following directory:

```
<$SPECROOT>/tomcat/webapps/spectrum/WEB-INF/console/config/custom-menu-config.xml
```

Copy this file into the <*$SPECROOT*>/custom/console/config/ directory. Add your custom menu items to this XML file.

The <launch-application> element lets you launch a specified executable and pass attribute values to that executable.

```
<launch-application>
    <command>myapp {0}</command>
    <param>
        <attribute>AttributeID.NETWORK_ADDRESS</attribute>
    </param>
</launch-application>
```

The previous example launches an application on the client computer. The <command> element specifies the command or executable to run. You can provide the path to the command or executable in either of the following ways:

- You can specify the path on each client using an environment variable. On Windows, select My Computer, Properties, Advanced, and then click Environment Variables.
- You can specify an absolute path to the executable or command. Verify that the path is the same on each OneClick client. Path statements in the Windows environment use a double backslash instead of a single backslash. For example:
  C:\\Windows\\system32\\cmd.exe

You can pass multiple parameters to the launched application or URL. The previous example launches an application named myapp and passes the IP address of the selected model.

The command element must conform to the following syntax rules:

- Use only spaces to delimit command arguments. To include a space within an argument, either place quotation marks around the argument or use the escape character '\' before the space.
- To embed quotes within an argument, place the escape character '\' before the quotation marks.
- If your command arguments contain commas, DX NetOps Spectrum automatically places the argument within quotation marks. For example, an argument is a numeric value containing commas.
- DX NetOps Spectrum replaces arguments that come back null or have a string length of zero with empty quotes.

The following example uses the <platform> element to specify commands for different platforms.

Example: <launch-application> Code

```
<launch-application>
   <platform>
   <os-name>Windows</os-name>
   <command>cmd.exe /c start "ping {0}" cmd /c "ping.exe {0}
   &#38;&#38;pause"</command>
   </platform>
   <platform>
   <os-name>SunOS</os-name>
   <command>>/usr/dt/bin/dtterm -e ping {0}</command>
   </platform>
   <param>
   <attribute>AttributeID.NETWORK_ADDRESS</attribute>
   </param>
</launch-application>
```

- **<os-name>**
  (Optional) Specifies the operating system
- **<command>**
  Specifies the command to execute on the specified operating system.

If you do not specify the <os-name>, the associated command is executed if no other platforms match.

At run time, DX NetOps Spectrum compares the specified OS names to the name that the ‘os.name’ Java property returns. DX NetOps Spectrum uses a best-match algorithm that lets you specify a prefix. The following operating system names are accepted:

- Windows for all Windows platforms
- Windows 9x for Windows 95/98 only
- Windows 2000 for Windows 2000 only
- Windows XP for Windows XP only
- Linux for the Linux platform
- Mac for the Macintosh platform

If none of the specified platforms matches, the associated menu item is disabled.

NOTE
For more information, see OneClick Customization .

Launch URLs in Web Browsers from the OneClick Interface

You can configure OneClick to launch specified URLs and to pass parameters to the URL. These parameters can be hard-coded values or values from model attributes.

NOTE
For more information, see OneClick Customization .

Launch OneClick Clients with Context

You can launch OneClick clients within the context of a certain topology, model, or a Global Collection. Contextual parameters and values can be included with the URL that launches OneClick. The following URL is an example of such parameters:

http://<*hostname*>/spectrum/oneclick.jnlp?<*parameter*>=<*value*>

The following parameters are available:

- **Topology**
  Supply a model handle, IP address, or name of a Global Collection. This parameter launches a OneClick client or reuses an existing one, selects the Explorer tab or the Topology tab, expands the tree to show the model, or selects the specified model in the Topology panel.
  Examples
  - http://<*hostname*>/spectrum/oneclick.jnlp?topology=0x3780003d
  - http://<*hostname*>/spectrum/oneclick.jnlp?topology=10.253.9.7
  - http://<*hostname*>/spectrum/oneclick.jnlp?topology=East
- **Explorer**
  Supply a model handle, IP address, or name of a Global Collection. This parameter launches a OneClick client or reuses an existing one, selects the Explorer tab, or expands the tree to show the model. The currently selected tab in the Contents panel reflects the new model.
  Examples
  - http://<*hostname*>/spectrum/oneclick.jnlp?explorer=0x3780003d
  - http://<*hostname*>/spectrum/oneclick.jnlp?explorer=10.253.9.7
  - http://<*hostname*>/spectrum/oneclick.jnlp?explorer=East
- **Alarm**
  Supply the integer alarm ID (to facilitate the integration with legacy applications), the complete global alarm ID (in the form 3f983d3d-2045-1000-012b-000bdb5a1c31), or <model handle>@<alarm ID>. This parameter launches a OneClick client or reuses an existing one, selects the Explorer or Alarms tab, expands the tree to show the model, and selects the alarm.
  Examples
  - http://<hostname>/spectrum/oneclick.jnlp?alarm=0x3780003d@7710
    where 0x3780003d@7710 is <modelhandle>@<alarm ID> and
  - http://<hostname>/spectrum/oneclick.jnlp?alarm=7710
    where 7710 is the integer <alarm ID>

If you are passing the integer alarm ID, pass the model handle also. The integer alarm ID is not always unique across SpectroSERVERs. The full global alarm ID, which is unique across SpectroSERVERs, is preferable. However, that ID is not always available to the application that is launching OneClick.

NOTE
When launching in context, a new instance of OneClick is not launched if an instance is already running on the host. Instead, the context is changed in the current instance of OneClick. For more information, see the OneClick Administrator.

Launch Applications with Process Daemon

A process launching and tracking daemon (processd) lets DX NetOps Spectrum control various processes that are running on a workstation. This daemon operates in the background. The processd automatically starts during DX NetOps Spectrum installation and whenever the system restarts. Once processd is started, it automatically starts and manages other processes. You can create an application that processd can launch as part of the DX NetOps Spectrum startup process.

The lib/SDPM/partslist directory is used by processd to determine executable locations and arguments. For application tracking, processd uses the lib/SDPM/runtime directory. The partslist directory lists everything that the process daemon can launch and monitor. The runtime directory lists everything that processd is tracking.

To let processd launch and track your application, create an install ticket file (.idb). Save the file in the partslist directory.

NOTE
For more information, see Distributed SpectroSERVER Administration.

Further References - Launching Applications from DX NetOps Spectrum

For more information about launching applications from DX NetOps Spectrum, see the following spaces:

- OneClick Customization provides details about launching applications in OneClick.
- OneClick Administration provides details about launching OneClick with context.
- Distributed Administration provides a complete reference for launching applications.
- Getting Started provides an overview of the DX NetOps Spectrum technology and the terminology that is used throughout the DX NetOps Spectrum documentation.

Create New Management Modules

Creating New Management Modules

The following section provides information and procedures to create new Management Modules:

Model Type Editor

To derive new model types and import MIBs, use the Model Type Editor. This tool is available from the Configure menu in the DX NetOps Spectrum Control Panel. The Model Type Editor lets you select the model type for your new model type and import proprietary MIBs. It also allows you to set selected attribute values within the model type.

NOTE
For more information, see Model Type Editor .

OneClick Topology Support

After creating model types using the Model Type Editor, you can specify how they appear in OneClick topologies using XML. The following list describes the customizations that you can perform:

- Create icons for model types
- Register model icons
- Configure icons for OneClick themes
- Design on and off-page reference icons
- Create icon labels
- Define text components
- Define selection components
- Define tool tips for model icons

NOTE
For more information, see OneClick Customization.

Event Configuration Files

Once you have edited the appearance of the new model type, build the support files that let the model type receive and process SNMP traps. These Event Configuration files include the AlertMap file, the EventDisp file, and all applicable Event Format and Probable Cause files. The AlertMap file maps SNMP alerts to DX NetOps Spectrum events. The EventDisp file contains information about event handling. For example, the event can be logged, can be turned into an alarm, or can be used to clear an alarm. The EventFormat file provides information about the event, and the Probable Cause file provides information about the alarm.

Use a text editor to edit any of these files. You can also use the Event Configuration tool to edit EventDisp, Event Format, and Probable Cause files.

NOTE
For more information, see Event Configuration

DX NetOps Spectrum Extension Integration (SEI) Toolkit

After you have created the support files, use the DX NetOps Spectrum Extension Integration (SEI) toolkit to create a virtual CD. The virtual CD installs your new management model on any SpectroSERVER.

Watches

When models of your newly derived model type have been instantiated, you can track and analyze information about their condition. Create one or more watches for a particular model to monitor and analyze the internal and external attribute values of that model. Watches can include expressions that incorporate one or more attribute values. These attribute values, or an expression that is based on these values, can then be measured against a defined threshold value.

Watch results can be used to generate events and alarms and can be logged for historical tracking and report information. Results can also be sent to script files. DX NetOps Spectrum polls attributes to evaluate the attribute values that are defined in a watch. Keep in mind that this polling can have an impact on network traffic and system resources. Delete the watches that are no longer useful.

NOTE
For more information, see Watches.

Further References - Using the GnSNMPDev Model Type

For more information about using the GnSNMPDev model type to create new model types, see the following DX NetOps Spectrum spaces:

- Certification provides complete information about the GnSNMPDev management module.
- Model Type Editor gives in-depth instructions for using the Model Type Editor.
- Getting Started provides an overview of the DX NetOps Spectrum technology and the terminology that is used throughout the DX NetOps Spectrum documentation.
- OneClick Customization describes how to create your own DX NetOps Spectrum views or customize generic DX NetOps Spectrum views.
- Extension Integration Developer provides details about using SEI to package integration for distribution.
- Event Configuration describes the Event Configuration utility, which lets you edit EventDisp, Event Format, and Probable Cause files from a graphical user interface. It also provides an in-depth look at EventDisp, Event Format, and Probable Cause files.
- Watches provides information about creating a watch. This space provides examples for generating events or alarms, sending this information to a script, or using this information in a report.

Distribute Integration Files

Distributing Integration Files:

Create an Index File

An index file defines the components of an integration, identifies their location, and specifies where to install them on the DX NetOps Spectrum server. Index files include references to all files that are necessary for the operation and installation of the integration on the DX NetOps Spectrum host. Depending on the type of integration that you have created, the index file includes references to different types of files.

You can include scripts in the index file that support the installation process. Custom installation scripts are launched during installation. These scripts can copy the contents of an installable text file into an existing text file on the DX NetOps Spectrum host, for example.

Create index files manually, or use the mmship tool.

Run mkmm

After creating your index file, build a package that contains all of the files that are referenced. Use the mkmm tool to build this package. This tool references the index file to create a VCD that contains all of files that must be installed on the SpectroSERVER.

The mkmm tool is saved in the SPECTRUM INSDK directory. However, you run this tool from the directory where your index file is saved. To run the tool, type the following command:

```
../INSDK/mkmm *.i
```

Run mkcd

The mkcd tool performs three final functions to complete an installable integration file. This script checks dependencies between purchasable parts and extension modules on the virtual CD. Any inconsistencies or errors are reported. The mkcd script adds a version number to the VCD, which enables installation. And it locks the VCD against the addition of files. The mkcd tool is located in the SPECTRUM INSDK.

Use the following syntax to execute the command in the directory that contained the VCD directory:

```
../INSDK/mkcd  -f <vcddir><version>
```

- **<vcddir>**

Specifies the name of the vcd directory that mkmm created (usually vcd).

- **<version>**
  Specifies the version number of your integration. The version number must not contain spaces.

Further References - SEI Toolkit

For more information about the SEI toolkit, see the following sections:

- DX NetOps Spectrum Extension Integration Developer provides details about using SEI to package an integration for distribution.
- Getting Started provides an overview of the DX NetOps Spectrum technology and the terminology that is used throughout the DX NetOps Spectrum documentation.

Use CLI to Exchange Data

Using CLI Components to Exchange Data with DX NetOps Spectrum

CLI Components

The CLI includes the following four separate components:

- A set of executable commands that let you communicate with the SpectroSERVER.
- A set of environment variables that can affect executable command behavior. Depending on the operating system, they can be required to enable the script or executable command to operate.
- A daemon that maintains communication with the SpectroSERVER.
- A set of sample scripts that include CLI commands.

When you use the CLI to connect to the SpectroSERVER, the communication daemon that acts as a CLI local server starts. You can then use the executable commands and environment variables on the command line or in a script. Each CLI command sends success or failure information to standard error. Normal output that results from a successful command is sent to standard output.

Each command also generates a return code of zero on success and a non-zero error code on failure. These return codes are useful in scripts to determine the success or failure of a command.

CLI Commands

The following list describes the available executable CLI commands:

- **ack alarm**
  Acknowledges an alarm.
- **connect**
  Connects to the SpectroSERVER.
- **create**
  Creates a new alarm, association, event, or model in a specified landscape.
- **current**
  Sets a model as the 'current' model or a landscape as the 'current' landscape to be acted on by other CLI commands, or displays the current model and current landscape.
- **destroy**
  Destroys (removes) an alarm, association, or model in a specified landscape.
- **disconnect**

Disconnects from the SpectroSERVER.

- **jump**
  Returns to a model and a landscape that were saved with the setjump command.
- **seek**
  Finds a model in a specified landscape.
- **setjump**
  Saves the current model and the current landscape under a user-defined text string label. The jump command can then be used to return to this model and landscape.
- **show**
  Displays information about objects in a specified landscape.
- **stopShd**
  Disconnects all users from SpectroSERVER and terminates the CLI locale server.
- **update**
  Updates the attributes of a model or model type.

Further References - Working with CLI

For more information about working with CLI, see the following sections:

- Command Line Interface provides a complete reference to the CLI environment and its commands.
- Getting Startedprovides an overview of the DX NetOps Spectrum technology and the terminology that is used throughout the DX NetOps Spectrum documentation.

Extending with the CORBA API

Overview of the DX NetOps Spectrum CORBA API

The DX NetOps Spectrum CORBA API provides programmatic access to the SpectroSERVER. Use the API to develop C ++ or Java client applications.

The CORBA API lets you develop your own client application that exchanges data with and manipulates data contained in the DX NetOps Spectrum knowledge base. This interface is especially useful for integrating web-based (Java) applications. The DX NetOps Spectrum CORBA API consists of Interface Definition Language (IDL) files for the interfaces we provide, as well as some helper classes.

Considerations

Use the CORBA API with caution. If you can use one of the non-programmatic toolkits described in this section to accomplish the goals of your integration, you should do so. This powerful interface requires an in-depth understanding of DX NetOps Spectrum functionality and the DX NetOps Spectrum knowledge base.

If you intend to use the CORBA API to integrate client applications that share alarm and event data with DX NetOps Spectrum, consider using the Southbound Gateway, AlarmNotifier, and SANM toolkits. These toolkits are specifically designed for this purpose, so you can save considerable time and effort by using them. The toolkits do not necessarily require deep knowledge of DX NetOps Spectrum.

Further References - Working with CORBA API

For further information about working with the DX NetOps Spectrum CORBA API, see the following:

- Development API Reference is an introductory reference for the DX NetOps Spectrum CORBA API.
- Getting Started provides an overview of the DX NetOps Spectrum technology and the terminology that is used throughout the DX NetOps Spectrum documentation.

Security Policy Statement

The DX NetOps Spectrum Security Policy Statement applies to the DX NetOps Spectrum product and is applicable as long as the product is used within the documented procedures defined in the product documentation.

The DX NetOps Spectrum Security Policy Statement details the encryption and hashing that is used by specific DX NetOps Spectrum components.

The DX NetOps Spectrum Security Policy Statement communicates the FIPS 140-2 statement for the DX NetOps Spectrum product. Specifically, it does the following:

- Clearly states what DX NetOps Spectrum modules are FIPS-compliant and which are FIPS-compatible
- Identifies FIPS certificate numbers for the encryption modules or hash algorithms used
- Communicates additional items that require extra physical security or protection
- Identifies the application boundaries surrounding the different application modules using encryption and or hashing
- Identifies what data is protected
- Communicates how keys are protected
- Explains how to enable FIPS mode on the software component

Definitions

The following terms are used in the DX NetOps Spectrum Security Policy Statement:

FIPS-compliant means that the component is capable of running FIPS-compliant encryption and hashing modules and offers the ability to run in FIPS mode.

FIPS-compatible means that the component uses FIPS-certified algorithms for encryption and hashing, but does not offer the ability to run in FIPS mode.

FIPS 140-2 Compatibility Matrix

The following table shows the extent to which DX NetOps Spectrum uses FIPS-compliant algorithms:

DX NetOps Spectrum Software Component	Module	Version	Certificate1	Algorithms2	Algorithm Cert#3	Mode4
SpectroSERVER User Password Storage**	BSAFE Crypto-J	5.1.1	1502	SHA-256	1549	Compatible
OneClick User Password Storage	BSAFE Crypto-J	5.1.1	1502	AES-256	1766	Compatible
Integration Password Storage	BSAFE Crypto-J	5.1.1	714	AES-256	1766	Compatible
eHealth Password Storage	BSAFE Crypto-J	5.1.1	714	AES-256	1766	Compatible

EEM Single SignOn (Proxy Password)	BSAFE Crypto-J	5.1.1	714	AES-256	1766	Compatible
CA Service Desk Password Storage	BSAFE Crypto-J	5.1.1	714	AES-256	1766	Compatible
MySQL Password Storage	BSAFE Crypto-J	5.1.1	714	AES-256	1766	Compatible
SRAdmin Data Transmission	BSAFE Crypto-C ME	2.0	608	3DES	378	Compatible
SNMPv3 Privacy Data Transmission	OpenSSL***	0.9.8	2097	3DES, AES-128, AES-256, SHA	1302	Compatible
Secure Domain Manager*	BSAFE Crypto-C ME	2.0	608	3DES, SHA	378	Compliant
Secure Domain Manager	OpenSSL	0.9.8	2097	3DES, SHA-256	1302	Compatible
SDConnector Data Transmission	BSAFE Crypto-C ME	2.0	608	3DES	378	Compatible
Certgen	OpenSSL	0.9.8	2097	3DES, SHA	1302	Compatible

Notes:

- * You can configure a different algorithm for Secure Domain Manager (SDM) and the SDM Connector. You do not have to use 3DES.
- ** Credentials of old user models using SHA are updated to SHA-256 on first-time login to 9.4. For the newly created models (from 9.4), credentials are hashed using SHA-256.
- *** OpenSSL module is part of CAPKI. In FIPS mode, it is FIPS-2 complaint as all consumer products use only FIPS approved algorithm from Crypto-C ME of CAPKI. These certificate and algorithm certs are from Crypto-C ME (4.0.1).
- You can find NIST certificate numbers at: http://csrc.nist.gov/groups/STM/cmvp/validation.html
- These are the only algorithms the software supports. You can find more information at: http://csrc.nist.gov/groups/STM/cavp/validation.html
- Verify algorithm certificate numbers by looking up the certificate number at NIST, opening the Security Policy, or reading the 'Level/Description' column associated with the Certificate number.
- N/A means the software does not offer the ability to operate in FIPS mode. Compatible or Compliant means the software is capable of operating in FIPS mode according to the definitions of those terms.

Detailed Component Descriptions

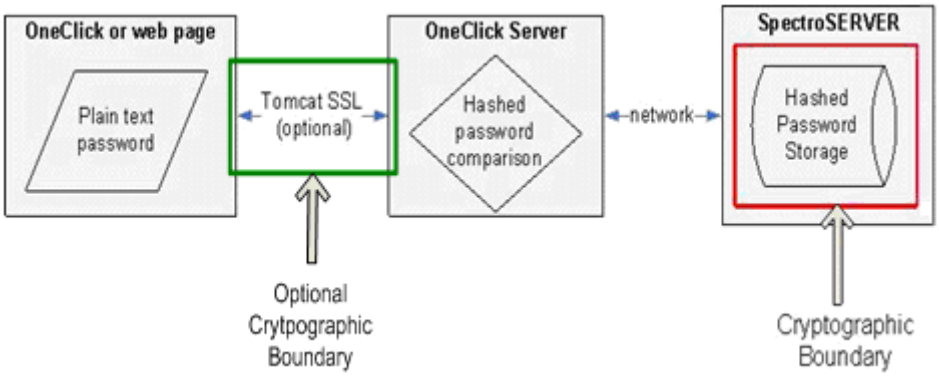
This chapter describes the encryption and hashing that is used by specific DX NetOps Spectrum components.

SpectroSERVER User Password Storage

From 9.4 onward, passwords are hashed using SHA-256, and are stored in the SpectroSERVER database for comparison when a user attempts to log in. Passwords of user models that were created before 9.4, are updated to SHA-256 after first-time login to 9.4 or above. All newly created user models in 9.4 and above versions support SHA-256 for hashing and authentication.

NOTE
Enable "Tomcat Secure Sockets Layer (SSL)" for protection over the wire.

The following image illustrates the Cryptographic Boundary for SpectroSERVER user password storage:



The DX NetOps Spectrum password is protected. For more information about configuring and using SSL, see the Create User Accounts and User Groups *section*.

Special Protection and Key Storage

The encryption key is internal to DX NetOps Spectrum.

Enable FIPS Modeguide

SHA-256 password hashing is enabled out of the box.

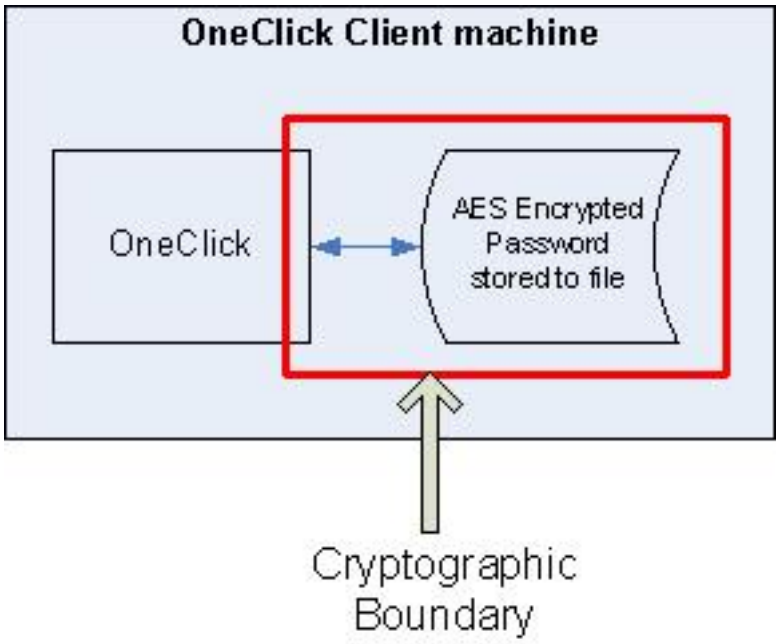
Changing Root Certificate

Not applicable at this time.

OneClick User Password Storage

When the "Remember my password" option is selected in the OneClick home page, OneClick passwords are stored to a file. It is part of Windows Security, and nothing to do with Spectrum. For Spectrum, the username and password are encrypted and stored in the SpectroSERVER database.

The Cryptographic Boundary for the OneClick password storage is as follows:



The OneClick username and password is encrypted with AES-256 in the file.

NOTE
For more information about the OneClick login, see the Using OneClick *section*.

Special Protection and Key Storage

The encryption key is internal to DX NetOps Spectrum.

Enable FIPS Mode

AES-256 password and username encryption is enabled out of the box.

Changing Root Certificate

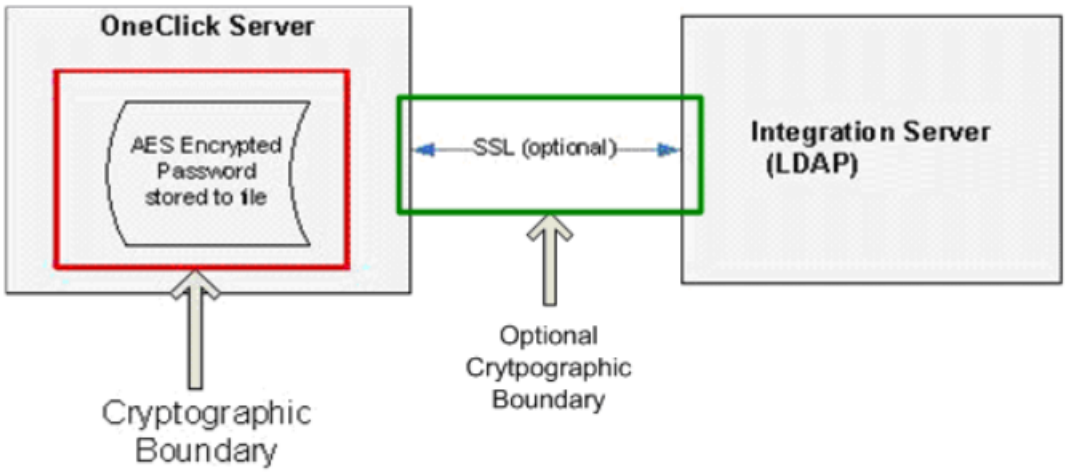
Not applicable at this time.

Integration Password Storage

Passwords for OneClick integrations are stored to a file and are encrypted using AES-256.

NOTE
For encryption over the network, enable SSL for integrations between DX NetOps Spectrum and CA eHealth, and DX NetOps Spectrum and Lightweight Directory Access Protocol (LDAP).

The Cryptographic Boundary for integration password storage is as follows:



The password is encrypted with AES-256 in the file.

> **NOTE**
> For more information about configuring and using LDAP, see the Create User Accounts and User Groups section.

Special Protection and Key Storage

The encryption key is internal to DX NetOps Spectrum.

Enable FIPS Mode

AES-256 password encryption is enabled out of the box.

Changing Root Certificate

Not applicable at this time.

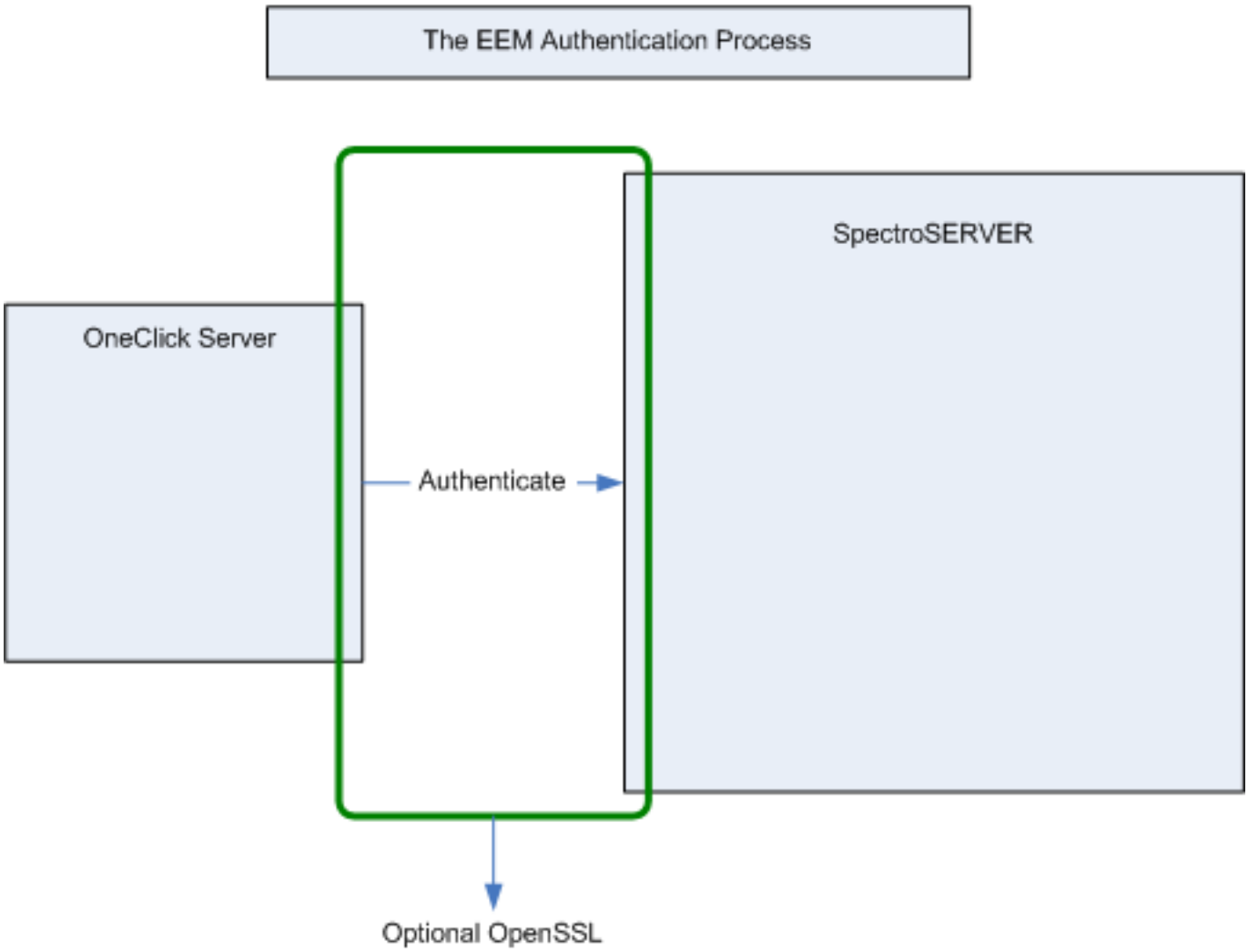
Embedded Entitlements Manager (EEM) Single Sign-On (Proxy Password)

Single Sign-On functions are used mainly for logging in, but can also be used to access cross-platform resources, such as CA eHealth reports.

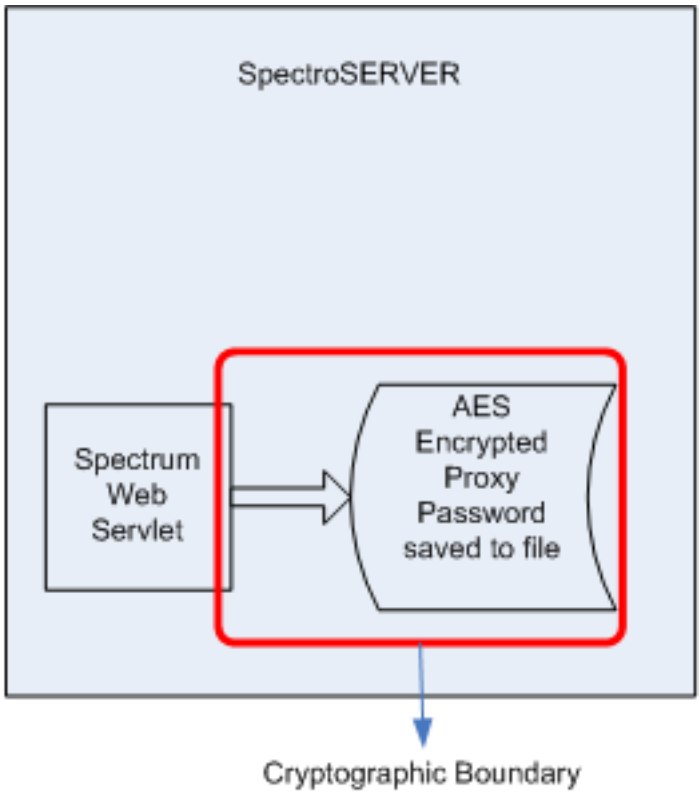
> **NOTE**
> If the token expires and a resource requires authentication, the authentication process repeats.

The following figures illustrate the Cryptographic Boundary for Embedded Entitlements Manager (EEM) single sign-on (proxy password).

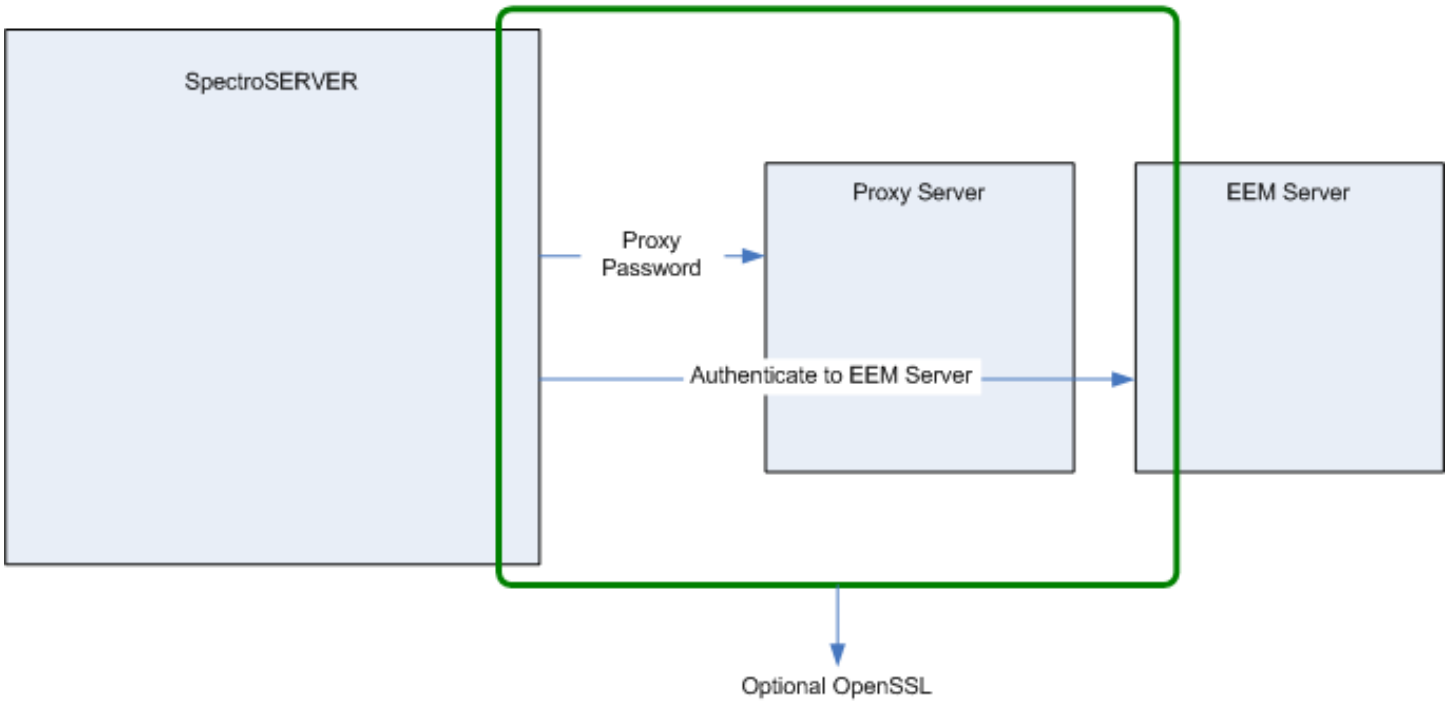
The following figure illustrates how the OneClick web server communicates with the SpectroSERVER to authenticate which OneClick web server transmissions can be encrypted with OpenSSL:



The following figure illustrates how the SpectroSERVER uses a configuration file to determine the authenticator. In this instance, the authenticator is configured to be Single Sign On through an EEM server, behind a proxy server. The password for the proxy server is AES-256 encrypted and is stored on the SpectroSERVER in a file:



The following figure illustrates how the proxy password is transmitted to the proxy server, which then allows the SpectroSERVER to communicate to the EEM server for authentication. These transmissions can also be encrypted with OpenSSL:



The proxy password is encrypted with AES-256 in the file. The transmission of the data between servers can protected by SSL, but is not required.

> **NOTE**
> For more information about the EEM login and configuration information, see How to Configure DX NetOps Spectrum/CA EEM Integration section. For more information about configuring the SSL, see the Configure OneClick for Secure Sockets Layer section.

Special Protection and Key Storage

There is no special protection for the file; however, the key is internal to DX NetOps Spectrum.

Enable FIPS Mode

AES-256 password encryption is enabled out of the box. It cannot be turned off. SSL is optional, and can be turned off.

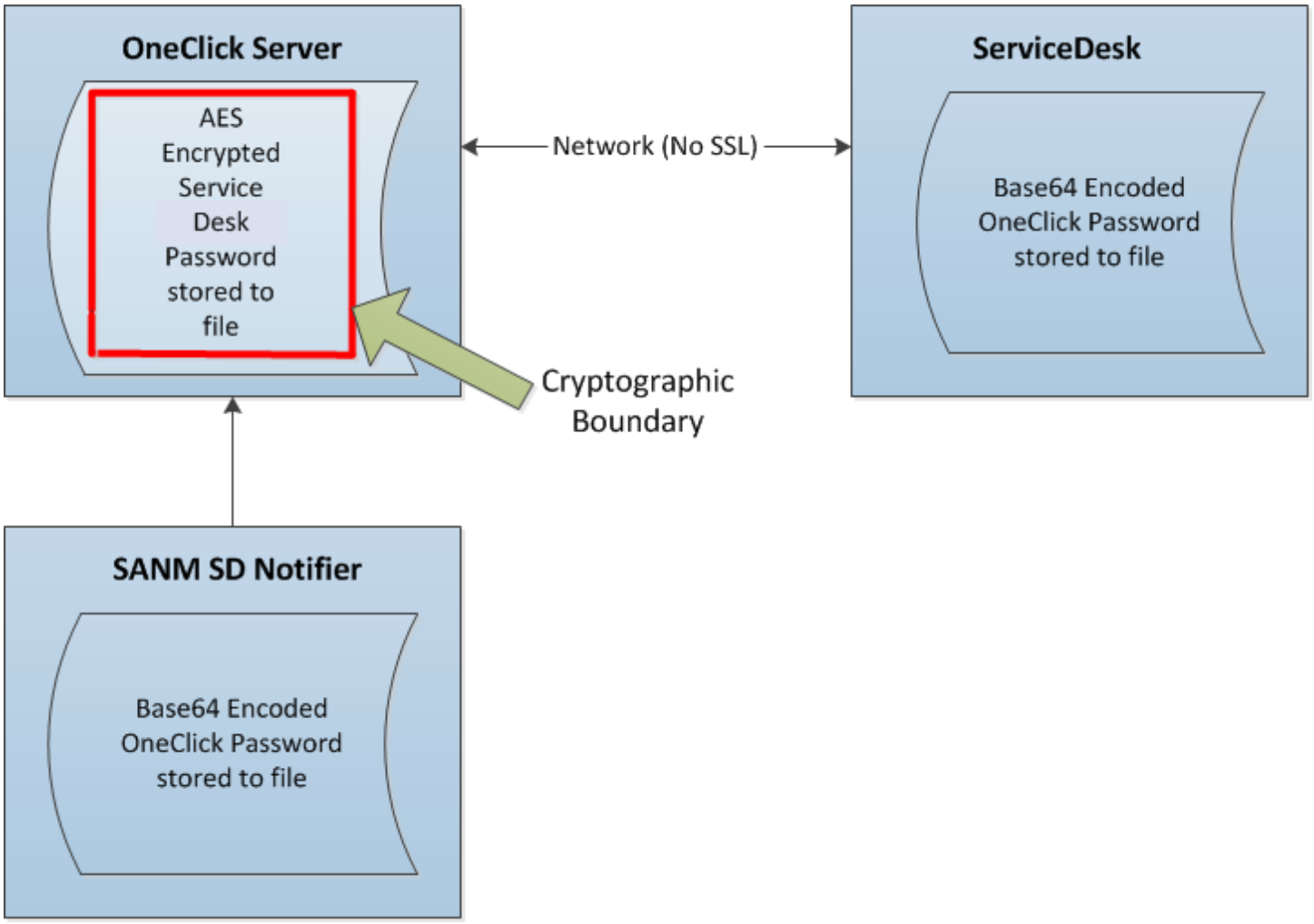
Changing Root Certificate

Not applicable at this time.

CA Service Desk Password Storage

The CA Service Desk password is stored to a file with AES-256 encryption. The OneClick password is stored to a file with Base64 encoding.

The Cryptographic Boundary for CA Service Desk password storage is as follows:



NOTE
For more information about integrating with CA Service Desk, see the DX NetOps Spectrum and CA Service Desk Integration section.

Special Protection and Key Storage

The encryption key is internal to DX NetOps Spectrum.

Enable FIPS Mode

AES-256 password encryption and Base64 encoding is enabled out of the box.

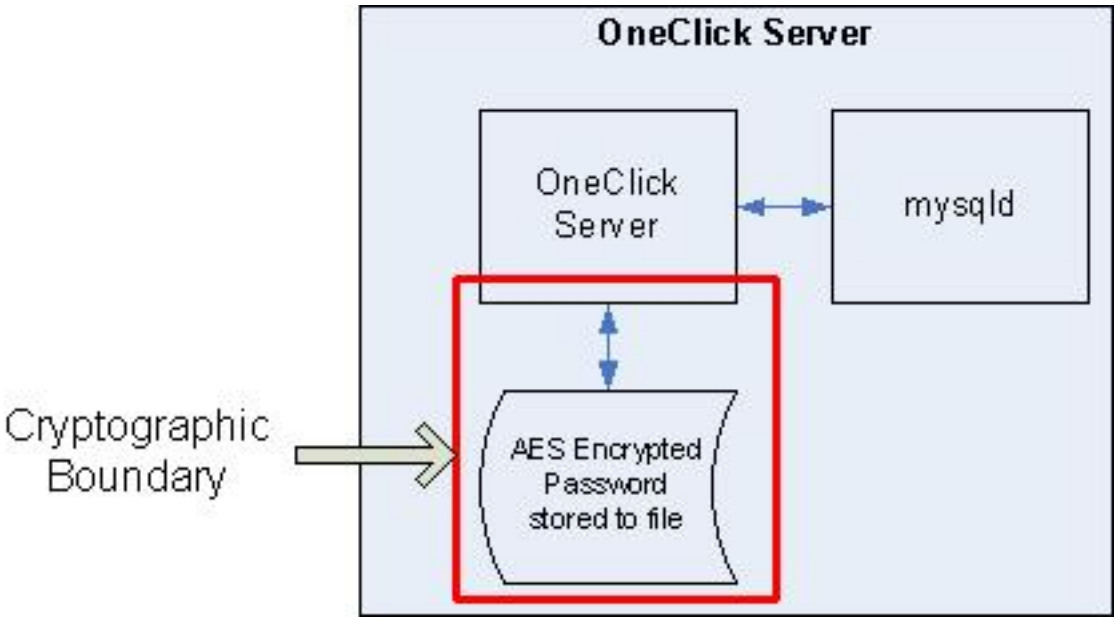
Changing Root Certificate

Not applicable at this time.

MySQL Password Storage

The mysql password is stored to a file with AES-256 encryption.

The Cryptographic Boundary for a MySQL password storage is as follows:



NOTE
For more information about configuring and using MySQL, see the Administration *section*.

Special Protection and Key Storage

The encryption key is internal to DX NetOps Spectrum.

Enable FIPS Mode

AES-256 password encryption is enabled out of the box.

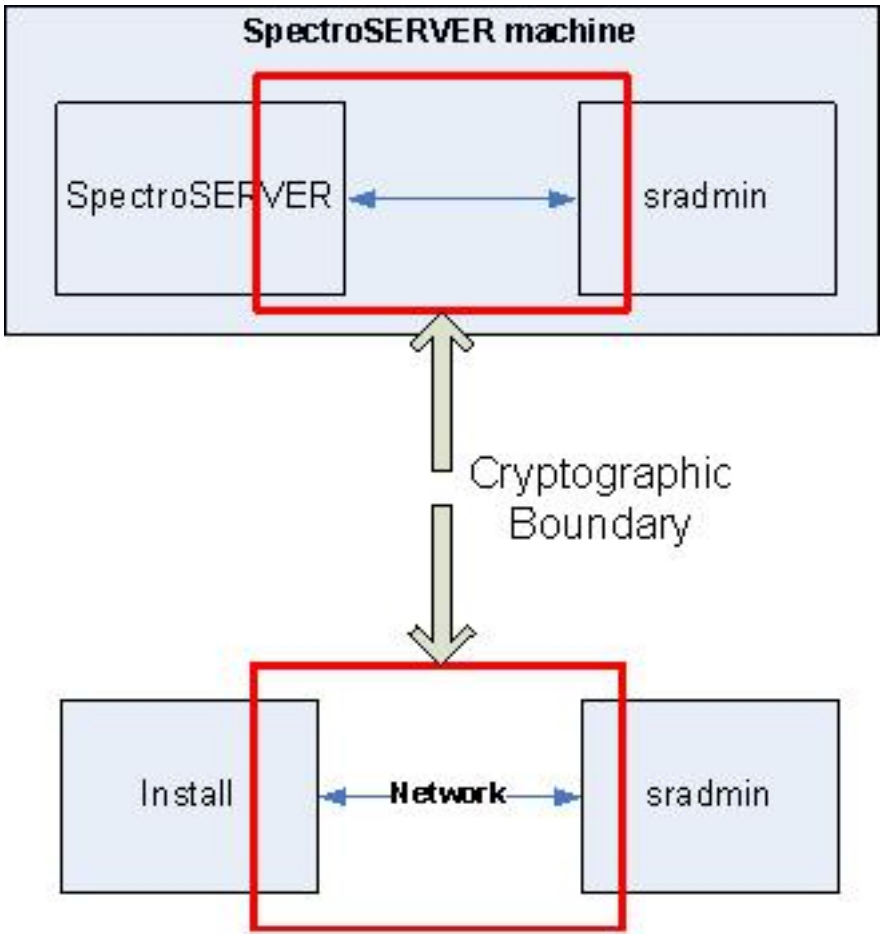
Changing Root Certificate

Not applicable at this time.

SRAdmin Data Transmission

The username and password for DX NetOps Spectrum Remote Administration (SRAdmin) is encrypted using 3DES.

The Cryptographic Boundary for SRAdmin data transmission is as follows:



The sradmin username and password are encrypted with 3DES and are sent over the network.

> **NOTE**
> For more information about configuring and using SRAdmin, see the Installation section.

Special Protection and Key Storage

The encryption key is internal to DX NetOps Spectrum and is also based off the time of the system.

Enable FIPS Mode

3DES username and password encryption is enabled out of the box.

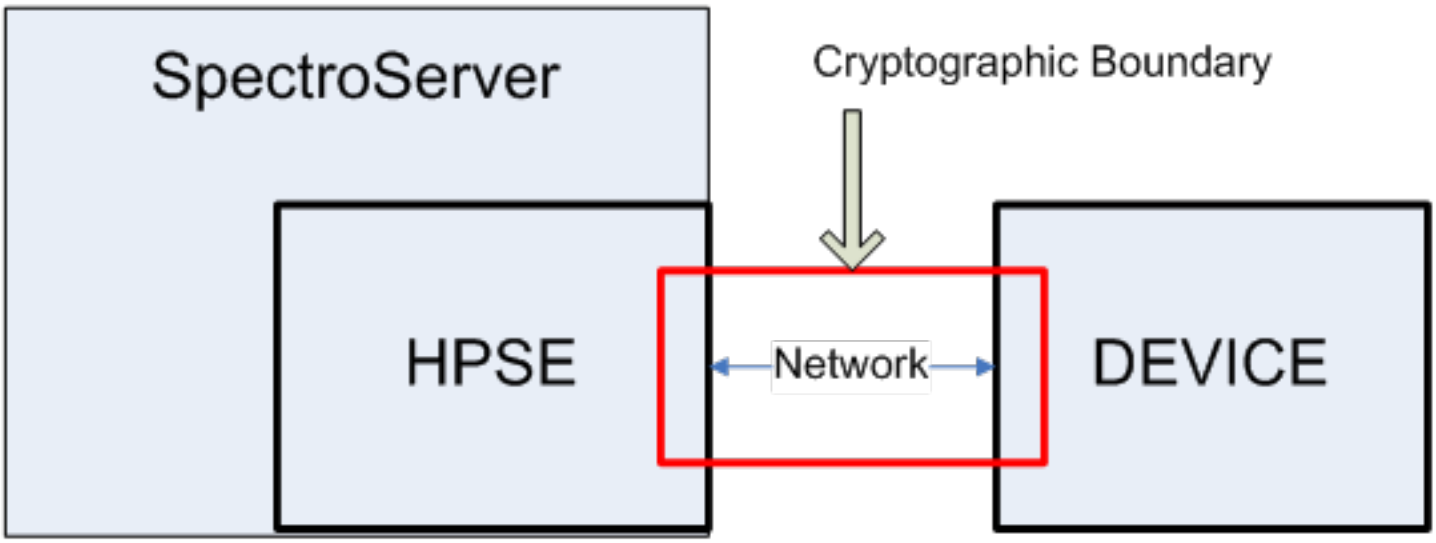
Changing Root Certificate

Not applicable at this time.

SNMPv3 Privacy Data Transmission

To query devices using SNMPv3 with the authentication and privacy, DX NetOps Spectrum can be configured so that the SNMPv3 messages are encrypted using 3DES, AES-128, or AES-256. DX NetOps Spectrum can also be configured to use SHA for authentication when querying devices using SNMPv3 with only authentication or authentication with privacy.

The Cryptographic Boundary for SNMPv3 privacy data transmission is as follows:



The SNMPv3 messages are encrypted using 3DES, AES-128 (AES), or AES-256 and are sent over the network.

NOTE
For more information about configuring and using SNMPv3, see the Modeling and Managing Your IT Infrastructure Administrator *section.*

Special Protection and Key Storage

The encryption key is internal to DX NetOps Spectrum.

Enable FIPS Mode

SNMPv3 does not support FIPS mode, but SNMPv3 supports FIPS-compliant algorithms.

To change the default privacy encryption algorithm for all device models to 3DES, AES or AES-256, the snmpv3_default_priv_protocol parameter in the <*$SPECROOT*>\SS\.vnmrc file must be set to 3DES, AES, or AES256.

For example:

```
snmpv3_default_priv_protocol=3des
```

or

```
snmpv3_default_priv_protocol=aes
```

or

```
snmpv3_default_priv_protocol=aes256
```

Alternatively, to override the default privacy encryption algorithm for a particular device model, append the encryption algorithm to the SNMP community string for that device model.

For example:

```
#v3/<authPW>:3DES^<privPW>/<user>
#v3/<authPW>:AES^<privPW>/<user>
#v3/<authPW>:AES256^<privPW>/<user>
```

To change the default authentication algorithm for all device models to SHA, the snmpv3_default_auth_protocol parameter in the <*$SPECROOT*>\SS\.vnmrc file must be set to "sha".

For example:

```
snmpv3_default_auth_protocol=sha
```

Alternatively, to override the default authentication algorithm for a particular device model that uses authentication only, append the authentication algorithm to the SNMP community string in the following format:

For example:

```
#v3/SHA^<authPW>/<user>
```

To override the default authentication algorithm for a particular device model that uses authentication with privacy, append the authentication algorithm to the SNMP community string in the following format:

```
#v3/SHA^<authPW>:<privPW>/<user>
#v3/<authPW>:AES^<privPW>/<user>
#v3/<authPW>:AES256^<privPW>/<user>
```

NOTE
For more information, see the Modeling and Managing Your IT Infrastructure Administrator *section.*

Changing Root Certificate

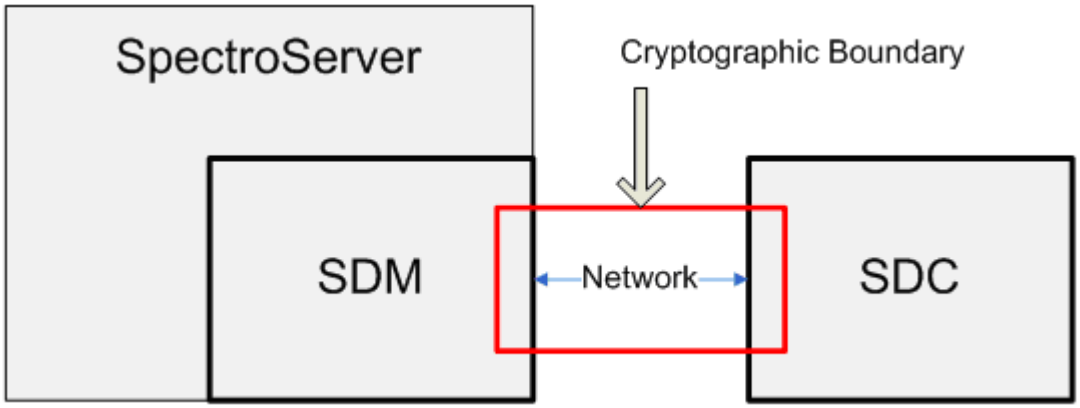
Not applicable at this time.

Secure Domain Manager

A bi-directional communication takes place between the Secure Domain Manager (SDM) and the Secure Domain Connector (SDC).

NOTE
When not running in FIPS mode, DX NetOps Spectrum, using SDM and SDC, runs in a FIPS-compatible state.

The Cryptographic Boundary for the Secure Domain Manager is as follows:



SNMP or ICMP requests and replies, and SNMP traps communication are protected. No other communication takes place.

NOTE
For more information about configuring and using the Secure Domain Manager, see the Secure Domain Manager (SDM) section.

Special Protection and Key Storage

The SDM private key is located at <*$SPECROOT*>/SDM/CERTS/SDMCAKey.pem. The private key requires administrator read and write privileges only.

Enable FIPS Mode

Locate the sdm.config file and add the - withfips option.

NOTE
Any time this option is changed, the entire system needs to be restarted.

Changing Root Certificate

Run the following command to create a new network security certificate for the SDManager:

```
CertGen.exe  - t cert  - c <Country Code>
```

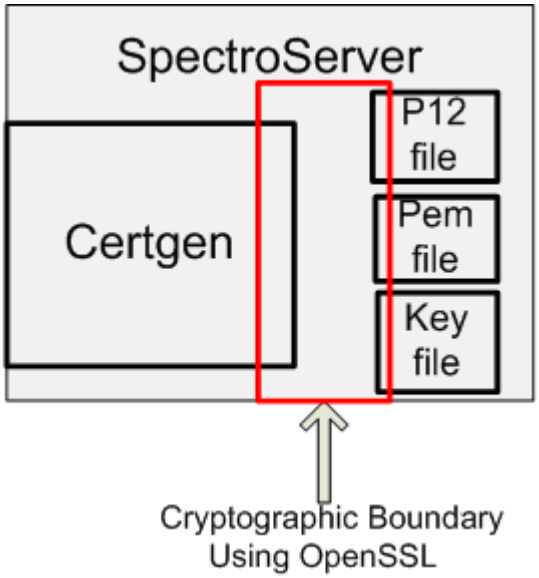
For added security, use the -p option to generate the certificate with a password as follows:

```
CertGen.exe  - t cert  - p <password> -c <Country Code>
```

Certgen

Certgen uses openSSL 0.9.8 to create certificate authorities, key files, and security certificates that are used to encrypt communications between SDM and SDC. The algorithm that is used to create the certificate is 3DES.

The Cryptographic Boundary for Certgen is as follows:



The p12 certificate is used primarily to protect the data being transferred between the Secure Domain Manager and the Secure Domain Connectors.

NOTE
For more information about configuring and using the CertGen, see the Secure Domain Manager section.

Special Protection and Key Storage

All certificate authority, private keys, and certificates are located at <*$SPECROOT*>/SDM/CERTS. All files require administrator read and write privileges.

Enable FIPS Mode

No FIPS mode required.

Changing Root Certificate

Run the following command to create a new network security certificate for the SDManager:

```
CertGen.exe  - t cert  - c <Country Code>
```

For added security, use the -p option to generate the certificate with a password as follows:

```
CertGen.exe  - t cert  - p <password> -c <Country Code>
```

Additional Resources

Provides additional resources to assist you in maximizing your product experience.

This section provides additional resources to assist you in maximizing your product experience; for example:

- Product Videos List
- Frequently Asked Questions
- Glossary
- Collect Troubleshooting Data
- CA Green Books

Product Videos

Access the following YouTube link to view the DX NetOps Spectrum videos:

- DX NetOps Spectrum YouTube Playlist

Frequently Asked Questions

Following is the list of frequently asked questions and their responses respectively:

Features

Q: What are the key updates to the current release of DX NetOps Spectrum?

A: For the complete list of new features and enhancements, review the Feature and Enhancements page.

Install and Upgrade

Q: Are there any changes to the operating systems that are supported in the current release of DX NetOps Spectrum?

A. Review the System Requirements for Installing DX NetOps Spectrum for more information.

Q: Is the current release a full release?

A. Yes, it is a full release.

Q. Is a base version required to install the current release of DX NetOps Spectrum?

A. No base version is required to install this release.

Q. Have the CA integration components changed?

A. Yes, review the Integration Compatibility Matrix page for the specific version support.

Q. What is the supported Java version in the current release of DX NetOps Spectrum?

A. Adopt OpenJDK 1.8u242

Third-Party Software Updates

Q. What are the third-party software agreements in the current release of DX NetOps Spectrum?

A. To view the list of the third-party software agreements applicable for this release, see the third-party license agreements page.

JasperSoft/CABI

Q. What version of CABI is supported in the current release of DX NetOps Spectrum?

A. See the Integration Compatibility Matrix page for the specific version support.

Glossary

Active Directory Domain Services (AD DS) server role

The *Active Directory Domain Services (AD DS) server role* stores directory data for all objects in your network. The AD DS server role also manages communication between users and domains, including authentication requests and directory searches.

action

An *action* is any operation that is not part of the basic set of operations that DX NetOps Spectrum defines for use with a model.

active node

An *active node* is a system in a cluster environment where application processes (as part of a resource group) are currently running. Within DX NetOps Spectrum Cluster Manager, an active node has resource groups as children. A solid workstation icon represents an active node in the Cluster Manager hierarchy.

ADES AIM Discovery

ADES AIM Discovery is the process that the ADES AIM performs to identify hosts in Active Directory and Exchange Server environments. ADES AIM Discovery uses the Global Catalogs to extract information that is based on user configurations. These configurations specify domain name, management entity (Active Directory, Exchange Server, or both) and management mode (domain-based or host-based). Hosts that ADES AIM Discovery identifies are made available to DX NetOps Spectrum for management.

ADES Host Manager

Active Directory and Exchange Server Host Manager (ADES Host Manager) is the DX NetOps Spectrum model that represents a host that contains the ADES AIM. Successful creation of this model indicates that all requisite intelligence to support ADES Manager has been installed on the host.

ADES Manager Discovery

ADES Manager Discovery is the modeling within DX NetOps Spectrum of Active Directory and Exchange Server hosts that ADES Manager is going to manage. ADES Manager Discovery also includes the modeling of any other required components for ADES Manager. ADES Manager Discovery obtains information about the hosts that are determined as available for management by ADES AIM Discovery.

alarm

An *alarm* is an indication that an abnormal condition exists for a model.

alarm severity

An *alarm severity* is a value found in an EventDisp file that indicates the condition of a model. Conditions represent the presence and seriousness of an alarm. The valid values are 0 through 6 and all represent a different colored condition.

alert

An *alert* is an unsolicited message that is sent from a managed element to the SpectroSERVER.

alert code

An *alert code* is a string that identifies an alert. An alert that is received from an SNMP source contains an alert code that consists of a generic trap followed by a dot (.) followed by an enterprise-specific trap.

AlertMap file

An *AlertMap file* exists for each model type. This file maps incoming SNMP trap data to DX NetOps Spectrum events.

application insight module (AIM)

The SystemEDGE agent provides a plug-in architecture through which it can load optional *application insight modules (AIMs)* when it initializes. AIMs are functional extensions to the SystemEDGE agent.

application programming interface (API)

An *application programming interface (API)* is a set of routines that are used to make calls to another software package.

association

An *association* is a link that is formed between two models by a relation.

asynchronous call

An *asynchronous call* is a call to a method that begins (but does not necessarily complete) a requested operation before allowing a program to continue. At some point, the requested operation completes and notifies the program. In the meantime, the program and the requested operation can both proceed simultaneously.

attribute

An *attribute* is the declarative knowledge in DX NetOps Spectrum that defines a model type. Attributes are defined using the Model Type Editor.

Attribute Editor

The *Attribute Editor* is a OneClick utility that lets you change attributes configured at the device level.

available host

An *available host* is an Active Directory or Exchange Server host that ADES AIM Discovery has identified and which qualifies for management by DX NetOps Spectrum ADES Manager.

DX NetOps Spectrum event

A *DX NetOps Spectrum event* is an event that is generated based on alerts received from MOM- or SCOM-managed hosts.

channel

A channel is a data transmission link between two or more points.

client

A *client* is the application process in a client-server architecture.

cluster

A *cluster* is a group of locally attached machines that provide distributed processing power and high availability. A cluster appears to clients as a single system image and IP address.

client-server architecture

Client-server architecture is a system design that is based on the relationship between a service-providing process (the *server*) and a service-using process (the *client*).

Cluster Manager discovery

Cluster Manager discovery is the modeling within DX NetOps Spectrum of your cluster components. After the cluster technology AIM is modeled successfully, Cluster Manager obtains information about the cluster components in your environment from the AIM. Using a list of machines that is obtained from the AIM, Cluster Manager uses AutoDiscovery to model each cluster node. All supporting cluster components (clusters, resource groups, and resources) are also modeled.

cluster node

A *cluster node* is an independent computer system that participates in a cluster. A cluster node can be active or inactive. The active node has application processes (as part of a resource group) currently running. An inactive node is a system that is allocated to a cluster but not currently processing any resources.

configuration

A *configuration* contains the parameters you specify to determine which network entities in your infrastructure you want *Discovery* to locate and identify for review, export, or modeling.

connection

A *connection* is a link between two modeled elements in a view.

container

A *container* is a graphical icon that you can use to depict a group of modeled devices by network technology such as LAN, Network, ATM, or to represent some other containment concept such as a Department.

CORBA (Common Object Request Broker Architecture)

CORBA is a software component architecture that lets various and different software components communicate. The software components can be written in different languages (C, C++, Java, Smalltalk, COBOL, ADA, Lisp, Perl, Tcl, Eiffel, Python). The software components can also reside on different machines and can be written for different operating systems.

custom installation script

A *custom installation script* is a script that can be included in a VCD. The script executes as the integration package is being installed on the SpectroSERVER.

database

A *database* is a collection of interrelated data that is organized to facilitate efficient and accurate inquiries and updates.

database availability group (DAG)

A *database availability group (DAG)* is a cluster of Exchange Mailbox servers that are used for continuous replication, providing failover at the database level. The databases on any of the DAG members can be replicated to the other DAG members. At any given time, the database is active on one DAG member only, while the databases on the other DAG

members are passive. A passive database can then be activated in the event of failure. The DAG feature was introduced in Exchange 2010.

database management system

A *database management system* is a software package that organizes and maintains a database.

developer ID

A *developer ID* consists of a 14-character developer name and a four-character prefix. The developer name must be alpha-numeric and can include underscores, however, no other punctuation marks are allowed. The prefix must also be alpha-numeric, however, no underscores or other punctuation marks are allowed. DX NetOps Spectrum uses *developer IDs* to verify that the objects, such as model types, attributes, or relations, created by users or integrators have unique identifiers. These objects can therefore be distributed to other users without conflict.

To obtain a developer ID, contact CA Support.

Development API

The *Development API* is the CORBA-based application program interface (API) to the SpectroSERVER. The Development API is also referred to as the SpectroSERVER Object Request Broker (SSORB) interface because it depends on an object request broker (ORB).

device

A *device* is a managed element.

device attributes

Device attributes are the configuration settings written to a device or interface.

Device Communications Manager (DCM)

The *Device Communications Manager (DCM)* is a multiprotocol communication engine in the SpectroSERVER that handles the communication with all managed elements, regardless of their protocol. The DCM translates the SpectroSERVER requests into protocols that the individual devices understand.

Discovery

Discovery is a OneClick feature that automates the process of discovering and modeling the entities in your IT infrastructure. You can create and edit Discovery and modeling configurations to customize and simplify the process. DIscovery also lets you filter and export the results of Discovery or modeling sessions.

Distributed SpectroSERVER (DSS)

The *Distributed SpectroSERVER (DSS)* is a modeling feature that uses the concept of landscapes to improve DX NetOps Spectrum performance when managing a large computing infrastructure. DSS technology distributes the load that management traffic introduces and lets you delegate management functions to remote workstations.

distributed SpectroSERVER (DSS) environment

A *distributed SpectroSERVER (DSS) environment* consists of more than one SpectroSERVER. This environment enables management of a large-scale infrastructure. The SpectroSERVERs in this environment can be located within a single physical location or in multiple physical locations.

domain

A *domain* is an Active Directory container structure that contains a collection of objects that share a common set of policies, name, and security database. A domain is at the lowest level of the logical structure of an entire network. The domain name identifies the domain.

domain controller

A *domain controller* is a host that is running AD DS. Typically multiple domain controllers host Active Directory within a domain, and you can manage your network resources from any domain controller within your domain.

domain tree

A *domain tree* is an Active Directory container structure that contains a collection of one or more domains in a network.

domain-based management

Domain-based management is a configuration option in the ADES AIM where all newly discovered, available hosts in the domain are automatically managed by default. Domain-based management is typically used with domains that are small enough for a single ADES AIM to manage.

Edit Mode

Edit Mode lets you edit the current view in DX NetOps Spectrum. Selecting Edit Mode displays the File and Edit options in the menu bar.

element management system

An *element management system* lets you provision, manage, and monitor elements in a network infrastructure.

event

An *event* is a significant message from the SpectroSERVER.

event code

An *event code* is a hexadecimal number that uniquely identifies an event.

event data template

An *event data template* is a series of integers that Southbound Gateway uses to format variable data coming from an alert.

event discriminators

Event discriminators are references to event variables that let you to generate alarms for events based on the values of the variables.

event format file

An *event format file* contains the message about the event that appears in the Events tab in the OneClick Console when the event occurs.

event message

The *event message* is the message that is displayed on the Events tab in OneClick when the event occurs.

event severity

Event severity is a numeric value that is found in an EventDisp file entry that indicates the seriousness of an event. Valid values are from 0 through 100, with 0 being the least severe.

event variable ID

An *event variable ID* is an ID that represents the event variable data. Use of this ID returns the event variable value.

EventAdmin

EventAdmin is the Southbound Gateway model type that represents a third-party management system.

EventDisp file

An *EventDisp file* exists for each model type and determines how DX NetOps Spectrum processes events for that model type.

EventModel

An *EventModel* is a model type in DX NetOps Spectrum that represents a unique alert source within a Southbound Gateway integration.

failover/failback (MSCS)

Failover is a transfer process where resource groups that are hosted on a particular node that fails move to another node in the cluster. The reverse process is "failback". Failback occurs when the failed node becomes active again, and the groups that were failed over to other nodes transfer back to the original node.

fall over/fall back (IBM PowerHA)

Fall over is a transfer process where resource groups that are hosted on a particular node that fails move to another node in the cluster. The reverse process is "fall back". Fall back occurs when the failed node becomes active again, and the groups that moved to other nodes transfer back to the original node.

forest

A *forest* is an Active Directory container structure that contains a collection of Active Directory objects, their attributes, and attribute syntax. A forest is at the highest level of the logical structure. A forest is a collection of domain trees sharing a common global catalog, directory configuration, directory, schema, and logical structure.

head-end device

The LSP *head-end device* is a Provider Edge router device used to create the LspHead model in MPLS Transport Manager. This model groups your LSPs by ingress device in the OneClick Navigation panel, making it easier to view and locate data about your LSPs within the MPLS environment.

host-based management

Host-based management is a configuration option in the ADES AIM where all newly discovered, available hosts in the domain are automatically not managed by default. Host-based management is typically used with domains that are large enough that multiple ADES AIMs manage them.

Hub Transport server role

The Exchange Server *Hub Transport server role* handles email flow and routing. All messages are delivered through this role, regardless of whether they are being delivered locally or remotely.

IBM PowerHA Cluster Manager

The *IBM PowerHA Cluster Manager* is the DX NetOps Spectrum model that represents a host that contains the HACMP AIM. The HACMP AIM monitors the IBM PowerHA cluster elements (clusters, nodes, resource groups, and resources) in your environment.

ICMP (Internet Control Message Protocol)

ICMP supports packets that contain error, control, and informational messages.

icon

An *icon* is a visual representation of something that is displayed on a screen.

inactive node

An *inactive node* is an available cluster node that has no resource groups currently running on it. In DX NetOps Spectrum, unlike a model in maintenance mode or hibernation mode, the inactive node model is fully functional. Data is gathered for the node, and any alarm activity or events for the node post to the model. Within the Cluster Manager hierarchy, an inactive node does not have any resource groups as children. A transparent icon represents an inactive node.

index file

An *index file* is a file that is created using the DX NetOps Spectrum Extension Integration (SEI) Toolkit and defines the components of the integration. The index file indicates where the integration components exist on the host and their location on the customer host.

Inductive Modeling Technology (IMT)

Inductive Modeling Technology (IMT) is a CA set of artificial intelligence techniques. These techniques allow a computing infrastructure of arbitrary complexity to be modeled such that every element is given intelligence.

inference handler

Inference handlers are the intelligence behind DX NetOps Spectrum. Inference handlers monitor the changes in the environment, the changes between related models, and the changes in relationships between models and attributes.

instance variable

The *instance variable* stores the instance portion of the OID. If the variable binding identifies a particular object from a table variable within the trap MIB, it likely includes an instance ID.

instance variable ID

An *instance variable ID* is the integer that identifies the instance variable in the OID map. Use of this ID returns the instance variable.

instantiate

In object-oriented design, to *instantiate* is to create a particular occurrence of something.

intelligence circuit

An *intelligence circuit* is a collection of inference handlers that defines the behavior of a model type.

knowledge base

The *knowledge base* is the collection of everything that DX NetOps Spectrum can model and managed including concepts, relationships, declarative knowledge, and procedural knowledge.

knowledge base management system

The *knowledge base management system* is the software that is used to define and manage the information in the knowledge base.

label

A *label* is a fixed-size field contained in a packet header that can be used as an exact-match key in determining how to forward a protocol data unit.

Label Switch Router (LSR)

A *Label Switch Router (LSR)* is a router in an MPLS network that switches the routing label on a data packet before forwarding it to the next hop in the LSP. The LSR uses a look-up table to determine the new label.

Label Switched Path (LSP)

A *Label Switched Path (LSP)* is the path within the MPLS network along which labeled packets are forwarded. Packets forwarded using a label are forwarded along the same path as other packets using the same label.

landscape

A *landscape* is all the data that is specific to any one virtual network machine (VNM) in a single network. The term also identifies the network domain that is managed by a single SpectroSERVER. In OneClick, a landscape is the network view of one SpectroSERVER.

legacy SNMP community string

The DX NetOps Spectrum *legacy SNMP community string* has been replaced in OneClick by access groups and privileges. The legacy SNMP community string can still be viewed and edited in OneClick under the Details tab for a selected user in the Users tab. The legacy SNMP community string is still used by non-OneClick DX NetOps Spectrum applications.

license

A *license* determines which privileges can be granted to holders of that license. Launching a OneClick client consumes any licenses granted to that user.

Mailbox server role

The Exchange Server *Mailbox server role* provides email storage (including user mailboxes), advanced scheduling services, and supports public folders. Continuous replication technology provides a reliable failover mechanism in the event of failure. In Exchange 2007, continuous replication failover is at the server level. With Exchange 2010 and the introduction of database availability groups (DAGs), failover is at the database level.

managed element

A *managed element* is an item or device whose status is being monitored and controlled. A managed element can be a network device, host system, application, service, or other computing infrastructure component.

managed host

A *managed host* is an Active Directory or Exchange Server host that DX NetOps Spectrum ADES Manager manages, as reflected by the Managed Host Table. The ADES AIM polls all managed hosts for Active Directory and Exchange Server metrics. Management of a host can be controlled by using the Universal Host Table view in DX NetOps Spectrum.

Managed Host Table (MHT)

The *Managed Host Table (MHT)* contains all hosts that the ADES AIM is managing (polling for Active Directory and Exchange Server metrics). To clarify, the Universal Host Table contains all hosts that an AIM *can* manage. The Managed Host Table contains those hosts that *are* managed. The MHT table resides on the ADES AIM and is viewable in DX NetOps Spectrum. ADES Manager uses the MHT as its basis for creating, deleting, or updating Active Directory and Exchange Server host models.

managed object

A *managed object* is a variable on a managed element containing one piece of information about the node. Each node can have several objects.

management agent

A *management agent* is an implementation of a management protocol that exchanges information for the managed element with the management station.

management entity

Management entity is a configuration option in the ADES AIM on a per-domain basis that controls what technologies to monitor by the ADES AIM: Active Directory, Exchange Server, or both.

management mode

Management mode is a configuration option in the ADES AIM that controls monitoring of the Active Directory and Exchanger Server environments on a per-domain basis. Domain-based monitoring automatically manages all newly discovered Active Directory and Exchange Server hosts in the domain. Host-based monitoring automatically does not manage all newly discovered Active Directory and Exchange Server hosts in the domain.

Microsoft Cluster Manager

The *Microsoft Cluster Manager* is the DX NetOps Spectrum model that represents a host that contains the MSCS AIM. The MSCS AIM monitors the Microsoft Clusters Service elements (clusters, nodes, resource groups, and resources) in your environment.

Management Station Access Provider (MSAP)

Management Station Access Provider (MSAP) is a software task that provides an object with access to a management station.

manual modeling

Manual modeling is the act of manually representing individual devices and their connections within a OneClick topology view.

meta-rules

Meta-rules specify which model types can participate in a relation.

migration

Migration is the movement of a resource group from one node to another. Different terms are used to describe migration depending on the cluster technology; for example, failover, fall over, failback, and fallback.

model

A *model* in DX NetOps Spectrum represents a modeled network element.

MIB (Management Information Base)

A *MIB (Management Information Base)* is a database that resides on a network device and represents that device as a hierarchical collection of objects. A MIB object represents an individual element of information, such as the uptime of a device. MIBs themselves are text files with special syntax.

mkcd

The *mkcd* is a tool that is a part of the DX NetOps Spectrum Extension Integration (SEI) Toolkit. This tool finishes the VCD by adding a version number and making the VCD installable on a DX NetOps Spectrum host.

mkmm

The *mkmm* is a tool that is a part of the DX NetOps Spectrum Extension Integration (SEI) Toolkit. This tool creates the VCD using the index file and the applicable component files.

model

A *model* is a collection of information that forms a specific occurrence of some basic defined type. In DX NetOps Spectrum, models are instances of model types.

model type

A *model type* is a template that describes the attributes, actions, and associations that are related to a managed element in DX NetOps Spectrum.

Model Type Editor (MTE)

The *Model Type Editor (MTE)* is the primary tool that defines the concepts, relationships, meta-rules, and declarative knowledge in the DX NetOps Spectrum knowledge base.

Model by Host Name

Model by Host Name is a modeling feature that you can use in the Universe, World, or TopOrg topology. This feature lets you manually model a new device by specifying the host name for the device.

Model by IP Address

Model by IP Address is a modeling feature that you can use in the Universe, World, or TopOrg topology. This feature lets you manually model a new device by specifying the device IP address.

Model by Model Type

Model by Model Type is a modeling feature you can use in the Universe, World, or TopOrg topology. This feature lets you manually model container icons or devices by a model type.

modeling methods

There are two *modeling methods* available for modeling your network infrastructure. You can automate the process using *Discovery,* or you can manually model the individual entities and later enhance the model presentation using the Topology Edit Mode tools.

modeling session

A *modeling session* occurs when you instruct *Discovery* to model the results of a *Discovery session*. It uses the modeling options specified to model the network entities discovered in that *configuration.*

Modular Object Oriented Threads (MOOT)

Modular Object Oriented Threads (MOOT) is a task control manager that is used in the SpectroSERVER.

Multiprotocol Label Switching (MPLS)

Multiprotocol Label Switching (MPLS) refers to a group of technologies that marries IP to Layer 2 technologies (such as ATM) by overlaying a protocol on top of IP networks.

Navigation Mode

Navigation Mode is the moving from one view to the next in DX NetOps Spectrum. The menu bar contains the File and View options when in Navigation Mode. Navigation Mode does not allow editing of a view.

Network management agent

A *network management agent* is an implementation of a management protocol that exchanges information about the managed element with the management station.

network management protocol

A *network management protocol* is the means by which the management station and the managed elements exchange information.

network management station

A *network management station* is the host system or workstation that is running the network management applications and protocol.

object-oriented design (OOD)

Object-oriented design (OOD) is design that encompasses the process of breaking a system into parts. Each part represents some class or object from the problem domain. OOD is applied by viewing the world as a collection of objects that cooperate with one another to achieve some desired functionality. OOD typically includes a notation for depicting logical or physical and static or dynamic models of the system under design.

OneClick Console client

The *OneClick Console client* is a Java JNLP application which provides network operators with a view into the details and health of the network.

OneClick web server

The *OneClick web server* is the server responsible for moving data between SpectroSERVERs and OneClick clients.

OID

An *OID* is an identifier for a managed object.

OID map

An *OID map* is the syntax that is used to map an alert variable to an event variable.

path

A *path* is a large communications pipe that pre-allocates bandwidth and allows for greater flexibility in establishing PVCs. A defined amount of bandwidth is leased from a service provider, and as many PVCs as necessary within the limits of that bandwidth can be established.

In an MPLS environment, a *path* is the route taken by a data packet through a series of devices in the network. This route is defined hop by hop.

permanent virtual circuit (PVC)

A permanent virtual circuit (PVC) is a logical connection that is manually created by a network administrator. This connection is maintained at all times even if it is not always in use. PVCs can exist without being a part of a PVP.

permanent virtual path (PVP)

A permanent virtual path (PVP) is a logical communications path that has a defined amount of leased bandwidth. The path is maintained at all times even if it is not always in use.

pipe

A *pipe* is an icon that represents a connection between two devices or ports.

pollable attribute

A *pollable attribute* is an attribute for which the VNM regularly queries the managed element to obtain current values. Attributes are defined as pollable or non-pollable through the Model Type Editor.

probable cause file

The *probable cause file* is an ASCII text file that defines the symptoms, probable causes, and recommended corrective actions for an alarm. When an event generates an alarm, the text in the associated probable cause file is displayed on the Alarm Details tab in OneClick.

procedural knowledge

In DX NetOps Spectrum, *procedural knowledge* is information that defines how a concept behaves or reacts to environmental changes.

protocol

A *protocol* is a set of rules that computers use to communicate with each other.

proxy management

Proxy management is the act of managing network devices using an alternate management source in place of or in addition to the device itself. For example, DX NetOps Spectrum can manage Active Directory or Exchange Server hosts by contacting them directly or through the ADES AIM.

relation

A *relation* is information describing the connection that models have with each other.

resource

A *resource* is a logical component or entity (for example, a file system or an application) that runs on only one node at a time. A resource can move from one cluster node to another.

resource group

A *resource group* is a collection of resources that forms a functional unit existing on a single cluster node.

role (SPECTRUM)

A *role* is a reusable set of user privileges that you can assign to an access group. For example, the default role (OperatorRW) grants the set of read/write privileges typically needed by a OneClick operator.

SANM

The Alarm Notification Manager (*SANM*) is a DX NetOps Spectrum component that enhances the functionality of DX NetOps Spectrum alarm-processing applications.

security community

A *security community* determines user access to secure models. DX NetOps Spectrum users are selectively granted access to models that are secured with a *security string*.

security string

Security strings are expressions that define security communities. Security communities define access to models, securing them from unauthorized users. Security strings are set at the model level for modeled elements in OneClick.

sequential window

A *sequential window* is a type of type time window in which non-overlapping time windows are examined, one after another, to determine if the requisite number of events has occurred within the time window. This type of time window is best suited for detecting a long, sustained train of events.

Service Level Agreement (SLA)

A *Service Level Agreement (SLA)* is typically a contractual agreement between a business and its subscribers guaranteeing a specified level of service, such as 99.9 percent availability. An SLA sometimes contains a penalty clause for noncompliance.

server

A *server* is a process that provides services in response to a client request in a client-server architecture.

server role

A *server role* is the primary duty that a server performs. Active Directory and Exchange Server use server roles to assign specific functions to specific hosts.

sliding window

A *sliding window* is a type of time window where if the specified number of events (or more) ever occurs within any window of the specified time period, the output event is created in response. This type of time window is best suited for accurately detecting a short burst of events.

SNMP (Simple Network Management Protocol)

SNMP (Simple Network Management Protocol) is a network protocol that is used to monitor devices and applications on a network.

SNMP trap

An *SNMP trap* is an occurrence that is either broadcast or directed to a network management application, notifying the application of device or network activity. SNMP-enabled devices or applications generate traps.

SpectroSERVER

The *SpectroSERVER* is the server responsible for providing network management services such as polling, trap management, notification, data collection, fault management, and so on. This server is also referred to as the Virtual Network Machine (VNM).

DX NetOps Spectrum Extensions Integration (SEI) Toolkit

The *SEI Toolkit* is a set of tools for packaging and distributing an integration so that it can be installed on other DX NetOps Spectrum host machines.

SSORB client

The *SSORB client* is an application that accesses the DX NetOps Spectrum CORBA interface, SSORB.

super user

A *super user* is a DX NetOps Spectrum user that has all possible DX NetOps Spectrum privileges and access in OneClick. A user that has been designated a super user is automatically granted all OneClick license roles and privileges.

switched virtual circuit (SVC)

A switched virtual circuit (SVC) is a temporary connection that is established and maintained only for the duration of a data transfer session.

synchronous call

A *synchronous call* is a call to a method that performs the entire requested action before a program can continue running. The program continues only after the completion of the called method.

trap

A *trap* is an *alert* that is generated by an SNMP-compliant device.

timing interval

A *timing interval* is the frequency with which the current view updates displayed model attribute information. The default is one request per model every 5 seconds for some number of attributes.

Traffic Engineering (TE)

Traffic Engineering (TE) is the application of constraint-based routing in which a traffic engineer uses a set of link characteristics to select a route and assigns specific traffic to that route.

Universal Host Table (UHT)

The *Universal Host Table (UHT)* contains all hosts that are available for management by an ADES AIM. Using the Universal Host Table view in DX NetOps Spectrum, you can control which hosts DX NetOps Spectrum ADES Manager manages.

user (Spectrum)

A *user* in OneClick can refer to either a OneClick user account or the actual individual associated with the account. This account is created by a OneClick administrator. It provides a single OneClick user with access to OneClick and stores information about the user, such as password, access, and privileges in the DX NetOps Spectrum database.

user group

A *user group* in OneClick is a logical grouping of users that are organized together for a common purpose. Users within the same group can share the privileges granted by the group. When you specify privileges at the group level, OneClick grants each group member those privileges in addition to any privileges they have at the user level.

value variable ID

A *value variable ID* is an ID that is used to retrieve the value of a variable binding that is sent with an SNMP trap.

variable bindings

A *variable binding* is variable data that is sent as a part of an SNMP trap.

VCD (virtual CD)

A *VCD (virtual CD)* is a series of files that the DX NetOps Spectrum Extension Integration (SEI) Toolkit creates. The VCD lets developers easily package and distribute their integrations.

view

A *view* is one of many representations of the network landscape.

virtual channel identifier (VCI)

The virtual channel identifier (VCI) is a field of a cell header that contains the address of the virtual channel.

virtual channel link (VCL)

A virtual channel link (VCL) is a unidirectional method of transport for ATM cells that begins at the point where a VCI value is assigned and ends at the point where the VCI value is translated or removed.

virtual network machine (VNM)

Within the SpectroSERVER, the virtual network machine (VNM) is the software level that provides access to data regardless of where the data is stored. The data can be stored in the database, the VNM memory, or any of the managed elements on the network. The VNM also embodies the DX NetOps Spectrum intelligence that is known as the Inductive Modeling Technology (IMT).

virtual path identifier (VPI)

The virtual path identifier (VPI) is a field of a cell header that contains the address of the virtual path.

virtual path trunk (VPT)

A *virtual path trunk (VPT)* is a unidirectional method of transport for ATM cells that begins at the point where a VPI value is assigned and ends at the point where the VPI value is translated or removed. A VPT is made up of a group of VCLs with the same VPI value.

virtual technology manager

A *virtual technology manager* is the SystemEDGE agent with a virtual technology AIM loaded. Virtual Host Manager uses virtual technology managers to manage virtual devices. For more information, see the Virtual Host Manager section.

AlarmNotifier

AlarmNotifier is a SpectroSERVER-client application that installs with core DX NetOps Spectrum components. The AlarmNotifier application connects to a single SpectroSERVER and invokes scripts that provide notifications about DX NetOps Spectrum alarm status.

CAMM Presenter (Huawei SingleCLOUD)

The *Huawei SingleCLOUD CAMM Presenter* model represents a CA Mediation Manager (CAMM) Presenter. The CAMM Presenter model allows configuration of the virtual IP addresses used by the CAMM Engines to communicate with Huawei SingleCLOUD GalaX. Each CAMM Presenter model can support multiple Huawei SingleCLOUD Managers.

CNA FIP (Huawei SingleCLOUD)

The *Huawei SingleCLOUD CNA FIP* represents the management interface of the CNA that is hosting the virtual machines. This model is assigned the IP address of the CNA FIP and lives within the host container.

Computing Node Agent (CNA) (Huawei SingleCLOUD)

The *Computing Node Agent (CNA)* is an administrative process used in the Huawei SingleCLOUD platform that resides on a server that hosts virtual machines. A CNA is represented by a Huawei SingleCLOUD Host model in DX NetOps Spectrum. The Huawei SingleCLOUD CNA FIP model is assigned the IP address of the CNA.

datacenter (VMware)

A *datacenter* serves as a container for your hosts, virtual machines, resource pools, or clusters. Depending on their virtual configuration, datacenters can represent organizational structures, such as geographical regions or separate business functions. You can also use datacenters to create isolated virtual environments for testing or to organize your infrastructure.

ESX host (VMware)

An *ESX host* is a physical computer that uses ESX Server virtualization software to run virtual machines. Hosts provide the CPU and memory resources that virtual machines use and give virtual machines access to storage and network connectivity.

ESX service console (VMware)

The *ESX service console* is a Linux kernel running on the ESX host that provides a management interface to the hosted virtual machines.

Hardware Management Console

The *Hardware Management Console (HMC*) is the IBM LPAR virtualization technology application used to configure IBM LPARs. This console provides centralized management of the IBM LPAR environment.

Huawei SingleCLOUD

The *Huawei SingleCLOUD* platform is an enterprise-grade turn-key offering that consists of a complete system of network, storage, servers and software for creating private or public clouds.

Huawei SingleCLOUD GalaX

Huawei SingleCLOUD GalaX is the software suite that collectively manages the Huawei SingleCLOUD. It includes the Operation and Management Module (OMM), which is responsible for managing the Universal Virtualization Platform (UVP).

Huawei SingleCLOUD Manager

The *Huawei SingleCLOUD Manager* represents a virtual IP address on the CAMM Presenter. CAMM monitors the Huawei SingleCLOUD GalaX, which is responsible for managing the Huawei SingleCLOUD virtual platform. Information for each Huawei SingleCLOUD GalaX being monitored by CAMM is provided through a virtual IP address on the CAMM Presenter and is represented by a Huawei SingleCLOUD Manager model.

Hyper-V Host

A *Hyper-V Host* is a physical computer that uses Microsoft Hyper-V virtualization software to run virtual machines. Hosts provide the CPUs and memory resources that Hyper-V virtual machines use. They also give these virtual machines access to storage and network connectivity.

Hyper-V management operating system

The Hyper-V management operating system is the original operating system running on the Hyper-V Host. Microsoft Hyper-V uses this operating system to configure the hosted Hyper-V virtual machines.

IBM LPAR

An *IBM LPAR* is a logical partition instance configured on the IBM LPAR Host that, like a physical computer, runs an operating system and applications. An IBM LPAR dynamically consumes resources on its physical host, depending on its workload and configuration.

IBM LPAR Host

An *IBM LPAR Host* is a physical computer that uses IBM LPAR virtualization software to host IBM LPAR instances. IBM LPAR Hosts provide the CPU and memory resources that IBM LPARs use. They also give these IBM LPARs access to storage and network connectivity.

IBM LPAR Manager

The *IBM LPAR Manager* in Virtual Host Manager is the CA eHealth SystemEDGE agent with the IBM LPAR AIM enabled. The IBM LPAR Managers are responsible for reporting on all of the configured IBM LPARs. Virtual Host Manager communicates with the IBM LPAR Managers to gather details about your IBM LPAR virtual environment.

Operation and Management Module (OMM)

The *Operation and Management Module (OMM)* is part of the Huawei SingleCLOUD GalaX software application that is responsible for managing the Huawei SingleCLOUD Hypervisor Universal Virtualization Platform (UVP).

pingable model (Spectrum models)

A *pingable model* is a generic type of network model created in DX NetOps Spectrum based on a non-SNMP model type. DX NetOps Spectrum can poll these devices to provide basic model management, but SNMP-capable monitoring is not available.

Universal Virtualization Platform (UVP)

The *Universal Virtualization Platform (UVP)* is the Huawei HyperVisor, a part of the Huawei SingleCLOUD solution that consists of the hosts and virtual machines that make up its cloud architecture.

vCenter

vCenter is a VMware application that provides centralized management, operational automation, and resource optimization for ESX environments.

VHM model

A *VHM model* in DX NetOps Spectrum represents a virtual entity managed by Virtual Host Manager. Instead of retrieving status and management information from SNMP like some traditional DX NetOps Spectrum models, a VHM model communicates with the proxy manager for its fault and virtual management capabilities. A VHM model can additionally communicate via SNMP if an SNMP agent is installed and configured on the modeled device.

virtual machine

A *virtual machine (VM)* is a software computer that, like a physical computer, runs an operating system and applications. A virtual machine dynamically consumes resources on its physical host, depending on its workload. Because virtual machines are flexible computing units, their deployment comprises a wide range of environments. Examples include environments such as data centers, cloud computing, test environments, or desktops and laptops. In data center implementations, they are used for server consolidation, workload optimization, or higher energy efficiency.

VMware Manager

The *VMware Manager* in Virtual Host Manager is the CA eHealth SystemEDGE agent with the vCenter Server AIM loaded. The VMware Manager is responsible for reporting on all of the configured virtual machines it manages. Virtual Host Manager communicates with the VMware Managers to gather details about your VMware virtual environment.

resource pool (VMware)

A *resource pool* defines partitions of physical computing and memory resources of a single host or a cluster. You can partition any resource pool into smaller resource pools to divide and assign resources to specific groups or for specific purposes. You can also hierarchically organize and nest resource pools.

Center Server (VMware)

VMware vCenter Server provides the central point of control for configuring, provisioning, and managing a virtual vSphere environment. vCenter Server runs as a service on Microsoft Windows Servers and Linux Servers.

virtual NIC (VMware)

A *virtual NIC* is a virtual Ethernet adapter on a virtual machine. The guest operating system communicates with the virtual Ethernet adapter through a device driver as if the virtual Ethernet adapter was a physical Ethernet adapter. The virtual Ethernet adapter has its own MAC address, one or more IP addresses, and responds to the standard Ethernet protocol like a physical NIC.

main location server (MLS)

The *main location server (MLS)* is the primary SpectroSERVER used to coordinate the information and events from all other SpectroSERVERs connected in a DSS environment.

Provider Edge (PE)

Provider Edge (PE) typically refers to the customer-facing router that is managed by the ISP. These routers interface with the customer network to carry network traffic from one customer server to another within the same LAN.

Service Level Agreement (SLA)

A *Service Level Agreement (SLA)* is typically a contractual agreement between a business and its subscribers guaranteeing a specified level of service, such as 99.9 percent availability. An SLA sometimes contains a penalty clause for noncompliance.

Virtual Forwarding Instance (VFI)

A *Virtual Forwarding Instance (VFI*) is a logical collection of VPLS Sites that are part of the same virtual packet forwarding instance. These VPLS Sites share a common Layer 2 forwarding database, much the same way that Layer 3 devices may share a common routing table.

Virtual Private LAN Service (VPLS)

Virtual Private LAN Service (VPLS) is a way to provide Ethernet-based multi-point to multi-point communication over IP/MPLS networks.

VPLS Site

A *VPLS Site* represents the service delivered to a customer over an interface in your VPLS environment.

Microsoft Operations Manager (MOM)

Microsoft Operations Manager (MOM) is an application that provides real-time events and alerts on the performance of Microsoft Windows servers and applications in your network.

MOM agent

The *MOM agent* is installed agent software that resides on a system managed by MOM. A MOM agent collects system events and statistics.

MOM Connector

MOM Connector is an application that synchronizes alarm data between DX NetOps Spectrum and Microsoft Operations Manager. The MOM Connector uses the DX NetOps Spectrum SSORB CORBA API to interface with DX NetOps Spectrum. It uses the MOM Connector Framework (MCF) to interface with Microsoft Operations Manager.

MOM host server

The *MOM host server* is responsible for management tasks, such as event-based alert generation.

SCOM agent

SCOM agent is installed agent software that resides on a system managed by Microsoft System Center Operations Manager. A SCOM agent collects system events and statistics.

SCOM Connector

The *SCOM Connector* is an application that synchronizes alarm data between DX NetOps Spectrum and System Center Operations Manager. The SCOM Connector uses the DX NetOps Spectrum SSORB CORBA API to interface with DX NetOps Spectrum. It uses the Operations Manager Connector Framework to interface with SCOM.

SCOM host server

SCOM host server is responsible for management tasks, such as event-based alert generation.

System Center Operations Manager (SCOM)

System Center Operations Manager (SCOM) is an application that provides real-time events and alerts on the health of servers and applications in your IT environment.

Use the CA Remote Engineer Tool to Collect Troubleshooting Data

CA Remote Engineer is a tool that automates the collection of diagnostic data for installed CA software products. CA Remote Engineer scans the system, collects log or configuration file information packages, and transmits the file to CA Support. This tool helps technicians troubleshoot problems with the system or software. This diagnostics tool simplifies the diagnostics process and ensures business continuity. The tool collects and shares CA Support Engineers the exact data they need from your environment to resolve the issues faster. The metrics that are gathered by Remote Engineer are protected by the customer support agreement which states that the gathered information is deleted within 30 days of the resolution of an issue.

The CA Remote Engineer tool collects configuration and diagnostic data that helps CA Support Engineers troubleshoot problems with the system or software.

Collect Troubleshooting Information

CA Remote Engineer collects the following configuration and diagnostic data to troubleshoot problems with the system or software:

- Product information: Version, product logs, and configuration information
- Hardware information: CPU, memory, ports, IRQ, interrupts, devices, and disk partitions
- Operating system information: OS version, patch levels, network information, disk information, CA product registry keys, running process information, and environment variables

Use CA Remote Engineer for DX NetOps Spectrum

From 10.3.1, the CA Remote Engineer tool is shipped with the product. The **CA_RemoteEngineer.zip** file is available in the DX NetOps Spectrum installation folder, extract the .zip file to a preferred location.

Perform the following steps to run Remote Engineer and collect troubleshooting data for your DX NetOps Spectrum:

1. Log in to the DX NetOps Spectrum computer where you want to run the diagnosis tool.
2. Unzip the **CA_RemoteEngineer.zip** file to a proper location on that server and go to the newly created '**RemoteEngineer**' directory.
3. Run the **RemoteEngineer.exe** file.
4. Select **CA Spectrum** from the **Product Name** drop-down.
5. Select the **Run Diagnostics** button.
6. Enter the Spectrum Installation Drive details then select **Enter**. The tool starts diagnosing for the following details of Spectrum:
   a. OS Version
   b. CPU
   c. Memory
   d. Disk Space
   e. Product Version
   f. Product Checks details.

   CA Remote Engineer collects troubleshooting data for the server and stores in the diagnostic_results folder in the Remote Engineer directory.

CA Remote Engineer Output

CA Remote Engineer writes the information that it collects in a local zip file in the CA Remote Engineer installation directory. Navigate to the **diagnostic_results** folder to review the results and detailed troubleshooting information.

Output file format:

- Product information:
  - Logs and configuration files (located in the same locations as in the installed product file structure)
  - CA Registry output in caprodinfo.xml
- Operating system and hardware information:
  - diaginfo.txt
  - machineinfo.xml

For more information about CA Remote Engineer, see the CA Remote Engineer Product Information page.

CA Green Books

CA Green Books provide knowledge focused on CA solution implementations and deployments. They deliver best practices and considerations based on real-world scenarios and knowledge compiled from global CA team experiences. Cross-functional teams of CA technical employees from field services, support and education collaborate with Technical Information to apply their expertise and create publications that deliver practical knowledge that goes beyond an "out-of-the-box" installation. Here is the link to the DX NetOps Spectrum Green Book:

Troubleshooting section for CA Support Partners

Documentation Legal Notice

Non-SNMP Monitoring

DX NetOps Mediation Manager monitors the performance for non-SNMP based devices, such as mobile wireless, fiber-optic switch, radio access, and 3G or 4G voice data. DX NetOps Mediation Manager supports a wide range of protocols to access data, for example, SOAP, SSH, XML, SQL, JMS, SFTP, and HTTP. DX NetOps Mediation Manager is portable across all platforms.

This section contains everything you need for non-SNMP monitoring from getting started to troubleshooting information.

Release Notes

Contents

20.2 Service Pack 2 (SP 2) New Features and Enhancements

This release of DX NetOps Mediation Manager includes the following new features and enhancements.

CA5620SamY1731 device pack enhancements

- Included jms plugin along with the engine.
- Splitted findToFile big xml file (size around 300MB) into small chunks (chunk size can be configurable) before performance conversion.
- Performance conversion logic enhanced to consider the logToDBAndFile stats as well.
- Optimized the performance conversion logic.
- Added new variable **SPLIT_GROUP_SIZE** with default size **2500** in Global Variables.

  NOTE
  Maximum Java Heap size can be configurable based on the findToFile file size.

  If findToFile size is around 300 MB then recommended MAXIMUM_JAVA_HEAP_SIZE value as 6144.

20.2 New Features and Enhancements

This release of DX NetOps Mediation Manager includes the following new features and enhancements.

New Device Packs

This release supports the following device packs:

CASBC Devicepack

This device pack collects performance metrics of:

- CallRecord
- SummaryRecord
- VoiceQualitySummaryRecord

CA5620SamY1731

This device pack collects cfmTwoWayDealy SAS metrics of SAM 5620 (NFM-P) through FindToFile and LogToFile by enabling JMS subscription. It collects the following performance metrics.

- MinimumResponse
- MaximumResponse
- SuccessfulAttempts
- MaxOneWayDelaySrcDest
- MinOneWayDelayDestSrc
- MaxOneWayDelayDestSrc
- MinOneWayDelaySrcDest
- PacketsLost
- PacketsSent
- PacketsArrivingAfterTimeout
- AverageJitter
- AvgResponseTime
- JitterOut
- JitterIn
- Jitter
- DelayVariation
- AvgOneWayDelaySrcDest
- AvgOneWayDelayDestSrc
- Latency
- AveragePercentPacketLoss

CAPaloAltoPanOS

This device pack collects data for the PaloAlto devices from the api and shows the performance stats for:

- Utilization
- SessionAverage
- SuccessfulAttempts
- SessionMaximum
- PacketBufferAverage
- PacketBufferMaximum
- PacketDescriptorAverage
- PacketDescriptorMaximum
- PacketDescriptorOnchipAverage
- PacketsArrivingAfterTimeout
- PacketDescriptorOnchipMaximum

Enhancements

The following device pack enhancements have been made in this release:

1. CAAVIController
   - AVI Controller device pack updated to support non-snmp devices for those controllers which are hosted in the cloud. Since DNS of the cloud devices maps to multiple IP's.
   - AVI controller device pack updated to support following controller metrics
     - controllerstatsavgcpuusage
     - controllerstatsavgdiskusage
     - controllerstatsavgmemusage

New Features

This release of DX NetOps Mediation Manager includes the following new features

- Delete devices from Component in DX NetOps MM using any API development environment
  You can delete one or more devices discovered in the device pack component using the delete API.

  NOTE
  Before Deleting the Devices, Stop the Engine.

  Scenario:
  – Primary Multi Controller is installed at **DX_NetOps_MM_HOST** with the default web server port number (8880).
  – Two Local Controllers are connected to the primary Multi Controller.
  – Delete device(s) from CAVerizonSatellite device pack

  Request URL Format:
  http://<DX_NetOps_MM_HOST>:<DX_NetOps_MM_WEB_PORT>/tim-web/rest/components/<Component_Id>/devices

  For the given scenario:
  http://localhost:8880/tim-web/rest/components/<Component_Id>/devices

  Content Type: Text or XML

  HTTP Verb: DELETE

  Sample Request Body:
  <Device>SAT-Device4-75644</Device>
  <Device>SAT-Device5-75645</Device>
  <Device>SAT-Device8-75648</Device>

  Sample Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
     <response url="components/ENGINE_CAVerizonSatellite/devices">
        <status>
            <code>200</code>
            <message>Success</message>
        </status>
    </response>
```

- Apache Tomcat upgraded to 8.5.51 version to address **Ghostcat CVE-2020-1938** issue related to AJP connector.

Known Issues

The following are the known issues for DX NetOps Mediation Manager 20.2.

- **The device pack does not appear in CA Performance Center 2.7 and later version.**
  This issue occurs when you upgrade CA Performance Management from 2.6 or lower version to 2.7 and later version but use CA Mediation Manager 2.6 or lower version.
  To resolve this issue, follow these steps:
  a. Log in to Mediation Manager after you upgrade to version 3.7, and click the **Device Packs** tab.
  b. Select the sub component ***PRESENTER_CAPM*****,** and click **Stop.**
  c. Ensure that the ***PRESENTER_CAPM*** is selected, and click
     Advanced
     The PRESENTER_CAPM window appears **.**
  d. Click **Configuration** on the right pane.
  e. Update the value in **OUTPUT_DIRECTORY** as follows:
     /opt/IMDataCollector/apache-karaf-2.4.3/MediationCenter/
  f. Click **Save**, **Close**.

g. On the Device Pack window, select **PRESENTER_CAPM,** and click **Start**.

WARNING

This issue can also occur if Mediation Manager is upgraded from version 2.7 to version 20.2 but **PRESENTER_CAPM** is not upgraded. In that scenario, either upgrade PRESENTER_CAPM or manually modify the OUTPUT_DIRECTORY value as specified in step 5.

- **DX NetOps Mediation Manager versions 2.4, 2.5, 2.6, 2.7, 2.8 fails to deploy device pack certificates in CA Performance Management version 3.0 or later.**
  This is a compatibility issue. To resolve this issue, upgrade CA Mediation Manager to version 3.0 or later because the Mediation Manager versions 2.4 through 2.8 are not compatible with CA Performance Management version 3.0 or later.
- **DX NetOps Mediation Manager** on Windows and Solaris Supports only Multi-Controller
  Even if the DX NetOps Mediation Manager Local Controller is installed on Windows and Solaris, you must manually copy the Queue folder (where the input files get collected) to the Data collector.
- **CAAVIController device pack supports single AVI Controller**
- **CERT deployment fails when the corresponding CERT folders are zipped at the parent level**
  The CERT deployment fails when the following folders are zipped at the parent level (CERT_<DevicePack_Name>).
  – components
  – device_mapper
  – metric_families
  – vendor_certs
  As a workaround, ensure that you zip the preceding four folders.
- While installing Mediation Manager in Windows environment, make sure that JRE path should be an environment variable with JAVA_HOME name and bin directory of JRE shall be in path.

Third-Party Software Acknowledgements

Third-party software was used in the creation of DX NetOps. All third-party software has been used in accordance with the terms and conditions for use, reproduction, and distribution as defined by the applicable license agreements. This section contains all third-party software license agreements for applications that are included as part of the current release of DX NetOps.

The following license agreements are available as an attachment. Click here to download the license agreements.

- AdoptOpenJDK_v8
- Ant 1.10.5
- aopalliance 1.0
- Apache Commons FileUpload 1.4
- Apache Commons DbUtils 1.6
- ASM 3.2
- Bootstrap 3.3.2
- cglib-nodep 2.1.3
- Commons Cli 1.2
- Commons Codec 1.11
- Commons Collections 3.2.1
- Commons Logging 1.1.1
- Commons net 3.6
- commons-beanutils 1.9.3
- commons-compress 1.18
- commons-io 2.6
- commons-lang 2.5
- Derby 10.14.2.0
- dom4j 1.6.1
- End User License Agreement (EULA) r5.1.3-GA
- extjs 3.4.0
- Ganymed SSH-2 build210
- google-gson 1.6
- gpars 1.2.1
- Groovy 2.5.5
- Javamail 1.4.4
- Jaxen 1.1 B8
- JBoss Netty 3.2.10 Final
- JDOM 1.0
- Jersey 2.25.1
- json-lib 2.4
- JSTL (The JSP Standard Tag Library) 1.1.2
- JSTL Reference Implementatoin 1.2
- Log4j 1.2.9
- Log4j 1.2.16
- Python
- RSyntaxTextArea 2.0.3
- Saxon-B 9.1.0.8
- SBLIM 2.1.3
- ServingXML 1.1.2
- SNMP4J 1.10.1
- SNMP4J-Agent 1.3.1
- Spring Framework 3.2.9 Release
- Spring Security 3.2.9 Release
- Tomcat 8.5.37
- Tomcat 8.5.51
- TrueZIP 6.6
- Velocity 1.7
- xercesImpl 2.6.2
- xml-apis 1.0.b2

Collecting Data for DevicePack Generation

DX NetOps Mediation Manager (DX NetOps MM) monitors the performance of non-SNMP devices in Element Management System, such as, mobile wireless, fiber-optic switch, radio access, and 3G or 4G voice data. CA Mediation Manager supports a wide range of protocols to access the data.

For example, REST, SOAP, SSH, XML, SQL, JMS, (S)FTP, and HTTP(S).

Element Management System

An Element Management System (EMS) consists of systems and applications for managing Network Elements (NE) on the Network Element-Management Layer (NEL) of the Telecommunications Management Network (TMN) model. Provide the following details of EMS.

- **Name of the EMS:** <ems_name>
- **Vendor Name:** <vendor_name>
- **Version:** <version_number>

For example,

- **Name of the EMS:** <Service Aware Manager (SAM) 5620>
- **Vendor Name:** <Nokia (Alcatel)>
- **Version:** <14R1>

Data Collection

Data collection provides details of DX NetOps MM DevicePack that collects data from the EMS system. DX NetOps MM supports SFTP, FTP, HTTP(S), REST, JMS, JDBC communication protocols.
Example: If an EMS creates archive of hourly data and the data is placed in an FTP Server, then the following details are required:

- **Communication Protocol:** <FTP/SFTP>
- **Hostname/IPAddress:** <CAMMSIM.ca.com>
- **(if applicable) Port:** <22>
- **Username:** <camm_ftp_user>
- **Password:** <ftp_password>

If the EMS has any proprietary communication protocol, provide the EMS Integration or API Guide.

Data Format

Provide the details in a data format.
For example, CSV, XML, JSON, COMMANDPROMPTOutput, or others.
Example of CSV data file format:

DeviceID	DeviceName	CollectionTime	CPU (%)	Memory Utilization (MB)	Total Memory (MB)
3145741	NE A	9/19/2016 22:15	10	20	1024
3145742	NE B	9/19/2016 22:15	20	30	1024

(Optional) Granularity of Data

The time period of receiving the data from the EMS.

For example, 5 minutes, 10 minutes or 60 minutes, and so on.

Frequency of Data file Generation

The frequency of the time the data file is generated.

For example, 5 minutes, 10 minutes, or 60 minutes, and so on.

Sample Data

The location of the sample files.
Example: CA Support Case FTP, any public Cloud Drive location such as, Microsoft OneDrive, Dropbox, and so on.

Metrics

The metric is the name that you want to view in the CA PC reports. If the columns are self-explanatory in the CSV data format file, then the metric name is not required.
Example:

The following table describes the metrics for the table mentioned in the **Data Format** section.

Metric Family Name	Metric Name	Units
EMS_Name_DeviceHealth	CPU	Percent
EMS_Name_DeviceHealth	Total Memory	Bytes

Units possible values: Percent, Packets, PacketsPerSecond, DiscardedPackets, ErroredPackets, Bits, BitsPerSecond, Bytes, BytesPerSecond, Seconds, Microseconds, Milliseconds, UnixTime, Observations

KPI Formulas

If you want DX NetOps MM or CAPM to derive any KPIs on top of the out of the box metrics, then provide the KPI formulas.
For example, Memory Utilization % = Memory Utilization/Total Memory x 100>

More Details to Develop a Device Pack

Provide any other details which help to develop a Device Pack.
For example, API Guides, EMS Vendor Contact details, VPN to your LAB where the EMS is hosted, and so on.

Getting Started

DX NetOps monitors the performance for non-SNMP based devices, such as mobile wireless, fiber-optic switch, radio access, and 3G or 4G voice data. DX NetOps supports a wide range of protocols to access data, for example, SOAP, SSH, XML, SQL, JMS, SFTP, and HTTP. DX NetOps is portable across all platforms.

Architecture and Components

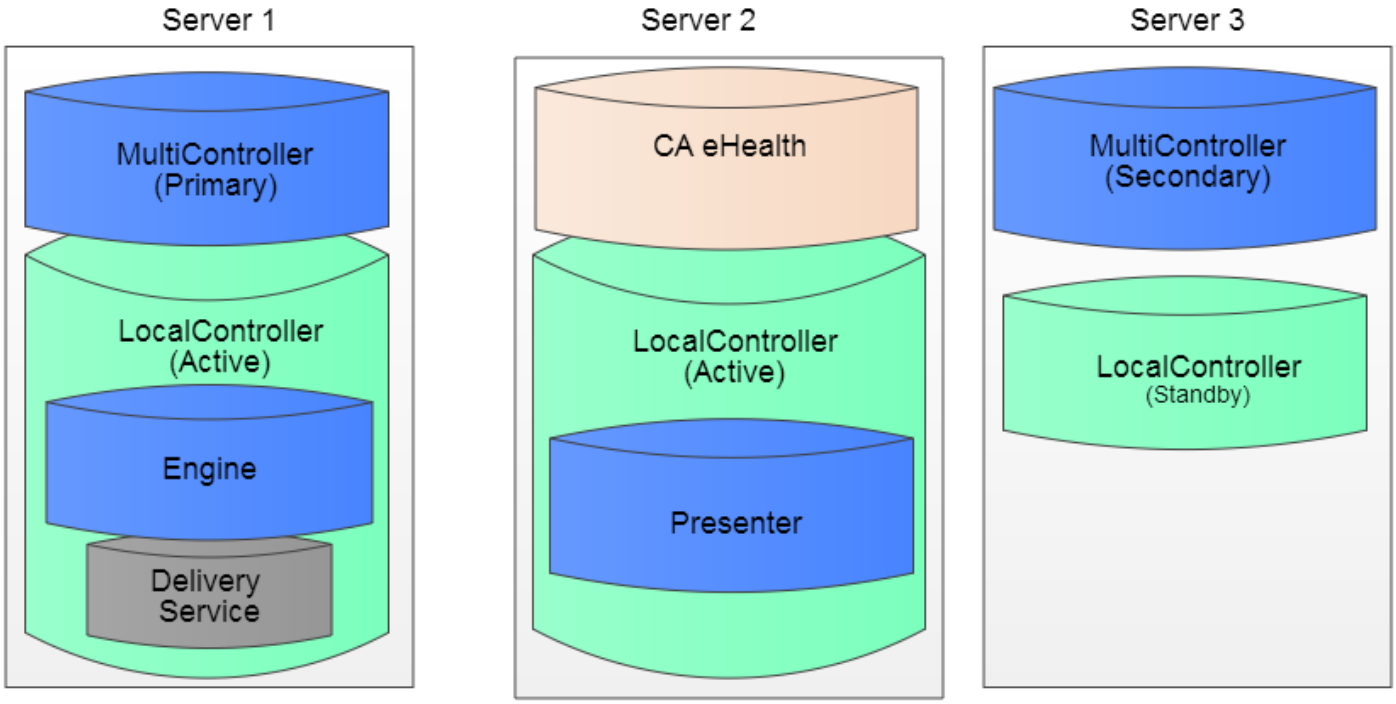
This article contains the following topics:

The architecture lets you quickly develop any required vendor-specific API plug-ins, named device packs, to collect the data directly from a device or from an Element Management System (EMS). DX NetOps enables deployment in a standalone or distributed architecture.

In a standalone deployment, the installation is usually performed on the management server being integrated. In a distributed deployment, the application is deployed across multiple servers to meet the high demands of current Communication Service Provider (CSP) environments.

The following diagram describes the general architecture of DX NetOps for CA eHealth in a multiserver installation:

Figure 83: DX NetOps Mediation Manager architecture--XML



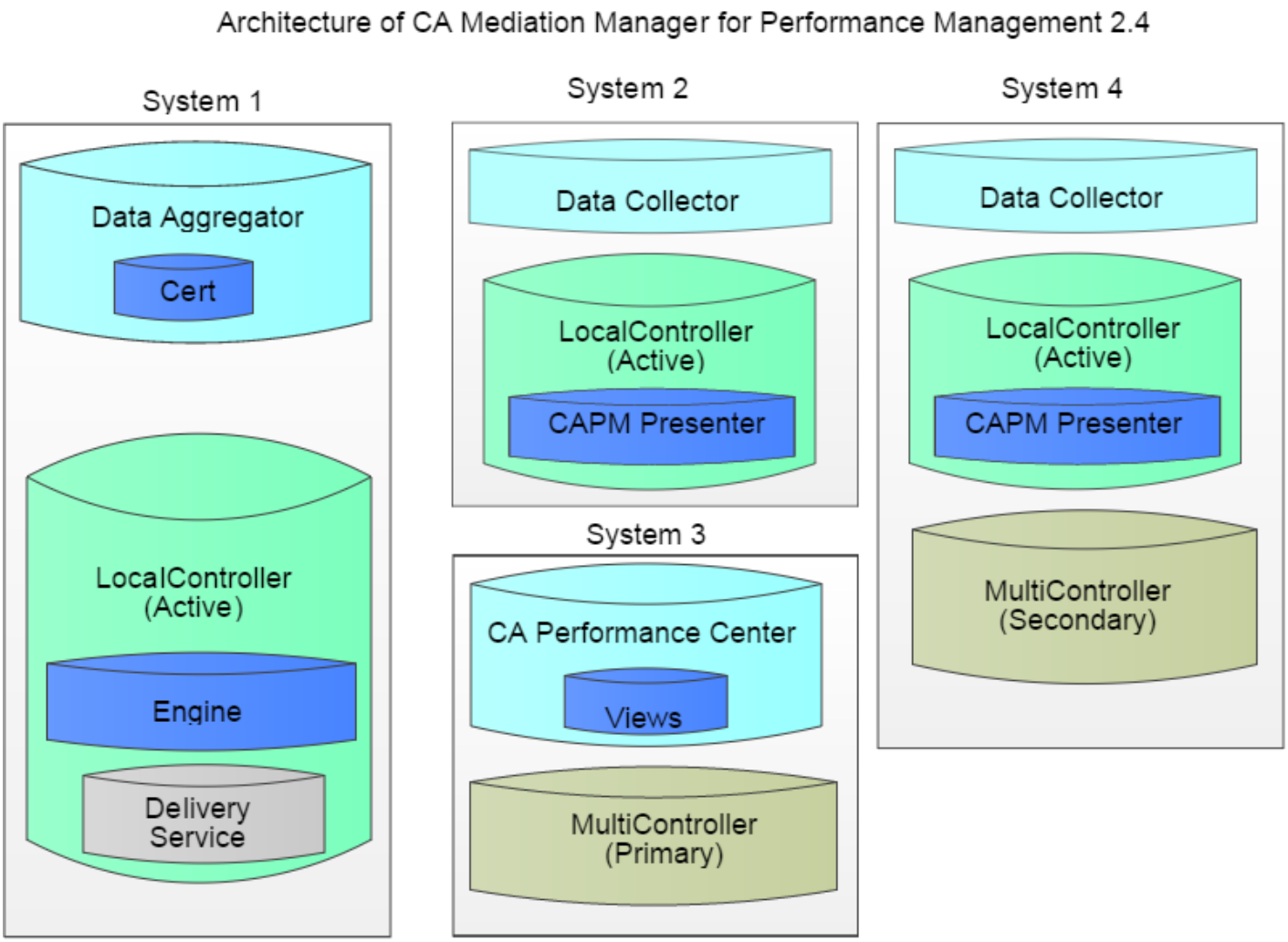
When you install DX NetOps Mediation Manager for Performance Management 2.3.4 or later in distributed systems,install the components of Performance Management and DX NetOps Mediation Manager as follows:

- Data Aggregator, Data Collector, and CA Performance Center in individual systems.
- Cert and Views respectively in the system that has Data Aggregator and CA Performance Center.
- CAPM Presenter and LocalController in every system that has a Data Collector.
- Primary MultiController in a system that has CA Performance Center.
- LocalController with Engine and Delivery Service in another system that has Data Aggregator.

> **NOTE**
> Install Primary MultiController and LocalController with Engine and Delivery Service in different systems based on the RAM and CPU sizes of the systems. If there is no sufficient memory, install Primary MultiController and LocalController with Engine and Delivery Service in individual systems.

The following diagram describes the general architecture of multiserver installation of DX NetOps for Performance Management with two Data Collectors:

Figure 84: DX NetOps MM for PM in 4 Systems



NOTE
An extra component, the Generic Executor is not shown in both the diagrams, but is described in this document.

The components of DX NetOps include the following set of core software entities:

- MultiController
- LocalController
- Web Manager
- Generic Executor
- Delivery Service

The subcomponents of DX NetOps include the following entities:

- Engine
- Presenter or CAPM Presenter

MultiController

You can deploy up to two MultiControllers, primary and secondary, in a cluster. Deploy at least one MultiController per cluster. A MultiController performs the following actions in your cluster environment:

- Monitors heartbeat messages from LocalController components on remote servers.
- Acts as the centralized licensing server for the cluster.
- Stores centralized configuration files for components in the cluster.

LocalController

Install one LocalController on each physical server in the cluster where a subcomponent (Engine or Presenter) resides. A LocalController performs the following actions:

- Communicates with the subcomponents that are installed on the server.
- Monitors heartbeat messages for subcomponents on the local server and automatically restarts subcomponents if they fail.
- Uses a delivery service to process output from the Engine. This service delivers XML documents in a compressed and encrypted format to a local or remote Presenter.

Web Manager

The Web Manager lets you manage the device pack deployment using its web-based interface. The interface displays the following information:

- Status of the running device packs.
- Status of the LocalControllers on which the device packs are installed.
- Status of the primary and secondary MultiController.

Generic Executor

All components in a cluster share a common set of functions for communication and execution. The Generic Executor starts the Engine and Presenter subcomponents and cleans the temporary and log files.

The Generic Executor starts at system startup and listens on a specific TCP port. To start a component like the MultiController, the DX NetOps Control Utility, cammCtrl, sends a MultiController XML configuration file to the Generic Executor. When the Generic Executor receives this data, it identifies and starts the MultiController component using the information in the configuration file.

Delivery Service

When an Engine finishes its poll cycle, it generates one or more DX NetOps-standard XML documents in the queue directory. The Delivery Service monitors the queue directory independently and distributes the data to one or more local or remote Presenters. Delivery Service determines the correct Presenter by sending a request to the MultiController through the LocalController and identifies the IP address and TCP port of the Presenter. The Delivery Service does not process the queue until the local or remote Presenter becomes available.

Engine

The Engine is the main, threaded polling engine in DX NetOps. You can deploy the Engine in the active or standby mode. The Engine performs the following actions:

- Gathers information from devices using XML, CSV, Telnet, SSH, and so on. The engine then processes the data to a DX NetOps-standard XML document.
- Deploys the DX NetOps-standard XML document to the queue for processing by the Delivery Service.

Presenter

The Presenter is required only for DX NetOps for eHealth and one Presenter is required for each device pack. It is a threaded presentation engine that performs the following actions:

- Receives the DX NetOps-standard XML document from the Engine.
- Formats the data to the required output format, such as CSV, XML, SNMP, and DDI.

CAPM Presenter

The CAPM Presenter is required only for DX NetOps for CA Performance Management. The CAPM Presenter must be installed in Data Collector that supports DX NetOps. Similar to the Presenter, CAPM Presenter receives the DX NetOps-standard XML document from the Engine. But, CAPM Presenter does not format the data to a required output format.

The number of CAPM Presenter depends on the number of complex device packs.

Default Ports and Data Flow

DX NetOps Default Ports

The following lists the default ports:

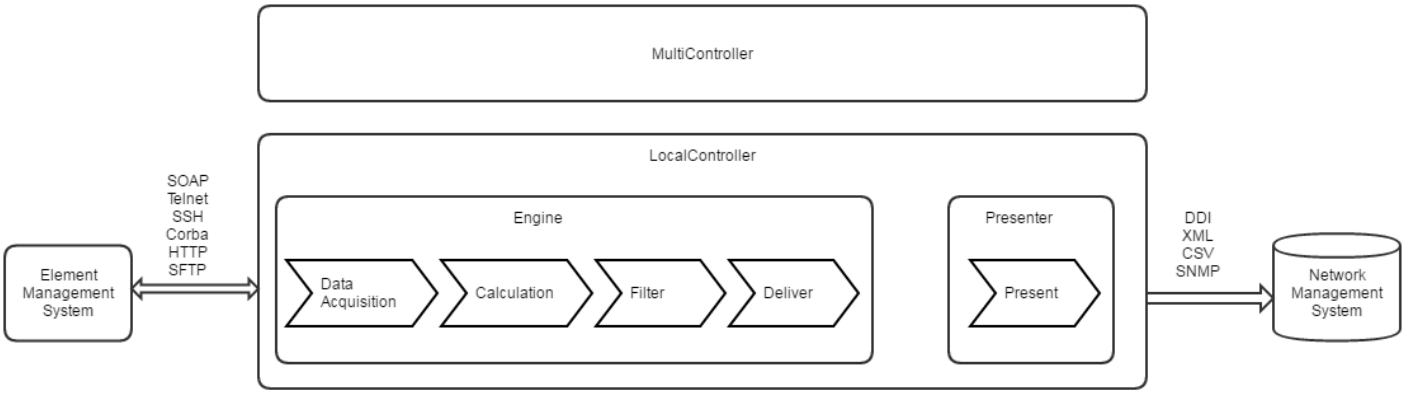
Component	Port	Type	Comments
MultiController	29599	TCP	Communicates with LocalController via heartbeats to maintain status.
LocalController	29598	TCP	Communicates with MultiController via heartbeats to maintain status.
Generic Executor	29560	TCP	Listens on this TCP port to receive XML configuration files from MultiController and LocalController.
Web Manager	8880	HTTP	Administrators browse to the Web Manager (located on the MultiConroller) server over port 8880.

Data Flow

- Engine polls the Element Management System.
- Engine converts data to XML format.
- Engine performs any calculation or filtering required.
- Engine delivers data in CAMM2XML format to the queue directory.
- Deleivery Service reads queue and transfers XML file to the Presenter(s) (Local or Remote) using the name service.
- Presenter reads the XML file and outputs the data in the required format.

The following diagram shows the typical data flow in DX NetOps.

Figure 85: Data Flow in CAMM



Installing

This section helps you install DX NetOps, device pack in DX NetOps, and device pack in DX NetOps for Performance Management.

DX NetOps Mediation Manager is a distributed application that includes multiple components across several servers. A successful deployment includes installing the these components in the following order:

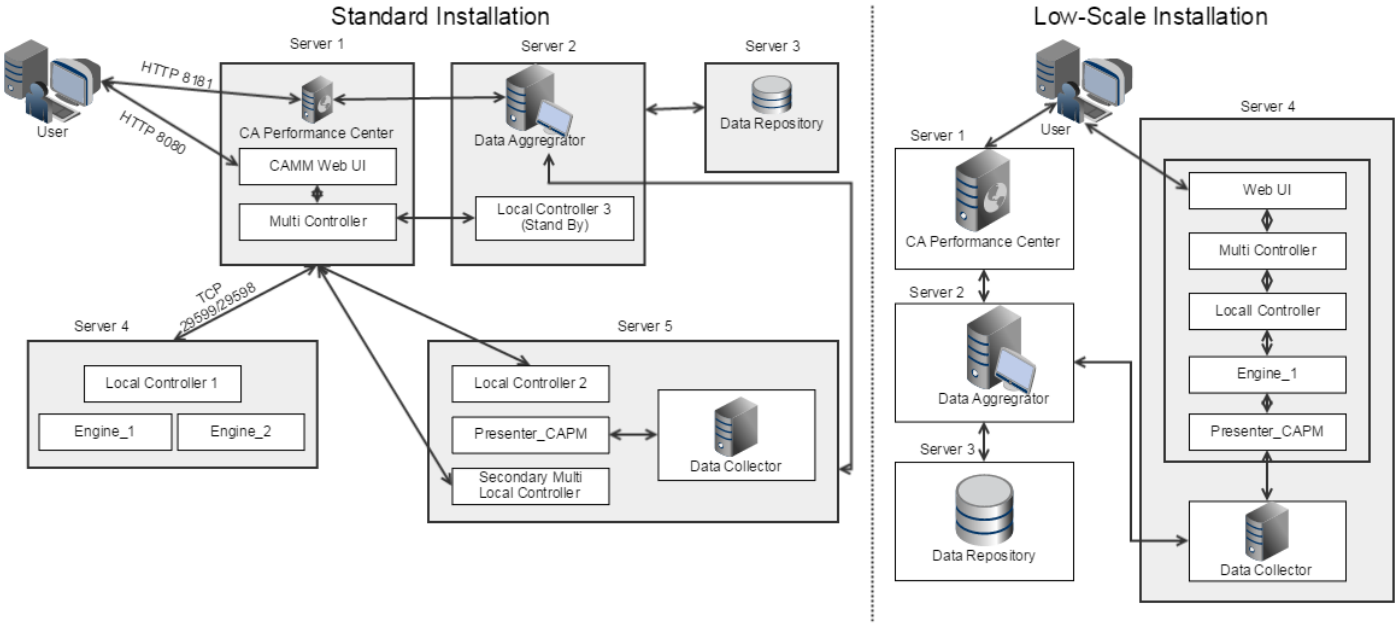
- CAPM (CAPC, DR, DA & DC)
- DX NetOps MM Multi Controller
- DX NetOps MM Local Controller
- DX NetOps MM Local Controller for Presenter on DC system
- DX NetOps MM Secondary Multi Controller*
- DX NetOps MM Local Controller for LC failover*

NOTE

The *** (asterisk)** indicates the failover of MC & any LC.

Your deployment strategy depends on the number of devices, the location of these devices, and which metrics you want to monitor. The following diagram shows the installation options:

Figure 86: DX NetOps MM Standard and Low-Scale Installation



System Requirements

Review the following information before you install the product:

Hardware Requirements

The following table describes the minimum hardware requirements for each supported operating system:

Operating System	Architecture	CPU	Memory	Disk
SUSE Linux Enterprise Server (SLES) 12 SP2	-	-	-	-
Oracle Linux (OL) 7.3 (Red Hat compatible kernel only)	-	-	-	-
Solaris 9 or 10	SPARC (64-bit)	1 x 1.4 GHz	4 GB	18 GB
Red Hat Enterprise Linux 6.x Red Hat Enterprise Linux 7.3	64-bit	1 x 2 GHz	4 GB	18 GB
Windows 2008	32-bit, 64-bit	1 x 2 GHz	4 GB	18 GB
Windows 2012	64-bit	1 x 1.8 GHz	4 GB	18 GB

NOTE
Maintain consistency between the JRE and the operating system architecture. For example, on 64-bit operating systems, the JRE you use to install and run DX NetOps must also be 64-bit.

Web Browser Requirements

A web browser is required to access DX NetOps Mediation Manager UI (User Interface). The following browser applications are supported:

- Microsoft Internet Explorer version 9 or above
- Mozilla Firefox (current version)
- Google Chrome (current version)

Java Support

- AdoptOpen JDK 1.8.0.212 (From CA Mediation Manage 3.7 SP 3)
- Oracle JRE 1.8.x (CA Mediation Manager 3.7)

Supported Platforms

CA Mediation Manager supports the following platforms:

- CA eHealth 6.x
- CA Performance Management r2.7 through 20.2

Compatibility Matrix

The following table provides the compatibility matrix for DX NetOps Mediation Manager:

	CA Performance Management 3.1	CA Performance Management 3.2	CA Perf
CA Mediation Manager 3.1	Compatible	Compatible	Compati
CA Mediation Manager 3.2	Compatible	Compatible	Compati
CA Mediation Manager 3.5	Compatible	Compatible	Compati
CA Mediation Manager 3.6	Compatible	Compatible	Compati
CA Mediation Manager 3.7	Compatible	Compatible	Compati
CA Mediation Manager 20.2	Compatible	Compatible	Compati

*Check the Known Issue section in Release Notes.

Install DX NetOps Mediation Manager

Use this article to install DX NetOps on the Windows or UNIX systems. This article contains the following topics:

Installation Planning

The installation prerequisites are as follows:

- Ensure that users of DX NetOps and CA Performance Management are in the same user group.
- Administrator privileges on Windows systems
- Comply with the system requirements, web browser requirements, supported platforms, and compatibility matrix. For more information, see System Requirements.
- Java Runtime Environment (JRE) version 1.8 or later
  We recommend that you obtain the latest version of JRE from the Java download site.

Install DX NetOps

This procedure describes the steps to install DX NetOps on Windows or UNIX systems.

Follow these steps:

1. Log in to the server with the user ID that you want the DX NetOps processes to use.

   NOTE
   We refer to the user ID throughout this document as CAMM_USER. If you use an invalid user ID, an error occurs and the installation fails.
2. Start the DX NetOps installation with the corresponding command based on your operating system:
   - UNIX systems
     ```
     $JAVA_HOME/bin/java  -jar CAMM-Installer-20.2.x.jar
     ```
   - Windows systems
     a. Insert the installation CD-ROM and open Windows Explorer.
     b. Locate and double-click the executable JAR file, CAMM-Installer-20.2.x.jar.
3. Click Next on the Welcome dialog.
4. Read the Important Information and click Next.
5. Review the licensing agreement, accept, and click Next.
6. Specify a target path for the installation. Or click Choose to browse to an installation location.

   NOTE
   When installing on a Windows System, the default installation directory is C:\Program Files\CA\CAMM.
   When installing on a UNIX System (Linux or Solaris), the default installation directory is /opt/CA/CAMM.
7. Select Typical for the new installation.
8. Select one or more required installation packages.
   - MultiController
   - LocalController
9. Configure the following DX NetOps foundation parameters:
   - **User ID**
     Specifies the user ID for the Generic Executor for the DX NetOps installation. The user ID is the CAMM_USER and defaults to the current user ID.
     Default: root

     WARNING

     Use the same user ID that you used to install Data Collector or the user must have read and write group access to $DATA_COLLECTOR_INSTALL/apache-karaf-x.y.z/MediationCenter.
   - **Port**
     Specifies the port where the Generic Executor listens.
     Default: TCP port 29560
10. Specify whether you install primary or secondary MultiController.
11. (Optional) Configure the following parameters of primary MultiController:

    NOTE
    For the first installation, configure the primary MultiController. You can configure the secondary MultiController in the subsequent step.

    - **MC IP**Specifies the IP address or host name for the primary MultiController.

      NOTE
      Make sure that the host name is resolved to an IP address for to function properly. supports IP V4.
    - **Failover Threshold**Determines the elapsed time, in seconds, with no received heartbeat messages, in seconds, and signals LocalController failure. When this threshold is reached, the MultiController activates the standby LocalController.
      Default: 600
    - **Will the other MC exist in the cluster?**Indicates that another MultiController exists or may exist in the cluster.

12. (Optional) Configure the following parameter for another MultiController in the same cluster:
    - **The Other MC IP**
      Specifies the secondary MultiController IP address, if you installed and configured a primary MultiController.
      If you installed a secondary MultiController in a previous step, this value is the primary MultiController IP address.
      If this server is a backup server, install the secondary MultiController on a different host server.
13. Configure the following parameters for the web authentication:
    - **User**
      Specifies the login name for the DX NetOps Web Manager.
      Default: admin
    - **Password**
      Specifies the login password for the DX NetOps Web Manager.
      Default: camm
    - **Web Port**
      Specifies the port for the web UI.
      Default: 8880
14. (Optional) Configure the following parameters for LocalController. At least one LocalController is required on any server where the Engine and Presenter subcomponents are running.
    - **LC IP**
      Specifies the IP address or host name for the LocalController.

      NOTE

      Make sure that the host name is resolved to an IP address for DX NetOps to function properly.

      DX NetOps supports IP V4.
    - **LC Port**
      Specifies the port for this LocalController.
      Default: 29598
    - **Failover Threshold**
      Determines the time (in seconds) that can elapse with no received heartbeat messages. The loss of heartbeat messages indicates the subcomponent failure. When the threshold is reached, the LocalController restarts the Engine, Presenter, or both components.
      Default: 600
15. Review the installation summary and click Install.
16. Click Done.
    The installation is complete.
17. Navigate to $CAMM_HOME and check the installer information in the version.xml file. The contents of the version.xml file resemble the following format:

```
<?xml version="1.0" ?>
<CAMM-Version>
   <Current>
     <Version><20.2.x>/Version>
     <Revision><Installer revision number></Revision>
   </Current>
</CAMM-Version>
```

18. To start the application, type http://<*PrimaryMCMachineIP*>:<*web-port*> in a browser.
    Where <*web-port*> is the port number that is configured during the DX NetOps Installation and <*PrimaryMCMachine IP*> is the IP address or hostname of the primary MultiController system.
19. Type the login credentials.
    The DX NetOps application starts.

    NOTE
    For more information about using the options in UI, see the Using section.

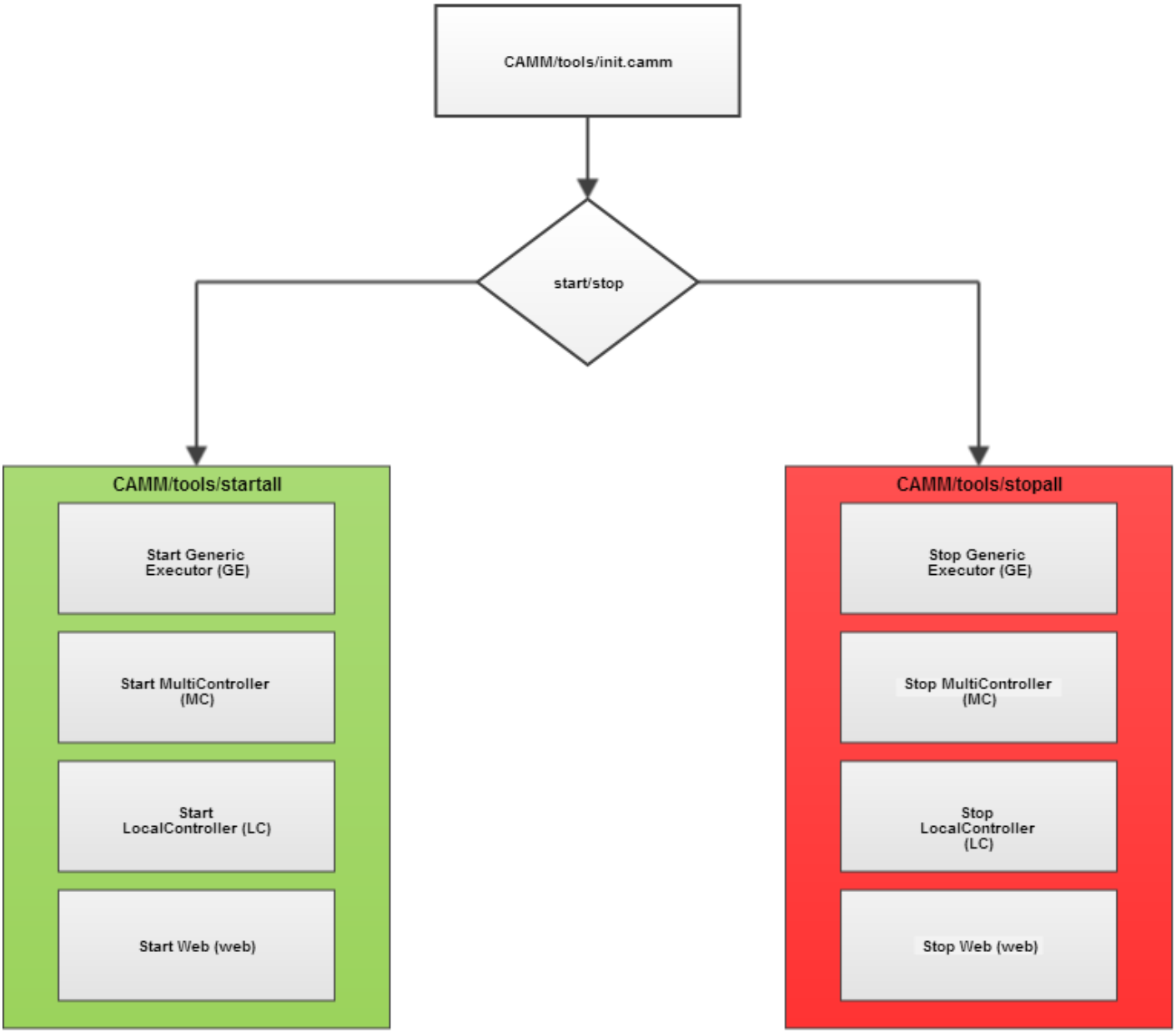
Starting and Stopping Services

This section describes how to start and stop services on the following operating systems:

During the DX NetOps installation, the Generic Executor and the Web Manager are registered as Services. The service names are CAMM-GE-{user}-{port}, and CAMM-tomcat7-8880. You can automatically start or stop DX NetOps at the system startup or shut down by executing specific commands as discussed in this section.

The following diagram illustrates the process flow of starting and stopping services:

Figure 87: Process flow diagram for starting and stopping services-XML



UNIX

You can execute the startall or stopall script or the init.camm script to start or stop DX NetOps. The init.camm script is in the Tools directory in the DX NetOps MM Home directory.

Execute the following init.camm.install script, as root or sudo su, to start or stop DX NetOps automatically at the system startup or shut down:

```
shell# tools/init.camm.install
```

Execute the following init.camm.uninstall script to remove the setting that automatically starts or stops DX NetOps:

```
shell# tools/init.camm.uninstall
```

Windows

Execute init.camm.install.bat to install the setting that automatically starts or stops DX NetOps:

```
C:/CAMM/tools/init.camm.install.bat
```

Execute init.camm.uninstall.bat to remove the setting that automatically starts or stops DX NetOps:

```
C:/CAMM/tools/init.camm.uninstall.bat
```

Install a Device Pack in DX NetOps Mediation Manager

Use this article to install a device pack in DX NetOps on the Windows or UNIX systems. This article contains the following topics:

A device pack is a set of XML configuration, ServingXml or Groovy, and XQuery files. Device packs pull data from a specific set of devices and convert it to a form that can be read by eHealth, CA Spectrum, CA Infrastructure Management, or Performance Management.

Install Engine and Presenter

You can install the Engine and Presenter through the DX NetOps Web UI.

Follow these steps:

1. Launch the DX NetOps Web Manager:
   ```
   http://<PrimaryMCMachineIP>:<web-port>
   ```
   Where <w*eb-port* > is the port number that is configured during the DX NetOps installation, and <*PrimaryMCMachine IP*> is the IP address or hostname of the primary MultiController system.
2. Enter the login credentials.

   NOTE
   Confirm that the working environment is CA eHealth.
3. Select the Device Packs tab and click Install**.**
4. Specify the destination of the LocalController using the LocalController drop-down.
5. Perform one of the following steps in the Device Pack configuration dialog:
   - If the device pack is located in the Device Pack repository that is shipped with the current release of DX NetOps, select the DevicePack Repository check box.
   - If the device pack is not located in the Device Pack repository and resides in the local system, click Browse and locate the device pack.
6. Select the ENGINE_<*devicepack*>.zip file or the PRESENTER_<*devicepack*>.zip file that you want to install.
7. Click Next.

8. Provide the configuration parameters in Global External Variables and Component Configuration for the Engine or Presenter being installed.

   NOTE
   For more information about specific device pack configuration files, see the corresponding Engine readme file by selecting ReadMe from the Web UI.
9. Select Finish.
   The device pack is installed.

Install CA eHealth Certification

To view the reports in CA eHealth, install eHealth certification. Installation of certification consists of the following two steps:

- Deploy Certification manually or automatically
- Manually import the elements to CA eHealth.

Deploy the eHealth Certification Automatically

Deploy the eHealth Certification automatically through DX NetOps Mediation Manager Web.

Follow these steps:

1. Launch the DX NetOps Mediation Manager Web UI:
   http://<*PrimaryMCMachineIP*>:<*web-port*>
   Where <*web-port* > is the port number configured during the DX NetOps Installation and <*PrimaryMCMachine IP*> is the IP address or hostname of the primary MultiController system.
2. Enter the login credentials.
3. Select the Device Packs tab and click Certification.
4. Specify the Certification destination from the Destination drop-down.
5. (Optional) Select OverWrite Existing cert check box.
6. Do one of the following steps:
   – If the Certification to install is in the eHealth Certification repository shipped with the current release of DX NetOps, select eHealth Cert.
   – If the Certification to install is in the local system, click Browse and locate the Certification.
7. Select the Certification and select Deploy.

Deploy the Certification Manually

Deploy the eHealth Certification manually.

Follow these steps:

1. Verify that you meet the following prerequisites:
   – The LocalController is installed on the CA eHealth server using the same user credentials that are used for installing eHealth.
   – The PRESENTER_<*devicepack*> component is installed on the LocalController that is running on the CA eHealth server.
   – $NH_HOME is set to the eHealth installation directory. To confirm, execute the following command:
   ```
   echo $NH_HOME
   ```
2. Copy the $CAMM_HOME/MC/eHealthCerts/CERT_<*devicepack*>.zip file to the $NH_HOME directory of your eHealth server.
3. Open a command prompt, type bash, and execute the following command:
   ```
   unzip CERT_<devicepack>.zip
   ```

4. Modify the $NH_HOME/modules/<*devicepack*>/camm.env script to provide the path details to JRE and the CAMM installation folder.
5. Create a soft link from $CAMM_HOME/output/PRESENTER_<*devicepack*> to $NH_HOME/modules/<*devicepack*>/ddiData.
6. Append the content of the $NH_HOME/db/data/variable.usr_<*devicepack*> file to $NH_HOME/db/data/variable.usr.
   - Before you combine the content, verify if there are any duplicate variables in both files. If you find duplicate variables, remove the duplicate variables and save the final $NH_HOME/db/data/variable.usr file.
   - Consider the final variables in the variable.usr file and map the variables.
   - Verify the mapping of the variable IDs in the variable.usr file and the elementTypeVariable.usr file.
   - Make sure that the variable IDs mapped in the elementTypeVariable.usr file are located in the variable.usr file.
7. Append the content of the $NH_HOME/db/data/elementTypeVariable.usr_<*devicepack*> file to $NH_HOME/db/data/elementTypeVariable.usr.

Import the Elements

Manually import the elements to CA eHealth.

Follow these steps:

1. Stop the CA eHealth server with the following command:
   ```
   nhServer stop
   ```
2. Convert the database schema for a software upgrade using the following command:
   ```
   nhConvertDb
   ```
   Wait until the database conversion executes successfully.
3. Start the eHealth server with the following command:
   ```
   nhServer start
   ```
4. Modify $NH_HOME/modules/<devicepack>/modules.defaults.init to suit your specific poll requirements.
5. Go to $NH_HOME/modules/camm<devicepackname> directory and execute the following command to discover your new elements:
   ```
   ./cammpoll  - c  - j -l
   ```
   eHealth automatically calls cammpoll to poll the elements.
   cammpoll is successful when you see the elements in eHealth OneClick in Manage Resources, Groups, <devicepack>.

   NOTE
   You only execute cammpoll manually once. Take note of any situation afterward in which you execute the cammpoll command manually, because it represents an error condition. In normal operations, eHealth executes the cammpoll command automatically.

Install Device Packs in DX NetOps Mediation Manager for Performance Management

A device pack for DX NetOps for Performance Management comprises of Engine, CAPM Presenter, Certification, and Views. Install these components to pull data from devices.

This article contains the following topics:

Prerequisites

Before you install device packs, confirm whether the following packages are installed:

- FTP or SFTP on the device server
- CA Performance Center
- Data Aggregator
- Data Collector

NOTE
For information about installing CA Performance Center and Data Aggregator, see the respective installation documents.

Install Device Packs in DX NetOps Mediation Manager for Performance Management 2.3.4 and later

You can install a device pack in DX NetOps for Performance Management 2.3.4 and later through the DX NetOps Web UI.

Follow these steps:

1. Launch the DX NetOps Web UI:
   http://<*PrimaryMCMachineIP*>:<*web-port*>
   Where <*web-port*> is the port number that is configured during the DX NetOps Installation and <*PrimaryMCMachine IP*> is the IP address or hostname of the primary MultiController system.
2. Type the login credentials.

   NOTE

   Confirm that the working environment is CA Performance Management.
3. Click the Device Packs node and select Install.
4. The Device Pack Selection dialog appears.
5. Perform one of the following steps:
   – If the device pack to install is in the Device Pack repository that is shipped with the current release of DX NetOps Mediation Manager, select Device Pack Repository.
   – If the device pack to install is in the local system, click Browse and locate the device pack.
6. Select a device pack and click Next.
   Cert and Views of the selected device pack are deployed and the dialog to configure CAPM Presenter appears.
7. Specify the LocalController destination from the drop-down.
8. Provide the parameters in Global External Variables and Component Configuration.

   WARNING

   - The PRESENTER_ID of the Engine must match with the COMPONENT_ID of the CAPM Presenter.
   - To update the text in COMPONENT ID, append text or numeral with underscore. Do not modify the existing text in COMPONENT ID.

   NOTE

   If you already installed a CAPM Presenter, you can skip the CAPM Presenter installation by clearing the check box at the top of the dialog. For more information about specific Device Pack configuration files, see the corresponding Engine readme file by selecting ReadMe from the Web UI.

   The polling time is defined in the fields, such as PERFORMANCE_POLL_RATE and INVENTORY_POLL_RATE.
   Format: Crontab format in which the time is specified in the seconds-minutes-hour order.
   Examples:

   */30 * * specifies that polling occurs every 30 seconds.
   * 5 * specifies that polling occurs at the fifth minute after an hour.
   0 0 2,14 specifies that polling occurs at 2 A.M and 2 P.M.
   - - - specifies that no polling is scheduled.

9. Provide the following parameters for tmp, logbase, and local in Cleanup Configuration to clean the files in the tmp, logbase, and local directories and click Next:
   expire
   Specifies the time after which the file is deleted.
   Format: <*number*><s/m/h/d/y>
   Where, s = second, m = month, h = hour, d = number of days, y = number of years
   Examples: 6s, 4m, 2h, 10d, or 1y
   match
   Specifies the matching string to be searched.
   Example: *.file
   achivePrefixSpecifies the prefix for the archived file.
   achiveSuffix
   Specifies the suffix for the archived file.
10. Provide the parameters for Engine and click Finish.
    CAPM Presenter and Engine are deployed.

Back Up and Restore

Back Up and Restore DX NetOps MM

When you migrate DX NetOps MM, ensure that you back up the following DX NetOps MM files and restore them:

- /opt/CA/CAMM/MC/repository
- /opt/CA/CAMM/COMPONENTS/ENGINE_<DevicePackName>/tmp/input/*

Back Up and Restore the Device Pack

Before you reinstall DX NetOps Mediation Manager or redeploy the MultiControllers, back up the device packs by executing the cammBackup script. On executing the backup script, all files in the <*Install_folder*>/MC/repository folder are backed up. By executing the cammRestore script, you can restore the device packs that are backed up.

WARNING

- The scripts work only on Linux and Solaris systems.
- Stop all device packs before executing the scripts.

Back up the Device Pack

Execute the following command in the system where MultiController is installed:

```
./cammBackup /<Destination_Folder>
```

Example: ./cammBackup /MC/repository

Restore the Backed up Device Packs

Execute the following command:

```
./cammRestore /<location and Name of the backedup files>
```

Example: ./cammRestore /tmp/cammBackup_2015_02_10_03_59_32.tar.gz

If you restore the device packs to a system other than from where it was backed up, ensure to rename the MultiController or LocalController. So that they both are synchronized and work correctly.

Upgrading

This section provides steps to upgrade DX NetOps, upgrade a device in DX NetOps, and upgrade device packs in DX NetOps for Performance Management.

Upgrade DX NetOps Mediation Manager

Upgrade DX NetOps when the latest release is available. Also, when you upgrade CA Performance Management, check the compatibility matrix in System Requirements and, if needed, upgrade DX NetOps Mediation Manager. If MultiController and LocalController are installed in discrete systems, upgrade the components in the following order:

- Primary MultiController
- Secondary MultiController
- LocalController

Follow these steps:

1. Start the DX NetOps installation with the respective command based on your operating system:
   - UNIX systems:
     ```
     # $JAVA_HOME/bin/java  -jar CAMM-Installer-20.2.x.jar
     ```
   - Windows systems:
     a. Stop CAMM Components
        ```
        %CAMM_HOME%/tools/stopall.bat
        ```
     b. Insert the installation media and open Windows Explorer.
     c. Locate and double-click the executable JAR file, CAMM-Installer-20.2.x.jar.
2. Click Next on the Welcome dialog.
3. Read the Important Information and click Next.
4. Review the licensing agreement and click Next.
5. Specify a target path to install DX NetOps. Alternatively, click Choose to browse for a location.

   NOTE

   When installing on a Windows System, the default installation directory is C:\Program Files\CA\CAMM.

   When installing on a Unix System (Linux or Solaris), the default installation directory is /opt/CA/CAMM.
6. Select Upgrade and click Next.
7. Review the upgrade summary and click Install to start the upgrade process.
8. Click Done.

   NOTE
   DX NetOps starts before the installer exits.

   The installer stops DX NetOps and starts the upgrade process. All existing files are overwritten.
   Files in the following folders are preserved:
   - $CAMM_INSTALL_DIR/MC/repository
   - $CAMM_INSTALL_DIR/Queue
   - $CAMM_INSTALL_DIR/output
   - $CAMM_INSTALL_DIR/COMPONENTS

     NOTE
     We recommend that you back up any custom scripts in the CAMM_INSTALL_DIR/tools folder.

   DX NetOps is upgraded.
9. Navigate to $CAMM_HOME and review the upgraded installer information in the version.xml file. The content format in the version.xml file is similar to the following format:

```
<?xml version="1.0" ?>
<CAMM-Version>
   <Current>
     <Version><20.2.x>/Version>
     <Revision><Installer revision number></Revision>
   </Current>
   <History>
      <Build>
         <Version>3.6.0 or 3.7.0</Version>
         <Revision>Installer revision number</Revision>
      </Build>
  </History>
</CAMM-Version>
```

Upgrade a Device Pack in DX NetOps Mediation Manager

You can upgrade a device pack by using the DX NetOps Web UI.

Follow these steps:

1. Launch the DX NetOps Web Manager:
   `http://<PrimaryMCMachineIP>:<web-port>`
   Where <w*eb-port* > is the port number that is configured during the DX NetOps installation, and <*PrimaryMCMachine IP*> is the IP address or hostname of the primary MultiController system.
2. Enter the login credentials.

   NOTE
   Confirm that the working environment is CA eHealth.
3. Click the Device Packs tab, select the subcomponent (Engine or Presenter) that you want to upgrade, and click Upgrade**.**
4. Perform one of the following steps:
   – If the Device Pack you want to upgrade is located in the Device Pack repository that is shipped with the current release of DX NetOps, select the Device Repository checkbox.
   – If the Device Pack you want to upgrade is not located in the Device Pack repository and resides in the local system, click Browse and locate the Device Pack.
5. From the LocalController drop-down, select the LocalController where you want to upgrade the Device Pack.
6. Select the ENGINE_<*devicepack*>.zip file that you want to upgrade and click Next.
7. Provide the configuration parameters in Global External Variables and Component Configuration for the Engine or Presenter being upgraded.

   NOTE
   For more information about specific device pack configuration files, see the corresponding engine readme file by selecting Readme from the Web UI.
8. Select Finish.
   The device pack is upgraded.

Upgrade Device Packs in DX NetOps Mediation Manager for Performance Management

A device pack consists of Engine, Certification, and Views. Upgrade a device pack in DX NetOps for CA Performance Management through the DX NetOps Mediation Manager Web UI.

Upgrade using the Typical Option

Use the Typical option to upgrade Engine and the corresponding CERT and Views seamlessly.

Follow these steps:

1. Launch the DX NetOps Web Manager:
   `http://<PrimaryMCMachineIP>:<web-port>`
   Where <w*eb-port* > is the port number that is configured during the DX NetOps installation, and <*PrimaryMCMachine IP*> is the IP address or hostname of the primary MultiController system.
2. Enter the login credentials.

   NOTE
   Confirm that the working environment is CA Performance Management.
3. Click the Device Packs tab, select the Engine that you want to upgrade, and click Upgrade, Typical**.**
4. Select the Device Pack Repository check box, select a device pack, and click Next.

   NOTE

   Do no click the Browse button. If you click Browse, you have to manually upgrade the individual components.
5. Review the configuration parameters in Global External Variables and Component Configuration for the Presenter and Engine being upgraded and update the new parameters.

   WARNING

   The PRESENTER_ID of the Engine must match with the COMPONENT_ID of the CAPM Presenter.

   NOTE
   In Global External Variables, the previously modified values for the existing parameters are preserved during the upgrade. So, modify the values for the existing parameters, only if needed. However, the new parameters are populated with the default values. So, modify the new values according to your requirement. To know the new parameters, click ReadMe in the Web UI; the list of device pack configuration files that consists of the parameters appear. Select DPInfo_<*devicepack*>.txt and click Open.
6. Click Finish.
   The device pack is upgraded.

Upgrade using the Custom Option

Use the Custom option to upgrade, Engine, Presenter, CERT, and Views individually. This method is most suited for the Presenter upgrade.

Follow these steps:

1. Launch the DX NetOps Web Manager:
   `http://<PrimaryMCMachineIP>:<web-port>`
   Where <w*eb-port* > is the port number that is configured during the DX NetOps installation, and <*PrimaryMCMachine IP*> is the IP address or hostname of the primary MultiController system.
2. Enter the login credentials.

   NOTE
   Confirm that the working environment is CA Performance Management.
3. Click the Device Packs tab, select the Engine or Presenter that you want to upgrade, and click Upgrade, Custom**.**
4. Perform one of the following steps:
   - If the Device Pack you want to upgrade is located in the Device Pack repository that is shipped with the current release of DX NetOps, select the Device Pack Upgrade check box and select a device pack.
   - If the Device Pack you want to upgrade is not located in the Device Pack repository and resides in the local system, click Browse, locate the device pack.

5. Click Engine, CERT, or View radio button and select the corresponding subcomponent, and click Next.
6. (Only for Presenter and Engine) Review the configuration parameters in Global External Variables and Component Configuration for the subcomponent being upgraded and update the new parameters.

   WARNING

   The PRESENTER_ID of the Engine must match with the COMPONENT_ID of the CAPM Presenter.

   NOTE
   In Global External Variables, the previously modified values for the existing parameters are preserved during the upgrade. So, modify the values for the existing parameters, only if needed. However, the new parameters are populated with the default values. So, modify the new values according to your requirement. To know the new parameters, click Help, DevicePacl ReadMe in the Web UI; the list of device pack configuration files that consists of the parameters appear. Select DPInfo_<*devicepack*>.txt and click Open.
7. Click Finish.
   The subcomponent is upgraded.

Migrating Device Pack

This section contains steps to migrate a device pack and self-monitoring device pack from one version of CA Performance Management to another. To move the Engine from one LocalController to another, see the Moving Device Pack Engine between LocalControllers section.

Limitation:
On account of limitations with the CA Performance Management, you cannot migrate a Presenter from a LocalController in a Data Collector to another Local Controller in another Data Collector

Device Pack Migration

This section provides you the steps to migrate the device pack in DX NetOps Mediation Manager for Performance Management.

The following table provides the installer type available for DX NetOps Mediation Manager:

	CA Infrastructure Management 2.3.3 or older	**CA Performance Management 2.3.4**	CA Performance Management 2.4 and 2.5
CA Mediation Manager 2.2.5 or older	ems-installer-<*version*>.zip	Not Compatible	Not Compatible
CA Mediation Manager 2.2.6	ems-installer-<*version*>.zip	CAMM-Installer-<*version*>.jar	Not Compatible
CA Mediation Manager 2.4 and 2.5	ems-installer-<*version*>.zip	CAMM-Installer-<*version*>.jar	CAMM-Installer-<*version*>.jar

When you upgrade DX NetOps Mediation Manager without upgrading CA Infrastructure Management or CA Performance Management releases, there is no need for migration of device packs. [Schematically, you move vertically down in the table.]

When you upgrade CA Performance Manager with or without upgrading DX NetOps Mediation Manager, you need to migrate the device pack. [Schematically, you move horizontally right-side in the table.]

The two possible scenarios and the device pack migration process are follows:

Scenario1: Upgrade CA Infrastructure Management r2.3.3 to CA Performance Management r2.5
Perform the following steps:

1. Install DX NetOps Mediation Manager 20.2.

2. Execute the migratePMtoCAMM script to migrate the engine. Execute the script in the machine where Data Collector is installed. The script is available at <*CAMM_Install_Folder*>/tools. For the step-by-step instructions, see the Migrate Device Pack Generated by the Device Pack Generator or Migrate Device Pack Existing in the Repository section.
3. Stop the Data Aggregator.
4. Execute the camm-tools-cert-migration-<*version*>-SNAPSHOT-jar-with-dependencies.jar script to migrate the certifications. The script is packaged with the Data Aggregator file (CA_DA_<version>_Linux.tar.gz). To execute the script, unzip CA_DA_<version>_Linux.tar.gz, which you download from the CA Performance Management page in CA Support Site. For more information, see CA Performance Management documentation.
5. Upgrade CA Infrastructure Management r2.3.3 to CA Performance Management r2.5. Refer to CA Performance Management 2.5 installation document.
6. Start the migrated device pack from DX NetOps Mediation Manger Web UI.

Scenario2: Upgrade CA Performance Management r2.3.4 to CA Performance Management r2.5
Perform the following step:

1. Stop the Data Aggregator.
2. Execute the camm-tools-cert-migration-<*version*>-SNAPSHOT-jar-with-dependencies.jar script to migrate the certifications. The script is packaged with the Data Aggregator file (CA_DA_<*version*>_Linux.tar.gz). To execute the script, unzip CA_DA_<version>_Linux.tar.gz, which you download from the CA Performance Management page in CA Support Site. For more information, see CA Performance Management documentation.
3. Upgrade CA Performance Management r2.3.4 to CA Performance Management r2.5. Refer to CA Performance Management 2.5 installation document.
4. Upgrade DX NetOps Mediation Manager to release 20.2.

> **NOTE**
>
> To migrate the self-monitoring device pack, follow the instructions in the Self-Monitoring Device Pack Migration section.

Verify Prerequisites

Before you migrate the device packs, confirm that the following criteria are met:

1. The installed version of CA Infrastructure Management is Release 2.3.3 or earlier.
2. The CA Performance Management 2.3.4 or later installation file is available.
3. DX NetOps Mediation Manager Release 20.2 is installed. A MultiController is installed on a dedicated server within the CA Infrastructure Management environment.
4. A LocalController is installed in every Data Collector that has a device pack installed.
5. Perform the following steps to check if IP address of the system in which the LocalController is installed appears in the runtime.xml file:
   a. Open the runtime.xml file from $CAMM_INSTALL/LC.
   b. Check if IP address is mentioned against the “myAddress” attribute.
   c. If IP address is not mentioned, replace the hostname with IP address and save the file.

Migrate Device Pack Generated by the Device Pack Generator

Migrate the device packs that are generated by the device pack generator by using the migratePMtoCAMM script.

Follow these steps:

1. Log in to the host where Data Collector is installed.
2. Navigate to $CAMM_INSTALL_DIR/tools.
3. Execute the migratePMtoCAMM migration script.
   The options to execute the migration script appear.

4. Execute the migratePMtoCAMM migration script with the option “d”
5. Type the device pack to migrate and press Enter.
   The device pack migrates successfully.

   NOTE

   The successfully migrated device pack is listed in Web UI. If the device pack is not available in Web UI or any error message is in the migration script output, please contact CA Support.
6. Upgrade the current version of CA Infrastructure Management to CA Performance Management 2.3.4 or later.
7. In CAMM Web UI, Click Device Packs and the Advanced option of the migrated device pack.
8. Select Repository, Control, and open the <*DevicePackName*>-Inventory.xml file in deviceConfig.
9. Confirm that “<Type>com.torokina.tim.plugin.inventory.InventoryListUpdateFunction</Type>“ is a single statement in <Acquisition desc="Inventory Update"> tag
10. Save <*DevicePackName*>-Inventory.xml file.
11. Start the device pack Engine followed by CAPM Presenter from the DX NetOps Mediation Manager Web UI.

    WARNING

    If the device pack did not start, contact CA Support.

Migrate Device Pack Existing in the Repository

Execute the migratePMtoCAMM script to migrate the device packs. Execute the script on every Data Collector where a device pack is installed.

Follow these steps:

1. Log in to the host where Data Collector is installed.
2. Navigate to $CAMM_INSTALL_DIR/tools.
3. Execute the migratePMtoCAMM migration script.
   The options to execute the migration script appear.
4. Execute the migratePMtoCAMM migration script with one of the following options: a, d, or t

   NOTE

   To migrate the CANECeNodeB, CACRBT, or CACRBTUSER device pack, execute the script with the option d.
5. Specify the device pack to migrate.
6. (For Test mode) Do one of the following steps:
   - To install the device pack Engine in the system where migratePMtoCAMM was executed, do the following step:
     - Copy the Engine_<*devicepack*>.zip file from MigratedIMDevicepacks directory and manually deploy in the system where the corresponding LocalController is running.
   - To install the device pack Engine in a system where migratePMtoCAMM was not executed, do the following steps:
     - Copy the files in the <*CAMM_Install dir*>/Components directory in the system where migratePMtoCAMM was executed.
     - Log in to the other system where you want to install the device pack Engine.
     - Create the Engine_<*devicepack*> directory in the <*CAMM_Install dir*>/Components directory.
     - Paste the copied files in the Engine_<*devicepack*> directory.
7. Upgrade the current version of CA Infrastructure Management to CA Performance Management 2.3.4 or later.
8. (For Test mode) Follow one of the steps to install CAPM Presenter. But, do not start the CAPM Presenter.
   - Copy the Presenter zip file from MigratedIMDevicepacks directory and install it in the LocalController that is on Data Collector.
   - Install CAPM Presenter from the DX NetOps Web UI.

WARNING

The PRESENTER_ID of the Engine must match with the COMPONENT_ID of the CAPM Presenter.

The number of CAPM Presenters depends on the number of complex device packs. If the amount of data to be processed is very large, the performance of the single CAPM Presenter is impacted. For the workaround, see the Frequently Asked Questions section.

9. (For Test and Deploy modes) Start the device pack Engine followed by CAPM Presenter from the DX NetOps Mediation Manager Web UI.

WARNING

If the device pack did not start, contact CA Support.

Self-Monitoring Device Pack Migration

You should migrate the self-monitoring device pack through the Web UI if you migrate the device pack from CA Infrastructure Management 2.3.3 and earlier to CA Performance Management 2.3.4 and above as specified in the Device Pack Migration section.

Follow these steps:

1. Launch the DX NetOps Mediation Manager web UI:
   http://<*PrimaryMCMachineIP*>:<*web-port*>
   Where <*web-port*> is the port number that is configured during the DX NetOps Mediation Manager Installation and <*PrimaryMCMachine IP*> is the IP address or hostname of the primary MultiController system.
2. Enter your login credentials.
3. Make sure that you are in the CA Performance Management environment.
4. Select Device Packs from the DX NetOps MM Cluster node in the Dashboard.
5. Select Install.
   The Device Pack Installation dialog appears.
6. Specify the destination from the LocalController drop-down.
7. Perform one of the following steps:
   – If the device pack to install is in the Device Pack repository that is shipped with the current release of DX NetOps Mediation Manager, select Device Pack Repository.
   – If the device pack to install is in the local system, click Browse and locate the device pack.
8. Select ENGINE_ CAMM_IM_SelfMonitoring.zip and click Next.
   Cert and Views of the selected device pack are deployed and the dialog to configure CAPM Presenter appears.
9. Provide the parameters in Global External Variables and Component Configuration and click Next.

WARNING

- The PRESENTER_ID of the Engine must match with the COMPONENT_ID of the CAPM Presenter.
- To update the text in COMPONENT ID, append text or numeral with underscore. Do not modify the existing text in COMPONENT ID.

NOTE

If you already installed a CAPM Presenter, you can skip the CAPM Presenter installation by clearing the check box at the top of the dialog. For more information about specific Device Pack configuration files, see the corresponding Engine readme file by selecting ReadMe from the Web UI.

   CAPM Presenter and Engine are deployed.
10. Click Finish when the installation is complete.
    The self-monitoring device pack is migrated.

11. Start CAMM_IM_SelfMonitoring from the Web UI.

Using

Use the Web Manager to select an environment that is integrated with DX NetOps and to manage the components and device packs deployed. Check the browser requirement in the Release Notes section to know the browser compatibility.

How to access and use DX NetOps?

- To start the application, type http://<*PrimaryMCMachineIP*>:<*web-port*> in a browser.Where <*PrimaryMCMachine IP*> is the IP address or hostname of the primary MultiController system, <*web-port*> is the port number that is configured during the DX NetOps Mediation Manager installation. **Default:** 8080.
- To log in the application, type the login credentials and click LOG IN.
  User ID: admin; **Password:** camm
- To manage the components or device packs or to modify the settings, click the corresponding tabs.
- To log out the application, click Log Out at the top right-corner.

Select Environment, Language, and Help

Contents

Environment Selection

The data that are collected and processed by DX NetOps is rendered in another environment. CA eHealth and CA Performance Management are the two environments that can be integrated with DX NetOps. The options in the Web interface dependDX NetOps on the environment selected.

You are prompted to select the environment when you log in to the web interface for the first time. However, you can change or update the environment anytime.

Selecting the Environment

After you install DX NetOps Mediation Manager, perform the following steps to select an environment.

1. Launch DX NetOps Mediation Manager and click the environment name that appears at the top-right corner.
2. Select the environment.
3. (for CA Performance Management) Specify the CA Performance Center details.
4. Click Save.

Changing or Updating the Environment

The name of the current environment appears at the top-right corner of the Management tab. To change the existing environment in which DX NetOps is installed or to update the parameters of the environment, click the environment name and specify the parameters.

> **NOTE**
>
> When no environment is selected, Change/Update Environment appears on the top-right corner.

Language Selection

DX NetOps supports the following languages:

- Chinese Simplified
- Chinese Traditional
- English
- French
- Japanese

Changing the Language

The name of the current language appears at the top-right corner of the Web UI. To change the existing language in which DX NetOps is installed, click the language and select a new language from the dialog.

Open API

The API link redirects to the page that has instructions to perform the following read-only operations using REST over HTTP:

- GET the installed and supported device packs
- GET the discovered devices
- GET the last performance poll data

Help Selection

DX NetOps provides link to access the device pack information file [DPInfo file] and also three channels to resolve their issues - Wiki, Support, and Communities.

Follow these steps:

1. Click Help at the top-right corner.
2. Select one of the following options:
   a. Click DX NetOps MM Wiki to access DX NetOps documentation in Wiki.
   b. Click Device Pack ReadMe to access the DPInfo file.
   c. Click About DX NetOps MM to access CA Support site and DX NetOps user communities.

Manage Controllers

MultiController (MC) and LocalController (LC) are the two controllers that manage the environment configuration. Use this page to start, monitor, or stop the controllers.

The following information describes the options in the Controllers tab of the DX NetOps Mediation Manager Web UI:

Start or Stop Starts or stops a LocalController or MultiController. To start or stop a controller, select the corresponding check box of the controller, and click Start or Stop.

RefreshRefreshes the page and not individual controllers.

GroupGroups the controllers that are based on the selected criteria. To group the controllers, click Group and select a criteria from the drop-down.

LoggerDisplays the logs. To view the log details of a controller, click the

(Logging) icon that is available in the corresponding row.

(Advanced)
Provides performance metrics for the Java Virtual Machine (JVM) and the system where the controller is running.

SubscribeRecords the log. This button appears in the Logging area when you click

for a particular controller.

ClearClears the log. This button appears in the Logging area when you click

for a particular controller.

FocusPins the logging area. This button appears in the Logging area when you click

for a particular controller.

OptionsProvides options to change the settings that are done in My Settings page for the Logging section.

Manage Device Packs

Device Pack collect information from a non-SNMP device and forward it to the Network Management System. Use this page to install, start, monitor, upgrade, manage, stop, or remove a device pack in your system.

The following information describes the options in the Device Packs tab of the DX NetOps Mediation Manager Web UI:

Install or Remove

Installs or removes the device packs from the existing repository. The default path that is used during the installation is $CAMM_HOME/MC/repository/device packs. If you used a different path during the installation, browse and select your device pack path.

To install a subcomponent, click Install, and specify the options in the Device Pack Configuration dialog.

To remove a subcomponent, click the corresponding check box of the subcomponent and click Remove.

Upgrade

Upgrades the version of the device pack.

To upgrade a subcomponent, click the corresponding check box of the subcomponent and click Upgrade.

Move

Moves the Engine from one LocalController to another.

Refresh

Reloads the page. It does not refresh the individual subcomponent.

GroupGroups the controllers that are based on the selected criteria. To group the controllers, click Group and select a criteria from the drop-down.

LoggerDisplays the logs. To view the log details of a controller, click the

(Logging) icon that is available in the corresponding row.


(Advanced)
Provides performance metrics for the Java Virtual Machine (JVM) and the system where the controller is running.

SubscribeRecords the log. This button appears in the Logging area when you click

for a particular controller.

ClearClears the log. This button appears in the Logging area when you click

for a particular controller.

FocusPins the logging area. This button appears in the Logging area when you click

for a particular controller.

OptionsProvides options to change the settings that are done in My Settings page for the Logging section.

Move Device Pack Engine

Use the Move feature to move a device pack Engine from one LocalController to another.

Follow these steps:

1. Select one or multiple Engines and click Stop.
2. Click Move, specify the LocalController as new destination, and click Move.
3. Do one of the following steps:
   - In the destination LocalController, if you want to prevent the download and process of the already polled data, continue from Step 4.
   - In the destination LocalController, if downloading and processing the already polled data is acceptable, continue from Step 7.
4. Browse to the following location in the destination LocalController: /opt/CA/CAMM/COMPONENTS/ENGINE_<*devicepack name*>
5. Create the tmp, input, performance, and inventory folders in the following order:
   /opt/CA/CAMM/COMPONENTS/ENGINE_<d*evicepack name*>/tmp/input/performance
   /opt/CA/CAMM/COMPONENTS/ENGINE_<*devicepack name*>/tmp/input/inventory
6. Cut the .historyFile.inventory from the following locations in the source LocalController and paste them in the corresponding folders [created in step 5] in the destination LocalControllers:
   /opt/CA/CAMM/COMPONENTS/ENGINE_<*devicepack name*>/tmp/input/performance
   /opt/CA/CAMM/COMPONENTS/ENGINE_<*devicepack name*>/tmp/input/inventory

   WARNING

   Perform steps 4 through 6 before the next poll. After the poll, the tmp, input, performance, and inventory folders are created automatically and the previous history files are downloaded causing unwanted memory usage.
7. Select the moved Engine in the Manage Device Packs tab and click Start.

Manage Settings

Launch the Web UI, click My Settings, and use the options to specify the settings to establish connection, collect logs, and send notification messages.

Connection

Use the following parameters to specify the settings to establish connection between MultiController and device:

RetriesSpecifies the number of times MultiController attempts to connect the server.

IntervalSpecifies the time lag between the two attempts.

Logging

Use the following parameters to specify the settings for recording the logs:

Refresh Rate Specifies how frequent the log page refreshes.

Buffer Size Specifies the character size of the log page.

Max Line Number Specifies the maximum line number to be accommodated in log.

Syntax Highlight Indicates if a syntax that appears in the log to be highlighted in different color.

Notification

Use the settings in the Notification area to continuously monitor the components and receive alerts when an anomaly is detected. For every component, you can select the anomaly for which you need notification. When an anomaly occurs, you will receive an email that contains the possible cause and solution.

Use the following parameters to receive notification if anomalies are detected in the DX NetOps Mediation Manager components:

Server Properties

Contains the following details of the server.

SMTP Server Address
Specifies the hostname or IP address of the SMTP server.

SMTP Server Port
Specifies the port to use for SMTP requests.

Email To
Specifies the primary email addresses to which the generated alert messages are sent. Separate multiple addresses by comma.
Format: user1@mydomain.com, user2@mydomain.com

Email CC
(Optional) Specifies the secondary email addresses to which the generated alert messages are sent. Separate multiple addresses by comma.
Format: username1@mydomain.com, username2@mydomain.com

Email From
(Optional) Specifies the email address from which the alert messages are sent to the addresses listed in Email To and Email CC. If no address is specified, email will be sent from the CAMM_Notification@ca.com address.
Format: sendername@mydomain.com

Enable Authentication
(Optional) Select the check box if the email server provided needs any authentication.

Username
Specifies a username to use when the email server challenges an SMTP request.

Password
Specifies a password to use when the email server challenges an SMTP request. The SMTP Username parameter is required.

Test
Tests the server details by sending an autogenerated email.

> **WARNING**
> Proceed further only if the test is successful.

DX NetOps MM Components

Lists the DX NetOps Mediation Manager components. Expand a node and click a component. From the <*component name*> Anomalies dialog, select the following anomalies:

MC stopped Notifies when the MultiController stopped working.

Error file created Notifies when an error file is created. The file contains “STD-OUT” or “STD-ERROR”. This file contains error messages, such as any third-party-library errors, that cannot be logged in regular log file.

Exception or error message is logged Notifies about an event that affects the natural functioning of the component and error message is logged.

Engine stopped Notifies when the Engine stopped working.

LC stopped Notifies when the LocalController stopped working.

Presenter stopped Notifies when the Presenter stopped working.

DX NetOps MM DC stopped Notifies when the Data Controller stopped working.

DX NetOps MM DC Queue size growing Notifies when the threshold limit of the Data Collector file collection is exceeded.

Programming

DX NetOps data can be classified as the data that are collected from the non-SNMP devices and the data that are processed by the DX NetOps components. These data are stored in corresponding XML files. Though you can manually access the files from the system where DX NetOps is installed, it is not recommended. So, the files are accessed through the Web-based UI to know the product status or to perform various operations.

API using REST over HTTP can be used only for DX NetOps for CA Performance Management. This feature helps in sharing the data among applications and easily integrating with Network Management System [NMS].

You can use API for the following purposes:

- Read the list of all supported device packs
- Read the default configuration of device pack components
- Read the list of all installed components
- Read the data from installed components
- Installing components
- Upgrading components
- Start or Stop the components

Access API using REST over HTTP

The URL and the other information are as follows:

API Hyperlink

- If you are logged in DX NetOps, click the API hyperlink on the top-right corner.
- Follow the instructions in the home page.

API Usage

In DX NetOps 2.5, only the GET method is supported for OpenAPI. The services that are provided through the GET method can also be accessed through a browser.

Follow these steps:

1. Open your web browser and navigate to the following address: http://<*base URL*>/<*relative URL*>
   base URL: http://<*CAMM_HOST*>:<*WebServerPort*>/tim-web/rest
   relative url: See the Supported Services for the list of relative URLs.

2. Type the login credentials, which is same as that of DX NetOps Web UI.

   NOTE
   Alternatively, to communicate with the DX NetOps API, create the REST client application or use any tool that contains the REST client functionality.

Supported Services

The supported services and the corresponding relative URLs are as follows:

Service Description	Relative URL
Get installed device packs	/devicepacks
Get supported device packs	/devicepacks/all
Get all discovered devices by one device pack	/devicepacks/<*id*>/devices
Get performance poll data of one device packs within this range	/devicepacks/<*id*>/data/performance

Example: http://<*PrimaryMCHostName*>:<*WebServerPort*>/tim-web/rest/devicepacks

Read the List of all Supported Device Packs

You can get the list of all the supported device packs in the CA Performance Management environment. This section provides you sample request URL and the corresponding response.

Scenario:

- Primary MultiController is installed at camm.ca.com with the default web server port number (8880).
- Two LocalControllers are connected to the primary MultiController.

HTTP Method: GET

Request URL

Format: http://<*CAMM_HOST*>:<CAMM_WEB_PORT>/tim-web/rest/devicepacks

For the given scenario: http://camm.ca.com:8880/tim-web/rest/devicepacks

Sample response

```
<response url="/devicepacks" version="2.x.y.z">

  <devicepacks>

      <devicepack id="CAMM_IM_SelfMonitoring"/>

      <devicepack id="CARuckus"/>

      <devicepack id="CAVISP_VSEP"/>

      <devicepack id="CACiscoStarentGGSN"/>

      <devicepack id="CAZTESSS"/>

      <devicepack id="CAVmamsa"/>
```

```
<devicepack id="CAAcmePacketSBC"/>

<devicepack id="CAAlcatelPresence"/>

<devicepack id="CACRBTUSER"/>

<devicepack id="CAStarentSGSN"/>

<devicepack id="CADatacollector"/>

<devicepack id="CAALU7750HRU"/>

<devicepack id="CACiscoASR5000EDR"/>

<devicepack id="CAAlcatel5620Sam_v11R5"/>

<devicepack id="CANortelSSL"/>

<devicepack id="CACitrixAR"/>

<devicepack id="CANECeNodeB"/>

<devicepack id="CAAlcatelEPC"/>

<devicepack id="CAJuniperERX"/>

<devicepack id="StarentEMS_All"/>

<devicepack id="CAZTEMSC"/>

<devicepack id="CANSNHLR"/>

<devicepack id="CAFlashNetworks"/>

<devicepack id="CAAlcatel5620Sam"/>

<devicepack id="CAEricssonSGW"/>

<devicepack id="CACiscoASR5000"/>

<devicepack id="CANetAppFiler"/>

<devicepack id="CANortelBCPOM"/>

<devicepack id="CACRBT"/>

<devicepack id="CASIAENMS5UX"/>

<devicepack id="CAMNS"/>
```

```
<devicepack id="CANokiaIMS"/>

<devicepack id="CAZTEAGCF"/>

<devicepack id="CAAlcatelXMSIMS"/>

<devicepack id="CACiscoCallManager"/>

<devicepack id="CAComverseInsight"/>

<devicepack id="CATekelekDSR"/>

<devicepack id="CAMetaswitchCluster"/>

<devicepack id="CAEricssonSGSN"/>

<devicepack id="CAALUDSRSDM"/>

<devicepack id="CANSNLBS"/>

<devicepack id="CAALUCP"/>

<devicepack id="CAEricssonMME"/>

<devicepack id="CAEXFOBrixWorx"/>

<devicepack id="CAHuaweiU2000PTN"/>

<devicepack id="CANortelMS2000"/>

<devicepack id="CAZTEMGW"/>

<devicepack id="CAGuavus"/>

<devicepack id="CATekalecPCRF"/>

<devicepack id="CAOpenetPCRF"/>

<devicepack id="CAZTESBC"/>

<devicepack id="StarentEMS"/>

<devicepack id="CANortelUSP"/>

<devicepack id="CAInsight"/>

<devicepack id="CACamiantMPE"/>
```

```
        <devicepack id="CAEricssonSoO"/>

        <devicepack id="CAAlcateleNodeB"/>

        <devicepack id="CAPEStats"/>

        <devicepack id="CAGenBandGateway"/>

        <devicepack id="CAServiceOverMicroWave"/>

        <devicepack id="CAHuaweiU2000WDM"/>

        <devicepack id="CAHuaweiCloud"/>

        <devicepack id="CAALUSA"/>

        <devicepack id="CAZTECSCF"/>

        <devicepack id="CANortelSST"/>

        <devicepack id="CANortelBCPPM"/>

        <devicepack id="CANortelGWC"/>

    </devicepacks>

    <status>

        <code>200</code>

        <message>Success!</message>

    </status>

 </response>
```

Read the List of all Installed Components

You can get the list of installed components in the CA Performance Management environment. This section provides you sample request URL and the corresponding response.

HTTP Method: GET

Scenario:

- Primary MultiController is installed at camm.ca.com with the default web server port number (8880).
- Two LocalControllers are connected to the primary MultiController.
- PRESENTER_CAPM, ENGINE_CAMM_IM_SelfMonitoring, and ENGINE_CAGuavus are installed.

Request URL

Format: http://<*CAMM_HOST*>:<*WebServerPort*>/tim-web/rest/components

For the given scenario: http://camm.ca.com:8880/tim-web/rest/components

Sample response

```
<response url="/components" version="2.x.y.z">

    <devicepacks>

        <devicepack id="PRESENTER_CAPM"/>

        <devicepack id="ENGINE_CAMM_IM_SelfMonitoring"/>

        <devicepack id="ENGINE_CAGuavus"/>

    </devicepacks>

    <status>

        <code>200</code>

        <message>Success!</message>

    </status>

</response>
```

GET the Last Performance Poll Data

You can get the details of all the last performance poll data of a device pack in the CA Performance Management environment. This section provides you sample request URL and the corresponding response.

Scenario:

- Primary MultiController is installed at camm.ca.com with the default web server port number (8880).
- Two LocalControllers are connected to the primary MultiController.
- CAGuavus device pack is installed.

Request URL

Format: http://<*PrimaryMCHostName*>:<*WebServerPort*>/tim-web/rest/devicepacks/<*id*>/data/performance

For the given scenario: http://camm.ca.com:8880/tim-web/rest/devicepacks/CAGuavus/data/performance

Sample response

```
<response url="/devicepacks/CAGuavus/data/performance" version="2.5.0.208">
  <data>
    <performance id="CAGuavus">
      <root deviceType="CAGuavus" pollId="ENGINE_CAGuavus-1429692600878">
        <group>
```

```
          <TIM-Device>Guavus_MIDM</TIM-Device>
          <TIM-deviceIP/>
          <TIM-Branch>Guavus_SAP</TIM-Branch>
          <TIM-BranchDescr>MIDM</TIM-BranchDescr>
          <TIM-utc>1429691700</TIM-utc>
          <TIM-Delta>600</TIM-Delta>
          <DATA-SAP_SubscriberIB_DC_volume>0</DATA-SAP_SubscriberIB_DC_volume>
          <DATA-MIDM_total_duration>0</DATA-MIDM_total_duration>
          <DATA-Memory_utilization>0</DATA-Memory_utilization>
          <DATA-MIDM_aggregated_data_total_volume>0</DATA-
MIDM_aggregated_data_total_volume>
          <DATA-MIDM_enriched_data_DC_volume>0</DATA-MIDM_enriched_data_DC_volume>
          <DATA-Hadoop>0</DATA-Hadoop>
          <DATA-Syslog_records_data_volume>0</DATA-Syslog_records_data_volume>
          <DATA-MIDM_data_processing_DC_duration>0</DATA-
MIDM_data_processing_DC_duration>
          <DATA-MIDM_aggregated_data_DC_volume>0</DATA-MIDM_aggregated_data_DC_volume>
          <DATA-CPU_utilization>0</DATA-CPU_utilization>
          <DATA-Insta_status>0</DATA-Insta_status>
          <DATA-MIDM_data_transfer_total_duration>0</DATA-
MIDM_data_transfer_total_duration>
          <DATA-MIDM_data_processing_total_duration>0</DATA-
MIDM_data_processing_total_duration>
          <DATA-Arcsight_record_data_volume>0</DATA-Arcsight_record_data_volume>
          <DATA-Node_status>0</DATA-Node_status>
          <DATA-Nodes_in_Hadoop>0</DATA-Nodes_in_Hadoop>
          <DATA-MIDM_enriched_data_total_volume>0</DATA-MIDM_enriched_data_total_volume>
          <DATA-HDFS_utilization>0</DATA-HDFS_utilization>
          <DATA-SAP_PilotPacket_DC_volume>0</DATA-SAP_PilotPacket_DC_volume>
          <DATA-MIDM_data_transfer_DC_duration>0</DATA-MIDM_data_transfer_DC_duration>
          <DATA-SAP_IPFIX_DC_volume>0</DATA-SAP_IPFIX_DC_volume>
          <DATA-Load_usage>0</DATA-Load_usage>
        </group>
      </root>
    </performance>
  </data>
  <status>
    <code>200</code>
    <message>Success!</message>
  </status>
</response>
```

Read the Default Configuration of Device Pack Components

You can get the Engine or Presenter configuration supported in the CA Performance Management environment. This section provides you sample request URL and the corresponding response.

HTTP Verb: GET

Scenario:

- Primary MultiController is installed at camm.ca.com with the default web server port number (8880).
- Two LocalControllers are connected to the primary MultiController.

Get Configuration of the Supported Engine

Request URL

Format: http://<*CAMM_HOST*>:<*WebServerPort*>/tim-web/rest/devicepack/<*devicePackID*>/engine/configuration

For the given scenario: http://camm.ca.com:8880/tim-web/rest/devicepacks/CAGuavus/engine/configuration

Sample response

```
<response url="/devicepacks/CAGuavus/engine/configuration" version="2.x.y.z">
    <status>
        <code>200</code>
    <message>Success!</message>
    </status>
    <variables>
        <variable id="PRESENTER_ID">PRESENTER_CAPM</variable>
        <variable id="SCP_OR_SFTP">SFTP</variable>
        <variable id="EMS_IP_LIST">[cammsim.ca.com]</variable>
        <variable id="EMS_USERNAME">simuser</variable>
        <variable id="EMS_PASSWORD">ca123</variable>
        <variable id="EMS_BASEDIRECTORY">/export/troptest-vm1/Simulations/QA/CAGuavus/Bulkstats</variable>
        <variable id="EMS_INV_MAX_DOWNLOAD">8</variable>
        <variable id="EMS_INV_MAX_BATCHSIZE">4</variable>
        <variable id="EMS_PERF_MAX_DOWNLOAD">8</variable>
        <variable id="EMS_PERF_MAX_BATCHSIZE">4</variable>
        <variable id="EMS_FILE_PATTERN">.*.*</variable>
        <variable id="MAX_THREADS">4</variable>
        <variable id="INVENTORY_POLL_RATE">0 0 1</variable>
        <variable id="PERFORMANCE_POLL_RATE">0 */5 *</variable>
        <variable id="HOSTNAMECHANGE_ENFORCED">false</variable>
        <variable id="EMS_IGNORE_HISTORY">false</variable>
        <variable id="UNWANTED_DEVICE_LIST"/>
        <variable id="UNWANTED_BRANCH_LIST">TIM-XXXX</variable>
        <variable id="EMS_SSHKEYFILE"/>
        <variable id="EMS_PASS_PHRASE"/>
        <variable id="EMS_KNOWNHOSTS"/>
        <variable id="EMS_IGNORE_INCOMPLETEFILES">false</variable>
        <variable id="EXCLUDE_DIR_PATTERN"/>
    </variables>
</response>
```

Get the Configuration of the Supported Presenter

Request URL

Format: http://<*CAMM_HOST*>:<*WebServerPort*>/tim-web/rest/devicepacks/<*devicePackID*>/presenter/configuration

For the given scenario: http://camm.ca.com:8880/tim-web/rest/devicepacks/CAGuavus/presenter/configuration

Sample response

```
<response url="/devicepacks/CAGuavus/presenter/configuration" version="2.x.y.z">
    <status>
        <code>200</code>
        <message>Success!</message>
    </status>
    <variables>
        <variable id="OUTPUT_DIRECTORY">/opt/IMDataCollector/apache-karaf-2.3.0/MediationCenter</variable>
        <variable id="MAX_THREADS">4</variable>
    </variables>
</response>
```

Read the Data from Installed Components

You can get data, such as the discovered devices, poll data, and Engine and Presenter configurations for the device packs installed in the CA Performance Management environment. This section provides you sample request URL and the corresponding response.

HTTP Method: GET

Get the Discovered Devices

Scenario:

- Primary MultiController is installed at camm.ca.com with the default web server port number (8880).
- Two LocalControllers are connected to the primary MultiController.
- CAGuavus device pack is installed.

Request URL

Format: http://<*CAMM_HOST*>:<*WebServerPort*>/tim-web/rest/components/<componentID>/data/inventory

For the given scenario: http://camm.ca.com:8880/tim-web/rest/components /CAGuavus /data/inventory

Sample response

```
<response url="/components/ENGINE_CAGuavus/data/inventory" version="2.x.y.z">

    <data>

        <inventory id="CAGuavus:Guavus_AZUSCA21NSA_A_GV_HPC7_20"/>

        <inventory id="CAGuavus:Guavus_FSLAB"/>

        <inventory id="CAGuavus:Guavus_AZUSCA21NSA_A_GV_HPC7_21"/>

        <inventory id="CAGuavus:Guavus_MIDM"/>

        <inventory id="CAGuavus:Guavus_ALL_DC"/>

        <inventory id="CAGuavus:Guavus_smlabgvsgp1"/>

        <inventory id="CAGuavus:Guavus_smlabgvuip2"/>
```

```
        <inventory id="CAGuavus:Guavus_smlabgvuip1"/>

        <inventory id="CAGuavus:Guavus_smlabgvsgp2"/>

        <inventory id="CAGuavus:Guavus_smlabgvccp2"/>

        <inventory id="CAGuavus:Guavus_smlabgvccp1"/>

        <inventory id="CAGuavus:Guavus_smlabgvcnp1"/>

        <inventory id="CAGuavus:Guavus_smlabgvcmp2"/>

        <inventory id="CAGuavus:Guavus_smlabgvcmp1"/>

        <inventory id="CAGuavus:Guavus_SANMATEO"/>

        <inventory id="CAGuavus:Guavus_smlabgvcnp2"/>

    </data>

    <status>

        <code>200</code>

        <message>Success!</message>

    </status>

</response>
```

Get the Last Performance Poll Data

Scenario:

- Primary MultiController is installed at camm.ca.com with the default web server port number (8880).
- Two LocalControllers are connected to the primary MultiController.
- CAGuavus device pack is installed.

Request URL

Format: http://<*CAMM_HOST*>:<*WebServerPort*>/tim-web/rest/components/<*Component_Id*>/data/performance

For the given scenario: http://camm.ca.com:8880/tim-web/rest/components/CAGuavus/data/performance

Sample response

```
<response url="/components/ENGINE_CAGuavus/data/performance" version="2.x.y.z">

    <data>
```

```
        <performance id="ENGINE_CAGuavus">

        <root deviceType="CAGuavus" pollId="ENGINE_CAGuavus-1439205000576">

            <group>

                <TIM-Device>Guavus_smlabgvcnp1</TIM-Device>

                <TIM-deviceIP/>

                <TIM-Branch>Guavus_SAP</TIM-Branch>

                <TIM-BranchDescr>smlabgvcnp1</TIM-BranchDescr>

                <TIM-utc>1439204100</TIM-utc>

                <TIM-Delta>900</TIM-Delta>

                <DATA-Arcsight_record_data_volume>0</DATA-Arcsight_record_data_volume>

                <DATA-MIDM_data_processing_DC_duration>0</DATA-
MIDM_data_processing_DC_duration>

                <DATA-MIDM_enriched_data_total_volume>0</DATA-
MIDM_enriched_data_total_volume>

                <DATA-MIDM_aggregated_data_DC_volume>0</DATA-
MIDM_aggregated_data_DC_volume>

                <DATA-Hadoop>0</DATA-Hadoop>

                <DATA-SAP_SubscriberIB_DC_volume>0</DATA-SAP_SubscriberIB_DC_volume>

                <DATA-Insta_status>0</DATA-Insta_status>

                <DATA-MIDM_data_transfer_DC_duration>0</DATA-
MIDM_data_transfer_DC_duration>

                <DATA-Memory_utilization>0</DATA-Memory_utilization>

                <DATA-MIDM_data_processing_total_duration>0</DATA-
MIDM_data_processing_total_duration>

                <DATA-HDFS_utilization>0</DATA-HDFS_utilization>

                <DATA-MIDM_aggregated_data_total_volume>0</DATA-
MIDM_aggregated_data_total_volume>
```

```
                <DATA-MIDM_data_transfer_total_duration>0</DATA-
MIDM_data_transfer_total_duration>

                <DATA-MIDM_enriched_data_DC_volume>0</DATA-MIDM_enriched_data_DC_volume>

                <DATA-Load_usage>0</DATA-Load_usage>

                <DATA-MIDM_total_duration>0</DATA-MIDM_total_duration>

                <DATA-Node_status>0</DATA-Node_status>

                <DATA-SAP_PilotPacket_DC_volume>0</DATA-SAP_PilotPacket_DC_volume>

                <DATA-CPU_utilization>0</DATA-CPU_utilization>

                <DATA-SAP_IPFIX_DC_volume>0</DATA-SAP_IPFIX_DC_volume>

                <DATA-Syslog_records_data_volume>0</DATA-Syslog_records_data_volume>

                <DATA-Nodes_in_Hadoop>0</DATA-Nodes_in_Hadoop>

            </group>

.

.

.

.

.

            <group>

                <TIM-Device>Guavus_smlabgvuip2</TIM-Device>

                <TIM-deviceIP/>

                <TIM-Branch>Guavus_SAP</TIM-Branch>

                <TIM-BranchDescr>smlabgvuip2</TIM-BranchDescr>

                <TIM-utc>1439204040</TIM-utc>

                <TIM-Delta>300</TIM-Delta>

                <DATA-Arcsight_record_data_volume>0</DATA-Arcsight_record_data_volume>
```

```
                <DATA-MIDM_data_processing_DC_duration>0</DATA-
MIDM_data_processing_DC_duration>

                <DATA-MIDM_enriched_data_total_volume>0</DATA-
MIDM_enriched_data_total_volume>

                <DATA-MIDM_aggregated_data_DC_volume>0</DATA-
MIDM_aggregated_data_DC_volume>

                <DATA-Hadoop>0</DATA-Hadoop>

                <DATA-SAP_SubscriberIB_DC_volume>0</DATA-SAP_SubscriberIB_DC_volume>

                <DATA-Insta_status>0</DATA-Insta_status>

                <DATA-MIDM_data_transfer_DC_duration>0</DATA-
MIDM_data_transfer_DC_duration>

                <DATA-Memory_utilization>0.16</DATA-Memory_utilization>

                <DATA-MIDM_data_processing_total_duration>0</DATA-
MIDM_data_processing_total_duration>

                <DATA-HDFS_utilization>0</DATA-HDFS_utilization>

                <DATA-MIDM_aggregated_data_total_volume>0</DATA-
MIDM_aggregated_data_total_volume>

                <DATA-MIDM_data_transfer_total_duration>0</DATA-
MIDM_data_transfer_total_duration>

                <DATA-MIDM_enriched_data_DC_volume>0</DATA-MIDM_enriched_data_DC_volume>

                <DATA-Load_usage>0</DATA-Load_usage>

                <DATA-MIDM_total_duration>0</DATA-MIDM_total_duration>

                <DATA-Node_status>0</DATA-Node_status>

                <DATA-SAP_PilotPacket_DC_volume>0</DATA-SAP_PilotPacket_DC_volume>

                <DATA-CPU_utilization>32.34</DATA-CPU_utilization>

                <DATA-SAP_IPFIX_DC_volume>0</DATA-SAP_IPFIX_DC_volume>

                <DATA-Syslog_records_data_volume>0</DATA-Syslog_records_data_volume>

                <DATA-Nodes_in_Hadoop>0</DATA-Nodes_in_Hadoop>
```

```
            </group>

        </root>

        </performance>

    </data>

    <status>

        <code>200</code>

        <message>Success!</message>

    </status>

</response>
```

Get the Configuration of the Installed Engine

Scenario:

- Primary MultiController is installed at camm.ca.com with the default web server port number (8880).
- Two LocalControllers are connected to the primary MultiController.
- CAGuavus device pack is installed.

Request URL

Format: http://<*CAMM_HOST*>:<*WebServerPort*>/tim-web/rest/components/<*component_Id*>/data/configuration

For the given scenario: http://camm.ca.com:8880/tim-web/rest/components/ENGINE_CAGuavus/data/configuration

Sample response

```
<response url="/components/ENGINE_CAGuavus/data/configuration" version="2.x.y.z">
    <status>
        <code>200</code>
        <message>Success!</message>
    </status>
    <variables>
        <variable id="PRESENTER_ID">PRESENTER_CAPM</variable>
        <variable id="SCP_OR_SFTP">SFTP</variable>
        <variable id="EMS_IP_LIST">[cammsim.ca.com]</variable>
        <variable id="EMS_USERNAME">simuser</variable>
        <variable id="EMS_PASSWORD">ca123</variable>
        <variable id="EMS_BASEDIRECTORY">/export/troptest-vm1/Simulations/QA/CAGuavus/Bulkstats</variable>
        <variable id="EMS_INV_MAX_DOWNLOAD">8</variable>
        <variable id="EMS_INV_MAX_BATCHSIZE">4</variable>
        <variable id="EMS_PERF_MAX_DOWNLOAD">8</variable>
        <variable id="EMS_PERF_MAX_BATCHSIZE">4</variable>
        <variable id="EMS_FILE_PATTERN">.*.*</variable>
        <variable id="MAX_THREADS">4</variable>
        <variable id="INVENTORY_POLL_RATE">0 0 1</variable>
```

```
        <variable id="PERFORMANCE_POLL_RATE">0 */5 *</variable>
        <variable id="HOSTNAMECHANGE_ENFORCED">false</variable>
        <variable id="EMS_IGNORE_HISTORY">false</variable>
        <variable id="UNWANTED_DEVICE_LIST"/>
        <variable id="UNWANTED_BRANCH_LIST">TIM-XXXX</variable>
        <variable id="EMS_SSHKEYFILE"/>
        <variable id="EMS_PASS_PHRASE"/>
        <variable id="EMS_KNOWNHOSTS"/>
        <variable id="EMS_IGNORE_INCOMPLETEFILES">false</variable>
        <variable id="EXCLUDE_DIR_PATTERN"/>
        <variable id="MAXIMUM_JAVA_HEAP_SIZE">2048</variable>
    </variables>
</response>
```

Get the Configuration of the Installed Presenter

Scenario:

- Primary MultiController is installed at camm.ca.com with the default web server port number (8880).
- Two LocalControllers are connected to the primary MultiController.
- PRESENTER_CAPM is installed.

Request URL

Format: http://<*CAMM_HOST*>:<*WebServerPort*>/tim-web/rest/components/<*component_id*>/data/configuration

For the given scenario: http://camm.ca.com:8880/tim-web/rest/components/PRESENTER_CAPM/data/configuration

Sample response

```
<response url="/components/PRESENTER_CAPM/data/configuration" version="2.x.y.z">

    <status>

        <code>200</code>

        <message>Success!</message>

    </status>

    <variables>

        <variable id="OUTPUT_DIRECTORY">/opt/IMDataCollector/apache-karaf-2.3.0/
MediationCenter</variable>

        <variable id="MAX_THREADS">4</variable>

        <variable id="MAXIMUM_JAVA_HEAP_SIZE">2048</variable>

    </variables>

</response>
```

Install Components

You can install a device pack (Engine, CERT, and Presenter) in the CA Performance Management environment with the default or customized configurations. This section provides you sample request URL and the corresponding response.

NOTE

CERT is automatically deployed in Data aggregator while installing the Engine.

HTTP Method: POST

Scenario:

- Primary MultiController is installed at camm.ca.com with the default web server port number (8880).
- Two LocalControllers are connected to the primary MultiController.
- CAGuavus device pack to be installed.

Request URL

Format: http://<*CAMM_HOST*>:<*WebServerPort*>/tim-web/rest/devicepacks/<*devicepack_ID*>

For the given scenario: http://camm.ca.com:8880/tim-web/rest/devicepacks/CAGuavus

Sample Request Body:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
 <request url="/devicepacks/CAGuavus">
    <devicepacks>
        <devicepack>
            <configurations>
                <engine id="ENGINE_CAGuavus">
                    <variables>
                        <Variable name="PRESENTER_ID" type="STR">PRESENTER_CAPM</
Variable>
                        <Variable name="SCP_OR_SFTP" type="STR">SFTP</Variable>
                        <Variable name="EMS_IP_LIST" type="STR">[cammsim.ca.com]</
Variable>
                        <Variable name="EMS_USERNAME" type="STR">simuser</Variable>
                        <Variable name="EMS_PASSWORD" type="PASS">ca123</Variable>
                        <Variable name="EMS_BASEDIRECTORY" type="STR">/export/troptest-
vm1/Simulations/QA/CAGuavus/Bulkstats</Variable>
                        <Variable name="EMS_INV_MAX_DOWNLOAD" type="STR">8</Variable>
                        <Variable name="EMS_INV_MAX_BATCHSIZE" type="STR">4</Variable>
                        <Variable name="EMS_PERF_MAX_DOWNLOAD" type="STR">8</Variable>
                        <Variable name="EMS_PERF_MAX_BATCHSIZE" type="STR">4</Variable>
                        <Variable name="EMS_FILE_PATTERN" type="STR">.*.*</Variable>
                        <Variable name="MAX_THREADS" type="STR">4</Variable>
                        <Variable name="INVENTORY_POLL_RATE" type="STR">0 0 1</Variable>
                        <Variable name="PERFORMANCE_POLL_RATE" type="STR">0 */5 *</
Variable>
```

```
                        <Variable name="HOSTNAMECHANGE_ENFORCED" type="STR">false</
Variable>
                        <Variable name="EMS_IGNORE_HISTORY" type="STR">false</Variable>
                        <Variable name="UNWANTED_DEVICE_LIST" type="STR"></Variable>
                        <Variable name="UNWANTED_BRANCH_LIST" type="STR">TIM-XXXX</
Variable>
                        <Variable name="EMS_SSHKEYFILE" type="STR"></Variable>
                        <Variable name="EMS_PASS_PHRASE" type="STR"></Variable>
                        <Variable name="EMS_KNOWNHOSTS" type="STR"></Variable>
                        <Variable name="EMS_IGNORE_INCOMPLETEFILES" type="STR">false</
Variable>
                        <Variable name="EXCLUDE_DIR_PATTERN" type="STR"></Variable>
                    </variables>
                    <configuration>
                        <MAXIMUM_JAVA_HEAP_SIZE>1024</MAXIMUM_JAVA_HEAP_SIZE>
                    </configuration>
                    <lcaddress>10.134.15.53</lcaddress>
                </engine>
                <presenter id="PRESENTER_ID">
                    <variables>
                         <Variable name="OUTPUT_DIRECTORY" TYPE="STR">/opt/
IMDataCollector/apache-karaf-2.3.0/MediationCenter</Variable>
                         <Variable name="MAX_THREADS" TYPE="STR">4</Variable>
                     </variables>
                     <configuration>
                        <MAXIMUM_JAVA_HEAP_SIZE>1024</MAXIMUM_JAVA_HEAP_SIZE>
                     </configuration>
                     <lcaddress>10.134.15.53</lcaddress>
                </presenter>
            </configurations>
        </devicepack>
    </devicepacks>
</request>
```

Sample response:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<response url="/devicepacks/CAGuavus" version="3.x.y.z">

    <devicepacks>

        <devicepack type="CAGuavus">

            <cert id="CERT_CAGuavus">
```

```
                <message>Success</message>
            </cert>
            <engine id="ENGINE_CAGuavus">
                <message>Success</message>
            </engine>
            <presenter id="PRESENTER_ID">
                <message>Success</message>
            </presenter>
        </devicepack>
    </devicepacks>
    <status>
        <code>201</code>
        <message>Success</message>
    </status>
</response>
```

Upgrade Components

You can upgrade a device pack (Engine, CERT, and Presenter) with the default or customized configurations in the CA Performance Management environment. This section provides you sample request URL and the corresponding response.

NOTE

CERT is automatically deployed in Data aggregator while upgrading the Engine.

HTTP Verb: PUT

Scenario:

- Primary MultiController is installed at camm.ca.com with the default web server port number (8880).
- Two LocalControllers are connected to the primary MultiController.
- CAGuavus device pack is installed and needs to be upgraded.

Request URL

Format: http://<*CAMM_HOST*>:<*WebServerPort*>/tim-web/rest/devicepacks/<*Devicepack_Name*>

For the given scenario: http://camm.ca.com:8880/tim-web/rest/devicepacks/CAGuavus

Sample Request Body:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
 <request url="/devicepacks/CAGuavus">
    <devicepacks>
        <devicepack>
            <configurations>
                <engine id="ENGINE_CAGuavus">
                    <variables>
                        <Variable name="PRESENTER_ID" type="STR">PRESENTER_CAPM</Variable>
                        <Variable name="SCP_OR_SFTP" type="STR">SFTP</Variable>
                        <Variable name="EMS_IP_LIST" type="STR">[cammsim.ca.com]</Variable>
                        <Variable name="EMS_USERNAME" type="STR">simuser</Variable>
                        <Variable name="EMS_PASSWORD" type="PASS">ca123</Variable>
                        <Variable name="EMS_BASEDIRECTORY" type="STR">/export/troptest-vm1/Simulations/QA/
CAGuavus/Bulkstats</Variable>
                        <Variable name="EMS_INV_MAX_DOWNLOAD" type="STR">8</Variable>
                        <Variable name="EMS_INV_MAX_BATCHSIZE" type="STR">4</Variable>
                        <Variable name="EMS_PERF_MAX_DOWNLOAD" type="STR">8</Variable>
                        <Variable name="EMS_PERF_MAX_BATCHSIZE" type="STR">4</Variable>
                        <Variable name="EMS_FILE_PATTERN" type="STR">.*.*</Variable>
                        <Variable name="MAX_THREADS" type="STR">4</Variable>
                        <Variable name="INVENTORY_POLL_RATE" type="STR">0 0 1</Variable>
                        <Variable name="PERFORMANCE_POLL_RATE" type="STR">0 */5 *</Variable>
                        <Variable name="HOSTNAMECHANGE_ENFORCED" type="STR">false</Variable>
                        <Variable name="EMS_IGNORE_HISTORY" type="STR">false</Variable>
                        <Variable name="UNWANTED_DEVICE_LIST" type="STR"></Variable>
                        <Variable name="UNWANTED_BRANCH_LIST" type="STR">TIM-XXXX</Variable>
                        <Variable name="EMS_SSHKEYFILE" type="STR"></Variable>
                        <Variable name="EMS_PASS_PHRASE" type="STR"></Variable>
                        <Variable name="EMS_KNOWNHOSTS" type="STR"></Variable>
                        <Variable name="EMS_IGNORE_INCOMPLETEFILES" type="STR">false</Variable>
                        <Variable name="EXCLUDE_DIR_PATTERN" type="STR"></Variable>
                    </variables>
                    <configuration>
                        <MAXIMUM_JAVA_HEAP_SIZE>2048m</MAXIMUM_JAVA_HEAP_SIZE>
                    </configuration>
                </engine>
                <presenter id="PRESENTER_ID">
                    <variables>
                         <Variable name="OUTPUT_DIRECTORY" TYPE="STR">/opt/IMDataCollector/apache-karaf-2.3.0/
MediationCenter</Variable>
                         <Variable name="MAX_THREADS" TYPE="STR">4</Variable>
                     </variables>
                     <configuration>
                        <MAXIMUM_JAVA_HEAP_SIZE>2048m</MAXIMUM_JAVA_HEAP_SIZE>
                     </configuration>
                </presenter>
            </configurations>
        </devicepack>
    </devicepacks>
</request>
```

Sample response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response url="/devicepacks/CAGuavus" version="3.x.y.z">
    <devicepacks>
        <devicepack type="CAGuavus">
            <cert id="CERT_CAGuavus">
                <message>Success</message>
            </cert>
            <engine id="ENGINE_CAGuavus">
                <message>Success</message>
            </engine>
            <presenter id="PRESENTER_ID">
                <message>Success</message>
            </presenter>
        </devicepack>
    </devicepacks>
    <status>
        <code>200</code>
        <message>Success</message>
    </status>
</response>
```

Start or Stop the Components

You can start or stop the installed device pack in the CA Performance Management environment. This section provides you sample request URL and the corresponding response.

HTTP Method: POST

Scenario:

- Primary MultiController is installed at camm.ca.com with the default web server port number (8880).
- Two LocalControllers are connected to the primary MultiController.
- CAGuavus device pack and PRESENTER_CAPM are installed.

Start the Device Pack Engine on all LocalControllers

Request URL

Format: http://<*CAMM_HOST*>:<*WebServerPort*>/tim-web/rest/components/<*component_id*>?action=start

For the given scenario: http://camm.ca.com:8880/tim-web/rest/components/ENGINE_CAGuavus?action=start

Sample response

```
<response url="/components/ENGINE_CAGuavus?action=start" version="2.x.y.z">
 <components>
  <component id="ENGINE_CAGuavus" status="started"/>
  <status>Successfully Started</status> </components>
 <status>
  <code>200</code>
  <message>Success!</message>
 </status>
</response>
```

Stop the Device Pack Engine on all LocalControllers

Request URL

Format: http://<*CAMM_HOST*>:<*WebServerPort*>/tim-web/rest/components/<*component_id*>?action=start

For the given scenario: http://camm.ca.com:8880/tim-web/rest/components/ ENGINE_CAGuavus ?action=stop

Sample response

```
<response url="/components/ENGINE_CAGuavus?action=stop" version="2.x.y.z">
 <components>
  <component id="ENGINE_CAGuavus" status="stopped"/>
  <status>Successfully Stopped</status>
 </components>
 <status>
  <code>200</code>
  <message>Success!</message>
 </status>
</response>
```

Start the Presenter on all LocalControllers

Request URL

Format: http://<*CAMM_HOST*>:<*WebServerPort*>/tim-web/rest/components/<*component_id*>?action=start

For the given scenario: http://camm.ca.com:8880/tim-web/rest/components/PRESENTER_CAPM?action=start

Sample response

```
<response url="/components/PRESENTER_CAPM?action=start" version="2.x.y.z">
 <components>
  <component id="PRESENTER_CAPM" status="started"/>
  <status>Successfully Started</status> </components>
 <status>
  <code>200</code>
  <message>Success!</message>
 </status>
</response>
```

Stop the Presenter on all LocalControllers

Request URL

Format: http://<*CAMM_HOST*>:<*WebServerPort*>/tim-web/rest/components/<*component_id*>?action=start

For the given scenario: http://camm.ca.com:8880/tim-web/rest/components/ PRESENTER_CAPM ?action=stop

Sample response

```
<response url="/components/PRESENTER_CAPM?action=stop" version="2.x.y.z">
 <components>
  <component id="PRESENTER_CAPM" status="stopped"/>
  <status>Successfully Stopped</status>
 </components>
 <status>
  <code>200</code>
  <message>Success!</message>
```

```
  </status>
</response>
```

Read List of LocalControllers

You can get the list of LocalControllers in the CA Performance Management environment. This section provides you sample request URL and the corresponding response.

Scenario:

- Primary MultiController is installed at camm.ca.com with the default web server port number (8880).
- One LocalController is connected to the primary MultiController.

HTTP Verb: GET

Request URL

Format: http://<*PrimaryMCHostName*>:<*WebServerPort*>/tim-web/rest/lc

For the given scenario: http://camm.ca.com:8880/tim-web/rest/lc

Sample response

```
<response url="/lc" version="2.x.y.z">

    <lcs>

        <lc id="LC-127.0.0.1" status="started"/>

    </lcs>

    <status>

        <code>200</code>

        <message>Success!</message>

    </status>

</response>
```

Configuring

This section deals with configuring the components in DX NetOps and EMS Integration Profiles for DX NetOps for Infrastructure Management.

Generic Executor Configuration

The Generic Executor typically requires no configuration after the initial installation. Only one Generic Executor is required per server unless you require the components to run with different user IDs.

The Generic Executor configuration is installed in the Generic Executor directory, named GE_<*userid*>, in the DX NetOps MM HOME directory. The <*userid*> is the user name that you specified during the installation.

The Generic Executor directory contains a file that is named LocalConfig-ge.xml.

When the Generic Executor assumes the role of a component, it executes using the same user ID configured in the LocalConfig-ge.xml.

Example of a LocalConfig-ge.xml file for UNIX

```
<?xml version="1.0" encoding="UTF-8"?>
<AppDaemon>
    <Names>
    </Names>
    <Paths>
        <Path name="tim.base">/opt/CA/CAMM</Path>
        <Path name="appHome">${tim.base}/GE_camm</Path>
        <Path name="configBase">${appHome}/tmp</Path>
    </Paths>
    <Binding>
        <Port>29560</Port>
        <UserId>camm</UserId>
    </Binding>
</AppDaemon>
```

Example of a LocalConfig-ge.xml file for Windows

For a Windows user, the primary Generic Executor configuration file is slightly different:

```
<?xml version="1.0" encoding="UTF-8"?>
<AppDaemon>
    <Names>
    </Names>
    <Paths>
        <Path name="tim.base">/opt/CA/CAMM</Path>
        <Path name="appHome">${tim.base}/GE_camm</Path>
        <Path name="configBase">${appHome}/tmp</Path>
    </Paths>
    <Binding>
        <Port>29560</Port>
        <UserId>camm</UserId>
    </Binding>
    <CompanyItems>
       <Item>
        <Name>MC</Name>
        <Config>${tim.base}/MC/LocalConfig-mc.xml</Config>
        <Port>29599</Port>
    </Item>
       <Item>
        <Name>LC</Name>
        <Config>${tim.base}/LC/LocalConfig-lc.xml</Config>
        <Port>29598</Port>
    </Item>
    </CompanyItems>
</AppDaemon>
```

The <*CompanyItems*> section added to the Generic Executor configuration file defines the locations and service ports for MultiController and LocalController. With <CompanyItems> defined, the Generic Executor checks these items every

one minute and if they are not running, the Generic Executor starts them automatically. This configuration enables the MultiController and the LocalController to start without external intervention.

The <*CompanyItems*> feature is disabled after installation by default. You can execute the camm.init.install command to enable it.

NOTE
Only the primary Generic Executor requires the <*CompanyItems*> section. Other Generic Executors must not contain this section.

How the Generic Executor Works

The Generic Executor is a reusable entity and is the foundation of DX NetOps components. When a server starts, it is mandatory that at least one Generic Executor component exists and is started.

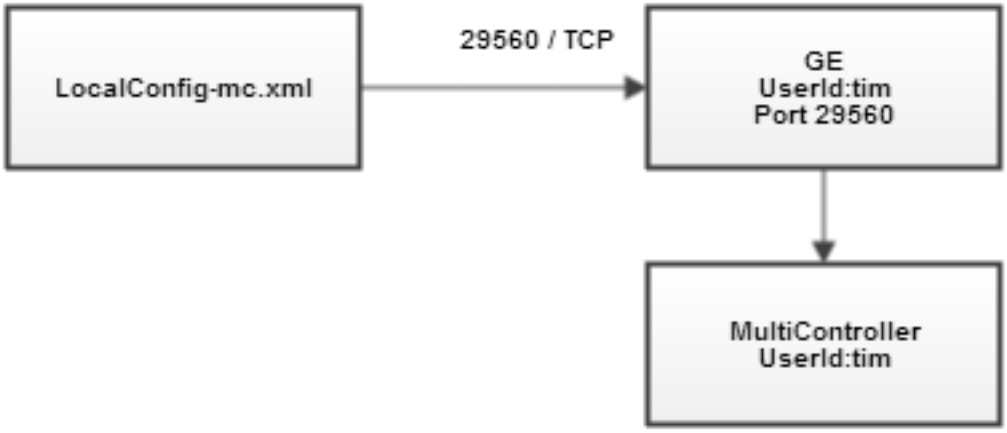
By default, the LocalConfig-ge.xml file contains the User ID of the CAMM_USER and the TCP port information. The following diagram illustrates the Generic Executor on TCP port 29560 and the CAMM_USER who installed DX NetOps:

Figure 88: How the Generic Executor Works_1--XML

If you require a MultiController to run on a server, you need a LocalConfig-mc.xml configuration file for the MultiController and the TCP port for the Generic Executor.

The following diagram illustrates how the LocalConfig-mc.xml file is sent to the Generic Executor on TCP port 29560, and the Generic Executor starts a new process for the MultiController:

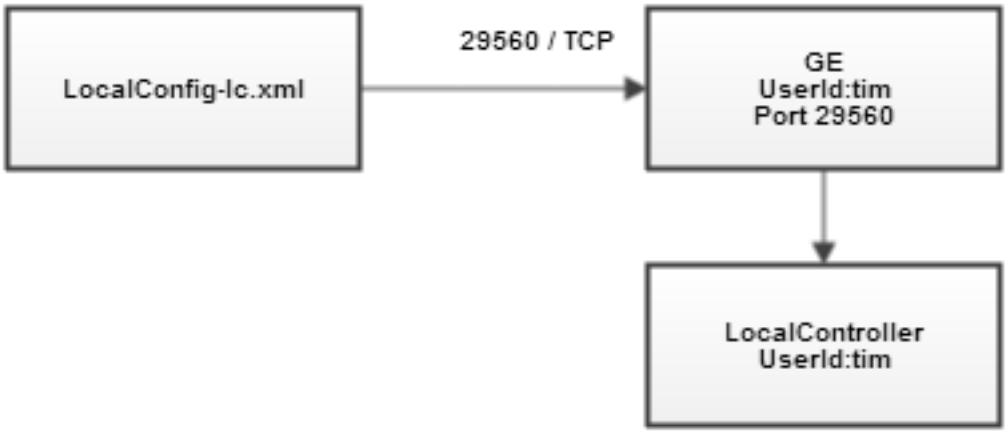
Figure 89: How the Generic Executor Works_2--XML



If a LocalController must run on a server, only a LocalConfig-lc.xml configuration file for the LocalController and the TCP port for the Generic Executor are required.

The following diagram illustrates how the LocalConfig-lc.xml file is sent to the Generic Executor on TCP port 29560, and the Generic Executor starts a new process for the LocalController:

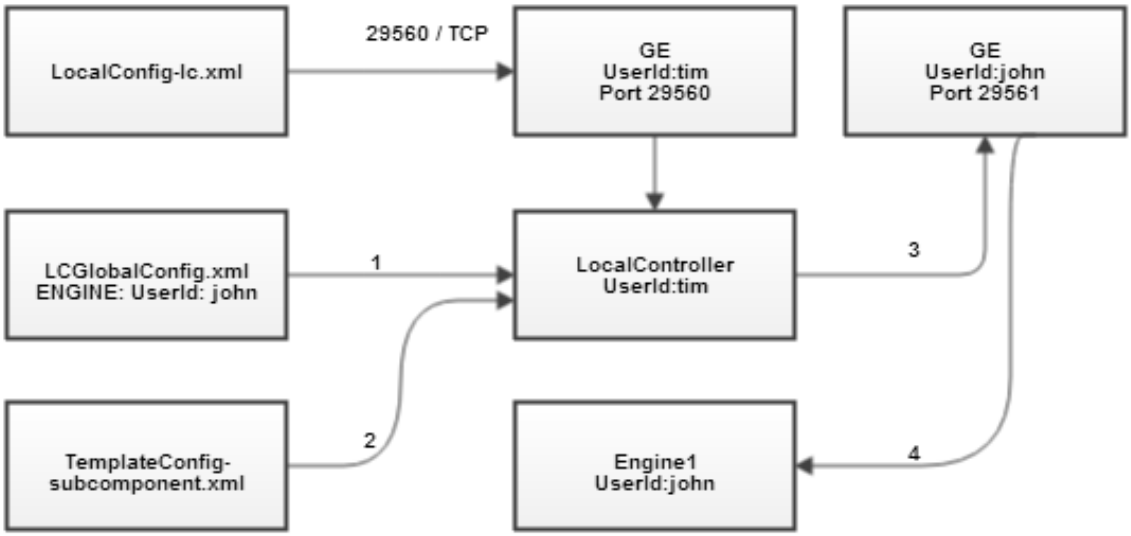
Figure 90: How the Generic Executor Works_3--XML



When the LocalController starts, it reads the LCGlobalConfig.xml file from its repository. The LCGlobalConfig.xml file contains the subcomponents and port information.

If an Engine is required to run on a server, the LocalController uses the LCGlobalConfig.xml file to determine which user ID it must run as. In the LocalController directory, the LocalController looks up the ExecutorMap.xml file. This action allows the LocalController to locate a Generic Executor running as the required user ID. The TemplateConfig-subcomponent.xml file is then combined with the specific configuration from the LCGlobalConfig.xml file. The combination is sent to the TCP port of the Generic Executor.

Figure 91: How the Generic Executor Works_4--XML



Generic Executor Configuration Options

The installer requires the following options for the configuration:

- **tim.base**
  Specifies the base directory of DX NetOps.
- **appHome**

Specifies the base directory of the DX NetOps Generic Executor.

- **configBase**
  Specifies the temporary directory for storing configurations of components that the Generic Executor executes now.
- **Port**
  Specifies the TCP port on which the Generic Executor listens.
  Default: TCP 29560
- **UserId**
  Specifies the User ID that owns the Generic Executor process and its subprocesses.

Generic Executor Startup Sequence

You can manually start the Generic Executor process (or service on Windows) using the startall or startall.bat script. However, if camm.init.install is executed, the Generic Executor starts automatically on system startup.

Add Another Generic Executor

Add another Generic Executor if you require a component to execute as a different User ID.

Follow these steps:

1.
2. Add a user to the LocalController that requires the additional user ID. The new user must be a member of the same group that owns the CAMM_HOME directory.

   NOTE
   Repeat this procedure for any standby LocalControllers.
3.
4. Log in as the new user.
5.
6. Execute the following command to add the new Generic Executor:

   ```
   (For UNIX) shell# /opt/CA/CAMM/tools/camm.ge.install -p <Port number>
   (For Windows) > %TIM_HOME%/tools/camm.ge.install.bat -p <Port number>
   ```

   Where, TIM_HOME is the DX NetOps installation directory and <*Port number*> is the port number to install the Generic Executor.
   The additional Generic Executor is added.

   NOTE
   A configuration file in the LocalController directory that is named ExecutorMap.xml is modified to provide a map between the required User ID and the TCP port.

When the Engine subcomponent is started, the LocalController references the User ID provided in the LCGobalConfig for that component. If the User ID is not the default User ID, the ExecutorMap is referenced and the TCP port for the alternative Generic Executor starts the subcomponent.

This reference is useful when the Engine uses user or host-based RSA key authentication to SFTP or SCP the information.

Enable SSL Communication Between Components

The TLS 1.0 had some known cryptographic design flaws. With this release, we enhanced the remote service to accept connections using TLS 1.2 encryption.

Use the validateSSL.sh script in **CAMM_Home/tools** directory to enable SSL communication between DX NetOps Mediation Manager components. You can either use the Signed Certificate and Key issues by Certificate Authority (CA) or external Keystore with CA Signed certificate, to initialize the SSL communications between the DX NetOps Mediation Manager components. When you select any option to enable SSL, the DX NetOps Mediation Manager stops the connected components. Perform the following steps on both MultiController and LocalController. You must restart the components manually for the change to take effect.

Follow these steps:

1. Run the `validateSSL.sh` script from the `<CAMM_Home>/tools` directory. The script prompts you to select the certificate.

   ```
   Please select any one from the below options:
    1. Do you have an existing signed certificate and key?
    2. You have a Keystore file and want to validate the keystore?
   ```
2. Select the option that suits your certificate validation.
3. At the prompt, specify the location and filename of the keystore.

   ```
   Specify the location and filename of the Keystore:
   ```

   For Example: /opt/SSLs/myserver.keystore
4. Specify the keystore and SSL certificate password when prompted.

   ```
   Please enter the Password for Keystore
   Please enter the Password for SSL Certificate>
   ```

   The script validates the keystore and displays the certificate information.
5. (Optional) Run the following command to validate the password encryption.

   ```
   cat ../GE_root/Keystore_Details.xml
   ```

   The script displays the encrypted Keystore details.
6. From the MultiController or LocalControlle, start the controller using the `./startall` command.
7. Navigate to the log directory on your MultiController or LocalController, to view the status.
8. Use the `cat <Log_Filename>.log` command to view status.
9. In the log file look for the following status.

   ```
   INFO: Initializing SSLContext with config file/opt/CA/CAMM/tools/myserver.keystore
   ```

   All further communication between the DX NetOps Mediation Manager components use the new SSL encryption.

MultiController Configuration

The MultiController contains the centralized license file and maintains a heartbeat with the components in the cluster.

The configuration for LocalControllers, Engines, and Presenters in the cluster is also located in the repository of the MultiController.

The MultiController listens for heartbeat operations from cluster members on TCP port 29599.

During the console installation, you can configure only the mandatory settings to install a basic MultiController. However, you can manually edit the MC or LocalConfig-mc.xml file.

Sample LocalConfig-mc.xml file (basic configuration)

```
<?xml version="1.0" encoding="UTF-8"?>
<LocalConfig>
      <Description>Configuration for Multi Controller</Description>
      <Names>
```

```
            <Name name="mainClass">com.torokina.tim.mc.Main</Name>
            <Name name="appName">CAMM-Multi-Controller</Name>
            <Name name="appShortName">MC</Name>
            <Name name="primaryMcAddress">127.0.0.1</Name>
            <Name name="secondaryMcAddress"></Name>
            <Name name="primaryMcPort">29599</Name>
            <Name name="secondaryMcPort">-1</Name><
            <Name name="myMode">primary</Name>
            <Name name="myAddress">127.0.0.1</Name>
            <Name name="mcPort">29599</Name>
            <Name name="otherMcAddress"></Name>
            <Name name="otherMcPort">-1</Name>
            <Name name="heartbeatFrequency">15</Name>
            <Name name="heartbeatTimeout">180</Name>
            <Name name="repositoryFrequency">15</Name>
      </Names>
      <Paths>
            <Path name="license">${tim.base}/license.lic</Path>
      </Paths>
<LocalConfig>
```

Sample LocalConfig-mc.xml file (hidden logging and cleanups configuration)

```
    <Logging>
        <LogLevel>INFO</LogLevel>
        <LogDirectory>${logbase}</LogDirectory>
        <ObjectLogging>
          <ObjectToLog>
            <ObjectName>com.torokina.tim.config</ObjectName>
            <ObjectLogLevel>TRACE</ObjectLogLevel>
          </ObjectToLog>
        </ObjectLogging>
    </Logging>
    <CleanUps>
        <CleanUp>
            <CleanUpName>clean-temporary-directory</CleanUpName>
            <CleanUpAction>delete</CleanUpAction>
            <CleanUpTarget>${tmp}</CleanUpTarget>
            <Parameter>
                <ParameterName>expire</ParameterName>
                <ParameterValue>3d</ParameterValue>
            </Parameter>
        </CleanUp>
        <CleanUp>
            <CleanUpName>archive-log-directory</CleanUpName>
            <CleanUpAction>archive</CleanUpAction>
            <CleanUpTarget>${logbase}</CleanUpTarget>
            <Parameter>
                <ParameterName>expire</ParameterName>
                <ParameterValue>3d</ParameterValue>
            </Parameter>
        </CleanUp>
        <CleanUp>
            <CleanUpName>clean-log-directory</CleanUpName>
```

```
            <CleanUpAction>delete</CleanUpAction>
            <CleanUpTarget>${logbase}</CleanUpTarget>
            <Parameter>
                <ParameterName>expire</ParameterName>
                <ParameterValue>7d</ParameterValue>
            </Parameter>
        </CleanUp>
    </CleanUps>
```

Specifying any of these fields in the correct XML structure in the LocalConfig-mc.xml file overwrites the default content. For example, the following configuration changes the default log level to FINEST.

Changing the default log level to FINEST

```
<?xml version="1.0" encoding="UTF-8"?>
<LocalConfig>
      ... ...
      <Logging>
        <LogLevel>FINEST</LogLevel>
        <LogDirectory>${logbase}</LogDirectory>
      </Logging>
      ... ...
</LocalConfig>
```

Sample LocalConfig-mc.xml (MultiController runtime) file

```
<?xml version="1.0" encoding="UTF-8"?>
<Runtime>
   <Names>
      <Name name="mainClass">com.torokina.tim.mc.Main</Name>
      <Name name="appName">CAMM-Multi-Controller</Name>
      <Name name="appShortName">MC</Name>
      <Name name="primaryMcAddress">127.0.0.1</Name>
      <Name name="secondaryMcAddress"/>
      <Name name="primaryMcPort">29599</Name>
      <Name name="secondaryMcPort">-1</Name>
      <Name name="myMode">primary</Name>
      <Name name="myAddress">127.0.0.1</Name>
      <Name name="mcPort">29599</Name>
      <Name name="otherMcAddress"/>
      <Name name="otherMcPort">-1</Name>
      <Name name="heartbeatFrequency">15</Name>
      <Name name="heartbeatTimeout">180</Name>
      <Name name="repositoryFrequency">15</Name>
      <Name name="lcPort">29598</Name>
      <Name name="manageable">469</Name>
   </Names>
   <Paths>
      <Path name="license">${tim.base}/license.lic</Path>
      <Path name="apphome">${tim.base}/${appShortName}</Path>
      <Path name="runtimeConfig">${apphome}/runtime.xml</Path>
      <Path name="tmp">${apphome}/tmp</Path>
      <Path name="logbase">${apphome}/logs</Path>
      <Path name="basedir">${tim.base}</Path>
   </Paths>
```

```
</Runtime>
```

The runtime.xml file is merged with the hidden configuration and then starts the MultiController component.

MultiController Configuration Options

Use the following options to configure MultiController:

- **Path**
  Lets you specify the path information for the following items:
  - **basedir**
    Specifies the DX NetOps base directory.
  - **apphome**
    Specifies the DX NetOps MultiController application home directory.
  - **tmp**
    Specifies the DX NetOps MultiController temporary files directory.
  - **logbase**
    Specifies the DX NetOps MultiController log directory.
  - **runtimeConfig**
    Specifies the DX NetOps MultiController runtime XML configuration that the Generic Executor provides.
- **Java**
  Lets you specify options for using Java.
  - **CommandPath**
    Specifies the full path to the Java executer that the Generic Executor calls to start the MultiController.
  - **ClassPath/JarBase**
    Lets you create a list with one or more entries to add to ClassPath.
  - **Options**
    Specifies the command-line options parsed to Java.
  - **Environment**
    Specifies the environment variables that execute the MultiController component.
  - **MainClass/Class**
    Specifies the main Java class to execute.
  - **MainClass/Args**
    Specifies the arguments parsed to the Java class.
- **Runtime**
  Lets you specify runtime options.
  - **Binding/BindAddress**
    Specifies the IP address to which the MultiController component binds.
    Value: Use IP address 0.0.0.0 for all MultiController components. For two or more IP addresses, use a comma-separated list.
    The MultiController binds to TCP port 29599.
  - **Binding/MyAddress**
    Specifies the IP address that the MultiController uses to identify itself.

    NOTE
    The IP address must be a valid IP address on this host.
  - **MultiControllerConfig/Mode**
    Specifies the operating mode of the MultiController.
    Value: Specify either Primary or Secondary.
  - **MultiControllerConfig/MCAddresses/Other**
    Specifies the IP address of the other MultiControllers in the cluster.
  - **MultiControllerConfig/Heartbeat/ParameterName == frequency**

Specifies the frequency of heartbeat messages that are sent to other MultiControllers.
  - **MultiControllerConfig/Heartbeat/ParameterName == timeout**
    Specifies the period that this MultiController waits for heart beats from the LocalController components. If the MultiController does not receive a heartbeat from a LocalController for 180 seconds, the MultiController triggers a failover to the first available LocalController.
- **Logging**
  Lets you specify logging options.
  - **LogLevel**
    Specifies the logging level in the output log files.
    Value: Specify DEBUG, TRACE, INFO, WARNING, or ERROR.
  - **LogDirectory**
    Specifies the logging level in the output log files.
    Value: Specify DEBUG, TRACE, INFO, WARNING, or ERROR.
  - **ObjectLogging/ObjectToLog/ObjectName**
    Specifies the Java class name on which to enable logging.
  - **ObjectLogging/ObjectToLog/ObjectLogLevel**
    Specifies the logging level for a Java class.
- **Cleanup**
  Lets you specify cleanup options.
  - **CleanUpName**
    Specifies the descriptive name of the cleanup.
  - **CleanUpAction**
    Specifies the cleanup action.
    Value: Specify Delete or Archive.
  - **CleanupTarget**
    Specifies the directory to clean up.
    Value: Can specify using a DX NetOps variable such as ${camm.variable}.
  - **Parameter/ParameterName - Parameter/ParameterValue**
    Specifies a parameter to expire and its expiration time in the format of <n><unit>. For example:
    10d = 10 days
    10h = 10 hours
    10m = 10 minutes

Start and Stop the MultiController Manually

The init.camm script automatically starts the MultiController process. You can use the cammCtrl utility to manually stop or start the MultiController component independently.

Follow these steps:

1. Log in as the CAMM_USER and go to the DX NetOps MM home directory.
2. Execute the following command to start the MultiController:
   ```
   /opt/CA/CAMM# tools/startall -c mc
   ```
3. Execute the following command to stop the MultiController:
   ```
   /opt/CA/CAMM # tools/stopall - c mc
   ```

LocalController Configuration

The LocalController is an essential service that is installed on each server in the cluster.

The LocalController performs the following key functions:

- Facilitates communication between subcomponents running on the local server and the remote MultiControllers.
- Monitors the performance and availability of Engines and Presenters and restarts any failed components.
- Listens for heartbeat operations from cluster members. By default, the LocalController listens on port 29598 or TCP.
- Sends heartbeat information to the MultiControllers.
- Starts, stops, and restarts local Engines and Presenters.

During the GUI installation, you can configure only the mandatory options that are required to install a basic LocalController. However, you may manually edit the LC or LocalConfig-lc.xml file.

Sample LocalConfig-lc.xml file (basic configuration)

```
<LocalConfig>
      <Names>
            <Name name="mainClass">com.torokina.tim.lc.Main</Name>
            <Name name="primaryMcAddress">127.0.0.1</Name>
            <Name name="secondaryMcAddress"></Name>
            <Name name="primaryMcPort">29599</Name>
            <Name name="secondaryMcPort">-1</Name>
            <Name name="myAddress">127.0.0.1</Name>
            <Name name="appName">CAMM-Local-Controller</Name>
            <Name name="appShortName">LC</Name>
            <Name name="lcPort">29598</Name>
            <Name name="heartbeatFrequency">15</Name>
            <Name name="heartbeatTimeout">180</Name>
      </Names>
      <Paths>
            <Path name="dsLocalConfig">${basedir}/DS/LocalConfig-ds.xml</Path>
      </Paths>
</LocalConfig>
```

Sample LocalConfig-lc.xml file (hidden logging and cleanups configuration)

```
    <Logging>
        <LogLevel>INFO</LogLevel>
        <LogDirectory>${logbase}</LogDirectory>
        <ObjectLogging>
          <ObjectToLog>
            <ObjectName>com.torokina.tim.config</ObjectName>
            <ObjectLogLevel>TRACE</ObjectLogLevel>
        </ObjectToLog>
        </ObjectLogging>
    </Logging>
    <CleanUps>
        <CleanUp>
            <CleanUpName>clean-temporary-directory</CleanUpName>
            <CleanUpAction>delete</CleanUpAction>
            <CleanUpTarget>${tmp}</CleanUpTarget>
            <Parameter>
                <ParameterName>expire</ParameterName>
                <ParameterValue>3d</ParameterValue>
            </Parameter>
        </CleanUp>
        <CleanUp>
            <CleanUpName>archive-log-directory</CleanUpName>
            <CleanUpAction>archive</CleanUpAction>
```

```
            <CleanUpTarget>${logbase}</CleanUpTarget>
            <Parameter>
                <ParameterName>expire</ParameterName>
                <ParameterValue>3d</ParameterValue>
            </Parameter>
        </CleanUp>
        <CleanUp>
            <CleanUpName>clean-log-directory</CleanUpName>
            <CleanUpAction>delete</CleanUpAction>
            <CleanUpTarget>${logbase}</CleanUpTarget>
            <Parameter>
                <ParameterName>expire</ParameterName>
                <ParameterValue>7d</ParameterValue>
            </Parameter>
        </CleanUp>
    </CleanUps>
```

Sample LocalConfig-lc.xml file (LocalController runtime)

```
<?xml version="1.0" encoding="UTF-8"?>
<Runtime>
   <Names>
      <Name name="mainClass">com.torokina.tim.lc.Main</Name>
      <Name name="primaryMcAddress">127.0.0.1</Name>
      <Name name="secondaryMcAddress"/>
      <Name name="primaryMcPort">29599</Name>
      <Name name="secondaryMcPort">-1</Name>
      <Name name="myAddress">127.0.0.1</Name>
      <Name name="appName">CAMM-Local-Controller</Name>
      <Name name="appShortName">LC</Name>
      <Name name="lcPort">29598</Name>
      <Name name="heartbeatFrequency">15</Name>
      <Name name="heartbeatTimeout">180</Name>
      <Name name="mcPort">29599</Name>
      <Name name="manageable">996</Name>
   </Names>
   <Paths>
      <Path name="dsLocalConfig">${basedir}/DS/LocalConfig-ds.xml</Path>
      <Path name="apphome">${tim.base}/${appShortName}</Path>
      <Path name="runtimeConfig">${apphome}/runtime.xml</Path>
      <Path name="tmp">${apphome}/tmp</Path>
      <Path name="logbase">${apphome}/logs</Path>
      <Path name="basedir">${tim.base}</Path>
   </Paths>
</Runtime>
```

LocalController Configuration Options

Use the following options to configure LocalController.

- **Path**
  Lets you specify the path information for the following items:
  – **basedir**

Specifies the DX NetOps base directory.
  - **apphome**
    Specifies the DX NetOps LocalController application home directory.
  - **tmp**
    Specifies the DX NetOps LocalController temporary files directory.
  - **logbase**
    Specifies the DX NetOps LocalController log directory.
  - **runtimeConfig**
    Specifies the DX NetOps LocalController runtime XML configuration file that the Generic Executor provides.
- **Java**
  Lets you specify options for using Java.
  - **CommandPath**
    Specifies the full path to the Java executer that the Generic Executor calls to start the LocalController.
  - **ClassPath/JarBase**
    Lets you create a list with one or more entries to add to ClassPath.
  - **Options**
    Specifies the command-line options parsed to Java.
  - **Environment**
    Specifies the environment variables that execute the MultiController component.
  - **MainClass/Class**
    Specifies the main Java class to execute.
  - **MainClass/Args**
    Specifies the arguments parsed to the Java class.
- **Runtime**
  Lets you specify runtime options.
  - **Binding/BindAddress**
    Specifies the IP address to which the LocalController component binds.
    Value: Use IP address 0.0.0.0 for all LocalController components. For two or more IP addresses, use a comma-separated list.
  - **Binding/MyAddress**
    Specifies the IP address that the LocalController uses to identify itself.

    NOTE
    This IP address must be a valid IP address on this host.
  - **Binding/BindPort**
    Specifies the IP address that the LocalController uses to identify itself.

    NOTE
    This IP address must be a valid IP address on this host. By default, the LocalController binds to TCP port 29598.
  - **LocalControllerConfig/Mode**
    Specifies the operating mode of the LocalController.
    Value: Specify either Active or Standby.
  - **LocalControllerConfig/MCAddresses/Primary**
    Specifies the IP address of the primary MultiController in the cluster.
  - **LocalControllerConfig/MCAddresses/Secondary**
    Specifies the IP address of the secondary MultiController in the cluster.
  - **LocalControllerConfig/Heartbeat/ParameterName == frequency**
    Specifies the frequency of the heartbeat messages sent to the MultiController.
  - **LocalControllerConfig/Heartbeat/ParameterName == timeout**

Specifies how long this LocalController waits for the heart beats from subcomponents (Engines and Presenters). If the LocalController does not receive a heartbeat from a subcomponent for 180 seconds, the LocalController restarts.

- **Logging**
  Lets you specify logging options.
  - **LogLevel**
    Specifies the logging level in the output log files.
    Value: Specify DEBUG, TRACE, INFO, WARNING, or ERROR.
  - **LogDirectory**
    Specifies the logging level in the output log files.
    Value: Specify DEBUG, TRACE, INFO, WARNING, or ERROR.
  - **ObjectLogging/ObjectToLog/ObjectName**
    Specifies the Java class name on which to enable logging.
  - **ObjectLogging/ObjectToLog/ObjectLogLevel**
    Specifies the logging level for a Java class.
- **Cleanup**
  Lets you specify cleanup options.
  - **CleanUpName**
    Specifies the descriptive name of the cleanup.
  - **CleanUpAction**
    Specifies the cleanup action.
    Value: Specify Delete or Archive.
  - **CleanupTarget**
    Specifies the directory to clean up.
    Value: Can specify using a DX NetOps variable such as ${camm.variable}.
  - **Parameter/ParameterName - Parameter/ParameterValue**
    Specifies a parameter to expire and its expiration time in the format of <n><unit>. For example:
    10d = 10 days
    10h = 10 hours
    10m = 10 minutes

Start and Stop the LocalController Manually

The init.camm script automatically starts the LocalController process. You can use the cammCtrl utility to manually stop or start the LocalController component independently.

Follow these steps:

1. Log in as the CAMM_USER and go to the CAMM home directory.
2. Execute the following command to start the LocalController:
   ```
   /opt/CA/CAMM# tools/startall  - c lc
   ```
3. Execute the following command to stop the LocalController:
   ```
   /opt/CA/CAMM # tools/stopall  - c lc
   ```

Engine and Presenter Configuration

The installation and configuration of the Engine and the Presenter is achieved using the Device Pack installation program provided with each device pack.

High Availability Configuration

High availability refers to a system or component that is continuously operational for a desirably long time and ensures operational continuity. For DX NetOps to be highly available, the main components must be operational always. This is achieved by the failover processing. This section deals with the failover processing and the configuration settings for the high availability.

Failover is a backup operational mode available in a multi-server architecture. In failover, when a primary component becomes unavailable because of failure or scheduled down time, the secondary components assume the functions of the primary components. In DX NetOps, there are three types of failover operations:

- MultiController Failover
- LocalController Failover
- Subcomponent Failover

MultiController Failure

In a multiserver architecture, when the secondary MultiController is started, it sends heartbeat messages to the primary MultiController and also periodically synchronizes the secondary MultiController repository with that of the primary MultiController. When the heartbeat to the primary MultiController fails, the secondary MultiController assumes that the primary MultiController is down. The secondary MultiController waits till the timeout period expires. If the primary MultiController did not respond after the timeout period, secondary MultiController assumes the role of the primary. When the primary MultiController resumes its function, the secondary MultiController sends all the information back to the primary MultiController.

Failover Settings

The parameters for the failover settings are in the LocalConfig-mc.xml file. If needed, change the following default values:

heartbeatFrequency

The time period in which the secondary MultiController sends the heartbeat messages to the primary MultiController. If the heartbeat messages to the primary MultiController is failed for more than six consecutive times, the secondary MultiController takes the active role.
<Name name="heartbeatFrequency">30</Name>
Default: 30 seconds

repositoryFrequency
The time period in which the secondary MultiController synchronizes its repository with the primary MultiController.
<Name name="repositoryFrequency">15</Name>
Default: 15 seconds

When the primary MultiController fails, the LocalController contacts the secondary MultiController according to the following failover settings in LocalConfig-lc.xml file. If needed, change the following default values:

heartbeatFrequency
The time period by which the LocalController sends the heartbeat messages to the primary MultiController.
<Name name="heartbeatFrequency">30</Name>
Default: 30 seconds

heartbeatTimeout
The time period by which the LocalController waits to send heartbeat to the primary MultiController. If the LocalController cannot send the heartbeat message to MultiController for 600 seconds, the LocalController switches to the standby mode. Note: In the standby mode, a LocalController does not have any Subcomponent running.
<Name name="heartbeatTimeout">600</Name>
Default: 600 seconds

LocalController Failure

A LocalController sends heartbeat messages to the MultiController. If the MultiController did not receive heartbeat from the LocalController, missed heartbeat is logged. If the missed heartbeat time exceeds the timeout period, the LocalController is placed in the expiry list and MultiController triggers the first standby LocalController.

> **NOTE**
>
> MultiController runs expiry check for LocalControllers every 10 seconds. If any LocalController is found in the expiry list, MultiController moves its repository to the standby LocalController present in the environment. If no standby LocalController is available, MultiController does not perform the swap operation.

In the case of LocalController failover, the LocalController where Sub Components were running comes back, but the Subcomponents do not switch back.

Failover Settings

The parameters for the failover settings are in the LocalConfig-lc.xml file. If needed, change the following default value:

heartbeatTimeout
The time MC waits for the heartbeat message from a LC.
<Name name="heartbeatTimeout">600</Name>
Default: 600 seconds

Subcomponent Failure

A LocalController monitors its running subcomponents (Engine and Presenter). If LocalController does not receive a response from a Subcomponent for 10 seconds (default), it restarts the Subcomponent.

To change the default failover settings, contact CA Support.

Log Files Configuration

The following sections describe how to configure log files in DX NetOps.

Configure the logging properties for all the DX NetOps components using the logging.properties file, which is created in the default log directory when the component is started. The logging.properties file is preconfigured to generate log files in the default log directory. However, you can edit the logging.properties file to redirect the log files in to another directory. Once the logging.properties file is modified, restart the component to load the modified logging properties. All logs (application logs and STD-ERROR/STD-OUTPUT logs) are generated in the directory that is specified in the logging.properties file.

logging.properties File - Examples by Component

The following examples describe the different logging.properties files:

MultiController example logging.properties file

```
com.torokina.common.logging.apache.FileHandler.directory=/opt/CA/CAMM/MC/logs
handlers=com.torokina.common.logging.apache.FileHandler
com.torokina.common.logging.apache.FileHandler.level=INFO
.level=INFO
com.torokina.common.logging.apache.FileHandler.prefix=CAMM-Multi-Controller-
```

LocalController example logging.properties file

```
com.torokina.common.logging.apache.FileHandler.directory=/opt/CA/CAMM/LC/logs
```

```
handlers=com.torokina.common.logging.apache.FileHandler
com.torokina.common.logging.apache.FileHandler.level=INFO
.level=INFO
com.torokina.common.logging.apache.FileHandler.prefix=CAMM-Local-Controller-
```

Delivery System example logging.properties file

```
com.torokina.common.logging.apache.FileHandler.directory=/opt/CA/CAMM/DS/logs
handlers=com.torokina.common.logging.apache.FileHandler
com.torokina.common.logging.apache.FileHandler.level=INFO
.level=INFO
com.torokina.common.logging.apache.FileHandler.prefix=CAMM-Delivery-System-
```

ENGINE_CAMM example logging.properties file

```
com.torokina.common.logging.apache.FileHandler.directory=/opt/CA/CAMM/COMPONENTS/ENGINE_CAMM/logs
handlers=com.torokina.common.logging.apache.FileHandler
com.torokina.common.logging.apache.FileHandler.level=INFO
.level=INFO
com.torokina.common.logging.apache.FileHandler.prefix=CAMM-ENGINE_CAMM-
```

Generic Executor example logging.properties file

By default the Generic Executor does not create a logging.properties file in its log directory. All logs are generated in the ~GE/logs directory. The following example shows how you can create the logging.properties file in the logs directory to redirect the Generic Executor logs (non-Windows platforms only):

```
#Properties for Logger
#Tue May 07 04:08:45 EDT 2013
com.torokina.common.logging.apache.FileHandler.directory=/opt/CA/CAMM/GE/logs
handlers=com.torokina.common.logging.apache.FileHandler
com.torokina.common.logging.apache.FileHandler.level=INFO
.level=INFO
com.torokina.common.logging.apache.FileHandler.prefix=CAMM-Generic-Executor-
```

Configuring Log File Cleanup

By default the cleanup action for each component is configured to run on the *logbase* directory. If the log files are redirected to another directory, modify the cleanup configuration for successful *Archive* or *Delete* actions.

Define cleanup actions in the *LocalConfig* xml-element in the configuration files. For the DX NetOps components, define the cleanup actions in the respective configuration files:

- **MultiController:** LocalConfig-mc.xml
- **LocalController:** LocalConfig-lc.xml
- **Delivery Service:** LocalConfig-ds.xml
- **Subcomponents (ENGINE or PRESENTER):** TemplateConfig-subcomponent.xml

> **NOTE**
> You can find the TemplateConfig-subcomponent.xml file in the <camm.base>/LC directory.

The following examples describe two sample cleanup configurations.

Example: Cleanup configuration file (*Delete* action) for the Delivery Service component

```
<LocalConfig>
<Description>Configuration for Delivery Module</Description>
 ...
```

```
...
<CleanUps>
 <!-- SAMPLE  DELETE  ACTION -->
 <CleanUp>
  <CleanUpName>Delete</CleanUpName>
  <CleanUpAction>delete</CleanUpAction>
  <CleanUpTarget>${apphome}/.local</CleanUpTarget> <!-- Directory Name -->
  <Parameter>
   <ParameterName>expire</ParameterName>
   <ParameterValue>7d</ParameterValue> <!-- 1y0m3d1h -->
  </Parameter>
  <Parameter>
   <ParameterName>includeDir</ParameterName>
   <ParameterValue>true</ParameterValue><!-- true/false -->
  </Parameter>
  <Parameter>
   <ParameterName>recursive</ParameterName>
   <ParameterValue>true</ParameterValue><!-- true/false -->
  </Parameter>
  <Parameter>
   <ParameterName>match</ParameterName>
   <ParameterValue>^[\d]+\.xml$</ParameterValue>
   <!-- Regular Pattern -->
  </Parameter>
 </CleanUp>
```

Example: Cleanup configuration file (*Archive* action) for the Delivery Service component

```
 <!-- SAMPLE  ARCHIVE  ACTION -->
 <CleanUp>
  <CleanUpName>Archive</CleanUpName>
  <CleanUpAction>archive</CleanUpAction>
  <CleanUpTarget>${logbase}</CleanUpTarget> <!-- Directory Name -->
  <Parameter>
   <ParameterName>expire</ParameterName>
   <ParameterValue>7d</ParameterValue> <!-- 1y0m3d1h -->
  </Parameter>
  <Parameter>
   <ParameterName>includeDir</ParameterName>
   <ParameterValue>true</ParameterValue><!-- true/false -->
  </Parameter>
  <Parameter>
   <ParameterName>recursive</ParameterName>
   <ParameterValue>true</ParameterValue><!-- true/false -->
  </Parameter>
  <Parameter>
   <ParameterName>match</ParameterName>
   <ParameterValue>CAMM-.*\.log</ParameterValue>
   <!-- Regular Pattern -->
  </Parameter>
  <Parameter>
   <ParameterName>achiveHome</ParameterName>
   <ParameterValue>${logbase}</ParameterValue> <!-- folder path -->
  </Parameter>
```

```
    <Parameter>
     <ParameterName>achivePrefix</ParameterName>
     <ParameterValue>Archive-</ParameterValue> <!-- prefix string -->
    </Parameter>
    <Parameter>
     <ParameterName>achiveSuffix</ParameterName>
     <ParameterValue>.zip</ParameterValue> <!-- suffix string -->
    </Parameter>
   </CleanUp>
  </CleanUps>
 <LocalConfig>
```

You can include any of the cleanup actions in the component configuration file. The Generic Executor performs all the cleanup actions. The Generic Executor stores cleanup files in the {camm.base }/GE_<User>/cleanup directory, where camm.base is the DX NetOps installation directory. Once the component configuration file is modified, restart the related component so that the modified configurations can take effect.

EMS Integration Profiles Configuration

EMS integration profiles specify how EMS inventory discovery operates in your Data Aggregator environment.

With EMS integration profiles, you specify the status, data collector, device pack, EMS IP, and Backup EMS IP. Specify one IP domain for each EMS integration profile you create. The IP domain that you specify is for the target EMS server, which manages multiple devices (typically 1,000 devices at a time). Data Aggregator processes the inventory data from the EMS server constantly and all at once. (With SNMP or ICMP, polling is done device-by-device.)

NOTE

- After you install the device pack, the EMS Integration Profiles option becomes visible in the user interface.
- Do not add the same EMS integration profile more than once for the same Data Collector.

Add EMS Integration Profiles

You can create EMS integration profiles to specify how EMS inventory discovery operates in your Data Aggregator environment.

NOTE
You must be logged in as an administrator to perform this task.

Creating EMS integration profiles without first setting the scope to a tenant puts the profile in the global space, accessible by all tenants. Running a discovery using a profile in the global space lets anyone see the discovery results, whether they set the scope to a tenant.

Therefore, set the scope to a tenant *before* you create an EMS integration profile to make that profile accessible only to a specific tenant. After you set the scope to a tenant, the tenant indicator appears at the top right of the page. You can then manually synchronize the tenant with CA Infrastructure Management, or wait for the automatic synchronization to occur. You cannot create the EMS integration profile until the tenant is synchronized with Data Aggregator.

NOTE
For more information about setting the scope to a tenant and synchronizing a tenant, see the *CA Performance Center Administrator Guide*.

Follow these steps:

1. Select Admin, Data Source Settings, and click a Data Aggregator data source in the CA Performance Center user interface.
2. Click EMS Integration Profiles from the Monitoring Configuration menu.

The EMS Integration Profiles page opens, displaying a list of available discovery profiles.

3. Click New.
   The Add EMS Discovery Profile dialog opens.
4. Enter the required information in the fields. The configuration fields that display depend on the device pack you select. Each device pack has unique global variables that you configure.

   NOTE
   For information about unique global variables for each product, see the CA Support site.
5. Click Save.
   The EMS integration profile is created.
   The inventory discovery does *not* run automatically when you click Save and the Enabled option is selected. The inventory discovery only runs when one of the following conditions is met:
   - The inventory poll schedule is reached.
   - The EMS integration profile is manually started.

   NOTE
   To edit EMS integration profile information, select an EMS integration profile in the Add EMS Discovery Profile dialog, click Edit, and update the information in the dialog. However, if you edit the information in the Inventory_Poll_Rate and Performance_Poll_Rate fields, the changes are not applied. If there is a change in the information for these two fields, delete and recreate the EMS integration profile.

Start EMS Discovery Manually

EMS Integration Profiles discover devices and their components in your network. You can manually start an EMS Integration profile to begin discovery.

NOTE
Alternatively, you can wait until the inventory poll schedule is reached for discovery to begin automatically.

Follow these steps:

1. Select Admin, Data Source Settings, and click a Data Aggregator data source in the CA Performance Center user interface.
2. Click EMS Integration Profiles from the Monitoring Configuration menu.
   The EMS Integration Profiles page opens, displaying a list of available discovery profiles.
3. Select one or more EMS integration profiles that you want to run a discovery on, and click Start.

   NOTE
   You can only run a discovery on an EMS integration profile that has a status of 'READY'.

   A confirmation dialog opens.
4. Click Yes.
   The Discovery starts. The Status column for the selected Discovery profiles indicates 'Started'.
   A confirmation dialog opens.
5. Click OK.
   Devices and all of their associated interfaces are discovered and polling begins. You are returned to the EMS Integration Profiles page.
   If discovery hangs for more than 10 minutes, Data Aggregator stops it. Data Aggregator considers a discovery to be hanging when no new devices are discovered within 10 minutes *and* the state for the selected discovery profiles have not changed within 10 minutes. An audit event is generated on the discovery instance item. If no devices were discovered successfully, the Status column for the selected discovery profiles indicates 'FAILURE'. If at least one device but not all devices were discovered successfully, the Status column indicates 'PARTIAL_FAILURE'.
   The discovered devices and components can take up to 5 minutes to synchronize with CA Performance Center. When the synchronization is complete, the discovered devices and components appear in the Inventory tab in CA Performance Center.

View EMS Discovery Results

You can view a summary of the number of all manageable EMS devices that were discovered.

Follow these steps:

1. Select Admin, Data Source Settings, and click a Data Aggregator data source in the CA Performance Center user interface.
2. Click EMS Integration Profiles from the Monitoring Configuration menu.
   The EMS Integration Profiles page opens, displaying a list of available discovery profiles.
3. Select an EMS integration profile instance for which you want to view discovery results, and click History.
   The EMS History results display as follows:
   - The Device table shows the monitored devices and time of creation for each.
   - The Element table shows the monitored interfaces and time of creation for each.

Start or Stop EMS Discovery Services

"Start" services are used for discovery on the EMS servers to review the inventory continuously. Although discovery can be scheduled, you can manually start, stop, or restart services on demand. For example, you can restart discovery after an EMS server has been down and brought back up, or after upgrading a device pack installation.

Stopping a service deletes any inactive data poll but waits for any active data polling to complete without any interruption. No EMS files are deleted. This action also disables any new polling from occurring as long as it is in the stopped state. The service remains in the stopped state until you start it again.

Restarting Data Collector has no effect on the status of any EMS integration profile.

> **NOTE**
> You must be logged in as an administrator to perform this task.

Follow these steps:

1. Select Admin, Data Source Settings, and click a Data Aggregator data source in the CA Performance Center user interface.
2. Click EMS Integration Profiles from the Monitoring Configuration menu.
   The EMS Integration Profiles page opens, displaying a list of available discovery profiles.
3. Select a profile, and click Start or Stop.
   A confirmation dialog opens.
4. Click Yes to confirm the action.
   The service starts or stops, depending on your choice. This service remains in the started or stopped state until you change it manually.

Add Event Rules

Event rules can be added using the Data Aggregator monitoring profiles dialog.

> **NOTE**
> For more information, see the *Data Aggregator Administrator Guide* or online help.

Administrating

This section deals with the migrating, monitoring, and customizing device packs.

Self-Monitoring Device Pack

Self-monitoring device pack monitors the state of its own components and the existing device packs.

Self-Monitoring Device Pack in DX NetOps Mediation Manager for Performance Management

NOTE
The information in this section applies only to DX NetOps for Performance Management 2.3.4. If you are using DX NetOps for Infrastructure Management 2.3.3 or the earlier releases, contact CA Support and request a Program Temporary Fix (PTF) to install the Self-Monitoring Device Pack. For example, if you are using DX NetOps for Infrastructure Management 2.3.2, request for PTF_2.3.2_09.

Self-monitoring device pack can monitor the state of its own Engine and the Engine of the existing device packs. To install the self-monitoring device pack in Data Collector for DX NetOps Mediation Manager for Performance Management, follow the procedure in the Installing section and select Engine_CAMM_IM_SelfMonitoring.zip to install the Engine. To monitor the device packs in multiple Data Collectors, install Self-Monitoring Device Pack in all Data Collectors.

The following types of reports in CA Performance Center are used to monitor the status of the Engine:

- Reports for All Engines
- Reports for the Selected Engine

Reports for All Engines

The following reports are available to monitor the status of all Engines.

- **JVM Memory Usage**
  Indicates the percentage of the average Java virtual machine (JVM) memory usage by all Engines.
- **Number of Threads used per Poll**
  Indicates the number of threads that all Engines use in a poll.
- **Number of Engines**
  Indicates the total number of Engines on a Data Collector that the Self-Monitoring Device Pack monitors.
- **Number of Devices Polled**
  Indicates the total number of devices that all Engines poll.

Reports for a Selected Engine

The following reports are available to monitor the status of a selected Engine:

- **JVM Memory Usage**
  Indicates the percentage of the JVM memory usage by the selected Engine
- **Number of Devices Polled**
  Indicates the total number of devices that the selected Engine polls.
- **Number of Threads used per Poll**
  Indicates the total number of threads that the selected Engine uses.
- **Files Processed during Conversion and Calculation**
  Indicates the total number of processed files, files that are failed, and successfully processed files during the Conversion and Calculation phase.
- **Files Processed during Filter**
  Indicates the total number of processed files, processed files that are failed, and successfully processed files during the Filter phase.
- **Processed Files to be Delivered to Data Aggregator**
  Indicates the total number of processed files, successfully processed files, and processed files that are not delivered to Data Aggregator.
- **Time Taken in each Phase**
  Indicates the time that is taken (in seconds) by the Engine during each phase. The Engine processes the data in the following phases sequentially:

- **Data acquisition:** The Engine connects to the Element Management Server (EMS) or the device to gather raw data by using protocols, such as SFTP, FTP, COPY, and HTTP. Data that are collected is written in a text file and processed in the next phase.
- **Conversion and Calculation:** The Engine converts the received file to the DX NetOps MM XML format. From each file, the Engine identifies key elements, such as Metric Family, Metrics, Device, Device IP, and Delta time.
- **Filter:** If required, the Engine performs filtering actions, such as removal of unwanted device or unwanted metric family. After the filtering process, the Engine generates a new file in same format for every input file.
- **Delivery system:** The files that are collected from previous phase are sorted by timestamp and sent to Data Aggregator for further processing.

  NOTE

  The processed files in the last phase are deleted if the Log level is not in the DEBUG mode.

- **Time Taken to Complete a Poll**
  Indicates the time that is taken (in seconds) by the selected Engine to complete a poll.
- **Size of Processed Files per Poll**
  Indicates the total size (in KB) of the processed files per poll.

The following custom dashboard reports are available to monitor the status of the selected Engine:

- **Number of Polled Items**
  Indicates the number of polled device components in the selected Engine.
- **Number of Metrics Polled**
  Indicates the number of metrics polled by the selected Engine.
- **Metrics Polled per Second**
  Indicates the number of metrics polled per second.
- **Average Size of Processed Files per Poll**
  Indicates the average size (in KB) of the processed files per poll.

Self-Monitoring Device Pack in CA Mediation Manager

Similar to a device pack, the self-monitoring device pack in CA Mediation Manager consists of Engine, Presenter, and Cert. Use the self-monitoring device pack to monitor the state of the LocalController and the components of device packs (Engine and Presenter). To install the self-monitoring device pack, follow the procedure in the Installing section and install ENGINE_CAMM, PRESENTER_CAMM, and CERT_CAMM. To monitor the device packs in multiple LocalController, install self-monitoring device pack in the corresponding LocalController.

The following types of reports in CA eHealth are used to monitor the status of a device pack:

- Reports for LocalController
- Reports for the Selected Engine
- Reports for the Selected Presenter

Reports for LocalController

The following reports are available to monitor the status of a LocalController:

EngineRSPercentIndicates (in percentage) the running status of the Engine.

EnginePollSpercentIndicates (in percentage) the last successful polling by the Engine.

TotalEngineMemoryUsageIndicates the total memory usage by all Engines

TotalEnginesIndicates the number of Engines that the self-monitoring device pack monitors.

TotalDevicesIndicates the number of devices that are being polled by all Engines.

PeakEngineCPUUsageIndicates the peak CPU usage among the Engines.

TotalPresentersIndicates the number of Presenters that the self-monitoring device pack monitors.

EngineDASPercentIndicates (in percentage) the successful data acquisition by the Engine.

PresenterRSPercentIndicates (in percentage) the running status of the Presenter.

Reports for Selected Engine

The following reports are available to monitor the status of the selected Engine:

pollDuration
Indicates the time taken by the selected Engine for the specific poll.

State
Indicates the state of the data delivered by the Engine to Delivery Service.

laststatus
Indicates the last poll status of the selected Engine.

Memory Usage
Indicates the memory usage (in percentage) by the selected Engine.

CPU Usage
Indicates the CPU usage (in percentage) by the selected Engine.

DAStatus
Indicates the data acquisition status by the selected Engine.

maxThreadNum
Indicates the number of threads the selected Engine uses.

deviceCount
Indicates the number of devices that the Engine monitors.

Reports for the Selected Presenter

The following reports are available to monitor the selected Presenter:

State
Indicates the state of the Presenter that the self-monitoring device pack monitors.

Memory Usage
Indicates the memory usage (in percentage) by the Presenter.

CPU Usage
Indicates the CPU usage (in percentage) by the Presenter.

avgProcessTime
Indicates the time taken by the selected Presenter to process the data.

maxThreadNum
Indicates the number of threads the selected Presenter uses.

avgQueueTime
Indicates the waiting time for the data to be processed by the selected Presenter.

Average Queue Size
Indicates the queue size for the selected Presenter.

How to Customize a Device Pack

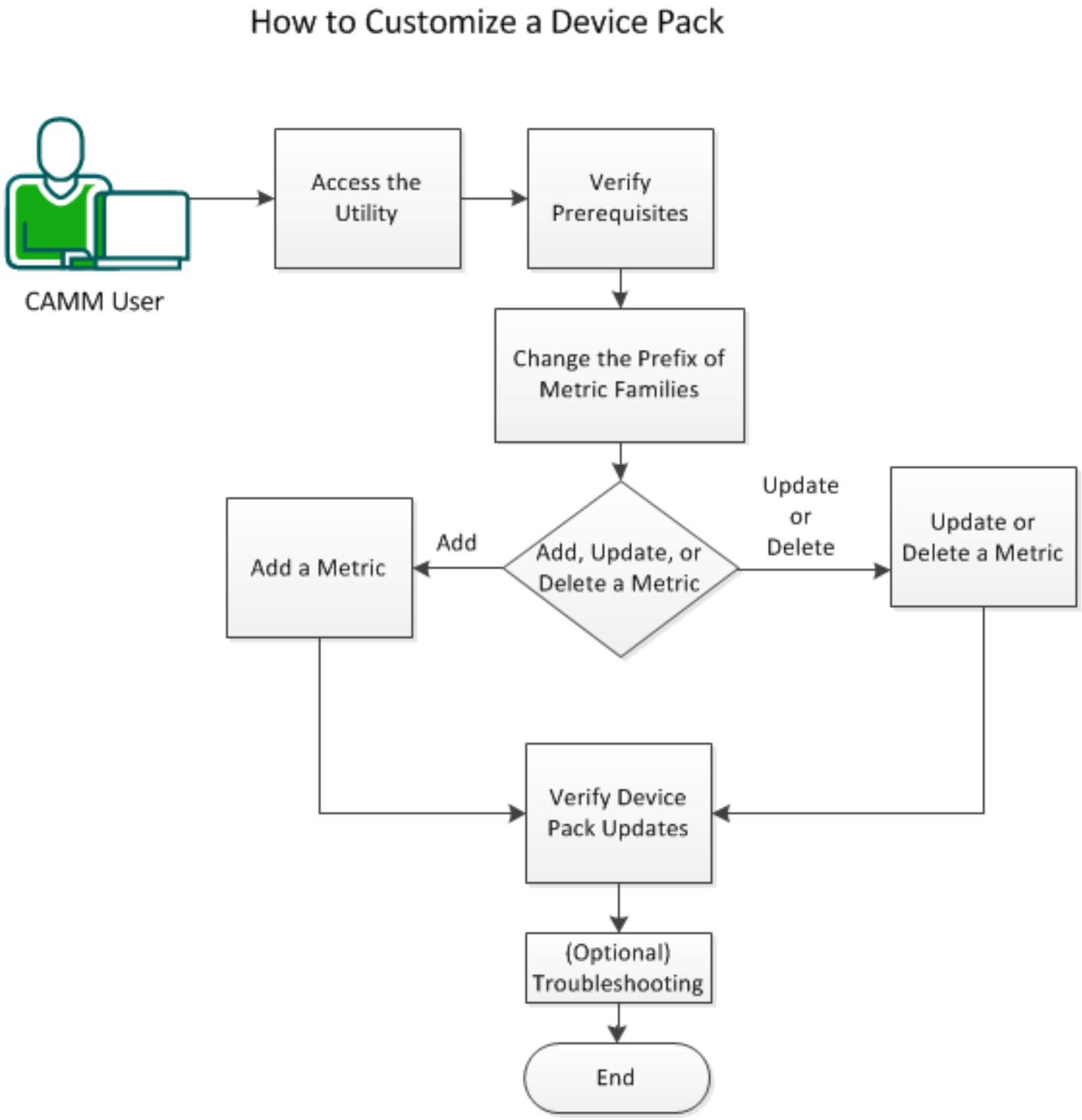
When a device is upgraded with a new metric, Data Aggregator cannot recognize the new metric by default. Therefore, in CA Performance Center, you cannot generate a report for the new metric. Additionally, when a metric is updated or deleted in a device, you cannot generate the metric-based report in CA Performance Center.

You can use the Device Pack Customization tool to customize the vendor certification and the metric family of the device by adding, deleting, or updating a metric of a device pack. You can add a prefix to a metric family name of a device pack. You can also create a key performance indicator (KPI) metric to derive new metrics from the existing metrics. If the device pack was upgraded, you can pull and merge the previous customization with the upgraded device pack.

Data Aggregator recognizes the new or updated metric and you can generate the metric-based report in CA Performance Center.

The following diagram shows the work flow for this task:

Figure 92: How to customize a device pack



Install the Device Pack Customization Tool

Install the Device Pack Customization tool in CA Performance Center. The installation process supports only the console mode.

Follow these steps:

1. Log in to the host where CA Performance Center is installed.
2. Copy the installation file, DevicePackCustomizationTool_<*version*>.zip, to a folder in CA Performance Center and unzip it.
3. Execute **java -jar dptool-installer-bin.jar**.
4. Press Enter repeatedly to scroll through the license agreement. At the end of the agreement, type **Yes** and press Enter to accept the license agreement.
5. Provide the CA Performance Center user name, password, and HTTP port number.
   Default: admin, admin, 8181
6. Delete the following files from the specified location:
   – dpcustomizationProcessor.jar in the <dataaggregator> /opt/IMDataAggregator/apache-karaf-2.x.y/deploy folder in Data Aggregator
   – dpcustomizationserver.war in the /opt/CA/PerformanceCenter/PC/webapps folder in CA Performance Center.
7. Type **Yes** to confirm the deletion of files.
   The device pack customization tool is successfully installed.
8. Access the tool from the following URL:
   ```
   http://<PC Hostname>:<PC Port#>/dpcustomization/login.htm
   ```
   Where <*PC Hostname*> is the CA Performance Center hostname and <*PC Port#*> is the port number that is configured for CA Performance Center.

Verify Prerequisites

The prerequisites for adding a new metric using the Device Pack Customization Tool are as follows:

- The Engine polls the device correctly.
- The Engine recognizes the new metric.

Check if the Engine Polls the Device Correctly

Make sure that the Engine polls the device correctly.

Follow these steps:

1. Log in to Data Collector.
2. Change the working directory to $DC_INSTALL/apache-karaf-2.3.0/data/log.
   $DC_INSTALL is the path where Data Collector is installed. This defaults to /opt/IMDataCollector
3. Execute the following command:
   ```
   grep ERROR ems.log
   ```
   If the Engine polls the device correctly, output has no error.

Check if the Engine Recognizes the New Metric

After confirming that the Engine polls the device correctly, check if the Engine recognizes the new metric. This section is applicable only if you use CA Performance Management release 2.3.3 or earlier.

Follow these steps:

1. Log in to Data Collector.

2. Open the org.ops4j.pax.logging.cfg file from $DC_INSTALL/apache-karaf-2.3.0/etc/ where $DC_INSTALL is the path where Data Collector is installed. This defaults to /opt/IMDataCollector.
3. Search for “# EMS Integration logging”.
4. Replace INFO to DEBUG only in the following lines:
   log4j.logger.com.ca.im.dm.mediation=INFO, ems
   log4j.logger.com.ca.im.dm.ems=INFO, ems
   log4j.logger.com.ca.im.data-mgmt.common.ems=INFO, ems
   log4j.logger.com.ca.im.data-collection-manager=INFO, ems
   log4j.logger.com.torokina=INFO, ems

   NOTE
   This change allows the Engine to dump the XML files that have information about the device and polled metrics.
5. Wait for the next performance poll.
6. After the next performance poll, change the working directory to:
   $DC_INSTALL/apache-karaf-2.3.0/MediationCenter/Queue.Pol/queue-<DevicePackName>.
7. Execute the following command:
   ```
   grep  - r <metric-name> .
   ```
   If the Engine recognizes the new metric, for each occurrence of <metric-name> from the XML files in the working directory, the <metric-name> is returned.

   WARNING
   You cannot execute the command if you miss the period at the end of the command.
8. Revert the changes in step 4.

   WARNING
   If you do not revert the changes in step 4, the Engine does not delete the temporary files and the disk may run out of space.

Merge Old Customizations

After you update a device pack, use the Merge option to absorb the old device pack customization to the updated device pack.

Follow these steps:

1. Log in to the Device Pack Customization tool.
2. Click File, Open, and select the device pack that was updated.
3. Click Merge.

Merge Old Customizations

After you update a device pack, use the Merge option to absorb the old device pack customization to the updated device pack.

Follow these steps:

1. Log in to the Device Pack Customization tool.
2. Click File, Open, and select the device pack that was updated.
3. Click Merge.

Customize a Device Pack

You can customize a device pack by performing one or both the following activities:

- Add or modify the prefix of a metric family.
- Add, update, or delete a metric in a metric family.

NOTE

- Whenever you update to the latest version of CA Performance Management, close the current session of Device Pack Customization Tool and launch the session again.
- Customization for the following device packs are supported, only if you use CA Performance Management 2.4 and above:
- – CAEXFOBrixWorx
- – CAAlcatelXMSIMS
- – CACamiantMPE
- – CAEricssonMME
- – CAEricssonSGW
- – CANokiaIMS
- – CANetAppFiler

Follow these steps:

1. Log in to the Device Pack Customization tool.
2. Click File, Open, and select a device pack to customize.
3. Click Customize.
   The metric families of the selected device pack appear in a tabbed page. Now, you can customize a metric family prefix or customize a metric.

Add or Update a Metric to a Device Pack

You can add or update a metric to a metric family of a device pack.

Follow these steps:

1. Click File, Open, and select a device pack.
2. Click Customize.
3. Double-click a metric family from the panel on the left-side.
   The metrics of the selected metric family appear in a tabbed page.

   NOTE
   - Hover the mouse over the column headers to view the corresponding tooltip. - Click the down arrow in Metric Name, Columns, and select or clear the check boxes to display or hide specific columns.
4. (Only to add a metric) Click Add, type the values for the following parameters and, and click Add:
   Name
   Specifies the metric name.

   WARNING

   See Step 7 in Check if the Engine Recognizes the New Metric. Provide the metric name that the Engine recognized. For a KPI metric, provide a unique name. The value can be a string consisting only of alphanumeric characters and _ (underscore) and must not be empty.

Documentation
Specifies the description of the metric.

5. (Only to update a metric) Double-click a row.
6. Provide the parameters in the fields of the following attributes:

Type
Specifies the supported data type of the metric.

Deltaed
Indicates whether Data Aggregator performs the delta calculation on the metric value.

Values: Yes, No

Effect of updating: When set to Yes, Data Aggregator performs the delta calculation by using the current and previous values of the metric.

Is dbColumn
Stores the value in the database table. IsDBColumn is used for metric attributes. Set the value to true when Polled is set to true.

Values: true, false

Effect of updating: If set to true, the data for this attribute or metric is stored in the database.

Baseline
Indicates whether to calculate a mean value for this attribute or not. If it is set to true, a corresponding BaselineList definition must be defined.

> **NOTE**
> Baseline requires that Standard Deviation is set to true.

Minimum
Indicates whether to calculate the minimum of this attribute during the rollup or not. Creates a 'min_' column in the database table.

Values: true, false

Effect of updating: If the value is set to true, the minimum value of the attribute is calculated and reported.

Maximum
Indicates whether to calculate the maximum of this attribute during the rollup or not. Creates a 'max_' column in the database table.

Values: true, false

Effect of updating: If the value is set to true, the maximum value of the attribute is calculated and reported.

Standard Deviation
Indicates whether to calculate the standard deviation of this attribute during the rollup or not. Creates a 'std_' column in the database table.

Values: true, false

Effect of updating: If the value is set to true, the standard deviation of the attribute is calculated and reported.

Variance
Indicates whether to calculate the variance of this attribute during the rollup or not. Creates a 'var_' column in the database table.

> **NOTE**
> This field is not supported for custom metric families.

Percentile
Indicates whether to calculate the nth percentile of this attribute during the rollup or not. Creates a 'pct_' column in the database table.

Values: 0 (zero), 95

Effect of updating: If the value is set to 95, the 95th percentile of the attribute is calculated and reported. Zero means that no calculation is performed, and the reporting field is not available.

Polled

Indicates whether the Data Aggregator collects the metric data during each poll cycle.

Values: true, false

Effect of updating: When set to True, Data Aggregator collects the metric data during each polling cycle. When set to False, the data is collected only upon discovery of an item.

Rollup Strategy

Specifies the operation that is performed every cycle during the rollup of the individually polled values. When Polled and IsDBcolumn are set to true, this parameter is required.

Values: Sum (a summation for counters), Avg (an average for gauges)

Effect of updating: The specified strategy is used to perform rollup calculations.

Units

Expresses the metric in reports by the specified standard of measurement.

KPI Expression

(Only for the KPI metric) Specifies the Key Performance Indicator (KPI) formula..

7. Click Update, Save.
   The metric appears in the metric family. The new and updated metric appears in a different color when you refresh the tool.

Provide KPI Expression

You can create a key performance indicator (KPI) metric to derive new metrics from the existing metrics. Currently, you can add a KPI metric only as a gauge. The Device Pack Customization tool supports only the addition, subtraction, and multiplication operators. Parentheses can be used in expressions. You can use the arithmetic operations to derive new metrics.

Example: (val1+val2)*5(val3)

> **NOTE**
> The Device Pack Customization tool does not support the division operator ("/" slash).

You can also use the SNMP functions to create a KPI metric. The Device Pack Customization Tool supports the following SNMP functions:

- **snmpProtectedDiv**
  Divides two values. Use the snmpProtectedDiv function instead of the division operator.
  Example: snmpProtectedDiv(OUTPUTERRORS, IFSPEED)
- **snmpMax**
  Compares two values and returns the maximum value.
  Example: snmpMax(val1, val2)
- **snmpRound**
  Rounds off the decimal value to the nearest integer.
  Example: snmpRound(IFSPEED).

Change the Prefix of Metric Families

Change the prefix of metric families of a device pack for an easier identification of the metric families during report generation.

Follow these steps:

1. Log in to the Device Pack Customization tool.
2. Click File, Open, and select a device pack.
3. Click Customize.

The metric families of the selected device pack appear in a tabbed page.
4. Click Edit.
5. Type the old prefix, if any.
6. Type the new prefix and click Update.
7. Click Save in the top-right corner of the Metric Families section.
   Example 1
   Old Prefix: <blank>
   New Prefix: CAZTMSC
   Effect of changing the prefix: CAZTMSC<*Metric Family Name*>
   Example 2
   Old Prefix: CAZTMSC
   New Prefix: ZTMSC<space>
   Effect of changing the prefix: ZTMSC <*Metric Family Name*>

NOTE

If you create reports-based on Select View Context in CA Performance Center, make sure that the metric family name with the prefix does not exceed the following limits:

- 128 characters in Infrastructure Management 2.3.1 and Infrastructure Management 2.3.3.
- 50 characters for Infrastructure Management 2.3.2.
- If the metric family name exceeds the character limit, the prefix is not added to any metric family name.

Delete a Metric in a Device Pack

You can delete metrics of a metric family in a device pack.

NOTE
Though you can delete a metric that is shipped with the device pack and associated with a report in CA Performance Center, you can view reports for the data that are collected before deleting the metric.

Follow these steps:

1. Click File, Open and select a device pack.
2. Click Customize.
3. Double-click a metric family from the panel on the left-side.
   The metrics of the selected metric family appear in a tabbed page.
4. Select a metric family.
5. Select a metric to delete and click Delete.
6. Click Save.

Verify Device Pack Updates

When you add, update, or delete a metric by using the Device Pack Customization Tool, the changes automatically appear in CA Performance Center. However, you can verify if the device pack is updated.

Follow these steps:

1. Access CA Performance Center from the following URL:
   http://<*PC Hostname*>:<*PC Port#*>
   Where <PC Hostname> is the CA Performance Center hostname and <PC Port#> is the port number configured for CA Performance Center.

2. Click Admin, Data Source Settings, Data Aggregator@<*DAHost Name*>, where <*DAHostname*> is the host name of <*caimda*>.
   The Monitored Devices page appears.
3. Click Monitoring Configuration, Metric Families.
   The Metric Families list appears.
4. Enter the device pack or Metric Family name in the Search View box at the bottom of the Metric Families list and press Enter.
   The metrics list appears.
5. Confirm if the metric is added, updated, or deleted.

Troubleshooting the Device Pack Customization Tool

The following information describes troubleshooting the Device Pack Customization Tool:

Metric families or metrics do not appear in the Device Pack Customization Tool

Symptom:

When I select a device pack in the tool, metric families or metrics are not listed.

Solution:

Capture the log and send it to Technical Support at http://ca.com/support.

New metric does not appear immediately in CA Performance Center

Symptom:

When I add a metric through the Device Pack Customization Tool, it does not appear immediately in CA Performance Center.

Solution:

Sometimes, a delay is experienced based on the refresh time. Wait for 15 minutes and if the problem persists, capture the log, and send it to Technical Support at http://ca.com/support.

Error 404

Symptom:

When I access the Device Pack Customization Tool, 'The Page cannot be found' message appears.

Reason:

The tool was not installed properly.

Solution:

Manually copy the jar file of the tool to the /opt/IMDataAggregator/apache-karaf-2.3.0/deploy directory in Data Aggregator. Also, make sure that Data Aggregator is not connected to CA Performance Center when the installer is running. If the problem persists, contact Technical Support at http://ca.com/support.

Uninstalling

This section provides steps to uninstall DX NetOps, uninstall a device pack in DX NetOps, and uninstall device packs in DX NetOps for Performance Management.

Uninstall DX NetOps Mediation Manager

The following procedure describes how to uninstall DX NetOps software.

Follow these steps:

1. (Optional) Stop the DX NetOps cluster using the following command:
   - Windows systems:
     ```
     %CAMM_HOME%/tools/stopall.bat
     ```
   - Unix systems:
     ```
     $CAMM_HOME/tools/stopall
     ```
2. (Optional) Wait for all DX NetOps components to stop.

   NOTE
   Uninstaller stops DX NetOps before it proceeds to remove DX NetOps.
3. Run the following command:
   - Windows systems:
     ```
     "%JAVA_HOME%/bin/java"  -jar
     %CAMM_HOME%/_CAMM_installation/uninstaller.jar
     ```
   - UNIX systems:
     ```
     "$JAVA_HOME/bin/java"  -jar
     $CAMM_HOME/_CAMM_installation/uninstaller.jar
     ```

   The Uninstall dialog appears.
4. Click Uninstall.
   The Uninstaller stops DX NetOps and starts the removal process. When the removal processing is complete, the Uninstall Complete window appears.
5. Click Done.
   DX NetOps is uninstalled.

Uninstall a Device Pack in DX NetOps Mediation Manager

A device pack can be removed from DX NetOps through the Web UI. You can remove both Engine and Presenter.

Follow these steps:

1. From the Web UI, select the Component (Engine or Presenter) that you want to uninstall.
2. Click Remove.
   The Remove Sub-Components dialog appears.
3. Select Yes.
   The device pack is uninstalled.

Uninstall a Device Pack in DX NetOps Mediation Manager for Performance Management

You can uninstall a device pack when you do not want to poll data for that device. This procedure completely removes the device pack from your system when followed sequentially.

NOTE
You cannot uninstall the device pack views. You can only reinstall or upgrade the device pack views.

Uninstall Device Packs in DX NetOps for CA Performance Management 2.3.4 and later

Remove Engine, CAPM Presenter, and Certification of device pack from DX NetOps through the Web UI.

Follow these steps:

1. Launch the web UI, click the Device Packs tab, select the subcomponent that you want to uninstall.
2. Click Remove.
   The Remove Sub-Components dialog appears.
3. Select Yes.

Frequently Asked Questions

Some of the frequently asked questions when working with DX NetOps or DX NetOps for Performance Management are as follows:

DX NetOps Mediation Manager

Contents

What are the default ports used by the DX NetOps components?

The default ports that are used by the components are as follows:

Component	Port	Type
MultiController	29599	TCP
LocalController	29598	TCP
Generic Executor	29560	TCP
Web Manager	8880	HTTP

Scripts to start, stop, and check the status of the components

How do I stop or start the services of all the components of DX NetOps?
Execute the following command to start the services of all components:

```
$CAMM_HOME/tools/startall
```

Execute the following command to stop the services of all components:

```
$CAMM_HOME/tools/stopall
```

Where $CAMM_HOME is the installation directory.

How do I stop or start the services of the individual components of DX NetOps?
Execute the following command to start the services of a component:

```
$CAMM_HOME/tools/startall -c <component>
```

Execute the following command to stop the services of a component:

```
$CAMM_HOME/tools/stopall -c <component>
```

Where $CAMM_HOME is the installation directory and <component> is GE, MC, LC, or WEB.
Example: $CAMM_HOME/tools/stopall -c MC

How can I manage a subcomponent of DX NetOps (for example, start, stop, restart, add, delete, or update a subcomponent)?
The functions and the corresponding commands for the subcomponent are as follows:
Execute the following command to start a subcomponent:

```
$CAMM_HOME/tools/cammSubCtrl -ma <MCaddress> -id <subcomponent_id>_CAMM
```

Execute the following command to stop a subcomponent:

```
$CAMM_HOME/tools/cammSubCtrl -ma <MCaddress> -id <subcomponent_id> -s
```

Example: $CAMM_HOME/tools/cammSubCtrl - ma 10.134.116.183 - id ENGINE_CAMM - s
Execute the following command to restart a subcomponent:

```
$CAMM_HOME/tools/cammSubCtrl -ma <MCaddress> -id <subcomponent_id> -r
```

Execute the following command to add a subcomponent:

```
$CAMM_HOME/tools/cammSubCtrl -ma <MCaddress> -la <LCaddress> --add -f <dpack file>
```

Execute the following command to delete a subcomponent:

```
$CAMM_HOME/tools/cammSubCtrl -ma <MCaddress> -la <LCaddress> --delete -id <subcomponent_id>
```

Execute the following command to update a subcomponent:

```
$CAMM_HOME/tools/cammSubCtrl -ma <MCaddress> -la <LCaddress> --update  -f <dpack file>
```

Where $CAMM_HOME is the installation directory.
<subcomponent> is ENGINE or PRESENTER.
mcaddress is the MultiController IP address (**Default:** Localhost).
lcaddress is the LocalController IP address (**Default:** Localhost).
subcomponent_id is the identifier of the subcomponent.

Manage Logs

Where are the log files of the DX NetOps components located?
The locations of the log files are as follows:

- **MultiController:** $CAMM_HOME/MC/logs/CAMM-Multi-Controller-<timestamp>.log
- **LocalController:** $CAMM_HOME/LC/logs/CAMM-Local-Controller-<timestamp>.log
- **Generic Controller:** $CAMM_HOME/GE_<username>/logs/*.log
- **Web:** $CAMM_HOME/WEBCAMM/logs/*.log
- **Engine:** $CAMM_HOME/COMPONENTS/ENGINE_<DEVICE_PACK>/logs directory
- **Presenter:** $CAMM_HOME/COMPONENTS/PRESENTER_<DEVICE_PACK>/logs directory

How do I increase the log level of a subcomponent?
Select a subcomponent in the web interface and change the Log Level option for the subcomponent from INFO to FINEST.

> **WARNING**
> Increasing the log level significantly increases the details being logged. As a result, more disk space is used. Monitor the disk space and revert to default settings when the problem is resolved.

Reports

What should I do if there are frequent gaps in reports though data is generated during the specified time?

This problem occurs if there is an offset in time between the system from which the input files are obtained and the system in which the Engine processes the input files. To avoid this problem, ensure that the time zone and time, accuracy up to seconds, are synchronized between the two systems.

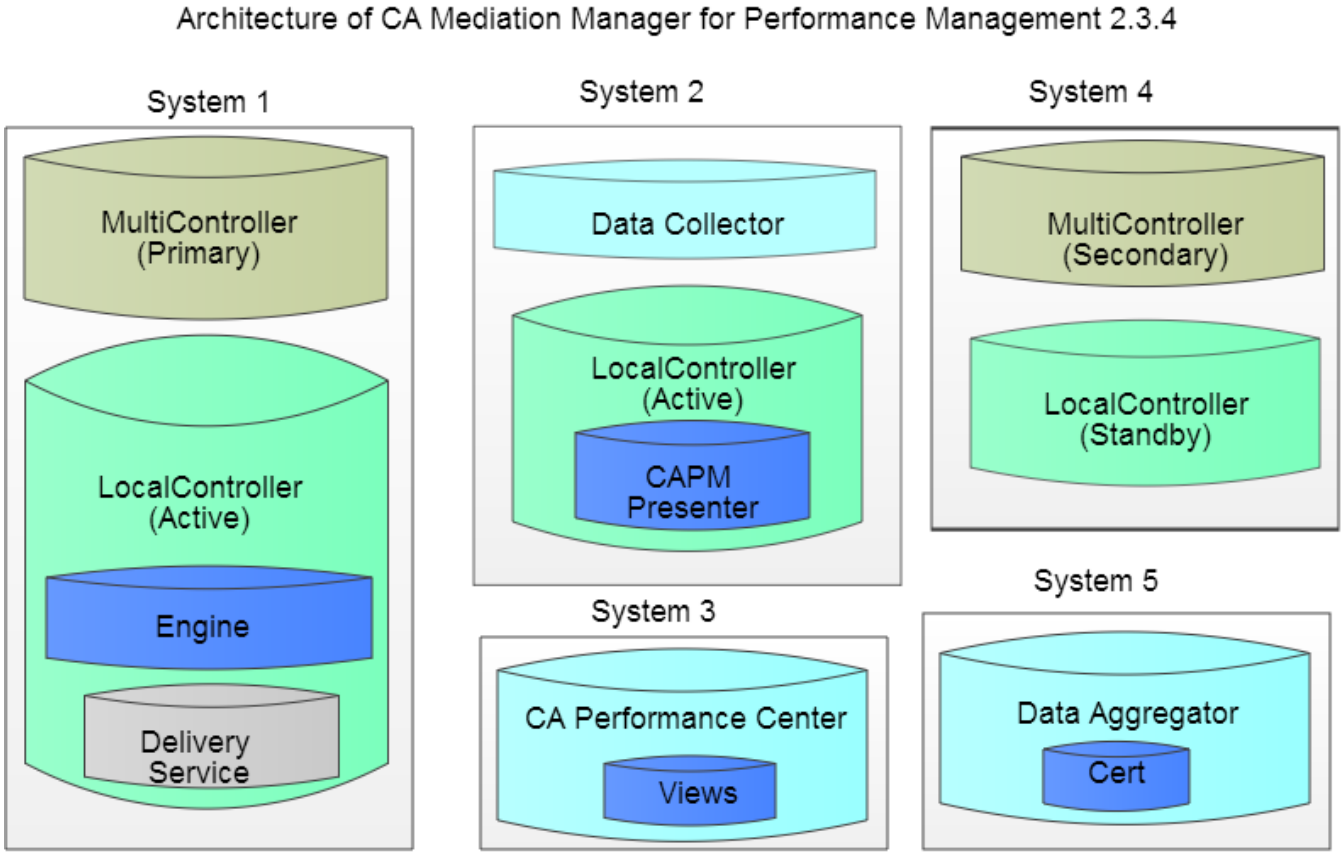
DX NetOps Mediation Manager for Performance Management

Contents

Architecture, Components, System Requirements

1. **How to distribute Primary MultiController, active LocalController, and a standby LocalController in a multiserver installation?**Install Primary MultiController and an active LocalController with Engine and Delivery Service in one system. A standby LocalController can be installed in any system that has sufficient RAM and CPU spaces. The following diagram shows one of the possible scenarios:

   Figure 93: DX NetOps MM for PM with 5 systems



2. **Is it mandatory to install Primary MultiController in the system that has CA Performance Center?**No. Install the Primary MultiController in the system that has CA Performance Center only if it has enough memory to accommodate new device packs and MultiController.
3. **Is there a system requirements guideline to install DX NetOps Mediation Manager for Performance Management?**Refer to the Sizing Guidelines document available at the CA Support site.
4. **How can I improve the performance of CAPM Presenter?**
   The number of CAPM Presenters that are required for your environment depends on the number of complex device packs that you intend to deploy. If the amount of data to be processed is very large, the performance of a CAPM Presenter can be impacted. In such deployments, install multiple CAPM Presenters. In DX NetOps Mediation Manager, you can use the same installation file of a subcomponent and can install multiple subcomponents by changing the name of the subcomponent. Therefore, install multiple CAPM Presenters by using the same Presenter zip file, which is created after migrating the device pack. Similarly, install multiple Engines and provide the path of the corresponding CAPM Presenter and host names.

Example 1:
To collect data from nine hosts (named H1 through H9), install multiple Engines and CAPM Presenters, and distribute the host names accordingly. One of the scenarios is as follows:

- Engine 1 contacting CAPM Presenter 1 and Hosts: H1, H2, and H3
- Engine 2 contacting CAPM Presenter 2 and Hosts: H4, H5, and H6
- Engine 3 contacting CAPM Presenter 3 and Hosts: H7, H8, and H9

Example 2:

- Engine 1 contacting CAPM Presenter 1 in Data Collector 1 and Hosts: H1, H2, H3, H4, and H5
- Engine 2 contacting CAPM Presenter 2 in Data Collector 2 and Hosts: H6, H7, H8, and H9

> **NOTE**
> All three Engines are of the same device pack but have different COMPONENT_ID parameters. All three CAPM Presenters are installed from the same zip file and have same COMPONENT_IDs as their Engine.

Scripts to start, stop, and check the status of the components

1. **How do I start, stop, or check the status of the CA Performance Center services?**Execute the following commands to stop the services:

   ```
   /etc/init.d/caperfcenter_console stop;
   /etc/init.d/caperfcenter_eventmanager stop;
   /etc/init.d/caperfcenter_devicemanager stop;
   /etc/init.d/caperfcenter_sso stop;
   ```

   Execute the following commands to start the services:

   ```
   /etc/init.d/caperfcenter_devicemanager start
   /etc/init.d/caperfcenter_sso start
   /etc/init.d/caperfcenter_eventmanager start
   /etc/init.d/caperfcenter_console start
   ```

   Execute the following commands to check the status of the services:

   ```
   /etc/init.d/caperfcenter_sso status
   /etc/init.d/caperfcenter_devicemanager status
   /etc/init.d/caperfcenter_eventmanager status
   /etc/init.d/caperfcenter_console status
   ```

2. **How do I start, stop, or check the status of the Data Aggregator services? How do I start, stop, or check the status of the Data Aggregator services?**
   Execute the following commands to start the services:

   ```
   /etc/init.d/dadaemon start
   ```

   Execute the following commands to stop the services:

   ```
   /etc/init.d/dadaemon stop
   ```

   Execute the following commands to check the status of the services:

   ```
   /etc/init.d/dadaemon status
   ```

3. **How do I start, stop, or check the status of the Data Collector services?**
   Execute the following commands to start the services:

   ```
   /etc/init.d/dcmd start
   ```

   Execute the following commands to stop the services:

   ```
   /etc/init.d/dcmd stop
   ```

   Execute the following commands to check the status of the services:

   ```
   /etc/init.d/dcmd status
   ```

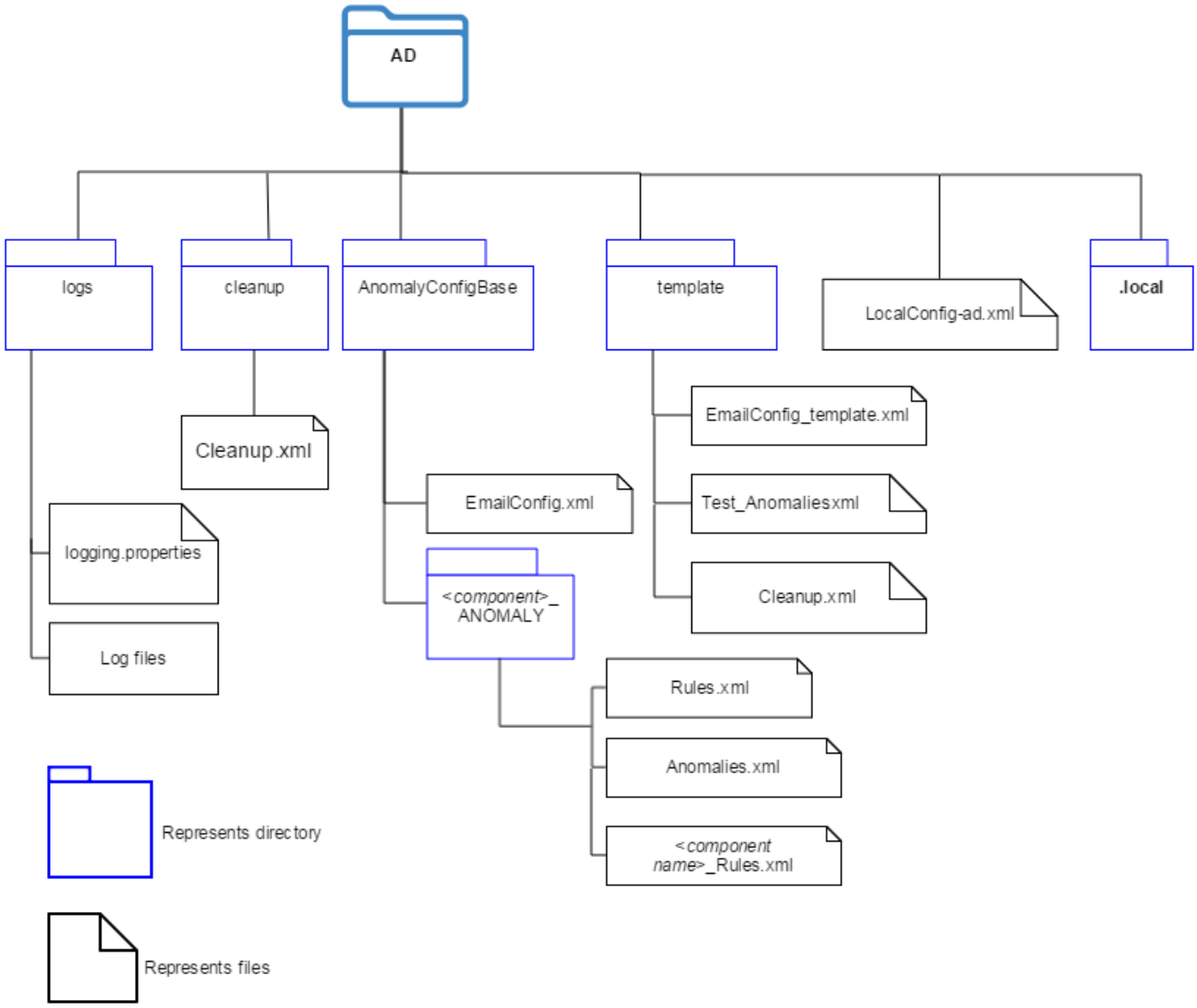
Device Pack Migration

1. **What are the options in migratePMtoCAMM and when do we use them?**
   The options to execute migratePMtoCAMM are a, d, and t. Use these options under the following scenarios:
   d: To migrate the device pack before upgrading the Data Collector. However, do not start the device pack till the Data Collector is upgraded to release 2.3.4. Also, always use option “d” to migrate the CANECeNodeB, CACRBT, or CACRBTUSER device pack.
   t: To test the device pack that is generated in a DX NetOps Mediation Manager setup, in another DX NetOps Mediation Manager setup. You can also use this option to test the device pack that is generated in the lab environment before upgrading Performance Management in the production environment.
   a: To migrate the device pack and start the device pack immediately after migration. However, before you use this option confirm that the Data Collector is upgraded to release 2.3.4.
2. **Will there be any data loss after migration?**
   No. If you are using DX NetOps Mediation Manager 2.2.6 and above in Performance Management 2.3.4, the migration script will handle the historical files and start processing after the point where the EMS Profile stopped.
3. **Why are there gaps in reports after executing the migration script?**
   After the migration, there should not be gaps in the report. However, if there is no data from the EMS server at any time, gaps appear in the reports.

Notification Feature

1. **Where are the configuration files of the Notification feature available?**
   The Notification feature is available starting DX NetOps release 2.5. If you install afresh or upgrade to the latest version, the configuration files for the Notification feature are added to the following location: <*CAMM_Install_folder*>/CA/CAMM/AD
2. **What is the folder structure for the configuration files of the Notification feature?**
   All configuration files for the Notification feature are in the AD folder, which is in the <*CAMM_Install_folder*> location. The following diagram provides the list of other files and subfolders:

Figure 94: Anomaly Detector Folders and Files



3. **What are the contents of the configuration files and their uses?**

   AnomalyConfigBase

   Contains the following folders that has the anomaly files for each component and the xml file to generate emails:

   – • MC_ANOMALY
   - LC_ANOMALY
   - Engine_ANOMALY
   - Presenter_ANOMALY
   - EmailConfig.xml

   <*component*>_ANOMALY

   <*component*> refers to MC, LC, Engine, or Presenter. By default, this folder contains two xml files: Rules.xml and Anomalies.xml. When you select an anomaly for a component in UI, <*component name*>_Rules.xml is created and it contains the anomalies with rules specified in the UI. According to the rules specified in UI, notification is sent. The <*component name*> refers to the component name in the UI.

EmailConfig.xml Contains server properties details and email addresses that are specified in the Server Properties area in UI. This file is generated based on the EmailConfig_Template.xml file when you specify the parameters in Server Properties. Anomaly notification is sent to the email addresses added to this file.

logs
Contains log files that the Notification feature generates.

template
Contains template xml files to trigger test email, perform cleanup, and store the server properties and email addresses.

LocalConfig-ad.xml Starts the Notification feature. This file also contains information about the default port number. Currently, the feature uses the port 29570. Therefore, ensure that the port is available for this feature. If the port 29570 is occupied and you want to allot different port for the Notification feature, stop the feature, change the port number in the LocalConfig-ad.xml file, start the feature, and change the port number in all other locations where DX NetOpscomponents are installed.

cleanupContains xml file to perform the cleanup of the log files. You can change the log file cleanup by changing the appropriate parameters.

.local
Contains the jmxPort number and this folder is created when the feature is started. This port remotely monitors Java resources (CPU, Memory, Heap, and Number of Classes) of this feature.

4. **When I upgrade DX NetOps, will the email server properties and anomaly settings be lost?**
   No.
5. **What happens to the selected anomalies when an Engine is moved from one LocalController to another LocalController?**
   ENGINE_<*devicepack*>_Rules.xml is deleted and created again. In this process, the previously selected parameters are lost.
6. **How can I know the anomalies for which I will receive email notifications?**Use one of the following methods to know the selected anomalies:
   - Through UI: In the My Settings, Notification area, expand the DX NetOps MM Components node and click a component. For the selected anomalies in the <*component*> Anomalies area, you receive notifications.
   - Through scripts: Execute the cammStatus -ad script from the <*CAMM Install*>/CA/CAMM/tools folder.
   - Through <*component name*>_Rules.xml: See the rules which have selected=”true”.
7. Can I change the anomalies name in **the <*component name*> Anomalies area** in UI?Yes. Follow these steps:
   1. Open the <*component name*>_Rules.xml file in the <*component*>_ANOMALY folder.
   2. Change the text in the name field.
      Example: In MC-Primary_Rules.xml, change "MC stopped" in the name field according to your choice.
   3. Save the file and exit.
8. **What are the parameters in the <*component name*>_Rules.xml file and which parameters can be changed?**
   The details of the parameters in the <*component name*>_Rules.xml are as follows:

   asserted
   Indicates whether the anomaly is detected or not. Do not change this parameter.

   frequencyIndicates the time interval according to which the rule is executed in the background. If anomaly is detected during the execution, email notification is sent.
   Units: Seconds

   id
   Specifies the component identifier. Do not change this parameter.

   nameSpecifies the anomaly name. This name appears in the Anomalies section in UI.

   selectedIndicates whether the component is selected through UI for anomaly detection.
9. **How can I change input parameters of a rule that is used to detect an anomaly?**
   Open the <*component name*>_Rules.xml file in the <*component_name*>_ANOMALY folder. In this file, except the asserted and id fields, you can update the other input parameters.

Example: In PRESENTER_CAPM_Rules.xml, the DX NetOps MM DC Queue size growing anomaly, you can modify the setThresholdInPercentage and setMaxTrend input parameters.

10. How do I stop **DX NetOps** to detect anomaly?
    In the Notification area, expand the DX NetOps MM Components node, click a component, clear the check box of the anomaly, and click Save.
11. **How to customize the test email message?**
    Open the Test_Anomalies.xml file in the <*CAMM_Install_Folder*>/AD/Template folder and change the subject or title.
12. **How can I customize the troubleshooting and causes messages according to my environment?**
    Open the Anomalies.xml file in the AD/AnomalyConfigBase/<*component*>_ANOMALY location and change the text.
13. **How to start and stop the notification feature?**
    Execute the following script to start the feature:
    ```
    startall -c ad
    ```
    Execute the following script to stop the feature:
    ```
    stopall -c ad
    ```
14. Can anomaly be detected even when the **DX NetOps components are** not running?
    Yes, unless the stopall script was executed to stop all services of DX NetOps.
15. **Is there any specific system requirements for the Notification feature to function?**
    No.
16. **Can I use this feature to monitor anomalies in CA Performance Management?**
    No.
17. **Can I use this feature if the primary MultiController is down?**
    No. Do not change the settings through UI when the primary MultiController is down.
18. **What happens to the selected anomalies if an Engine is removed from my environment?**
    All the anomalies that are selected for the Engine are deleted.
19. **Where is the log file for the Notification feature?**
    In the <*CAMM_Intall_Folder*>/AD/logs location.
20. **How do I change the log levels for this feature?**
    Open logging.properties from the <*CAMM_Intall_Folder*>/AD/logs location and change INFO to FINE/FINEST
21. **How to edit the default port number?**
    Follow these steps:
    a. Stop the feature by executing the stopall -c ad script.
    b. Open LocalConfig-ad.xml from the <*CAMM_Intall_Folder*>/AD location.
    c. Change the port number in the LocalConfig-ad.xml file.
    d. Start the feature.
    e. Change the port number in all other locations where DX NetOps components are installed.
22. **How can I monitor the system resources used by this feature?**Follow these steps:

1. a. Install JDK 1.7 or above in a system where DX NetOps is running.
   b. Double-click jconsole.exe in <*JDK_HOME*>\bin to open JConsole.
   c. Select Remote Process in the JConsole New Connection dialog.
   d. Copy the JMX connection URL from the log file. Example: service:jmx:rmi:///jndi/rmi://127.0.0.1:10979/timJmxRmi
   e. Replace the “127.0.0.1” IP address with the actual IP address where the Notification process is running.
   f. Replace "10979" with the value in the jmxPort.xml file in the <*CAMM_Install_Folder*>/CAMM/AD/.local location.
   g. Type the login credentials and click Connect.
      JConsole displays the utilization details of “Memory”, “CPU”, “Threads” & “Classes”.

Device Pack Information

The device pack information [DPInfo] files contain the supported metric, specific instructions, and known issues. Click here to download the HTML format of the DPInfo files.

Click here to download the Excel workbook that contains the list of device packs and the supported CA eHealth and CA Performance Management versions.

Greenbook

This section provides problem determination and troubleshooting guidelines for the DX NetOps Mediation Manager product. CA also recommends that you interview the customer using the Troubleshooting Methodology for CA Support Partners. The Troubleshooting Methodology for CA Support Partners provides a script for obtaining basic information to assist you with problem determination for all CA products and solutions.

If you have completed the problem determination interview and used the resource information provided in this guide and the problem or issue reported by the customer is not resolved, please contact CA Support for assistance. You can open a case or contact our Customer Care representatives.

Common Troubleshooting and Actions

Common Issue

The most common issue is data missing or data discrepancy in the report. This may include an individual graph on a report not containing data through to all reports containing no data. Use the following steps to diagnosing these and other types of issues:

- Confirm the status of DX NetOps MM components:
- Log in to the Web Manager. Ensure the status of each Component and Sub Component is active (represented by a green tick). If any component is down, attempt to start them using ‘startall’ command ($CAMM_HOME/tools/startall).
- If any Sub Component is down, attempt to start it through the GUI by clicking on ‘Start’.
- If any Component or Sub Component continues to fail, increase the logging level as explained in “Using the Log files” section, reproduce the problem and collect the following log files:
  - $CAMM_HOME/MC/repository/*
  - $CAMM_HOME/<(sub)component>/log/*<current_date>.log

Other Issues

Engine common problems are related to:

- Lose of connectivity to the EMS
- The input data format is changed on the EMS

For connectivity to EMS lost:

- Check Log files in $CAMM_HOME/COMPONENTS/ENGINE_<device_pack>/logs directory, specifically the STD-ERROR-<timestamp>.log files.
- Check the file $CAMM_HOME/COMPONENTS/ENGINE_<devicePack>/.local/EngineStatistics.xml. It indicates the performance, success or failure of every part of the Poll.
- Check the file $CAMM_HOME/COMPONENTS/ENGINE_<devicePack>/repository/work/InventoryDeviceList.xml

For change of input EMS data format:

- In addition to the above log files, EngineStatistics.xml and InventoryDeviceList.xml files, also check the Engine output is making it to the Delivery Service by stopping the Presenter and checking the Queue directory.

Presenter common problems are related to:
Data not arriving from the Delivery Service
The common problem when Data are not seen in the Presenter is when the processed data is not delivered by the Delivery Service. The troubleshooting steps in this case are the following:

- Check the log files in the log directory specific to the device pack presenter specifically focusing on the STDERR log files, e.g. $CAMM_HOME/COMPONENTS/PRESENTER_<device_pack>/logs/*
- Check if the Engine output is making it to the Delivery Service by stopping the Presenter and checking the $CAMM_HOME/Queue directory.

Stop the Presenter and check if the processed data from the Engine is reaching the Delivery Service.
The processed file should be present under the $CAMM_HOME/Queue directory.

- Check to see if CSV, XML or DDI files are present in the $CAMM_HOME/output directory
- If data are expected to be presented in the SNMP format, try querying the SNMP MIB using snmpwalk.

Troubleshooting Greenbook

Contents

Basic Troubleshooting

- Connect to the DX NetOps MM Web Manager (see Using the DX NetOps MM Web Manager) to gather information about the DX NetOps MM Cluster and collect the following info:
  - Are all the components running?
  - Is it a Single Server install? If not, how many LC's are there?
  - Which Device Pack is the customer having problems with?
- Review the logs of the device pack Engine to see if the issue is in the Engine and check if restarting Engine solves the problem.
- Review the logs of the device pack Presenter to see if the issue is in the Presenter and check if restarting Presenter solves the problem.
- If the "DX NetOps MM for DX NetOps MM" Device Pack is installed on the server, DX NetOps MM Engine polls itself to indicating the current Run state of each component giving you an overall performance of the entire DX NetOps MM environment.
- By default CAMM is installed under $CAMM_HOME\COMPONENTS\ENGINE_CAMM and $CAMM_HOME \COMPONENTS\PRESENTER_CAMM directory.

How to Stop and Start DX NetOps MM Components and Sub Components

Visit the following pages to learn about the scripts to start, stop, and check the status of the components:

- DX NetOps Mediation Manager
- DX NetOps Mediation Manager for Performance Management

Using the DX NetOps MM Web Manager

CAMM Web Manager reflects the high level operational status and health of DX NetOps MM components such as MultiController, LocalController and Sub Components such as Device Pack's Engine and Presenter. However, it does not indicate if Sub Components are actually collecting data. The utilization of resources on the server by each Component is also indicated. If one or more of the components are not running, they can be restarted from the web interface.

Log into DX NetOps MM Web Manager from the URL http://server:port (8880 is default port). The default username is 'admin' and the password is 'camm'.

Using the Log Files

The CAMM directory structure is created in a logical manner. Every Component (MultiController, LocalController, Generic Executor, Delivery Service) and Sub Component (Engine & Presenter) has a log directory.

The logs are automatically archived as per settings specified in the LocalConfig-ge.xml: $CAMM_HOME/GE_<user>/LocalConfig-ge.xml

Logs for DX NetOps MM main Components are located in the following directories:

- MultiController**:** $CAMM_HOME/MC/logs/CAMM-Multi-Controller-<timestamp>.log
- LocalController**:** $CAMM_HOME/LC/logs/CAMM-Local-Controller-<timestamp>.log
- GenericController**:** $CAMM_HOME/GE_<username>/logs/*.log
- **Device Pack:** $CAMM_HOME/COMPONENTS/<device-pack-name>/logs/*.*
- **Web:** $CAMM_HOME/WEBCAMM/logs/*.log

The “STD-ERROR-<timestamp>.log” file located in the same directories contains the error messages. This is often the most important file to look at when troubleshooting an issue.

Logs for DX NetOps MM Sub Components (Engine & Presenter):
The Sub Components of each device pack installed on the server has its own sub directory under the COMPONENTS directory. The Device Pack names are included in the name of the directory. Each Sub Component has a logs directory under the specific device pack ENGINE or PRESENTER sub directories. Log files for each component would indicate if there is a problem with the Device Pack. For example, if it failed to collect/parse data from the EMS.

- **Device pack Engine:** $CAMM_HOME/COMPONENTS/ENGINE_<DEVICE_PACK>/logs directory
  Example: /opt/CA/CAMM/COMPONENTS/ENGINE_CAMM/logs/CAMM-Application-<YYYY-MM-DD>.log
- **Device Pack Presenter:** $CAMM_HOME/COMPONENTS/PRESENTER_<DEVICE_PACK>/logs directory.
  Example: /opt/CA/CAMM/COMPONENTS/PRESENTER_CAMM/logs/CAMM-Application-<YYYY-MM-DD>.log

The “STD-ERROR-<timestamp>.log” file located in the same directories contains the error messages. This is often the most important file to look at when troubleshooting an issue.

Increasing log verbosity

All the above mentioned Components and Sub Components log files by default contain informational level messages (INFO). The granularity of logging can be increased temporarily for troubleshooting purposes by modifying the corresponding “logging.properties” file located in the $CAMM_HOME/<sub-component>/logs directory of each Component and Sub-Component.

Follow these steps:

1. Edit the “logging.properties” file in the same logs directory; replace the two instances of INFO with FINEST and save the file.
   For example, change it from the default:

```
 com.torokina.common.logging.apache.FileHandler.directory=E\:\\CA\\CAMM/DS/logs
handlers=com.torokina.common.logging.apache.FileHandler
 com.torokina.common.logging.apache.FileHandler.level=INFO
.level=INFO
com.torokina.common.logging.apache.FileHandler.prefix=CAMM-Delivery-System-
```

   To this:

```
 com.torokina.common.logging.apache.FileHandler.directory=E\:\\CA\\CAMM/DS/logs
handlers=com.torokina.common.logging.apache.FileHandler
com.torokina.common.logging.apache.FileHandler.level=FINEST
.level=FINEST
```

```
com.torokina.common.logging.apache.FileHandler.prefix=CAMM-Delivery-System-
```

2. Enable the new logging level by either restarting the component or sub-component (stopall/startall or cammCtrl) or by running the following command from within the relevant log directory to enable without restart: $CAMM_HOME/tools/cammCtrl a 127.0.0.1 p <port> -l ./logging.properties

   WARNING

   Increasing the log level will result in significantly more detail being logged and consequently more disk space usage. Make sure to monitor the disk space and revert back to default INFO settings as soon as reproduction of the problem has been completed.

Documentation Legal Notice

Flow Monitoring

Network Flow Analysis gives you an enterprise-wide view into the composition of traffic on every link and helps you detect threatening traffic patterns in the making.

It provides network traffic analysis with real-time visibility into the traffic throughout your enterprise. Network groups can quickly identify the source of performance problems, validate the impact of planned and unplanned changes within the network, and avoid unnecessary WAN costs. In addition, management can make accurate decisions regarding cost reduction, capacity planning, troubleshooting, and network traffic analysis across the enterprise. You can access as much as one year of flow data for your entire network.

This section contains everything you need for flow monitoring from getting started to troubleshooting information.

Release Notes

This section contains information on the following topics:

New Features and Enhancements

The current DX NetOps release contains the following new features and enhancements:

10.0.5 New Features and Enhancements

OData API Enhancements

In NFA 10.0.5, OData API is enhanced with many new features.

The enhancements are included in the following existing APIs:

- Routers APIs
- Interfaces APIs
- Interface aggregations APIs
- Custom virtual interfaces APIs
- Application mapping APIs

The following are the new APIs that are exposed through NFA OData APIs:

- Port priority rules APIs
- Reserved seating APIs
- AS Names APIs

InSpeed/OutSpeed for Viptela Devices

You can now view the interface bandwidth up/down stream for the Viptela devices in the Physical & Virtual page under Administration. Currently NFA uses the 'ifTable' to capture a single bandwidth value and the same speed is shown as both IN and OUT for an interface. With this feature, the interface bandwidth up/down stream values that are present in the Viptela MIB are displayed as IN and OUT speed for the Viptela devices. See the following field mappings for the InSpeed and OutSpeed in the 'VIPTELA-OPER-VPN.mib'.

interfaceBandwidthDownstream - InSpeed

interfaceBandwidthUpstream - OutSpeed

The value in the interfaceSpeedMbps field is displayed as InSpeed and OutSpeed when the interfaceBandwidthDownstream and interfaceBandwidthUpstream does not have any data.

10.0.4 New Features and Enhancements

SNMP Router Refresh

From NFA 10.0.4, you can refresh the SNMP Router through the OData API. For detailed documentation, refer SNMP Router Refresh page.

10.0.3 New Features and Enhancements

Manage Address-Hostname

The NFA maintains only the latest resolved hostname against the IP address previously. From NFA 10.0.3, you can maintain the historical data of the IP address and hostname association.

NOTE

More Information:

Manage Address-Hostname

IP Address and Hostname Historical Data

IP Address and Hostname API

NFA ODataAPI QueryBuilder

The ODataAPI is a flexible tool that lets users easily extract data from the DX NetOps database. The ODataAPI enables integration between DX NetOps data and external applications. The ODataAPI is a public API that uses the QueryBuilder GUI. The QueryBuilder is a guided URL builder that lets you create custom Query URLs to extract and explore performance data. The URLs return customized data in the specified format. You can view the data in a browser or process the data in a custom web application.

More Information: NFA OData QueryBuilder

Platform Support

- Red Hat Enterprise Linux 7.5, or 7.6 on a 64-bit processor

10.0.2 New Features and Enhancements

The current DX NetOps the release includes the following new and updated features.

Support for AdoptOpenJDK Java

CA Technologies, a Broadcom Company, is moving towards adopting more open source technologies in its products. As a part of this strategy, various products have started using open-source implementations of Java. To align with this corporate direction, DX NetOps has adopted AdoptOpenJDK (1.8.0.212), replacing Oracle JDK.

Edit a Report Without Opening the Report

From this release, you can edit a custom report without opening the report. You can use the **Edit** link against a specific report and change the report criteria. You can find the edit options in the following reports:

- Custom Reports
- Flow Forensics Reports
- Analysis Reports
- Site to Site reports

Generate Flow Forensic Report for Routers Common to Multiple Harvesters

With this enhancement, you can get flow forensic report from either specific harvester or from all harvester when you select the following ROUTER filter types.

- Router Address
- Router Address and Interface In
- Router Address and Interface Out
- Router Address and Interface In or Out

When you generate a report based on one of these filters, NFA prompts you to enter the router IP. Enter the router IP address. DX NetOps displays a drop-down with a list of associated harvesters. Based on your selection, the NFA displays a report from All Harvesters (Default) or from the selected harvester.

NOTE

- Harvesters Drop-down is displayed only if the specified IP is available on any of the harvesters.
- Harvesters Drop-down is displayed even if the router belongs to only one harvester.

More Information: Create or View a Flow Forensics Report

NAT Segment Group Reports

From the DX NetOps 10.0 Service Pack 2, you can view NAT Segment Group reports. Network address translation (NAT) method maps one IP address into another by changing the network address information in the IP header of packets.

More Information: NAT Segment Group

Support for Additional NetFlow Fields in Options Template

The current release supports sampling rates specified by SAMPLING_INTERVAL (34) and SAMPLING_ALGORITHM (35) fields in the Options template.

More Information: Set Up the Routers

10.0.1 New Features and Enhancements

The DX NetOps the release includes the following new and updated features

Configure MySQL User Password

You can now manage the MySQL user password using the NFA MySQL User Password Change Utility.

More Information: Configure MySQL User Password

Application Mapping API

Using the application mapping, you can discover and map all your entities and their inter-dependencies. CA Network Flow Analysis supports the ToS, host, and subnet applications.

More Information: Application Mapping

Delete Available Interfaces

You can delete an interface from the available interfaces.

More Information: Available Interface

Support for Microsoft Excel as OData Client

This release supports OData as a Data source for Microsoft Excel client only for entity collections.

Enabling Router Name Prefix in Interface Selection Tab

The interface selection tab in Custom and Analysis reports now shows an Interface name with Router Name/IP as a prefix. Now users can identify the correct interface to include/exclude in the report, when multiple interfaces have same name.

10.0.0 New Features and Enhancements

The DX NetOps the release includes the following new and updated features

Support for AWS VPC Flow Logs

When you enable the Amazon Web Services virtual private cloud flow, you can view the following network traffic flow reports:

- Source and destination IPv4 address
- Source and destination ports
- Protocols used
- Bytes and packets transferred

More Information: AWS VPC for CA Network Flow Analysis

Support for DX NetOps OData API

This release supports APIs that enable you to extract data from the DX NetOps or perform Administrative Operations. The DX NetOps APIs use the OData v4.0 industry standard. OData enables integration between DX NetOps and other external applications.
You can perform the following actions using the DX NetOps OData API:

- Generate data in smaller sets and view the result in multiple pages.
- Use custom query to override the system default values for any specific query.
- Use built-in filtering options to filter the required data. For example, you can retrieve a list of routers that are rebooted in last 24 hours.
- Extract data for the entities and navigate to the associated entities. For example, you can retrieve data for protocol_traffic within a time duration.

You can perform various Administrative Operations on the DX NetOps entities. For more information, see Network Flow Analysis OData API.

Microsoft .NET Framework Update

Dependency on Microsoft .NET Framework 3.5.1 for using DX NetOps is removed.

More Information: Compatibility Matrix.

Handling of Router Refresh Scenario - Fresh Installation

When there is a change in SysObjectID of a router sending NetFlow to DX NetOps, DX NetOps treats the change as a router refresh. Router Refresh detection can happen in one of the following scenarios:

- Changing the configuration on the router with SysObjectID change.
- Adding or deleting the router interface with SysObjectID change.
- Using the same IP when replacing an existing router with a new router with SysObjectID change.

In the previous version of DX NetOps whenever a Router Refresh is detected in DX NetOps, the following changes are made to the interfaces/ routers:

- CA NFA prefixes the existing interfaces of that router with string OLD -
- CA NFA creates a set of interfaces and starts showing data on them.

Eventually, you see a router with multiple interfaces with duplicate ifIndex values.

When this information is synced up to CA PM, the interfaces were getting consolidated in CA PM due to duplicate ifIndex values. Hence it was not possible to view either historical or live data on those interfaces.

From the current release, the following changes are made in Router Refresh.

When a router Refresh is detected, the following changes are made to the interfaces/ routers:

- CA NFA prefixes the existing interfaces of that router with string OLD - and suffix with timestamp to identify multiple routers Refresh.
- CA NFA sets the LifeCycleState of Router to RETIRED in database.
- CA NFA creates a router with the same IP in CA NFA Console with LifeCycleState ACTIVE.
- The new interface is attached to the New router.
- If there is another router refresh detection on the same router, the ACTIVE router is marked as RETIRED and a new ACTIVE router is created.

Eventually, you can see more than one router with the same IP, whenever a device refresh occurs.

Handling of Router Refresh Scenario - After Upgrade

When you upgrade to the current release from any supported version of CA NFA, the following changes occur.

When a router Refresh is detected before upgrade, [Interfaces with duplicate ifIndex attached to one router], the following changes are made to the interfaces/ routers:

- CA NFA adds a router entry with LifeCycleState as RETIRED in the reporter DB for each such router
- All the inactive interfaces [Interfaces with OLD - prefix] of each router point to the Retired routers

NOTE

- If more than one router refresh happened on a router before upgrade, you can see only one Retired router with all the Inactive Interfaces.
- When you upgrade from CA NFA 9.3.6 and CA NFA 9.3.8 to the current release, you may not view the prefix OLD for refreshed devices as the prefix is not done in the base releases.

Compatibility Matrix

DX NetOps is compatible with the following products/ application:

NOTE
For more information about supported DX NetOps releases, see the DX NetOps Interoperability.

Product	Version	NFA 10.0.5	NFA 10.0.4	NFA 10.0.3	NFA 10.0.2	NFA 10.0.1
DX NetOps Performance Management		See: DX NetOps Interoperability				
DX NetOps Virtual Network Assurance		See: DX NetOps Interoperability				
Microsoft .NET Framework	4.6.2	**yes**	**yes**	**yes**	**yes**	**yes**
Microsoft Internet Explorer	11	**yes**	**yes**	**yes**	**yes**	**yes**
Unified Infrastructure Management (UIM)	20.1.0	**yes**	**yes**	**x**	**x**	**x**
	9.2.0	**yes**	**yes**	**yes**	**x**	**x**
	9.1.0	**yes**	**yes**	**yes**	**yes**	**x**
	9.0.2	**yes**	**yes**	**yes**	**yes**	**yes**
	9.0.1	**yes**	**yes**	**yes**	**yes**	**yes**
	8.5.1	**yes**	**yes**	**yes**	**yes**	**yes**
nfa_inventory probe (for UIM)	v1.41	**yes**	**yes**	**yes**	**yes**	**yes**
DX Operational Intelligence	1.3 (OI Connector 1.3)	**yes**	**yes**	**yes**	**x**	**x**
	1.3 (OI Connector 1.2.1)	**yes**	**yes**	**yes**	**yes**	**yes**

NOTE
OI connector 1.2.1 does not support CA NFA 10.0.1, if you change the default password for the MySQL user using the MySQL User Password utility.

Product Accessibility Features

CA Technologies is committed to ensuring that all customers, regardless of ability, can successfully use its products and supporting documentation to accomplish vital business tasks.

Product Names and Abbreviations

This documentation references the following products and abbreviations:

- Access Control List (ACL)
- CA Anomaly Detector (CA AD)
- CA Application Delivery Analysis (CA ADA)
- CA NetQoS Performance Center (CA NPC)
- CA Performance Center (CA PC)
- CA Performance Management (CA PM)
- CA Unified Communications Monitor (CA UC Monitor)
- CA Unified Infrastructure Management (CA UIM)
- CA Unified Infrastructure Management Portal (UMP)
- Quality of Service (QoS)
- Simple Network Management Protocol (SNMP)
- Simple Network Management Protocol Data Monitoring Probe (SNMPcollector)
- Type of Service (ToS)
- Unified Data Management (UDM)
- Unified Service Manager (USM)
- Virtual Machine (VM)

Third-Party Software License Agreements

This section contains third-party software license agreements for applications that are added/included as part of the current release of DX NetOps

To view the Third-Party Software License Agreements, click here.

Ace Editor 1.2.0
Adobe Flex SDK (Software Development Kit) 3.2
AdoptOpeJDK 8u232 (in NFA 10.0.3)
AdoptOpeJDK 8u212 (in NFA 10.0.2)
Apache Olingo 4.0.0
Apache Commons Beanutils 1.8.3
Apache Commons CLI v.1.3.1
Apache Commons Codec 1.4
Apache Commons Collections 3.2
Apache Commons Collections 3.2.2
Apache Commons IO 2.0.1
Apache Commons IO 2.5
Apache Commons JXPath 1.3
Apache Commons-lang 2.5
Apache commons-lang3 3.5
Apache Commons Logging 1.1.1
Apache Commons Pool 1.5.4
Apache Commons Pool 1.5.5
Apache Commons Pool 1.5.6
Apache Commons Collections 4.2
Apache CXF 2.3.2
Apache Cxf Spring Boot Starter jaxrs 3.2.6
Apache Geronimo Javamail 1.7.1
Apache geronimo-jms_1.1_spec-1.1.1.jar
Apache Geronimo WS-Metadata 2.0-spec 1.1.2
Apache HttpClient 4.3.6
Apache Jakarta Commons DBCP 1.4
Apache Jakarta Commons HttpClient 3.1

Apache Log4j 1.2.14
Apache Neethi 2.0.4
Apache Odata Commons api 4.5.0
Apache Odata Commons Core 4.5.0
Apache Odata Server Api 4.5.0
Apache Odata Server Core 4.5.0
Apache Odata Server Core Ext 4.5.0
Apache Odata Client Core 4.5.0
Apache Odata Client Api 4.5.0
Apache ServiceMix Bundles Spring Framework 5.0.9
Apache Spring Boot Starter Web 2.0.5
Apache Spring Boot Starter Artemis 2.0.5
Apache Spring Boot Starter Web Services 2.0.5
Apache Spring Boot Starter jdbc 2.0.5
Apache Xalan-J 2.6.0
Apache Xerces-C++ 3.0.1
Apache xercesImpl 2.9.1
Apache xml-commons-resolver 1.2
Apache xml-commons xml-apis 1.3.04
Apache XMLGraphics Commons 1.4
Apache XMLSchema 1.4.7
Apache xmlsec 1.4.4
Apache xmltooling 1.3.1
ASM 3.3
ASM 5.0.1
BC Flips 1.0.1
Blowfish Utilities
Bootstrap 3.3.5
bootstrap-daterangepicker 2.0
bootstrap-table 1.8.1
clipboard.js 0.2.0
Codeproject Rijndael Dated November 2002
Commons Collections 3.2.1
Commons Codec 1.4
Commons Dbcp 1.4
Commons net 3.5
Commons net 3.7
Ch.qos.logback:logback-classic.jar 1.2.3
Ch.qos.logback:logback-core.jar 1.2.3
Javax.annotation:javax.annotation-api.jar 1.3.2
Javax.jms:javax.jms-api.jar 2.0.1
Javax.ws.rs:javax.ws.rs-api.jar 2.1
DnDns 1.0.1.0
ext JS -5.1.1.451
FileSaver.js 1.1
geronimo 1.1.2
geronimo annotation 1.0 spec 1.1.1
Guava 18.0
Guava 19.0
HikariCP 3.2.0
ICU4C 4.4.1
Java Deep-Cloning Library 1.7.4
Javax servlet api 3.0.1

JAXB (Java Architecture for XML Binding) 2.1.13
JDOM 1.0
Jettison 1.2
Jetty 7.2.2
Jetty 9.2.17
Jetty 9.4.18
Joda-time 1.6.2
Joda-time 2.8.2
jQuery 2.1.4
jsr311-api 1.1.1
JSTL (The JSP Standard Tag Library) 1.2
JSW (Java Service Wrapper) 3.5.35
JSR 250 (Javax Annotations)
Myfaces 1.1.4
MySQL 5.7.26
MySQL 5.7.31
MySQL Connector Java 5.1.39
MySQL Connector ODBC 5.3.14
not-yet-commons-ssl 0.3.9
Objenesis 1.2
OpenSAML 2.4.1
OpenSSL Toolkit v0.9.8h
OpenSSL 1.0.2h
OpenSSL 1.1.1g
org.apache.servicemix.bundles.spring-tx 3.2.14
org.apache.servicemix.bundles.spring-jdbc 3.2.14
Rickshaw 1.5.1
RockSaw 1.1.0
Simple Logging Facade for Java (SLF4J) 1.6.1
SNMP4J 2.3.4
SNMP4J-Agent 2.1.1
snmp_pp 2.8
SpringFramework v.3.1.0
Slf4j 1.7.25
Tomcat Jasper 8.5.34
Tooltipster 3.3.0
tomahawk 1.1.5
Velocity 1.5
WSDL4J (Web Services Description Language for Java Toolkit) 1.6.2
wss4j1.5.11
XMLSchema 2.1.0
XStream 1.4.8
Licenses

Key Terms and Concepts

Contents

1-minute (high-resolution) data

1-minute (high-resolution) data is detailed information that is collected from each Harvester and is provided to the NFA console for display in views and reports. The data includes top protocols for each interface; traffic for the top hosts and conversations; top conversations for the top protocols; and top protocols, hosts, and conversations for the top ToS values. The 1-minute data is stored on the Harvester server in the `archive` database.

15-minute (historical) data

15-minute (historical) data is longer-range information that is collected for each interface. The information includes the protocols, hosts, and conversations for each interface. Summary data is also collected for the ToS, the top protocols for the top ToS values, and the top hosts and conversations for the top ToS values. The data is stored in the database.
`archive15`

Administrator

An A*dministrator,* in the context of this document, is a person who is responsible for administering the product in the NFA console. An Administrator also manages elements in the Performance Center Console that are related to DX NetOps, such as SNMP profiles, groups, users, and roles.

application mapping

Application mapping is a rule-based technique for combining the traffic for an application to facilitate reporting for the application. Application mapping rules are based on factors that can include the traffic origin (host, subnet and mask, and/or port), ToS, and protocol.

Autonomous System

Autonomous System (AS) refers to a connected group of Internet Protocol (IP) routing prefixes. The IP routing prefixes have a single, clearly defined routing policy and are controlled by one or more network operators. Meaningful AS data is available in reports only when routers and interfaces are configured to export it.

baseline

A *baseline* is a record of typical behavior, which is computed from past behavior. Baselines help you compare changes over time and predict future data or performance. Comparing current values to baseline projections is useful for determining whether current values are typical. The baseline in a trend plot is computed by using data from the six weeks before the selected date range, excluding the data point already in the trend plot.

conversation

A *conversation* is a session of subnet-to-subnet or user-to-user (host-to-host) traffic. TheDX NetOps the console displays conversation information, so you can find out whether a particular conversation is causing a traffic spike on an interface, for example. You can create and run reports to identify the top volume-based conversations.

custom virtual interface

A *custom virtual interface* (CVI) is an abstract representation of a network interface, which corresponds to one or more subnets of actual physical interfaces. CVIs can give you visibility into network traffic for a carrier cloud. Set up CVIs for data center traffic that is transferred to subnets through an MPLS carrier cloud when the flow is enabled on the routers in the data center.

dashboards

Dashboards are dynamic report-building pages in the Performance Center Console. Dashboards are accessible from the **Dashboards** tab (CA PC) or **Reports** tab (NPC). Each dashboard is a collection of views that present data from registered data sources on a single web page. The layout, views, time interval, and group context of each dashboard can be customized.

data sources

Data sources are the products that provide data for display in the Performance Center Console. Data sources also provide some configuration data that is stored in the Performance Center. DX NetOps is designed to be a data source for Performance Center.

drilldown report

A *drilldown report* is a more detailed report that you display by clicking a link in a report. You can open a drilldown report by clicking an interface name in an Enterprise Overview page report, for example. Properly credentialed users also can drill down from Performance Center views to detailed reports in the NFA console.

drill down

To *drill down* is to navigate from one data view to another, more detailed data view or context page. The new page displays data from the same time frame, for the same managed item or set of items. You can drill down to details in DX NetOps from views in Performance Center.

filter

A *filter* in a report is a set of selection criteria that are used to focus a report on the desired data.

firewall

A *firewall* server acts as a gateway between a local area network (LAN) and a large network that is not secure--such as the Internet. A firewall server typically runs a software package that inspects inbound and outbound packets, and decides whether to allow the packets to pass.

flow

A *flow* is a set of IP packets that pass a network observation point during a certain time interval. In DX NetOps, flow may consist of NetFlow v5, v7, or v9 or one of the following flow types that conforms to the standards for NetFlow v5, v7, or v9: sFlow version 5; or IPFIX, J-Flow, cFlow, or Huawei NetStream flow.

For data from non-sampled flows to appear in reports of 15-minute (historical) data, these minimum fields are required:

- One of the following: 1 - IN_BYTES, 85 - IN_PERMANENT_BYTES, 231 - FW_INITIATOR_OCTETS, or 232 - FW_RESPONDER_OCTETS
- All of the following: 4 - PROTOCOL, 7 - L4_SRC_PORT, 8 - IPV4_SRC_ADDR, 10 - INPUT_SNMP, 11 - L4_DST_PORT, 12 - IPV4_DST_ADDR, and 14 - OUTPUT_SNMP

group

A *group* is a collection of managed items that are organized in a tree structure. A global administrator can use Performance Center to create custom groups of the managed items that an operator can see. These managed items can be applications, servers, networks, routers, and interfaces, for example.

Harvester

A *Harvester* is a component in a distributed deployment of DX NetOps, which collects raw flows from the routers. In a two-tier architecture deployment, the Harvester processes and stores the 1-minute and 15-minute data.

host

A *host* is a specific computer engaged in an exchange across the network. In some cases, a host represents a managed services provider whose IT staff manages and monitor the networks and systems of multiple customers. In DX NetOps, hosts are identified by name or IP address. You can track host activity to find out whether a specific server or end-user system is responsible for significant traffic on an interface, for example. You can create and run reports about the traffic that is generated or is received by specified hosts.

IIS

IIS is the Web server that is part of the Microsoft Windows Server application. IIS consists of several services, including Simple Mail Transfer Protocol (SMTP). In versions of IIS before 5.0, IIS is an abbreviation for Internet Information Server. In version 5.0 and later, IIS is an abbreviation for Internet Information Services.

interface

An *interface* is a point of connection, such as a Serial, Frame Relay, Fast Ethernet, ATM, or PVC interface. DX NetOps reports on any logical interface that is enabled on a supported router that has flow enabled. The NFA console displays the interfaces that are monitored in your environment.

IP domains

IP domains are logical collections of data from different devices and networks. Domains let your enterprise conduct separate monitoring of IP addresses with associated interfaces or monitor applications that belong to separate customer networks. A global administrator can monitor IP domains from a single Console, but operators view data only for the domains that they have permission to view. Administrators create custom IP domains in the Performance Center Console. Administrators can use the NFA console to assign Harvesters, routers, interfaces, CVIs, and some other elements to IP domains.

LDAP

LDAP, or Lightweight Directory Access Protocol, is a software protocol for locating organizations, individuals, and other resources, such as files and devices in a network. LDAP is based on a client/server model. The LDAP client makes a Transmission Control Protocol (TCP) connection to an LDAP server, and then sends requests and receives responses over this connection.

NetFlow

NetFlow is a transaction between two hosts, which uses a unique pair of port numbers and IP addresses and which includes certain network traffic information. A Cisco router can be configured to export flow information by sending UDP packets that contain flow statistics to one or more collectors such as the Harvesters. DX NetOps supports NetFlow versions 5, 7, and 9 and sFlow version 5. DX NetOps also supports IPFIX, J-Flow, cFlow, and Huawei NetStream that complies with the standards for NetFlow v5, v7, or v9.

NFA console

The *NFA console* is a component in a distributed deployment of DX NetOps, which provides a web-based user interface for reports and for some administrative functions. The NFA console creates reports from Enterprise Overview data, which is stored locally and from the 1-minute resolution data and 15-minute resolution data that it retrieves from other components.

Performance Center

Performance Center is a term this documentation uses to refer to CA Performance Center and CA NetQoS Performance Center collectively. DX NetOps is designed to be used with one of these programs. Page names or functions that are specific to a Performance Center version may be identified by the full program name or acronym. *CA PC* is used as an acronym for CA Performance Center and *NPC* is used for CA NetQoS Performance Center.

permission groups

Permission groups define the scope of the managed items that each user or operator can monitor. Administrators can create and assign custom groups of items to match each user's area of responsibility, such as applications, servers, networks, routers, and interfaces. Administrators assign permission groups in Performance Center to give users access to default or custom groups.

product privilege

A *product privilege* is a type of permission that is associated with a user account in Performance Center. The product privileges grant access to features in the Performance Center Console, the NFA console, and any other data sources. The administrators who manage user accounts assign product privileges in the Performance Center Console.

protocol

A *protocol* is a standard for regulating communication between computers. Common protocols include: HTTP, SNMP, FTP, and VoIP. The information that is displayed may include the top protocols in and out for a particular interface. This information can help identify which application is causing network traffic. You can also create and run reports to determine which protocols and applications are used by different groups in your organization.

QoS (Quality of Service)

QoS (Quality of Service) is a defined level of performance--quality of transmission and service availability--in a data transmission system.

report

A *report* is a display of collected data, which you view in the NFA console from the **Enterprise Overview**, **Interfaces**, **Custom Reporting**, **Flow Forensics**, **Analysis**, and **Site to Site** pages. You can print or save reports in PDF format. You can also export reports as comma-separated value (CSV) files. An Administrator can set up some reports to be sent by email at scheduled intervals.

reporting information base (RIB)

The *reporting information base (RIB)* is a system of web services and XML files that describe and provide the data for views and dashboards in the CA Performance Center Console. This data originates from data sources, such as DX NetOps. The RIB capability provides an operating environment for cross-product, federated, and third-party reporting. RIB uses a single data access web service with SQL-like capabilities.

reporting period

A *reporting period* is a user-specified time range for data to be included in a DX NetOps report. The time options vary with each report type, but the report period could consist of hours, days, weeks, or months.

Reserved Seating

Reserved Seating is a rule-based technique for ensuring that reports include the traffic that interests you, even if the traffic volume or rate is low. The rules create 'reserved seats' in reports for data that matches the target ports and protocols.

role

A *role* controls access to product features in the NFA console and the Performance Center Console. In a well-planned deployment, roles let users access the features they need to perform their duties. Roles also restrict access to features that operators and administrators do not need. The administrator who manages user accounts assigns roles in the Performance Center Console.

Single Sign-On

Single Sign-On is the authentication scheme that provides a one-time login to authenticate users in the suite of related products. Once users are authenticated, they can navigate among the products without signing in again.

SMTP

SMTP (Simple Mail Transfer Protocol) is the Transfer Control Protocol/Internet Protocol (TCP/IP) protocol that is used for sending and receiving e-mail in data networks.

SNMP

SNMP (Simple Network Management Protocol) is a network management protocol that is used almost exclusively in data networks. SNMP is a method for monitoring and controlling network devices, as well as managing configurations, statistics collection, performance, and security.

SNMP profiles

SNMP profiles are definitions that contain the information for using SNMP securely to query device MIBs (Management Information Bases). Each connection to a device is made by using an SNMP profile. Administrators create SNMP profiles as needed in the Performance Center Console. In a multi-tenant CA Performance Center environment, SNMP profiles are tenant-specific. In this type of environment, each Harvester uses one of the SNMP profiles that are set up for its parent tenant.

Summary views

Summary views provide an overview of high-level information, such as averages from groups of managed items. Summary views often provide drilldown paths to more detailed, related pages.

synchronization

Synchronization, or global synchronization, is a Performance Center process that exchanges configuration and other data with DX NetOps. For example, if an administrator creates user accounts or SNMP profiles, the associated data is pushed down to the NFA console through synchronization. Synchronization occurs every 5 minutes automatically. Administrators also can perform a full or partial synchronization on demand.

threshold

A *threshold* is a user-definable limit. Meeting or exceeding a threshold may trigger an alarm. Thresholds are also used in some views to determine the status colors for items. For example, the Interface Utilization view on the **Enterprise Overview** page uses user-definable utilization thresholds for the status colors of the top interfaces.

trap

A *trap* is a message that indicates a threshold has been reached or that another user-defined condition has occurred. An SNMP agent sends traps to the NFA console or to a network management system (NMS). The Watchdog agent defines a number of traps for system and application management.

trend line

A *trend line* is a projection of the future performance of an element that is based on data from past performance. DX NetOps constructs the trend line as the best straight line through the data points of the baseline period.

two-tier architecture

Two-tier architecture refers to a type of DX NetOps deployment. The components work together to collect, process, and store flow data; display the data in reports; and generate traps, events, and scheduled reports.

A two-tier architecture deployment consists of the NFA console and one or more Harvesters (Windows or Linux). These components may be located on separate servers or on a stand-alone server.

view

Views, or *data views*, present report data, usually as a bar graph, pie chart, table, trend chart, or stacked trend chart. A view is created on the fly when you display data in the NFA console or the Performance Center Console. For example, the **Enterprise Overview** page in the NFA console consists of a collection of views. In some cases, you can export the view data to a file in `CSV` format or create a PDF report from it.

Web user interface

The DX NetOps the web user interface appears as the NFA console, which lets an operator access DX NetOps views and reports from a web browser. Administrators for DX NetOps use this interface to perform a number of administrative functions.

Known Issues

This section describes the known issues and workaround, if any:

Known Issue in release 10.0.5

Filter and OrderBy system queries do not work for custom properties in NFA OData API

Symptom: Filter and OrderBy system queries do not work for custom properties in NFA OData API.

Solution: No known solution as of now.

There are no other known issues in the current release. Password Reset issue and DBPasswordUtils is not working in windows 2012/2016 harvester issue can be found while upgrading, this occurs when the password has been changed and upgrading to 10.0.5.

Known Issue in release 10.0.4

DBPasswordUtils is not working in windows 2012/2016 harvester

Symptom: The utility gets hanged and the passwords cannot be changed.

Solution: Replace the ReporterAnalyzer.ini file from console and copy to the harvester machine and rerun the DBPasswordUtils.

There are no other known issues in the current release. Password Reset issue can be found while upgrading, this occurs when the password has been changed and upgrading to 10.0.4.

Known Issue in release 10.0.3

There are no new known issues in the current release. Password Reset issue can be found while upgrading from 10.0.2 version. This occurs when the password has been changed and upgrading to 10.0.3.

Known Issue in release 10.0.2

Password reset to default password on upgrade from 10.0.1 to 10.0.2 on SB.

Symptom:

NFA Services are down.

Solution:

Re-run the password utility after the upgrade.

> **NOTE**
> This known issue in Network Flow Analysis 10.0.2 is applicable to the releases 10.0.2 and later. This issue occurs when upgrading from 10.0.x to 10.0.2 and later versions.

Reaper service does not start on Linux

Symptom:
Reaper service is down.

Solution:
Navigate to `<NFA_HOME>/DBUsers/` directory and run the `dos2unix ReporterAnalyzer.ini` command. Start the Reaper service.

Known Issue in release 10.0.1 and Previous Release

Application Failed While Creating Response from the Data

Symptom:

Expand queries gives an error for olingo v4.1.0 when XML is the output format.

Solution:

No known solution as of now.

Pump Service Fails to Start after Upgrade

Symptom:

The pump service fails to start after upgrade.

Solution:

Verify the connection errors in the pump log. Ensure all the services are up and running and restart the pump service.

If you notice any exception that is related to “Invalid file format”, delete the content in `$INSTALL_DIR/REPORTER/datafiles/input/Staging` and start the pump service.

Report Service Does not Start after Upgrade

Symptom:

The Report service fails to start after upgrade.

Solution:

If you notice any MySqlException exception in the Report Service log, restart the Report service.

Delay in Display of AWS VPC Flow Data

Symptom:

The AWS VPC flow data does not show up immediately in the CA NFA console.

Solution:

There is a delay of ten minutes to display the AWS VPC Flow data received from CA VNA.

Known Issue in release 9.5.0

The known issues that were identified in DX NetOps 9.5.0 are applicable to the current release too. For more information, see the Known Issues 9.5.0 page.

Fixed Issues

Issues Fixed in DX NetOps 10.0.5

Defect ID	Support Case	Symptom	Resolution Text
DE453462	20296423	For the Linux machines, when a disk is mounted it is sent as a fixed disk and NFA stores it in the DiskUtilization table. While the disk is unmounted NFA does not record the event and the existing record is not deleted from table showing an alert.	While getting the disk utilization details from the SNMP calls, the existing fixed details for that particular harvester are deleted. And the new disk utilization details are inserted.
DE468870	32057726	When drilling down to NFA from CAPC (as a power user), the Flow Forensics, and the Site to Site tabs are hidden. The issue frequently occurs when you navigate between the tabs in NFA.	With this fix, admin users can view the Flow Forensics, and the Site to Site tabs when drilling down to NFA from CAPC.
DE472164	32108663	For MySQL users, if the password contains "D"(capital D) and while encrypting the password it is converted to special character ":". When NFA OData API forms the connection url to connect database, issues occur due to the special characters in the url. In few OData API calls urls are formed dynamically and in few cases urls are formed using props and settings.	Code changes are made to query the calls using OData Api even the password in the ReporterAnalyzer .ini file contains ":".

DE474732	32168437	When using the OData API to create a CVI, some subnets are not created properly in the database. This issue occurs when subnet octets range is more than 127, example: 128.128.128.0/24.	Modified the OData API query to properly insert the subnet ranges.
DE475864	20081275	The Top Interfaces charts in EOV page show high Rates and Utilization for interfaces.	Code changes are made to show the correct Rates and Utilization for interfaces for the Top Interfaces charts in EOV page.
DE474918	32143693	Viptela devices interface up/ down stream speeds are not accurate.	Code changes are made to display the correct up/down stream values of the interface bandwidth for the Viptela devices.
DE472115	32170411	FLT files are creating unwanted interfaces.	Removed the FLT file interface creation code in the NFAConsole to avoid the duplication of the interfaces.
DE460119	31879645, 31825621	As part of merge interfaces, after the console/harvester services are restarted, the merge activity gets aborted intermediately.	Code changes are made to verify the merge activity when the harvester/console services are restarted. Also implemented the retry mechanism when the merge activity goes wrong after the restart.
DE480460	32265965	RouterName and RouterID are missing in Interfaces API after upgrading from 10.0.2.	RouterName and RouterID are now available in the Interfaces API.
DE446065	20292015	While processing few custom virtual subnet rules which has IPv6 source/destination getting an exception "java.lang.IllegalArgumentException: Length of IP Address and Subnet Mask do not match!"	NFA does not support custom virtual interfaces rules for IPv6 flows whose source/destination is IPv6. Added checks to validate any such flows to discard while processing.

Issues Fixed in DX NetOps 10.0.4

Defect ID	Support Case	Symptom	Resolution Text
DE427109, DE401883	1260915	AD errors in log after upgrading to 10.0.1	Code changes are done to handle the custom storage engine’s bad templates with invalid data length.
DE449372	20325365	When a device is deleted from NFA's 'Enable Interfaces' page, does not get deleted from the CAPC, the NFA is attached to.	A device when deleted from 'Enable Interfaces' page in NFA UI gets deleted (along with its interfaces) from the CAPC, the NFA is attached to.

DE431573	20064968, 20019669	After adding a router to a harvester sometimes, we get the exception 'Expected flow sequence cannot be null' is seen in harvester wrapper log file.	Worked on the root cause of the issue and now the exception 'Expected flow sequence cannot be null' will not be seen in harvester wrapper log file.
DE424445	20023828	While loading 15-min resolution data NQMysql service crashes, if the database file is invalid under ReaperArchive15 directory.	Code changes are made to resolve Netqos NQMysql service crash so that the data for 15-min resolution gets loaded successfully.
DE454079	31830273	Odata api has ifindex type as Singed Int 32 while as per standards ifindex of interfaces can be of unsigned int 32.	Modified the way odata handles ifindex of routers.
DE441128	20101447, 20283369	FLT files are processing too slow.	Added index to few columns of the reporter.routers table and optimized few queries which are involved in processing of .flt files.
DE439463	20107064, 31829166	Odata api while deleting interface/s and router/s runs delete queries in reporter and harvester tables and also sends agentids instead of interaceids while deleting interfaces from reporter.interfaces table.	Modified delete interface odata api and router delete odata api so that wrong interfaces will not get deleted from reporter.interfaces table during these operations.

Issues Fixed in DX NetOps 10.0.3

Defect ID	Support Case	Symptom	Resolution Text
DE433457	20064513	IIS causes web page to not work: logo is not displaying on custom report pdf on console after I enable X-Content-Type-Options" value="nosniff".	Enable X-Content-Type-Options" value="nosniff" and change the image format jpg to gif. The report now displays the logo with nosniff.
DE428904	20050972	Deleting a router from Enable Interfaces page, deletes router from all the harvesters.	Code changes are made to deletes router only from the respective harvester.
DE427756	20037583	Egress Multicast data is not shown on the input interface, even if the input interface is non-zero.	Egress Multicast data is now shown on the input interface as well when the input interface is non-zero.
DE413121, DE424390	01336769, 20022801	Some devices do not get refreshed automatically.	Devices get priority properly to get refreshed periodically.
DE435729	20077629	Merge Interface rest API gives "start time and end time not available" response while merging two interfaces.	Code changes are done to return an appropriate response while merging two interfaces.

DE436715	20090739	Values are changed for SNMP profiles during full sync in CA PM, if the privacy protocol is AES256_3DES.	Code changes are made to show the correct value after full sync with CA PM when the privacy protocol is for AES256_3DES.
DE433120	20105234	On Linux, reaper service is down after harvester installation. The reaper reads incorrect DB password from ini file due to windows line encodings.	As part of the fix, the dos2unix command is applied to the DBUsers\ReporterAnalyzer.ini file.

Issues Fixed in DX NetOps 10.0.2

Defect ID	Support Case	Symptom	Resolution Text
DE405363	01259410	SNMP information like interface speed is not shown in 'Enable Interfaces' page for Huawei devices.	The 'Enable Interfaces' page for Huawei devices now displays the SNMP information.
DE415974	01352611	• Harvester service stops processing incoming flows if mask value from netflow is greater than 128. • Harvester service stops processing incoming flows if any srcIpv6/ dstIpv6 host address from netflow is received as ::ffff:xx.xx.xx.xx(ipv4 mapped ipv6 address).	Harvester service runs successfully and processes all the incoming flows.
DE343953	00957440, 00931541, 00960805, 00960786	Query service crashes when multiple reports are running in parallel.	Reports now run in parallel without crashing query service.

Issues Fixed in DX NetOps 10.0.1

Defect ID	Support Case	Symptom	Resolution Text
DE343953	00931541, 00957440, 00960786, 00960805,00951251, 01126402, 01255217	Custom reports with ToS Group filters fail with the error "Failed to complete baseline calculation" as report status.	Reports complete successfully without errors.
DE408219	01313998	Unable to apply ToS group filters on Custom reports, hence the report shows unrelated results.	The report now shows relevant results for the ToS Group filter you apply.
DE394610	01224491	Status message in custom reporting tab does not honor the user time zone.	Status message in custom reporting tab now shows date and time relevant to the user time zone.
DE416011	01350365	No data is loaded under Collections Sources in the Anomaly Detector tab.	The Collection Sources in the Anomaly Detector tab now show the appropriate data.

Issues Fixed in DX NetOps 10.0.0

Defect ID	Support Case	Symptom	Resolution Text
DE320627	00861196/ 00869246	On Router Refresh detection, the data is still pointing to Old interfaces.	On Router Refresh detection, the data points to New interfaces.
DE360946	01012887	Column Headers for Host Summary Volume in the custom reports do not match the columns when exported to CSV.	Column Headers for Host Summary Volume in the custom report match the columns when exported to CSV.
DE364215	01083039	Search in Add Interface filter page of Custom reports does not work properly when the interface list is expanded.	The search works properly even when the interface list is expanded.
DE364955	01087939	Column Headers do not match for Protocol reports when exported to CSV.	Column Headers matches the Protocol reports when exported to CSV.
DE367156	01101644	The pages for Physical & Virtual Interfaces, Enable Interfaces loads slowly.	The pages for Physical & Virtual Interfaces, Enable Interfaces loads faster.
DE369513	01111560	Flow Statistics page shows that router reboots more than the actual.	Flow Statistics page shows the router reboot correct count.
DE369591	01113363	The AdminTrapsInitial.asp page is vulnerable to scripting attacks.	The fields are now encrypted to avoid the scripting attacks.
DE373591	01132632	The Conversation charts in CA PC error out due to RIB query failures.	The Conversation charts load properly in CA PC.
DE381174	01108921	The charts in CA NFA show data at a higher rate for Viptela devices.	The data for Viptela devices is shown correctly.
DE386567	01195069	CA NFA no longer detects the proper sampling rate for the router.	CA NFA detects and honors the proper sampling rate for the routers.
DE391191	01070997	Data gaps are seen in the flow statistics page.	Data gaps are not seen now.
DE394250	01245443	CA NFA 9.5.0 shows less data for some routers, compared to the data for the same routers in CA NFA 9.3.8.	CA NFA shows correct data for all the routers.

Getting Started

DX NetOps gives you an enterprise-wide view into the composition of traffic on every link and helps you detect threatening traffic patterns in the making.

It provides network traffic analysis with real-time visibility into the traffic throughout your enterprise. Network groups can quickly identify the source of performance problems, validate the impact of planned and unplanned changes within the network, and avoid unnecessary WAN costs. In addition, management can make accurate decisions regarding cost reduction, capacity planning, troubleshooting, and network traffic analysis across the enterprise. You can access as much as one year of flow data for your entire network.

DX NetOps may be integrated as a data source for CA Performance Center (CA PC) or CA NetQoS Performance Center (CA NPC). The Performance Center Console displays report data from DX NetOps and any other programs that are integrated as data sources.

Welcome to the TechDocs Platform

The TECH DOCS PORTAL is the sole source for DX NetOps documentation.

Is there a recommended web browser?

Use Chrome or Firefox for the best experience. Internet Explorer 8 or higher is also supported.

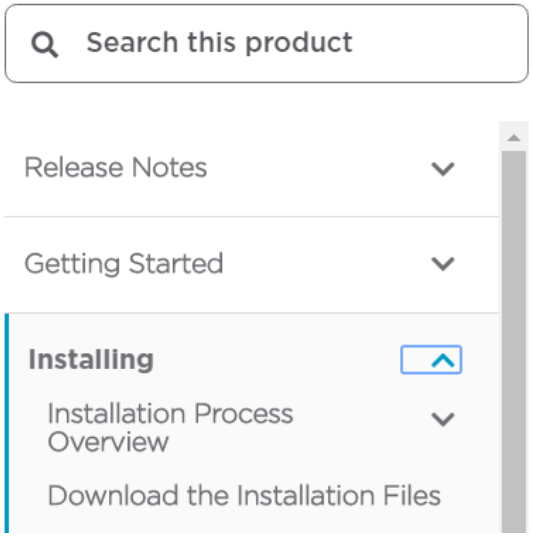
What is the home page?

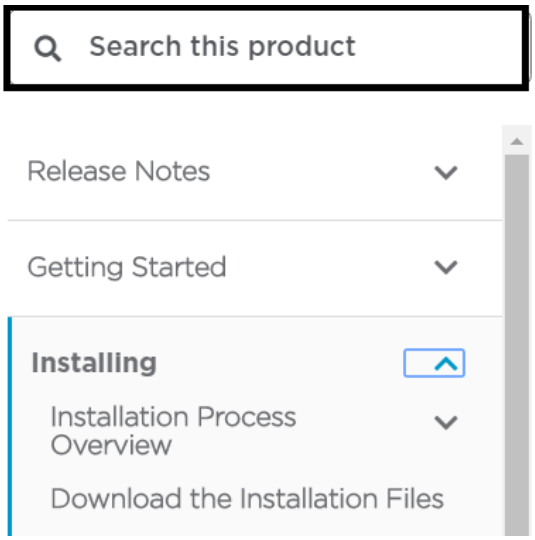
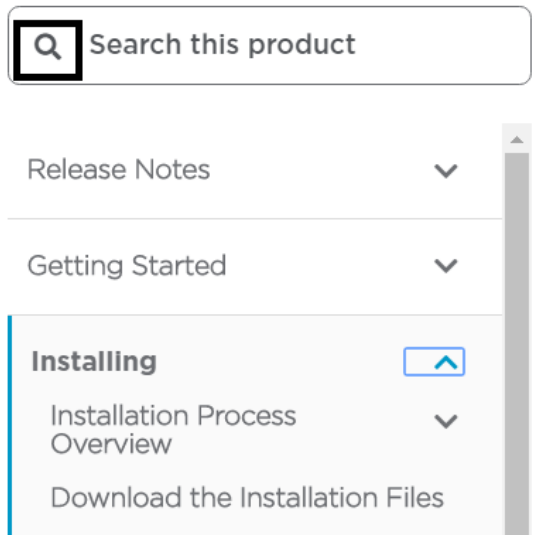
The home page is a front-page to quickly get you to the information you want:

- Search for information
- Browse a table of contents
- Click navigation tiles to key sections
- Switch between versions or languages
- Export the whole information set to PDF
- View live announcements

How do I find content?

The search box on top of the table of contents in the left tree pane searches this product only. You can use wildcards, such as "?" for single characters and "*" for undefined multiple characters, and matched phrase or excluded term searches.

Search	Here
Left explorer tree (like a Table of Contents)	

"Search for" field Returns five topics that best match the search term.	
Advanced Click the magnifying glass icon to list all topics within this product that include the search term.	
Partial word searches	For partial word searching, use: • **CTRL-F** (contents within an article) • Your favorite search engine (limit search to "broadcom.com" for best results)

How do I create printable or offline output?

From the upper right corner of a page, click **PDF** to download printable documents of the pages you are viewing. To download the sections or entire documentation as PDF, navigate to the starting page of the section or documentation home page and download the PDF.

How do I switch to another version or language?

Click **Versions** to select the documentation for a different version.

Click **Languages** to select the documentation in a supported language.

How do I switch to another product?

Click the TECHDOCS link in the upper left corner to return to the root TECH DOCS PORTAL page. From there, you can select any product documentation set from the product drop-down list.

How do I add ask a question or post my own advice on a page?

We want to hear your comments. Click in the **Write a comment** section at the bottom of any page to enter your thoughts, technical question, or opinion. Like a forum, all viewers can see what you post as a comment, and CA Technologies content authors are notified so that they can quickly respond. If you are logged in with your CA credentials, your log in name is posted with your comment. If you are not logged in, enter the displayed characters to post anonymously.

How do I give general feedback?

Use the additional features at the bottom of the page to rate the usefulness of the page or to make suggestions to help us improve it. You can also rate and comment on the overall wiki space.

Introduction to CA Network Flow Analysis

DX NetOps provides network traffic analysis with real-time visibility into the traffic throughout your enterprise. You can access as much as one year of flow data for your entire network.

DX NetOps gives you an enterprise-wide view into the composition of traffic on every link and helps you detect threatening traffic patterns in the making. Network groups can quickly identify the source of performance problems, validate the impact of planned and unplanned changes within the network, and avoid unnecessary WAN costs. In addition, management can make accurate decisions regarding cost reduction, capacity planning, troubleshooting, and network traffic analysis across the enterprise.

DX NetOps can be integrated as a data source for either CA Performance Center or CA NetQoS Performance Center, whichever program your enterprise uses. The Performance Center Console displays report data from DX NetOps and any other programs that are integrated as data sources.

This documentation describes administration tasks that you perform in the DX NetOps console. Other administration tasks, such as the management of users, roles, permissions, SNMP profiles, and some types of groups, might be performed in the Performance Center Console.

Install Performance Center and register DX NetOps as a data source so you can use all the features. You can access Performance Center from the DX NetOps console as soon as you register the product.

> **NOTE**
> The term *Performance Center* refers to CA Performance Center and CA NetQoS Performance Center collectively. Program-specific page names or functions could be identified by the full program name or acronym, which is *CA PC* for CA Performance Center and *CA NPC* for CA NetQoS Performance Center.

Capabilities of CA Network Flow Analysis

DX NetOps capabilities help you analyze network traffic and make informed decisions about resolving issues. The capabilities described in the following list can give you valuable data for capacity planning, troubleshooting, and traffic analysis.

- Identify network traffic that exceeds a specified threshold so you can manage your network proactively.
- Identify bandwidth requirements for applications and users so you can evaluate network capacity precisely.
- Immediately identify the interfaces, hosts, and applications that generate the most traffic in your enterprise. This information is essential for short-term and long-term troubleshooting.
- View the impact of application rollouts on WAN links and measure application traffic growth using protocol-level trend analyses, application baseline trend comparisons, and percent-growth tables. These analyses help you make more informed infrastructure investments.
- Pinpoint the exact cause of a network problem by examining 100 percent of all NetFlow and IPFIX traffic from the last four hours.
- Review automatic alerts and detailed reports so you discover network problems quickly.
- Design and run reports that are based on criteria that you select.
- View real-time NetFlow and IPFIX monitoring reports and alarms for every interface on the network for past 30 days with 1-minute granularity.
- Establish baselines for protocol and flow data so you can compare current data with past performance.
- Analyze trends in applications, hosts, and conversations per class of service. This information helps you optimization your network infrastructure for application performance.
- Drill into raw flows per interface to assist with troubleshooting.
- Review trend settings for historical data and for future projections to perform more effective capacity planning.

Introduction to Performance Center

DX NetOps can be integrated as a data source for either CA Performance Center or CA NetQoS Performance Center, but it is not required. The Performance Center Console displays report data from DX NetOps and any other programs that are integrated as data sources.

When your administrator adds DX NetOps as a data source for Performance Center, several changes occur:

- The NFA console banner contains a link to the Performance Center Console (either *CA PC* or *CA NPC*).
- If your user account enables it, you can see DX NetOps views in customizable dashboards and context pages in the Performance Center Console.
- Properly credentialed users also can drill in to details in the NFA console from views in Performance Center.

Notes:

- The term *Performance Center* refers to CA Performance Center and CA NetQoS Performance Center collectively. Program-specific page names or functions might be identified by the full program name or acronym, which is *CA PC* for CA Performance Center and *CA NPC* for CA NetQoS Performance Center.
- To learn more about Performance Center views and customization options, see the wiki for your Performance Center.

The CA Network Flow Analysis User Interface

The primary DX NetOps user interface is the NFA Console, a web interface that you use to view the collected data, and perform administrative tasks.

DX NetOps analyzes, formats, and displays collected data in the following page views:

- **Enterprise Overview Page**

Summarizes information about the interfaces that exceed or are close to exceeding a utilization threshold that product administrators establish. The **Enterprise Overview** page also shows the top interfaces, protocols, and hosts for your enterprise. The timeframe for this report is the most recent 24 hours available to the NFA console.

- **Interfaces Page**

Lists available routers and their component interfaces. Click an interface to drill down to more detailed information about the interface. You can select a timeframe and a report type.

- **Custom Reporting Page**

Displays the current list of defined Custom Reports and provides options for running, creating, editing, and managing the reports. You can use a wizard to create Custom Reports. Several report types are available, including interface, protocol, ToS, host, and conversation. You can combine the types to produce the results you want. You can save and run these reports on demand or on a schedule.

- **Flow Forensics Page**

Displays the current list of defined Flow Forensics reports and provides options for running, creating, editing, and managing the reports. You can open the **Report Settings** dialog to create a Flow Forensics report. You can add filters and can specify the data collection timeframe. For example, you can analyze protocols, hosts, or conversations on your network. You can export the data on the screen to a file in comma-separated value (CSV) format. You can save and run these reports on demand or on a schedule.

- **Analysis Page**

Displays the current list of defined Analysis reports and provides options for running, creating, editing, and managing the reports. You can use a wizard to create an Analysis report. An Analysis report lets you establish a threshold for comparing collected data, identifying potential bottlenecks, anomalies, and viruses. For example, you could use an Analysis report to see which interfaces exceeded 70 percent utilization over a certain timeframe. You can save and run these reports on demand or on a schedule.

- **Site to Site Page**

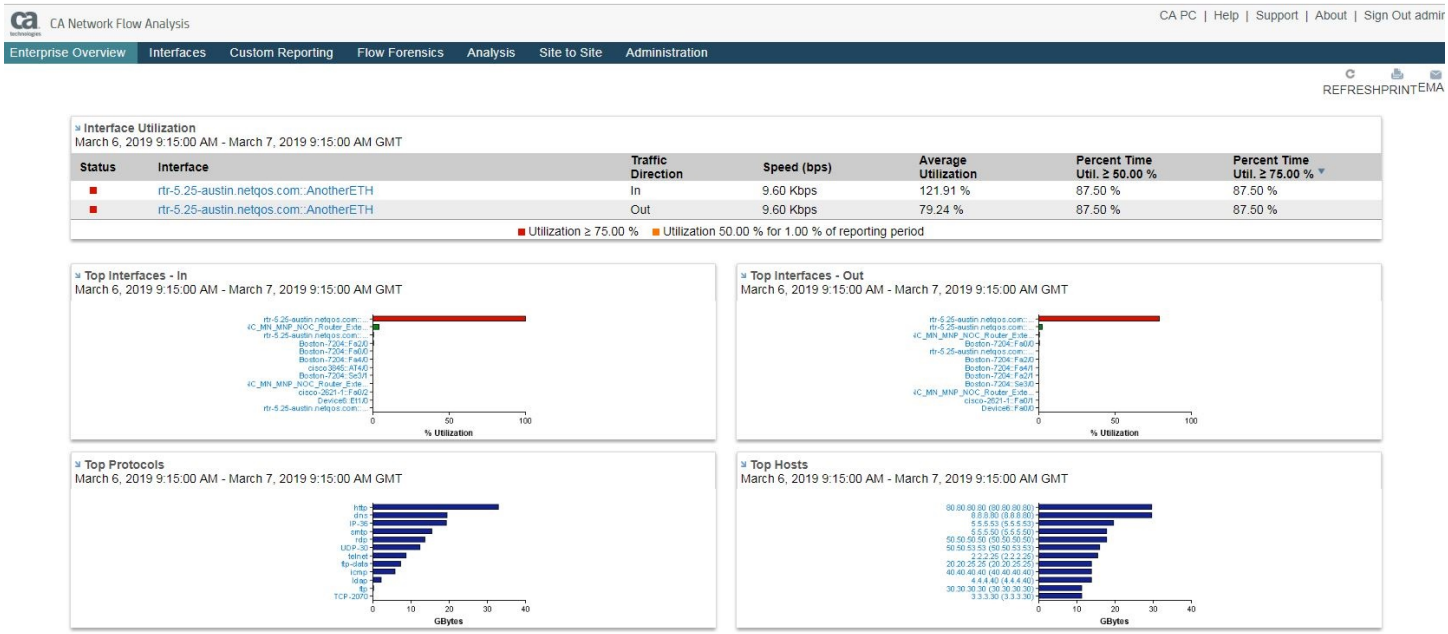
Displays the current list of defined Site to Site reports and provides options for running, creating, editing, and managing the reports. Site to Site reports enable you to view volumes of data between two or more sites. Sites can be defined as a collection of subnets that can also be discontiguous. You can save and run these reports on demand or on a schedule.

- **AdmInistration Page**

Provides access to administrative tasks for Administrators of DX NetOps.

Enterprise Overview Page

The **Enterprise Overview** page shows a set of built-in views of real-time network performance data, to help identify high-level patterns, anomalies, trends, and issues with network traffic. To display the **Enterprise Overview** page, click **Enterprise Overview** in the NFA console menu.



Depending on your access settings, the **Enterprise Overview** page may contain data for some or all of the following views:

- **Interface Utilization**: A table of data about the interfaces with the highest utilization levels
- **Top Interfaces - In**: Utilization and volume information for the interfaces that have high volumes of inbound traffic
- **Top Interfaces - Out**: Utilization and volume information for the interfaces that have high volumes of outbound traffic
- **Top Protocols**: Total volume of traffic associated with the most heavily used protocols
- **Top Hosts**: Traffic volumes of the most active hosts (inbound, outbound, and total traffic)

The **Enterprise Overview** displays data for the most recent 24-hour period that is available. The reporting timeframe is noted under the title of each view.

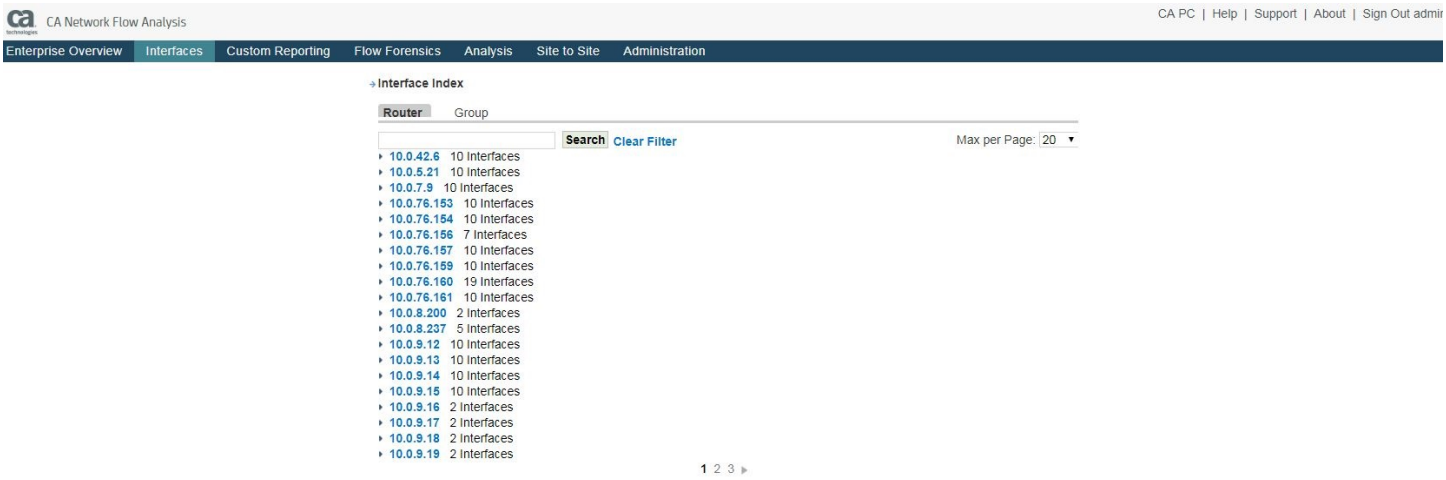
Available Actions for the Enterprise Overview Page

You can perform the following actions on the **Enterprise Overview** page, provided that you have the required access:

- Display data about each interface, protocol, or host in bar graphs by opening Tooltips.
- Open additional reports about each interface, protocol, or host by clicking the bars or the blue drilldown links.
- In the **Interface Utilization** view, customize the status thresholds.
- Export the data for each view to a .CSV file.
- Update the data by using the **Refresh** icon.
- Print the **Enterprise Overview** page to a PDF file (Administrator or Power User accounts only).
- Email PDF files of the current views or schedule PDFs to be sent out to one or more recipients on a schedule (Administrator or Power User accounts only).

Interfaces Page

You can use **Interfaces** page reports to review traffic for specific interfaces, to identify the cause of network problems or to anticipate upcoming problems. You can get an overview of network traffic from the **Enterprise Overview** reports, then get more detail from the **Interfaces** page reports.

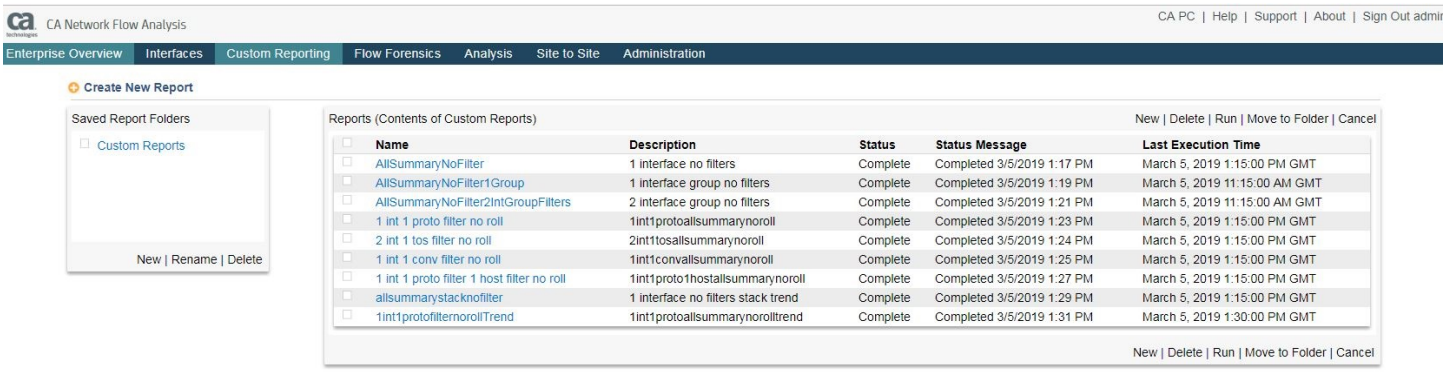


Custom Reporting Page

Custom Reports can answer specific technical and business questions in your environment, such as the following questions:

- Which applications are used most heavily in the regional offices?
- What is the total volume of traffic for global operations?
- Does the new data center have the capacity to handle additional servers?

To get started, select **Custom Reporting** from the NFA console menu.



The **Create New Report** page opens, which lists the existing reports. If you have the required permissions, the page contains functions for creating, managing, and running Custom Reports. You can generate updated versions of reports, add new report definitions, and change report settings.

The **Create New Report** page includes the following functions:

Saved Report Folders:

- *New:* Create a report folder.
- *Rename:* Change the name of a report folder.

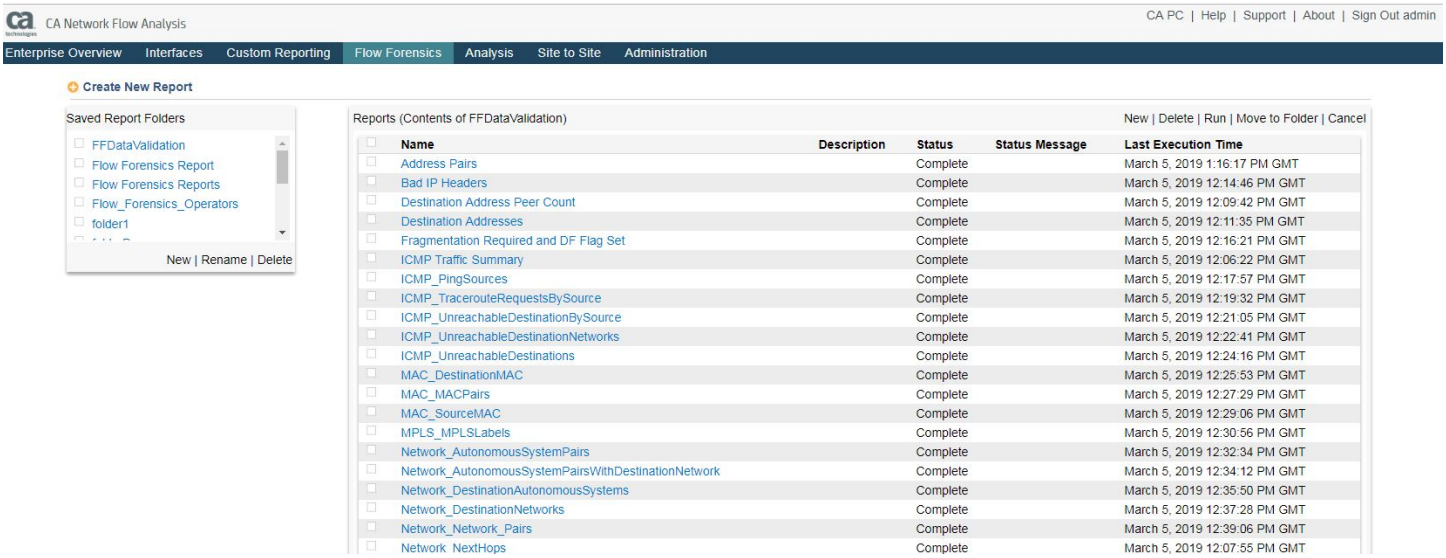
Reports:

- *New:* Create a Custom Report definition.
- *Run:* Regenerate the data for one or more Custom Reports.
- *Move to Folder:* Change the parent folder for one or more report definitions.
- *Cancel:* Stop one or more reports from running.

Flow Forensics Page

You can use Flow Forensics reports to leverage the detailed data that Flow Forensics provides, such as reporting data from multiple interfaces in real time.

To open the Flow Forensics page, click **Flow Forensics** in the NFA console menu.



When you click the **Flow Forensics** tab, the **Saved Report Folders** pane is displayed on the left and the existing reports are displayed on the right.

You can perform the following tasks in the **Saved Report Folders** pane: create folders, rename folders, and delete custom folders. You cannot delete the default **Flow Forensics Reports** folder, which is built in to DX NetOps. Create additional folders to provide more extended organization for your Flow Forensics reports.

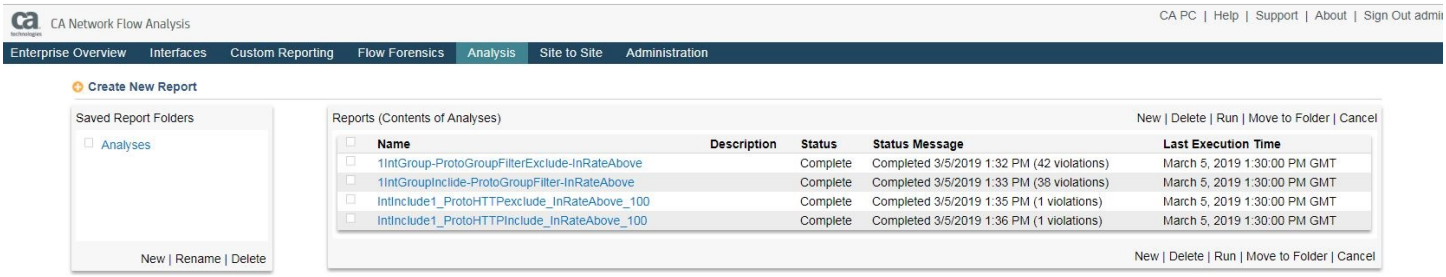
The right pane displays the Flow Forensics report definitions in the currently selected folder. Use the links at the top and bottom of the pane to delete or run selected reports, as well as to create or move reports.

Analysis Page

You can troubleshoot problems as they occur by using Analysis reports to identify issues before users in your environment are adversely affected.

An Analysis report is designed to compare collected network data to a threshold so you can identify potential bottlenecks, anomalies, and viruses. Analysis reports help you identify potential problems before they become serious issues. You can schedule these reports to run regularly, which means you can continually analyze your network traffic for potential issues.

To open the Analysis page, click **Analysis** in the NFA console menu.



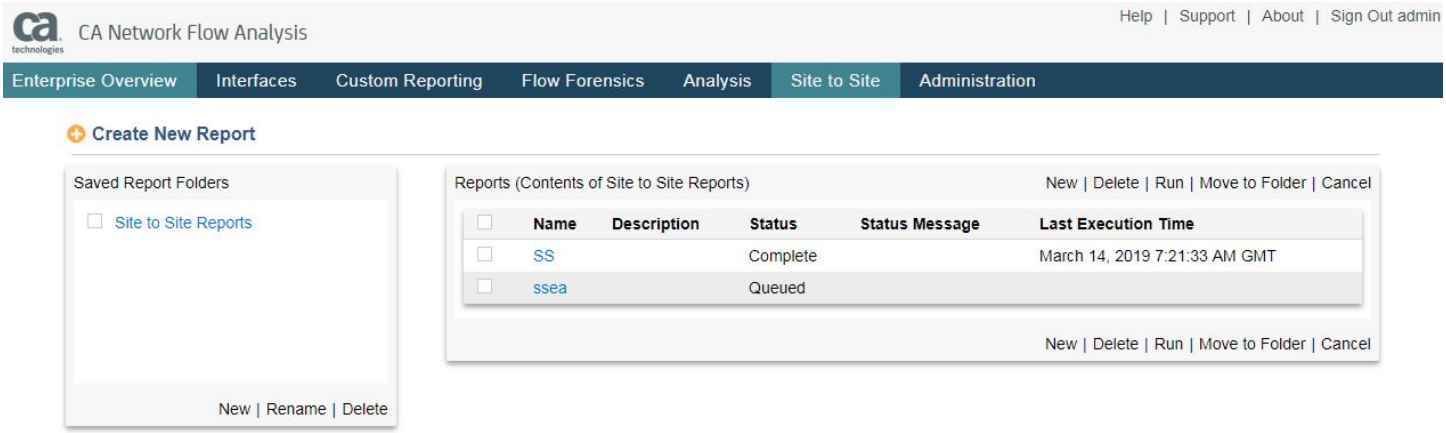
Site to Site Page

Site to Site reports enable you to view volumes of data between two or more sites. A site may be defined as a collection of subnets that can also be discontiguous.

Use Site to Site reports to compare the following between pairs of sites:

- Bytes in
- Rate in
- Bytes out
- Rate out

To get started, select **Site to Site** from the NFA console menu.



Introduction to Flow Cloner

You can use the Flow Cloner tool to forward flow data from a flow-enabled Harvester to another collection device, such as a Harvester in a different deployment. For example, the Flow Cloner could send flows to an Intrusion Detection System (IDS). By using the Flow Cloner, you can send the same data to two collection devices without burdening your routers with sending the data twice.

The Flow Cloner listens for packets in promiscuous mode, then forwards them to the IP addresses that you designate. In this mode, the Flow Cloner passes the packets along to any other process that is listening for them. A Harvester that is co-installed with a running Flow Cloner sees all the packets that are destined for it.

Installing

This section is for systems administrators who are installing DX NetOps. When installation is complete, you will have set up the following:

- NFA Console
- NFA Harvester
- (Optional) Flow Cloner

NOTE

More information:
(Optional) CA Anomaly Detector

System Recommendations and Requirements

Installation Process Overview

In a *stand-alone* deployment, a single server hosts the NFA console and the Harvester.

In a *distributed* deployment, DX NetOps components are distributed among multiple servers.

Workflow for Installing a Stand-Alone Deployment

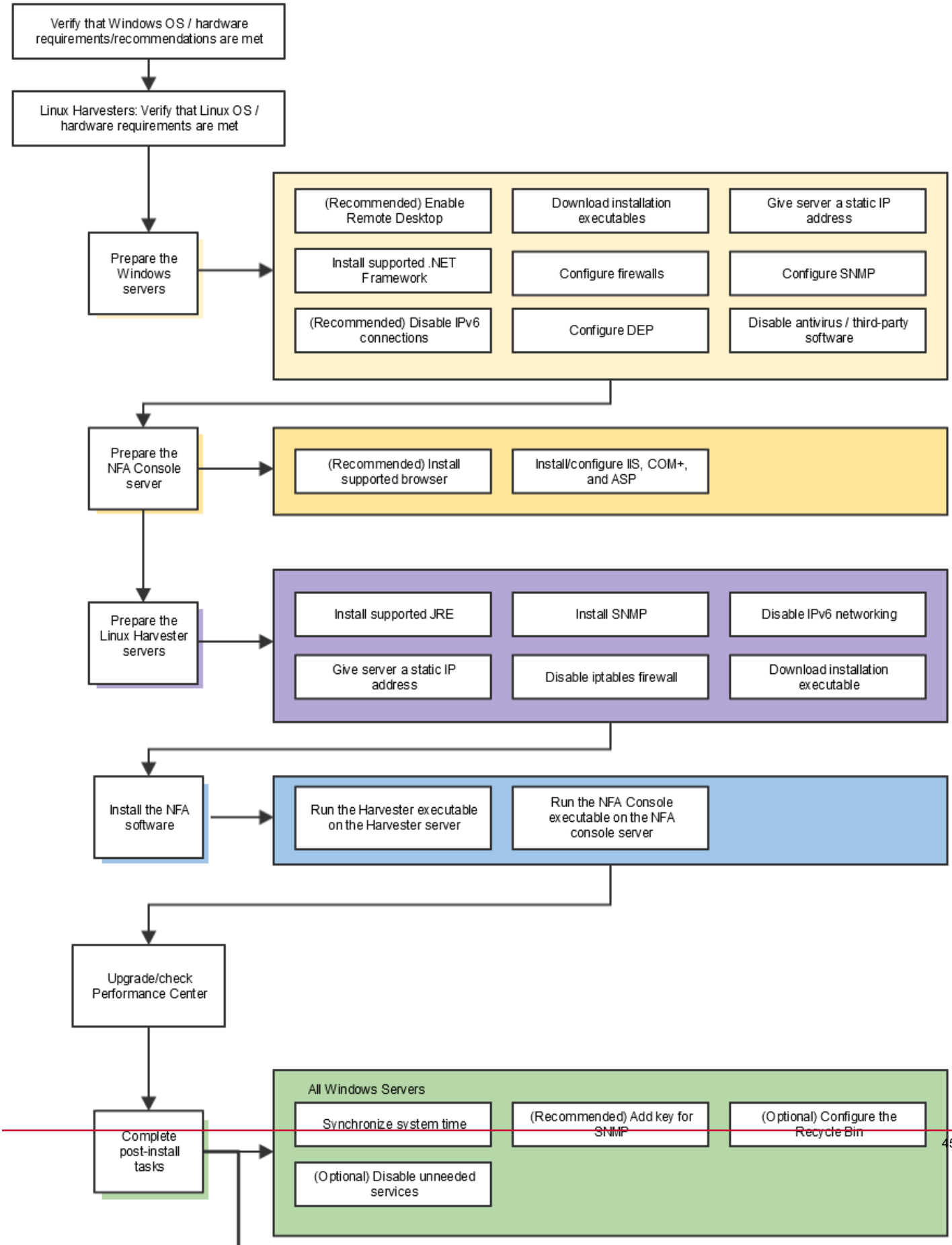
Use the following diagram as a general checklist for installing a stand-alone deployment of DX NetOps.

Figure 95: NFA stand-alone

Workflow for Installing a Distributed Deployment

Use the following diagram as a general checklist for installing a distributed deployment of DX NetOps.

Figure 96: NFA Distrib



Download the Installation Files

Copy the current version of the installation/upgrade files to the installation server so you are certain to have access to the files.

1. Get the files for installing or upgrading the components:
   a. Log in to support portal.
   b. Navigate to the **Download Management**.
   c. Click **Network Flow Analysis**.
   d. Select **CA Network Flow Analysis MULTI-PLATFORM**, x.x.x from Release drop-down list.
   e. Click DOWNLOAD NOW.
   The ISO file is downloaded.

   NOTE
   An ISO file is an archive file that contains the contents of an optical disk. Each one of the available ISO files contains the files for installing or upgrading the component named in the file link.
2. Perform one of the following tasks:
   – Burn the ISO files to a CD-ROM or DVD.
   – Extract the contents of the ISO files by using an ISO image software application. Many free ISO image applications are available.
3. Extract the appropriate files to the installation servers:

Stand-alone servers:

- – • CA Network Flow Analysis
    - `NFHarvesterSetupx.x.x.exe`
    - `RAConsoleSetupx.x.x.exe`
  - CA Anomaly Detector
    - `ADSetupx.x.x.exe`

Distributed deployments:

1. – • Windows Harvester servers in distributed deployments:
    - `NFHarvesterSetupx.x.x.exe`
  - Linux Harvester servers in distributed deployments:
    - `NFHarvesterSetupx.x.x.bin`
  - CA Anomaly Detector
    - `ADSetupx.x.x.exe`
  - NFA console servers in distributed deployments:
    - `RAConsoleSetupx.x.x.exe`

You can install or upgrade the software locally or remotely.

System Recommendations and Requirements

This section describes the hardware and operating system recommendations and requirements for the DX NetOps component servers. Configure and secure the operating system as described here.

We tested the product with the following hardware configuration. Your requirements might vary depending on the characteristics and volume of interfaces, applications, and operators in your network.

NOTE

- The recommended specifications described here apply to both physical and virtual deployments. The specifications represent an optimal configuration. You can run DX NetOps successfully on configurations that do not meet these specifications, although your performance might vary.
- Performance is improved by running the software and the operating system on separate drives. However, it is possible to install and run the software and operating system on the same drive.
- For the following per harvester recommendations, the Autonomous System (AS) is set to 50 and Type of Service (TOS) is set to 50 for the flow data (Flows per minute, Interfaces, Routers). For high variability of AS and TOS numbers Flows per minute has to be reduced.

Large Scale Environments

An example of a large scale environment:

- 100 million or greater flows (NetFlow v9**)
- 70,000 interfaces sending flows
- 1,000 or greater routers sending flows
- 24 harvesters

**This is based on v9 NetFlows, which included the following data:

(`srcaddr` , `srcport` , `dstaddr` , `dstport` , `port` , `input` , `output` , `dpkts` , `doctets` , `duration` , `tos` , `TCP_FLAGS` , `SRC_MASK` , `DST_MAST` , `IN_SRC_MAC` , `OUT_SRC_MAC` , `IN_DST_MAC` , `OUT_DST_MAC` , `icmp_type` , `SRC_VLAN` , `DST_VLAN` , `mpls_label_1` , `mpls_label_1` , `mpls_top_label_type` , `mpls_top_label_ip_addr` , `mpls_label_10` , `nexthop` , `src_as` , `dst_as`)

Architecture - 2 Tier

High-level Recommendations for Harvesters

- Windows 2016
- Windows 2012R2
- SSD storage - especially for Harvesters with greater than 4 million flows / 5000 interfaces / 50 routers
- 8 CPUs (2 quad-core CPUs) - 2.8 GHz
- Reserved resources (CPU and memory) in VM environment
- Max Disk Read / Write latency kept to < 10 ms
- Do not install DX NetOps on the operating system drive (for example, on `Windows` , not on `C:`)

Per Harvester recommendations, based on Flow Rate, Interfaces, Routers

Flows per minute	Interfaces	Routers	Memory	Disk Read IOPS needed	Disk Write IOPS needed	Disk Space needed***
9000000	500	5	16 GB	approx. 300	approx. 200	1.5 TB
8000000	1000	10	16 GB	approx. 300	approx. 200	1.5 TB
7000000	2000	20	16 GB	approx. 300	approx. 200	1.5 TB
4000000	5000	50	12 GB	approx. 100	approx. 200	1.0 TB
3000000	5000	100	12 GB	approx. 50	approx. 150	750 GB
2000000	5000	100	12 GB	approx. 50	approx. 150	750 GB

NOTE

Linux supports 9000000 flows per minute for IPv4 and 8000000 flows per minute for IPv6.

***This includes default values to limit data retention

- Raw NFA file storage - 24 hours
- 1-minute data - 30 days
- 15-minute data - 365 days

High-level Recommendations for Console

- Windows 2016
- Windows 2012R2
- 4 CPUs - 2.8 GHz
- 8 GB memory
- Reserved resources (CPU and memory) in VM environment
- Do not install DX NetOps on the operating system drive (for example, on `Windows` , not on `C:`)

Medium Scale Environments

An example of a medium scale environment:

- 16 million flows (NetFlow v9**)
- 20,000 interfaces sending flows
- 200 routers sending flows
- 4 harvesters

**This is based on v9 NetFlows, which included the following data:

(`srcaddr` , `srcport` , `dstaddr` , `dstport` , `port` , `input` , `output` , `dpkts` , `doctets` , `duration` , `tos` , `TCP_FLAGS` , `SRC_MASK` , `DST_MAST` , `IN_SRC_MAC` , `OUT_SRC_MAC` , `IN_DST_MAC` , `OUT_DST_MAC` , `icmp_type` , `SRC_VLAN` , `DST_VLAN` , `mpls_label_1` , `mpls_label_1` , `mpls_top_label_type` , `mpls_top_label_ip_addr` , `mpls_label_10` , `nexthop` , `src_as` , `dst_as`)

Architecture - 2 Tier

High-level Recommendations for Harvesters

- Windows 2016
- Windows 2012R2
- 8 CPUs (2 quad core CPUs) - 2.8 GHz
- Reserved resources (CPU and memory) in VM environment
- Max Disk Read / Write latency kept to < 10 ms
- Do not install DX NetOps on the operating system drive (for example, on `Windows` , not on `C:`)

Per Harvester recommendations, based on Flow Rate, Interfaces, Routers

Flows per minute	Interfaces	Routers	Memory	Disk Read IOPS needed	Disk Write IOPS needed	Disk Space needed***
4000000	5000	50	12 GB	approx. 100	approx. 200	1.0 TB
3000000	5000	100	12GB	approx. 50	approx. 150	750 GB
2000000	5000	100	12 GB	approx. 50	approx. 150	750 GB

***This includes default values to limit data retention

- Raw NFA file storage - 24 hours
- 1-minute data - 30 days
- 15-minute data - 365 days

High-level Recommendations for Console

- Windows 2016
- Windows 2012R2
- 4 CPUs - 2.8 GHz
- 8 GB memory
- Reserved resources (CPU and memory) in VM environment
- Do not install DX NetOps on the operating system drive (for example, on `Windows` , not on `C:`)

Small Scale Single Box Environments

An example of a small scale environment:

- 2 million flows (NetFlow v9)
- 5,000 interfaces sending flows
- 100 routers sending flows

Architecture - Stand-Alone (Single Box)

High-level Recommendations for Stand-Alone (Single Box)

- Windows 2016
- Windows 2012R2
- 4 CPUs - 2.8 GHz
- 16 GB memory
- Reserved resources (CPU and memory) in VM environment
- Max Disk Read / Write latency kept to < 10 ms
- Do not install DX NetOps on the operating system drive (for example, on `Windows` , not on `C:`)

Single Box recommendations, based on Flow Rate, Interfaces, Routers (includes the Console)

Flows per minute	Interfaces	Routers	Memory	Disk Read IOPS needed	Disk Write IOPS needed	Disk Space needed***
2000000	5000	100	16 GB	approx. 50	approx. 150	1 TB

***This includes default values to limit data retention

- Raw NFA file storage - 24 hours
- 1-minute data - 30 days
- 15-minute data - 365 days

NOTE
More information:

View Flow Statistics

Windows Servers

Verify that your hardware meets the specifications that are noted here.

Setting or Component	Description
Operating System	Microsoft Windows Server 2012 R2 Standard Edition on a 64-bit processor Microsoft Windows Server 2016 Standard Edition on a 64-bit processor
Operating System updates	Latest Service Pack and all important updates installed. Install only important Windows updates and service packs. Do not install an unsupported Web browser.
Disk space	C: drive with 40 GB of available space for the operating system We recommend installing DX NetOps on a separate drive that is dedicated to DX NetOps. Verify that the drive contains the following disk space available: • 41 GB for the installation or upgrade files • NFA console or stand-alone server: 200 GB or more available for data • Harvester server: 1 TB of available space for data
CPU	• NFA console or stand-alone server: One 2.8-GHz quad core processor • Harvester server: Two 2.8-GHz quad core processors (8 CPUs)
Memory	• NFA console: 12-GB RAM • Stand-alone server: 16 GB RAM • Harvester server: 12 GB or 16 GB RAM based on flow rate (link)
Hard drives	• NFA console or stand-alone server: Three 146-GB hard drives in RAID5 configuration • Harvester Server: Six 300-GB hard drives in RAID5 configuration For highest performance, use SSD drives. You can step down to 15,000 RPM or 10,000 RPM drives, if you can tolerate less performance. We do not recommend going lower than 10,000 RPM.
Ports	1-Gb Ethernet port Run the following command on harvester, if the harvester is having multiple NIC's. Before running, ensure you have the correct IP address of the Harvester that matches in the Administration -> Harvester page of NFA `Mysql -P3308 -D harvester -t -e "update parameter_descriptions set defaultValue='x.x.x.x' where parameter='harvesterIPAddress';"` Restart the CA NFA Harvester service. If you enter the wrong IP address in the above command it results in devices being created on Harvester IP addresses that you did not want. Make sure you have the correct Harvester IP address before running the commands.
Screen resolution	Minimum display resolution of 1024x768 (XGA)

Web browser	Microsoft Internet Explorer version 11 Mozilla Firefox (latest version) Google Chrome (latest version) • NFA console or stand-alone server: Browser optional • Clients that log in to the NFA console: Browser required • Some other browsers or browser versions might work for logging in from a client, but they have not been tested. **Notes:** The following browser versions have known issues: • Microsoft Internet Explorer version 9 - The **Log In** dialog and screen might have display artifacts: • The **Log In** button text might be white, although the button is functional. The **Log In** button is located in the bottom right corner of the dialog and has a blue box around it. If you position your cursor inside the box, the cursor changes to Hand mode. The login function is active whenever the cursor is in the Hand mode. The following browser versions have known issues: • The screen behind the **Log In** dialog might not be rendered in a uniform blue color. • Microsoft Internet Explorer v10 - If you save reports as PDF files, or set up scheduled reports to send PDF files, the PDFs are not searchable. The PDFs are rendered as pictures, which do not include searchable text. • Working in the CA Performance Center Console: Use Internet Explorer with Compatibility View turned off. You can work in the NFA console with Compatibility View turned on or off. Changing the Compatibility View option for the current session: • If Internet Explorer Developer Tools installed, press **F12** on your keyboard. Select the option that does not contain the phrase "Compatibility View".
Features, settings, and additional software	• .NET Framework 4.5, which is enabled by default in the Windows Server 2012 operating system. • .NET Framework 4.6.2, which is enabled by default in the Windows Server 2016 operating system. • **For NFA 10.0.0 and 10.0.1 release:** Java Runtime Engine (JRE) 1.8.0_192, which is installed automatically during the installation or upgrade of the DX NetOps software. • **For NFA 10.0.2 release:** AdoptOpenJDK JRE 1.8.0_212 • **For NFA 10.0.3 and 10.0.4 releases:** AdoptOpenJDK JRE 1.8.0_232

NOTE
More information:

- For more specific recommendations on large, medium, and small-scale environments, see System Recommendations and Requirements.

Prepare the Windows Servers

Before you begin the installation, verify that the following conditions are met. Failure to meet these requirements and recommendations can result in data loss, increased downtime, software conflicts, or installation failure.

Follow these steps:

1. Log in to a Windows server as a user who is a member of the Administrators group.

2. Download the DX NetOps Prerequisite Tool.
   The tool ensures that all the prerequisites are enabled or disabled on Windows Server 2016 server.

   NOTE

   For **Windows Server 2012 R2 server**, ensure that you install .Net 4.6 before executing this prerequisite tool.
3. Extract the contents of the zip file.
4. Run the check script. No changes are made to the system currently. The script is only a check.
5. When the check script finishes, check the output for information and instructions. If the check script determines that changes are necessary, it generates a fix script to make those changes.
6. Run the fix script, if necessary.
7. Launch the fix script in a **Command Prompt** window, so that the output is visible for review.
8. When finished, check the output for instructions on rebooting and re-running the check script.
9. When there no further changes are necessary; Install the Software.

(Optional) Verify the Prerequisites Manually

If anything fails to work while using the tool, or if it is preferable to take care of the prerequisites manually, see the following pages:

Additional Requirements

- If you use Performance Center, verify that a supported version is installed in your deployment. Install a supported version of Performance Center on a separate server after you install DX NetOps
- When you apply Windows updates, restart all servers to ensure that the updates are applied.
- For a Harvester running on a Windows VM with greater than 4 million flows/minute, make the following environment changes:
  – Navigate to **Device Manager, Network adapters, vmxnet3 Ethernet Adapter, Properties, Advanced** tab.
  – Configure the following properties on the VMXNET 3 interface.

Property	Value
Receive Side Scaling	Enabled
Small Rx Buffers	8192
Rx Ring 1 Size	4096

Verify the following requirements on each Windows server:

Stand-Alone Server	**Distributed NFA Console Server**	**Distributed Harvester Server**
If possible, meet Windows hardware recommendations.		
Meet Windows operating system requirements.		
Configure Regional settings.		
Assign a static IP address to each server. Set the Harvester server IP address to match the flow export destination that is assigned to each router.		
Install the supported version of .NET Framework 3.5 SP1 *.		
Install .NET Framework 4.6.2 *.		
(Recommended) Enable Remote Desktop Connection to allow remote access.		
(Recommended) Install a supported browser **.		
Configure the firewall.		

Install IIS, COM+, and ASP **.
Configure SNMP **.
(Recommended) Disable IPv6 connections.
Configure DEP **.
Disable the following third-party software: Antivirus, server monitoring, and maintenance software until the installation is complete. If you enable antivirus scans later, exclude the DX NetOps installation path and its subdirectories.

* If this requirement is not met, the installation program either does not open or does not complete successfully.
** If the server fails to pass the check for this requirement, a warning message opens.

Configure Data Execution Prevention (DEP)

Data Execution Prevention (DEP) helps to prevent code executing from data pages. To configure the appropriate DEP policy level:

Follow these steps:

1. Log in as a user who has administrator privileges for DX NetOps.
2. Open the **Control Panel** and click the **System** link.
3. Click the **Advanced system settings** link.
4. Click the **Advanced** tab in the **System Properties** dialog.
5. In the **Performance** area, click **Settings**.
6. Click the **Data Execution Prevention** tab in the **Performance Options** dialog.
7. Select **Turn on DEP for essential Windows programs and services only**.
8. Save your settings and exit:
   a. Click **Apply** in the **Performance Options** dialog.
      A message opens and informs you that you must restart your system to implement the new settings. Click **OK**.
   b. Click **OK** in the **Performance Options** dialog.
   c. Click **OK** in the **System Properties** dialog.
9. (Optional) Restart your system before you install or upgrade the software.
   If you proceed without restarting the system, the prerequisite test displays a warning about the DEP configuration.

Install .NET Framework

In both fresh installation or during upgrade, the current version of DX NetOps requires .NET Framework 4.6.2 on all Windows servers. As an administrator or member of administrator group install the .NET Framework on all of the Windows servers.

Follow these steps:

1. Download the .NET Framework update.
   – Web installer:
     https://www.microsoft.com/en-us/download/details.aspx?id=53345
   – Offline installer:
     https://www.microsoft.com/en-us/download/details.aspx?id=53344
2. Run the installer.
3. Follow the instruction on the installer wizard and complete the installation.

Install IIS, ASP, COM+, and SNMP

Install the following required components on a stand-alone server or NFA console server:

- IIS
- ASP
- IIS 6 Management Compatibility
- COM+ Network Access
- SNMP

> **NOTE**
> Before proceeding with the installation, ensure that vcredist_x86.exe is installed on the server. Please install vcredist_x86.exe from here, if it is not installed.

For Windows 2012 Server, follow these steps:

1. Log in as a user who has administrator privileges for CA Network Flow Analysis.
2. Open **Server Manager**.
3. In the **Dashboard**, select **Add roles and features**.
   If the **Before you begin** page displays, click **Next**.
4. In the **Select installation type** page, select **Role-based or feature-based installation**.
   Click **Next**.
5. In the **Select destination server** page, select **Select a server from the server pool**. Select the appropriate server from the **Server Pool** list, then click **Next**.
6. In the **Select server roles** page, select **Application Server**. Click **Next**.
7. In the **Select features** page:
   a. Expand **.NET Framework 4.5 Features**.
      a. Select **.NET Framework 4.5**, if not already checked.
      b. Select **ASP.NET 4.5**, if not already checked.
      c. Expand **WCF Services**, and select **HTTP Activation**.
   b. Select **SNMP Service**.
      a. Select **SNMP WMI Provider**.
         Click **Add Features** when prompted.
   c. Expand **User Interfaces and Infrastructure**.
      a. Select **Desktop Experience**.
         Click **Add Features** when prompted.
   d. Click **Next**.
8. In the **Application Server** page, click **Next**.
9. In the **Select role services** page:
   a. Select **COM+ Network Access**.
   b. Select **Web Server (IIS) Support**.
      Click **Add Features** when prompted.
   c. Click **Next**.
10. In the **Web Server Role (IIS)** page, click **Next**.
11. In the **Select role services** page:
    a. Expand the **Application Development** section.
       a. Select **ASP**.
    b. Expand the **Management Tools** section.
       a. Select **IIS 6 Management Compatibility**.
    c. Click **Next**.

12. The **Confirm installation selections** page summarizes your actions and displays related messages.
    If you are prompted to specify an alternate source path, click **Specify an alternate source path**.
    a. Specify the path to the SxS Sources on the Windows Install CD, then click **OK**.
13. Install the role services and options you selected:
    a. Click **Install**.
    The **Results** page opens when the installation or upgrade is complete.
    b. (Optional) Click **Print, e-mail, or save the installation/upgrade report**, review the information, then close the page.
    c. Click **Restart Server**.

For Windows 2016 Server, follow these steps:

1. Log in as a user who has administrator privileges for DX NetOps.
2. Open **Server Manager**.
3. In the **Dashboard**, select **Add roles and features**.
   If the **Before you begin** page displays, click **Next**.
4. In the **Select installation type** page, select **Role-based or feature-based installation**.
   Click **Next**.
5. In the **Select destination server** page, select **Select a server from the server pool**. Select the appropriate server from the **Server Pool** list, then click **Next**.
6. In the **Select server roles** page, select **Web Server (IIS)**, and click **Next**.
7. In the **Select features** page:
   a. Expand **.NET Framework 4.6 Features**.
      a. Select **.NET Framework 4.6,** if not already checked.
      b. Select **ASP.NET 4.6**, if not already checked.
      c. Expand **WCF Services**, and select **HTTP Activation**.
   b. Expand **SNMP Service**.
      a. Select **SNMP WMI Provider**.
      Click **Add Features** when prompted.
   c. Click **Next**.
8. In the **Web Server Role (IIS)** page, click **Next**.
9. In the **Select role services** page:
   a. Expand the **Application Development** section.
      a. Select **ASP**.
   b. Expand the **Management Tools** section.
      a. Select **IIS 6 Management Compatibility**.
   c. Click **Next**.
10. The **Confirm installation selections** page summarizes your actions and displays related messages.
    If you are prompted to specify an alternate source path, click **Specify an alternate source path**.
    a. Specify the path to the SxS Sources on the Windows Install CD, then click **OK**.
11. Install the role services and options you selected:
    a. Click **Install**.
    The **Results** page opens when the installation or upgrade is complete.
    b. (Optional) Click **Print, e-mail, or save the installation/upgrade report**, review the information, then close the page.
    c. Click **Restart Server**.

Configure SNMP on Windows Servers

The Simple Network Management Protocol (SNMP) service is required by the Watchdog services. To configure the SNMP service on the Windows servers in your deployment:

Follow these steps:

1. Log in as a user who has administrator privileges for DX NetOps.
2. To display the list of community names for the SNMP service:
   a. Select **Start**, **Administrative Tools**, **Services**.
   The **Services** window opens.
   b. Right-click the **SNMP Service** and select **Properties**.
   The **SNMP Service Properties** dialog opens.
   c. Select the **Security** tab.
3. Verify that the appropriate community name is in the **Accepted community names** list. The default community name is "public".
4. If the appropriate community name is not listed, add it:
   a. Click **Add**.
   The **SNMP Service Configuration** dialog opens.
   b. Set the following options:
      - Community rights: Select **Read Only**.
      - Community Name: Enter **public** or a custom community name. Use the same community name throughout the DX NetOps deployment:
        `snmpd.conf` file on each Linux server
        SNMP service on each Windows server
        Watchdog Settings page of the NFA console
   c. Click **Add**.
   The **SNMP Service Configuration** dialog closes. The **SNMP Service Properties** dialog displays the new name in the **Accepted community names** list.
5. Save your changes and exit:
   a. Click **Apply**, **OK** in the **SNMP Service Properties** dialog.
   Your changes are saved and the dialog closes.
   b. Select **File**, **Exit** in the **Services** window.
   The **Services** window closes.

Disable IPv6 Connections on Windows Servers

This release does not support connections to IPv6-formatted addresses. If connection to IPv6-formatted addresses is enabled, data collection fails.

The instructions are based on the assumption that each server has a single network interface card, which is the recommended configuration.

Follow these steps:

1. Log in as a user who has administrator privileges for DX NetOps.
2. Open the **Network Connections** window:
   a. Select **Start**, **Control Panel**.
   b. Click **Network and Internet**.
   c. Click **Network and Sharing Center**.
   d. Click **Change adapter settings**.
   The **Network Connections** window opens and shows the currently configured connections.

3. Right-click the connection.
4. Select **Properties** from the menu.
   The **Properties** dialog opens.
5. Clear the **Internet Protocol Version 6 (TCP/IPv6)** check box, if it is selected.
6. Click **OK**.
   The dialog closes and your changes are saved.
7. Select **Organize**, **Close** in the **Network Connections** window.
   The window closes.

Firewall Configuration

For DX NetOps to work properly in a firewall-protected environment, certain ports must be open. The following topics summarize the ports that must be open to allow communication among the DX NetOps components. To perform these tasks, log in as a user who is a member of the Administrators group.

Ports to Open for a Stand-Alone System

Open the following ports for a stand-alone system to allow DX NetOps communications to function properly.

From	To	Port [Function]
NFA console	Outbound	TCP 25 [SMTP email reports] UDP 53 [DNS]
Harvester	Routers (SNMP interface, read-only)	UDP 161 [SNMP polling]
Harvester	Trap destination	UDP 162 [traps]
Router	Harvester	UDP 9995 [flow]
Administrators and operators	NFA console	TCP/HTTP 80 [UI access and SNMP web services] TCP/HTTP 8381 [Single Sign-On]
CA PC / CA NPC Console	NFA console	TCP/HTTP 80 [device and interface synchronization with CA PC / CA NPC] TCP 8681 [data import for DX NetOps views in CA PC / CA NPC, CA UIM] TCP 8981 [NFA data read via OData API]
Administrators	Each server	TCP 3389 [Remote Desktop, if Remote Desktop is used] TCP 5800, 5801, 5900, 5901 [VNC, if VNC is used]

Ports to Open for a Two-Tier Distributed Deployment

NFA console and Harvesters on separate servers

From	To	Port [Function]
NFA console	Outbound	TCP 25 [SMTP email reports] UDP 53 [DNS]

NFA console	Harvester	TCP 3307 [CA MySQL] TCP 3308 [MySQL] TCP 8066 [SOAP web service calls] TCP 8067 [Flow Statistics web service calls] TCP 8080 [File web server port for collecting Harvester files] UDP 161 [Watchdog service]
Harvester	NFA console	TCP 8067 [Flow Statistics web service calls]
Harvester	Routers (SNMP interface, read-only)	UDP 161 [SNMP polling]
Harvester	Trap destination	UDP 162 [traps]
Router	Harvester	UDP 9995 [flow]
Administrators and operators	NFA console	TCP/HTTP 80 [UI access and SNMP web services] TCP/HTTP 8381 [Single Sign-On]
CA PC / CA NPC Console	NFA console	TCP/HTTP 80 [device and interface synchronization with CA PC / CA NPC] TCP 8681 [data import for DX NetOps views in CA PC / CA NPC] TCP 8981 [NFA data read via OData API]
Administrators	Each server	TCP 3389 [Remote Desktop, if Remote Desktop is used] TCP 5800, 5801, 5900, 5901 [VNC, if VNC is used]

Web Browser Support

For client systems that are used to log into the NFA console: We recommend Microsoft Internet Explorer version 11 or the latest version of Mozilla Firefox. Other browsers or browser versions may work with the NFA console, but have not been tested.

NFA console server: Install Microsoft Internet Explorer version 11 or Mozilla Firefox (latest version).

To set up DX NetOps and work with data in the CA Performance Center console, use Internet Explorer with Compatibility View turned off. You can use Internet Explorer in the NFA console with Compatibility View turned on or off.

If Internet Explorer Developer Tools are installed, you can use **F12** to access Compatibility View options for the current browser session.

Follow these steps:

1. Press **F12**.
   A new pane opens in the lower half of the window.
2. Click the **Browser Mode** item on the main menu.
3. Select the Internet Explorer option that does not contain the phrase "Compatibility View."

Configure Regional Settings

Having incorrect regional settings on the server can cause various errors in the web interface. DX NetOps requires that the regional settings be set to US Regional settings with the proper **sShortDate** and **sLongDate** settings in the registry.

Follow these steps:

1. Enter the **Windows** regional settings menu through the **Control Panel**.
2. Select **English (United States)** as the **Format**.
   Click **OK**.
3. In the **Registry Editor**, use **Edit, Find** to search for the following registry keys and update them to the following values:
   - **sLongDate**: `dddd, MMMM dd, yyyy`
   - **sShortDate**: `M/d/yyyy`
   - **sLanguage**: `ENU`
   - **Locale**: `00000409`
   - **LocaleName**: `en-US`
   - **sCountry**: `United States`
   - **sDecimal**: `.`
4. Use the **F3** key to continue searching the registry for more instances of these keys.
5. Repeat until the **Registry Editor** states it has finished searching the registry.
   The regional settings change in the **Control Panel** might not update every registry entry, which can affect the database if all entries are not updated manually.
6. After all changes are made, restart the server to apply the registry changes.

Linux Servers

For a distributed deployment, DX NetOps supports running the Harvester on dedicated Linux servers. If you install the Harvester software on a Linux system, verify that your hardware meets the recommendations and requirements that are noted here.

Setting or Component	Description
Operating System	Red Hat Enterprise Linux 6.8 or 7.3, 7.4 on a 64-bit processor **(for all NFA releases)**, and Red Hat Enterprise Linux 7.5, or 7.6 on a 64-bit processor **(for NFA 10.0.3 and 10.0.4)**
Language	English, Chinese (Simplified), French (France), or Japanese language The appropriate language packs are required for localized deployments.
Disk Space	Root partition that contains 40 GB of available space Partition for DX NetOps that contains the following amounts of available space: • 41 GB for the installation or upgrade files • 4 GB for swap • 1 TB for data
CPU	Two 2.26-GHz quad-core processors
Memory	Minimum 12 GB RAM
Hard drives	Six 300-GB, 10,000-RPM SAS hard drives in RAID5 configuration If you do not have enough available space in the `/tmp` directory (at least 4 GB) and you cannot configure it, relocate the directory. Export the `IATEMPDIR` environment variable (for the Install Anywhere temporary directory) to set a new location, and select a directory with sufficient available space.
Ports	1-Gb Ethernet port
Screen resolution	Minimum display resolution of 1024x768 (XGA)

Features, settings, and additional software	AdoptOpenJDK JRE 1.8.0.232, for Network Flow Analysis 10.0.3 and 10.0.4 releases. AdoptOpenJDK JRE 1.8.0.212, for CA Network Flow Analysis 10.0.2 release. Java Runtime Engine (JRE) 1.8.0_192, which is installed automatically during the installation or upgrade of the DX NetOps software till NFA 10.0.1 and previous releases.

Prepare the Linux Servers

Before you begin the installation, verify that the following conditions are met. Failure to comply with these requirements can result in data loss, increased downtime, software conflicts, or a failed installation.

- Before you begin **NFA 10.0.3 GUI installation on RHEL 7.6**, run the `yum install redhat-lsb*` command.
- Before you begin the tasks, log in to a Linux server with root privileges.
- System Requirements: Verify that the installation servers meet the requirements and recommendations.
- Run the **yum install -y fontconfig** command, to install the CA Network Flow Analysis using the User Interface mode.
- Before running the installer, install the libaio package if it is not part of the already installed OS.
- Verify that each of the Harvester Linux servers is ready for the installation by:
  - Assigning a static IP address to each server. Set the Harvester server IP address to match the flow-export destination for each router.
  - Disabling the following third-party software: Antivirus, server monitoring, and maintenance software. If you enable antivirus scans later, exclude the DX NetOps installation path and its subdirectories.

    NOTE

    - If SNMP is not running, the installation program displays a warning. You can bypass the warning and install SNMP later.
    - Polling fails if DNS resolution is not configured.
- For a Linux Harvester running on a VM with greater than 4 million flows/minute, make the following environment changes:
  a. Set the following kernel parameter in the `/etc/sysctl.conf` file, to match the socket buffer size of 32 MB.
     ```
     net.core.rmem_max=33554432
     ```
  b. If ethtool is not available on the server, install the ethtool using the following command:
     ```
     yum install ethtool
     ```
  c. Set the interface rx buffer to the max 4096.
     ```
     ethtool -G <interface> rx 4096
     ```
     Where *<interface>* is the interface receiving the netflow, for example eth0.

     NOTE

     Localization Notes:

     - To support non-Latin characters such as Japanese and Simplified Chinese, any command-line clients that you use for installation must be configured for UTF-8 encoding. If UTF-8 encoding is not enabled, these characters may not display properly.
     - The appropriate language packs are required for localized deployments.
     - Regional Settings must use a period (.) to indicate a decimal value. If your deployment is localized to French, change the decimal symbol to a period in the **Region and Language: Customize Format** dialog.

Disable IPv6 Networking on Linux Servers

Disable IPv6 networking on each Linux server that has a Harvester installed.

NOTE
Complete this task before you add the Harvester in the NFA console. If IPv6 is enabled when you add a Harvester in the NFA console, the Harvester automatically binds with an IPv6-format address. This prevents DX NetOps from receiving the Harvester data.

Verify IPv6 Status

Verify if IPv6 is enabled or not. In case IPv6 is enabled, disable the IPv6, else continue to Disable the iptables Firewall.

Follow these steps:

1. Log in with root privileges.
2. Execute the following command:
```
$ ip a | grep inet6
```
3. In case you find the following as the result, then IPv6 is enabled:
```
inet6 ::1/128 scope host
inet6 fe80::e922:bcdf:e150:labb/64 scope link
```
4. If IPv6 is disabled, you should see no output if you run this command.

Disable IPv6

Perform the following steps to disable IPv6.

Follow these steps:

1. Log in with root privileges.
2. Open the /etc/sysctl.conf file with the following command:
```
$ sudo vim /etc/sysctl.conf
```
   a. Add the following lines to it:
```
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
```
   b. Now save the file and reboot your computer with the following command:
```
$ sudo reboot
```
   c. Verify that IPv6 is disabled by running the following command:
```
$ ip a | grep inet6
```

Disable the iptables Firewall

We recommend that you disable the iptables firewall and stop the iptables service on each Linux server that has a Harvester installed. Disabling iptables ensures that all the required ports are open and that the iptables firewall does not impact performance adversely.

NOTE
If your enterprise requires the use of iptables, make sure that you open all of the applicable firewall ports in the firewall configuration list. Also, make sure that you have full localhost-to-localhost access. This step is required because DX NetOps uses RMI (Remote Method Invocation) access.

Complete the following steps to disable all levels of iptables and allow communication among DX NetOps components.

Disable iptables in RHEL 6

Follow these steps:

1. Log in as root or with a sudo user account.
2. Run the following commands in a command prompt window:

```
service iptables stop

chkconfig iptables off

chkconfig  --list |grep iptables
```

3. Review the output of the last command to make sure that all of the iptables levels are off, as shown in the following example:

```
iptables 0:off 1:off 2:off 3:off 4:off 5:off 6:off
```

Disable iptables in RHEL 7

Follow these steps:

1. Log in as root or with a sudo user account.
2. Check the status of iptables, using the following command:

```
systemctl status firewalld
```

3. If the iptables are in enable status, stop the iptables using the following command. If they are disabled, ignore the further steps.

```
systemctl stop firewalld
```

4. Disable the iptables using the following command:

```
systemctl disable firewalld
```

> **NOTE**
>
> **More information:**
>
> Firewall configuration list

DNS Name Resolution

DX NetOps requires DNS name resolution. If host names do not resolve to their corresponding IP addresses, SNMP polling fails.

To determine whether an IP address resolves to a host name on a Linux server, enter the `hostname -i` command in a command prompt window, as shown in the following example:

```
[root@NFAHARV]# hostname -i
```

If the command fails to return the corresponding IP address, you can edit the `/etc/hosts` configuration file on the Harvester server manually. Add a line for the local server, which associates the host name and IP address.

Follow these steps:

1. Log in to the Harvester server as root or with a sudo user account.
2. Open the `/etc/hosts` file in a text editor.
3. Add a line that associates the IP address with the local host name, as shown in the following example:

```
[root@NFAHARV]# more /etc/hosts;

# Do not remove the following line, or various programs

# that require network functionality will fail.

127.0.0.1 localhost localhost.localdomain localhost

::1 localhost6.localdomain6 localhost6

10.0.0.10 NFAHARV

where:

 10.0.0.10 = IP address of the Harvester server

 NFAHARV = Host name of the Harvester server
```

4. Save and close the `/etc/hosts` file.
5. Restart the DX NetOps services on the Harvester server, including the following services:
   – NFA CollpollWS (nfa_collpollws): CA NFA Collection and Poller Webservices
   – NFA Proxies (nfa_proxies): CA NFA DNS/SNMP Proxies
   – NFA Poller (nfa_poller): CA NFA Poller

   Your change takes effect immediately.

Install SNMP on Linux Servers

Install the Net-SNMP. Net-SNMP is required to support Watchdog functionality.

Install net-snmp

Follow these steps:

1. Install net-snmp RPM using the following command.

```
yum install -y net-snmp net-snmp-utils
```

2. Follow the prompt to complete the `net-snmp` installation.

Configure SNMP on RHEL 6

Follow these steps:

1. Open the /etc/snmp/snmpd.conf file using a text editor.
2. Find the following line:

```
com2sec notConfigUser default public
```

3. Ensure the following lines exist, else add them:

```
view all included .1

access notConfigGroup "" any noauth exact all none none

smuxpeer .1.3.6.1.4.1.674.10892.1

trapsink <IP_address> <desired_community_string>
```

4. Open the /etc/sysconfig/iptables.config file using a text editor.
5. Add the following lines to open the SNMP ports

```
-A INPUT -p udp -m udp --dport 161 -j ACCEPT

-A INPUT -p udp -m udp --dport 162 -j ACCEPT
```

6. Restart the snmpd service, using the `service snmpd restart` command.
7. Restart iptables service, using the `service iptables restart` command.

Configure SNMP on RHEL 7

Follow these steps:

1. Set up a minimal configuration:

```
cd /etc/snmp

cp -p snmpd.conf snmpd.conf.dist

echo "rocommunity public">snmpd.conf

echo "syslocation here" >>snmpd.conf

echo "syscontact root@localhost" >>snmpd.conf
```

2. Activate at boot and start the SNMP service:

```
systemctl enable snmpd && systemctl start snmpd
```

3. Execute a simple test:

```
snmpwalk -v 1 -c public -O e 127.0.0.1
SNMPv2-MIB::sysDescr.0 = STRING: Linux rhel7.example.com 3.10.0-54.0.1.el7.x86_64 #1
 SMP Tue Nov 26 16:51:22 EST 2013 x86_64
SNMPv2-MIB::sysObjectID.0 = OID: NET-SNMP-MIB::netSnmpAgentOIDs.10
```

NOTE

More information:

- The `snmpd.conf` file contains information on adding viewing rights for HostResources-MIB and UCD-SNMP-MIB, or check online for information on setting up read permissions.

Install the Software

Before you begin the DX NetOps installation:

- Verify that the servers meet the requirements and recommendations.
- Stop other programs from running during the installation.
- Ensure that no one else is logged in to the server during the installation.

Install the components:

After installation is complete, perform Post-Installation or Upgrade Tasks.

Install the Components on a Stand-Alone Server

Complete the steps in this topic to install all of the components on a single Windows server or virtual machine.

Install the Harvester

This section describes the procedure to install the harvester.

Follow these steps:

1. Verify that the installation server is prepared.
2. Log in as a user who has administrator privileges for DX NetOps.
3. Start the Harvester phase of the installation: Double-click the `NFHarvesterSetupx.x.x.exe` file.
4. Verify that the appropriate language is selected, then click **OK**.
   The **Welcome** screen opens.
5. Click **Next**.
   The **CA NFA Harvester License Agreement** screen opens.
6. Review and Click **Next** to accept the license agreement:
   Prerequisite tests run to look for problems on the server. If a problem is found, an error message opens. A critical problem causes the program to exit. A **Prerequisite Check Warning** message or other warning message opens for non-critical problems, which gives you the option to make corrections now or after the installation or upgrade is complete.
7. Review the test results in the **Prerequisite Check Warning** message, if it opens:
   a. Correct problems now or wait until the program finishes.
   b. Click **OK** to close the message.

8. Verify or specify the installation directory and Click **Next**:
   a. (Optional) Click **Choose** in the **Choose Install Folder** screen to change the installation location.
   Default Location: `C:\CA\NFA`.

      NOTE

      We recommend that you install DX NetOps components on a dedicated drive and not the operating system drive. The CA NFA console is installed to the same directory that you choose for the Harvester.

   The **Pre-Installation Summary** screen opens.
9. Review the pre-installation information and click **Install**.
   The **Installing Harvester** screen opens. When the installation is complete, the **Install Complete** screen opens and reports any errors that occurred.
10. (Optional) If errors occurred during the installation, see the following log for details:
   install_path`\Harvester_Install_`*<timestamp>*`.log` (where *<timestamp>* is the time the log was created)
11. Click **Done** in the **Install Complete** screen.
   The Harvester installation program closes.

Install the CA NFA console

This section describes the procedure to install the CA NFA Console.

Follow these steps:

1. Start the NFA console installation software: Double-click the `RAConsoleSetupx.x.x.exe` file in Windows Explorer.
   The language selection screen opens.
2. Verify that the appropriate language is selected, then click **OK**.
   The **Welcome** screen opens.
3. Click **Next** in the Welcome screen.
   The **NFA Console License Agreement** screen opens.
4. Review and Click **Next** to accept the license agreements:
   The **Third-Party License Agreement** screen opens.
5. Read the third-party license agreement and Click **Next**.
   Prerequisite tests run. If an error message opens that requires attention, see Troubleshooting.
6. Review the test results in the **Pre-requisite Check Warning** message, if it opens:
   a. Fix any noncritical problems now or wait until the program finishes.
   b. Click **OK** to close the message.
      The **Singlebox Confirmation** message opens and asks you to confirm that you want a stand-alone deployment.
7. Review the information and click **OK**.
   The **Pre-Installation Summary** screen opens.
8. Review the pre-installation information and click **Install**.
   The **Installing NFA** screen opens. When the NFA console installation is complete, the **Install Complete** screen opens.
   a. Select **Yes, restart my system**.
   b. Click **Done**.
      Installation is complete.

Next: Complete the following:

- Initial configuration
- Post-installation tasks

Install a Distributed Deployment

The topics in this section describe how to install the software on each component server. The recommended installation order is Harvesters first, then the NFA console.

To install a two-tier distributed deployment:

- Install the Harvester on a Windows Serveror
  Install the Harvester on a Linux Server
- Install the NFA Console

Install the Harvester on a Windows Server

Distributed deployments have separate servers for the NFA console and the Harvester. Complete the steps in this topic to install the Harvester on a dedicated Windows server or virtual machine.

In a distributed deployment, each Harvester is on a separate server. To install a Harvester on a dedicated Windows server or virtual machine, complete the following. These steps apply to a two-tier distributed deployment.

Follow these steps:

1. Verify that the server is prepared.
2. Log in as a user who has administrator privileges for DX NetOps.
3. Start the installation: Double-click the `NFHarvesterSetupx.x.x.exe` file in Windows Explorer on the Harvester server.
   The language selection screen opens.
4. Verify that the appropriate language is selected, then click **OK**.
   The **Welcome** screen opens.
5. Click **Next** in the **Welcome** screen.
   The **License Agreement** screen opens.
6. Review and accept the license agreement:
   a. Read the license agreement and scroll down.
   b. If you want to continue under the terms of the license agreement, click the option to accept it.
   c. Click **Next**.
      Prerequisite tests are run to identify problems on the server. If a problem is found, an error message opens.
7. If the **Prerequisite Check Warning** message opens, review the test results:
   a. Correct problems now or wait until the program finishes.
   b. Click **OK** to close the message.
      Once the server passes the required checks and you close any noncritical messages, the **Choose Install Folder** screen opens and displays the default root installation path.
8. Verify or specify the installation directory:
   a. (Optional) Click **Choose** in the **Choose Install Folder** screen to change the installation location.
      The default location is `C:\CA\NFA` . We recommend that you install DX NetOps components on a drive that is dedicated to DX NetOps, not the operating system drive. The NFA console and the Harvester are installed to the same directory.
   b. Click **Next** when the installation path setting is correct.
      The **Pre-Installation Summary** screen opens.
9. Review the pre-installation information, then click **Install**.
   The **Installing Harvester** screen opens. When the installation is complete, the **Install Complete** screen opens and reports any errors that occurred.
10. (Optional) If errors occurred during the installation, see the following log for details:

install_path\Harvester_Install_<*timestamp*>.log (where <*timestamp*> is the time the log was created)

11. Click **Done** in the **Install Complete** screen.
    The Harvester installation program closes.

Next:

- To install an additional Harvester on another Windows server, repeat these steps.
- Install the NFA Console.

Install the Harvester on a Linux Server

A two-tier distributed deployment of DX NetOps may include one or more Linux Harvester servers. To install the Harvester software on a dedicated Linux server or virtual machine, complete the steps in this topic.

Follow these steps:

1. Verify that the server is prepared.
2. Log in to the Harvester server as `root` .
   You can install the software locally or remotely; for example, by using `ssh` when you are logged in with root privileges.

   NOTE
   If you do not have root access, use an account with sudo privileges.
3. Open a command prompt window.
4. Run the following command to change the`ulimit` for the open files limit:

   ```
   ulimit -n ulimit_number
   ```

   Example:

   ```
   ulimit -n 65536
   ```

5. Prepare the installation/upgrade file for execution:
   a. Log in to the Harvester server as `root` .
      You can install or upgrade the software locally or remotely; for example, by using `ssh` when you are logged in with root privileges. If you do not have root access, use an account with sudo privileges.
   b. Execute the `chmod` command on the file in a terminal window:

      ```
      chmod u+x NFHarvesterSetupx.x.x.bin
      ```

   c. (Optional) Execute the list command to verify that the file is executable:

      ```
      ls -al
      ```

      The file permission settings are displayed.
6. Run the installation or upgrade software:

   ```
   ./NFHarvesterSetupx.x.x.bin
   ```

7. If the Linux server or virtual machine has GUI installed then follow the steps mentioned in Install Harvester through GUI.

8. If the Linux server or virtual machine is not GUI enabled, follow the steps mentioned in Install Harvester through Console.

Install Harvester through Console

1. Click Enter and accept the license agreement.
2. Select the installation folder or click enter to continue with the default installation location.
3. Review the Pre-Installation Summary.
4. Click Enter.

The Harvester installation is completed through Console.

Install Harvester through GUI

1. Verify that the appropriate language is selected, then click **OK**.
   The **Welcome** screen opens.
2. Click **Next** in the **Welcome** screen.
   The **License Agreement** screen opens.
3. Review and accept the license agreement:
   a. Read the license agreement and scroll down.
   b. If you want to continue under the terms of the license agreement, click the option to accept it.
   c. Click **Next**.
      Prerequisite tests are run to identify problems on the server. If a problem is found, an error message opens.
4. If the **Prerequisite Check Warning** message opens, review the test results:
   a. Correct problems now or wait until the program finishes.
   b. Click **OK** to close the message.
      Once the server passes the required checks and you close any noncritical messages, the **Choose Install Folder** screen opens and displays the default root installation path.
5. Verify or specify the installation directory:
   a. (Optional) Click **Choose** in the **Choose Install Folder** screen to change the installation location.
      We recommend that you install DX NetOps components on a partition that is dedicated to DX NetOps. The NFA console is installed to the same directory that you choose for the Harvester.
   b. Click **Next** when the installation path setting is correct.
      The **Pre-Installation Summary** screen opens.
6. Review the pre-installation information, then click **Install**.
   The **Installing Harvester** screen opens, which shows the progress. When the installation is complete, the **Install Complete** screen opens and reports any errors that occurred.
7. (Optional) If errors occurred during the installation, see the following log for details:
   install_path`/Harvester_Install_<`*timestamp*`>.log` (where <*timestamp*> is the time the log was created)
8. Click **Done** in the **Install Complete** screen.
   The Harvester installation program closes.

Next:

- To install an additional Harvester on another Linux server, repeat these steps.
- Install the NFA console.

Install the NFA Console

Distributed deployments use separate servers for the NFA console and Harvesters. Complete the steps in this topic to install the NFA console on a dedicated Windows server or virtual machine.

Follow these steps:

1. Verify that the installation server meets the following requirements:
   – The server is prepared.
   – The software is installed on the Harvester servers.
2. Log in to the NFA console server as a user who has administrator privileges for the system and for DX NetOps.
3. Start the installation: Double-click the `RAConsoleSetupx.x.x.exe` file in Windows Explorer on the NFA console server.
   The language selection screen opens.
4. Verify that the appropriate language is selected, then click **OK**.
   The **Welcome** screen opens.
5. Click **Next**.
   The **License Agreement** screen opens.
6. Review and accept the license agreements:
   a. Read the NFA console license agreement and scroll down.
   b. If you want to continue, click the option to accept the license agreement.
   c. Click **Next**. The **Third-Party License Agreement** screen opens.
   d. Read the third-party license agreement and scroll down.
   e. If you want to continue, click the option to accept the third-party license agreement.
7. Click **Next**. Prerequisite tests run.
   If the server fails any noncritical tests, the **Prerequisite Check Warning** message opens.
8. If the **Prerequisite Check Warning** message opens, review the test results:
   a. Correct problems now or wait until the program finishes.
   b. Click **OK** to close the message.
9. Click **Next**.
   The **Choose Install Folder** screen opens.
10. (Optional) Click **Choose** to change the program installation location when prompted or enter a new path manually.
11. Click **Next**. The **Pre-Installation Summary** screen opens.
12. Review the pre-installation information, then click **Next**.
    The **Installing NFA** screen opens and installation starts. Once installation starts, you cannot cancel it.
    When the program finishes, the **Install Complete** screen opens and reports any errors.
13. (Optional) If errors occurred, see the installation log:
    install_path`\NFA_Install_<`*timestamp*`>.log`
14. Exit from the installation program:
    a. Select one of the restart options:
       - **Yes, restart my system**: Restart the system when you click **Done**.
       - **No, I will restart my system myself**: Defer the restart to be performed manually.
    b. Click **Done**.
       The installation program closes. If you selected the option to restart now, the system restarts and the installation is finalized.

Next: Complete the following:

- Initial configuration
- Post-installation tasks

Install Performance Center

Your deployment can include either CA Performance Center or CA NetQoS Performance Center, but it is not required. After you install DX NetOps, you would register it as a data source in the Performance Center Console. You use the Performance Center Console to perform certain administrative tasks. You also use it to view DX NetOps data next to data

from other data sources. Until you register the product as a data source, some of the function links on the **Administration** page are disabled.

Verify that you have one of the following programs installed:

- Compatibility Matrix , installed on a Linux server that does not have a Harvester installed, or
- CA NetQoS Performance Center 6.2, installed on a server that is running Windows Server 2008 R2 Standard edition

You cannot co-locate CA Performance Center or CA NetQoS Performance Center with the current release of any DX NetOps component or CA Anomaly Detector.
If you uninstall DX NetOps from a server that has any related software installed, the related software is disabled.

See the documentation for your Performance Center type for information on installation.

NOTE

More information:

- CA Performance Management

Install Flow Cloner

Once you have the Flow Cloner installed and configured, the flows going to the Harvester are forwarded whenever the CA NFA Flow Cloner service is running. The service starts by default whenever the server is rebooted. You can change this setting to run the service on demand. The configuration file must identify at least one destination IP address or the service will not start.

Install the Flow Cloner on the Harvester server in a distributed deployment or on the single server in a stand-alone deployment.

NOTE
The Flow Cloner has not affected Harvester performance significantly during testing. If you use the Flow Cloner on a high-flow Harvester server, we recommend monitoring performance.

Prerequisites

Before you install the Flow Cloner, verify that your installation server meets the following prerequisites:

- The server is configured and installed or upgraded.
- The server already has software installed and configured to act as one of the following components:
  - Windows Harvester server in a DX NetOps distributed deployment
  - Single server in a stand-alone deployment.
- The server has at least 12.8 MB of disk space available on the target drive for the Flow Cloner.
- You exited from all other programs.
- No other user is logged in to the server.

Installing Flow Cloner

Follow these steps:

1. Log in to the Windows-based Harvester installation server as a user with administrator privileges.
   The installation server must have the DX NetOps Harvester software installed.
2. Locate the installation program file: *install_path*\setup\FlowClonerSetupx.x.x.exe .
3. Start the installation program: For example, double-click the FlowClonerSetupx.x.x.exe file in Windows Explorer.
4. Click **Next** in the **Welcome** screen that opens.
   The **Pre-Installation Summary** screen opens and shows the installation path and disk space requirements. The Flow Cloner will be installed in the same root installation directory that is used for the Harvester or stand-alone software.

5. Click **Install**.
   The **Install Complete** screen opens when the installation is complete.
6. Click **Done**.
   The installation program closes. An installation log file named `FlowCloner_Install_<timestamp>.log` is created in the root installation directory.

Configuring Flow Cloner Options

To configure the Flow Cloner, modify its default initialization (`.ini`) file. The `.ini` file contains a header line followed by a line for each destination host (each host that will receive the cloned packets). You must specify at least one destination host. If you do not specify values for the header fields, the default values are used.

Follow these steps:

1. Log in to the Flow Cloner installation server as a user who has administrator privileges.
2. Open the following file in a text editor: *install_path*`\Netflow\FlowCloner\flowclonedef.ini`
   The `.ini` file has a header line followed by a line for each host that will receive packets.
3. Customize the header line:
   The header content must be contained in the first non-commented and non-blank single line in the file.
   – To use the default value for the input NIC, replace the entire header line with the following token:

     ```
     /use defaults
     ```

     You can follow the `/use defaults` token with a comment, as shown in the following example:

     ```
     /
     use defaults ; use first available NIC and port 9995 to listen and send flows on the first
     ```

     The program uses the first available NIC. The hosts listen for the original flows and cloned flows on port UDP 9995. The `/use defaults` token takes effect only if the header does not contain any other tokens.
   – (Optional) To specify the listening port, enter the `/port= token` , followed by the port number. The Harvester that receives the original flows listens on UDP 9995 unless you use the `/port` token to specify a different port.
     Default: UDP 9995
   – (Optional) To specify the destination port, enter the `/dest port= token` , followed by the port number. The hosts that receive the cloned flows listen on UDP 9995 unless you use the /dest port token to specify a different port. All of the hosts listen for the cloned flows on the same port.
     Default: UDP 9995
   – (Optional) To specify the Input NIC, enter the `/listen ip= token` , followed by the IP address for the NIC on which the Flow Cloner listens for packets.
     Default: First functional IP address of the host
4. Specify one or more hosts that will receive the cloned packets:
   Enter each host on a separate line, which consists of the `dest ip= token` and IP address of the destination host. You can put the destination host lines in any order.
   Example:

   ```
   /dest ip=10.0.0.100 ; send cloned packets to 10.0.0.100
   ```

   If the IP address is missing, the line is ignored.
5. Save and close the `FlowCloneDef.ini` file.
6. Start the CA NFA Flow Cloner service on the Harvester server.

The Flow Cloner is enabled and attempts to forward packets to each valid destination that you specified. Flow cloning continues until you stop the CA NFA Flow Cloner service manually.
The CA NFA Flow Cloner service is configured to start automatically on reboot and start sending cloned flow data. To operate the Flow Cloner only on demand, change this configuration in the **Services** window. The service can run only if the configuration file identifies at least one destination IP address.

NOTE

More information:

Flow Cloner Configuration Files

Initial Configuration

After the DX NetOps software is installed, some configuration is needed. The articles that follow provide information to help you with the initial configuration.

Generate or Configure Certificates for Use by CA Network Flow Analysis

DX NetOps (CA NFA) can be configured to use X.509 certificates to enable various levels of secure transport across the product. The NFA Console software includes the `openssl` command-line utility that can be used to:

- Generate certificate signing requests
- Sign certificates (either self-signing or acting as a Certificate Authority (*CA*))
- Manage certificate storage

Regardless of what types of security you are configuring, you can use the `openssl` tool to fulfill your certificate management needs.

We recommend obtaining certificates for use with DX NetOps from your trusted Certificate Authority. If you choose not to use certificates signed by a trusted Certificate Authority, follow these instructions to generate your own certificates for use with CA NFA.

The general procedure involves:

1. Creating a certificate on the NFA Console server that acts as a Certificate Authority certificate.
2. Using that certificate to sign certificates that are generated for the NFA Console and for each Harvester in your deployment.

Follow these steps:

On the Console:

WARNING

- The `openssl req -new` command asks you to enter information to incorporate into the certificate request. The **Common Name** must be unique in each request.
- The number you use as the argument for `-set_serial` in the `openssl x509` commands (generating a certificate signed by the console acting as a *CA*) must be unique in each request.

1. Create a directory named *install_path*`\certs` for storing certificates and certificate stores.
   Execute the commands in the following steps from this directory.
2. Set the `OPENSSL_CONF` environment variable for the `openssl` configuration file.
   ```
   set OPENSSL_CONF=install_path\Tools\openssl\bin\openssl.cfg
   ```

3. Use `openssl` to generate a self-signed certificate. This certificate acts as the certificate authority (*CA*) certificate for the console and for all harvesters. The `openssl` command can be found on the NFA Console, in the *`install_path`*`\tools\openssl\bin` directory.
   a. Generate the private key:
      ```
      openssl genrsa -des3 -out nfa-ca-key.pem 2048
      ```
   b. Generate a Certificate Signing Request:
      ```
      openssl req -new -key nfa-ca-key.pem -out nfa-ca.csr
      ```
   c. Remove passphrase from the private key:
      ```
      copy nfa-ca-key.pem nfa-ca-key.pem.orig
      openssl rsa -in nfa-ca-key.pem.orig -out nfa-ca-key.pem
      ```
   d. Generate the self-signed certificate:
      ```
      openssl x509 -req -days 1825 -in nfa-ca.csr -signkey nfa-ca-key.pem -out nfa-ca-cert.pem
      ```
4. Use `openssl` to generate a new certificate for the console, signed by the *CA* certificate you just generated.
   a. Generate the private key:
      ```
      openssl genrsa -des3 -out nfa-console-key.pem 2048
      ```
   b. Generate a Certificate Signing Request:
      ```
      openssl req -new -key nfa-console-key.pem -out nfa-console.csr
      ```
   c. Remove passphrase from the private key:
      ```
      copy nfa-console-key.pem nfa-console-key.pem.orig
      openssl rsa -in nfa-console-key.pem.orig -out nfa-console-key.pem
      ```
   d. Generate a certificate signed by the console acting as a *CA*:
      ```
      openssl x509 -req -days 1825 -in nfa-console.csr -CA nfa-ca-cert.pem -CAkey nfa-ca-key.pem -set_serial
       <unique_num> -out nfa-console-cert.pem
      ```
5. Use `openssl` to generate a PKCS12 keystore that contains both the *CA* certificate and the console certificate that is signed by the *CA* certificate.
   a. Generate keystore with console private key and console certificate:
      ```
      openssl pkcs12 -export -in nfa-console-cert.pem -inkey nfa-console-key.pem -out nfa-console-
      keystore.pfx
      ```
   b. Generate truststore with console certificate and *CA* certificate:
      ```
      openssl pkcs12 -export -in nfa-console-cert.pem -inkey nfa-console-key.pem -certfile nfa-ca-cert.pem -
      out nfa-console-truststore.pfx
      ```
6. Use `openssl` to generate a new certificate signed by the *CA* for each Harvester.
   a. Generate private key:
      ```
      openssl genrsa -des3 -out nfa-harvester-key.pem 2048
      ```
   b. Generate a Certificate Signing Request:
      ```
      openssl req -new -key nfa-harvester-key.pem -out nfa-harvester.csr
      ```
   c. Remove passphrase from private key:
      ```
      copy nfa-harvester-key.pem nfa-harvester-key.pem.orig
      openssl rsa -in nfa-harvester-key.pem.orig -out nfa-harvester-key.pem
      ```
   d. Generate a certificate signed by the console acting as a CA:
      ```
      openssl x509 -req -days 1825 -in nfa-harvester.csr -CA nfa-ca-cert.pem -CAkey nfa-ca-key.pem -
      set_serial <unique_num> -out nfa-harvester-cert.pem
      ```
7. Use `openssl` to generate a pkcs12 keystore and truststore that contains both the *CA* certificate and the harvester certificate that is signed by the *CA* certificate for each Harvester.
   a. Generate the keystore with harvester private key and harvester certificate:
      ```
      openssl pkcs12 -export -in nfa-harvester-cert.pem -inkey nfa-harvester-key.pem -out nfa-harvester-
      keystore.pfx
      ```
   b. Generate the truststore with harvester certificate and *CA* certificate:

```
openssl pkcs12 -export -in nfa-harvester-cert.pem -inkey nfa-harvester-key.pem -certfile nfa-ca-cert.pem -out nfa-harvester-truststore.pfx
```

8. Verify that the certificates created are without errors.

```
openssl verify -CAfile nfa-ca-cert.pem nfa-console-cert.pem nfa-harvester-cert.pem
```

On each Harvester:

1. Create a directory named *install_path*\certs for storing certificates and certificate stores.
2. Copy the *CA* certificate file and the Harvester certificate files and stores (keystore and truststore) to the certs directory created in step 1.

NOTE
If you are using certificates signed by an actual Certificate Authority, you can use these steps to generate the required *.pfx files in the *install_path*\certs directory. Place copies of your certificate and private key files in pem format in the *install_path*\certs directory. The *.pfx and *.pem files are required for securing DX NetOps.

Enable HTTPS for CA Network Flow Analysis

DX NetOps (CA NFA) can be configured for secure communication through the web interface.

NOTE

Some of the following settings might be available during an upgrade.

Prerequisites

Create the required certificate files using the procedure in

Generate or Configure Certificates for Use by CA Network Flow Analysis

Enable HTTPS on the Console

Follow these steps:

1. Install the *CA* certificate (generated as a prerequisite) as a trusted Certificate Authority (*CA*) for your server.
   a. Double-click the nfa-console-truststore.pfx .

b. Run the import wizard to import the certificate as trusted by the local machine.

NOTE
More information:

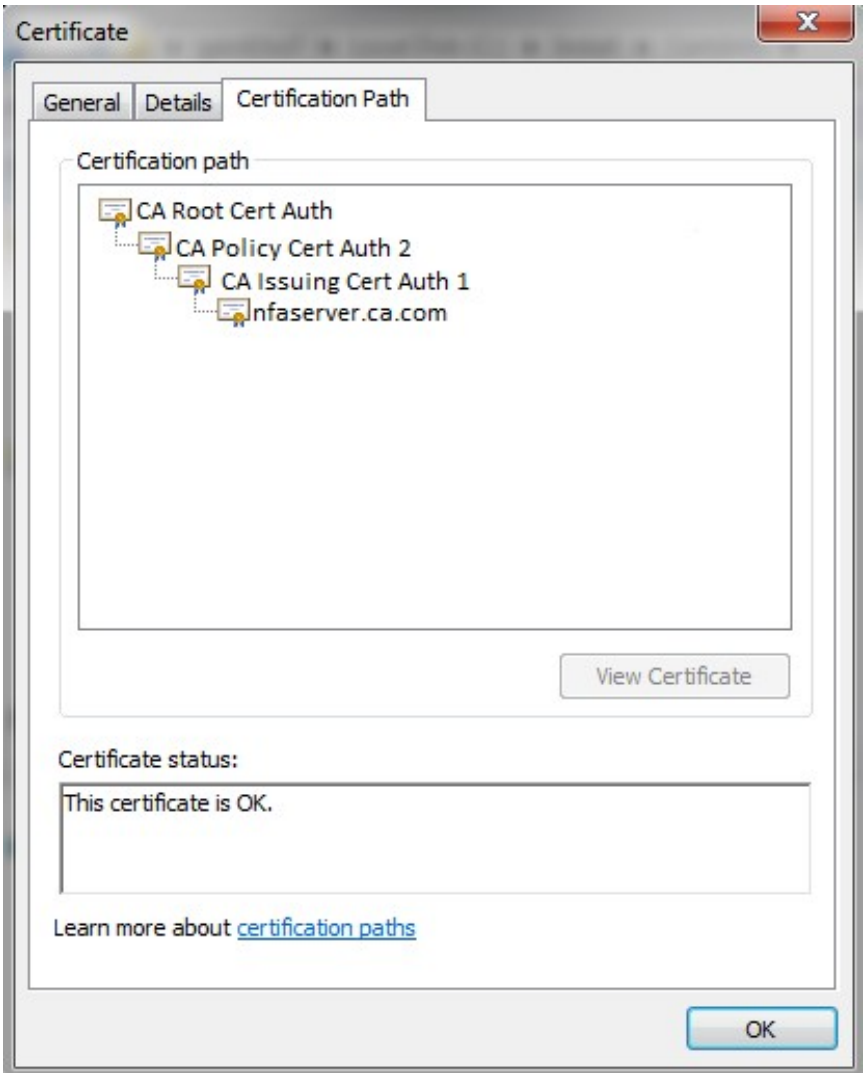
See "To import a certificate" in How to Use the Certificates Console

2. Install the Console SSL certificate.
   a. Install a signed certificate in IIS Manager Server Certificates.
      a. In the **Internet Information Service (IIS) Manager**, go to the **Features** view.
      b. Open **Server Certificates**.
      c. Under **Actions**, click **Import** to import the `nfa-console-keystore.pfx` file.

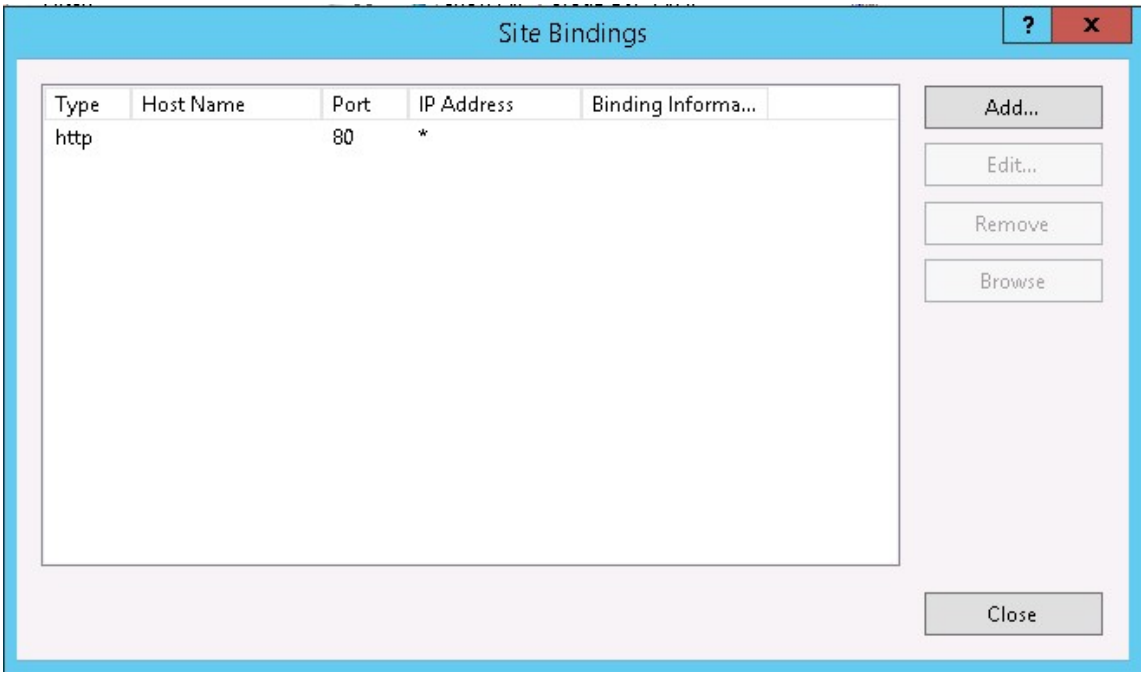
NOTE
More information:

Install an Internet Server Certificate

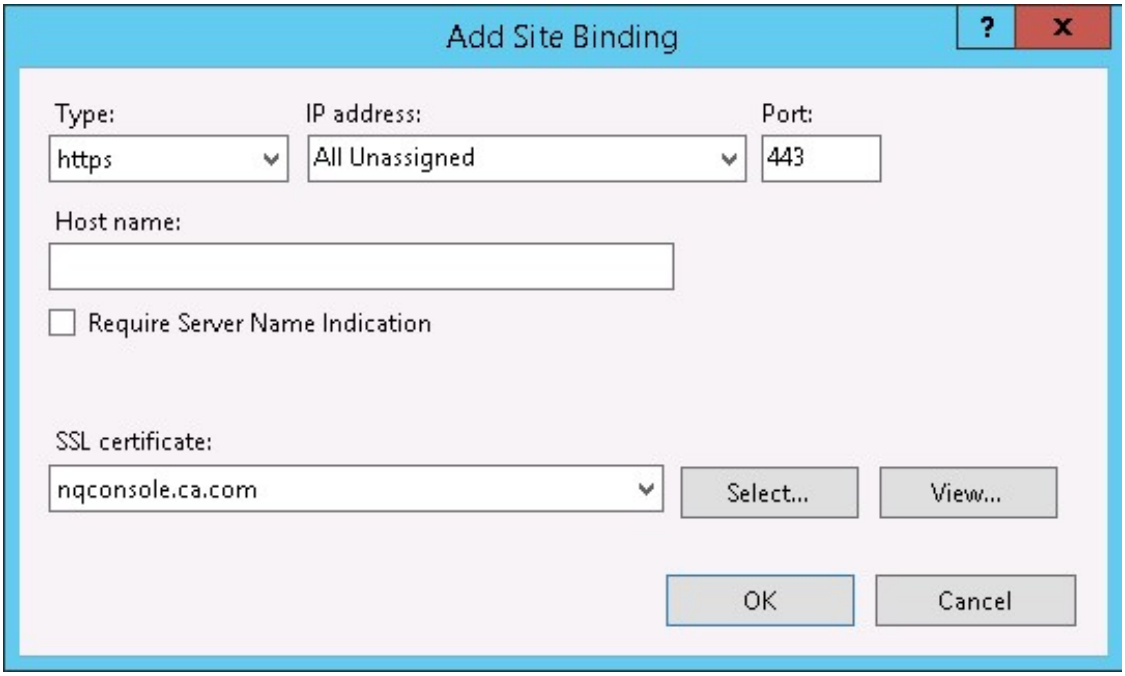
   b. To confirm that the certificate is properly installed, open the certificate and select the **Certification Path** tab. Select each certificate displayed in the Certification path and check that the **Certificate status** field shows “This certificate is OK”. Contact the certificate provider if **Certificate status** field displays errors.



3. Configure the HTTPS port IIS Application. By default, IIS does not have a binding for HTTPS.
   a. In the **Internet Information Services (IIS) Manager**, navigate to the **Default Website**.
   b. Under **Actions**, click **Bindings**.
   c. In the **Site Bindings** dialog, click **Add**.



d. Select the signed certificate from the **SSL certificate** list.



WARNING
Do not disable http-port 80 binding. DX NetOps will not work properly if http is disabled.

4. Edit the product configuration XML file.
   install_path\Portal\SSO\webapps\sso\configuration\ReporterAnalyzer.xml
   a. In the `SignInPageProductDefaultUrl` section, change `Scheme` from http to https.
   b. Enter 443 for the `Port` (blank by default).
   Example:

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<Configuration>
      <SingleSignOnEnabled>True</SingleSignOnEnabled>
      <SingleSignOnProductCode>ra</SingleSignOnProductCode>
      <SignInPageProductTitle><![CDATA[NetQoS<sup><font class="Superscript">®</font></sup>
       ReporterAnalyzer<sup><font class="Superscript">™</font></sup>]] ></SignInPageProductTitle>
      <SignInPageProductDescription>Network Traffic Analysis</SignInPageProductDescription>
      <SignInPageProductDefaultUrl>
        <Scheme>https</Scheme>
        <Port>443</Port>
        <PathAndQuery>/ra/default.aspx</PathAndQuery>
      </SignInPageProductDefaultUrl>
      <SingleSignOnWebServiceUrl>
        <Scheme>http</Scheme>
        <Port></Port>
        <PathAndQuery>/ReporterDataSource/SingleSignOnWS.asmx</PathAndQuery>
      </SingleSignOnWebServiceUrl>
</Configuration>
```

5. Configure Single Sign-On SSL scheme and port.
   Run *install_path*\Portal\sso\bin\SsoConfig.exe

```
SSO Configuration:

1. CA Performance Center

2. CA Network Flow Analysis

Choose an option >2
```

```
SSO Configuration/CA Network Flow Analysis:

1. LDAP Authentication

2. SAML2 Authentication

3. Performance Center

4. Single Sign-On

5. Test LDAP

6. Export SAML2 Service Provider Metadata

Choose an option >4
```

```
SSO Configuration/CA Network Flow Analysis/Single Sign-On:

Anonymous User Enabled: Disabled

Anonymous User ID: 2

Localhost User Sign-In Page Enabled: Disabled

Localhost User Enabled: Enabled

Localhost User ID: 1

Cookie Timeout Minutes: 20

Encryption Decryption Key: #$utP9%z

Encryption Algorithm: DES

Failed Sleep Seconds: 3

Remember Me Enabled: Enabled

Remember Me Timeout Days: 15

Scheme: http

Port: 8381

Virtual Directory: sso

1. Remote Value

2. Local Override

Choose an option > 2



SSO Configuration/CA Network Flow Analysis/Single Sign-On/Local Override:

1. Anonymous User Enabled:

2. Anonymous User ID:
```

```
3. Localhost User Sign-In Page Enabled:

4. Localhost User Enabled:

5. Localhost User ID:

6. Cookie Timeout Minutes:

7. Encryption Decryption Key:

8. Encryption Algorithm:

9. Failed Sleep Seconds:

10. Remember Me Enabled:

11. Remember Me Timeout Days:

12. Scheme:

13. Port:

14. Virtual Directory:

Select a Property > 12


Enter u to update to new value > u


Enter new value > https


SSO Configuration/CA Network Flow Analysis/Single Sign-On/Local Override:

1. Anonymous User Enabled:

2. Anonymous User ID:

3. Localhost User Sign-In Page Enabled:

4. Localhost User Enabled:

5. Localhost User ID:

6. Cookie Timeout Minutes:

7. Encryption Decryption Key:
```

```
8. Encryption Algorithm:

9. Failed Sleep Seconds:

10. Remember Me Enabled:

11. Remember Me Timeout Days:

12. Scheme: https

13. Port:

14. Virtual Directory:

Select a Property > 13


Enter u to update to new value > u


Enter new value > 8382



Enter q to quit SsoConfig
```

6. Backup and edit the SSO `start.ini` file.
   Edit *install_path*`\Portal\SSO\start.ini`.
   a. Uncomment the `--module=ssl` line so that it is active:
      `--module=ssl`
   b. Modify the `--module=http` line to be https:
      `--module=https`
7. Configure the SSO `jetty-https.xml`, `jetty-ssl.xml`, and `jetty-ssl-context.xml` files.
   a. Copy the `jetty-https.xml` template from
      install_path`\Portal\Jetty\etc\jetty-https.xml`
      to
      install_path`\Portal\SSO\etc\jetty-https.xml`
   b. Copy the `jetty-ssl.xml` template from
      install_path`\Portal\Jetty\etc\jetty-ssl.xml` to

      install_path`\Portal\SSO\etc\jetty-ssl.xml`
   c. Copy the `jetty-ssl-context.xml` template from
      install_path`\Portal\Jetty\etc\jetty-ssl-context.xml` to

install_path\Portal\SSO\etc\jetty-ssl-context.xml

d. Edit *install_path*\Portal\SSO\etc\jetty-ssl.xml .
In the addConnector section, set the port to 8382.

```
<Set name="port">8382</Set>
```

e. Edit *install_path*\Portal\SSO\etc\jetty-ssl-context.xml .
Edit the sslContextFactory section to contain the following lines.

```
<Set name="KeyStorePath">install_path/certs/nfa-console-keystore.pfx</Set>

<Set name="KeyStorePassword">yourkeypassword</Set>

<Set name="KeyStoreType">pkcs12</Set>

<Set name="KeyStoreProvider"><Property name="jetty.sslContext.keyStoreProvider"/
></Set>

<Set name="KeyManagerPassword">yourkeypassword</Set>

<Set name="TrustStorePath">install_path/certs/nfa-console-truststore.pfx</Set>

<Set name="TrustStorePassword">yourkeypassword</Set>

<Set name="TrustStoreType">pkcs12</Set>
```

NOTE

Use the keystore/truststore password created in Generate or Configure Certificates for Use by CA Network Flow Analysis for both the KeyStorePassword and TrustStorePassword .

8. Reboot the NFA Console to apply the changes.
9. Confirm that you can access the NFA Console using https.

Enable TLS for MYSQL Connections

NOTE

This section is applicable only to secure MYSQL communication.

Transport Layer Security (TLS) is a cryptographic protocol that provides communications security over a network.

MySQL performs encryption on a per-connection basis. With MySQL 5.7.22, TLS 1.2 is supported both for the database and for connections into the database.

Prerequisites

Before enabling TLS, create the required certificate files.

- Generate or Configure Certificates for Use by CA Network Flow Analysis

Enable TLS on a Standalone System

Follow these steps:

1. Edit *install_path*\MySql\my.ini and uncomment these lines:
   ```
   require_secure_transport=true
   tls_version=TLSv1,TLSv1.1,TLSv1.2
   ssl-ca=install_path/certs/nfa-ca-cert.pem
   ssl-cert=install_path/certs/nfa-console-cert.pem
   ssl-key=install_path/certs/nfa-console-key.pem
   ```
   Edit all occurrences of these lines. Use appropriate names for the *.pem files, and restrict the versions of TLS by removing those you do not wish to allow.
2. Change the ReporterAnalyzerWebSite Application Pool settings.
   a. Open **Internet Information Services (IIS) Manager**, select **Sites**, **Default Web Site**.
   b. Click **Basic Settings** in the Actions pane.
   The Edit Site window appears.
   c. Click **Test Settings** to verify the site connection.
   After the connection is successful, exit the Site Connection.

      NOTE

      If the connection is unsuccessful, follow the below steps **(step d - step h)**.
   d. Click **Connect as...** on the **Edit Site** window.
   The Connect As window appears.
   e. Select the **Specific user**, and click **Set** to enter the username and password.
   f. Click **OK** and exit the site connection.
   g. Click **Restart for Default Web Site**.
   h. Click **Ok**.
3. Edit *install_path*\Netflow\bin\netqosmy.ini and uncomment these lines:
   ```
   require_secure_transport=true
   tls_version=TLSv1,TLSv1.1,TLSv1.2
   ssl-ca=install_path/certs/nfa-ca-cert.pem
   ssl-cert=install_path/certs/nfa-console-cert.pem
   ssl-key=install_path/certs/nfa-console-key.pem
   ```
4. Edit all occurrences of these lines. Use appropriate names for the *.pem files, and restrict the versions of TLS by removing those you do not wish to allow.
5. Edit *install_path*\DBUsers\ReporterAnalyzer.ini :
   a. Everywhere requireSSL is set to false, set it to true.
   b. Make sure the keystore property points to the keystore created in:
   Generate or Configure Certificates for Use by CA Network Flow Analysis
   Use the same pkcs12 store for all, for example:
   install_path/certs/nfa-console-keystore.pfx
   c. Make sure the truststore property points to the truststore created in:

Generate or Configure Certificates for Use by CA Network Flow Analysis
Use the same pkcs12 store for all, for example:
install_path/certs/nfa-console-truststore.pfx

d. Set thekeystore/truststorepassword properties to the password used to generate them.
e. Set `SslCa` to the *install_path*/certs/ `nfa-ca-cert.pem` file.
f. Set `SslCert` to the *install_path*/certs/nfa-console-cert.pem file.
g. Set `SslKey` to the *install_path*/certs/nfa-console-key.pem file.

6. Restart the following services in this order.
   - CA MySql
   - NetQoS NQMySql
   - CA NFA Harvester
   - CA NFA Collection and Poller Webservices
   - CA NFA Data Retention
   - CA NFA DNS/SNMP Proxies
   - CA NFA File Server
   - CA NFA Poller
   - CA NFA Reaper

Enable TLS on a Distributed System

Follow these steps:

On the Console

1. Edit *install_path*\MySql\my.ini and uncomment these lines:

```
require_secure_transport=true
tls_version=TLSv1,TLSv1.1,TLSv1.2
ssl-ca=install_path/certs/nfa-ca-cert.pem
ssl-cert=install_path/certs/nfa-console-cert.pem
ssl-key=install_path/certs/nfa-console-key.pem
```

   Edit all occurrences of these lines. Use appropriate names for the `*.pem` files, and restrict the versions of TLS by removing those you do not wish to allow.
2. CChange the ReporterAnalyzerWebSite Application Pool settings.
   a. Open **Internet Information Services (IIS) Manager**, select **Sites**, **Default Web Site**.
   b. Click **Basic Settings** in the Actions pane.
   The Edit Site window appears.
   c. Click **Test Settings** to verify the site connection.
   After the connection is successful, exit the Site Connection.

      NOTE

      If the connection is unsuccessful, follow the below steps **(step d - step h)**.

   d. Click **Connect as...** on the **Edit Site** window.
   The Connect As window appears.
   e. Select the **Specific user**, and click **Set** to enter the username and password.
   f. Click **OK** and exit the site connection.
   g. Click **Restart for Default Web Site**.
   h. Click **Ok**.
3. Reboot the server to restart all services.

On the Harvester

1. Edit *install_path*\MySql\my.ini (or my.cnf on Linux) and uncomment these lines:

   ```
   require_secure_transport=true

   tls_version=TLSv1,TLSv1.1,TLSv1.2

   ssl-ca=install_path/certs/nfa-ca-cert.pem

   ssl-cert=install_path/certs/nfa-harvester-cert.pem

   ssl-key=install_path/certs/nfa-harvester-key.pem
   ```

   Edit all occurrences of these lines. Use appropriate names for the *.pem files, and restrict the versions of TLS by removing those you do not wish to allow.
2. Edit *install_path*\Netflow\bin\netqosmy.ini (or Netflow/my.cnf on Linux) and uncomment these lines:

   ```
   require_secure_transport=true
   tls_version=TLSv1,TLSv1.1,TLSv1.2
   ssl-ca=install_path/certs/nfa-ca-cert.pem
   ssl-cert=install_path/certs/nfa-harvester-cert.pem
   ssl-key=install_path/certs/nfa-harvester-key.pem
   ```

   Edit all occurrences of these lines. Use appropriate names for the *.pem files, and restrict the versions of TLS by removing those you do not wish to allow.
3. Edit *install_path*\DBUsers\ReporterAnalyzer.ini :
   a. Everywhere requireSSL is set to false, set it to true.
   b. Make sure the keystore property points to the keystore created in:
   Generate or Configure Certificates for Use by CA Network Flow Analysis
   Use the same pkcs12 store for all, for example:
   install_path/certs/nfa-harvester-keystore.pfx
   c. Make sure the truststore property points to the truststore created in:
   Generate or Configure Certificates for Use by CA Network Flow Analysis
   Use the same pkcs12 store for all, for example:
   install_path/certs/nfa-harvester-truststore.pfx
   d. Set the keystore/truststore password properties to the password used to generate them.
   e. Set SslCa to the *install_path*/certs/ nfa-ca-cert.pem file.
   f. Set SslCert to the *install_path*/certs/nfa-harvester-cert.pem file.
   g. Set SslKey to the *install_path*/certs/nfa-harvester-key.pem file.
4. Restart the following services in this order.
   – CA MySql
   – NetQoS NQMySql
   – CA NFA Harvester
   – CA NFA Collection and Poller Webservices
   – CA NFA Data Retention
   – CA NFA DNS/SNMP Proxies
   – CA NFA File Server
   – CA NFA Poller
   – CA NFA Reaper

Enable TLS 1.2 for HTTPS Connection

Transport Layer Security (TLS) is a cryptographic protocol that provides communications security over a network.

Prerequisites

Create the required certificate files before enabling TLS.

- Generate or Configure Certificates for Use by CA Network Flow Analysis

Follow these steps:

> **NOTE**
>
> Ensure that you follow these steps properly or you might lose the RDP connection.

1. Run gpedit.msc from the NFA installed system.
2. Navigate to Computer Configuration, Administrative Templates, Windows Components, Remote Desktop Services, Remote Desktop Session Host, Security.
3. Double click **Require use of specific security layer for remote (RDP) connections.**
4. Click **Enabled**.
5. From the Security Layer drop-down list, select **Negotiate.**
6. Click **OK**.
7. Back up the registry and create the following registry for TLS 1.0 and TLS 1.1:
   a. Key: HKLM SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS 1.0\Server
      Value: Enabled
      Value type: REG_DWORD
      Value Data: 0
   b. Key: HKLM SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS 1.1\Server
      Value: Enabled
      Value type: REG_DWORD
      Value Data: 0
8. Navigate to the *install_path\Portal\SSO\etc*.
9. Edit the *jetty-ssl-context.xml* file and add the following lines after tag *<Set name="ExcludeCipherSuites">.*

```
<Set>

.

.

<Array type="String">
 <Item>SSL_RSA_WITH_DES_CBC_SHA</Item>
 <Item>SSL_DHE_RSA_WITH_DES_CBC_SHA</Item>
 <Item>SSL_DHE_DSS_WITH_DES_CBC_SHA</Item>
 <Item>SSL_RSA_EXPORT_WITH_RC4_40_MD5</Item>
 <Item>SSL_RSA_EXPORT_WITH_DES40_CBC_SHA</Item>
 <Item>SSL_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA</Item>
 <Item>SSL_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA</Item>
 <Item>SSL_RSA_WITH_RC4_128_MD5</Item>
 <Item>TLS_RSA_WITH_RC4_128_MD5</Item>
 <Item>TLS_RSA_WITH_RC4_128_SHA</Item>
```

```
 <Item>SSL_RSA_WITH_RC4_128_SHA</Item>
 <Item>TLS_ECDHE_RSA_WITH_RC4_128_SHA</Item>
 <Item>SSL_ECDHE_RSA_WITH_RC4_128_SHA</Item>
 <Item>SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA</Item>
 <Item>TLS_DHE_RSA_WITH_AES_128_CBC_SHA</Item>
 <Item>TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA</Item>
 <Item>TLS_DHE_RSA_WITH_AES_128_CBC_SHA256</Item>
 <Item>TLS_DHE_RSA_WITH_AES_128_GCM_SHA256</Item>
</Array>
</Set>
<!--<Get name="sslContextFactory">-->
    <Set name="excludeProtocols">
     <Array type="java.lang.String">
     <Item>SSL</Item>
     <Item>SSLv2</Item>
     <Item>SSLv2Hello</Item>
     <Item>SSLv3</Item>
     <Item>TLSv1</Item>
     <Item>TLSv1.1</Item>
     </Array>
    </Set>
```

NOTE

The following steps 10 and 11 are required only for CA PC integration using HTTPS or TLS.

10. Edit *install_path*/RIB/start.ini .
    a. Comment the following lines to disable HTTP communication.

    ```
    --module=http
    jetty.port=8681
    ```

    b. Uncomment the following lines and edit as necessary (the `keystore.password` and `keymanager.password` should be the same value) to enable https communication.

    ```
    --module=https
    jetty.keystore=install_path/certs/nfa-console-keystore.pfx
    jetty.keystore.password=somepassword
    jetty.keymanager.password=somepassword
    jetty.truststore=install_path/certs/nfa-console-truststore.pfx
    jetty.truststore.password=somepassword
    https.port=8681
    ```

11. Edit *install_path*/RIB/etc/jetty-ssl.xml , and perform the following:
    a. To restrict to only certain transport layer protocols, uncomment and edit the following before the closing configures tag (this restricts to only TLS 1.2):

    ```
    <Call name="addExcludeProtocols">

        <Arg>

         <Array type="java.lang.String">
    ```

```
        <Item>SSL</Item>

        <Item>SSLv2</Item>

        <Item>SSLv2Hello</Item>

        <Item>SSLv3</Item>

        <Item>TLSv1</Item>

        <Item>TLSv1.1</Item>

      </Array>

     </Arg>

  </Call>
```

b. To exclude the ciphers, add the following lines after <*Set name="ExcludeCipherSuites"*> tag:

```
<Set>

  <Array type="String">

     <Item>SSL_RSA_WITH_DES_CBC_SHA</Item>

     <Item>SSL_DHE_RSA_WITH_DES_CBC_SHA</Item>

     <Item>SSL_DHE_DSS_WITH_DES_CBC_SHA</Item>

     <Item>SSL_RSA_EXPORT_WITH_RC4_40_MD5</Item>

     <Item>SSL_RSA_EXPORT_WITH_DES40_CBC_SHA</Item>

     <Item>SSL_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA</Item>

     <Item>SSL_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA</Item>

  <Item>SSL_RSA_WITH_RC4_128_MD5</Item>

  <Item>TLS_RSA_WITH_RC4_128_MD5</Item>

  <Item>TLS_RSA_WITH_RC4_128_SHA</Item>

  <Item>SSL_RSA_WITH_RC4_128_SHA</Item>
  <Item>TLS_ECDHE_RSA_WITH_RC4_128_SHA</Item>

  <Item>SSL_ECDHE_RSA_WITH_RC4_128_SHA</Item>
```

```
<Item>SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA</Item>

<Item>TLS_DHE_RSA_WITH_AES_128_CBC_SHA</Item>

<Item>TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA</Item>

<Item>TLS_DHE_RSA_WITH_AES_128_CBC_SHA256</Item>

<Item>TLS_DHE_RSA_WITH_AES_128_GCM_SHA256</Item>

  </Array>

</Set>
```

12. Confirm that ports 8382 and 8681 are open if firewalls are enabled.
13. Restart the following services in this order.
    - CA MySQL
    - NetQoS NQMySql
    - CA NFA Harvester
    - CA NFA Collection and Poller Webservices
    - CA NFA Data Retention
    - CA NFA DNS/SNMP Proxies
    - CA NFA File Server
    - CA NFA Poller
    - CA NFA Reaper

Security Settings for Protection Against BEAST and Weak Diffie-Hellman Moduli

This section describes how to protect the system from the following security attacks are identified:

- Information potentially disclosed due to weak Diffie-Hellman moduli
- Eavesdropping possible due to support of RC4 cipher

To protect the system from these security attacks, add the following to the registry:

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers
\RC4 128/128]

"Enabled"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers
\RC4 40/128]

"Enabled"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers
\RC4 56/128]

"Enabled"=dword:00000000
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL
\KeyExchangeAlgorithms\Diffie-Hellman]

"ServerMinKeyBitLength"=dword:0x00000800
```

AWS VPC for CA Network Flow Analysis

The current release of DX NetOps lets you view the following Amazon Web Services virtual private cloud network traffic flow reports:

- Source and destination IPv4 address
- Source and destination ports
- Protocols used
- Bytes and packets transferred

NOTE

Configure only one Harvester per VNA plug-in to collect the data from CA VNA. In the hybrid network, ensure that the IP address is unique and different from the IP address used in the cloud.

Prerequisites

Ensure that the CA Virtual Network Assurance and AWS plug-in is installed. Also, ensure that you can connect to the CA VNA server from your CA NFA instance. You can enable AWS VPC for CA Network Flow Analysis only on Windows.

Configure AWS VNA plug-in

Set up the connection from your CA NFA to the CA VNA, to enable communication between the two servers. The CA NFA uses the connection to fetch the network traffic flow reports. Perform the following steps on the CA NFA Harvester.

Follow these steps:

1. Open the `vnadatacollector.properties` file that is located at the `C:\CA\NFA\Netflow\bin` directory.
2. Add the values for the following properties:
   VNAIPAddress
   specifies the IP address of the VNA server
   VNAPort
   Specifies the port of the VNA server
3. Save and close the file.
4. Start the `CA NFA VnaDataCollector` service.

   NOTE
   By default, the `CA NFA VnaDataCollector` service is in Stopped state.

Configure Single Sign-On

Unsecured

The Single Sign-On tool authenticates you at start-up so you can log in once and can use several related products without logging in again. If you log in to CA Performance Center, for example, you can jump to information in DX NetOps without logging in again.

Single Sign-On is installed automatically when you install DX NetOps.

Notes:

- The Single Sign-On tool uses Lightweight Directory Access Protocol (LDAP) if LDAP is set up. The Single Sign-On tool does not use LDAP by default.
- For more information about configuring the Single Sign-On tool for LDAP authentication and other options, see Single Sign-On in the CA Performance Management documentation.

Secured

If you need to enable secure communication between CA Performance Center (CA PC) and DX NetOps (CA NFA) in your environment, complete the following tasks (in order):

1. Generate or Configure Certificates for Use by CA Network Flow Analysis
2. Enable TLS 1.2 for HTTPS Connection
3. Enable HTTPS for CA Network Flow Analysis

After completing these tasks, complete the steps in Set Up HTTPS for CA Single Sign-On . Prior to configuring CA NFA as a data source in CA PC, you will need to establish a trust relationship between CA PC and the NFA Console. This can be accomplished either by importing the “official” Certificate Authority certificate in use in your environment into the CA PC trust store, or by importing the CA NFA Certificate Authority certificate generated in item 1 (above) into the CA PC trust store.

Follow these steps:

1. Copy the certificate (`nfa-ca-cert.pem` in this example) to the CA PC server.
2. Run the `keytool` command to import the certificate.
   ```
   <CAPC_install_dir>/jre/bin/keytool -import -alias <some_alias> -trustcacerts -file nfa-ca-cert.pem -
   keystore <CAPC_install_dir>/jre/lib/security/cacerts
   ```
3. Run the `keytool` command to verify/print the newly imported certificate(s).
   ```
   <CAPC_install_dir>/jre/bin/keytool -list -keystore <CAPC_install_dir>/jre/lib/security/cacerts | grep -i
    <some_alias>
   ```
4. Configure DX NetOps as a data source in CA PC, making sure to choose https as the protocol.
   See the CA PC documentation for details.

Configure the Product to Work with Performance Center

Some essential administration tasks may be performed in either Performance Center or the NFA console.

Configuring Using Performance Center

To configure DX NetOps properly in Performance Center (either CA Performance Center or CA NetQoS Performance Center), perform the following tasks:

- Register the product as a data source for Performance Center.
- Set up the following elements in the Performance Center console:

  - SNMP profiles
  - (*Optional*) IP domains
  - User accounts, permissions, and roles
  - (*Recommended*) Groups
- Configure flow collection
  - Add Harvesters.
  - Enable the routers and interfaces to export flows.
- (*Optional*) Configure traps.
- (*Optional*) Verify that DX NetOps is receiving data.

Once you register the product and complete the administration tasks, operators can see meaningful reports in the NFA console and in the Performance Center console.

Register CA Network Flow Analysis

Register the product in the Performance Center Console, on the **Manage Data Sources** page (CA PC) or the **Data Source List** page (NPC).

> **NOTE**
> For information about the number of data sources that you can use, see the *Release Notes* for your Performance Center version.

Follow these steps:

1. Verify that no one else is running a session of DX NetOps. If multiple users write to the database simultaneously, problems can result.
2. Log in to the Performance Center Console as a user who has the Administrator role.
3. Click **Admin**, **Data Sources**.
   The current list of registered data sources are shown on the **Manage Data Sources** page (CA PC) or the **Data Source List** page (NPC).
4. Click **Add** (CA PC) or **New** (NPC).
   The **Add Data Source** dialog opens.
5. Select the type of data source you want to add from the **Source Type** list.

   > **NOTE**
   > All CA products that can be registered as data sources are shown in the **Source Type** list. The list is not filtered to hide products that are installed already.
6. Enter the **Host Name** of the data source.
   The hostname is the IP address or DNS hostname of the server that hosts the database for the data source. For a distributed deployment of the product, enter the hostname of the NFA console or stand-alone server.
7. Enter the port to use for contacting DX NetOps.
   For more information about the port setting, see Single Sign-On in the CA Performance Management documentation.
8. Select the protocol to use for contacting the data source. Select **https** if your network uses SSL for communications. Verify that you have configured the system correctly before you select the **https** option.

   > **NOTE**
   > For more information about using SSL for communication between the product and Performance Center, see Single Sign-On in the CA Performance Management documentation.
9. (Optional) Enter a **Display Name** for the DX NetOps data source.
   If you do not enter a name, a default name is created by combining the data source type and hostname. For example, you can use a default name like `NetworkFlowAnalysis@xxx.x.x.xx` or you can use a name like `NetworkFlowAnalysis_NewYork`.
10. Confirm whether the web console address is the same as the **Host Name**. If it is not, take the following steps:

- Clear the **Same as Data Source** check box.
- Provide the web console hostname, port, and protocol.

11. Click **Save**.
    The updated list shows the data sources that are registered.

Test Data Source Connections

In most cases, the status indicates that data source registration has completed successfully. If the status indicates an error, use the test feature on the **Manage Data Sources** page (CA PC) or **Data Source List** page (NPC).

The **Test** button initiates a test to confirm that a new data source is registered and connected correctly. The test checks for version compatibility and verifies that the data source is not registered with a different instance of the software.

If the test fails, verify that the DNS hostname or IP address of the data source server is correct.

Verify SNMP Profiles

DX NetOps sends secure SNMP information to Performance Center at registration. Subsequently, this information is transformed into SNMP profiles. *SNMP profiles* are definitions that contain the information necessary to enable secure queries of device MIBs using SNMP. Profile information is updated at each synchronization (every five minutes).

SNMP populates bandwidth, interface type, and router/interface description information.

During initial product setup, verify that the available SNMP profiles are adequate to monitor your environment. The available SNMP profiles are listed on the **Manage SNMP Profiles** page (CA PC) or **SNMP Profile List** page (NPC).

View the SNMP Profiles List

You can view a list of SNMP profiles that have already been defined. The list includes high-level information about the contents of each profile.

Tenant-Specific Profiles in CA Performance Center:

- If no tenant definitions have been created, the definitions in the **SNMP Profile List** are shared among all registered data sources. The global administrator sees a list of SNMP profiles that are not explicitly associated with a tenant.
- Tenant administrators only see the items that are associated with their tenant.

Follow these steps:

1. Log in as a user with the Administrator role.
2. Display the list of SNMP profiles:
   - CA PC: Select **Admin, System Settings: SNMP Profiles**.
   - NPC: Select **Admin, NetQoS Settings: SNMP Profiles**.

   The page displays the current list of SNMP profiles.
   The following information is listed for each profile:
   - **Order**
     Determines the order in which the secure information contained in an SNMP profile is used to try to query a selected device. If the query fails, the next profile is used, in priority order.
   - **Profile Name**
     Defines a name for the SNMP profile. Profile names must be unique, cannot be duplicated across SNMP versions, and are not case-sensitive.
   - **Port** (CA PC Only)
     Identifies the port that is used to make SNMP connections to devices.

Default: UDP 161.

- **SNMP Version**
  Specifies the version of SNMP that the profile uses. Because SNMPv1 and SNMPv2C are similar from a security standpoint, they share a single option. SNMPv3 is a separate option.
- **Authentication Protocol**
  (SNMPv3 Only) Specifies the authentication protocol to use when contacting devices associated with this profile. The following algorithms for authenticating SNMPv3 packets are supported:
  - **None** (do not attempt authentication)
  - **MD5** (Message Digest 5)
  - **SHA** (Secure Hash Algorithm)
- **Privacy Protocol**
  Identifies the encryption protocol that is used to contact associated devices, if any. Always **None** if no authorization protocol is in use.
- **Use by Default**
  Indicates whether the information in this profile is used when not explicitly assigned to a device. If disabled, this profile is excluded from discovery in data sources that support the exclusion of profiles.

To perform any action on this page, select a profile, then click a button.

Add SNMP Profiles

Administrators can create SNMP profiles in the Performance Center Console. You can create SNMPv1/v2C or SNMPv3 profiles.

Follow these steps:

1. Log in to the Performance Center Console as a user with the Administrator role.
2. Display the list of SNMP profiles:
   - CA PC: Select **Admin, System Settings: SNMP Profiles**.
   - NPC: Select **Admin, NetQoS Settings: SNMP Profiles**.

   The page displays the current list of SNMP profiles.
3. Click **New**.
   The **Add SNMP Profile** dialog opens.
4. Complete the fields and change default settings as needed. Some fields apply only to SNMPv3 or SNMPv1/v2C.
   - **Profile Name**
     Defines a name for the SNMP profile. Profile names must be unique, cannot be duplicated across SNMP versions, and are not case-sensitive.
   - **SNMP Version**
     Specifies the version of SNMP that the profile uses. Because SNMPv1 and SNMPv2C are similar from a security standpoint, they share a single option. SNMPv3 is a separate option.
   - **Port**
     (Optional for SNMPv1/v2C) Identifies the port that is used to make SNMP connections to devices associated with this profile.
     Default: UDP 161.
   - **User Name**
     (SNMPv3 Only) Identifies the user for the profile, whose secret keys were used potentially to authenticate and encrypt the SNMPv3 packets. The **User Name** is a character string.
   - **Context Name**
     (SNMPv3 Only) Specifies a collection of management information that is accessible by an SNMP entity. The **Context Name** is necessary for providing end-to-end identification and for retrieving data from an SNMPv3 agent. The **Context Name** is an octet string.
   - **Community Name**

(SNMPv1/v2C Only) Defines a secure string that lets the data source query the MIB of the associated device. The community that you supply must provide read-only access to the device MIB.

NOTE
In the default SNMP profile, the community is 'public'.

- **Verify Community Name**Confirms the secure community string (name).
- **Authentication Protocol**
  (SNMPv3 Only) Specifies the authentication protocol to use when contacting devices associated with this profile. The following algorithms for authenticating SNMPv3 packets are supported:
  - **None** (do not attempt authentication)
  - **MD5** (Message Digest 5)
  - **SHA** (Secure Hash Algorithm)
- **Authentication Password**
  (SNMPv3 Only) Specifies the password for authentication using SNMPv3 and the selected authentication protocol.

  NOTE
  For SNMP profiles that are used with DX NetOps, make sure you specify a password that is at least eight characters long. If the password does not meet this requirement, the SNMP profile will be invalid and no SNMP data will be returned when the SNMP profile is used for polling. In this case the corresponding interfaces and devices in views, reports, and dashboards are missing device names, interface names, interface speeds, and interface utilization data.
- **Verify Authentication Password**
  Confirms the authentication password.
- **Privacy Protocol**
  (Optional) Specifies the encryption protocol to use for data flows sent to any devices or servers associated with this profile. Select one of the following protocols to create a valid SNMP profile for DX NetOps polling:
  - **None** (do not encrypt communications)
  - **DES**
  - **AES (128-bit encryption)**
  - **Triple DES**

  NOTE
  If **AES 192** and **AES 256** options are listed, do not select either of those options: They are not supported for DX NetOps. If the SNMP profile you create is not valid, no SNMP data is returned for devices and interfaces that use it. In this case the corresponding interfaces and devices in views, reports, and dashboards are missing device names, interface names, interface speeds, and interface utilization data.

  The privacy protocol option is not enabled until authentication is enabled for this profile.
- **Privacy Password**
  Defines the password that is used when exchanging encryption keys. See the Note in **Authentication Password** for a possible length requirement.
- **Verify Privacy Password**
  Defines the password used when exchanging encryption keys.
- **Use by default for new devices** (CA PC)
  Enable this profile for auto-discovery (NPC)
  Specifies whether the profile is used by default to contact any new devices that are discovered from monitored traffic. If it fails, the next profile in priority order is used. Disable this parameter to exclude a profile from discovery.

5. Click **Save**.
6. You return to the list of SNMP profiles. The new profile appears in the list.
   Performance Center performs a global synchronization to send the profile information to DX NetOps.

Verify IP Domains

The IP domains feature helps to address potential IP address conflicts. Domain identifiers indicate that two managed items that otherwise appear as duplicate IP addresses are actually two *different* managed items.

IP domains also let a global administrator in Performance Center control which managed items are visible to and accessible by a particular administrator or user.

Information about custom IP domains is sent down to the data sources during synchronization. Domains are available for use during configuration. You can use the NFA console Administration functions to add interfaces, custom virtual interfaces (CVIs), routers, and Harvesters to the custom domains that you create.

During initial setup, verify that the existing IP domains are adequate to monitor your environment. To see the domains, complete one of the following actions in the Performance Center Console:

- (CA PC) Select **Admin, IP Domains**, and review the domains on the **Manage IP Domains** page.
- (NPC) Select **Admin, Groups**, and expand the **All Groups** tree on the **Manage Groups** page.

View the IP Domains List

IP domains are required for monitoring multiple environments with overlapping IP addresses. Set up all the domains that you need before you begin to export flow data.

Follow these steps:

1. Log in to the Performance Center Console as a user with the required administrative role rights.
2. Display the current domains by completing one of the following actions:
   - (CA PC) Select **Admin, Custom Settings**: **IP Domains**.
     The **Manage IP Domains** page opens and displays the current domains.
   - (NPC) Select **Admin, NetQoS Settings**: **Groups**, and expand the **All Groups** tree on the **Manage Groups** page. The current domains are shown under **All Domains**. To display the parameters for a domain in CA NetQoS Performance Center, select the domain and click the **Properties** tab.

If you have not created any custom IP domains, only the **Default Domain** appears in the list. This predefined domain has a 'null' setting for all parameters.

Any custom domains that you have created include values for the following parameters:

- **Name**
  Identifies the domain.
- **Description**
  (Optional) Describes this domain namespace, such as naming the enterprise that owns it.
- **Primary DNS Address**
  Is the IP address of the primary name server for this domain.
- **Primary DNS Port**
  Is the port number that the primary name server uses.
- **Secondary DNS Address**
  Is the IP address of the secondary name server for this domain. Can be the same as the primary address.
- **Secondary DNS Port**
  Is the port number that the secondary name server uses.

Add Custom IP Domains

Administrators can set the domain assignment for Harvesters, routers, interfaces, and custom virtual interfaces (CVIs). We recommend that you set up any custom tenants and domains you need before you add the Harvesters.

Having the appropriate IP domains set up helps to achieve the following goals:

- Assign the correct tenant-domain when you add Harvesters so that their routers and interfaces inherit the correct associations. The routers have the appropriate SNMP profiles available to poll their interfaces.
- Make specific content accessible only to the operators who need to monitor it.
- Enable Administrators to create domain-specific ToS labels, protocol groups, and Autonomous System (AS) names in DX NetOps.
- Avoid IP address conflicts.
  For example, suppose a router with a single IP address has interfaces that belong to different enterprises. The domain identifiers clarify that the interfaces are different managed items, even though they have the same IP address.

The **Default Domain** is created automatically. The **Default Domain** includes any items that are not assigned to a custom domain.

Follow these steps:

1. Log in to the Performance Center Console as a user with the required administrative role rights.
2. Display the current domains by completing one of the following actions:
   – (*CA PC*) Select **Custom Settings: IP Domains**.
     The **Manage IP Domains** page opens and displays the current domains.
   – (*CA NPC*) Select **Admin, NetQoS Settings: Groups**, then expand the **All Groups** tree on the **Manage Groups** page.
     The current domains are shown under **All Domains**.

   If you have not created any custom IP domains, only the **Default Domain** appears in the list.
3. Create a new domain:
   – (*CA PC*) Click **New**.
     The **IP Domains Administration** dialog opens.
   – (*CA NPC*) Right-click **All Domains** and select **Add New Domain**.
     The **Add Domain** dialog opens.
4. Supply information for the following parameters:
   – **Domain Name**
     Identifies the domain.
   – **Description**
     (Optional) Describes this domain namespace, such as naming the enterprise that owns it.
   – **Device Name Alias**
     (*CA PC only*) Indicates the alias to use for a managed device. A device alias is a user-configured name that is applied to the associated managed item in CA Performance Center. Click Browse to navigate to and import a CSV file of aliases. The CSV file contains a comma-separated list of IP address-to-device alias mappings.
     Aliases that are associated with the primary IP address of a device take precedence over aliases that are associated with any secondary IP addresses. Look for the primary IP address in the Address column of the Inventory Devices list. We recommend always using the primary IP address of the device in the CSV file.
     For example:

     ```
     172.24.36.107,Austin Router
     ```

     Browse to select the file and click **Open**.
     If you include aliases for devices you are managing already, it can take up to 5 minutes to begin synchronizing these aliases with CA Performance Center.

     NOTE
     To remove an alias, import a CSV file that includes the IP address for the device and a blank alias column. To change an alias, modify the alias entry in the CSV file and reimport the file.
   – **Interface Description Override**

(*CA PC only*) Indicates the alternate description to use for an interface. Interface descriptions appear in CA Performance Center already, but you can provide an alternate description. Click **Browse** to navigate to and import a CSV or TXT file of alternate descriptions. The file contains a comma-separated list of values that include the device IP address, interface name, interface description, and alternate interface description (alias) mappings.
For example:

```
172.24.36.107,ethernet_7,vmxnet3 Ethernet Adapter,Connection to Dallas
```

NOTE
Use the primary IP address of the associated device in the CSV or TXT file. Secondary IP addresses are not supported. Look for the primary IP address in the **Address** column of the **Inventory Devices** list.

Browse to select the file and click **Open**.
If you include alternate descriptions for interfaces you are managing already, it can take up to 5 minutes to begin synchronizing these descriptions with CA Performance Center.

NOTE
You can use the same alternate interface descriptions for more than one interface.To remove an alternate description, import a CSV or TXT file that includes the IP address for the device, the interface name, the interface description, and a blank alias column. When you remove an alternate description, the original interface description reappears in CA Performance Center views.

WARNING
If you use a spreadsheet program to remove all of the alternate descriptions from a `CSV` file, include a column heading for the interface description override column in the imported file. If you do not include this column heading, the original interface descriptions will not reappear in CA Performance Center views.

To change a description, modify the alias entry in the `CSV` or `TXT` file and re-import the file.

- **DNS Settings check box**
  (*CA PC only*) If selected, displays the Primary DNS/Port and Secondary DNS/Port options.
- **Primary DNS Address**
  Is the IP address of the primary name server for this domain.
- **Primary DNS Port**
  Is the port number that the primary name server uses.
- **Secondary DNS Address**
  Is the IP address of the secondary name server for this domain. Can be the same as the primary address.
- **Secondary DNS Port**
  Is the port number that the secondary name server uses.
- **Enable DNS Proxy Address**
  (*CA NPC only*) Indicates whether the proxy address is enabled for this IP domain.
- **DNS Proxy Address**
  (*CA NPC only*) Is the IP address of the DNS proxy server.
  This setting is required only if your network is located behind a DNS proxy server.

5. Click **Save**.The new IP domain appears on the page.
6. Repeat the steps as required to add more IP domains.

Configure Flow Collection

The next step is to configure the routers in DX NetOps and verify that they are sending data to the Harvesters.

DX NetOps can begin to collect flows after you complete the following tasks:

- Recommended: Add the domains that you need.

  NOTE
  More information:

  Add Custom IP Domains
- Add the Harvesters.
- Configure the routers and interfaces to export flow data.

Add One or More Harvesters

Add one or more Harvesters to enable data to be processed and displayed.

We recommend registering DX NetOps and setting up the domains before you add any Harvesters.

Follow these steps:

1. Open the NFA console, logged in with Administrator rights. For example, enter the following address in a browser:
   `http://<ipaddress>/ra/`
   User name: `admin`
   Password: `admin`
2. Open the **Harvester** page:
   a. Select **Administration** from the NFA console menu.
      The **Administration** page opens.
   b. Select **System, Harvester** from the menu on the left side of the page.
      The **Harvester** page opens and displays the current list of Harvesters.
3. Click **Add**.
   The **Add Harvester** dialog opens.
4. Enter the following information:
   – **IP Address**
     Address of the Harvester server.
   – **Description**
     Identifying text about the Harvester, which appears in the Harvester page table.
   – **Domain**
     Parent tenant and domain combination for the Harvester and for any routers and interfaces that begin to supply data to the Harvester.
     Changing this setting affects the tenant-domain association for new routers that begin exporting flow data and any new interfaces that begin generating flow.
     In a multi-tenant environment, the Harvester tenant affects which SNMP profiles are available for routers to poll interfaces.
     The domain affects which operators and reports have access to the data from routers and interfaces.
     This option is visible only in an environment that contains multiple domains.
5. Click **Save**.
   The new Harvester is added and appears in the Harvester list, provided that the IP address passes the connection test. If the test connection to the web service fails, an error message opens.
   The usual process is to add one or more Harvesters, then configure the router interfaces to export flow to the Harvesters. If you configure the routers to export flow to the Harvesters first, the NFA console immediately begins to collect data from the new Harvester. In this case, the domain for the routers is set at the time you add the parent Harvester.

   NOTE
   Make sure the Harvesters you add have not been deleted from the **Harvester** page previously. To add a Harvester instance successfully in DX NetOps after deleting it, the Harvester installation server must be re-imaged and the Harvester software must be re-installed.

Verify the Harvester Domain

Verify that each Harvester is associated with the appropriate domain before you set up routers to export flow data. If you have not already done so, set up any needed custom tenants and domains before you proceed.

> **NOTE**
> The tenant feature is applicable only to deployments that include CA Performance Center. If your deployment uses CA NetQoS Performance Center, the tenant setting is **Default Tenant**.

Follow these steps:

1. Open the **Harvester** page:
   a. Select **Administration** from the NFA console menu.
      The **Administration** page opens.
   b. Select **System, Harvester** from the menu on the left side of the page.
      The **Harvester** page opens and displays the current list of Harvesters.
2. Click **Edit** on the row for the Harvester that you want to edit.
   The **Edit Harvester** dialog opens.
3. (*Optional*) Change the **Domain** setting (tenant-domain combination) as needed.
   Default: Default Tenant \ Default Domain.
   You can also change the **IP Address** and **Description**.
   If no custom IP domains have been created, the **Harvester** table includes only the **IP Address** and **Description** columns.
4. Click **Save** when your changes are complete.
   Your changes are saved immediately.

Set Up the Routers

Enable NetFlow on each DX NetOps router by completing the following steps. You can configure routers to export any of the following flow protocols:

- NetFlow v5, v7, v9, and Random Sampled NetFlow
- sFlow version 5
- IPFIX, J-Flow, cFlow, and NetStream flow that complies with the standards for NetFlow v5, v7, or v9

Notes:

- Configure flow from each source to be exported to a single Harvester. If flow from one source is exported to multiple Harvesters, a number of problems result. If this occurs, contact CA Support for help.
- NetFlow provides a broad view of your network packet streams by creating flow records for all packets. The data from these flow records represents all packets. Sampled NetFlow/IPFIX and sFlow take samples from your packet streams, producing fewer flow records and lessening the impact to a collector. The lower your sampling rate, the less precise the data is likely to be.
- Cisco documentation notes that "Sampled NetFlow does not allow random sampling and thus can make statistics inaccurate when traffic arrives in fixed patterns." Therefore, if you choose NetFlow v9 Sampling, you must use Random Sampled NetFlow. Refer to the Cisco documentation for your hardware and IOS version for more details.

The CA Network Flow Analysis 10.0.2 supports SAMPLING_INTERVAL (34) and SAMPLING_ALGORITHM (35) additional NetFlow fields in Options Template for sampling rates. For the SAMPLING_ALGORITHM (35), only random sampling algorithm (0x02) is supported. A netflow from a SourceId must have one of the following set of fields:

- FLOW_SAMPLER_ID(48), FLOW_SAMPLER_MODE (49), FLOW_SAMPLER_RANDOM_INTERVAL (50) fields or
- SAMPLING_INTERVAL (34) and SAMPLING_ALGORITHM (35)

For data from non-sampled flows to appear in reports of 15-minute (historical) data, these minimum fields are required:

- One of the following:

  - 1 - IN_BYTES
  - 85 - IN_PERMANENT_BYTES
  - 231 - FW_INITIATOR_OCTETS
  - 232 - FW_RESPONDER_OCTETS
- 4 - PROTOCOL
- 7 - L4_SRC_PORT
- 8 - IPV4_SRC_ADDR/ 27 - IPV6_SRC_ADDR
- 10 - INPUT_SNMP
- 11 - L4_DST_PORT
- 12 - IPV4_DST_ADDR/ 28 - IPV6_DST_ADDR
- 14 - OUTPUT_SNMP

Follow these steps:

1. Back up the current router configuration.
2. Configure NetFlow export for each interface individually:
   a. Set the flow export version.
   b. Set the flow source IP address. Cisco recommends that you configure a loopback source interface. The IP addresses of non-loopbacked interfaces can change.
   c. Set the flow destination IP address and set the destination port to 9995. If you are using a custom value for the harvester listening port, use that value as the destination port. The port values must match or the Harvester does not receive flow data.
   d. Set the flow expiration timeout to 1 minute.
3. Enable flow for each interface.
   - NetFlow v5 or v5-compatible flow:
     - Monitoring multiple interfaces on a router: Use either all ingress or all egress. Use the same option for all of the interfaces. Ingress and egress values may vary slightly due to routers dropping packets and changing ToS values as traffic travels between interfaces.
     - Monitoring a single known interface on a router: Use ingress and egress. This option results in fewer total flows from the router to the Harvester and puts less load on the network and the Harvester.
   - NetFlow v9 or v9-compatible flow:
     The Harvester identifies and deduplicates multiple flows on a single router, so you can use ingress and egress on multiple interfaces. You may find it most efficient to use this option for two or three interfaces. You have the option to enable ingress and egress across all interfaces, but this configuration may put an unnecessary burden on the Harvester.
4. Configure SNMP index persistence on each router that supports this feature.

NOTE
More information:

- NetFlow Version 9 Flow-Record Format

Configure ifTypes

The interfaces from which DX NetOps (CA NFA) will process flow data are configurable based on ifType. There are interface types that typically will not export flow data, and DX NetOps ignores these ifTypes:

- 1
- 18
- 37
- 100
- 101
- 102
- 103
- 104
- 134

However, you may have a different router configuration, and need to modify the ifTypes on this list.

Follow these steps:

1. Locate the `poller.properties` file.

   install_path\Netflow\bin\poller.properties

   Open the file in a text editor.
2. The ifTypes that CA NFA will ignore are included on the following line:

   ```
   ifTypesToIgnore=1,18,37,100,101,102,103,104,134
   ```

   Edit this list to add or remove ifTypes as necessary. The list must contain integer values, separated by commas, and containing no spaces.
3. Save the edited file.
4. Restart the poller service (CA NFA Poller).

NOTE
More information:

- IANAifType-MIB Definitions

Verify That Data Is Received

To verify that data is received, complete the following tasks:

Verify That the Interfaces Are Enabled

Once you configure DX NetOps to receive flow data, verify that the expected interfaces are monitored.

Follow these steps:

1. Open the NFA console, logged in with Administrator rights.
2. Open the **Available Interfaces** page:
   a. Select **Administration** from the NFA console menu.
      The **Administration** page opens.
   b. Select **Interfaces, Enable Interfaces** from the menu on the left side of the page.
      The **Available Interfaces** page opens.
3. Expand the router details to display the interface list: Click the arrow to the left of the router name.
4. Review the list to review whether the interfaces are enabled.
5. Change the interface status: Select the check box next to one or more interfaces, then click **Enable** or **Disable**.
   The selected interfaces are immediately enabled or disabled.
6. Repeat these steps for each router.

Verify That the Interfaces Are Visible in the NFA Console

Make sure that the configured interfaces are visible in the NFA console.

1. Verify that interfaces are visible on the **Enterprise Overview** page:
   a. Log in to the NFA console.
   b. Click **Enterprise Overview** in the main menu.
   c. Make sure the report views show interfaces.
2. Verify that interfaces are visible in the **Interface Index**:
   a. Click **Interfaces** in the NFA console menu.
      The **Interface Index** opens.
   b. Make sure that the **Interface Index** includes the interfaces that you expect to see.
      You can locate interfaces by expanding router details to view interfaces. Alternatively, you can use the **Search** box to find routers or interfaces by entering all or part of a name or description.
3. If interfaces are not visible, perform the following preliminary troubleshooting tasks:
   – Verify that the DX NetOps services are running on the NFA console server.
   – Review the logs. View the logs in the NFA console or open the logs from:

     install_path\reporter\Logs

Change the Domain of Interfaces and CVIs

Interfaces and CVIs inherit their initial tenant-domain setting from the parent router and Harvester when the Harvester is added and the program begins to collect data from the router and interfaces. If the Harvester is not associated with a custom domain, the routers and interfaces are assigned to the Default Domain as their data begins to be collected.

You can edit the settings for interfaces and CVIs at any time. The domain setting does not have to match the parent router or Harvester. Changing the domain can affect which operators and reports have access to the interfaces' data.

> **NOTE**
> The tenant feature is applicable only to deployments that include CA Performance Center. If your deployment uses CA NetQoS Performance Center, the tenant setting is **Default Tenant**.

Follow these steps:

1. Open the **Active Interfaces** page:
   a. Select **Administration** from the NFA console menu.
      The **Administration** page opens.
   b. Select **Interfaces, Physical & Virtual** from the menu on the left side of the page.
      The **Active Interfaces** page opens.
2. Locate and select the check box next to one or more interfaces that you want to associate with a tenant-domain.
   – To search for parent routers, interfaces, or CVIs, enter all or part of a router IP address, a router or interface name, or an interface description in the **Search** field, and then click **Search**. Expand the router details.
   – To navigate to an interface or CVI manually, go to the page that contains the parent router and click the arrow next to the router name. The router details expand to show the interfaces and CVIs.
3. Click **Edit**.
   The editing dialog opens. The **Domain** selection list is included in the dialog only if multiple domains exist.
4. Select a tenant-domain option from the **Domain** list.
5. Click **Save**.
   The dialog closes. The changes are shown on the **Active Interfaces** page.

> **NOTE**
> You can also change the tenant-domain setting for Harvesters and routers.

Configure Traps

Trap configuration is complete when you have finished the following tasks:

- Create the traps that you need.
- Enable traps to be displayed as events in the Performance Center Console:
  a. Open the **Application Settings** page in the NFA console.
  b. Set the **Trap Destination** value to match the IP address of one of the following servers:
     - (*CA PC*) NFA console or stand-alone server that is registered as a data source
     - (*CA NPC*) Event Manager server
- (*Optional*) Enable Watchdog trap notifications to be sent to your trap receiver:
  Open the **Watchdog Settings** page in the NFA console. Configure values for the **Trap Destination**, **Email Address**, and other Watchdog settings.
- (*Optional*) Verify that the events are displayed on the Performance Center Console, on the **Events** page (CA PC) or the **Event List** page (CA NPC). If the events are not shown as expected, verify that the following conditions are met:
  – The logs show that events have been generated and have been forwarded to the Event Manager.
  – The Event Manager host name is resolvable by the DNS server for DX NetOps.
  – The **Trap Destination** value on the **Application Settings** page in the NFA console matches the IP address of one of the following servers:
    - (*CA PC*) NFA console or stand-alone server that is registered as a data source
    - (*CA NPC*) Event Manager server
  – (*CA NPC Only*): The Event Manager is installed.

Set Up User Accounts

One predefined user account, admin, is included with the installation of DX NetOps. The admin account has full administrative privileges.

The administrator must create a user account for each person who will use the product--administrators and operators. Custom user accounts enhance security and take advantage of the narrowly defined role rights that determine access to product features and data.

Custom user accounts are best deployed in a well-planned system that includes custom groups. Custom groups are assigned as permissions to let product operators view only the data, menus, and dashboards that they need to perform their daily tasks.

View a List of User Accounts

You can see a high-level overview of user account settings in the Performance Center Console. Until you create custom user accounts only the two factory user accounts are available.

Follow these steps:

1. Log in to the Performance Center Console as a user with the required administrative role rights.
2. Click **Admin, User Settings**: **Users**.
   The current list of user accounts is displayed on the **Manage Users** page (CA PC) or the **User List** page (NPC).
   The table includes the following information about each user account:
   – **Name**
     Is a login name for the user account.
   – **Role**
     Is the role assigned to the user account.
   – **CAPC Privilege / NPC Privilege**

Identifies the level of access to registered data sources, such as DX NetOps.

- **Permission**
  Lists the permission groups that are assigned to this account. You can view permission groups as nested locations in the **Groups** tree.
  Default: '/All Groups'.
- **Status**
  Indicates whether the user account is enabled or disabled.

  NOTE
  Additional status values may be listed in the Console for CA NetQoS Performance Center: Built-in (configured automatically) and Online (currently logged in).

To perform any action on this page, click one of the buttons along the bottom.

Add User Accounts

Add a user account for each person who will operate the products. For security purposes, operators should not share user accounts.

NOTE
This topic describes the steps to perform this task in CA Performance Center. If you register DX NetOps as a data source for CA NetQoS Performance Center, the steps are slightly different.

Follow these steps:

1. Log in to the Performance Center Console as a user with the required administrative role rights.
2. Confirm that the required roles and groups exist.
3. Select **Admin, User Settings**: **Users**.
   The current list of user accounts is displayed.
4. Click **New**.
   The **Create New User** wizard (CA PC) or **Add User** page (CA NPC) opens.
5. Enter information for the following account parameters:
   - **Name**
     Is a login name for the user account. Limited to 50 characters.
   - **Description**
     (Optional) Describes the user account to help you identify it.
   - **Email Address**
     (Optional) Associates an email address with the user account.
   - **Preferred Language** *(CA PC only)*
     Specifies the language spoken by the operator associated with the user account.
   - **Authentication Type**
     Identifies the authentication method that applies to this user account. The method must match Single Sign-On configuration. Select one of the following:
     - **Performance Center** *(CA PC) or* **Product** (NPC): The default authentication scheme deployed by Performance Center.
     - **External**: A third-party authentication scheme, such as LDAP or SAML.
   - **Password**
     Defines a password for the user account. The password is limited to 32 characters.
   - **Time Zone**
     Corresponds to the time zone in which the user will view data.
     Default: UTC (Coordinated Universal Time).
   - **Role**
     Is the role assigned to the user account.
   - **Account Status** *(CA PC only)*

Determines whether the account is activated for use (Enabled).

- **User Options** *(CA NPC only)*
  Determines whether the account is activated for use (Enabled) and whether the user is allowed to share views with other products (Allow user to generate view URLs).

6. Assign access permissions to the user.
7. Add permission groups to the user account.
8. Click **Save**.
   The new user appears in the list of user accounts.

Role Rights

The rights that are assigned to each role determine user access to dashboards and menus. For example, role rights control the types of views that users can see and control whether users can export data, customize settings, and set up schedules for sending email reports.

Administrators can grant additional rights to users by editing their role. The **Edit Role** dialog lists role rights that are assigned to roles. The **Manage Users** or **User List** page shows the role that is assigned to each user.

> **NOTE**
> Do not remove the administrative role rights from your primary administrator account. Administrative access to the console is required.

Add Role Rights for Users

If the predefined user roles do not fit your requirements, you can add custom user roles. Ideally, you create the roles that each unique product operator needs to be able to perform his or her job responsibilities.

Custom roles work best within a system of custom groups. Custom groups let you precisely grant access to dashboards and product features while restricting access to sensitive data. The same groups that you create to organize data can serve as “permission groups” when you set up user account permissions.

A new role has no rights until you add them.

Follow these steps:

1. Log in to the Performance Center Console as a user with the required administrative role rights.
2. Navigate to the **Manage Roles** or **Roles List** page.
   The page displays the current list of roles.
3. Click **New**.
   The **Add Role** dialog opens.
4. Supply the required information and make selections in the fields provided:
   - **Name**
     (Optional) Identifies the role. Limited to 45 characters.
   - **Description**
     (Optional) Describes the role. For example, identifies the job-related duties that the associated user performs.
   - **Role Status**
     Select **Enabled** to make the role active. Required to give users with this role the access granted by role rights.
5. Specify the menus that will be visible to users with the new role:
   a. Select **Menu Set** (CA PC) or select a menu or product from the list at the bottom of the dialog (NPC).
   b. Click **Edit**.
      The **Edit Menu Set** dialog opens. Menus in the **Available Menus** list can be added to the role.
   c. Click an item on the left that you want to add to the role, then click the right arrow.
      Use **Shift + Click** or **Ctrl + Click** to select multiple items.
      Each selected item moves to the **Selected Menus** list.

d. (Optional) Use the Up and Down arrows to move items around in the list. The order of menus in the list determines their order on the **Dashboards** tab.
e. Click **OK**.
You return to the **Add Role** page.

6. Set the Performance Center rights for the role:
   a. Select **Performance Center** (CA PC) or **NetQoS Performance Center** (NPC).
   b. Click **Edit**.
   A dialog opens, which you use to select Performance Center access rights.
   c. Click an item on the left that you want to add to the role, then click the right arrow.
   The access right moves to the **Selected Rights** list.
   d. (Optional) Use the Up and Down arrows to move items around in the list. The order of role rights determines their priority in cases where rights overlap.
   e. Click **OK**.
   You return to the **Add Role** page.
7. Set the DX NetOps rights for the role:
   a. Select the name of the registered DX NetOps instance.
   b. Click **Edit**.
   A dialog opens, which you use to select access rights for DX NetOps in the same way you selected access rights for Performance Center.
   c. When the access rights are set up correctly, click **OK**.
   The new role is created and appears in the **Role List**.
8. Repeat the previous step to set the rights for any additional data source that you want to include.
9. Click **Save** on the **Add Role** page.
   You return to the **Manage Roles** page (CA PC) or **Roles List** page (NPC).

> **NOTE**
> When you finish creating a role, assign it to a user account as a separate step. Roles are inoperative until they are assigned to user accounts. Only users with the 'Administer Users' and 'Administer Roles' role rights can assign roles to user accounts.

Assign Permission Groups to User Accounts

Individual operators require data access permissions to monitor data in the products. Access permissions are based on groups. You can assign access permissions according to your plan for custom groups. Your goal as the administrator is to make sure that all operators see only the data they require to do their job.

For example, suppose you create custom groups and assign them as permissions to IT staff. When staff members log in to Performance Center, they can view data from the systems that are assigned to them.

Follow these steps:

1. Log in to the Performance Center Console as a user with administrative privileges.
2. Click **Admin, User Settings: Users**.
   The **Manage Users** page (CA PC) or **User List** page (CA NPC) opens.
3. Select a user account that you want to change, and click **Edit**.
   The **Edit User** wizard or dialog opens.
4. Display the permission groups:
   – (CA PC) Click the **Access Permissions** button.
   – (CA NPC) Locate the **Permission Groups** pane in the middle of the page.
   The group settings are displayed.
5. Add permission groups to the user account

- Expand the groups in the **Available Groups** tree on the left so that subgroups are shown.
- Select a group or subgroup.
- Click the right arrow or **Add** button to add the group.
- Repeat as necessary.

The selected permission groups appear in the **Selected Groups** pane.

6. Select the default group for the user; the data that appears by default in the dashboards for the user:
   - (CA PC) Right-click the target group and select **Make Default**.
   - (CA NPC) Select the target group and click **Make Default**.
7. Click **Save**.
   The changes are saved to the user account, and you return to the **Manage Users** page.
   When the user logs in, data from the default group appears in dashboards by default.

Assign Product Privileges

Each registered data source has its own product privilege setting, which grants unique privileges within that product interface. Administrators give users product privileges for each data source. For example, the product privilege determines whether a user can log in to DX NetOps or drill down from a Performance Center view to details in the NFA console. Privileges are specific to the data source instance.

The default administrator account, admin, is locked to prevent changes to product privileges. This account must have Administrator privileges for all registered data sources. If you select a group of accounts that includes the admin account, you cannot edit the product privileges for any of the selected accounts.

DX NetOps Product Privileges

A user must have product privileges for the DX NetOps data source to log in to the NFA console. Product privileges also determine whether a user can access the **Administration** page, and can perform certain functions:

- **Administrator**
  Gives access to the **Administration** page in the NFA console and to all functions. Functions include creating and managing user accounts, roles, groups, SNMP profiles, and scheduling for reports.
- **Power User**
  Gives user-level access and any additional abilities that the Role setting grants. For DX NetOps, the Power User privilege is equivalent to the Administrator privilege.
- **User**
  Gives access to **Top Interfaces** reports and **Interface Utilization** reports on the **Enterprise Overview** page.
  A User with the appropriate Permission Group settings also has access to the following reports:
  - **Top Hosts** and **Top Protocols** reports on the **Enterprise Overview** page, if the user also has access to All Groups
  - **Interfaces** page reports for the interfaces that are accessible to the user
  - Existing reports on the **Custom Reporting**, **Flow Forensics**, **Analysis**, and **Site to Site** pages
  - Menus that an administrator has assigned to the User role

  The Role and Permission Group settings determine whether the User also can run existing reports, create reports, and manage reports. To create reports, a User must have access to All Groups.
- **None**
  Has no access to a data source. The user who has this product privilege cannot log in to the NFA console or drill down from a Performance Center view to the NFA console. By default, all users have this product privilege setting for all data sources.

  NOTE
  The same user account can have different privileges for different data sources.

Set Up Groups

We recommend that you create custom groups to help manage items in the Performance Center Console. Custom groups are required to let operators see performance data from the routers they manage.

Properly configured, groups can prevent operators from viewing particular types of data for security reasons. The administrator can selectively grant users access to data in their area of responsibility, such as a physical location or subnet.

Create Custom Groups

Before you start creating groups, plan a strategy and a structure. Consider the types of access permissions that operators require to perform their monitoring duties. If necessary, you can discuss your organizational and monitoring goals with a CA technical representative.

Create groups under the **All Groups** node in the **Groups** tree, or within an existing custom or site group. You cannot add groups to system groups, which appear "locked" in the **Groups** tree.

You can add a maximum of 2000 child groups to a parent group.

Follow these steps:

1. Log in to the Performance Center Console as a user with the required administrative role rights.
2. Navigate to the **Manage Groups** page.
   The page displays current groups in a tree structure.
3. Expand nodes in the **Groups** tree to find a location for the new group.
4. Right-click the node, and select **Add Group** (CA PC) or **Add New Group** (NPC).
   The **Add Group** window opens with the **New** tab selected by default.
5. Supply values for the following parameters:
   - **Group Name**
     Specifies a name for the group. Do not use the following special characters in group names: /&\,%.
   - **Description**
     (Optional) Helps you identify the group.
6. Confirm the setting for the following parameter:
   - **Include the children of managed items**
     Adds the children of managed items automatically when the items are added to this group. If this option is disabled and you add a router, the router interfaces are not included. As a result, the data from those interfaces is not visible in drilldown views.
     Default: **Selected** (CA PC) or **Not Selected** (NPC).

     NOTE
     Clear this option for a custom group that contains routers or the group will not be usable in in the NFA console.
7. Select **Custom** or **Site** from the **Group Type** list.
   If you selected **Site** as the type, specify values for the additional parameters that appear, including **Location**.
8. Click **Save**.
   The new group appears in the **Groups** tree.
   The group contains no items until you add them. You have two options for adding items to a custom group:
   - Manually populate the group by adding items in the **Manage Groups** interface.
   - Create rules to manage group membership.

Results of Unregistering

On rare occasions, you might need to unregister DX NetOps. For example, you would unregister a DX NetOps instance before registering it with a different Performance Center instance.

If you unregister DX NetOps, the following rules apply:

- Users: Users who are not associated with DX NetOps are deleted. Existing User IDs remain unchanged. You cannot add new users or edit user account settings while unregistered.
- Roles: Roles are not deleted. Users continue to have their previous roles. Existing Role IDs remain unchanged. You cannot change the roles or permissions for existing users while unregistered.
- Groups:
  - Groups that do not exist in DX NetOps are deleted. You cannot add or change groups while unregistered.
  - Nested groups that are associated with an interface are displayed as interface groups in the NFA console.
  - Groups that are not associated with an interface are displayed as permissions.
- Single Sign-On and LDAP: Single Sign-On and LDAP values remain unchanged.

NOTE

More information:

- P repare to Change the Performance Center Version

Enable Multi-Tenancy

To create separate monitoring environments that you administer from a single user interface, add custom tenants. A tenant represents a customer environment that a managed service provider (MSP) administers. Each tenant environment is independent, and can contain multiple users and roles that are not shared among tenants.

Prerequisites

- Install DX NetOps 9.5.0.
- Install CA Performance Center supported version. For supported version, see Compatibility Matrix.
- Add DX NetOps (CA NFA) to CA Performance Center (CA PC) as a data source.

Creating a Tenant in CA PC

Create a Tenant

Follow these steps:

1. Log in to CA PC as the administrator.
2. In the **Administration** menu, click **Tenants**.

   NOTE
   More information:

   Manage Tenants
3. On the **Manage Tenants** page, click **New**.
4. Give the tenant a **Name** that reflects the tenant usage (in the following examples, *tenant7*). Tabbing through the fields (up to the password fields) creates the default *tenant7_admin* and *tenant7_user*. Enter the credentials for the *tenant7* administrator and *tenant7* user.
5. Click **Save**.
6. On the **Manage Tenants** page, select the tenant you just created (*tenant7*) and click **Administer**.

This takes you to **Manage Users** for that tenant, indicated in the box in the upper-right corner of the screen that says "Administering Tenant: *tenant7*".

7. On the **Manage Users** page, select *tenant7_admin* and click **Edit**.
   a. In **1 Account Details**, verify that the **Role** is Administrator and change the password if desired. Click **Next**.
   b. In **2 Access Permissions**, click **Next**.
   c. In **3 Administer Group**, click **Next**.
   d. In **4 Product Privileges**, set the **Network Flow Analysis@...** privilege as **User**. Click **Save**.
8. On the **Manage Users** page, select *tenant7_user* and click **Edit**.
   a. In **1 Account Details**, verify that the **Role** is IT Operator and change the password if desired. Click **Next**.
   b. In **2 Access Permissions**, click **Next**.
   c. In **3 Administer Group**, click **Next**.
   d. In **4 Product Privileges**, set the **Network Flow Analysis@** privilege as **User**. Click **Save**.

Define Roles for the Tenant

Follow these steps:

1. Select **Administration, Roles**.
2. On the **Manage Roles** page, select **IT Operator** and click **Edit**.
   a. In the **Edit Role** page, select **Performance Center** and click **Edit**.
      a. Under **Available Rights**, select **Drill into Data Sources** and click the arrow to move it to **Selected Rights**.
      Note: Scroll to the bottom of the **Selected Rights** to see that it was added.
      b. Click **OK**.
   b. In the **Edit Role** page, select **Network Flow Analysis@...** and click **Edit**.
      a. Edit as necessary so that all rights except **Manage Reports** are under **Selected Rights**.
      b. Click **OK**.
   c. Click **Save**.
3. On the **Manage Roles** page, select **Administrator** and click **Edit**.
   a. In the **Edit Role** page, select **Performance Center** and click **Edit**.
      a. If **Selected Rights** does not include **Drill into Data Sources**, move it from **Available Rights**.
      b. Click **OK**.
   b. In the **Edit Role** page, select **Network Flow Analysis@...** and click **Edit**.
      a. Edit as necessary so that all rights are listed under **Selected Rights**.
      b. Click **OK**.
   c. Click **Save**.

Define the Domain For This Tenant

1. Select **Administration, IP Domains**.
2. In the **Manage IP Domains** page, click **New**.
3. Enter the necessary information for the domain for this tenant.
   It is helpful to create a domain name that references the name of the tenant; for example, *Tenant7-IPDomain*.
4. Click **Save**.

Sync CA PC with DX NetOps

1. Select **Administration, Data Sources**.
2. Select **Network Flow Analysis@...** and click **Resync**.
3. In the **Resync Data Source** dialog, click **Resync**.

Create Report Folders in DX NetOps

1. Log in as the tenant admin user (*tenant7_admin*).
2. Click the **Custom Reporting** tab.
3. Click **New** to create a new folder for tenant reports.
   It is helpful to create a name that references the name of the tenant; for example, *tenant7_CustomReportingFolder*.
4. Click **OK**.
5. Click the **Analysis** tab.
6. Click **New** to create a new folder for tenant reports.
   It is helpful to create a name that references the name of the tenant; for example, *tenant7_AnalysisReportFolder*.
7. Click **OK**.
8. Resync with CA PC.

Add Tenant Groups in CA PC

1. As the non-tenant admin user, select **Administration, Groups**.
2. Expand the nodes in the **Groups, Defined Tenants** tree to find the tenant of interest (*tenant7*).
3. Under the tenant, expand **Inventory** to find **Service Provider Defined Groups**.
4. Right-click and select **Add Group**.
5. Select the **Existing** tab.
6. Expand **Inventory**, **Data Sources**, **Network Flow Analysis@...**
   a. Under **Custom Report Groups**, select the tenant report folder (***tenant7_CustomReportingFolder*** **Folder**).
   b. Click **Select**.
   c. Under **Analysis Groups**, select the tenant report folder (***tenant7_AnalysisReportFolder*** **Folder**).
   d. Click **Select**.
7. Resync with DX NetOps.

Add Routers and Interfaces to the Tenant

1. As the non-tenant admin user in DX NetOps, select **Interfaces, Physical & Virtual** from the menu on the left side of the **Administration** page.
2. On the **Active Interfaces** page, select the checkbox next to the router and click **Edit**.
   a. Under **Domain**, select the domain created for this tenant (*tenant7 \ Tenant7-IPDomain*).
   b. Click **Save**.
3. Uncheck the router, if it is still checked.
4. Expand the router to show the list of interfaces.
5. Select the desired interfaces, or select the checkbox in the heading row to select all interfaces.
6. Click **Edit**.
   a. Under **Domain**, select the domain created for this tenant (*tenant7 \ Tenant7-IPDomain*).
   b. Click **Save**.
7. Confirm that the interfaces show the correct domain in the **Domain** column.

Configure the Product Using CA Network Flow Analysis

To configure the product without Performance Center, you would perform the following tasks:

- Set up the following elements in the NFA console:

  - SNMP profiles
  - User accounts, permissions, and roles
  - (*Recommended*) Groups
- Configure flow collection
  - Add Harvesters.
  - Enable the routers and interfaces to export flows.
- (*Optional*) Configure traps.
- (*Optional*) Verify that DX NetOps is receiving data.

NOTE

You cannot configure IP Domains in DX NetOps.

Additional Administrative Tasks

You may want to complete a number of additional administrative tasks, depending on your environment and on the number of users. You should complete some tasks as soon as DX NetOps is running. Consider the following tasks:

- **Verify that required settings are complete**
  Make sure that you have completed all the installation tasks. Certain settings are required to enable functionality, such as emailing reports and triggering SNMP traps.
- **Adjust settings to improve performance**
  Certain report settings can affect performance, such as the frequency of DNS host name resolution.
- **Check interface speeds**
  If interfaces are at or above 100 percent utilization in the NFA console (for example, as seen on the **Enterprise Overview** page), consider adjusting the interface speeds.
- **Control router display**
  You can exclude domains from the display to control which routers are displayed in the NFA console.
  Note: This action is applicable only in a deployment that includes multiple domains.
- **Disable monitoring of router-generated traffic**
  If you do not want reports to show broadcast traffic for each router, change the **Pump Broadcast/Multicast** application setting to **False**.
- **Monitor product components**
  The Watchdog services help you monitor components. Make sure that the appropriate settings are configured to notify you of component issues as soon as possible. Specify thresholds, an email address for receiving messages, and other settings.
- **Adjust security settings**
  To export reports to *comma-separated value* (CSV) files successfully, the security settings in your environment may make it necessary to add the IP address of the NFA console to your list of trusted sites.

Post-Installation or Upgrade Tasks

Complete the following post-installation tasks:

- Install Performance Center in your deployment.
- Configure SNMP on Linux Harvester servers.
- Exclude the following directories from real-time scans:
  - `C:\Windows\Temp`
  - *`install_path`* and all subdirectories

Real-time scans of these directories can corrupt the database.
- Do not implement drive space compression. Drive space compression can cause database losses and can degrade system performance.
- We recommend that you install Adobe Flash Player on systems with desktops that access the NFA console and install Adobe Reader on systems with desktops that access the PDF documentation.
- (*Optional*) Configure CA Anomaly Detector.

The tasks described here assume that the steps for pre-installation and installation or upgrade are complete.

Stand-Alone Server	Distributed NFA Console Server	Distributed Harvester Server (Windows)
Synchronize system time		
(*Recommended*) Update the list of trusted internet sites *		
(*Recommended*) Modify router ACLs **		(Recommended) Modify router ACLs **
(*Recommended*) Disable UAC		
(*Recommended*) Configure Web content expiration		
(*Recommended*) Create a TrapConfiguration key		
(*Optional*) Configure the Recycle Bin		
(*Optional*) Disable unneeded Windows services.		

* Complete this task for the systems that access the NFA console and the stand-alone server or NFA console server.
** In a distributed deployment, verify that the router access control lists (ACLs) are configured to enable the Harvesters to perform SNMP polling.

NOTE

More information:

- Get Adobe Flash Player
- Get Adobe Reader

Configure SNMP on Linux Servers

To configure a Linux server for a Harvester, complete the following tasks:

- Set up the SNMP configuration file.
- Configure SNMP to start automatically on boot.
- Start the snmpd service.

Follow these steps:

1. Log in as root and open a shell prompt.
2. (Highly Recommended) Use the following steps to set up the SNMP configuration file. This configuration file is needed for Watchdog SNMP polling.

   NOTE
   If you have a custom (non-default) snmp configuration file at
   `/etc/snmp/snmp.conf`
   , you might want to skip this step and update your existing configuration file instead. In this case, consult with an administrator to update the required settings to match the settings in the example configuration file. For example, make sure the
   `rocommunity`
   value is set as shown in the example configuration file.

If you use a custom community name as the `rocommunity` value, use the same community name throughout the DX NetOps deployment:
- The `snmpd.conf` file on each Linux Harvester server
- SNMP service on each Windows server
- The **Watchdog Settings** page of the NFA console

a. (Recommended) Back up the configuration file in `/etc` , for example by entering the following command:

```
cp /etc/snmp/snmpd.conf /etc/snmp/snmpd.conf.bak
```

b. Change to the `Netflow` directory:

```
cd install_path/Netflow
```

Where *install_path* is the target directory for installing the Harvester: `/opt/CA/NFA/` or a custom location

c. Copy the `snmpd.conf` file in the `Netflow` directory to the `/etc/snmp` directory, overwriting the existing file:

```
cp -i snmpd.conf /etc/snmp
```

d. Confirm the overwrite operation when prompted.

e. Verify that the configuration file is in place:

```
ls -l /etc/snmp/snmpd.conf
```

f. Verify that the configuration file has the correct permissions:

```
chmod 600 snmpd.conf
```

3. Configure SNMP to start automatically on every boot by entering the following command:

```
chkconfig snmpd on
```

4. Start the SNMP service in either of the following ways:
   - Enter the command:

```
service snmpd start
```

   - Navigate to **Services** in the user interface, select **snmpd**, **Start**, then click **Save**.
     The SNMP service starts with the community name that is defined in the `snmpd` file.

Synchronize System Time

Synchronize the system time among all servers that have DX NetOps components installed, unless the system time is synchronized automatically. We also recommend that you synchronize the system time for any Linux servers in your deployment, including the server that hosts CA Performance Center.

This topic describes an approach for synchronizing system time on Windows Server 2012 servers.

Follow these steps:

1. Log in as a user who has administrator privileges for DX NetOps.

2. Right-click the date or time on the right edge of the taskbar.
   The **Clock and Calendar** dialog opens.
3. Click **Change date and time settings**.
   The **Date and Time** dialog opens.
4. Select the **Internet Time** tab.
5. Click **Change settings**.
   The **Internet Time Settings** dialog opens.
6. Select the **Synchronize with an Internet time server** check box.
7. Select the server with which you want to synchronize. The default selection is `time.windows.com`.
8. Click **Update Now**.
   The system time is synchronized with the selected server.
9. Click **OK** in the **Internet Time Settings** dialog.
10. Click **OK** in the **Date and Time** dialog.

> **NOTE**
> If you have collection devices in different time zones, set each device to its local time zone. Times are converted to Greenwich Mean Time (GMT).

Update the List of Trusted Internet Sites

We recommend that you add the NFA console server to the list of trusted internet sites, unless your browser security settings allow unrestricted access to internet sites.

Note: The steps in this task are written for Internet Explorer 11.

Follow these steps:

1. Log in as a user who has administrator privileges for DX NetOps.
2. Launch Internet Explorer.
3. Click **Tools**, **Internet Options**.
   The **Internet Options** window opens.
4. Select the **Security** tab.
5. Click the **Trusted Sites** icon.
6. Click **Sites**.
   The **Trusted Sites** dialog opens.
7. Enter **http://<*NFAServerName-or-IP*>** in the **Add this Web site to the zone** field.
8. Click **Add**.
   Your change is saved and the site is added to the **Websites** list.
9. Exit:
   a. Click **Close**.
   b. Click **OK** in the **Internet Options** window.
      The **Internet Options** window closes.

Modify the Access Control Lists

We recommend that you configure the router access control lists (ACLs) to ensure that Harvesters (stand-alone, distributed, Windows and Linux) can perform SNMP polling.

> **NOTE**
> If you configure flow to be exported from loopback interfaces, verify that DX NetOps can access the IP addresses of those interfaces.

Disable User Account Control (UAC)

We recommend that you disable User Account Control (UAC) on any Windows stand-alone server or NFA console server. UAC is not fully supported for the current version of DX NetOps. Enabling UAC on the stand-alone server or NFA console server can result in unexpected behavior.

Follow these steps:

1. Log in as a user who has administrator privileges for DX NetOps.
2. Click **Start**, **Control Panel**, **User Accounts**.
   The **User Accounts** window opens.
3. Click **Change User Account Control settings**.
   The **User Account Control Settings** dialog opens.
4. Move the slider bar to the bottom **Never notify** level, if it is not already at this level.
   UAC is set to be disabled for all local accounts on the server.
5. Click **OK**.
   You return to the **User Accounts** tasks page.
6. Close the window.

Configure Web Content Expiration

We recommend that you configure IIS on a stand-alone or NFA console to ensure that fresh web content is displayed. With the **Expire Web Content Immediately** setting enabled, the browser displays an updated page from the server rather than cached content.

Follow these steps:

1. Log in as a user who has administrator privileges for DX NetOps.
2. Select **Start**, **Administrative Tools, Internet Information Services (IIS) Manager**.
   The **Internet Information Services Manager** window opens.
3. Display the options for expiring web content:
   a. Click the server name in the **Connections** pane.
      The server features are displayed.
   b. Double-click the **HTTP Response Headers** icon in the **HTTP Features** group.
      The window displays the current HTTP Response Headers.
   c. Click **Set Common Headers** in the **Actions** pane.
      The **Set Common Headers** dialog opens.
4. Select the following options:
   – **Expire Web content** check box
   – **Immediately**
5. Exit:
   a. Click **OK** to save your changes and close the dialog.
   b. Close the **Internet Information Services Manager** window.

Create a TrapConfiguration Key

Create an empty TrapConfiguration key in the Windows Registry on the Harvester to prevent the SNMP service from logging false positive events.

Follow these steps:

1. Log in as a user who has administrator privileges for DX NetOps.
2. Open a command prompt window.

3. Run the following command:

```
reg add HKLM\SYSTEM\CurrentControlSet\Services\SNMP\Parameters\TrapConfiguration
```

If the command executes successfully, the return value is: "The operation completed successfully."
The TrapConfiguration registry key is created in the following location:
`HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SNMP\Parameters`

Configure the Recycle Bin

We recommend that you configure the Recycle Bin to remove deleted files from the server immediately. The default behavior is for the system to save copies of deleted files in the Recycle Bin.

This should be done on any computer where DX NetOps is installed.

Follow these steps:

1. Log in as a user who has administrator privileges for DX NetOps.
2. Right-click the **Recycle Bin** icon on the desktop.
3. Select **Properties** from the menu.
   The **Recycle Bin Properties** dialog opens.
4. On the **General** tab, select the drive where DX NetOps is installed.
5. Select **Don't move files to the Recycle Bin. Remove files immediately when deleted**.
6. Click **Apply**.
7. Repeat these steps for each additional drive that you want to configure.
8. Click **OK**.

Disable Unneeded Windows Services

You have the option to disable services on the Windows servers in your deployment that are not needed by the product. This step is designed to help secure your servers. This step is not required. If the following services are needed for another reason, do not disable them.

Follow these steps:

1. Log in as a user who has administrator privileges for DX NetOps.
2. Open the **Services** window: Select **Start**, **Administrative Tools**, **Services**.
   The **Services** window opens.
3. Right-click the following services and select **Manual** or **Disabled**.
   Do not select **Stop** or the services will restart whenever the server is rebooted.

Windows Server Services That You Can Disable:

Application Layer Gateway Service	Application Management	Certificate Propagation
Distributed Link Tracking Client	Distributed Transaction Coordinator	DNS Client
Function Discovery Resource Publication	Human Interface Device Access	IP Helper
Link-Layer Topology Discovery Manager	Microsoft Iscsi Initiator Service	Multimedia Class Scheduler
Netlogon	Network List Service	Network Location Awareness
Portable Device Enumerator Service	Print Spooler	Remote Access Auto Connection Manager
Remote Access Connection Manager	Remote Registry	Resultant Set of Policy Provider
Secondary Logon	Smart Card	Smart Card Removal Policy

Special Administration Console Helper	SSDP Discovery	Tablet PC Input Service
Telephony	Volume Shadow Copy	Windows Audio
Windows Audio Endpoint Builder	Windows CardSpace	Windows Color System
WinHTTP Web Proxy Auto-Discovery Service	WMI Performance Adapter	

Configure MySQL User Password

From the CA NFA 10.0.1 release, you can manage the MySQL user password using the NFA MySQL User Password Change Utility. You must change the MySQL user password in CA NFA Harvester and in the CA NFA console. Ensure that the password for a user is same in both CA NFA Harvester and CA NFA Console.

Follow these steps:

1. In the CA NFA Harvester server and CA NFA Console, navigate to `<CA_NFA_HOME>\Tools` directory.
2. Stop the CA NFA services using the following commands:

   Windows:

   ```
   Harvester: harvester_services.bat stop
   Console: console_services.bat stop
   ```

   Linux:

   ```
   Harvester: harvester-services.sh stop
   ```

3. Open the `DBPasswordUtils.exe` file.
   A command prompt opens to let you change the password.
4. At the prompt, select the user for which you intend to change the password.

   ```
   Select the DB User to change the Password

   1. netqos

   2. harvest

   3. archive

   4. Exit

   Enter your choice (No.):
   ```

   The utility displays the conditions to set the password.
5. At the prompt, enter the required password:

   ```
   Enter the Password for <selected_user_name> DB User :
   ```

   The utility sets the desired password for the selected user.
6. (Optional) Repeat the steps to set password for more users.
7. At the prompt, select option to exit the command prompt.

8. Restart the CA NFA services using the following commands:

   Windows:

   ```
   Harvester: harvester_services.bat start
   Console: console_services.bat start
   ```

   Linux:

   ```
   Harvester: harvester-services.sh start
   ```

9. Restart the IIS server using the `iisreset` command.
   You have changed the MySQL password for the selected users.

   NOTE

   More Information: For more information about troubleshooting any connection issues with CA Anomaly Detector, see CA NFA and CA Anomaly Detector Connection Issue.

Uninstalling the Software

DX NetOps 9.3.0 and later includes an option to uninstall the product, which you can use to remove DX NetOps after an installation or upgrade.

Notes:

- The Uninstaller has no Undo option: Once you uninstall the software, you cannot restore the deleted files automatically.
- You should be able to install and uninstall the DX NetOps software once or twice without incident. If you have ongoing problems, contact CA Support instead of installing and uninstalling the software repeatedly.

  WARNING
  Do not use the Uninstall option if you have upgraded from CA NetQoS ReporterAnalyzer 9.0.1.

Prerequisites

Before you begin uninstalling the DX NetOps software from a server, verify that the component is working properly.

Verify that the appropriate databases are present, as listed in the following table.

Database	Location
reporter	*install_path*\MySql\data\reporter on the stand-alone or NFA console server
harvester	*install_path*\MySql\data\harvester on the stand-alone or Harvester servers
archive15	*install_path*\Netflow\datafiles\ReaperArchive15 on the stand-alone or Harvester servers
data_retention	*install_path*\MySql\data\data_retention on the stand-alone or Harvester servers
archive	*install_path*\Netflow\datafiles\ReaperArchive on the stand-alone or Harvester servers
nsas	*install_path*\MySql\datafiles\HarvesterArchive on the stand-alone or Harvester servers

Verify that the DX NetOps services and MySQL are running, as listed in the following table:

Service	Stand-Alone	Harvester	Console
CA NFA Collection and Poller Webservices (nfa_collpollws on Linux)	Yes	Yes	N/A
CA NFA Data Retention (nfa_dataretention on Linux)	Yes	Yes	N/A
CA NFA DNS/SNMP Proxies (nfa_proxies on Linux)	Yes	Yes	Yes
CA NFA File Server (nfa_filewebservice on Linux)	Yes	Yes	Yes
CA NFA Harvester (nfa_harvester on Linux)	Yes	Yes	N/A
CA NFA Poller (nfa_poller on Linux)	Yes	Yes	N/A
CA NFA Pump	N/A	Yes	Yes
CA NFA Reaper (nfa_reaper on Linux)	N/A	Yes	N/A
CA NFA RibSource	Yes	N/A	Yes
CA MySql (mysql on Linux)	Yes	Yes	Yes
NetQoS NQMySql (nfa_mysqlCSE on Linux)	Yes	Yes	Yes
NetQoSVnaDataCollector	N/A	Yes	N/A
NetQoS Reporter Manager	Yes	N/A	Yes
NetQoS Reporter/Analyzer General Services	Yes	N/A	Yes
NetQoS Reporter/Analyzer Pump	Yes	N/A	Yes
NetQoS Reporter/Analyzer Query Services	Yes	N/A	Yes
NetQoS Reporter/Analyzer Report	Yes	N/A	Yes
NetQoS Reporter/Analyzer Watchdog	Yes	N/A	Yes

Uninstall DX NetOps

This procedure describes how to uninstall the DX NetOps software by using the Uninstaller. You also can uninstall the software from the Windows **Add or Remove Programs** window, where it is listed under the publisher CA Technologies, Inc.

NOTE
These steps assume that you are uninstalling DX NetOps from a standalone or distributed deployment server that has no other related software installed.

Follow these steps:

1. Log in as a user who has administrator privileges for DX NetOps.
2. Back up your data and configuration files.
3. Exit from all applications - no exceptions.
4. Start the Uninstaller: In *install_path*\Uninstall
   – Stand-alone system: Double-click `Uninstall NFA Console` to uninstall the NFA console first, then double-click `Uninstall NFA Harvester` to uninstall the Harvester.
   If you attempt to uninstall the Harvester software first, an error message displays.
   – Distributed deployment: Double-click `Uninstall NFA Console` (NFA console server), then `Uninstall NFA Harvester` (Harvester server).
   The **Uninstall** window opens.

   NOTE
   Leave the file system undisturbed while uninstallation is in progress. Do not attempt to view the progress in Windows Explorer, for example.
5. Click **Uninstall**.
   The Uninstaller removes all of the program and data files, including the following DX NetOps and MySQL elements:
   – Data
   – Services
   – Registry entries
   – Shortcuts, links, and aliases created by the installer (user-created shortcuts or links cannot be removed by the uninstaller)
   – Most files
   – Some directories
   When the process is complete, the screen displays a list of the directories and files that were not deleted.
   Once the program finishes, the **Uninstall Complete** screen opens.
6. Click **Done** to close the **Uninstall Complete** screen.
7. Wait a few minutes to allow the helper process to finish the final cleanup.
   Some files are not deleted until this phase is finished. Once the final cleanup is finished, the Uninstaller itself is deleted.
8. Reboot the computer to complete the uninstall process.
9. Check the following to verify the uninstall:
   a. Verify that the Registry keys in the following location are deleted:
      `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\NetQoS`
   b. Verify that the DX NetOps services are removed.
   c. Verify that the DX NetOps programs (such as NFA or Harvester and MySQL) are no longer visible from the Control Panel. If they are, select each program individually and click **Uninstall**.

Notes:

- The uninstallation log is at the root level of the original installation path. For example, the Harvester uninstallation log is at:

  install_path\Harvester_Uninstall_<timestamp>.txt
- You may want to manually delete any DX NetOps directories and files that are still present.
- If you make an unsuccessful attempt to reinstall the software, contact CA Support.

Upgrading

This section is for systems administrators who are upgrading DX NetOps. When complete, you will have updated the following:

- NFA Console (Windows Server 2016, Standard Edition on a 64-bit processor)
- NFA Harvester (Windows Server 2016, Standard Edition on a 64-bit processor, or Linux server)
- (Optional) Flow Cloner (Windows Server 2016, Standard Edition on a 64-bit processor)

NOTE
More information:

(Optional) CA Anomaly Detector

Upgrade CA Network Flow Analysis

You can upgrade from the following version to the current version of DX NetOps:

- DX NetOps 9.3.3, 9.3.6, 9.3.8, 9.5.0, 10.0.0
- Compatibility Matrix
- (Optional) CA Anomaly Detector 9.3.3, 9.3.6, 9.5.0, 10.0.0

When you upgrade the software, you continue to use the same architecture:

- Stand-alone to stand-alone
- Distributed 2-tier to distributed 2-tier
- You cannot upgrade to current version of DX NetOps with a distributed 3-tier architecture.
  You can convert from a 3-tier architecture to a 2-tier architecture on DX NetOps 9.3.3, then upgrade to the current version of CA NFA.

DX NetOps 9.3.3 is the earliest software version that you can upgrade directly to the current version of CA NFA. If you have an earlier version, upgrade to 9.3.3 or 9.3.6 before you proceed.

NOTE

Before upgrading, Administrators should review all users that have the Manage Reports right, to determine what level of access they require after upgrade. Prior to DX NetOps 9.3.1, users with the Manage Reports right had visibility to all interfaces and reports in the system. Starting with DX NetOps 9.3.1, users with the Manage Reports right only have access to those reports, interfaces, and report folders where the Administrator has explicitly granted access.

Administrators can use the automatically generated groups that correspond to each report folder to control access to reports in that folder. Administrators should be aware that when they grant access to a specific report to a user with the Manage Reports right, that user will have access to all reports in the folder where that report resides.

Notes:

- Upgrade DX NetOps before any upgrade you make to Performance Center.
- Do not install or upgrade any DX NetOps component on a server that has Performance Center installed. You can co-locate the NFA console or stand-alone deployment with CA Anomaly Detector, but not with any other related software.
- If you plan to switch between CA NetQoS Performance Center (NPC) and CA Performance Center (CA PC), unregister before you upgrade DX NetOps. Unregister before a switch from CA NPC to CA PC or a switch from CA PC to CA NPC.
- Windows NT LAN Manager (NTLM) is not supported by the Single Sign-On tool.
- If you are using HTTPS, you need to perform the configuration steps found in the I nstallin g section.

NOTE

More information:

Upgrade from CA NFA 9.0.161

If you are on NFA 9.0.161, we recommend upgrading from 9.0.161 to 9.1.4, to 9.2.1 and then to 9.3.3 follow the steps below:

1. Before starting any upgrade, regardless of what version you are running, you MUST make backups of all of your Databases on the RA Console, Harvester, and DSA following the steps on pages 50-51 of the 9.1.3 upgrade Guide.
2. Prepare your servers by installing Java 1.6u41 on all RA servers, and then run the consoletool-exe.jar on on the RA Console server.
3. Upgrade from 9.0.161 to 9.1.4 following the NFA 9.1.3 Upgrade Guide.

   NOTE

   The CA NFA 9.1.4 was a minor release so there was 9.1.4 Upgrade Guide published.
4. Once the upgrade to NFA 9.1.4 completes, there is a period of time where your historical data will be migrated to the new database format. You need to wait for this historical data migration to complete on each of your DSA's before proceeding with the upgrade to 9.2.1To verify that your Historical Data migration completed, run the following command on your DSA's:

   ```
   mysql -P3308 -D nqrptr -t -e "select from_unixtime(value) from settings where
    name='migrationCompletedTime';"
   ```

   If this query returns a current time stamp, it means the historical migration completed and you can upgrade to NFA 9.2.1.
5. After you verify step 4 and verify that you can see historical data, cleanup any left over tables on the DSA by following the steps in this KB on how to Cleanup old NQRPTR files to prevent future upgrade problems.
6. Backup your 9.1.4 databases, follow the steps on pages 52-53 of the 9.1.3 Upgrade Guide.
7. Then you can upgrade from 9.1.4 to 9.2.1 following the NFA 9.2.1 Upgrade Guide.
8. If you have NPC installed on the same server as NFA, you will need to migrate NPC to a new server before upgrading to NFA 9.3.3 using this document:Tech Tip: NFA 9.3 Upgrade Tip - How to Migrate Netqos Performance Center(NPC) to a New Server
9. Once this completes backup All databases as documented in the NFA 9.3.3 Guide.
10. Upgrading to NFA 9.3.3 will require an upgrade of MySql to version 5.6. This may add additional time to your upgrade since all of your databases will be backed up while the upgrade runs. This will also require additional free space during the upgrade to ensure there is enough room to backup your databases as noted in this document: Tech Tip: NFA New Mysql version 5.6 pre-upgrade considerations
11. Then upgrade to NFA 9.3.3 following the 9.3.3 Upgrade Guide.

Converting From a Three-Tier to a Two-Tier Architecture

This release does not support a three-tier architecture. Before upgrading to 9.5.0, convert a DX NetOps 9.3.3 three-tier deployment to a two-tier deployment.

Prerequisites

DSA

On each DSA:

1. Download the DX NetOps three-tier to two-tier conversion tool and extract the contents of the zip file.
   250
   Copy the `3TConverter.ps1` script to a directory on the DSA.
2. Determine the Windows PowerShell version.
   a. Open the **Windows PowerShell** command window.
   b. Enter the following command to set the execution policy:
      ```
      Set-ExecutionPolicy RemoteSigned
      ```
      Enter **Y** at the prompt to change the execution policy.
   c. Enter the following command to determine the version:
      ```
      $PSVersionTable.PSVersion
      ```
      Windows Server 2008 R2 systems typically have PowerShell 2.0, while Windows Server 2012 R2 systems have PowerShell 3.0 or 4.0.
   d. Close the **Windows PowerShell** command window.
3. Download and install MySQL Connector/Net, `mysql-connector-net-<version>.msi` .
   PowerShell 2.0:
   https://dev.mysql.com/get/Downloads/Connector-Net/mysql-connector-net-6.7.9.msi
   PowerShell 3.0 or 4.0:
   https://dev.mysql.com/get/Downloads/Connector-Net/mysql-connector-net-6.9.9.msi
4. If there are Harvesters running on Linux in the three-tier architecture, download and install the most recent version of PuTTY 64-bit (version 0.69 at the time of writing). It is easiest to use the MSI Windows Installer version.
   http://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html Select to add PuTTY to the PATH, if necessary.

Windows Harvesters

On each Windows Harvester:

1. Share the *install_path*`\Netflow\datafiles\ReaperArchive15` directory with "Administrators" or users with Administrative rights.
   a. Navigate to `install_path\Netflow\datafiles` .
   b. Right-click the `ReaperArchive15` directory and select **Properties**.
   c. Select the **Sharing** tab and click **Share**.
   d. Select **Administrators** from the list and click **Share**.
2. Verify that there is enough disk space to copy the `ReaperArchive15` files from the DSA systems to the Harvester. The amount of required disk space could be as much as is currently configured on a single DSA.

Linux Harvesters

On each Linux Harvester:

1. Verify that all Linux Harvester hosts have the same user and password with read/write access. The conversion script uses a single user/password combinations to access all Linux Harvesters.
2. If the Harvester is not installed in the default installation directory (`/opt/CA/NFA`), create a symbolic link from `/opt/CA/NFA` to the installation directory.
   For example, if the Harvester is installed in `/home/NetQoS` :
   ```
   ln -s /home/NetQoS /opt/CA/NFA
   ```
3. Verify that there is enough disk space to copy the `ReaperArchive15` files from the DSA systems to the Harvester. The amount of required disk space could be as much as is currently configured on a single DSA.

Convert the DSAs

1. Stop all NFA services on the Console, each DSA, and each Harvester. *DO NOT* stop the MySQL service.
   Console:
   – CA NFA File Server
   – CA NFA RibSource
   – CA Performance Center SSO
   – NetQoS Reporter Manager Service
   – NetQoS Reporter/Analyzer General Services
   – NetQoS Reporter/Analyzer Pump Service
   – NetQoS Reporter/Analyzer Query Services
   – NetQoS Reporter/Analyzer Watchdog
   – NetQoS ReporterAnalyzer Report Service
   Harvester:
   – CA NFA Collection and Poller Webservices
   – CA NFA Data Retention
   – CA NFA DNS/SNMP Proxies
   – CA NFA File Server
   – CA NFA Harvester
   – CA NFA Poller
   – CA NFA Reaper
   DSA:
   – CA NFA Data Retention
   – CA NFA DSA Loader
   – CA NFA Pump
2. On each DSA, open a **Windows PowerShell** command window and browse to the directory containing the `3TConverter.ps1` script.
   Execute the script with the following arguments:
   ```
   ./3TConverter.ps1 -nfaconsole <nfa_console_ip> -installDir <install_path> [-linuxInstallDir
   <linux_install_path>] [-numDays <number_of_days_to_copy>]
   ```
   The optional `-numDays` parameter specifies the most recent number of days of 15-minute data to copy to the Harvesters. If this parameter is not set, all data is copied. We recommend setting the parameter to a low number so that the system can be brought back up in less time. After the system is back up, you can execute the script again to copy the remainder of the historical data while the system is receiving new flow data.
   a. If there are Linux Harvesters in the configuration, you are prompted for a user and password.
   b. Enter **y** to accept the **rsa** key when prompted.
   For example:
   ```
   .\3TConverter.ps1 -nfaconsole 10.2.0.15 -installDir C:\CA\NFA -linuxInstallDir /opt/CA/NFA -numDays 1
   ```
   Wait for the script to finish executing.
3. Restart all NFA services.
4. If the script was previously run with a `-numDays` parameter, execute the script again (on each DSA) without the `-numDays` parameter to convert the remaining 15-minute data.
5. After all 15-minute data are copied, stop sharing the *install_path*`\Netflow\datafiles\ReaperArchive15` directory on each Windows Harvester.
   Note: On Windows Server 2008 R2 systems, sharing leaves a lock on the `ReaperArchive15` directory. To remove the lock:
   a. Right-click `ReaperArchive15` and select **Properties**.
   b. Select the **Security** tab, then click **Advanced**.
   c. Select the **Administrators** group, then click **Change Permissions**.

d. Select **Include inheritable permissions from this object's parent**, then click **Apply**.
e. Exit from the **Properties** dialog.

Troubleshooting Conversion Issues

3TConverter.ps1 is Not Recognized

Error:

```
The term '.\3TConverter.ps1' is not recognized as the name of a cmdlet, function, script file, or operable
 program…
```

Solution:

`cd` to the directory that contains the `3TConverter.ps1` script and rerun the script.

All Directories Have Been Previously Copied

The `3TConverter.ps1` script creates and updates a file with the list of all DSA `ReaperArchive15` directories that have been successfully copied. This file is located in each DSA's installation directory, named `copiedArchive15Folders.txt` . If all files have already been copied, this message displays and the script exits.

If for some reason you need to recopy the files, delete the `copiedArchive15Folders.txt` file and delete the directories on the Harvester that need to be recopied, then rerun the script.

Execution of Scripts is Disabled

Error:

```
File 3Tconverter.ps1 cannot be loaded because the execution of scripts is disabled on
this system. Please see "get-help about_signing" for more details.
```

Solution:

1. Enter the **Windows PowerShell** command:
   `Set-ExecutionPolicy RemoteSigned`
2. Enter **Y** when prompted.

Specified Install Directory Does Not Exist

Solution:

Locate the NFA installation directory for the DSA and provide the full path to the directory in the `-installDir` argument. By default, the DSA is installed in `C:\CA\NFA` .

Unable to Query for Routers

Error:

```
Unable to query for routers: … Exception: Cannot find type [MySql.Data.MySqlClient.MySqlConnection]: verify
 that the assembly containing this type is loaded.
```

Solution:

Download and install MySQL Connector/Net.

Unable to Query for Routers

Error:

```
Unable to query for routers: … Exception: Exception calling "Open" with "0" argument(s): Unable to connect to
 any of the specified MySQL hosts.
```

Solution:

- Verify that the IP address specified for the `-nfaconsole` argument matches the IP address of the NFA Console for this configuration.
- Verify that the "CA MySql" service is running on the NFA Console.

Unable to Access Linux Harvester

Error:

```
Unable to access Linux Harvester: … Exception: The term 'pscp.exe' is not recognized as the name of a cmdlet,
 function, script file, or operable program…
```

or

```
Unable to access Linux Harvester: … Exception: The term 'plink.exe' is not recognized as the name of a cmdlet,
 function, script file, or operable program…
```

Solution:

- Download and install PuTTY.
- Ensure that PuTTY is added to the PATH.
- Close and reopen Windows PowerShell to refresh the environment.

Access Denied or Unable to Access Linux Harvester

Solution:

Verify that the provided user and password are correct and have read/write access on all Linux Harvester systems.

Unable to Copy Files to Windows Harvester

Solution:

Verify that that all Windows Harvesters have their `ReaperArchive15` directories shared with "Administrators".

Unable to Make ReaperArchive15/ Directory

Solution:

Verify that the provided Linux installation directory is correct and the same for all Linux Harvesters. Otherwise, verify that all necessary symbolic links have been created.

No Flow Data, Historical or New, on Windows Harvesters

Solution:

1. Verify that all NFA services are up on the Harvester and the Console.
2. Open a Remote Desktop session on the Windows Harvester.
3. Open the **File Explorer** and look for a "lock" icon on the `ReaperArchive15` directory. To remove the lock:
   a. Open the folder **Properties**.
   b. Select the **Security** tab, then click **Advanced**.

c. Select the **Administrators** group, then click **Change Permissions**.
d. Select **Include inheritable permissions from this object's parent** and click **Apply**.
e. Click **Ok** several times to exit from the **Properties** dialog.

4. Verify the `reporter` and `harvester` database settings.
   reporter Database Settings
   harvester Database Settings

No Flow Data, Historical or New, on Linux Harvesters

Solution:

1. Verify that all NFA services are running on the Harvester and the Console.
2. Verify that the owner and group of the `ReaperArchive15` directory and subdirectories are set to "nfa". If not, run the command:
   `chown -R nfa:nfa <linuxInstallDir>/Netflow/datafiles/ReaperArchive15/*`
3. Verify the `reporter` and `harvester` database settings.
   reporter Database Settings
   harvester Database Settings

reporter Database Settings (Console)

Table	Parameter	Value
agent_definitions	DataServerID	'0'
agents_aggregate	DataServerID	'0' or no data
agents_virtual	DataServerID	'0' or no data
parameter_descriptions	reporterArchitecture	'TwoTiers'
parameter_descriptions	dsaSettingsLastDeployed	'NULL'
parameter_descriptions	dsaSettingsLastModified	'NULL'
system_settings	reporterArchitecture	'TwoTiers'
system_settings	dsaSettingsLastDeployed	'0'
system_settings	dsaSettingsLastModified	'0'
database_servers	*No entries.*	

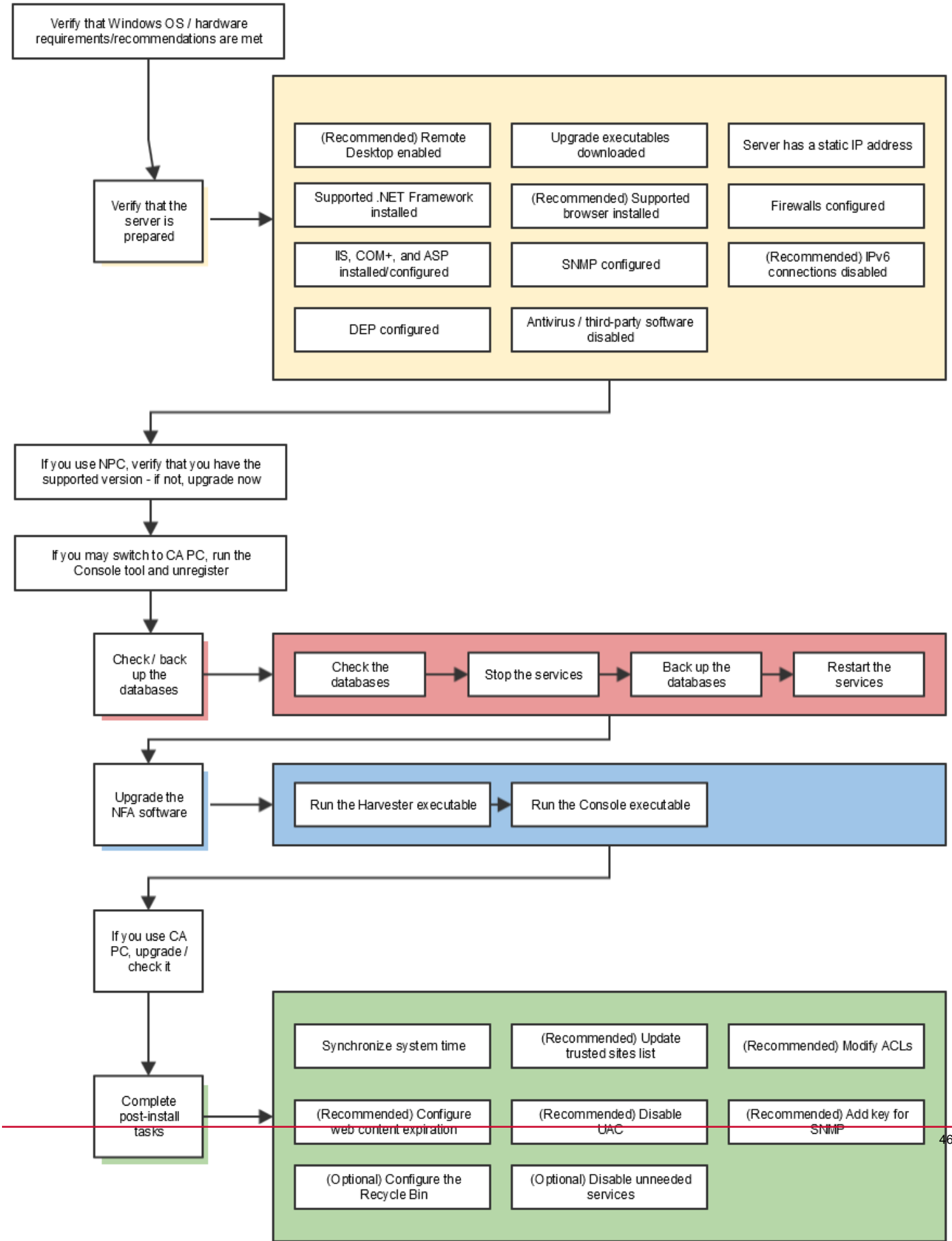
harvester Database Settings (Harvester)

Table	Parameter	Value
parameter_descriptions	enableArchive15	'Y'
parameter_descriptions	enableRollup	'Y'

Workflow for Upgrading a Stand-Alone Deployment

Use the following diagram as a general checklist for upgrading a stand-alone deployment.

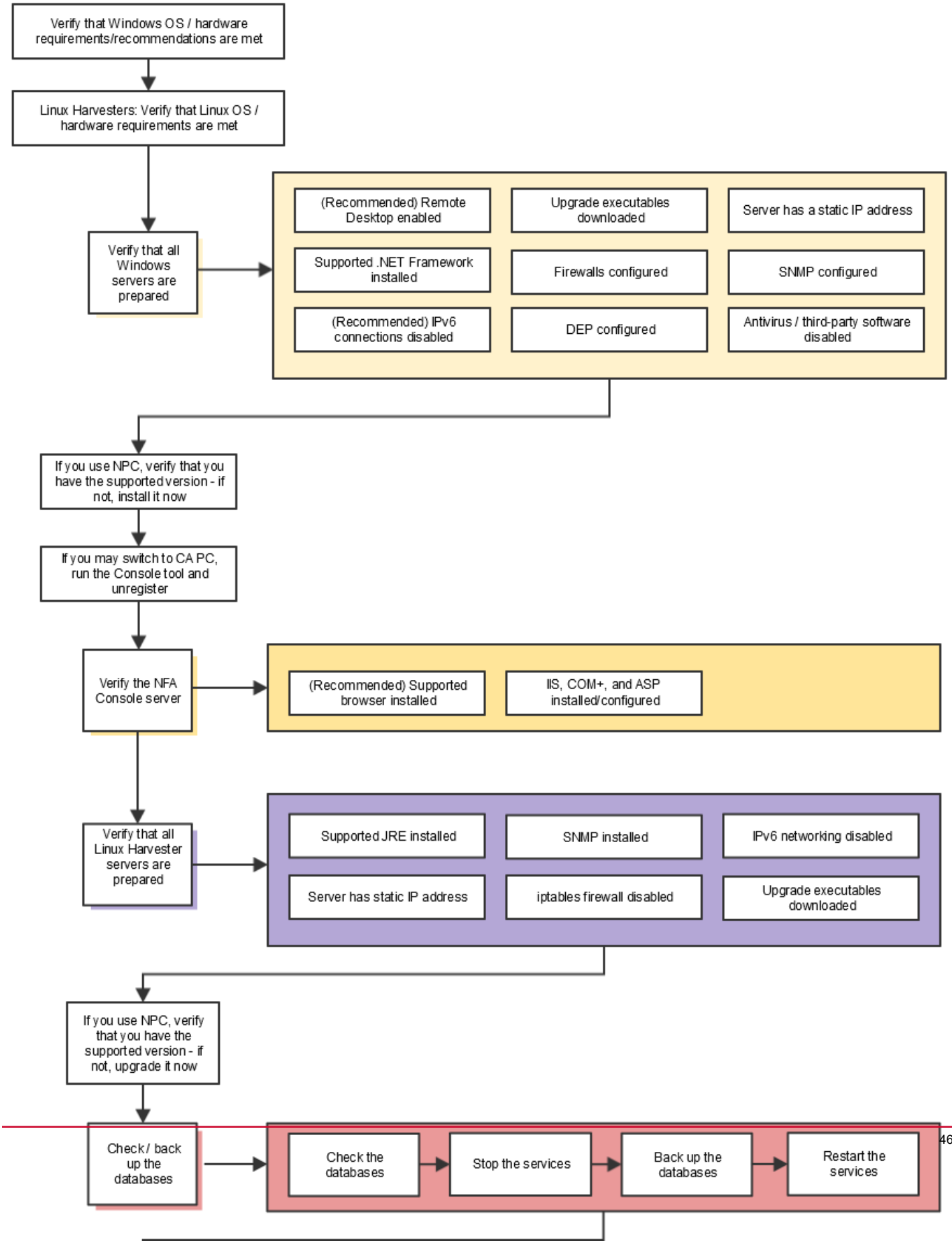
Figure 97: Upgrade stand-alone NFA_9.3.7



Workflow for Upgrading a Distributed Deployment

Use the following diagram as a general checklist for upgrading a distributed deployment.

Figure 98: Upgrade distributed NFA_9.3.7



Download the Upgrade Files

Copy the installation/upgrade files to the installation server so you are certain to have access to the files.

1. Get the files for installing or upgrading the components:
   a. Log in to support.ca.com.
   b. Navigate to the **Download Center**. For example, select **Download Center** from the **Support** menu in the left pane.
   c. Select the following navigation options:
      - **Select a Product**: Select "DX NetOps - MULTI-PLATFORM" to display the links for the NFA console, Harvester (Windows), Harvester (Linux), and CA Anomaly Detector installation and upgrade ISO files.
      - **Select a Release**: Select ***<version>***
   d. Download the ISO files from the **Product Components** list that is displayed.

      TIP
      An ISO file is an archive file that contains the contents of an optical disk. Each one of the available ISO files contains the files for installing or upgrading the component named in the file link.
2. Perform one of the following tasks:
   – Burn the ISO files to a CD-ROM or DVD.
   – Extract the contents of the ISO files by using an ISO image software application. Many free ISO image applications are available.
3. Extract the appropriate files to the installation servers:
   – Stand-alone servers:
      - `NFHarvesterSetup<version>.exe`
      - `RAConsoleSetup<version>.exe`
   – Windows Harvester servers in distributed deployments:
      - `NFHarvesterSetup<version>.exe`
   – Linux Harvester servers in distributed deployments:
      - `NFHarvesterSetup<version>.bin`
   – NFA console servers in distributed deployments:
      - `RAConsoleSetup<version>.exe`

You can install or upgrade the software locally or remotely.

Verify that the Windows Servers Are Prepared

Before you begin the upgrade, verify that the following conditions are met. Failure to comply with these requirements can result in data loss, increased down time, software conflicts, or a failed upgrade.

NOTE
More information:

- The system requirements are updated for DX NetOps 9.5.0. See System Recommendations and Requirements.
- You must have backups of your installation. See Backing Up and Restoring Data.

Verify that the following conditions are met:

- Installation servers have fully operational installations of DX NetOps software that is supported for upgrade.
- Your deployment includes a supported version of Performance Center, if you are using Performance Center.
- You have the required backups of your installation.
- The Windows servers meet the requirements in the following table.

Stand-Alone Server	Distributed NFA Console Server	Distributed Harvester Server
Windows operating system requirements are met		
(Recommended) Windows hardware recommendations are met		
(Recommended) Remote Desktop connection is enabled to allow remote access		
Upgrade executables are downloaded to the servers		
Static IP address is assigned to each server. Set the Harvester server IP address to match the flow export destination that is assigned to each router.		
Supported version of .NET Framework 3.5 is installed *		
Supported version of .NET Framework 4.6.2 is installed *		
(Recommended) Supported browser is installed **		
Firewalls are configured		
IIS, COM+, and ASP are installed **		
SNMP is configured **		
IPv6 addresses are disabled		
DEP is configured		
The following third-party software is disabled until the upgrade is complete: Antivirus, server monitoring, and maintenance software. If you enable antivirus scans later, exclude the DX NetOps installation path and its subdirectories.		
Databases are checked		
Services are stopped, databases and required files/directories are backed up, and services are restarted		

* The upgrade program does not open or does not complete successfully unless this requirement is met.
** If the server fails to pass this check, a warning message opens.

General Notes:

- Stop other programs from running during the installation or upgrade.
- When you apply Windows updates, restart all servers to ensure that the updates are applied.
- Ensure that no one else is logged in to the server during the installation or upgrade.

Upgrade Microsoft Windows Server 2008 to Microsoft Windows Server 2012 R2

This section describes the steps to migrate CA NFA with data to Microsoft Windows Server 2012 R2:

- Microsoft Windows Server 2012 R2 Standard Edition, when upgrading from a previous supported version of DX NetOps (CA NFA) which is installed on Microsoft Windows Server 2008 R2 Standard Edition.
- Microsoft Windows Server 2012 R2 Standard Edition, when performing a new installation of DX NetOps.

WARNING

- **Do not perform an in-place upgrade of the Windows operating system**. To upgrade a Microsoft Windows Server 2008 system which is hosting an existing version of DX NetOps, you must provision a new server with Windows Server 2012 R2 Standard Edition.
- Plan to migrate the existing IP addresses of the DX NetOps monitoring device to the new server.

In the following instructions,

- (WS2008) means to perform the task on the existing Windows Server 2008 CA NFA system.
- (WS2012) means to perform the task on the new Windows Server 2012 CA NFA system.

Two-Tier Deployment

Follow these steps:

1. Install Windows Server 2012 R2 Standard Edition on new servers.
2. (WS2012) Do a clean install of current version of the CA NFA on the new servers.
3. For each Windows Harvester:
   a. (WS2008) Stop the following services:
      - CA NFA Collection and Poller Webservices
      - CA NFA Data Retention
      - CA NFA DNS/SNMP Proxies
      - CA NFA File Server
      - CA NFA Harvester
      - CA NFA Poller
      - CA NFA Reaper
      - CA NFA Flow Cloner (if Flow Cloner is installed)
   b. (WS2008) Create the directory *install_path*\MySql_Backups .
   c. (WS2008) Execute the following commands from a Command Prompt window:

```
cd install_path\MySql\bin
mysqldump --routines --events -u root -P 3308 harvester
> install_path\MySql_Backups\harvester_backup.sql
mysqldump --routines --events -u root -P 3308 poller > install_path\MySql_Backups
\poller_backup.sql
mysqldump --routines --events -u root -P 3308 data_retention
> install_path\MySql_Backups\data_retention_backup.sql
```

   d. (WS2008 and WS2012) Copy the *install_path*\MySql_Backups directory to the Windows Server 2012 system.
   e. (WS2008 and WS2012) Copy the *install_path*\Netflow\datafiles directory to the Windows Server 2012 system.
   f. (WS2008) Shutdown the Windows Server 2008 system.
   g. (WS2012) Stop the following services:
      - CA NFA Collection and Poller Webservices
      - CA NFA Data Retention
      - CA NFA DNS/SNMP Proxies
      - CA NFA File Server
      - CA NFA Harvester
      - CA NFA Poller
      - CA NFA Reaper
   h. (WS2012) Execute the following commands from a Command Prompt window:

```
cd install_path\MySql\bin
mysql -P 3308 -u root -e "drop database harvester"
mysql -P 3308 -u root -e "create database harvester"
mysql -P 3308 -u root harvester < install_path\MySql_Backups\harvester_backup.sql
mysql -P 3308 -u root -e "drop database poller"
```

```
mysql -P 3308 -u root -e "create database poller"
mysql -P 3308 -u root poller < install_path\MySql_Backups\poller_backup.sql
mysql -P 3308 -u root -e "drop database data_retention"
mysql -P 3308 -u root -e "create database data_retention"
mysql -P 3308 -u root data_retention < install_path\MySql_Backups
\data_retention_backup.sql
```

i. (WS2012) Restart the NFA Services that you stopped in step g.
j. (WS2012) Reconfigure the Windows Server 2012 system to use the Windows Server 2008 IP address.

4. For the Windows Console:
   a. (WS2008) Stop the following services:
      - CA NFA RibSource
      - CA Performance Center SSO
      - NetQoS Reporter Manager Service
      - NetQoS Reporter/Analyzer General Services
      - NetQoS Reporter/Analyzer Pump Service
      - NetQoS Reporter/Analyzer Query Services
      - NetQoS Reporter/Analyzer Watchdog
      - NetQoS ReporterAnalyzer Report Service
   b. (WS2008) Create the directory *install_path*\MySql_Backups .
   c. (WS2008) Execute the following commands from a Command Prompt window:

```
cd install_path\MySql\bin
mysqldump --routines --events --skip-lock-tables -u root -P 3308 reporter
> install_path\MySql_Backups\reporter_backup.sql
```

   d. (WS2008 and WS2012) Copy the *install_path*\MySql_Backups directory to the Windows Server 2012 system.
   e. (WS2008) Shutdown the Windows Server 2008 system.
   f. (WS2012) Stop the following services:
      - CA NFA RibSource
      - CA Performance Center SSO
      - NetQoS Reporter Manager Service
      - NetQoS Reporter/Analyzer General Services
      - NetQoS Reporter/Analyzer Pump Service
      - NetQoS Reporter/Analyzer Query Services
      - NetQoS Reporter/Analyzer Watchdog
      - NetQoS ReporterAnalyzer Report Service
   g. (WS2012) Execute the following commands from a Command Prompt window:

```
cd install_path\MySql\bin
mysql -P 3308 -u root -e "drop database reporter"
mysql -P 3308 -u root -e "create database reporter"
mysql -P 3308 -u root reporter < install_path\MySql_Backups\reporter_backup.sql
```

   h. (WS2012) Restart the NFA services that you stopped in step f.
   i. (WS2012) Reconfigure the Windows Server 2012 system to use the Windows Server 2008 IP Address.
5. For the Anomaly Detector:
   a. (WS2008) Stop the following Anomaly Detector Services:

- CA NFA Host Resolver Service
- CA NFA Hunter Tracker Service
- CA NFA Ribsource

b. (WS2008) Create the directory *install_path*\MySql_Backups .
c. (WS2008) Execute the following commands from a Command Prompt window:

```
cd install_path\MySql\bin
mysqldump --routines --events -u root -P 3308 nsas > install_path\MySql_Backups
\nsas_backup.sql
```

d. (WS2008 and WS2012) Copy the *install_path*\MySql_Backups directory to the Windows Server 2012 system.
e. (WS2008) Shutdown the Windows Server 2008 system.
f. (WS2012) Stop the following Anomaly Detector Services:
   - CA NFA Host Resolver Service
   - CA NFA Hunter Tracker Service
   - CA NFA Ribsource
g. (WS2012) Execute the following commands from a Command Prompt window:

```
cd install_path\MySql\bin
mysql -P 3308 -u root -e "drop database nsas"
mysql -P 3308 -u root -e "create database nsas"
mysql -P 3308 -u root nsas < install_path\MySql_Backups\nsas_backup.sql
```

h. (WS2012) Restart the Anomaly Detector services that you stopped in step f.
i. (WS2012) Reconfigure the Windows Server 2012 system to use the Windows Server 2008 IP Address

6. (WS2012) Upgrade each CA NFA system to the current version.

Standalone Deployment

Follow these steps:

1. Install Windows Server 2012 R2 Standard Edition on a new server.
2. (WS2012) Do a clean install of current version of CA NFA on the new server.
3. (WS2008) Stop the following services:

- CA NFA Collection and Poller Webservices
- CA NFA Data Retention
- CA NFA DNS/SNMP Proxies
- CA NFA File Server
- CA NFA Harvester
- CA NFA Poller
- CA NFA Reaper
- CA NFA RibSource
- CA Performance Center SSO
- NetQoS Reporter Manager Service
- NetQoS Reporter/Analyzer General Services
- NetQoS Reporter/Analyzer Pump Service
- NetQoS Reporter/Analyzer Query Services
- NetQoS Reporter/Analyzer Watchdog
- NetQoS ReporterAnalyzer Report Service
- CA NFA Host Resolver Service (if Anomaly Detector is installed)
- CA NFA Hunter Tracker Service (if Anomaly Detector is installed)
- CA NFA Flow Cloner (if Flow Cloner is installed)

4. (WS2008) Create the directory *install_path*\MySql_Backups .
5. (WS2008) Execute the following commands from a Command Prompt window:

```
cd install_path\MySql\bin
mysqldump --routines --events -u root -P 3308 harvester > install_path\MySql_Backups
\harvester_backup.sql
mysqldump --routines --events -u root -P 3308 poller > install_path\MySql_Backups
\poller_backup.sql
mysqldump --routines --events -u root -P 3308 data_retention
> install_path\MySql_Backups\data_retention_backup.sql
mysqldump --routines --events --skip-lock-tables -u root -P 3308 reporter
> install_path\MySql_Backups\reporter_backup.sql
```

   If Anomaly Detector is installed:

```
mysqldump --routines --events -u root -P 3308 nsas > install_path\MySql_Backups\nsas_backup.sql
```

6. (WS2008 and WS2012) Copy the *install_path*\MySql_Backups directory to the Windows Server 2012 system.
7. (WS2008 and WS2012) Copy the *install_path*\Netflow\datafiles directory to the Windows Server 2012 system.
8. (WS2008) Shutdown the Windows Server 2008 system.
9. (WS2012) Stop the following services:

- CA NFA Collection and Poller Webservices
- CA NFA Data Retention
- CA NFA DNS/SNMP Proxies
- CA NFA File Server
- CA NFA Harvester
- CA NFA Poller
- CA NFA Reaper
- CA NFA RibSource
- CA Performance Center SSO
- NetQoS Reporter Manager Service
- NetQoS Reporter/Analyzer General Services
- NetQoS Reporter/Analyzer Pump Service
- NetQoS Reporter/Analyzer Query Services
- NetQoS Reporter/Analyzer Watchdog
- NetQoS ReporterAnalyzer Report Service
- CA NFA Host Resolver Service (if Anomaly Detector is installed)
- CA NFA Hunter Tracker Service (if Anomaly Detector is installed)

10. (WS2012) Execute the following commands from a Command Prompt window:

```
cd install_path\MySql\bin
mysql -P 3308 -u root -e "drop database harvester"
mysql -P 3308 -u root -e "create database harvester"
mysql -P 3308 -u root harvester < install_path\MySql_Backups\harvester_backup.sql
mysql -P 3308 -u root -e "drop database poller"
mysql -P 3308 -u root -e "create database poller"
mysql -P 3308 -u root poller < install_path\MySql_Backups\poller_backup.sql
mysql -P 3308 -u root -e "drop database data_retention"
mysql -P 3308 -u root -e "create database data_retention"
mysql -P 3308 -u root data_retention < install_path\MySql_Backups
\data_retention_backup.sql
mysql -P 3308 -u root -e "drop database reporter"
mysql -P 3308 -u root -e "create database reporter"
mysql -P 3308 -u root reporter < install_path\MySql_Backups\reporter_backup.sql
```

If Anomaly Detector is installed:

```
mysql -P 3308 -u root -e "drop database nsas"
mysql -P 3308 -u root -e "create database nsas"
mysql -P 3308 -u root nsas < install_path\MySql_Backups\nsas_backup.sql
```

11. (WS2012) Restart the NFA Services that you stopped in step 9.
12. (WS2012) Reconfigure the Windows Server 2012 system to use the Windows Server 2008 IP address.
13. (WS2012) Upgrade each CA NFA component to the current version.

Upgrade Microsoft Windows Server from 2012 R2 to Microsoft Windows Server 2016

This section describes the steps to migrate CA NFA with data to the Microsoft Windows Server 2016:

- Microsoft Windows Server 2016 Standard Edition, when upgrading from a previous supported version of DX NetOps (CA NFA) which is installed on Microsoft Windows Server 2012 R2 Standard Edition.
- Microsoft Windows Server 2016 Standard Edition, when performing a new installation of DX NetOps.

  WARNING

  - **Do not perform an in-place upgrade of the Windows operating system**. To upgrade a Windows Server 2012 system which is hosting an existing version of DX NetOps, you must provision a new server with Windows Server 2016 Standard Edition.
  - Plan to migrate the existing IP addresses of the DX NetOps monitoring device to the new server.

In the following instructions,

- (WS2012) means to perform the task on the existing Windows Server 2012 R2 CA NFA system.
- (WS2016) means to perform the task on the new Windows Server 2016 CA NFA system.

Two-Tier Deployment

Follow these steps:

1. Install Windows Server 2016 Standard Edition on new servers.
2. (WS2016) Do a clean install of current version of the DX NetOps on the new servers.
3. For each Windows Harvester:
   a. (WS2012) Stop the following services:
      - CA NFA Collection and Poller Webservices
      - CA NFA Data Retention
      - CA NFA DNS/SNMP Proxies
      - CA NFA File Server
      - CA NFA Harvester
      - CA NFA Poller
      - CA NFA Reaper
      - CA NFA Flow Cloner (if Flow Cloner is installed)

   b. (WS2012) Create the directory *install_path*\MySql_Backups .
   c. (WS2012) Execute the following commands from a Command Prompt window:

```
cd install_path\MySql\bin
mysqldump --routines --events -u root -P 3308 harvester
 > install_path\MySql_Backups\harvester_backup.sql
mysqldump --routines --events -u root -P 3308 data_retention
 > install_path\MySql_Backups\data_retention_backup.sql
mysqldump --routines --events -u root mysql proc > proc.sql
```

   d. (WS2012 and WS2016) Copy the *install_path*\MySql_Backups directory to the Windows Server 2016 system.
   e. (WS2012 and WS2016) Copy the *install_path*\Netflow\datafiles directory to the Windows Server 2016 system.
   f. (WS2012) Shutdown the Windows Server 2012 system.
   g. (WS2016) Stop the following services:

- CA NFA Collection and Poller Webservices
- CA NFA Data Retention
- CA NFA DNS/SNMP Proxies
- CA NFA File Server
- CA NFA Harvester
- CA NFA Poller
- CA NFA Reaper

h. (WS2016) Execute the following commands from a Command Prompt window:

```
cd install_path\MySql\bin
mysql -P 3308 -u root -e "drop database harvester"
mysql -P 3308 -u root -e "create database harvester"
mysql -P 3308 -u root harvester < install_path\MySql_Backups\harvester_backup.sql
mysql -P 3308 -u root -e "drop database data_retention"
mysql -P 3308 -u root -e "create database data_retention"
mysql -P 3308 -u root data_retention < install_path\MySql_Backups
\data_retention_backup.sql
```

i. (WS2016) Restart the NFA Services that you stopped in step g.

j. (WS2016) Reconfigure the Windows Server 2016 system to use the Windows Server 2012 IP address.

4. For the Windows Console:

a. (WS2012) Stop the following services:

- CA NFA RibSource
- CA Performance Center SSO
- NetQoS Reporter Manager Service
- NetQoS Reporter/Analyzer General Services
- NetQoS Reporter/Analyzer Pump Service
- NetQoS Reporter/Analyzer Query Services
- >NetQoS Reporter/Analyzer Watchdog
- NetQoS ReporterAnalyzer Report Service

b. (WS2012) Create the directory `install_path\MySql_Backups`.

c. (WS2012) Execute the following commands from a Command Prompt window:

```
cd install_path\MySql\bin
mysqldump --routines --events --skip-lock-tables -u root -P 3308 reporter
 > install_path\MySql_Backups\reporter_backup.sql
mysqldump --routines --events -u root mysql proc > proc.sql
```

d. (WS2012 and WS2016) Copy the `install_path\MySql_Backups` directory to the Windows Server 2016 system.

e. (WS2012) Shutdown the Windows Server 2012 system.

f. (WS2016) Stop the following services:

- CA NFA RibSource
- CA Performance Center SSO
- NetQoS Reporter Manager Service
- NetQoS Reporter/Analyzer General Services
- NetQoS Reporter/Analyzer Pump Service
- NetQoS Reporter/Analyzer Query Services
- NetQoS Reporter/Analyzer Watchdog
- NetQoS ReporterAnalyzer Report Service

g. (WS2016) Execute the following commands from a Command Prompt window:

```
cd install_path\MySql\bin
mysql -P 3308 -u root -e "drop database reporter"
mysql -P 3308 -u root -e "create database reporter"
mysql -P 3308 -u root reporter < install_path\MySql_Backups\reporter_backup.sql
mysql -u root mysql < proc.sql
```

h. (WS2016) Restart the NFA services that you stopped in step f.
i. (WS2016) Reconfigure the Windows Server 2016 system to use the Windows Server 2012 IP Address.

5. For the Anomaly Detector:
   a. (WS2012) Stop the following Anomaly Detector Services:
      - CA NFA Host Resolver Service
      - CA NFA Hunter Tracker Service
      - CA NFA Ribsource
   b. (WS2012) Create the directory *install_path*\MySql_Backups .
   c. (WS2012) Execute the following commands from a Command Prompt window:

```
cd install_path\MySql\bin
mysqldump --routines --events -u root -P 3308 nsas > install_path\MySql_Backups
\nsas_backup.sql
```

   d. (WS2012 and WS2016) Copy the *install_path*\MySql_Backups directory to the Windows Server 2016 system.
   e. (WS2012) Shutdown the Windows Server 2012 system.
   f. (WS2016) Stop the following Anomaly Detector Services:
      - CA NFA Host Resolver Service
      - CA NFA Hunter Tracker Service
      - CA NFA Ribsource
   g. (WS2016) Execute the following commands from a Command Prompt window:

```
cd install_path\MySql\bin
mysql -P 3308 -u root -e "drop database nsas"
mysql -P 3308 -u root -e "create database nsas"
mysql -P 3308 -u root nsas < install_path\MySql_Backups\nsas_backup.sql
```

   h. (WS2016) Restart the Anomaly Detector services that you stopped in step f.
   i. (WS2016) Reconfigure the Windows Server 2016 system to use the Windows Server 2012 IP Address
6. (WS2016) Upgrade each CA NFA system to current version.

Standalone Deployment

Follow these steps:

1. Install Windows Server 2016 Standard Edition on a new server.
2. (WS2016) Do a clean install of the current version of the DX NetOps on the new server.
3. (WS2012) Stop the following services:

- CA NFA Collection and Poller Webservices
- CA NFA Data Retention
- CA NFA DNS/SNMP Proxies
- CA NFA File Server
- CA NFA Harvester
- CA NFA Poller
- CA NFA Reaper
- CA NFA RibSource
- CA Performance Center SSO
- NetQoS Reporter Manager Service
- NetQoS Reporter/Analyzer General Services
- NetQoS Reporter/Analyzer Pump Service
- NetQoS Reporter/Analyzer Query Services
- NetQoS Reporter/Analyzer Watchdog
- NetQoS ReporterAnalyzer Report Service
- CA NFA Host Resolver Service (if Anomaly Detector is installed)
- CA NFA Hunter Tracker Service (if Anomaly Detector is installed)
- CA NFA Flow Cloner (if Flow Cloner is installed)

4. (WS2012) Create the directory *install_path*\MySql_Backups .
5. (WS2012) Execute the following commands from a Command Prompt window:

```
cd install_path\MySql\bin
mysqldump --routines --events -u root -P 3308 harvester > install_path\MySql_Backups
\harvester_backup.sql
mysqldump --routines --events -u root -P 3308 data_retention
 > install_path\MySql_Backups\data_retention_backup.sql
mysqldump --routines --events --skip-lock-tables -u root -P 3308 reporter
 > install_path\MySql_Backups\reporter_backup.sql
mysqldump --routines --events -u root mysql proc > proc.sql
```

   If Anomaly Detector is installed:

```
mysqldump --routines --events -u root -P 3308 nsas > install_path\MySql_Backups\nsas_backup.sql
```

6. (WS2012 and WS2016) Copy the *install_path*\MySql_Backups directory to the Windows Server 2016 system.
7. (WS2012 and WS2016) Copy the *install_path*\Netflow\datafiles directory to the Windows Server 2016 system.
8. (WS2012) Shutdown the Windows Server 2012 system.
9. (WS2016) Stop the following services:

- CA NFA Collection and Poller Webservices
- CA NFA Data Retention
- CA NFA DNS/SNMP Proxies
- CA NFA File Server
- CA NFA Harvester
- CA NFA Poller
- CA NFA Reaper
- CA NFA RibSource
- CA Performance Center SSO
- NetQoS Reporter Manager Service
- NetQoS Reporter/Analyzer General Services
- NetQoS Reporter/Analyzer Pump Service
- NetQoS Reporter/Analyzer Query Services
- NetQoS Reporter/Analyzer Watchdog
- NetQoS ReporterAnalyzer Report Service
- CA NFA Host Resolver Service (if Anomaly Detector is installed)
- CA NFA Hunter Tracker Service (if Anomaly Detector is installed)

10. (WS2016) Execute the following commands from a Command Prompt window:

```
cd install_path\MySql\bin
mysql -P 3308 -u root -e "drop database harvester"
mysql -P 3308 -u root -e "create database harvester"
mysql -P 3308 -u root harvester < install_path\MySql_Backups\harvester_backup.sql
mysql -P 3308 -u root -e "drop database data_retention"
mysql -P 3308 -u root -e "create database data_retention"
mysql -P 3308 -u root data_retention < install_path\MySql_Backups
\data_retention_backup.sql
mysql -P 3308 -u root -e "drop database reporter"
mysql -P 3308 -u root -e "create database reporter"
mysql -P 3308 -u root reporter < install_path\MySql_Backups\reporter_backup.sql
mysql -u root mysql < proc.sql
```

If Anomaly Detector is installed:

```
mysql -P 3308 -u root -e "drop database nsas"
mysql -P 3308 -u root -e "create database nsas"
mysql -P 3308 -u root nsas < install_path\MySql_Backups\nsas_backup.sql
```

11. (WS2016) Restart the NFA Services that you stopped in step 9.
12. (WS2016) Reconfigure the Windows Server 2016 system to use the Windows Server 2012 IP address.
13. (WS2016) Upgrade each CA NFA component to current release.

Verify that the Linux Servers Are Prepared

Before you begin the upgrade, verify that the following conditions are met. Failure to comply with these requirements can result in data loss, increased down time, software conflicts, or a failed upgrade.

- System Requirements: Verify that the upgrade servers meet the Linux requirements and recommendations.

NOTE
Harvester Linux servers must be upgraded to RHEL 6.8, 7.3, or 7.4 before the upgrade.

- Software Requirements: Verify that the upgrade servers have fully operational DX NetOps software that is supported for upgrade.
- Verify that each of the Harvester Linux servers is ready for the upgrade by:
  - Assigning a static IP address to each server. Set the Harvester server IP address to match the flow export destination that is assigned to each router.
  - Configuring SNMP
    If SNMP is not running, the upgrade program displays a warning. You can bypass the warning and configure SNMP after the upgrade, however.
  - Disabling the iptables firewall
  - Disabling IPv6 networking
  - Configuring DNS resolution. Polling fails if DNS resolution is not configured.
  - Stopping services
  - Backing up the databases
- To support non-Latin characters such as Japanese and Simplified Chinese, any command line clients that you use for installation must be configured for UTF-8 encoding. If UTF-8 encoding is not enabled, these characters may not display properly.
- The appropriate language packs are required for localized deployments.
- Regional Settings must use a period (.) to indicate a decimal value. If your deployment is localized to French, change the decimal symbol to a period in the **Region and Language**: **Customize Format** dialog.

NOTE
More information:

System Recommendations and Requirements

Check and Back Up the Databases

These steps should be followed to check and back up your databases before upgrading to DX NetOps 9.5.0:

Check the MySQL Databases

Checking the database tables corrects some problems and helps to avoid failures and recovery assistance with CA Support. Run the `mysqlcheck` command before the upgrade to verify that the database tables are set up properly.

You can run the `mysqlcheck` command to check the following databases:

- reporter: Located on each stand-alone or NFA console server (which typically contains some large tables)
- harvester: Located on each stand-alone or Harvester server
- data_retention: Located on each stand-alone or Harvester server
- nsas: Located on the CA Anomaly Detector server, which can be the stand-alone, NFA console, or separate server

Checking large database tables can be time-consuming. If you run the check on an entire database, each table in the database is locked in read-only state sequentially. The table that is being checked is unavailable for write operations.

You can run `mysqlcheck` without stopping MySQL. The MySQL daemon process (`mysqld`) can continue to run on Linux servers and the MySQL service can continue to run on Windows servers.

Follow these steps:

1. Log in to one of the DX NetOps servers as a user with administrator privileges. On a Linux Harvester server, log in as `root` .

2. Check the following databases:
   - Stand-alone server: harvester, reporter, and data_retention databases
   - Harvester server (distributed deployment): harvester, and data_retention databases
   - NFA console server (distributed deployment): reporter database
   - CA Anomaly Detector server: nsas database (if CA Anomaly Detector is installed on the stand-alone server or the NFA console server)
3. Enter one of the following `mysqlcheck` commands at a command or shell prompt:
   - To check the tables in all of the applicable databases on the server:

     ```
     mysqlcheck --all-databases
     ```

   - To check all of the tables in a single database:

     ```
     mysqlcheck --databases db_name
     ```

     where *db_name* = Name of the database that you want to check**Example**:

     ```
     mysqlcheck --databases reporter
     ```

   You do not need to specify the path to the database. The `mysqlcheck` command will find any or all databases that use the default port (port 3308). The custom storage engine does not support the use of the `mysqlcheck` command for its archive and archive15 databases. The command fails to run even if you specify the correct port (port 3307) for the connection to these databases.
   The command checks each table, attempts to repair any problems, then analyzes and optimizes the table. The return text lists the database tables that were checked and reports the status for each table.
   If the table passed the check, **OK** follows the table name. If a warning is returned and is followed by **OK**, the problem was resolved. If unresolved errors occur, contact CA Support.

Stop the Services

Before you back up the databases and upgrade the DX NetOps software, stop services to prevent new data from being sent to the NFA console until the upgrade is complete. Failure to stop the services does not cause the upgrade to fail, but some collected data is not processed.

Stop Services on Windows Servers

To prepare for backing up the databases, stop the services on all of the Windows servers in your DX NetOps deployment.

Follow these steps:

1. Log in as a user who has administrator privileges for DX NetOps.
2. Open the **Services** window: Click **Start, Control Panel, Administrative Tools, Services**.
3. Stop the Harvester service (NetQoS Harvester service or CA NFA Harvester) on each Harvester server.
4. Wait 15 minutes for data file processing to complete.
5. Stop the remaining DX NetOps services on each Windows server:

Service	Stand-Alone	Harvester	Console	Anomaly Detector
CA NFA Collection and Poller Webservices	Yes	Yes	N/A	N/A

CA NFA Data Retention	Yes	Yes	N/A	N/A
CA NFA DNS/SNMP Proxies	Yes	Yes	N/A	N/A
CA NFA File Server	Yes	Yes	N/A	N/A
CA NFA Harvester	Yes	Yes	N/A	N/A
CA NFA Host Resolver Service *	N/A	N/A	N/A	Yes *
CA NFA Hunter Tracker Service *	N/A	N/A	N/A	Yes *
CA NFA Poller	Yes	Yes	N/A	N/A
CA NFA Pump	N/A	Yes	Yes	N/A
CA NFA Reaper	Yes	Yes	N/A	N/A
CA NFA RibSource	Yes	N/A	Yes	N/A
NetQoS MySql	Yes	Yes	Yes	N/A
NetQoS NQMySql	Yes	Yes	N/A	N/A
NetQoS Reporter Manager Service	Yes	N/A	Yes	N/A
NetQoS Reporter/ Analyzer General Services	Yes	N/A	Yes	N/A
NetQoS Reporter/ Analyzer Pump Service	Yes	N/A	Yes	N/A
NetQoS Reporter/ Analyzer Query Services	Yes	N/A	Yes	N/A
NetQoS Reporter/ Analyzer Watchdog	Yes	N/A	Yes	N/A
NetQoS ReporterAnalyzer Report Service	Yes	N/A	Yes	N/A

* If CA Anomaly Detector is installed on the stand-alone DX NetOps server or NFA console server, stop these CA Anomaly Detector services.

The services and data collection stop. The data files are processed within 15 minutes.

6. Check the following directory on the NFA console server:
   install_path\Netflow\datafiles\HarvesterWork
   When the `HarvesterWork` folder is empty, you can back up the database.
   The services are restarted automatically during the upgrade process.

Stop Services on Linux Servers

To prepare for the database backups, stop the services on any Linux Harvester servers that are in your product deployment.

Follow these steps:

1. Log in as root or with a sudo user account.

2. Stop the nfa_harvester (CA NFA Harvester) service on each Linux Harvester server.
3. Wait 15 minutes for data file processing to complete.
4. Stop the following services on each Linux Harvester server:
   – mysql (NetQoS MySQL)
   – nfa_collpollws (CA NFA Collection and Poller Webservices)
   – nfa_dataretention (CA NFA Data Retention)
   – nfa_filewebservice (CA NFA File Server)
   – nfa_mysqlCSE (NetQoS NQMySql Custom Storage Engine)
   – nfa_poller (CA NFA Poller)
   – nfa_proxies (CA NFA DNS/SNMP Proxies)
   – nfa_reaper (CA NFA Reaper)

   After you stop the services, the Time BIN (`.tbn`) files are collected and processed within 15 minutes.
5. Check the following directory on the NFA console server: *`install_path`*`\Netflow\datafiles\HarvesterWork`
   When the `HarvesterWork` folder is empty, you can back up the database.
   The services are restarted automatically during the upgrade process.

Back Up the Databases and Restart the Services

Before you upgrade, back up the databases and files that are listed in the following table.

Important:

- Run backups concurrently. If you restore data from backups that have different timestamps, problems can result. Ensure that your backed-up data files are timestamped with the same hour.
- Store backups to a remote location to guard against the possibility of a hardware or operating system failure on the main server. For example, back up the databases to an administrative share or mapped network drive.

Database	Stand-Alone	Harvester (Distributed)	NFA Console (Distributed)	Anomaly Detector
reporter: Enterprise Overview data; NFA console configuration data	Important	N/A	Important	N/A
harvester: Harvester configuration data	Important	Important	N/A	N/A
nsas: Anomaly Detector configuration data	N/A	N/A	N/A	Important
archive15: Historical (15-minute) data	Recommended	Recommended	N/A	N/A
Customized Files: Configuration or other files that have been customized	Important	Important	Important	N/A
Customized data_retention: Settings to regulate data retention	Important if customized	Important if customized	N/A	N/A
HarvesterArchive: Flow Forensics data	Optional, rarely backed up	Optional, rarely backed up	N/A	N/A
archive: Realtime (1-minute) data	Optional, rarely backed up	Optional, rarely backed up	N/A	N/A

The following list describes the databases and their locations.

- reporter: Back up the previous 24 hours of Enterprise Overview data, NFA console configuration settings, and synchronization information.
- harvester: Back up the Harvester configuration data.
- nsas: If CA Anomaly Detector is installed on the same server as the NFA console or the stand-alone server, back up the configuration data for running CA Anomaly Detector. If CA Anomaly Detector is located on its own server, you can perform this backup when you upgrade CA Anomaly Detector.
  Path: *install_path*\MySql\data\nsas
- archive15: Back up the historical (15-minute) data that is stored for the reporting routers and interfaces. This backup is optional, but many administrators do back up the 15-minute data.
  Path: *install_path*\Netflow\datafiles\ReaperArchive15 directory
- Customized configuration files: Back up any customized configuration files--files that you customized or that were customized by CA Support. In addition, back up any customizations that you made to the website or reports.
  The DX NetOps configuration files typically have a `.config`, `conf.`, or `.ini` extension and are located in the product installation path. Other customizations may include `.css` files and report logos.
- archive: Back up the realtime (1-minute) data.
  Path: *install_path*\Netflow\datafiles\ReaperArchive directory
- Customized data_retention: If you customized any data retention settings, back up the data retention configuration data. It is unusual to customize data retention settings except with the assistance of CA Support. Changes to data retention settings can cause problems from rising demands on drive space.

Follow these steps:

1. Log in as a user who has administrator privileges for DX NetOps.
2. Connect to the server:
   a. Open a Remote Desktop session.
   b. Initiate a Terminal Services or VNC session to the installation server.
3. Back up each target directory or database, as described in:
   Backing Up and Restoring Data
4. Restart the services.

Upgrade a Stand-Alone Server

A *stand-alone deployment* consists of a single server that hosts both the Harvester and the NFA console. Complete the steps in this topic to upgrade the Harvester and NFA console on a single Windows server or virtual machine.

> **NOTE**
> The program checks for server problems at various points during the installation or upgrade. If a problem is found, an error message opens. A critical problem causes the program to exit. A warning message opens for non-critical problems, which you can correct at any time. The prerequisite checks look for general indicators that problems exist, but they do not warn you about all problems. You are responsible for preparing the server properly and for completing all required post-installation steps.

Follow these steps to complete the Harvester phase:

1. Verify that the server is upgrade-ready.
2. Log in to the server as a user who is a member of the Administrators group.
3. Stop the pump service on the NFA console server:
   a. Click **Start, Programs, Administrative Tools, Services**.
   b. Right-click the NetQoS Reporter/Analyzer Pump service.
   c. Select **Stop** in the right-click menu. The service stops.
4. Log in to the Harvester server as a user who is a member of the Administrators group.

5. Start the Harvester phase of the upgrade: Double-click the `NFHarvesterSetupx.x.x.exe` file. The language selection screen opens.
6. Verify that the appropriate language is selected, then click **OK**.
   The **Welcome** screen opens.
7. Click **Next**.
   The **CA NFA Harvester License Agreement** screen opens.
8. Review and accept the license agreement:
   a. Read the license agreement and scroll down.
   b. If you want to continue, click the option to accept the license agreement.
   c. Click **Next**.
      Prerequisite tests look for problems and may cause an error message to open.
9. If the **Prerequisite Check Warning** message opens, review it, correct or note any non-critical problems, then click **OK**.
   The **Upgrading Existing Installation** message opens and the **Choose Install Folder** screen opens. This screen displays the original root installation path as the default setting.
10. Verify that the specified installation directory is correct, then click **Next**.

    WARNING

    If you do not use the original installation path, the upgraded software will not run properly

    If the program does not find certain expected directories in the installation path, an error message opens and the upgrade stops. This problem does not occur when the previous software installation is fully functional.
    The **Pre-Installation Summary** screen opens.
11. Review the pre-installation information, then click **Install**.
    The **Installing Harvester** screen opens. When the Harvester upgrade is complete, the **Install Complete** screen opens and reports any errors that occurred.
12. (Optional) If errors occurred during the upgrade, see the following logs for details:
    – General installation log: *`install_path`*`\Harvester_Install_<`*`timestamp`*`>.log` (where <*timestamp*> is the time that the log was created)
    – Upgrade migration log: *`install_path`*`\migrator.log`
13. Click **Done** in the **Install Complete** screen.
    The Harvester upgrade program closes.

Follow these steps to complete the NFA console phase:

1. Start the NFA console upgrade software: Double-click the `RAConsoleSetupx.x.x.exe` file in Windows Explorer.
   The program starts and the language selection screen opens.
2. Verify that the appropriate language is selected, then click **OK**.
   The **Welcome** screen opens.
3. Click **Next**.
   The **NFA Console License Agreement** screen opens.
4. Review and accept the license agreements:
   a. Read the NFA console license agreement and scroll down.
   b. If you want to continue, click the option to accept the license agreement.
   c. Click **Next**.
      The **Third-Party License Agreement** screen opens.
   d. Read the third-party license agreement and scroll down.
   e. If you want to continue, click the option to accept the third-party license agreement.
   f. Click **Next**.
5. If the **Prerequisite Check Warning** message opens, review it, correct or note any non-critical problems, then click **OK**.

The **Singlebox Confirmation** message opens and asks you to confirm that you want a stand-alone deployment.

6. Review the information and click **OK**.
   The **Pre-Installation Summary** screen opens.
7. Review the pre-installation information, then click **Install**.
   The **Installing NFA** screen opens. When the NFA console upgrade is complete, the **Install Complete** screen opens.
8. Exit from the upgrade program:
   a. Select one of the restart options:
      - **Yes, restart my system**: Restart the system when you click **Done**.
      - **No, I will restart my system myself**: Defer the restart to be performed manually.
   b. Click **Done**.
      The upgrade program closes. If you selected the option to restart now, the system restarts and the upgrade is finalized.
9. (Optional) Check the revision history to verify that the software is upgraded to the correct version:
   a. Open a Command Prompt window.
   b. Start MySQL by entering the following command:
      ```
      mysql
      ```
   c. Display the revision history by entering the following commands:
      ```
      select * from harvester.revision_history;
      select * from reporter.revision_history;
      ```
10. If you did not choose to automatically restart the system, reboot now.
11. If you are using Performance Center, do a full synchronization.

Upgrade a Distributed Deployment

In a distributed deployment, DX NetOps components are distributed among multiple servers. The articles in this section describe how to upgrade each component server.

To upgrade a two-tier distributed deployment, complete the following procedures:

- Upgrade the Harvester on a Windows Server, or
  Upgrade the Harvester on a Linux Server
- Upgrade the Console

You cannot upgrade a three-tier distributed deployment. However, you can convert a three-tier DX NetOps 9.3.3 deployment to a two-tier architecture, then upgrade to current version.

The steps in these articles are written for the recommended upgrade order: Harvester upgrades, then the NFA console upgrade.

NOTE
The program checks for server problems at various points during the installation or upgrade. If a problem is found, an error message opens. A critical problem causes the program to exit. A warning message opens for non-critical problems, which you can correct at any time. The prerequisite checks look for general indicators that problems exist: They do not warn you about all problems. You are responsible for preparing the server properly and for completing all required post-installation steps.

NOTE

More information:

- Converting From a Three-Tier to a Two-Tier Architecture

Upgrade the Harvester on a Windows Server

In a distributed deployment, each Harvester is installed on a separate server. To upgrade a Harvester on a dedicated Windows server or virtual machine, complete the steps in this topic. These steps apply to a two-tier distributed deployment. You cannot upgrade a three-tier distributed deployment to DX NetOps 9.5.0 and later versions.

Follow these steps:

1. Verify that the server is upgrade-ready.
2. Log in to the NFA console server as a user who is a member of the Administrators group.
3. Stop the pump service on the NFA console server:
   a. Click **Start**, **Programs**, **Administrative Tools**, **Services**.
   b. Right-click the NetQoS Reporter/Analyzer Pump service and select **Stop**. The service stops.
4. Log in to the Harvester server as a user who is a member of the Administrators group.
5. Start the upgrade: Double-click the `NFHarvesterSetupx.x.x.exe` file in Windows Explorer on the Harvester server.
   The language screen opens.
6. Verify that the appropriate language is selected, then click **OK**.
   The **Prior Installation Detected** message opens.
7. Review the message and click **OK**.
   The **Welcome** screen opens.
8. Click **Next**.
   The **License Agreement** screen opens.
9. Review and accept the license agreement:
   a. Read the license agreement and scroll down.
   b. If you want to continue, click the option to accept the license agreement.
   c. Click **Next**.
      Prerequisite tests look for problems and may cause an error message to open.
10. If the **Prerequisite Check Warning** message opens, review it, correct or note any non-critical problems, then click **OK**.
    The **Choose Install Folder** screen opens and displays the original root installation path as the default setting.
11. Verify that the specified installation directory is correct, then click **Next**.

    NOTE
    If you do not use the original installation path, the upgraded software will not run properly.

    If the program does not find certain expected directories in the installation path, an error message opens and the upgrade stops. This problem does not occur when the previous software installation is fully functional.
    The **Pre-Installation Summary** screen opens.
12. Review the pre-installation information, then click **Install**.
    The **Installing Harvester** screen opens. When the upgrade is complete, the **Install Complete** screen opens and reports any errors that occurred.
13. (Optional) If errors occurred, see the following logs for details:
    – General installation log: *install_path*`\Harvester_Install_<`*timestamp*`>.log`
    – Upgrade migration log: *install_path*`\migrator.log`
14. Exit from the upgrade program:
    a. Select one of the restart options:
       - **Yes, restart my system**: Restart the system as soon as you click **Done**.
       - **No, I will restart my system myself**: Defer the restart to be performed manually.
    b. Click **Done**.
       The upgrade program closes. If you selected the option to restart now, the system restarts and the upgrade is finalized.

15. (Optional) Verify that the following conditions are met:
    - Harvester services are running.
    - Harvester is receiving data.
    - The revision history shows that the component is upgraded to the correct version. To display the revision history, complete the following:
      a. Start MySQL by entering the following command in a Command Prompt window:
         ```
         mysql
         ```
      b. Display the revision history by entering the following command:
         ```
         select * from harvester.revision_history;
         ```

Next:

- To upgrade another Harvester, repeat these steps on the additional Harvester server.
- To continue upgrading a two-tier deployment, upgrade the Console server.

Upgrade the Harvester on a Linux Server

A two-tier distributed deployment may include one or more Linux Harvester servers. To upgrade the Harvester software on a dedicated Linux server or virtual machine, complete the steps in this topic.

Follow these steps:

1. Verify that the server is upgrade-ready.
   See Linux Servers page for compatible RHEL versions.
2. Log in to the NFA console server as a user who is a member of the Administrators group.
3. Stop the pump service on the NFA console server:
   a. Click **Start**, **Programs**, **Administrative Tools**, **Services**.
   b. Right-click the NetQoS Reporter/Analyzer Pump service and select **Stop**. The service stops.
4. Log in to the Harvester server as `root` user.
   You can install the software locally or remotely; for example, by using `ssh` when you are logged in with root privileges. If you do not have root access, use an account with sudo privileges.
5. Open a command prompt window.
6. Run the following command to change the `ulimit` for the open files limit:
   ```
   ulimit -n ulimit_number
   ```
   Example:
   ```
   ulimit -n 65536
   ```
7. Prepare the installation/upgrade file for execution:
   a. Log in to the Harvester server as `root` .
      You can install or upgrade the software locally or remotely; for example, by using `ssh` when you are logged in with root privileges. If you do not have root access, use an account with sudo privileges.
   b. Execute the `chmod` command on the file in a terminal window:
      ```
      chmod u+x NFHarvesterSetupx.x.x.bin
      ```
      example: chmod u+x NFHarvesterSetup9.5.0.bin
   c. (Optional) Execute the list command to verify that the file is executable:
      ```
      ls -al
      ```
      The file permission settings are displayed.
8. Run the installation or upgrade software:
   ```
   ./NFHarvesterSetupx.x.x.bin
   ```
   example: ./NFHarvesterSetup9.5.0.bin

The language selection screen opens.

9. Verify that the appropriate language is selected, then click **OK**.
   The **Prior Installation Detected** message opens.
10. Review the message and click **OK**.
    The **Welcome** screen opens.
11. Click **Next**.
    The **License Agreement** screen opens.
12. Review and accept the license agreement:
    a. Read the license agreement and scroll down.
    b. If you want to continue, click the option to accept the license agreement.
    c. Click **Next**.
       Prerequisite tests look for problems and may cause an error message to open.
13. If the **Prerequisite Check Warning** message opens, review it, correct or note any non-critical problems, then click **OK**.
    The **Choose Install Folder** screen opens. This screen displays the original root installation path as the default setting.
14. Verify that the specified installation directory is correct, then click **Next**.

    NOTE
    If you do not use the original installation path, the upgraded software will not run properly.

    If the program does not find certain expected directories in the installation path, an error message opens and the upgrade stops. This problem does not occur when the previous software installation is fully functional.
    The **Pre-Installation Summary** screen opens.
15. Review the pre-installation information, then click **Install**.
    The **Installing Harvester** screen opens. When the upgrade is complete, the **Install Complete** screen opens and reports any errors that occurred.
16. (Optional) Review the errors by checking the installation log
    install_path/Harvester_Install_<timestamp>.log
17. Click **Done**.
    The upgrade program closes. The Harvester is upgraded and the DX NetOps services are started automatically.
18. (Optional) Verify that the following conditions are met:
    – (Two-tier architecture deployment) Harvester services are running.
    – Harvester is receiving data.
    – The revision history shows that the component is upgraded to the correct version. To display the revision history, complete the following:
      a. Start MySQL by entering the following command in a Command Prompt window:
         ```
         mysql
         ```
      b. Display the revision history by entering the following command:
         ```
         select * from harvester.revision_history;
         ```

Next:

- To upgrade another Harvester, repeat these steps on the additional Harvester server.
- To continue upgrading a two-tier deployment, upgrade the Console server.

Upgrade the NFA Console

Distributed deployments use separate servers for the NFA console and Harvesters. Complete the steps in this topic to upgrade the NFA console on a dedicated Windows server or virtual machine.

Follow these steps:

1. Verify that the server meets the following requirements:
   – The server is upgrade-ready.
   – The Harvester servers have been upgraded.
2. Log in to the NFA console server as a user who has administrator privileges for the system and for DX NetOps.
3. Start the upgrade: Double-click the `RAConsoleSetupx.x.x.exe` file in Windows Explorer on the NFA console server.
   The language selection screen opens.
4. Verify that the appropriate language is selected, then click **OK**.
   The **Welcome** screen opens.
5. Click **Next**.
   The **License Agreement** screen opens.
6. Review and accept the license agreements:
   a. Read the NFA console license agreement and scroll down.
   b. If you want to continue, click the option to accept the license agreement.
   c. Click **Next**.
      The **Third-Party License Agreement** screen opens.
   d. Read the third-party license agreement and scroll down.
   e. If you want to continue, click the option to accept the third-party license agreement.
   f. Click **Next**.
7. If the **Prerequisite Check Warning** message opens, review the test results:
   a. Correct the problems now or wait until the upgrade program finishes.
   b. Click **OK**. The **Upgrading Existing Installation** message opens.
8. Review the information:
   a. Verify that the existing and post-upgrade version information is correct, then click **OK**.
      The message reopens and reports the root installation path. The upgrade program uses the original path, which is `C:\CA\NFA` by default.
   b. Review the path information, then click **OK**. The **Choose Install Folder** screen opens.
9. (*Optional*) Click **Choose** to change the program installation location when prompted or enter a new path manually.
   The default location is `C:\CA\NFA` . Use the same installation path for the Harvester and for NFA console servers. We recommend that you install DX NetOps components on a non-system drive.
   The **Pre-Installation Summary** screen opens.
10. Review the information, then click **Install**.
    The **Installing NFA** screen opens. When the upgrade is complete, the **Install Complete** screen opens and reports any errors.
11. (*Optional*) See the installation log for error information:
    install_path`\NFA_Install_<`*timestamp*`>.log`
12. Exit from the upgrade program:
    a. Select one of the restart options:
       - **Yes, restart my system**: Restart the system when you click **Done**.
       - **No, I will restart my system myself**: Defer the restart to be performed manually.
    b. Click **Done**.
       The upgrade program closes. If you selected the option to restart now, the system restarts and the upgrade is finalized.
13. (*Optional*) Check the revision history to verify that the software is upgraded to the correct version:
    a. Open a **Command Prompt** window.
    b. Start MySQL by entering the following command:
       ```
       mysql
       ```

c. Display the revision history by entering the following command:

```
select * from reporter.revision_history;
```

14. If you did not choose to automatically restart the system, reboot now.
15. If you are using Performance Center, do a full synchronization.

Migrate NFA Harvester from RHEL 6.8 or 7.3 to RHEL 7.4

This section describes the steps to migrate CA NFA with data from RHEL 6.8 or 7.3 to RHEL 7.4.

- Upgrade your existing CA NFA installation on all servers (Harvester and Console) to current version of CA NFA.
- Do a clean install of current version of CA NFA Harvesters on the RHEL 7.4.
- The server must meet the minimum hardware and software requirements.

To migrate NFA Harvester data from RHEL 6.8 or 7.3 to RHEL 7.4, follow these steps:

1. Back up and restore the harvester data mentioned in Backing Up and Restoring Data.
2. From the Web browser pointed to the NFA console, select **Administration, System: Harvester.** Click **Edit** for this Harvester and change the IP address of the Harvester to reflect the new server's IP address. Do NOT delete the Harvester, as this will result in a deletion of all routers and interfaces associated with that Harvester.
3. Retarget the routers to send NetFlow to the latest version of CA NFA Harvester.

> **NOTE**
> Ensure that permissions are retained during the process of copying the files over. Manual update of folder and file permissions may be necessary.

Prepare to Change the Performance Center Version

If you are using Performance Center, the NFA console or stand-alone server must be registered as a data source for a supported version of CA Performance Center (CA PC) or CA NetQoS Performance Center (CA NPC). If you plan to to switch between CA NPC and CA PC, prepare for the switch as follows:

1. Prepare for unregistering from your current Performance Center version.
2. Unregister from Performance Center.
3. Perform the DX NetOps upgrade.

> **WARNING**
> We generally do not recommend unregistering. Many customizations are lost when you unregister and register again. Read this topic carefully to prepare for these events.

General Guidelines

- If you need to upgrade CA Performance Center, upgrade DX NetOps first.
- If you plan to switch between types of Performance Center, unregister before you upgrade DX NetOps. For example, unregister before you switch from CA NetQoS Performance Center to CA Performance Center or you switch from CA Performance Center to CA NetQoS Performance Center.

Results of Unregistering from CA Performance Center

If you unregister DX NetOps from CA NetQoS Performance Center and register again with CA Performance Center, the following rules apply.

- Domains:

  - The default domain is retained. Groups, users, devices (routers), interfaces, protocol names, and ToS labels that are assigned to the default domain retain their assignments.
  - Custom domains are deleted.
  - Groups, users, devices (routers), and interfaces in custom domains are reassigned to the default domain.
  - Protocol names, ToS labels, AS names, and IP addresses in custom domains become inaccessible.
- User accounts:
  - User accounts are retained if the users have valid product privilege settings (User, Power User, or Administrator) for DX NetOps. User accounts with no product privilege for DX NetOps are deleted.
  - User accounts that are associated with custom domains will be associated with the default domain instead.
  - If the user account `nqadmin` exists and has a product privilege that is lower than Administrator, this user account acquires the Administrator product privilege.

  You cannot add new users or edit user account settings after you unregister.
- Roles:
  - Custom and default roles are retained and continue to be associated with user accounts.
- Groups: Retained if the following conditions are met:
  - The group is a default group in CA NetQoS Performance Center or is a group that was pushed up to CA NetQoS Performance Center. 'Dynamic groups,' for example, cross-product groups, are deleted.
  - The group has contents; that is, the group is not empty.

  Custom and default groups require cleanup on the **Manage Groups** page in CA Performance Center after you unregister and register with the newer software. The following changes may occur:
  - Some group names change slightly. For example, the group 'All Interfaces' is renamed 'Interfaces.'
  - Structures may flatten so that a group that was nested under another group is not nested.
  - Groups may be relocated:
    A custom group that originally was shown on the **Manage Groups** page under **All Groups**/**System Groups**/**Data Sources**/**ReporterAnalyzer** is under **Network Flow Analysis** in the new group tree. The new location is **All Groups**/**Inventory**/**Data Sources**/**Network Flow Analysis**.
    If the custom group originally was not under **ReporterAnalyzer**, it is moved to sit under **Network Flow Analysis**.
  - Duplicate groups may be created.
  - Empty custom groups are deleted.
- SNMP profiles: SNMP profiles from CA NetQoS Performance Center 6.2 are retained with no changes in their status.
- Single Sign-On (SSO) customizations: LDAP and other SSO customizations are retained.
  If you change to a new SSO version, update the SSO configuration settings as described in Single Sign-on in the CA Performance Management documentation.

Follow these steps:

1. Log in to the console for CA NetQoS Performance Center as a user who is a member of the Administrators group.
2. Review your records of the customizations in CA NetQoS Performance Center, such as:
   - User accounts and their roles, product permissions, groups, and domain access
   - Custom roles and standard roles that have been customized, including any assignments for top-level menus, dashboards, and dashboard menus
   - Group structure and naming conventions
   - Custom domains and their contents, such as groups, devices, interfaces, SNMP profiles, report folders, AS names, protocol names, ToS labels, and IP addresses

   Remember that these elements may require checking, restructuring, or restoration.
3. (*Optional*) Prepare for the effects of unregistering:
   Consider giving a unique name to each group so it is easy to restore the group hierarchy. For example, name each group in a way that indicates its relationship to other groups.
   To ensure that no groups are deleted, flatten the group hierarchy before you unregister.
4. Open the **Data Source List** page by clicking **Admin**, **NetQoS Settings: Data Sources**.

5. Select the Reporter Analyzer or DX NetOps data source.
6. Click **Delete**.

Upgrade and Check Performance Center

If you are using Performance Center, make sure the NFA console server or stand-alone server is registered as a data source for:

- CA Performance Center version supported version, or
- CA NetQoS Performance Center 6.2

If you plan to upgrade Performance Center, do it now. For instructions, see the upgrading instructions for your Performance Center.

If DX NetOps is not registered as a data source, some of the function links on the **Administration** page are disabled. After you register, review the results in the Performance Center Console and the NFA console. Make any adjustments that are needed.

NOTE

More information:

- CA Performance Management

Using

DX NetOps is designed to give you the information you need for comprehensive visibility into your enterprise network.

The primary DX NetOps user interface is the NFA console, a web interface that you use to view the collected data, as well as perform administrative tasks.

NFA Console

To start using the NFA console, open a browser window and enter the server name or IP address of the server that hosts the NFA console:

```
http://<IP_Address>
```

You are prompted to log in when you first access the NFA console. You can get the DX NetOps server name or IP address and your login information from your Administrator.

If you are logging in for the first time, you can use the predefined administrator account:

- Login: **admin**
- Password: **admin**

When you log in to the NFA console, the **Interfaces** page opens. You can open the primary pages from the NFA console menu:

- **Enterprise Overview** Page: Quickly determine whether any of the interfaces in your network are nearing or have surpassed an acceptable utilization level.
- **Interfaces** Page: Select one or more interfaces and run reports.
- **Custom Reporting** Page: Define and run Custom Reports.
- **Flow Forensics** Page: Define and run reports on raw data.
- **Analysis** Page: Create proactive troubleshooting reports designed to compare collected network data to a threshold, identifying potential bottlenecks, anomalies, and viruses.
- **Site to Site** Page: Define and run Site to Site reports.
- **Administration** Page: Perform administrative tasks for DX NetOps .

As you use the navigation links to move between console pages, some product settings are saved and other settings are cleared as you leave the page. For example, when you navigate to **Enterprise Overview**, **Interfaces**, or **Flow Forensics** pages, the previous settings are preserved. As you navigate to **Custom Reporting**, **Analysis**, **Site to Site**, or **Administration**, your previous settings are cleared and the default values are restored.

Console Tips and Shortcuts

The NFA console simplifies viewing and using DX NetOps data. Some of the common NFA console features are:

Use Drilldown Links

When you view reports in the NFA console, many items are displayed in blue. Click the blue links for interfaces, hosts, conversations, and other items to drill down to additional information about the items.

For example, from the Interface report page you can open other detailed reports specific to the selected interface, host, conversation, or other traffic type.

Display Tooltips

You can display detailed information for some items in report views by positioning your cursor over the item. For example, place your cursor over a bar that represents an interface to open a Tooltip. The Tooltip shows additional details about the interface, such as its parent router, description, and number of bytes.

Sort Tables by Column Heading

Sortable column headings are provided for tables of interfaces, reports, or other items. Click a column heading to sort the data. Click a second time to switch the sort mode between descending and ascending order.

Jump Down

Some selection dialogs include a blue **Jump Down** arrow, which you can use to jump to the bottom of the page. For example, you can use the **Jump Down** arrow to locate the **Save** button quickly in the **Interface Group Selection** dialog.

Menu Arrow

Depending on your access settings, the built-in views on the NFA console pages may have a blue **Menu** arrow to the left of the view title. This opens a menu containing selections specific to the view.

Change the Interface for a Report

When you click an interface link on the **Interfaces** page, you drill down to an interface report. To view a report for a different interface, click the **[change]** link at the top of the page, then select another interface from the **Interface Index**. The report is updated to show data for the selected interface.

Search for a Router, Interface, or Interface Group

The **Interface Index** page includes a Search utility that you can use to filter the list and locate a router or interface. To perform a search, enter a text string in the text box and click **Search**.

You can include the wild card * as part of your search term. (The wild card * by itself is not a valid entry.) For example, to search for an IP address you can enter 10.0.7* to display only the addresses that begin with 10.0.7. In this example, the filtered list could include 10.0.7.1, but would not include 10.0.8.1.

The page displays a list of items that match the filter expression.

- Under the listed items, click the **Next** arrow or click a page number to display another page of list items.
- To display a different number of items per page, select a different value from the **Max per Page** list.

Save Report Data to CSV Files

You can save a report to a comma-separated value (CSV) file.

Follow these steps:

1. Display the data that interests you in one of the following locations: the **Enterprise Overview** page; an interface drilldown view; or a report you run on the **Custom Reporting**, **Flow Forensics**, **Analysis**, or **Site to Site** page.
2. Click the blue arrow next to a report view name.
3. Select the **Export to CSV** option from the menu that opens.
   The **File Download** dialog opens.
4. Click **Save**.
   The **Save As** dialog opens.
5. Specify a name and location for the `.csv` file, then click **Save**.

Open Online Help

Online help provides useful information that is easy to access when you are working in the NFA console. You can access the help system at any time by clicking the **Help** link that is near the top-right corner.

Email Reports

You can send a displayed report in an email immediately, or you can set up a schedule to generate an updated, complete report automatically as a PDF file.

The Email icon is included at the top of all report views:

- **Enterprise Overview** page views.
- Interface drilldown views.
- Reports that you run from the **Custom Reporting**, **Flow Forensics**, Analysis, and **Site to Site** pages.

If you are logged in with Administrator rights, you can email any of the reports by clicking the **Email** icon and you can schedule reports to be sent as PDFs by email.

> **NOTE**
> To email reports, an email server must be configured for DX NetOps. If no email server is configured, an error message is displayed when you attempt to use the email function. For information about setting up an email server, contact your Administrator.

Follow these steps:

1. Display a completed report.
2. Click the **Email** icon at the top-right corner of the report page.
   The **Email Information** dialog opens.
3. Enter the following information:
   - **Send To**
     Enter the email address to which you want to send the report page. Separate multiple email addresses with commas.
   - **Subject**
     Enter the subject line for the email.
   - **Message**
     Enter a message to explain the report or the purpose of the email.

   Scheduling Options: Select one of the following options:
   - **Send Now**
     Send the report by email immediately.
   - **Send on a Schedule**
     Schedule the report to generate and send on multiple days a week or to send once a week, month, quarter, or year. If you select **Send on a Schedule**, select one of the following options:
     - **Send Daily**: Select which days of the week to send the email.
     - **Send Weekly**: Select which day of the week to send the email.
     - **Send Monthly**: Sets the email to be sent on the last day of the month.
     - **Send Quarterly**: Select the month that designates the end of the first quarter to send the email. The email is sent on the last day of each reporting quarter.
     - **Send Yearly**: Select the last month of the year. The email is sent on the last day of the year.

     NOTE
     Scheduled emails generate a report PDF by using a stored URL address. The saved report definition is used to generate the scheduled report that is sent.
4. Click **OK**.
   - **Send Now**: The email is sent immediately with the current report page attached as a PDF file.
   - **Send on a Schedule**: The email schedule is configured. The report is generated and sent according to the schedule.
     If you have administrator privileges, you can view, edit, or delete the email schedules that you configure in the Administration pages of the NFA console.

Print Reports

You can print a report view from the browser window or save the report as a PDF file.

The **Print** icon is included at the top of all report views:

- **Enterprise Overview** page views
- Interface drilldown views
- Reports that you run from the **Custom Reporting**, **Flow Forensics**, **Analysis**, and **Site to Site** pages

If you are logged in with Administrator rights, you can print any of the reports by clicking the **Print** icon.

Follow these steps:

1. Select the completed report to display it.
2. Click the **Print** icon at the top-right corner of the report page.
   A printable version of the report opens in a new browser window.
3. In the browser toolbar, click the **Printer** icon.

Your browser displays a **Print** dialog box, which you use to select a printer and set other printing options.

4. Click **OK** to print the PDF file.

Refresh the View Data

Turn on **Refresh** mode for the report page to make it automatically update to reflect the most recently collected 15-minute data. When the **Refresh** icon is green and revolving, the page is in Refresh mode. To disable Refresh, click the icon again. Once you enable **Refresh**, data continues to be refreshed even when you navigate away from the page. If you leave the **Enterprise Overview** page, be sure to turn off **Refresh**.

NOTE
The Refresh function is available for real-time data you view in DX NetOps, in the **Enterprise Overview** page and in Interface drilldown reports.

Using Enterprise Overview

The views on the **Enterprise Overview** page give you an overview of network traffic across the enterprise. These articles describe the enterprise-level built-in reports.

Interface Utilization

The **Interface Utilization** view lists the interfaces throughout the enterprise that are the most heavily used. The view is a table summary of the interfaces whose utilization exceeds the user-configured thresholds. This view is located on the **Enterprise Overview** page.

Find Interface Utilization Information in the Performance Center Console

The Performance Center Console has a similar view, **Interfaces Over Threshold**. This view is located by default on the **Infrastructure Overview** dashboard in CA Performance Center and on the **Traffic Analysis** page in CA NetQoS Performance Center.

Interface Utilization Data

The **Interface Utilization** view displays the following information for each interface listed:

- **Status**
  Indicates utilization level: Green (Normal), Orange (Warning or Elevated), Red (Critical).
- **Interface**
  Identifies the interface by name.
- **Traffic Direction**
  Identifies whether the reported traffic is inbound or outbound.
- **Speed**
  Lists the speed that is set for the interface by the Administrator.
- **Average Utilization**
  Lists the percentage of total utilization of the interface on average.
- **Percent Time Utilization ≥ Warning Level**
  Identifies the percentage of the time that interface utilization meets or exceeds the Warning level. The default Warning level is 50 percent utilization for 25 percent of the reporting period. The **Interface Utilization** list contains only the interfaces that meet the Warning level. If an operator changes this setting, the contents of the interface list may change.
- **Percent Time Utilization ≥ Critical Level**

Identifies the percentage of the time that interface utilization meets or exceeds the Critical level. The default Critical level is 75 percent utilization of the interface capacity for 25 percent of the reporting period. The operator can change the Critical level.

- **Legend**
  Describes the criteria that determine which interfaces are displayed and what the status icons represent. The default settings for the icons are defined as follows:
  – Critical (red) status: The interface utilizes 75 percent or more of its bandwidth over 25 percent of the reporting period.
  – Warning/Elevated (orange) status: The interface utilizes 50 percent or more of its bandwidth over 25 percent of the reporting period.

Configuring the Interface Utilization Display

If your user account has the Administrator or Power User role, you can change the configuration of the data displayed in the **Interface Utilization** view of the **Enterprise Overview** page.

For example, you could display the orange status indicator next to interfaces that utilize 65 -74.9 percent of their bandwidth (instead of 50-74.9 percent) for at least 25 percent of the reporting period.

Follow these steps:

1. Click the menu arrow next to the **Interface Utilization** label and select **Configure**.
   The legend at the bottom of the **Interface Utilization** view changes to include editable fields.
2. Enter utilization threshold values for the following settings:
   – **Red Utilization %**
   Set the minimum threshold for assigning a Critical status to interfaces. Interfaces are flagged at the Critical level when their utilization approaches or exceeds this value.
   – **Orange Utilization %**
   Set the minimum threshold for assigning a warning status to interfaces. Interfaces are flagged at the Warning level when their utilization is at or above this value, but is below the minimum threshold for Critical status.
   – **For ... % of reporting period**
   Set the reporting period percentage to use for calculating the thresholds.
   For example, if the value is 25, the report includes only the interfaces that have a utilization level above the threshold for 25 percent of the reporting period. For the default reporting period of 24 hours, this list includes interfaces at or above the threshold value for six hours or more during that period.
   – **Show**
   (Optional) Change the number of interfaces that are displayed on each list page.
3. Click **Submit Changes** when the configuration changes are complete.
   The **Interface Utilization** table and legend are updated to reflect your modified threshold settings.

Top Interfaces

The **Top Interfaces** views display the interfaces that have the most inbound and outbound traffic in your network during the reporting period. These two views are located on the **Enterprise Overview** page.

Each bar is identified on the left by its parent router and interface name. By default, a red bar indicates that the interface exceeds a utilization threshold of 75 percent. An orange bar indicates that the interface utilization is less than 75 percent, but has a utilization of at least 50 percent. A green bar indicates that the interface utilization is less than 50 percent.

You can perform the following tasks in the **Top Interfaces - In** and **Top Interfaces - Out** views:

- Review the information in the graphical display.
- Display a Tooltip with additional details about the interface by holding your cursor over an interface bar or name.

Tooltips display the parent router name, interface name, interface description, flow volume, flow rate, interface utilization, and inbound speed.

- Drill down to display details on corresponding **Interface** page views by clicking a bar or name.
  For example, an interface that exceeds 75 percent utilization for outbound traffic could have degraded application performance. To investigate the issue, click the interface name in the graph. A more detailed report for the interface opens.
- Export the view data to a `.CSV` file.
- Change the utilization thresholds by editing the threshold values in the **Interface Utilization** view.

Find Similar Information in the Performance Center Console

The Performance Center Console has a similar view, **Top IP Interface Utilization (Flow)**. In CA Performance Center, this view is located on the **Infrastructure Overview** dashboard by default. In the CA NetQoS Performance Center Console, the view is located on the **Traffic Analysis** report page by default.

Top Protocols and Top Hosts

The bottom view of the **Enterprise Overview** page displays the protocols and hosts in your enterprise that are most heavily utilized. These views display the protocols and hosts that are transferring the most data in your enterprise and how many bytes of data that each one is transferring.

The **Top Protocols** and **Top Hosts** views provide quick visual indicators of the most utilized protocol and hosts in your network. You can use the links to drill down to more detailed information about a specific protocol or host.

For example, suppose a protocol has a high volume of traffic. To investigate whether the traffic is correctly routed over the network, you click the protocol name in the graph and drill down to details.

Display Tooltips in the **Top Protocols** and **Top Hosts** views by holding your cursor over a name or bar.

- Protocol: Display the protocol name (keyword), encapsulation (transport protocol and port number), and total traffic volume.
- Host: Display the host name, IP address, and volume of inbound, outbound, and total traffic.

Find Similar Views in the Performance Center Console

The Performance Center Console has similar views, the **Top Enterprise Protocols by Volume** and **Top Enterprise Hosts by Volume**. These views are on the **Infrastructure Overview** dashboard (CA PC) and the **Enterprise Dashboard** (NPC).

Drill Down to Protocol Details

In the **Top Protocols** report view on the **Enterprise Overview** page, click an individual protocol to see a more detailed summary report for the protocol, including its top interfaces and hosts. You can use this report to perform a high-level investigation of the related interfaces.

To open an **Interface Protocols** report and see detailed protocol data for the interface, click the interface name or bar. You can drill down to the **Interface Protocols** report from the view in the NFA console. Properly credentialed users also can drill down from the view in the Performance Center Console.

Drill Down to Details About a Host

In the **Top Hosts** view on the **Enterprise Overview** page, you can click a host name to see a more detailed summary report for the host, including its top interfaces and protocols. You can use this report to perform a high-level investigation of the protocols and interfaces for the host traffic.

In the **Top Interfaces for Host** report view, click the name of an interface to open an Interface Hosts report and see detailed protocol data specific to that interface.

To drill down to details, click the host name or bar. Properly credentialed users also can drill down from the view in the Performance Center Console.

Interface Reports

You can use **Interfaces** page reports to review traffic for specific interfaces. You can get an overview of network traffic from the **Enterprise Overview** reports, then get more detail from the **Interfaces** reports. The following topics describe how to view reports on the **Interfaces** pages:

Open Interface Reports

Interface reports show details about a specific interface. If you click an interface name on the **Enterprise Overview** page, for example, you drill down to details on the **Interface** report page. You can change the report presentation mode or open other types of interface reports.

The top right corner of the report page has following options:

- **Refresh**
  Click the **Refresh** icon to make the page update automatically. In Refresh mode, the page shows the most up-to-date data available. The **Refresh** icon is green and revolving as long as the page is in Refresh mode. Data continues to be refreshed even if you navigate away from the page.
  To turn off Refresh mode, click the icon again. Make sure to turn off the Refresh mode before you leave the page.

  NOTE
  The Refresh function is available for real-time data that you view on the **Enterprise Overview** page and in interface drilldown reports.
- **Email**
  Send the reports by email to one or more users (Administrator or Power User accounts only).
- **Print**Print the entire page of reports (Administrator or Power User accounts only).

The **Flow Forensics** links near the top and bottom of the report page open the page for configuring and running Flow Forensics reports. A Flow Forensics report lets you view detail for the raw data flows.

Use the Interface Index

When you log in to the NFA console, the **Interfaces** page opens and shows the **Interface Index**. You can use the **Interface Index** to select an interface and view reports for it. You can search for an interface on the **Router** or **Group** tab.

NOTE
The **Group** tab appears only if the product is registered as a data source for Performance Center. If the product is not registered, the **Interface** tab appears in place of the **Group** tab.

Open an Interface Index Router Tab

You can use the **Router** tab in the **Interface Index** to locate an interface by router. The **Interface Index** shows the available routers in an alphanumerically sorted list. By default, 20 routers are shown on each page of the list.

Follow these steps:

1. Select **Interfaces** in the NFA console menu.
   The **Interfaces** page opens and shows the **Interface Index**.
2. Make sure the **Router** tab is displayed.
3. Locate the interface that interests you. Use any of the following options to locate the interface:

- Click the name of the parent router to expand a list of its interfaces.
- **Search**: Search for whole or partial text strings. Searching filters the list to show only the entries that have an **Interface** or **Description** column value that matches your search term.
- **Max per Page**: Change the **Max per Page** setting to show more routers and interfaces on each page.

4. Click the interface that interests you.
   An **Interface Overview** report page opens, which displays data for the selected interface.
5. (Optional) Change the format or type of data displayed in the report:
   a. Click the gray bar on the left.
      The **Presentation** menu opens.
   b. Select the desired presentation (display and metric) options.
   c. (Optional) Click the bar again to hide the menu.

Open an Interface Index Group Tab

You can locate and select an interface by using the **Group** tab in the **Interface Index**. The **Group** tab shows the groups organized in a tree. The tree can help you locate a specific interface, especially when in an enterprise that has large numbers of routers and interfaces.

The **Group** tab appears only if the product is registered as a data source for Performance Center. If the product is not registered, the **Interface** tab appears in place of the **Group** tab.

Follow these steps:

1. Select **Interfaces** in the NFA console menu, if the **Interface Index** page is not already visible.
   The **Interfaces** page opens and shows the **Interface Index**. This is the first page that you see when you log in to the NFA console.
2. Click the **Group** tab.
   The group tree is displayed in the left pane.
3. Locate the interface that interests you. Use any of the following options to locate the interface:
   - Click the group that interests you.
     To expand the contents of a high-level group, click its arrow icon. For example, you could click the **Inventory** group or expand the Inventory group and select a domain sub-group.
     If the selected group or its sub-groups contain interfaces, the list of the interfaces opens in the right pane.
   - Sort by Column Heading: Click a column heading to re-sort the list. The entire list is sorted, not just the items on the current page.
   - Filter by: Click a **Filter by** option to display active, inactive, or all interfaces.
   - Search: Search for whole or partial text strings. Searching filters the list to show only the entries that have an **Interface** or **Description** column value that matches your search term.
   - Max per Page: Change the **Max per Page** setting to show more routers and interfaces on each page.
4. Click the interface that interests you.
   An **Interface Overview** report page opens, which displays data for the selected interface.
5. (Optional) Change the format or type of data displayed in the report:
   a. Click the gray bar on the left.
      The **Presentation** menu opens.
   b. Select the desired presentation (display and metric) options.
   c. (Optional) Click the bar again to hide the menu.

Interface Tab

The **Interface** tab displays a list of the available interfaces in an alphanumerically sorted list. By default, 10 interfaces are shown on each page. To re-sort the table data, click a column heading.

Open an Interface Report from Other Pages

In addition to opening an interface report from the **Interface Index**, you can drill down to detailed reports by clicking links in other reports. For example, you can drill down to details by clicking an interface name in an **Enterprise Overview** page report.

You can also drill down to interface reports from some Performance Center Console views. This option is available for views of DX NetOps data, provided that your administrator has enabled the drill down function.

Interface Report Types

If you select an interface in the **Interface Index** or you click a drilldown link, a report opens in the **Interface** pages. The initial report is an **Overview** report. To display a different report type, select an option from the **For this interface, show me:** list in the upper left corner.

Use the icons at the top to refresh, print, or email the report page.

> **NOTE**
> If the product is registered with CA Performance Center you can click the blue menu arrow next to the report title and select **CA PC Interface Performance**. The CA Performance Center Console opens to show **Interface Pages: Details** page for the Interface.

Interface Overview Report

An interface **Overview** report is a broad summary of information about the selected interface.

Follow these steps:

1. Display an interface report in either of the following ways:
   - Locate and click an interface on the **Interfaces** page.
   - Click an interface link in an existing view; for example, on the **Enterprise Overview** page.
2. Make sure that **Overview** is selected as the report type at the top of the page.
   The report page shows the following interface data:
   - Protocol data that is inbound and outbound on the interface
   - ToS data that is inbound and outbound on the interface
   - Data that travels to and from hosts
   - Total volume of conversation data

   If you want to display another report type, select an option from the report menu at the top-left corner of the page.

   > **NOTE**
   > The graph in the **Timeframe** panel only displays **Protocol Trend** data for all traffic on the interface that the user has selected. Selecting a different report type does not change the data displayed in the **Timeframe** panel.
3. (*Optional*) Change the type of data presentation and measurement by using the presentation options.
   - **Mixed Chart** of **Rate**, **Volume**, or **Utilization** data:
     - Stacked trend charts of incoming and outgoing protocol and ToS data
     - Pie charts that show the volume of data to and from hosts
     - Pie chart that shows the total volume of conversation data
   - **Mixed Trend** of **Rate**, **Volume**, or **Utilization** data:
     - Stacked trend charts of incoming and outgoing protocol and ToS data
     - Trend summary charts that show the volume of data to and from hosts
     - Trend summary chart that shows the total volume of conversation data

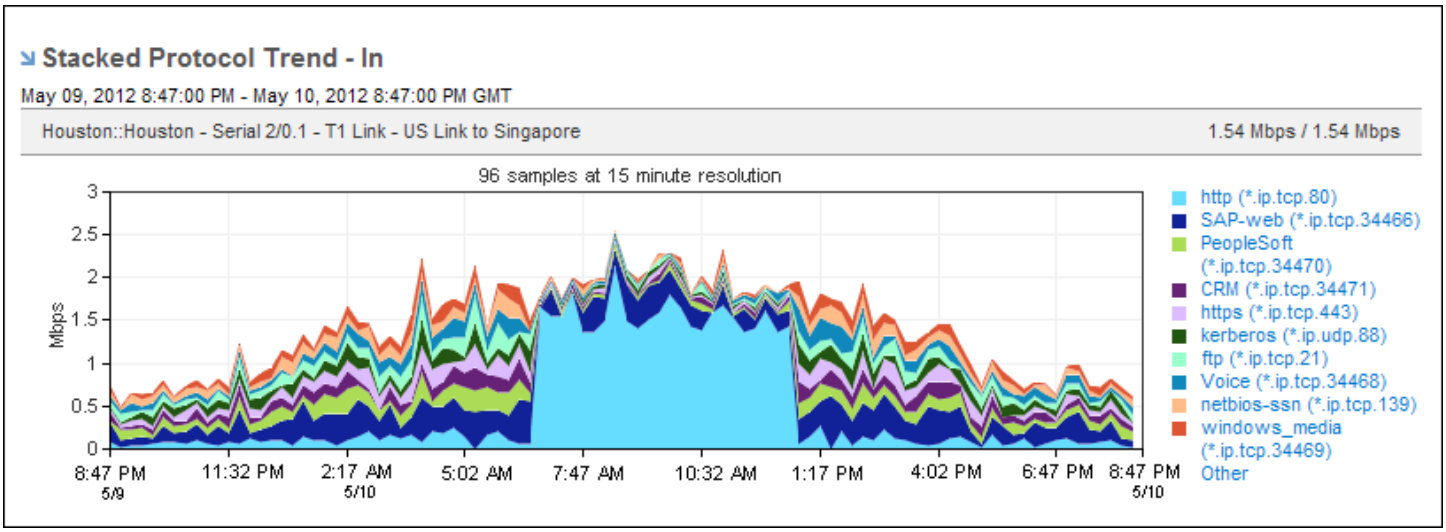
Use the **Show Top** setting to specify the maximum number of conversations to include. (Default setting)

- **Pie Chart**: Shows pie charts of the following data:
  - Incoming and outgoing protocol and ToS data
  - Volume of data to and from hosts
  - Total volume of conversation data

4. (*Optional*) Change the reporting period: Open the **Timeframe** dialog by clicking the timeframe link.
   The reporting period is the most recent 24-hour period by default.

Top N Protocols Report

A Top N Protocols report provides information about the protocols that generate the most traffic on a specific interface. This topic describes how to display a Top N Protocols report.



Follow these steps:

1. Display an interface report in either of the following ways:
   - Locate and click an interface on the **Interfaces** page.
   - Click an interface link in an existing view, for example, on the **Enterprise Overview** page.
2. Select **Protocols** from the report type menu at the top of the page.
3. Make sure that the report scope is set to **Top N Protocols**. The **Top N Protocols** link should appear next to the report type at the top.
   The report page is updated to show protocol data in stacked trend charts by default. The report includes views for inbound, outbound, and total protocol data.
4. (Optional) Change the type of data presentation and measurement by using the **Presentation** options.
   - **Stacked Trend Chart** of **Rate**, **Volume**, or **Utilization** data
   - **Trend Chart** of **Rate**, **Volume**, or **Utilization** data:
     Use the **Show Top** setting to specify the maximum number of conversations to include. (Default setting)
   - **Pie Chart**
   - **Summary Table** of **Rate**, **Volume**, or **Utilization** data

   Each option displays data that is inbound and outbound on the selected interface.
5. (Optional) Change the reporting period: Open the **Timeframe** dialog by clicking the timeframe link.
   The reporting period is the most recent 24-hour period by default.

NOTE
You also can display **Protocol Summary** views in a Custom report.

Open a Drilldown Protocol Report

To drill down to more detailed, protocol-specific data for the selected interface, click the name of a protocol in any of the Top N Protocol views. An overview report opens for the protocol on the selected interface.

Display an Interface Report for a Single Protocol

To drill down to details about a specific protocol, click one of the protocol links in a Top N Protocols view. A report opens, which you can display in **Overview**, **Details**, **Hosts**, or **Conversations** mode. The options in the **Presentation** menu are mode-specific. You can display any of the following types of data and views:

- **Overview** mode **Mixed Chart** or **Mixed Trend** chart: **Rate**, **Volume**, or **Utilization** data for the protocol on the selected interface
- **Overview** mode **Pie Chart** in the following views: **From** (outbound on the interface), **To** (inbound on the interface), and **Total**
- **Details** mode **Multi-Period Trend** chart: **Rate**, **Volume**, or **Utilization** data in any combination of the following views, which are shown with or without baselines:
  - (Displayed by default) **Last Hour**, **Last 2 Hours**, **Last 8 Hours**, and **Daily**
  - **Weekly**, **Monthly**, and **Yearly**
- **Details** mode **Calendar Chart**: Rate in either of the following views:
  - **Direction In**: Protocol data coming into the interface
  - **Direction Out**: Protocol data going out from the interface
- **Hosts** mode **Trend Chart**: **Rate**, **Volume**, or **Utilization** data for the number of top hosts that you specify
- **Hosts** mode **Pie Chart** in the following views: **From** (outbound on the interface), **To** (inbound on the interface), and **Total**
- **Hosts** mode **Summary Table**: Table of **Rate**, **Volume**, or **Utilization** data for hosts who used the protocol on the selected interface
- **Conversations** mode **Trend Chart** or **Summary Table**: **Rate**, **Volume**, or **Utilization** data for conversations that used the protocol on the selected interface
- **Conversations** mode **Pie Chart**: Data for all conversations that used the protocol on the selected interface

To return to the summary view of all protocols on the interface, click the link that is named for the currently selected protocol and click **Select Top N Protocols**.

Find Protocol Views in the Performance Center Console

Protocol data from DX NetOps is displayed in the following locations in the Performance Center Console. (The built-in dashboards and report pages that are noted here show the views by default.)

- Top Enterprise Protocols by Volume
  - (CA PC) **Infrastructure Overview** and **Network Overview** dashboards; **Summary** context view in a custom dashboard
  - (NPC) **Enterprise**, **Traffic Analysis**, **Network Overview**, and custom dashboards
- Top Protocols (Bar) and Top Protocols (Pie)
  - (CA PC) Custom dashboards
  - (NPC) Custom dashboards; **Interface Pages**: Interface QoS and custom tab views
- Top Protocols (Table)
  - (CA PC) Custom dashboards
  - (NPC) Custom dashboards; **Interface Pages** custom tab views
- Stacked Protocol Trend

- (CA PC) **Interface Pages: IP Performance** and **CBQoS** report pages
- (NPC) **Interface Pages: Interface Capacity**, **Interface QoS**, and custom tab views

Top N ToS Report

The Top N ToS report provides information about the Types of Service (ToS) markings of the packets that generate the most traffic for the selected interface. This topic describes how to display a Top N ToS report.

Follow these steps:

1. Display an interface report in either of the following ways:
   - Locate and click an interface on the **Interfaces** page.
   - Click an interface link in an existing view--for example, on the **Enterprise Overview** page.
2. Select **ToS** as the report type at the top of the page.
3. Make sure the report scope is set to **Top N ToS**. The **Top N ToS** link should appear next to the report type setting. The report page is updated to show ToS data in stacked trend charts. The report includes views for inbound, outbound, and total ToS data.
4. (Optional) Change the type of data presentation and measurement by using the Presentation options.
   - **Stacked Trend Chart** of **Rate**, **Volume**, or **Utilization** data (Default setting)
   - **Trend Chart** of **Rate**, **Volume**, or **Utilization** data: Use the **Show Top** setting to specify the maximum number of conversations to include.
   - **Pie Chart**
   - **Summary Table** of **Rate**, **Volume**, or **Utilization** data

   Each option displays data that is inbound and outbound on the selected interface.
5. (Optional) Change the reporting period: Open the **Timeframe** dialog by clicking the timeframe link. The reporting period is the most recent 24-hour period by default.

Display an Interface Report for a Single ToS

To drill down to more detailed, ToS-specific data for the selected interface, click a **ToS** link in a **Top N ToS** view. An overview report for the ToS value on that interface opens.

You can view the report in **Overview** or **Details** mode by selecting one of the following report types:

- **Overview** mode **Mixed Chart** or **Mixed Trend** chart: **Rate**, **Volume**, or **Utilization** data for the ToS on the selected interface
- **Overview** mode **Pie Chart** views:
  - **ToS Protocol Summary - In** or **Out**: Data going in or out of the interface for the protocols that use the ToS
  - **ToS Hosts Summary - From** or **To**: Data going from or to the hosts that use the ToS
  - **ToS Conversations Summary - Total**: Data for all conversations that use the ToS (coming in and going out of the interface)
- **Details** mode: Charts of **Rate**, **Volume**, or **Utilization** data in any combination of the following views, which are shown with or without baselines:

  - (Displayed by default) **Last Hour**, **Last 2 Hours**, **Last 8 Hours**, and **Daily**
  - **Weekly**, **Monthly**, **Yearly**
- **Protocols** mode **Trend Chart**: **Rate**, **Volume**, or **Utilization** data, which are shown with or without baselines
- **Protocols** mode **Summary Table**: **Rate**, **Volume**, or **Utilization** data
- **Protocols** mode **Stacked Trend Chart** or **Pie Chart** views of data for the protocols that used the ToS: **In** (inbound on the interface), **Out** (outbound on the interface), or **Total**
- **Hosts** mode **Trend Chart**: **Rate**, **Volume**, or **Utilization** data for the number of top hosts you specify
- **Hosts** mode **Summary Table**: Table of data for hosts who used the ToS on the selected interface
- **Hosts** mode **Pie Chart**: Summary views of data for the hosts that used the ToS: **From** (data from the host), **To** (data to the host), or **Total**
- **Conversations** mode **Trend Chart**: **Rate**, **Volume**, or **Utilization** data for the number of top conversations you specify
- **Conversations** mode **Summary Table**: Table of **Rate**, **Volume**, or **Utilization** data for conversations that used the ToS on the selected interface
- **Conversations** mode **Pie Chart**: Data for all conversations that used the ToS on the selected interface

To return to the summary view, click the link named for the currently selected ToS and click **Select Top N Hosts**.

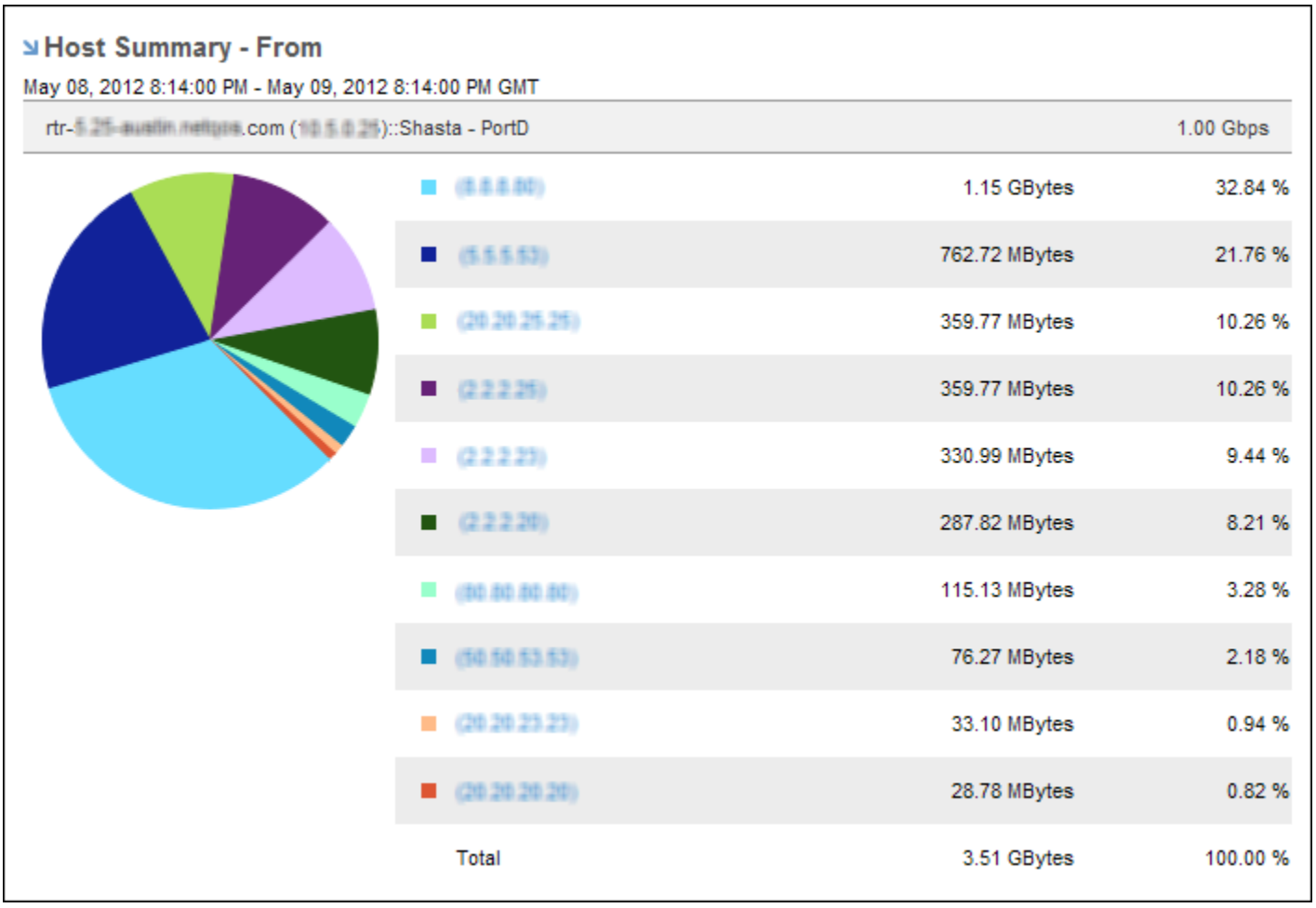
Find ToS Views in the Performance Center Console

ToS data from DX NetOps is displayed in the following Performance Center Console locations. (The built-in dashboards and report pages that are noted here show the views by default.)

- **Stacked ToS Trend**
  - (CA PC) **Interface Pages: CBQoS** report page or a custom dashboard
  - (NPC) **Interface Pages: Interface QoS** or custom tab views
- **ToS Summary Pie**
  - (CA PC) Custom dashboard; **Interface Pages: IP Performance** tab
  - (NPC) **Interface Pages** or custom tab views
- **ToS Summary Table**
  - (CA PC) Custom dashboard
  - (NPC) **Interface Pages: Interface QoS** or custom tab views

Top N Hosts Report

The **Top N Hosts** report provides information about the hosts that generate the most traffic on the selected interface.



Follow these steps:

1. Display an interface report in either of the following ways:
   - Locate and click an interface on the **Interfaces** page.
   - Click an interface link in an existing view, for example, on the **Enterprise Overview** page.
2. Select **Hosts** as the report type at the top of the page.
3. Make sure the report scope is set to **Top N Hosts**. The **Top N Hosts** link should appear next to the report type setting. The report page is updated to show host summary pie charts by default. The report includes views for the following host data:
   - **From**: Top hosts who sent data to the interface.
   - **To**: Top hosts who received data from the interface.
   - **Total**: Top hosts who either sent data to the interface or received data from the interface.
4. (Optional) Change the data presentation type and the data measurement type by using the Presentation options.
   - **Trend Chart** of **Rate**, **Volume**, or **Utilization** data: Show trend charts for data that travels to and from the hosts. Use the **Show Top** setting to specify the maximum number of conversations to include.
   - **Pie Chart**: Show pie charts for data that travels to and from the hosts, with lists of hosts and their data volumes. (Default setting)
   - **Summary Table** of **Rate**, **Volume**, or **Utilization** data: Show a table of data that travels to and from the hosts.
5. (Optional) Change the reporting period: Open the **Timeframe** dialog by clicking the timeframe link. The reporting period is the most recent 24-hour period by default.

Display an Interface Report for a Single Host

To drill down to more detailed, host-specific data for the selected interface, click a host name in a **Top N Hosts** view. An overview report for the host on that interface opens. You can view the host details or the host protocols.

You can view this type of report in **Details** mode or **Protocols** mode, which have the following options:

- **Details** mode: Charts of **Rate**, **Volume**, or **Utilization** data in any combination of the following views:
  - (Displayed by default) **Last Hour**, **Last 2 Hours**, **Last 8 Hours**, and **Daily**
  - **Weekly**, **Monthly**, and **Yearly**
- **Protocols** mode **Trend Chart** or **Summary Table**: **Rate**, **Volume**, or **Utilization** for the protocols that the hosts used. You see the data coming into the interface and the data going out from the interface.
- **Protocols** mode **Stacked Trend Chart**: Data for all protocols that the hosts used (coming into and going out of the interface).

To return to the summary view, click the link that is named for the currently selected host and click **Select Top N Hosts**.

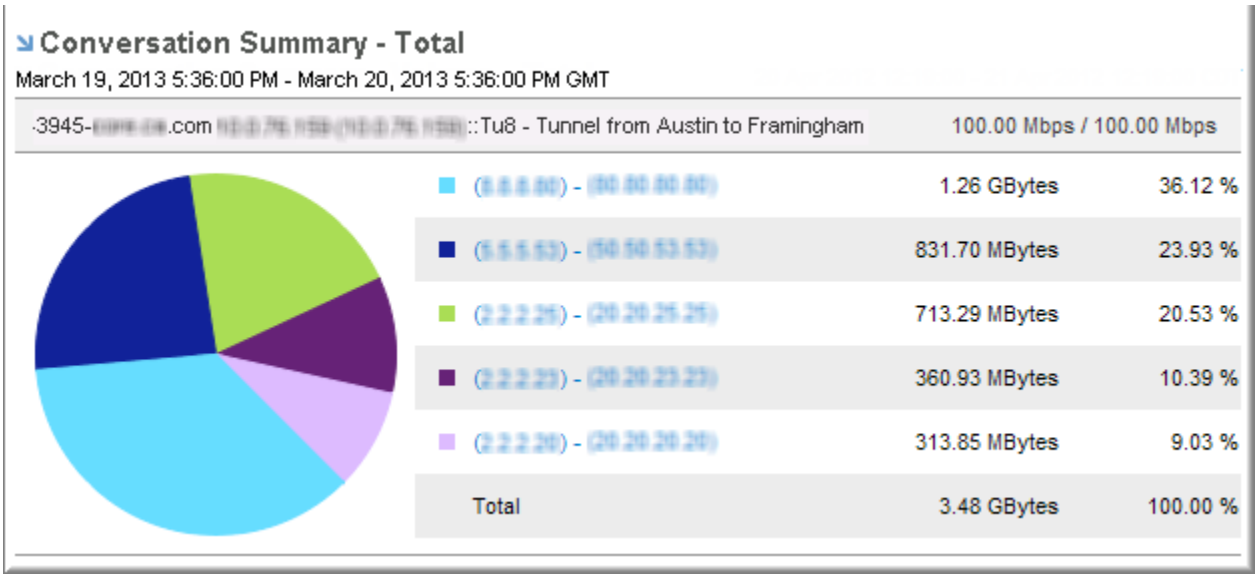
Find Host Views in the Performance Center Console

Host data from DX NetOps is displayed in the following Performance Center Console locations. (The built-in dashboards and report pages that are noted here show the views by default.)

- Top Enterprise Hosts by Volume or Top Hosts (Bar)
  - (CA PC) **Infrastructure Overview**, **Network Overview**, and custom dashboards
  - (NPC) **Enterprise**, **Traffic Analysis**, **Network Overview**, and custom dashboards
- Top Hosts (Bar)
  - (CA PC) Custom dashboards; **Interface Pages: IP Performance** report page
  - (NPC) Custom dashboards; **Interface Pages: Interface Capacity**, **Interface QoS**, and custom tab views
- Top Hosts (Pie)
  - (CA PC) Custom dashboards; **Interface Pages: IP Performance** report page
  - (NPC) Custom dashboards; **Interface Pages: Interface QoS**, and custom tab views
- Top Hosts (Table)
  - (CA PC) Custom dashboards
  - (NPC) Custom dashboards; **Interface Pages** custom tab views

Top N Conversations Report

The **Top N Conversations** report provides information about the hosts that generate the most traffic on the selected interface.



Follow these steps:

1. Display an interface report in either of the following ways:
   – Locate and click an interface on the **Interfaces** page.
   – Click an interface link in an existing view, for example, on the **Enterprise Overview** page.
2. Select **Conversations** as the report type at the top of the page.
3. Make sure the report scope is set to **Top N Conversations**. The **Top N Conversations** link should appear next to the report type setting.
   The report page is updated to show a conversation summary pie chart by default.
4. (Optional) Change the data presentation type and the data measurement type by using the Presentation options.
   – **Trend Chart** of **Rate**, **Volume**, or **Utilization** data: Show a trend chart for data that travels to and from the source host in the conversations. (Default setting)
   Use the **Show Top** setting to specify the maximum number of conversations to include.
   – **Pie Chart**: Show a summary pie chart with a list of conversations and their data volumes.
   – **Summary Table** of **Rate**, **Volume**, or **Utilization** data: Show a table of data that travels to and from the source host in the conversations.
5. (Optional) Change the reporting period: Open the **Timeframe** dialog by clicking the timeframe link.
   The reporting period is the most recent 24-hour period by default.

Display an Interface Report for a Single Conversation

To drill down to details about a specific conversation, click one of the conversation links in the **Top N Conversations** view. A report opens, which you can configure to show any of the following data:

You can view this type of report in **Details** mode or **Protocols** mode, which have the following options:

- **Details** mode: Charts of **Rate**, **Volume**, or **Utilization** data in any combination of the following views:
  – (Displayed by default) **Last Hour**, **Last 2 Hours**, **Last 8 Hours**, and **Daily**
  – **Weekly**, **Monthly**, or **Yearly**
- **Protocols** mode **Trend Chart** or **Summary Table**: **Rate**, **Volume**, or **Utilization** for the protocols that the conversations used. You see the data coming into the interface and the data going out from the interface.
- **Protocols** mode **Stacked Trend Chart**: Data for all protocols that the conversations used (coming into and going out of the interface).

To return to the summary view of all conversations on the interface, click the link that is named for the currently selected conversation and click **Select Top N Conversations**.

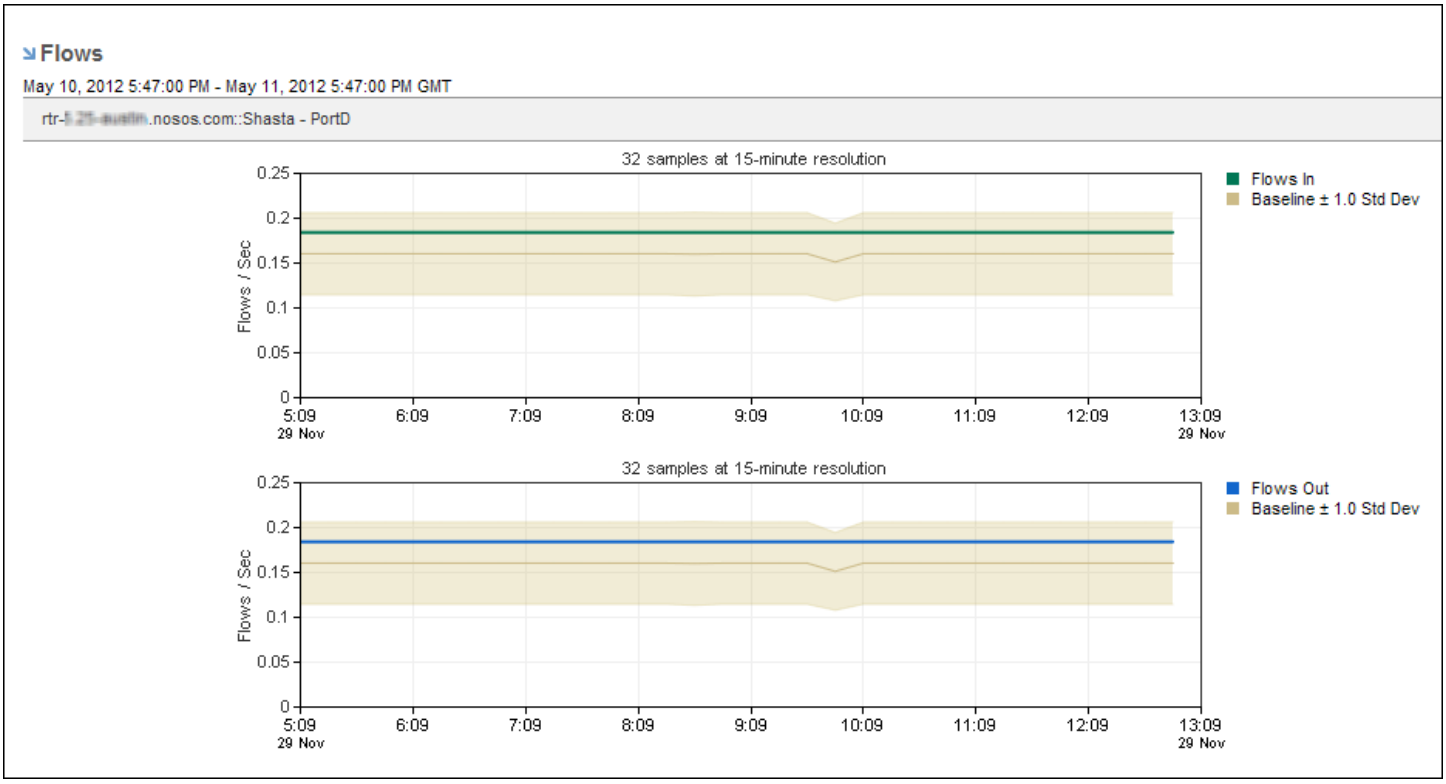
Find Conversation Views in the Performance Center Console

Conversation data from DX NetOps is displayed in the following Performance Center Console locations. (The built-in dashboards and report pages that are noted here show the views by default.)

- Top Conversations (Bar)
  - (CA PC) Custom dashboards
  - (NPC) **Interface Pages: Interface Capacity** and custom tab views
- Top Conversations (Pie)
  - (CA PC) Custom dashboards; **Interface Pages: IP Performance** report page
  - (NPC) **Interface Pages: Interface QoS** and custom tab views
- Top Conversations (Table)
  - (CA PC) Custom dashboards
  - (NPC) **Interface Pages** custom tab views

Flows Report

A **Flows** report is a trend plot that shows the rate of the flows that enter and leave the selected interface. A **Flows** report helps you find patterns or anomalies.



Viruses typically generate large increases in flow counts. The flow rate can indicate the load on the Harvester.

Follow these steps:

1. Display an interface report in either of the following ways:

- Locate and click an interface on the **Interfaces** page.
- Click an interface link in an existing view--for example, on the **Enterprise Overview** page.

2. Select **Flows** as the report type at the top of the page.
   The report page is updated to show flow trend charts.
3. (Optional) Change the data measurement type by using the Presentation options:
   Rate (default setting) or **Volume**
   Each option displays trend charts of data that is inbound and outbound on the selected interface.
   You can also select **Show Baselines** to view +/- 1 Standard Deviation. The baseline is computed by calculating the average and standard deviation for a maximum of 10 samples (the last six weeks and the last four days). This rolling baseline feature provides a visual representation of a current and historical trend overlay. When a current trend line is above or below the baseline, the performance is out of the norm. When the current trend is within the baseline, the performance is within the range of historical behavior.
4. (Optional) Change the reporting period: Open the **Timeframe** dialog by clicking the timeframe link.
   The reporting period is the most recent 24-hour period by default.

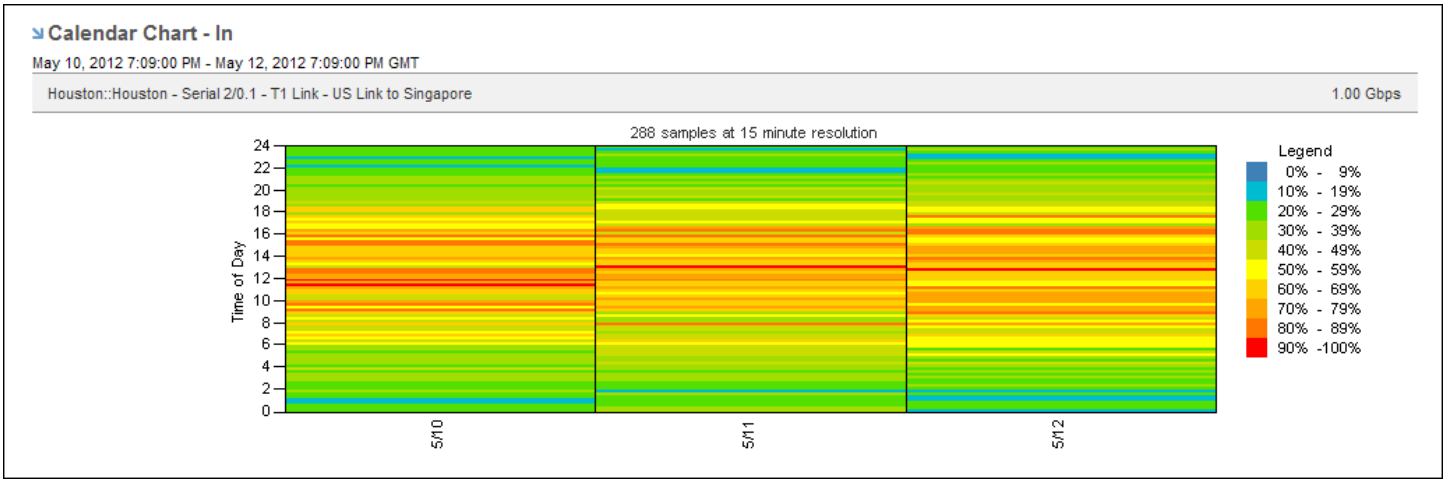
Find Flows Views in the Performance Center Console

The Top Flows by Interface view is displayed in the following Performance Center Console dashboards.

- (CA PC) **Infrastructure Overview** dashboard (by default)
- (CA PC) **Management Overview** dashboard (by default)
- (CA PC) Summary context view in a custom dashboard
- (NPC) Custom dashboard

Utilization (Calendar Chart) Report

The **Utilization** (**Calendar Chart**) report maps the utilization percentage of the selected interface over time. Utilization is calculated on either inbound or outbound traffic, depending on the selected **Presentation** mode.



This view makes it easy to detect recurring data patterns. Finding a pattern can help you identify the source of high traffic rates and potential performance issues. You might discover that the high traffic rates you thought were intermittent actually follow a pattern. The view can show the hour of each day when utilization is the highest, for example.

Each color represents a severity range that is calculated as a percentage of total capacity. High utilization is shown in orange and red. Low utilization is shown in green and blue.

Follow these steps:

1. Display an interface report in either of the following ways:

- Locate and click an interface on the **Interfaces** page.
- Click an interface link in an existing view; for example, on the **Enterprise Overview** page.

2. Select **Utilization** from the report type menu at the top of the page.
   The report page is updated.
3. (*Optional*) Choose the direction of the traffic by using the **Presentation** options:
   - **Direction In**: Show traffic that is inbound on the interface.
   - **Direction Out**: Show traffic that is outbound on the interface.
4. (*Optional*) Change the reporting period: Open the **Timeframe** dialog by clicking the timeframe link.
   The reporting period is the most recent 24-hour period by default.

   NOTE
   The **Calendar Chart** report uses 15-minute resolution when the **Time Period** is **Last Hour** or **Last 2 Hours**.

Find Calendar Chart Views in the Performance Center Console

To see the utilization calendar chart for an interface in the Performance Center Console, add it to a custom dashboard or report page:

- (CA PC) Custom dashboard with **Calendar Heat Chart (Flow) - In** or **- Out**
- (NPC) **Interface** pages: Custom report page with **Calendar Chart (Flow) - Total**

Growth Report

The **Growth Report** provides historical growth statistics for the top protocols on the selected interface, so you can see which applications have increasing capacity needs. This knowledge helps you make decisions about adding capacity in the future.

Follow these steps:

1. Display an interface report in either of the following ways:
   - Locate and click an interface on the **Interfaces** page.
   - Click an interface link in an existing view; for example, on the **Enterprise Overview** page.
2. Select **Growth Report** as the report type at the top of the page.
   The report page is updated to show a **Growth Report** table.
3. (*Optional*) Choose the direction of the traffic to display by using the **Presentation** options.
   - **Direction In**: Show traffic that is inbound to the interface.
   - **Direction Out**: Show traffic that is outbound from the interface.
4. (*Optional*) Change the reporting time period:
   a. Select **Last 6 Weeks** (default setting) or **Last 6 Months** in the **Presentation** options.
      - **Last 6 Weeks**: Sets the report to show six weeks of data. The six most recent weeks are selected by default.
      - **Last 6 Months**: Sets the report to show six months of data. The six most recent months are selected by default.

   b. (*Optional*) Use the **Through week of** or the **Through month of** option to select an alternative ending week or month for the six-week or six-month time period.
   c. (*Optional*) Restrict the reporting range by selecting a time filter from the **Time Filter** option list.
      Time filters are created by the Administrator for DX NetOps. For example, the Administrator could create a time filter to restrict the report data to business hours and business days. If the **Time Filter** list is empty, your Administrator has not created any time filters.
      Select **Time Filter: None** when not using a time filter.

Capacity Planning Report

Capacity Planning reports show traffic trends, which are useful for planning. Future traffic is calculated by analyzing previous traffic. **Capacity Planning** reports project trends for rate, volume, and utilization for the following data on the selected interface:

- Overall traffic (IP Summary)
- Protocol traffic
- ToS traffic

> **NOTE**
> In some cases, when there are short-term variations in the metric of interest that are not reflective of the longer-term trend, the capacity planning tables may show odd numbers.

You can specify several display and calculation options for the report:

- Time span of the historical (actual) data, which is shown with a white background on the view
- Time span of the projected data, which is shown with a gray background on the view
- Time range of the data that is used to calculate projections and (optionally) a time filter to exclude certain time periods from calculations
- Analysis type algorithm: **Daily Percentile** or **Daily Average**
- Threshold line, which is configured based on your selected percentage and selected bandwidth or speed

Follow these steps:

1. Display an interface report in either of the following ways:
   – Locate and click an interface on the **Interfaces** page.
   – Click an interface link in an existing view--for example, on the **Enterprise Overview** page.
2. Select **Capacity Planning** as the report type at the top of the page.
3. Select the report scope:
   – **IP Summary**: Overview of traffic on the selected interface
   – **Protocols**: Protocol traffic on the selected interface
   – **ToS**: ToS traffic on the selected interface
   The report page is updated.
4. (Optional) Change the reporting time frame and calculation options by using the **Trend Settings** options.
5. (Optional) Change the data presentation type and the data measurement type by using the **Presentation** options: **Rate** (default setting), **Volume**, or **Utilization**
   Each option displays views of data that is inbound (**In**) and outbound (**Out**) on the interface.

Views on the Capacity Planning Report Page

You can configure the **Capacity Planning** report views to display historical data and calculate projections for three types of interface data on the selected interface:

- IP Summary - Overview of traffic
- Protocols - Protocol traffic
- ToS - ToS

To change the data type, select a different option from the secondary menu at the top of the page.

The views that are shown on the report page are determined by two sets of options:

- Secondary report mode: **IP Summary** (default setting), **Protocols**, or **ToS**
- Presentation mode: **Rate** (default setting), **Volume**, or **Utilization**

IP Summary Report Page

The **IP Summary** report page displays the following view and table.

- **IP Summary Trend - Total** - Overall traffic inbound and outbound on the selected interface
  Two trend charts are included:
  – **Rate presentation** - **Rate In** and **Rate Out** trend charts show the data rates, expressed in a scale that is appropriate for the highest-rate value in the view (Y-Axis).
  – **Volume presentation** - **Bytes In** and **Bytes Out** trend charts show the data volume, expressed in a scale that is appropriate for the highest volume in the trend chart (Y-Axis).
  – **Utilization presentation** - **Utilization In** and **Utilization Out** trend charts show the utilization of interface capacity, which is measured in percentages (Y-Axis).
  The X-Axis shows date and time progression.
  The charts show color-coded lines:
  – Green line - Rate or volume of inbound data, or the utilization of inbound capacity
  – Blue line - Rate or volume of outbound data, or the utilization of outbound capacity
  – Red line - Threshold, which is configured by setting the Percentage value in the Trend Settings dialog (Calculations area)
  The historical data has a white background; the projected data has a gray background.
- **IP Summary Table**
  The **IP Summary** table contains the following columns:
  – **Direction**: Traffic direction with respect to the interface, either inbound (In) or outbound (Out)
  – **Trend**: Icon that shows whether the traffic is increasing or decreasing
  – **Daily Change**: Change over a day that is calculated for the rate (bps), volume (Bytes), or utilization (%)
  – **Days Until Threshold**: Number of days until the traffic is expected to reach the known capacity of the interface
  – **Date of Threshold**: Calendar date on which the traffic is expected to reach the known capacity of the interface

Protocols or ToS Report Page

The **Protocol** and **ToS** report pages display stacked trend charts of the historical traffic for a maximum of 12 protocols or ToS. A stacked trendline shows the predicted future traffic for each protocol or ToS.

The report page displays the following views and tables.

- **Protocol Stacked Trend** or **ToS Stacked Trend** (In and Out versions) - Inbound and outbound protocol traffic on the selected interface
  – Rate presentation - Data rates, expressed in a scale that is appropriate for the highest-rate value in the view (Y-Axis).
  – Volume presentation - Data volume, expressed in a scale that is appropriate for the highest volume in the view (Y-Axis).
  – Utilization presentation - Utilization of capacity for the selected interface, which is measured in percentages (Y-Axis).
  The X-Axis shows date and time progression.
  The red Threshold line shows the outbound threshold level, which is set in the **Percentage** field of the **Trend Settings (Calculations)** dialog.
  The historical data has a white background; the projected data has a gray background.
- **Protocol** or **ToS Table** (In and Out versions)
  The tables contain the following columns:

- **Protocol Name**: (Protocol table only) Name of the protocol, which may consist of its keyword and encapsulation (transport protocol and port number).
- **Type of Service**: (ToS table only) ToS label, which may be a default label or a label your administrator has configured
- **Trend**: Icon that shows whether the traffic is increasing or decreasing
- **Growth %**: Percentage of traffic increase
- **Current %**: Percentage of the total traffic that the protocol or ToS is using at the end date of the historical analysis (the current date)
- **Projected %**: Percentage of the total traffic that the protocol or ToS is using at the end date of the traffic projection

The **Trend** and **Daily Change** columns are defined in the **IP Summary** table description.

Trend Settings for Capacity Planning Reports

When you open a **Capacity Planning** report, the **Trend Settings** controls are at the top of the page. **Display** options are on the left and **Calculations** options are on the right. Use these settings to control the data display and projection calculations in the report.

Trend Settings Display Options

Use the **Display** options to specify the historical time period and the projection time period displayed in the report. This setting does not determine the data that is used to make projection calculations.

Follow these steps:

1. In the **Historical Display** section, select a time period from the **Show actual data from** list.
   You can select a time period that is relative to today's date, such as **Last 7 days**, **Last 1 month**, **Last 2 months**, and **Last 3 months**. Alternatively, you can select **Custom** to specify a specific date.

   NOTE
   The report displays the data starting at this time but does not necessarily use this time as the starting point to calculate the projection. For example, if you want the projection to calculate from six months of historical data but want to display only seven days of that data. In this scenario, you would select **Last 7 days** in this menu.
2. Accept the default start date or specify a custom start date in any of the following ways:
   - Click the calendar icon and choose a date from the calendar pop-up.
   - Enter a date value in the **Start** box in the DD/MM/YYYY format.
   - Enter a custom start time in the box on the right. Use the 24-hour time format.
3. Accept the default time period or select a time period from the **Show projection for list** in the **Projection Display** section.
   You can select a time period relative to the **Historical Display** end date (such as 1 month) or you can select **Custom** to specify a date.
4. Accept the default end date or specify a custom end date for the projection in any of the following ways:
   - Click the calendar icon and choose a date from the calendar pop-up.
   - Enter a date value in the **End** box in the DD/MM/YYYY format.
   - Enter a custom end time in the box on the right. Use the 24-hour time format.
5. Click **Set**.
   The report views below the **Trend Settings** regenerate and reflect your changes.

   NOTE
   If you make further changes, click **Set** again.

Trend Settings Calculations Options

Use the **Calculations** options to specify the data, calculation method, and threshold for calculating the projection data. These time period settings do not affect the projection data displayed in the report.

Follow these steps:

1. In the **Projection Slope Calculation** section, use the **Use data from** list to select the historical time period you want use to calculate the projection data.
   You can select a time period that is relative to the current date, such as **Last 7 days**, **Last 1 month**, **Last 2 months**, or **Last months**. Alternatively, you can select **Custom** to select a specific start date.

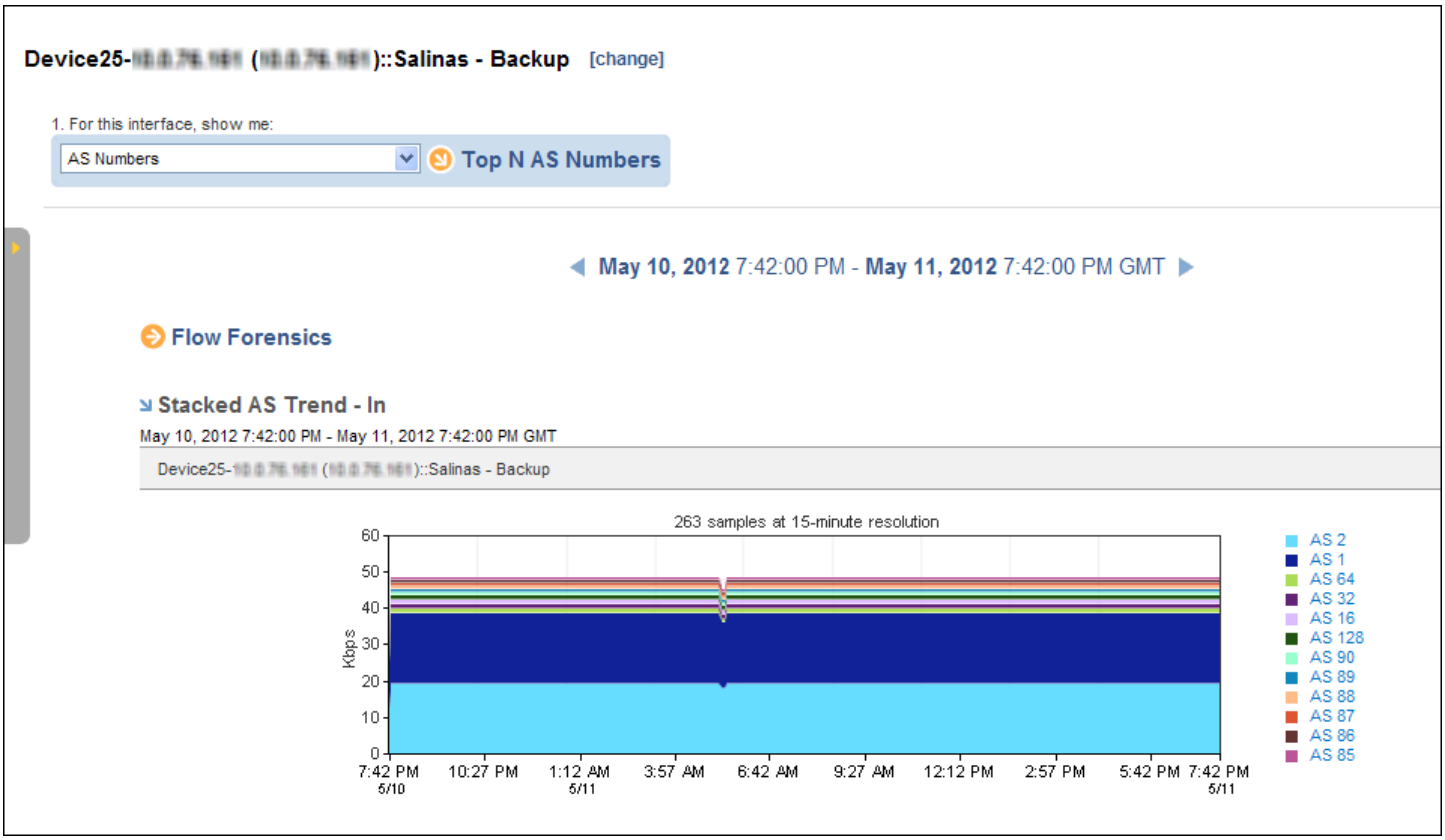
   NOTE
   The report uses this starting time for calculations but does not necessarily use this time to display historical data in the report. For example, if you want the projection to calculate from 6 months of historical data but want to display only seven days of that data. In this scenario, you select **Last 6 months** in this menu.
2. Use the **Time Filters** list to select a filter for restricting the data.

   NOTE
   User accounts that are assigned the Power User or the Administrator role can create custom time filters.
3. Accept the default start date or specify a custom start date in any of the following ways:
   - Click the calendar icon and choose a date from the calendar pop-up.
   - Enter a date value in the **Start** box in the DD/MM/YYYY format.
   - Enter a custom start time in the box on the right. Use 24-hour time format.
4. Select a calculation method from the **Calculate using** list and enter a value in the **Percentage** box.
   You can select **Daily Percentile** or **Daily Average**. To use the **Daily Percentile** method, enter a value in the **Percentage** field.
5. Enter a value in the **Percentage** box and select a measurement from the **Of** list in the **Threshold Line** section.
   The default measurement is **Actual Bandwidth**.
6. Click **Set**.
   The **Trend** report views regenerate and reflect your changes.

   NOTE
   If you make further changes, click **Set** again.

Top N Autonomous System Numbers Report

The **Top N AS** report provides AS (Autonomous System) Next Hop data. The Next Hop data provides high-level summaries of total volume and rate statistics for each AS number on a particular interface. This type of reporting is valuable for managing network routes efficiently. The information helps service providers control costs by comparing traffic usage through transit networks versus traffic through networks with peer agreements.



Follow these steps:

1. Display an interface report in either of the following ways:
   - Locate and click an interface on the **Interfaces** page.
   - Click an interface link in an existing view--for example, on the **Enterprise Overview** page.
2. Select **AS Numbers** as the report type at the top.
3. Make sure the **Top N AS Numbers** link appears next to the report type setting.
   The report page is updated.
4. (Optional) Change the data presentation type and the data measurement type by using the **Presentation** options.
   - **Stacked Trend Chart** of **Rate**, **Volume**, or **Utilization** data
   - **Trend Chart** of **Rate**, **Volume**, or **Utilization** data
   - **Pie Chart**
   - **Summary Table** of **Rate**, **Volume**, or **Utilization** data

   Each option displays views of data that is inbound (**In**) and outbound (**Out**) on the interface and data that is coming from (**From**) the previous hop and going to (**To**) the next hop.
5. (Optional) Change the reporting period: Open the **Timeframe** dialog by clicking the timeframe link.
   The reporting period is the most recent 24-hour period by default.

Work with Interface Reports and Data Views

Each **Interface** report includes specific view types that display the data for the selected interface or group of interfaces. When you open an **Interface** report, the data is displayed in the report views uses the default **Presentation** options and settings.

Change the Interface for a Report

If you click an interface link on the **Interface** page, you drill down to an interface report. You can use the **[change]** link to view a report for a different interface.

Follow these steps:

1. Click the **[change**] option next to the interface name at the top of the report.
   The **Interface Index** opens.
2. Locate the interface that interests you in either of the following ways:
   – Expand the parent router and select an interface from the list of details.
   – Use the **Search** field to locate an interface.
3. Click the link in the **Interface** column.
   The **Interface Index** closes. You return to the **Interfaces** report page, which displays data for the selected interface.

Set the Time Period for a Report

The **Enterprise Overview** displays data for the previous 24-hour period. **Interface** reports also display data for the previous 24 hours by default. You can change the reporting period by using the time period controls at the top of the report page.

When you view **Interfaces** report pages, you can change the time period for the displayed data. For example, if you notice an issue in a 1-day report, you might want to change the time period to be the last seven days. Expanding the time period can reveal whether the issue occurs daily.

The current time period is shown at the top of most **Interfaces** report pages. These interface reports reflect 1-minute resolution data. For a 24-hour time period, the report shows the data points in the previous 24 hours, including the final full 1-minute data point. For example, if you generate or refresh the report on May 11, 2012 at 16:11 GMT, the time period for the report is May 10, 2012 16:10:00 through May 11, 2012 16:10:00.

The time period link is on all top-level **Interfaces** report pages except **Growth Report** and **Capacity Planning**, which have other time options. In addition, if you drill down to a report page to investigate details, the drilldown report page does not include a time period link.

- The **Growth Report** uses a 6-weeks or 6-month time period.
- The **Capacity Planning** report uses two different time periods for calculations and reporting data.

> **NOTE**
> Each user account has an assigned time zone, which determines how time is labeled in reports. For example, if a user who has a time zone of Central Standard Time (CST) views a report with data for 8:00 to 9:00 A.M., the data for 8:00 to 9:00 A.M. CST appears in the report. An Administrator can modify this setting for the user account.

The time period at the top of the report page is a link to the options you use to change the reporting time period.

Scroll the Time Period Interval

You can keep the current time span length for the interface report (such as 24 hours, one week, one month, or custom), but can shift the time period backward or forward. For example, you can scroll the time period back to show data for the week before the week now on display.

To scroll the time period back by one increment, click

the **Back** icon. 

To scroll the time period forward by one increment, click the **Forward** icon.

Specify a Built-In Time Period

You can specify a different time period for **Interface** reports by using one of the built-in options. The built-in time periods let you quickly expand or restrict the reporting time period relative to the current date and time. For example, you can extend the time period to find out if an observed event happens as part of a repeated pattern.

Follow these steps:

1. Click the time period link at the top of the **Interfaces** report page.
   The **Timeframe** options expand.
2. Select a time period for the report from the **Time Period** list.
   The time period is relative to the current date and time. A **Daily** time period produces a report about the 24 hours that precede the current date and time. A **Weekly** time period produces a report the week that precedes the current date and time.
3. Select a time filter from the **Time Filter** list.The available time filters are set up by the Administrator for DX NetOps. Time filters limit the time span for reported data, for example to include only regular business hours.Select **Time Filter: None** when not using a time filter.
4. Click **Set**.The active time period displayed in the **Timeframe** pane changes to give you a preview of the data that is included. The preview helps you verify that you are capturing any specific trends that are of interest.
5. Click **Close** at the top-right corner of the pane.The options are hidden.

Specify a Custom Start and End Time

You can customize the time period for the current report. You can select a specific start date, end date, and time of day for collecting report data, such as the following times:

- Specific hour.
- Specific day.
- Unique week time period by specifying a day within the Saturday-to-Sunday 7-day time period that you want.
- Unique one-month time period by selecting the month start and end dates.
- Unique quarter-year time period by selecting the quarter start and end dates.
- Unique one-year time period by selecting the year start and end dates.

Follow these steps:

1. Click the time period at the top of the report page.
   The **Timeframe** pane expands to display the time period options.
2. Select **Custom** from the **Time Period** list. The end date and time are set to the current time by default.
3. Select **Time Filter: None** when not using a time filter.
4. Select the start date from the **Start Date** day, month, and year lists, or click the calendar icon to locate and select a date.
5. Select a start time from the **Start Date** Hour and Minute lists. The current time (hour:minutes) is selected by default. Hours are shown in 24-hour format.
6. Select an end date from the **End Date** day, month, and year lists, or click the calendar icon to locate and select a date.
7. Select an end time from the **End Date** Hour and Minute lists or click the calendar icon to locate and select a date. The current time (hour:minutes) is selected by default.
8. Click **Set**.
   The preview in the **Timeframe** pane updates to reflect your changes. This view helps you make sure you are capturing the needed data.
9. Click **Close** at the top-right corner of the pane.

The options are hidden.

Set the Presentation Options

You can use the **Presentation** menu to choose the way data is presented in a drilldown **Interface** report.

To open the **Presentation** menu, click the gray bar on the left edge of the page.

The **Presentation** menu opens and gives you access to the options that are appropriate for the selected report type. The display options in the top part of the menu may include one of the following report types:

- Trend Chart
- Pie Chart
- Summary Table

The selected presentation format determines which metric options are available. For example, if you choose a trend chart as the display type, you can choose rate, volume, or utilization for the metric.

Keep Settings at Top: As you scroll up or down the page, the **Presentation** menu moves to remain in view by default. To position the **Presentation** menu in a fixed position at the top of the page, select **Keep Settings at Top**.

Set the Display Option

Display options determine the presentation of the data in the report views. Each **Interface** report has a default display type, but you can use the **Presentation** options to change the display type. The display types that are available depend on the report.

- Stacked Trend Charts
  - Top N Protocols
  - Top N ToS
- Trend Charts
  - Top N Protocols
  - Top N ToS
  - Top N Hosts
  - Top N Conversations
  - Flows
- Mixed Chart
  - Overview
  - Capacity Planning
- Mixed Trend

  - Overview
- Pie Charts
  - Overview
  - Top N Protocols
  - Top N ToS
  - Top N Hosts
  - Top N Conversations
- Summary Tables
  - Top N Protocols
  - Top N ToS
  - Top N Hosts
  - Top N Conversations
- Growth
  - Growth Report
- Calendar Charts
  - Utilization

Set the Rate, Volume, Utilization Option

Interface reports support the display of rate, volume, and utilization metrics. The metrics available for the report depend upon the type of data included in the report, and sometimes the selected display option.

Show Other Option

When you display a Top-N report for protocols, hosts, and conversations, you can include data for items other than the top ten. To include the other items, select the **Show Other** check box in the **Presentation** options. The data for the other protocols, hosts, or conversations is then rolled up and assigned a label of **Other**.

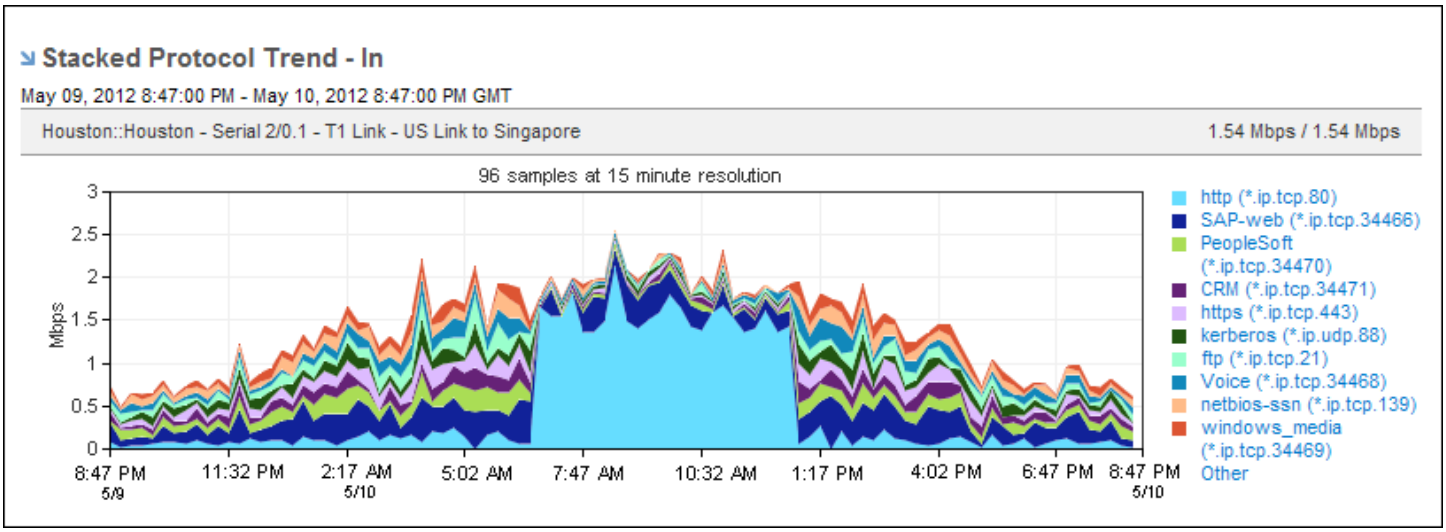
Analyze Interface Report Data

The Interface reports included in DX NetOps include numerous view types that are designed to help you identify and analyze your network traffic. These real-time report views provide the previous hour of performance data by default in a 1-minute granularity view. By using this data, operations center personnel can determine possible causes for new issues and can troubleshoot the issues quickly.

Protocol Trend Views

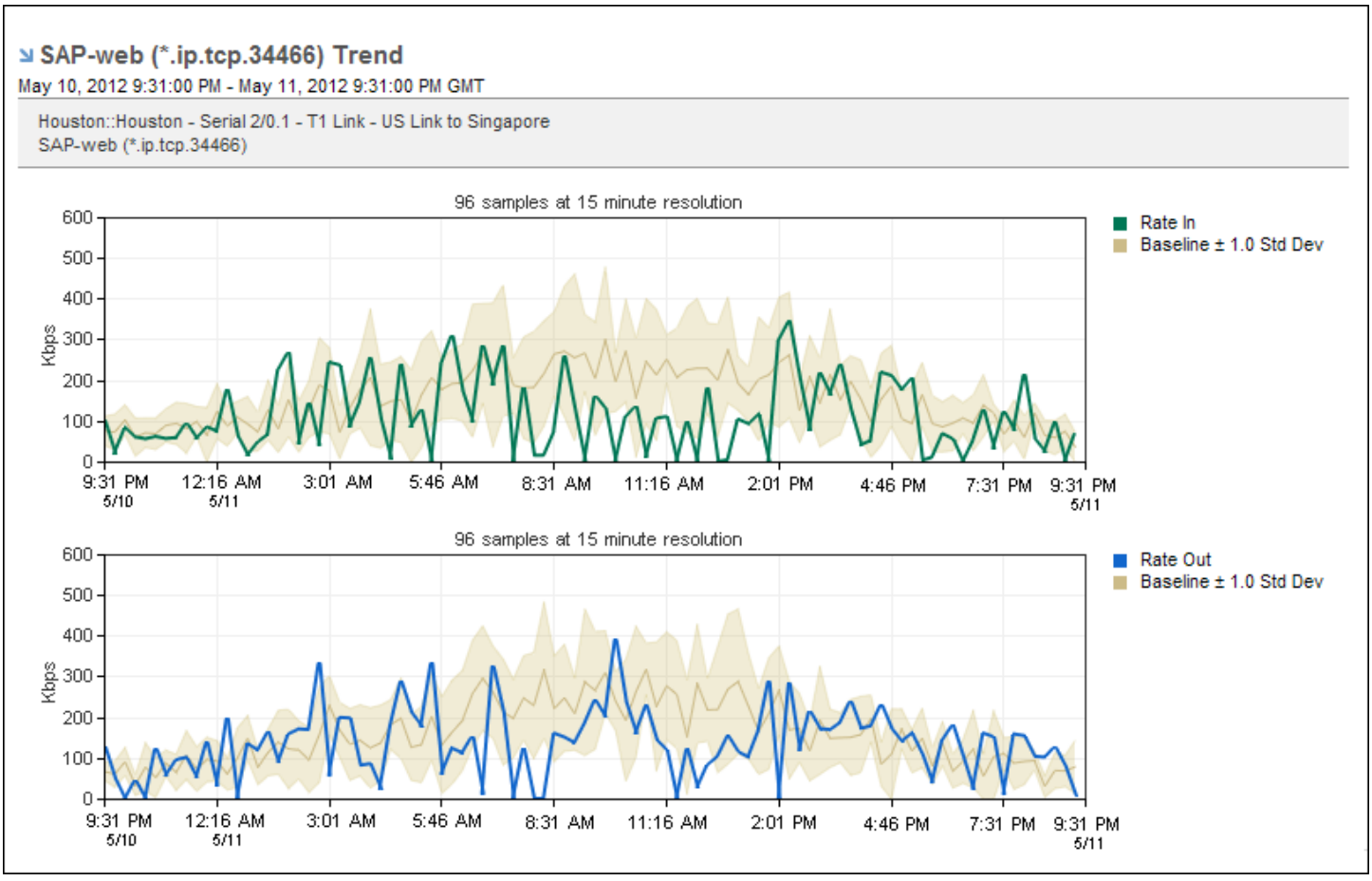
The data in **Stacked Protocol Trend** views helps you determine which type of traffic is consuming bandwidth on the interface. These views also show the way bandwidth consumption changed over the selected time period.

Click the name of a protocol in the legend to display a protocol report with interface data specific to that protocol.



The **Stacked Protocol Trend** views display data for the top ten protocols on the interface. To display data for protocols other than the top ten protocols, select **Show Other** in the **Presentation** menu. Data for protocols other than the top ten is rolled up and assigned a label of **Other**.

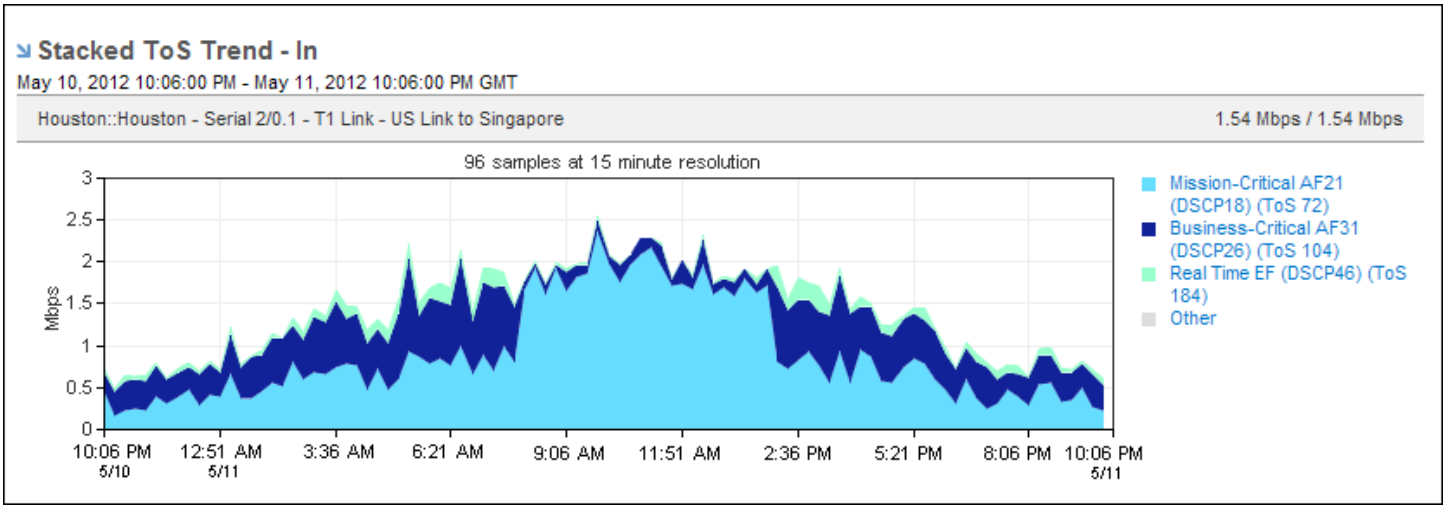
Change the presentation display to a trend chart to display each protocol as an individual trend plot. A trend chart is useful for comparing the protocol data patterns against a baseline.



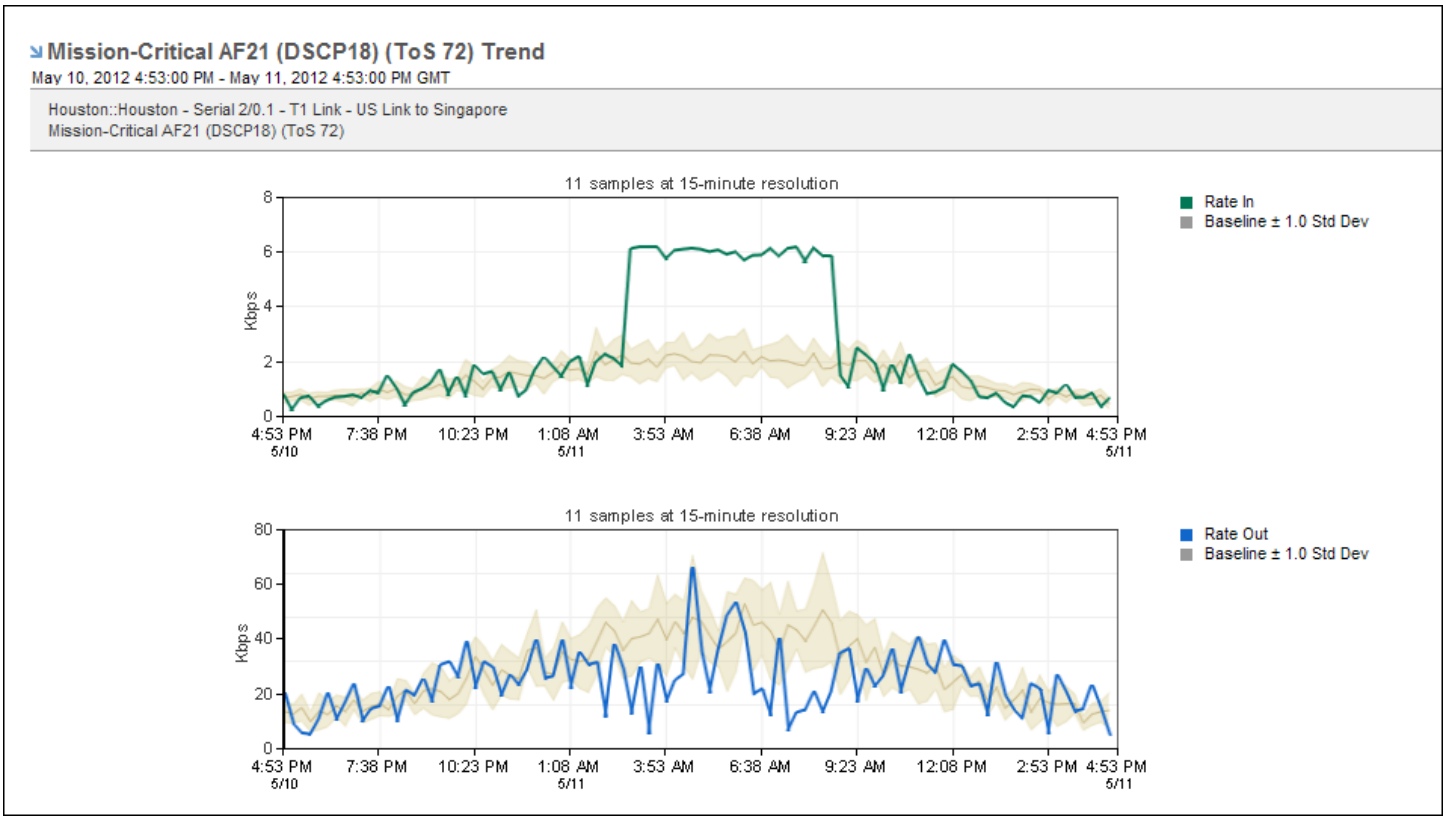
ToS Trend Views

The data in **Stacked ToS Trend** views helps you determine the amount of traffic on the interface by ToS designation and its change over the selected time period.

Click the name of a ToS in the legend to display the ToS report with interface data specific to that ToS.

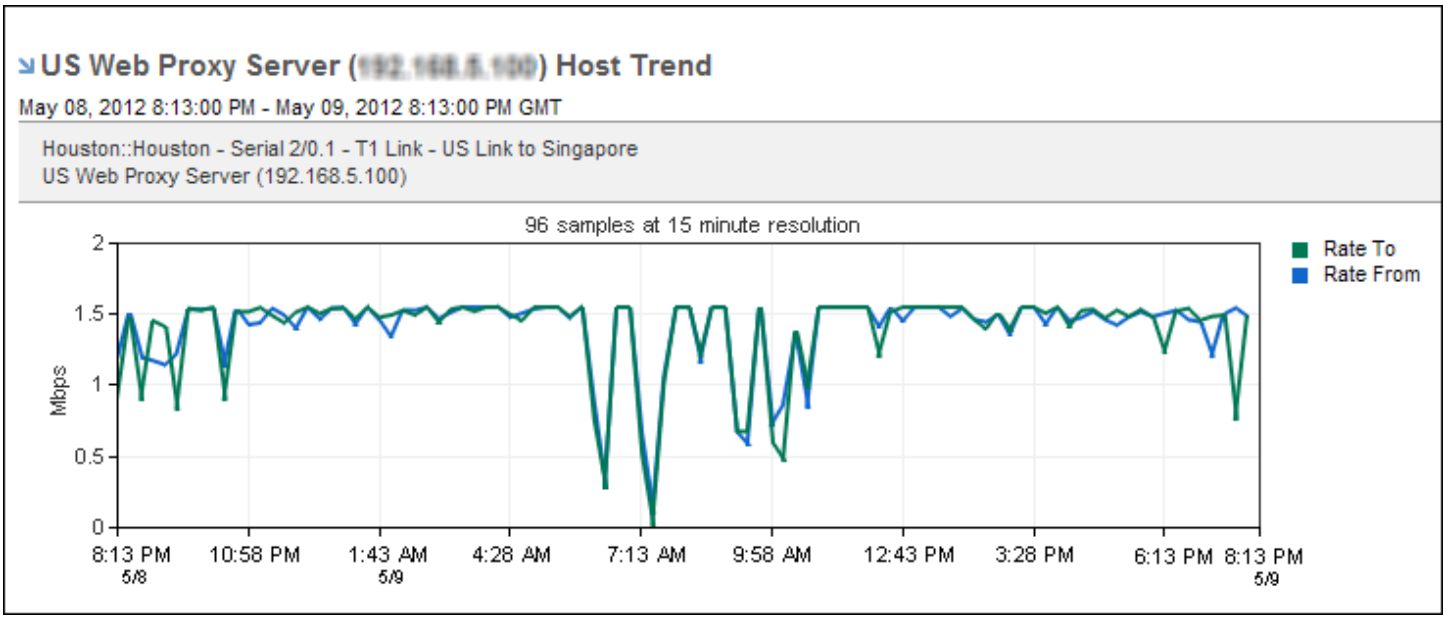


Change the presentation display to a trend chart to display each ToS as an individual trend plot. A trend chart is useful for comparing the ToS data patterns against a baseline.



Host Trend Views

Host data views are displayed as summary views (pie chart or summary table) by default. You can view individual trend plots for each of the hosts on the selected interface. To view individual trend plots, select the **Trend Chart** display type in the **Presentation** options for the **Top-N Hosts** report.

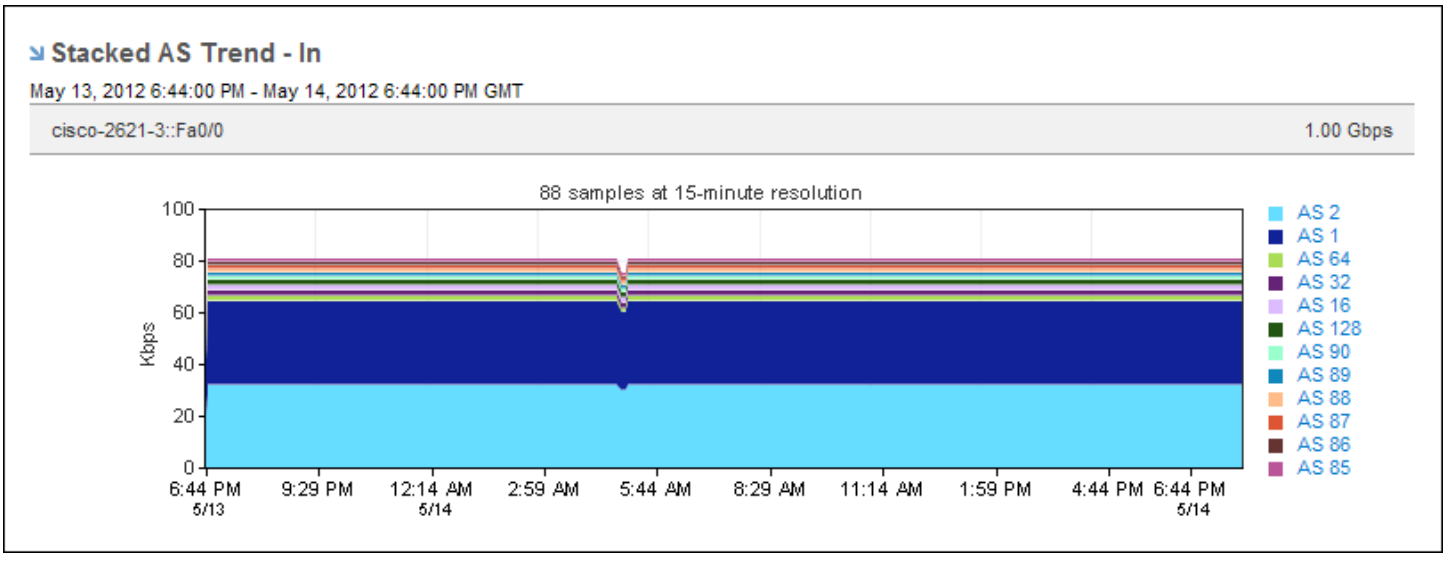


Host trend charts plot a line for each direction of the host traffic on the interface.

AS Number Trend Views

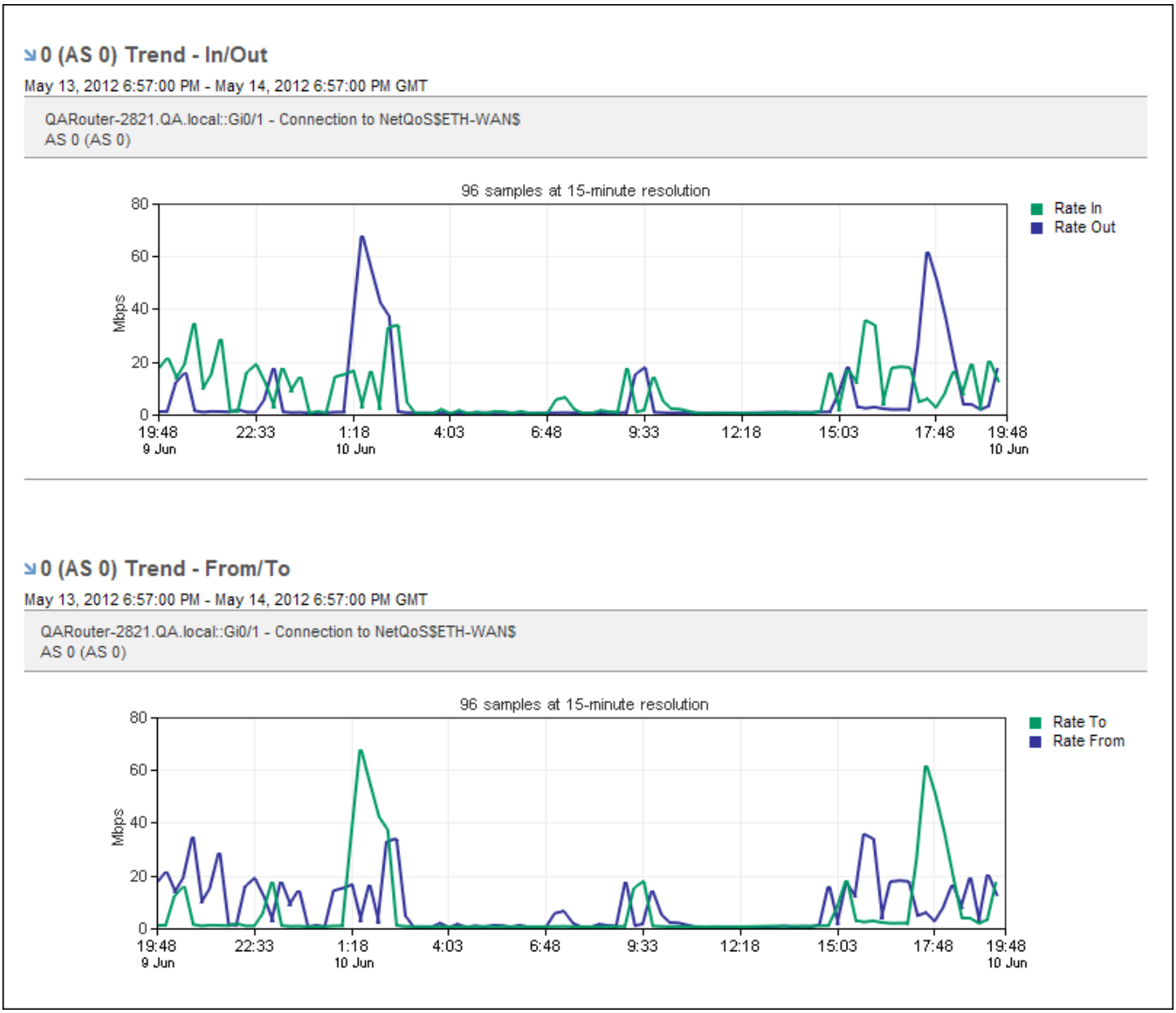
The data in the **Stacked AS Trend** views helps you determine the usage of numbered routes in a network. Use this AS (Autonomous System) Next Hop data to troubleshoot issues.

Click the name of an AS number in the legend to display an AS Next Hop Summary Table specific to that AS number.



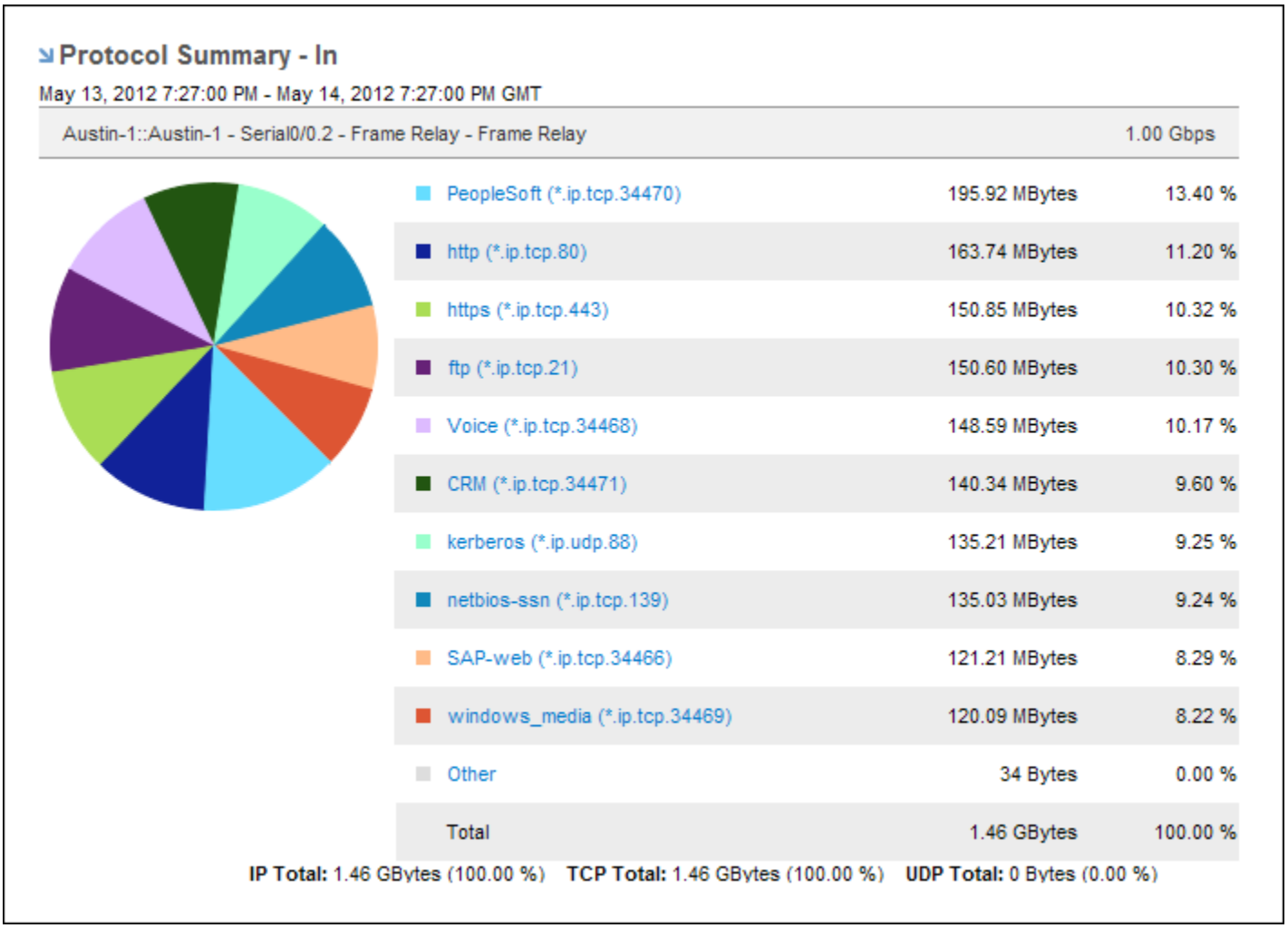
The **Stacked AS Trend** views display data for the top 10 AS numbers on the interface. To display data for AS numbers other than the top ten, select **Show Other** in the **Presentation** menu. Data for AS numbers other than the top ten is rolled up and assigned a label of **Other**.

Change the presentation display to a trend chart to display each AS number as an individual trend plot. A trend chart is useful for comparing the inbound and outbound AS number data patterns.



Protocol Summary Views

The **Protocol Summary** views can help you determine which protocols produce the most traffic over the interface during the selected time period. These views are an overview and provide a good starting point for troubleshooting issues.



Protocol Summary views are included on Interface pages for specific interfaces when the selected report type is **Overview** or **Protocols**, the protocol range is **Top N Protocols**, and the **Presentation** mode is **Pie Chart**.

You also can include **Protocol Summary** views in Custom Reports.

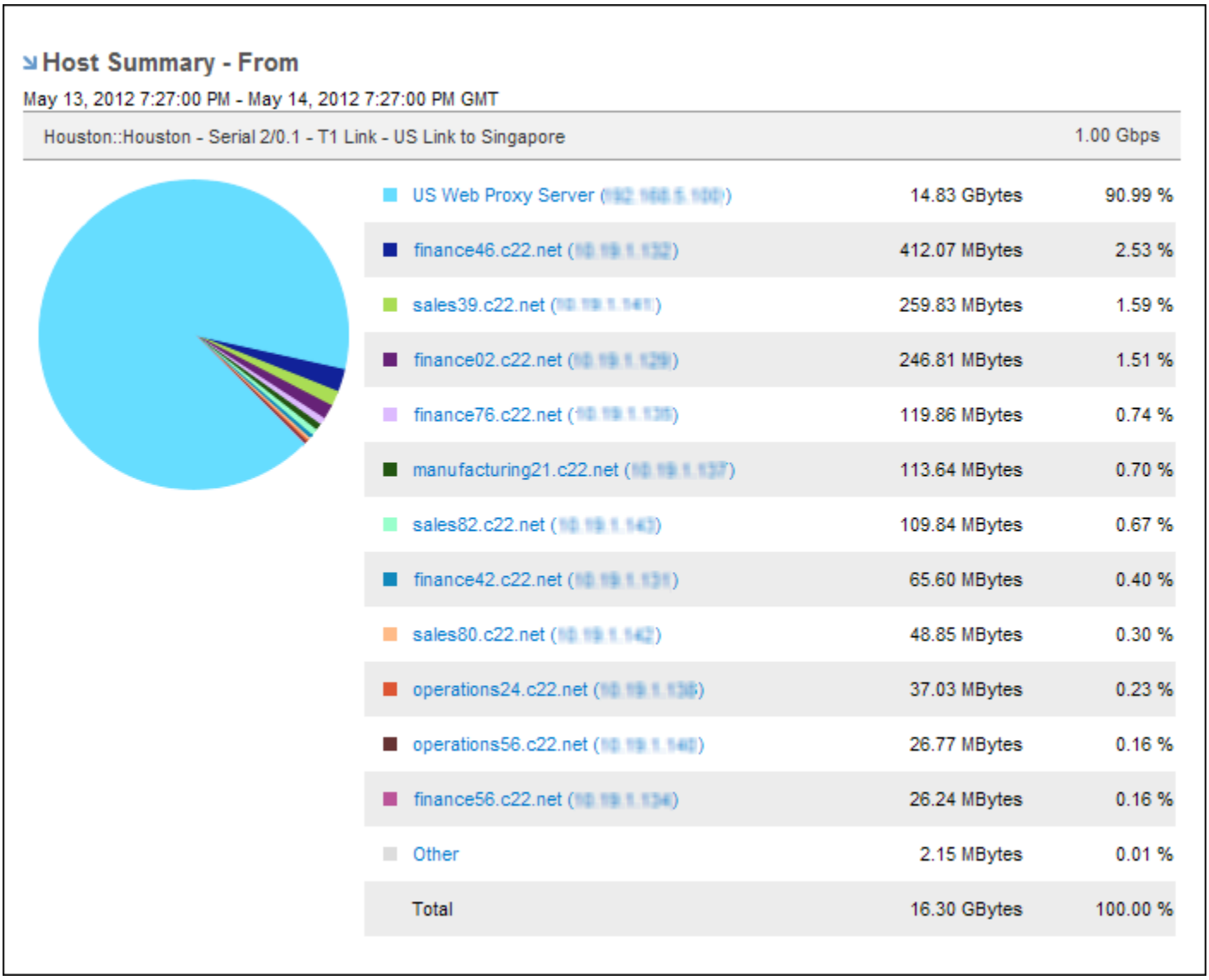
You can display **Top N Protocols** information on **Interface** pages in the following presentation modes:

- Pie charts of protocol summary data on **Overview** reports (inbound and outbound traffic) and **Protocols** report pages (inbound, outbound, and all traffic).
- Stacked trend charts of rate, volume, or utilization data on **Overview** and **Protocols** reports.
- Trend charts of rate, volume, or utilization data on **Protocols** reports.
- Summary tables of rate, volume, or utilization data on **Protocols** reports.

Host Summary Views

The **Host Summary** views can help determine which hosts have the highest traffic volume over the interface during a time period. Access to detailed host information helps pinpoint traffic sources and quickly determine whether traffic is normal. You may be able to resolve a problem quickly because you know which hosts are responsible.

Click the name of a host in a view to display the **Hosts** report with interface data that is specific to the host.



Host Summary views are included on Interface pages for specific interfaces when the selected report type is **Overview** or **Hosts**, the host range is **Top N Hosts**, and the **Presentation** mode is **Pie Chart** or **Mixed Chart**.

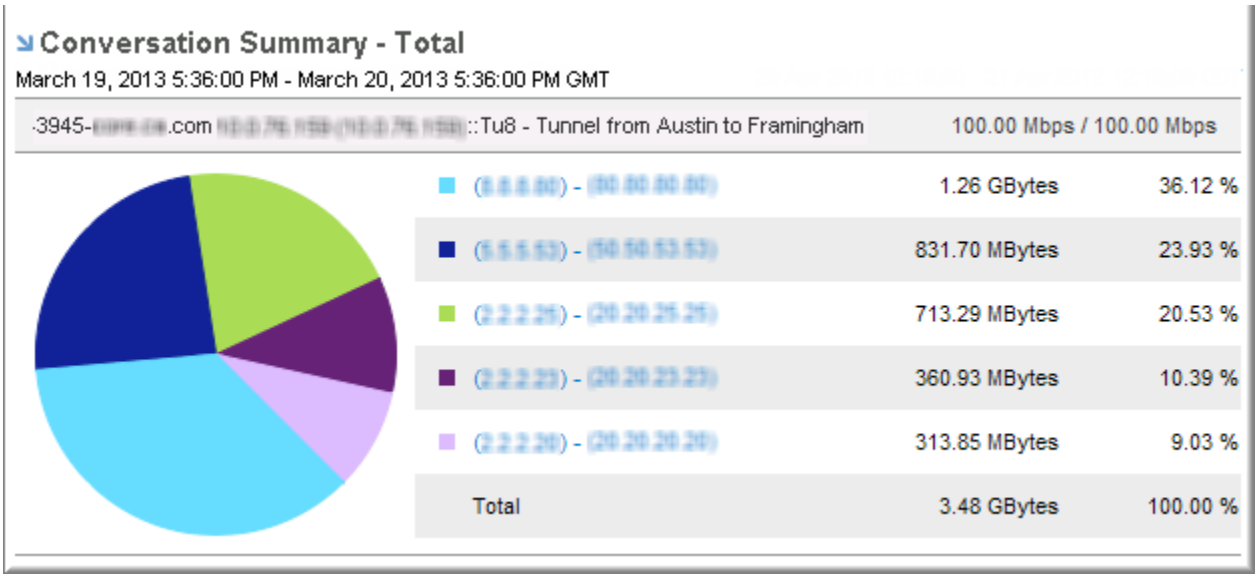
You also can include **Host Summary** views in Custom Reports.

You can display **Top N Hosts** information on **Interface** pages in the following presentation modes:

- Pie charts of host summary data on **Overview** reports (traffic from the host and to the host) and **Hosts** report pages (traffic from the host, traffic to the host, and all traffic).
- Trend charts of rate, volume, or utilization host data on **Overview** or **Hosts** reports.
- Summary tables of rate, volume, or utilization host data on **Hosts** reports. The data in a summary table provides additional data for the hosts, including **Maximum To/From** and **Average To/From** values.

Conversation Summary Views

The data displayed in the **Conversation Summary** views is designed to help you determine the conversations with the most volume of traffic over the interface during the selected time period. Use this information to troubleshoot issues and determine if there is a saturation issue or an area of poor performance.

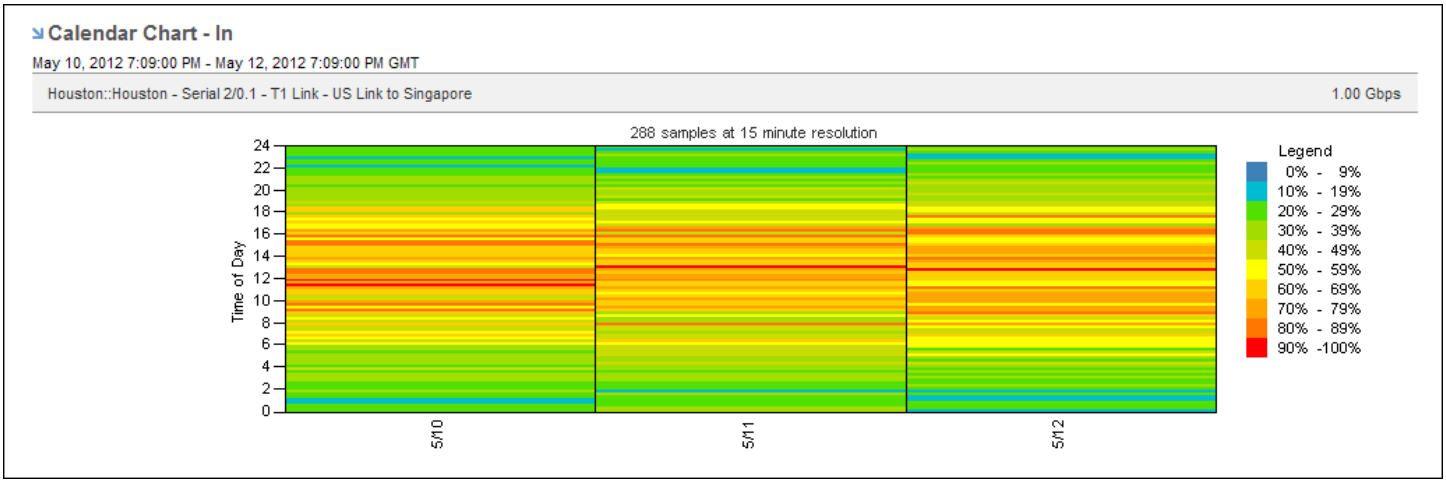


Click the name of a conversation in the view to display the **Conversations** report with interface data specific to that conversation.

These views display data for the top ten conversations on the interface by default. To display data for other conversations, select **Show Other** in the **Presentation** menu. Data for conversations other than the top ten is rolled up and assigned a label of **Other**.

Utilization Calendar Charts

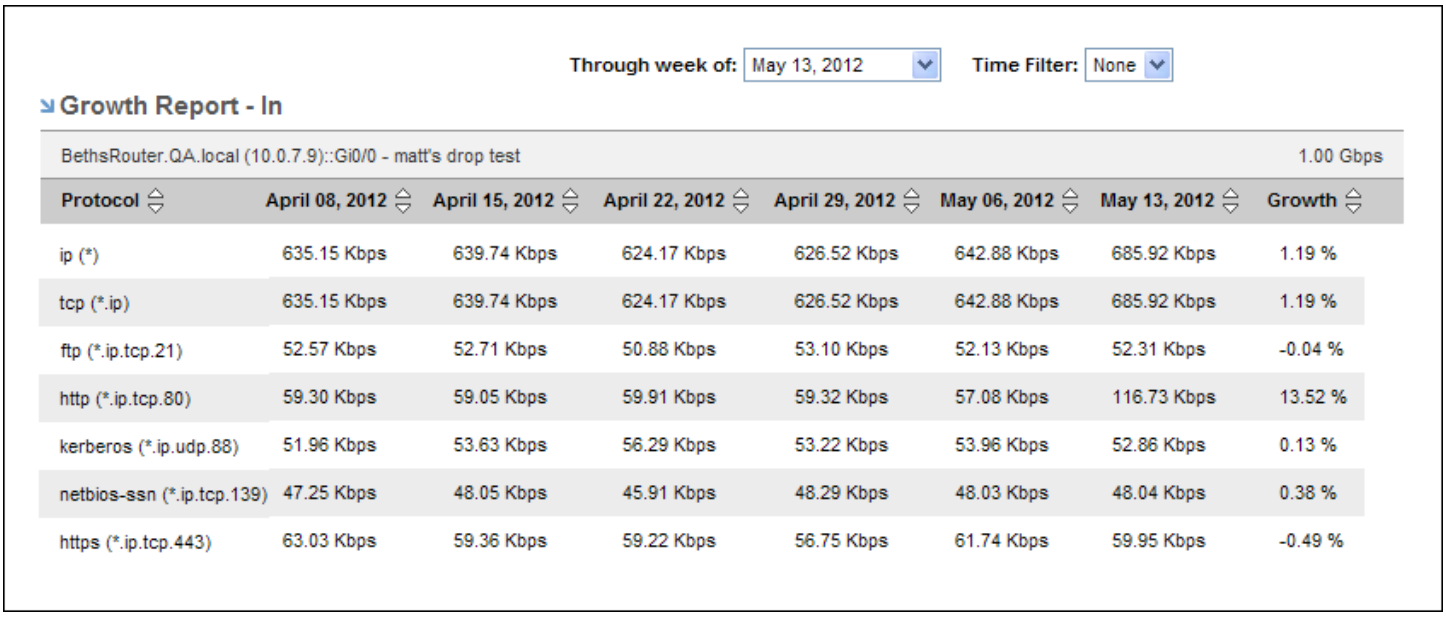
The **Calendar Chart** helps you detect utilization issues on an interface, determine when the problem started, and pinpoint the time of day the problem occurs. In the example chart, you can see that the utilization level for this interface is often at or above 70 percent. The high utilization level means that the performance of the applications sending data over this link is likely to be degraded at those times. Calendar Charts can also reveal patterns from day to day and from week to week so that you can determine whether a problem regularly occurs at a particular time.



Growth Reports

The **Growth Report** includes a specialized summary table. The table helps to identify applications that consume increasing amounts of bandwidth over time. You also can use the table data to determine whether future growth requires

additional capacity. Display data for the previous six weeks or the previous six months and apply any available time filters. You can choose the units to display (such as bits per second) and whether to include inbound traffic, outbound traffic, or all traffic.



Through week of: May 13, 2012 Time Filter: None

Growth Report - In

BethsRouter.QA.local (10.0.7.9)::Gi0/0 - matt's drop test 1.00 Gbps

Protocol	April 08, 2012	April 15, 2012	April 22, 2012	April 29, 2012	May 06, 2012	May 13, 2012	Growth
ip (*)	635.15 Kbps	639.74 Kbps	624.17 Kbps	626.52 Kbps	642.88 Kbps	685.92 Kbps	1.19 %
tcp (*.ip)	635.15 Kbps	639.74 Kbps	624.17 Kbps	626.52 Kbps	642.88 Kbps	685.92 Kbps	1.19 %
ftp (*.ip.tcp.21)	52.57 Kbps	52.71 Kbps	50.88 Kbps	53.10 Kbps	52.13 Kbps	52.31 Kbps	-0.04 %
http (*.ip.tcp.80)	59.30 Kbps	59.05 Kbps	59.91 Kbps	59.32 Kbps	57.08 Kbps	116.73 Kbps	13.52 %
kerberos (*.ip.udp.88)	51.96 Kbps	53.63 Kbps	56.29 Kbps	53.22 Kbps	53.96 Kbps	52.86 Kbps	0.13 %
netbios-ssn (*.ip.tcp.139)	47.25 Kbps	48.05 Kbps	45.91 Kbps	48.29 Kbps	48.03 Kbps	48.04 Kbps	0.38 %
https (*.ip.tcp.443)	63.03 Kbps	59.36 Kbps	59.22 Kbps	56.75 Kbps	61.74 Kbps	59.95 Kbps	-0.49 %

By default, the table is sorted by the total rate of the most recent week. You can sort according to any column by clicking a column heading. You also can change from descending to ascending order by clicking the triangle next to the column heading.

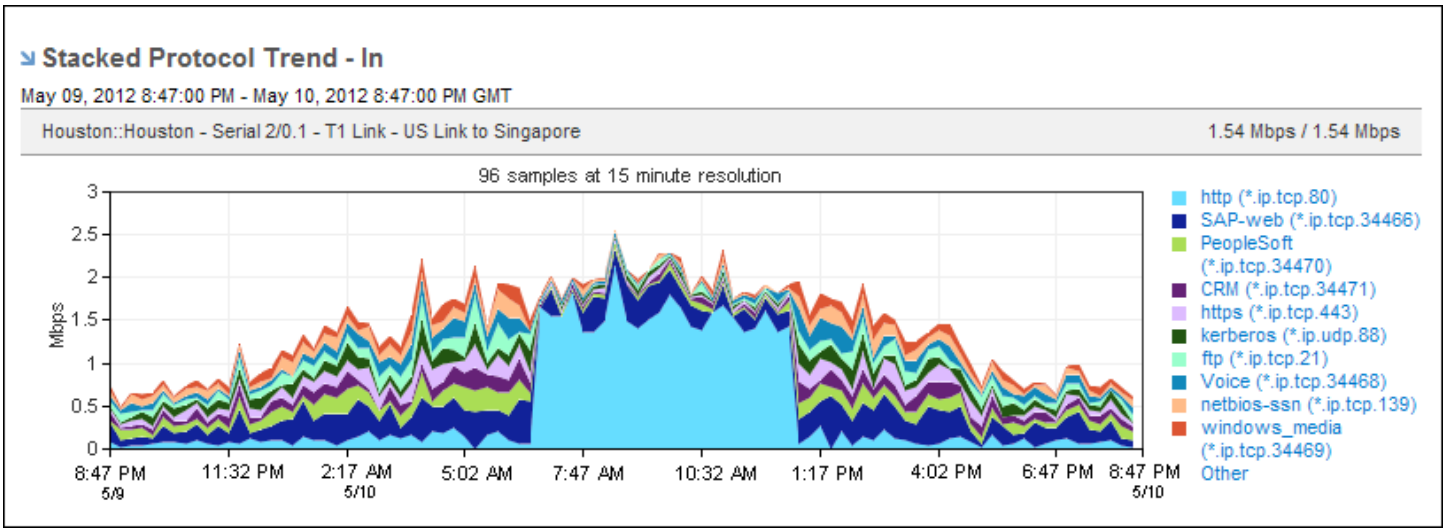
Display Charts and Graphs

When you view information about a specific interface, host, conversation, or protocol, several types of report presentations are available for displaying the data such as charts, graphs, and tables. Each report type is suited to particular data types and situations.

Stacked Trend Charts

The stacked trend plots for Protocol and ToS summary data are excellent for establishing the types of applications that use the most bandwidth for a particular interface. Stacked trend plots also help you compare the use of each application with others.

The legend on the right of the stacked trend chart lists the protocols or ToS values so you can see the amount of data that is transferred for each category. In the example, notice the surge of inbound web (http) traffic.



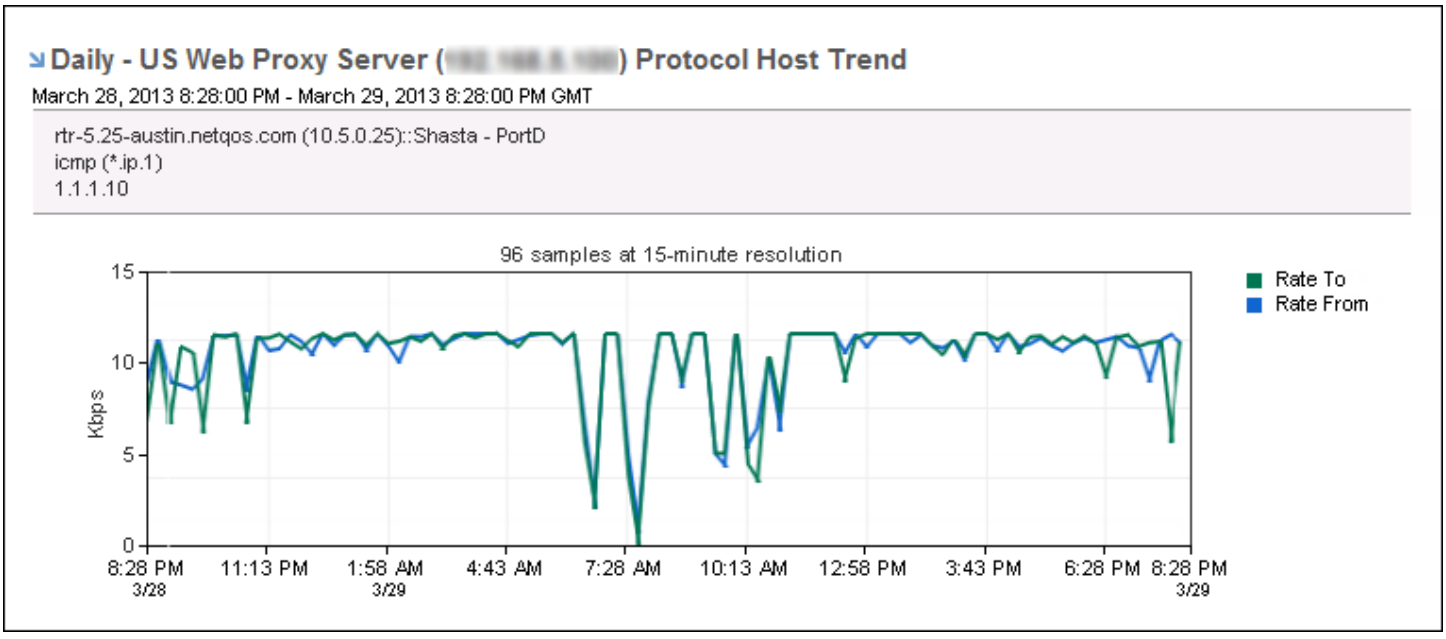
The example report duration is daily. You might want to change the duration to monthly to determine whether the surge is a one-time occurrence or occurs regularly at a particular time. By default, views and reports show the most recent 24 hours of data. To change the time period or apply a time filter, use the **Timeframe** options.

Trend Charts

The trend chart presentation format is available for interface data that is filtered by Protocol, ToS, Hosts, and Conversations. When you choose the trend chart presentation for a **Protocol Summary**, trend plots are shown for each of the top protocols.

Use trend charts to see traffic spikes and dips and to determine whether those patterns are consistent with your expectations for the interface. Click the name of a protocol, host, or conversation to view a more detailed trend chart.

By default, views and reports show the most recent 24 hours of data. To change the time period or apply a time filter, use the **Timeframe** options.



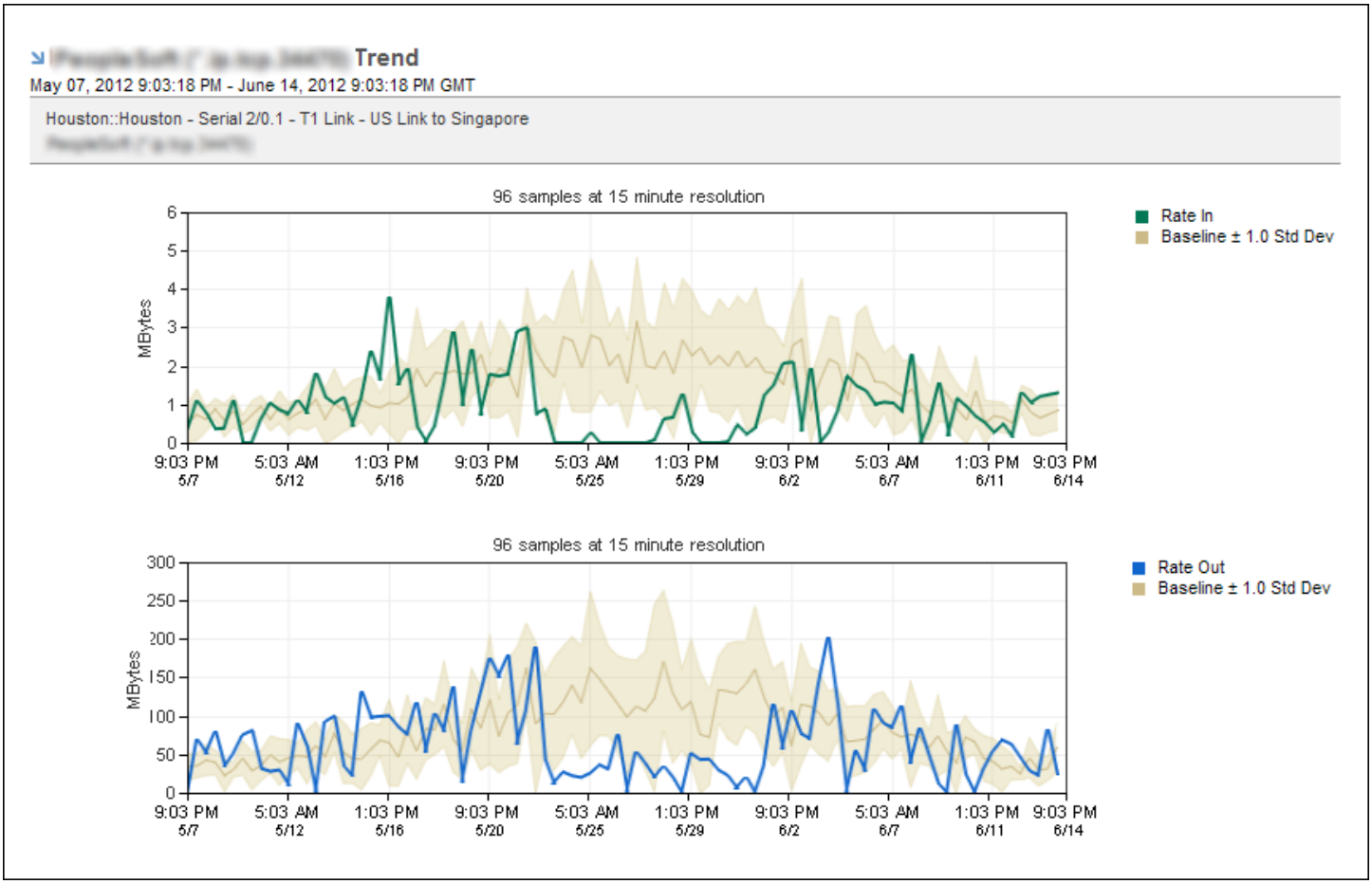
The preceding example shows a trend chart on the **Interface** page with the following settings:

- Report type: Protocols
- Protocol range: Specific protocol
- Report subtype: Hosts
- Hosts range: Top N Hosts
- Presentation type: Trend Chart
- Measurement type: Rate

Baselines

Protocol data views also support baselines for trend plots of some protocol, ToS and flow views. To display baselines, choose from the following settings:

- Protocols report type, Top N Protocols range, Trend Chart presentation type, Show Baselines option selected.
- ToS report type, Top N ToS range, Trend Chart presentation type, Show Baselines option selected.
- Flows report type, Show Baselines option selected.

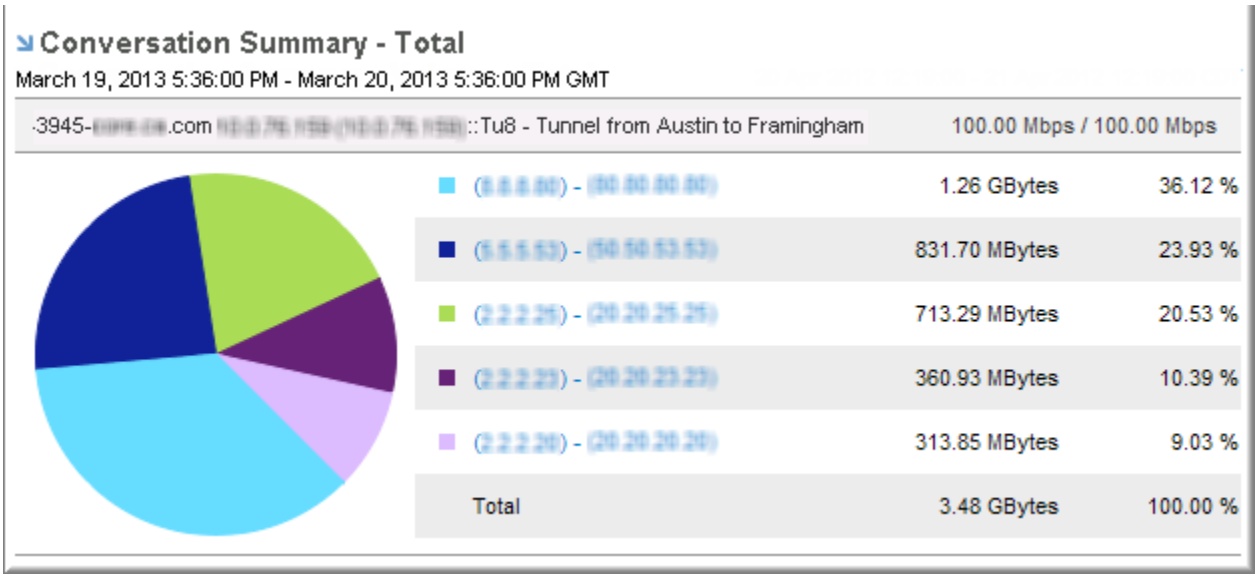


The preceding example shows trend plots on the **Interfaces** page with the following settings:

- Report type: Protocols
- Protocol range: Top N Protocols or a specific protocol with Overview as the report subtype
- Presentation type: Trend Chart or Mixed Chart
- Measurement type: Rate
- Show Baselines: Selected

Pie Charts

Pie charts provide a visual comparison of the protocols, ToS, hosts, or conversations on the interface, making it easy to see which ones use the most or least amount of bandwidth. Pie charts also include a listing of numeric data.

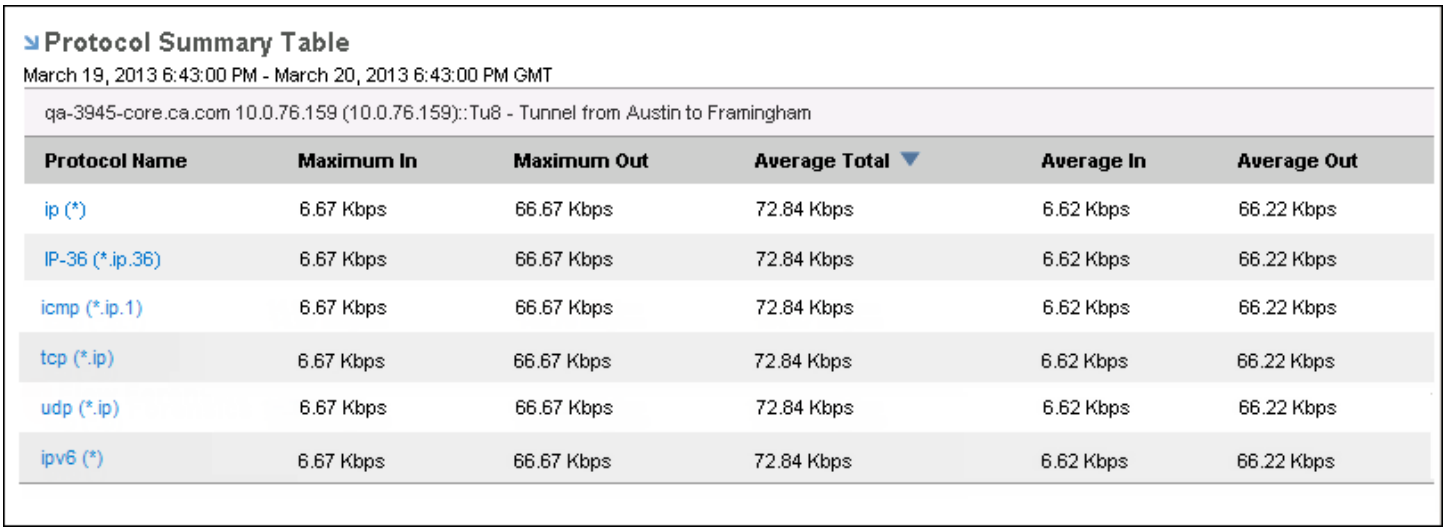


By default, views and reports show the most recent 24 hours of data. To change the time period or apply a time filter, use the Timeframe options.

Summary Tables

A summary table is a good presentation choice when you want Protocol or ToS summary data that can be easily sorted and saved to CSV format for use in a spreadsheet program. Summary tables are available for the following **Top N** report types on **Interface** pages: **Protocols**, **ToS**, **Hosts**, **Conversations**, and **AS Numbers**.

In the following example, the **Protocol Summary Table** lists protocols from the highest to lowest flow volume. You can change the sort order of the list by clicking a column name. Click again to change the order from ascending to descending. The sort column is marked with an arrow.

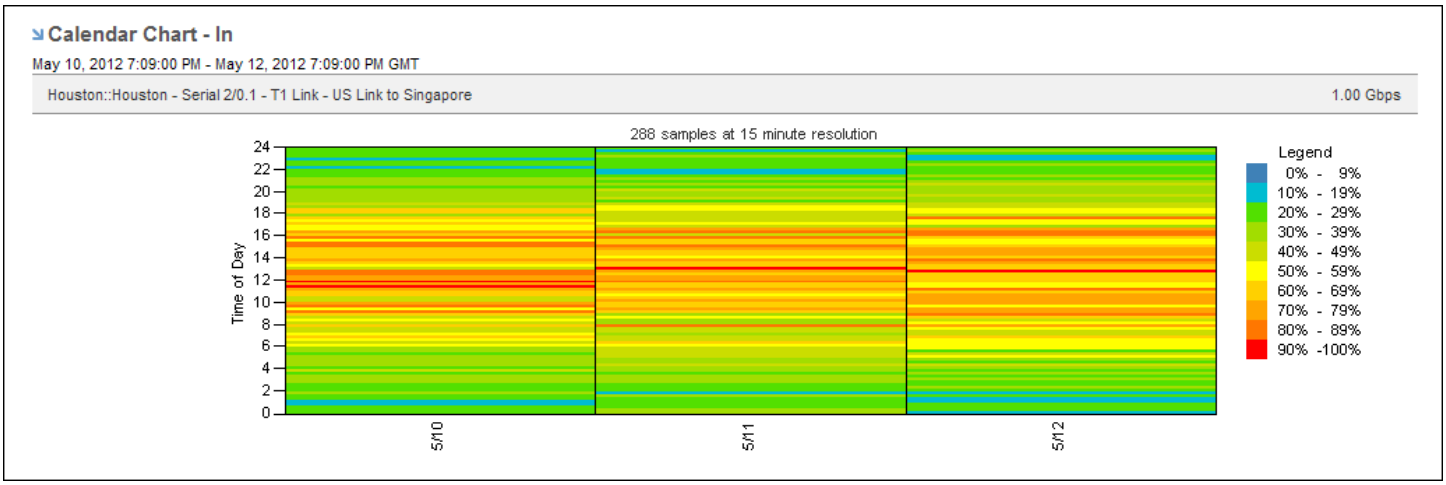


Protocol Name	Maximum In	Maximum Out	Average Total ▼	Average In	Average Out
ip (*)	6.67 Kbps	66.67 Kbps	72.84 Kbps	6.62 Kbps	66.22 Kbps
IP-36 (*.ip.36)	6.67 Kbps	66.67 Kbps	72.84 Kbps	6.62 Kbps	66.22 Kbps
icmp (*.ip.1)	6.67 Kbps	66.67 Kbps	72.84 Kbps	6.62 Kbps	66.22 Kbps
tcp (*.ip)	6.67 Kbps	66.67 Kbps	72.84 Kbps	6.62 Kbps	66.22 Kbps
udp (*.ip)	6.67 Kbps	66.67 Kbps	72.84 Kbps	6.62 Kbps	66.22 Kbps
ipv6 (*)	6.67 Kbps	66.67 Kbps	72.84 Kbps	6.62 Kbps	66.22 Kbps

Calendar Charts

When you choose to view Utilization summary data for an interface, the data is displayed as a calendar chart. You can show inbound, outbound, or total traffic in the chart, choose the month to display, and apply any available time filters.

Excessive utilization of an interface is shown as a block of red extending across the column for the day and rows indicating the time of day. Low utilization is shown in green. Varying shades of each color show the severity degree. The color for each severity range corresponds to a range of utilization values (calculated as percentages of total capacity). The legend explains how the colors are used in the calendar chart.



The calendar chart helps you detect utilization problems with the selected interface, determine when the problem started, and pinpoint the time of day the problem occurs. In the example, the interface utilization level is often at or above 70 percent. The high utilization level means that the performance of the applications sending data over this link is likely to be degraded at those times. Calendar charts can also reveal patterns from day to day and week to week so that you can determine whether a problem occurs regularly at a particular time.

Mixed Display Options

The **Interface Overview** report has additional presentation display types to accommodate a wider array of data--protocol, ToS, host, and conversation data. By using one of the mixed presentation types, you can view the report with the most useful presentation for each report view.

Mixed Chart

When the **Mixed Chart** is selected in the **Presentation** options, the report page displays both stacked trend and pie charts. Protocol and ToS data is presented using stacked trend charts. Host and conversation data is presented using pie charts.

Mixed Trend

When the **Mixed Trend** is selected in the **Presentation** options, the report page displays both stacked trend and trend charts. Protocol and ToS data is presented using stacked trend charts. Host and conversation data is presented using standard trend charts.

Custom Reports

Custom Reports can answer specific technical and business questions in your environment, such as:

- Which applications are used most heavily in the regional offices?
- What is the total volume of traffic for global operations?
- Does the new data center have the capacity to handle additional servers?

To get started, select **Custom Reporting** from the NFA console menu.

The Create New Report Page

The **Create New Report** page lists the existing reports. If you have the required permissions, the page contains functions for creating, managing, and running Custom Reports. You can generate updated versions of reports, add new report definitions, and change report settings.

The **Create New Report** page includes the following functions:

Saved Report Folders:

- **New**: Create a report folder.
- **Rename**: Change the name of a report folder.

Reports:

- **New**: Create a Custom Report definition.
- **Run**: Regenerate the data for one or more Custom Reports.
- **Move to Folder**: Change the parent folder for one or more report definitions.
- **Cancel**: Stop one or more reports from running.

Report Types and Usage

Custom Reports can show a variety of data, including the following types of information:

- *Interface:* View the total volume of traffic that is generated across particular interfaces.
- *Protocol:* Discover which applications are used by various business groups.
- *ToS:* View the distribution of applications by type of service (ToS).
- *Host:* View the activity or usage levels for applications on a server.
- *Conversation:* View the top conversations across interfaces or for a particular protocol.

Report on Network Utilization

Create a custom interface report to select the included interfaces so that you can report on network traffic by:

- Location, such as global, regional, country, state, and individual offices
- Business unit or department, such as Accounting, Marketing, and Sales
- Specific interface groups, such as EMEA, WAN, ATM interfaces, and Ethernet interfaces

Report on Application Distribution

Create a custom protocol report to focus on the applications and protocols in use in your organization. You can generate protocol distribution reports to:

- Show the applications that are used in each region.
- Understand which applications are used the most frequently.
- Determine whether any applications can be removed from the network or de-prioritized.
- Verify that rogue applications are not active on the network.

Report on ToS Distribution

Create a custom ToS report to identify the ToS categories for the most heavily utilized applications in your environment. You can generate ToS reports to:

- Understand which ToS is most widely used in your environment.
- Show the ToS categories in use in each regional location.

Report on Server Activity

Create a custom host report to identify the amount of traffic sent to and from particular servers in your organization. You can generate host reports to:

- Verify that application servers are utilized properly.
- Determine whether a server is vulnerable to any security risks.
- Determine whether a specific data center can handle additional servers.

Set Up Custom Reports

Depending on your user account settings, you may be able to perform a number of actions for Custom Reports:

- View existing reports that have been run.
- Run existing report definitions on demand.
- Set up a schedule for running and sending reports by email (typically for Administrators and Power Users only).
- Create a Custom Report by stepping through the Custom Report wizard.
- Create a Custom Report by using an existing report as a starting point.
- Make changes to the following parts of an existing report definition:
  - Report name, parent folder, and description
  - Report summary types: interface, ToS, protocol, host, and conversation
  - Presentation views used for the summary data
  - Reporting period, data resolution, and time filter
  - Report recurrence schedule and email recipients
- Set the following elements to be included or excluded as report data sources:
  - Interfaces or interface groups
  - Protocols and ToS values
  - A host and subnet
  - A conversation
  - A subnet mask for aggregating data

Scheduled Reports

You can schedule a custom report over a particular duration of time. These duration reports are the only reports which can be scheduled to repeat. You can email a report on a scheduled basis, but the same report is emailed every time unless someone manually runs the report for a different time period.

A scheduled report begins the next scheduled report at the `scheduled_next` value in the database. This value is visible as the **Next run** in the **Reporting Period** when editing a Custom Report. When the report starts this value is set as the **End Time** of the report.

The `scheduled_last` value in the database is updated with this value. This time is the last time the report, which is currently running, was scheduled to run.

Using the report time zone, this time is converted to Unix Time and all further calculations are done in Unix Time. A scheduled report will display the results in GMT regardless of the time zone used to specify the start time for the report.

If a time filter is specified it will alter the data which is displayed but not the duration of the report. The time filter allows the user to create a report that shows data only for a time period of interest. That portion of the reporting period which falls outside the time period of interest will have no data displayed.

A scheduled report can be saved or saved and queued. If the report is saved, it first runs at the next scheduled time for the report. If the report is saved and queued, the first run starts immediately and covers the duration specified in the report, with the end time the point in time when the report was queued, not the scheduled time of the report.

Adding information about the duration, time filters, interfaces, protocols, etc. into the description of the report will help a person reviewing the report understand the data being presented.

NOTE

More information:

Create a Custom Report

Use the Custom Report wizard to create a Custom Report step by step. The wizard guides you to select many options, such as the specific items on which to report, the presentation views, the reporting period, and, optionally, a schedule for running the report automatically.

Follow these steps:

1. Select **Custom Reporting** from the NFA console menu.
2. Click **Create New Report**.
   The **Custom Report** wizard opens and shows the options **Create a new custom report** and **Copy an existing report**.
3. Click **Create a new custom report**, and click **Next**.
   Alternatively, you can select **Copy an existing report** and modify an existing report.
   The **Select Interfaces** page opens.
4. Click one of the following options to select interfaces or interface groups for the report:
   - **Add Interface Filter**: Select one or more individual interfaces from the **Interface Index**.
   - **Add Interface Group Filter**: Select one or more interface groups from the **Interface Group Selection** list.

   The selected interfaces are added to the interface list.
5. Accept the default value or set the Inclusion value for each interface or interface group: **Include** sets the program to use report data from the interface or group. **Exclude** sets the program to bypass data from the interface or group.
6. Click **Next**.
   The **Specify Filters & Rollup** page opens.
7. (Optional) Specify the settings on the **Specify Filters & Rollup** page: Specify filters for gathering or excluding report data:
   - **Add Protocol Filter**: Select individual protocols from the **Protocol Index**.
   - **Add Protocol Group Filter**: Select protocol groups from the **Protocol Group Index**.
   - **Add ToS Filter**: Select individual ToS values from the **ToS Index**.
   - **Add ToS Group Filter**: Select ToS groups from the **ToS Group Index**.
   - **Add Host Filter**: Specify a host IP address and mask.
   - **Add Conversation Filter**: Identify the IP addresses and mask for each party in the conversation pair.
8. Accept the default value or set the Inclusion value for each filter you specified:
   - **Include**: For each filter listed, use only the data of the listed type. For example, use data from the listed protocol group, but not from other protocol groups.
   - **Exclude**: For each filter listed, do not use the data of the listed type. For example, do not use data from the listed protocol group, but do use data from other protocol groups.

9. Select at least one type of summary data to make available for display in the report. Your selections make various report sections available in a later wizard page, but do not require you to include them. Select one or more check boxes in the **Summary Types** section:
   - **Interface Summary**: For example, view the volume of traffic for particular interfaces.
   - **ToS Summary**: For example, view the distribution of applications by type of service (ToS).
   - **Protocol Summary**: For example, view which applications are used by each business group.
   - **Host Summary**: For example, view the application activity or use levels for a server.
   - **Conversation Summary**: For example, view a list of the top conversations across interfaces for a particular protocol.
10. (Optional) Specify a mask for rolling up data by subnet, if you specified a host or conversation filter: Click the **Rollup data using this mask** check box, then select a mask from the list.
11. Click **Next**.
    The **Configure Layout** page opens.
12. (Optional) Add one or more report sections to the report page layout:
    - Select a value from the **Summary Type** list.

      NOTE
      The list of available summary types is limited to the summary types you specified on the previous page.
    - Select a value from the **Presentation** list and the **Measurements** list (if you selected the presentation type as **Table**, **Trend Chart**, or **Stacked Trend Chart**).
    - Click **Add**.

    The new section is added at the end of the report.
13. (Optional) Delete one or more report sections from the report page layout by clicking the X icon next to the section name.
14. (Optional) Re-order the report sections in any of the following ways:
    - Drag and drop the section.
    - Click the Top icon to move the section to the beginning of the report.
    - Delete sections and add them again in the correct order.
15. Click **Next**.
    The **Specify Schedule** page opens.
16. Select the type of reporting period from the **Period** list in the **Specify Schedule** page:
    - **Duration**: Limits the reporting period to an amount of time, ending at the time the report runs. Enter the number of time units in the **Last** box.
      Select a unit of time from the list (days, weeks, months, or years).
      You can set up a schedule for a Duration report or you can run the report on demand.
    - Start and end: Specify a **Start** date and **End** date either by using the calendar icons or by selecting hour and time values from the lists. Hour values are expressed in 24-hour format.
17. Enter the number of time units in the **Resolution** box. Select a value from the list for **Start** and **End**.
    A Start-and-End report runs on demand and cannot be set up to run on a schedule.
18. Accept the default setting or enter the number of time units in the **Resolution** box. Select a unit of time from the list (minutes, hours, days, weeks, months, or years).
19. (Optional) Select a time filter from the list, if your Administrator has created a time filter that is appropriate for your report.

    TIP
    The time filter information does not display on the report as part of the date and time. Give the report a name and description that reflects the selected time filter.
20. (Optional) Select the **Schedule** check box and specify the following options:

- **Schedule**: Select the type from the **Schedule** list (Daily, Weekly, Monthly, Quarterly, Yearly).
- **Daily**: Select the day or days of the week, time of day, and time zone for report generation.
- **Weekly**: Select the day of the week, time of day, and time zone for report generation.
- **Monthly**: Select either the date or the week in the month and day of the week. Select the time of day, and time zone for report generation.
- **Quarterly**: Select a month that ends the first reporting quarter, time of day, and time zone.
- **Yearly**: Select a month that ends the first reporting year, time of day, and time zone.

21. (Optional) Enter the email addresses of all the report recipients in the format *name@domain*. Separate multiple addresses with a comma or semi-colon.

    NOTE
    The options in the **Recurrence** section are available only if you select **duration** as the reporting period type.

22. Click **Next**.
    The **Enter Name** page opens.
23. Identify the report and its location and click **Next**:
    - **Folder**: Accept the default folder or select a different folder to contain the new report.
    - **Name**: Give the new report a name, which appears in the **Reports** list.
    - **Description**: (Optional) Add a description to help identify the report. For example, use the description to identify scheduled reports and to indicate distinguishing features of Duration reports. Include any time filter information, for ease of identification.

    The **Summary & Submit** page opens.
24. Review the information in the **Report Definition Summary**.
    - **Save**: Save the report definition and return to the **Custom Reporting** page. The report will run on the schedule selected.
    - **Save and Queue Report**: Queue the report and return to the **Custom Reporting** page. The report will run now for the data available for the duration selected.

Review Settings for Custom Reports

You can review the **Report Definition Summary** of a Custom Report to ensure that the report definitions are appropriate. From the **Report Definition Summary** page, you can access the **Custom Report Wizard** pages and can modify the report.

Follow these steps:

1. Click the name of the report you want to modify in the **Reports** list on the main **Custom Reporting** page.

   NOTE
   Select a report that is not currently in execution--a report that does not have the status **Running**.
2. Display the **Report Definition Summary** page of the **Custom Report** wizard, if it is not already open. The steps for displaying the **Report Definition Summary** are dependent on the status of the report you select:
   - **Complete**: The report runs and opens. Click the **Edit** button on the left side of the **Report Settings** section to display the **Report Definition Summary**.
   - **Defined**: The **Report Definition Summary** opens automatically.
3. Review the current report settings and click the name of the section you want to modify.
   For example, click **Interface Filters**, **Protocol Filters**, or **ToS Filters** to add new filters or change existing filter settings.
4. The associated page opens, giving you access to the options that you can change.
5. When the changes are complete, click **Queue Report** to regenerate the report with the new settings.
   The **Reports** list opens and the modified report shows a status of **Queued**. When the report has been regenerated and is ready to be viewed, the status displays as **Complete**.
   When you modify a scheduled report, it is queued to be generated at the next scheduled runtime.

Customize Which Interfaces Are in Custom Reports

Before you specify other settings for a new Custom Report, you must add at least one interface or interface group. The **Select Interfaces** page provides options for specifying individual interfaces and interface groups, which you can use to create useful reports quickly.

Select Interfaces and Interface Groups

You can select interfaces from the **Interface Index** or you can select interfaces by group. When you select an interface group, all interfaces in the group and any child groups are included in the report.

> **NOTE**
> Custom groups are defined and managed in CA Performance Center. If DX NetOps is not registered as a data source for CA Performance Center, the only available interface groups are the default groups that are defined by type.

Follow these steps:

1. Select **Custom Reporting** from the NFA console menu if the **Custom Reporting** page is not already open.
2. Put the report that you want to change in editable mode: Click the name of the report. If the report has been run previously, it will run again. In this case, click **Edit** in the **Report Settings** section at the top of the report page.
   The **Report Definition Summary** page opens, which contains links to the Custom Report wizard pages.

   > **NOTE**
   > You can edit a report that appears in the **Reports** list as long as it has a status other than **Running**.
3. (Optional) Click **Add Interface Filter** and select one or more interfaces to include in the report. You can add individual interfaces, interface groups, or both. You must specify some kind of interface filter, however.
   The **Select Interfaces** page of the Custom Report wizard opens.
   - Expand the interface list for the desired router, then select the check box for each interface you want to include in the report.
   - Click **Submit** to add all selected interfaces to the list and close the **Interface Index**.
4. (Optional) Click **Add Interface Group Filter** and select one or more interface groups to include in the report. You can add individual interfaces, interface groups, or both. You must specify some kind of interface filter, however.
   The **Interface Group Selection** page opens.
   - Select check boxes for the groups to add from the **Interface Group Selection** list.
   - Click the blue arrow near the upper right corner to jump to the bottom of the list, then click **Save**.

   You return to the **Select Interfaces** page of the Custom Report wizard. The **Select Interfaces** list now includes the interfaces and interface groups you selected.
5. Check the list of interfaces and interface groups and their Inclusion values.

   > **NOTE**
   > Items that have a value of **Include** are in the report. You can temporarily remove an interface or interface group from the report by setting the **Inclusion** value for that item to **Exclude**.
6. Click **Save Changes**.
   Your changes are saved and you return to the **Report Definition Summary**.

   > **NOTE**
   > To save the report definition, you must have at least one interface set to be included.
7. (Optional) Return to the report list by clicking one of these buttons:
   - **Return to Listing**: Return to the **Custom Reporting** page without queuing the report to run.
   - **Queue Report**: Queue the current report to run and return to the **Custom Reporting** page.

Delete Interfaces and Interface Groups

You can delete filters from Custom Report definitions as needed. When you delete interfaces or interface groups from a report definition, those interfaces are not included in the report data.

Follow these steps:

1. Select **Custom Reporting** from the NFA console menu if the **Custom Reporting** page is not already open.
2. Put the report in editable mode: Click the name of the report. If the report has been run previously, it runs again. In this case, click **Edit** in the **Report Settings** section at the top of the report page.
   The **Report Definition Summary** page opens, which contains links to the Custom Report wizard pages.
3. Click **Interface Filters** to open the **Select Interfaces** page of the Custom Report wizard.
4. Select the check box in the interface list next to each interface you want to delete.
5. Click the **Remove Selected Filters** icon that is located above the check boxes.
   
6. Click **Save Changes**.
   Your changes are saved and you return to the **Report Definition Summary**.

   NOTE
   If you select a scheduled Custom Report that uses a deleted interface or interface group, the "Unknown interface group" message is displayed. If all the associated interfaces or interface groups are deleted, the report will not run.
7. (Optional) Click one of the following buttons to return to the report list:
   - **Return to Listing**: Return to the report list on the **Custom Reporting** page.
   - **Queue Report**: Queue the report to run and return to the **Custom Reporting** page.

Exclude Interfaces or Interface Groups Temporarily

To remove interfaces from a report temporarily, set the Include value for the interfaces to Exclude. For example, suppose that part of your network is temporarily offline and the offline interfaces distort the report data. You can exclude the interfaces, then change the value back to ‘Include’ when the interfaces are online again.

Follow these steps:

1. Select **Custom Reporting** from the NFA console menu if the **Custom Reporting** page is not already open.
2. Put the report in editable mode: Click the name of the report. If the report has been run previously, it runs again. In this case, click **Edit** in the **Report Settings** section at the top of the report page.
   The **Report Definition Summary** page opens, which contains links to the Custom Report wizard pages.
3. Click **Interface Filters** to open the **Select Interfaces** page of the Custom Report wizard.
4. In the **Select Interfaces** list, set the **Inclusion** option to **Exclude**.
5. Click **Save Changes**.
   Your changes are saved and you return to the **Report Definition Summary**.
6. (Optional) Check your changes in the summary, then click one of the following buttons to return to the report list:
   - **Return to Listing**: Return to the report list on the **Custom Reporting** page.
   - **Queue Report**: Queue the report to run and return to the **Custom Reporting** page.

Specify Custom Report Filters

Use Custom Report filters to determine the type of data to include, such as protocol, ToS, host, and conversation data.

You can use various filters to limit the data in a Custom Report, but not all filter combinations are valid. For example, you can filter report data to show either hosts or conversations, but not both.

Valid filter combinations are shown in the list that follows. The filter or combination of filters you apply to a report determine which summary types are available, which are also included in the list:

- *No filters:* Interface, ToS, Protocol, Host, and Conversation summaries
- *Protocol filters:* Interface, ToS, Protocol, Host, and Conversation summaries
- *ToS filters:* Interface, ToS, Protocol, Host, and Conversation summaries
- *Protocol and ToS filters:* Interface, ToS, and Protocol summaries
- *Protocol and host filters:* Interface, Protocol, Host, and Conversation summaries
- *Protocol and conversation filters:* Interface, Protocol, Host, and Conversation summaries
- *ToS and conversation filters:* Interface, ToS, Host, and Conversation summaries
- *ToS and host filters:* Interface, ToS, Host, and Conversation summaries
- *Conversation filters:* Interface, ToS, Protocol, Host, and Conversation summaries
- *Host filters:* Interface, ToS, Protocol, Host, and Conversation summaries

Rules for Utilization Measurements

The following rules apply to utilization measurements:

- Inbound and outbound interface speeds must be set.
- Utilization is applicable only for interface, ToS, and protocol views and for data that has no host or conversation filter applied.
- Protocol or ToS utilization is limited to a single interface.
- Total utilization is not available.

Add, Delete, or Change Protocol Filters

You can add a protocol filter to include report data for the protocols defined in your DX NetOps installation.

You can add a protocol group filter to restrict reported data to one or more protocol groups. Protocol groups are useful for streamlining Custom Report definitions. For example, protocol groups can help users easily report on network traffic that custom applications generate. The Administrator can create a group for each of the applications that includes the range of ports that are used for the application. Users can also choose from the default protocol groups that are provided automatically.

Follow these steps:

1. Select **Custom Reporting** from the NFA console menu if the **Custom Reporting** page is not already open.
2. Put the report in editable mode: Click the name of the report. If the report has been run previously, it runs again. In this case, click **Edit** in the **Report Settings** section at the top of the report page.
   The **Report Definition Summary** page opens, which contains links to the Custom Report wizard pages.
3. Click **Protocol Filters** to open the **Specify Filters & Rollup** page of the Custom Report wizard.
4. (Optional) Select one or more individual protocols to include: Click **Add Protocol Filter**
   The **Protocol Index** dialog opens.
   - (Optional) Select the display mode for the protocols from the **Select by** list:
     - **Protocol Name**: View the protocols groups by name, either divided into alphabetized lists or displayed together in the **All** list (Default setting).
     - **Port Number**: View the protocols in groups by port number. Click the arrows to expand the contents of individual port number groups or click **Expand All** to show the contents of all groups.
     - **Show**: Set the view to include TCP Ports, UDP Ports, or TCP & UDP Ports, when the **Select by** is **Port Number**.
   - Locate and select the protocols in the protocol lists.

To use a protocol that is not listed, add the protocol to the list: Click **Add Protocol**, then specify the new protocol name, port, type, and description.
   - Click **Submit**. To locate the **Submit** button quickly, click the blue arrow to jump to the bottom of the page.
5. (Optional) Select one or more protocol groups to include: Click **Add Protocol Group Filter**
   The **Protocol Group Index** dialog opens.
   - Select check boxes for the protocol groups to add from the list.
   - Click **Submit**. To locate the **Submit** button quickly, click the blue arrow to jump to the bottom of the page.

   You return to the **Select Filters & Rollup** page of the Custom Report wizard.
6. Check the list of filters and their **Inclusion** values.
   Items that have a value of **Include** are in the report. You can remove data for a protocol or protocol group temporarily by setting the Inclusion value for the item to **Exclude**.
7. Click **Save Changes**.
   Your changes are saved and you return to the **Report Definition Summary**.
8. (Optional) Return to the report list by clicking one of these buttons:
   - **Return to Listing**: Return to the **Custom Reporting** page without queuing the report to run.
   - **Queue Report**: Queue the report to run and return to the **Custom Reporting** page.

Add or Modify ToS Filters

You can add ToS filters or ToS group filters to restrict report data. You can filter for individual ToS values, use any ToS groups your Administrator has created, or use the default ToS group, All ToS.

ToS groups are useful for creating Custom Reports for specific classes of applications. For example, suppose that you want to watch applications that your IT department has classified as Gold Class applications. To facilitate reports on Gold Class applications, the Administrator creates a ToS group that includes the ToS values for those applications.

Follow these steps:

1. Select **Custom Reporting** from the NFA console menu if the **Custom Reporting** page is not already open.
2. Put the report in editable mode: Click the name of the report. If the report has been run previously, it runs again. In this case, click **Edit** in the **Report Settings** section at the top of the report page.
   The **Report Definition Summary** page opens, which contains links to the Custom Report wizard pages.
3. Click **ToS Filters**.
4. The **Specify Filters & Rollup** page of the Custom Report wizard opens.
5. (Optional) Select one or more individual ToS values to include: Click **Add ToS Filter**.
   The **Protocol Index** dialog opens.
   - Select the check box for each ToS value you want to include.
   - Click **Submit**. To locate the **Submit** button quickly, click the blue arrow to jump to the bottom of the page.
6. (Optional) Select one or more ToS groups to include: Click **Add ToS Group Filter**.
   The **ToS Group Index** dialog opens.
   - Select check boxes for the ToS groups you want to add.
   - Click **Save**.

   You return to the **Select Filters & Rollup** page of the Custom Report wizard.
7. Check the list of filters and their Inclusion values.
   Items that have a value of **Include** are in the report. You can remove data for a ToS value or ToS group temporarily by setting the Inclusion value for that item to **Exclude**.
8. Click **Save Changes**.
   Your changes are saved and you return to the **Report Definition Summary**.
9. (Optional) Return to the report list by clicking one of these buttons:
   - **Return to Listing**: Return to the **Custom Reporting** page without queuing the report to run.
   - **Queue Report**: Queue the report to run and return to the **Custom Reporting** page.

Add or Modify Host and Conversation Filters

You can add host or conversation filters to limit the report data to specific hosts or conversations, but you cannot add both host and conversation filters. You can apply a single type of filter or you can use one of the following valid filter combinations:

- Protocols/Protocol Groups + ToS/ToS Groups
- Protocols/Protocol Groups + Hosts
- Protocols/Protocol Groups + Conversations
- ToS/ToS Groups + Hosts
- ToS/ToS Groups + Conversations

Add or Modify Host Filters

Adding host filters to a report changes what table you are querying, switching from the `protocol_traffic` table to the `host_traffic` table. The `host_traffic` table has two entries per interface per flow: source/destination for both in and out interface, with perspective (in/out) set appropriately. So, the `host_traffic` table contains twice as many records as the `protocol_traffic` table. Any data that is returned appears up to twice as big because the data is host-oriented.

Follow these steps:

1. Select **Custom Reporting** from the NFA console menu if the **Custom Reporting** page is not already open.
2. Put the report in editable mode: Click the name of the report. If the report has been run previously, it runs again. In this case, click **Edit** in the **Report Settings** section at the top of the report page.
   The **Report Definition Summary** page opens, which contains links to the Custom Report wizard pages.
3. Click **Host/Conversation Filters** to open the **Specify Filters & Rollup** page of the Custom Report wizard.
4. Click **Add Host Filter**.
   The **Apply Host Filter** dialog opens.
5. Specify the following values, then click **Submit**.
   - **Host/Network IP**: Enter the IP address of the host network whose data will be included.
   - **Mask**: Select a mask from the list.

   For example, enter 192.168.1.0 with a mask of 255.255.255.0 to include data from all hosts on the class C network 192.168.1.0. The report will not include any data from hosts at other network addresses.
   You return to the **Select Filters & Rollup** page of the Custom Report wizard.
6. Check the list of filters and their Inclusion values.
   If you specify a single host filter with a value of **Include**, the report contains data only for that host. If you set a host filter to have a value of **Exclude**, the report contains data for all other hosts in the selected set of interfaces.
7. Click **Save Changes**.
   Your changes are saved and you return to the **Report Definition Summary**.
8. (Optional) Return to the report list by clicking one of these buttons:
   - **Return to Listing**: Return to the **Custom Reporting** page without queuing the report to run.
   - **Queue Report**: Queue the report to run and return to the **Custom Reporting** page.

Add or Modify Conversation Filters

Follow these steps:

1. Select **Custom Reporting** from the NFA console menu if the **Custom Reporting** page is not already open.
2. Put the report in editable mode: Click the name of the report. If the report has been run previously, it runs again. In this case, click **Edit** in the **Report Settings** section at the top of the report page.
   The **Report Definition Summary** page opens, which contains links to the Custom Report wizard pages.
3. Click **Host/Conversation Filters** to open the **Specify Filters & Rollup** page of the Custom Report wizard.
4. Click **Add Conversation Filter**.

The **Apply Conversation Filter** dialog opens.

5. Specify the following values for each host in the conversation pair, then click **Submit**.
   - **Host/Network IP**: Enter the IP address of each host network in the conversation pair in the Host/Network IP boxes.
   - Select a mask from the list to the right of the Host/Network IP value.

   For example, specify the two hosts in the conversation by entering 192.168.1.1 with a mask of 255.255.255.225 and 192.168.1.0 with a mask of 255.255.255.0. The report will include conversation data between all hosts on the 192.168.1.1 network and the 192.168.1.0 network.
   You return to the **Select Filters & Rollup** page of the Custom Report wizard.
6. Check the list of filters and their Inclusion values.
   If you specify a single conversation filter with a value of **Include**, the report contains data only for that conversation.
   If you set the conversation filter to **Exclude**, the report contains all data for other conversations in the selected set of interfaces.
7. Click **Save Changes**.
   Your changes are saved and you return to the **Report Definition Summary**.
8. (Optional) Return to the report list by clicking one of these buttons:
   - **Return to Listing**: Return to the **Custom Reporting** page without queuing the report to run.
   - **Queue Report**: Queue the report to run and return to the **Custom Reporting** page.

Delete Custom Report Filters

You can delete filters from Custom Reports to reflect the deletion of protocols, hosts, or TOS values from your network. Any filter you delete is no longer used to define the report data.

Follow these steps:

1. Select **Custom Reporting** from the NFA console menu if the **Custom Reporting** page is not already open.
2. Put the report in editable mode: Click the name of the report. If the report has been run previously, it runs again. In this case, click **Edit** in the **Report Settings** section at the top of the report page.
   The **Report Definition Summary** page opens, which contains links to the Custom Report wizard pages.
3. Click **Host/Conversation Filters** to open the **Specify Filters & Rollup** page of the Custom Report wizard.
4. Select the check box next to each filter that you want to delete.
5. Click the **Remove Selected Filters** icon at the top-left corner of the list.
6. Click **Save Changes**.
   Your changes are saved and you return to the **Report Definition Summary**.
7. (Optional) Return to the report list by clicking one of these buttons:
   - **Return to Listing**: Return to the **Custom Reporting** page without queuing the report to run.
   - **Queue Report**: Queue the report to run and return to the **Custom Reporting** page.

Exclude Custom Report Filters

For each filter you define in a Custom Report, you can opt to include either:

- Only the data that matches the filter criteria
- All the data that does not match the filter criteria

To set the type of filtering action, you set the **Inclusion** value for each filter in the **Specify Filters** list. To display matching data, set the Inclusion value to **Include** (the default setting). To display data that does not match, set the **Inclusion** value to **Exclude**.

Follow these steps:

1. Select **Custom Reporting** from the NFA console menu if the **Custom Reporting** page is not already open.

2. Put the report in editable mode: Click the name of the report. If the report has been run previously, it runs again. In this case, click **Edit** in the **Report Settings** section at the top of the report page.
   The **Report Definition Summary** page opens, which contains links to the Custom Report wizard pages.
3. Click **Host/Conversation Filters** to open the **Specify Filters & Rollup** page of the Custom Report wizard.
4. Locate the filter whose data you want to exclude and select **Exclude** from the **Inclusion** list.
5. Click **Save Changes**.
   Your changes are saved and you return to the **Report Definition Summary**.
6. (Optional) Return to the report list by clicking one of these buttons:
   – **Return to Listing**: Return to the **Custom Reporting** page without queuing the report to run.
   – **Queue Report**: Queue the report to run and return to the **Custom Reporting** page.

Define Custom Report Periods and Schedules

When you define a Custom Report, you must specify the reporting time period and the resolution of the reporting data. You can define a specific, nonrecurring time period (for a start-and-end report) or you can choose a timespan that ends at the report runtime (for a duration report). You also have the option to set a duration report to regenerate on a recurring schedule and to have the automated reports sent out by email.

Specify a Reporting Period for Custom Reports

The Custom Report Wizard provides a **Specify Schedule** page that defines the report time period. For a manually generated report, you can simply select a time period, resolution, and optional time filter. If you want to generate the report automatically at scheduled intervals, you can also use the Schedule option.

Follow these steps:

1. Select **Custom Reporting** from the NFA console menu if the **Custom Reporting** page is not already open.
2. Put the report in editable mode: Click the name of the report. In the **Report Settings** section at the top of the report page, click **Edit**.
   The **Report Definition Summary** page opens, which contains links to the **Custom Report Wizard** pages.
3. Click **Reporting Period** to open the **Specify Schedule** page of the **Custom Report Wizard**.

   NOTE
   Custom Reports display data using the user time zone setting. The time zone selected when scheduling the report is used as the time zone to report against, but the resulting report displays the user setting.
4. Select the type of reporting period from the **Period** list on the **Specify Schedule** page:
   – **duration**: Include data from the block of time that immediately precedes the report runtime. Enter a number of days, weeks, months, or years.
   You can set up a schedule for a duration report or you can run the report on demand. To run a scheduled report on demand, you can make a copy of the report and can disable the schedule in the copy.
   – **start & end**: Include data from a specific, nonrecurring timespan. Use the calendars to specify a **Start** date and **End** date or select hour and time values from the lists.
   Hour values are expressed in 24-hour format.
   A start-and-end report runs on demand. You cannot set up a schedule for a start-and-end report.
5. Set the resolution (granularity for data collection) on the **Specify Schedule** page: Accept the default setting or enter the number of time units in the **Resolution** box. Select a unit of time from the list (minutes, hours, days, weeks, months, or years).
   The default resolution varies depending on the length of the reporting period. For example, if you specify a duration of one month for the period, the default resolution is 8 hours. If you specify a duration of one day, the default resolution is 15 minutes.
6. (Optional) Select a time filter from the list on the **Specify Schedule** page, if your Administrator has created an appropriate time filter.
7. Click **Save Changes**.

Your changes are saved and you return to the **Report Definition Summary**.

8. (Optional) Return to the report list by clicking one of these buttons:
   - **Return to Listing**: Return to the **Custom Reporting** page without queuing the report to run.
   - **Queue Report**: Queue the report to run and return to the **Custom Reporting** page.

Specify Schedules for Auto-Generated Custom Reports

Use the **Schedule** option to set the Custom report to regenerate at specified times.

For example, suppose that you want to check network traffic during the monthly backups that occur on the last Sunday of each month. You schedule a report to be regenerated on the last Sunday of every month. Suppose that operating system updates occur on the 15th of every month. To check the network traffic during those updates, you schedule a report to be regenerated on the 15th of each month.

> **NOTE**
> The options in the **Recurrence** section are available only if you select **duration** as the reporting period type.

Follow these steps:

1. Select **Custom Reporting** from the NFA console menu if the **Custom Reporting** page is not already open.
2. Put the report in editable mode: Click the name of the report. In the **Report Settings** section at the top of the report page, click **Edit**.
   The **Report Definition Summary** page opens, which contains links to the **Custom Report Wizard** pages.
3. Click **Reporting Period** to open the **Specify Schedule** page of the **Custom Report Wizard**.
4. Click the **Schedule** check box and choose a recurrence interval from the list:
   - **Daily**: Select the day or days of the week the report will run.
   - **Weekly**: Select the day of the week the report will run.
   - **Monthly**: Select either a date or a week and day combination to specify the one day per month the report will run.
   - **Quarterly**: Select the month that ends the first quarter in which the report will run. Starting with the specified quarter, the report will run on the last day of each quarter.
   - **Yearly**: Select the month that ends the first year the report will run. The report runs on the last day of the year.

   For all schedule interval types, select the time of day and time zone for the report to run.
5. (Optional) **Email Results To**: Enter the email addresses of anyone who should receive the report by email. Use the format *name@domain*. Separate multiple addresses with a comma or semi-colon.
6. Click **Save Changes**.
   Your changes are saved and you return to the **Report Definition Summary**.
7. (Optional) Return to the report list by clicking one of these buttons:
   - **Return to Listing**: Return to the **Custom Reporting** page without queuing the report to run.
   - **Queue Report**: Queue the report to run and return to the **Custom Reporting** page.

View Custom Reports

When you have defined the Custom Reports that you need, you are ready to generate a new report and view it in the NFA console. You can run and view the following types of reports:

- Unscheduled duration reports on demand.
- The most recent versions of scheduled duration reports that have been generated automatically.
- The most recent versions of start-and-end reports

Follow these steps:

1. Select **Custom Reporting** from the NFA console menu if the **Custom Reporting** page is not already open.
   The **Custom Reporting** page contains two panes. The left pane lists the folders that store saved reports. The right pane lists the reports in the currently selected folder.

2. Click the name of the folder that contains the report you want to view.
3. Click the report name in the **Reports** list.
   The report runs. The results are displayed when report generation is complete.
4. (Optional) Change the type of presentation by selecting an option from the **Report Type** list.
   The default presentation type is **Custom Layout**. The other available options are dependent on the filters that are defined for your report.
   When you select an option other than **Custom Layout**, a **Presentation** menu is added on the left side of the page. You can select from the presentation types and from the data types to modify your view. For example, you could change from a pie chart to a stacked trend chart that shows volume.

Flow Forensics Reports

Flow Forensics reports let you drill down to raw data flows and see a level of detail that is not available in other reports. You can jump to detailed information about any of the fields in a data packet for any monitored interface.

You can run a Flow Forensics report to drill down and view raw flow data. The data is parsed into meaningful reports.

Use Flow Forensics to browse raw flow data for a specified time period. You can filter the results by using a number of fields. You can export the displayed data to a file in comma-separated value (CSV) format.

Flow Forensics reports let you report on all of the flow data that is collected in your environment. You can analyze every protocol, host, and conversation on your network. Other report types are designed to show the most active interfaces, protocols, hosts, and conversations or to show individual instances of these items.

Flow Forensics reports can provide a comprehensive analysis of all traffic for the following categories, for example:

- Protocols that are active for one or more specified hosts
- All protocols that are active on the network
- All hosts that access one or more specified hosts
- Volume of traffic to or from one or more specified hosts

NOTE

More information:

Flow Forensics Report Types

The following topics describe the available Flow Forensics reports.

Address Report Group

The **Address** Flow Forensics reports have the following fields.

Report	Src Addr	Dest Addr	Bytes	Rate (Bits)	% Total (Bytes)	Flows	Pkts	Rate (Pkts)	% Total (Pkts)	# of Src Addr	# of Dest Addr	IP Protocol	Dest Port
Address Pairs	Y	Y	Y	Y	Y	Y	Y	Y	Y				
Destination Address Peer Count		Y				Y				Y			

Destination Addresses		Y	Y	Y	Y	Y	Y	Y	Y				
Source Address Peer Count	Y										Y		
Source Address Peer Count with Destination Port	Y					Y					Y	Y	Y
Source Addresses	Y		Y	Y	Y	Y	Y	Y	Y				

- **Address Pairs Report**
  Displays the following information about pairs of IP addresses that exchanged traffic:
  - IP addresses of the source host and destination host in the conversation
  - Data volume and rate, shown in bytes (or kilobytes, megabytes) and packets
  - Percentage of total traffic that the data represents, shown in bytes (or kilobytes, megabytes) and packets
- **Destination Address Peer Count Report**
  Displays the following information about each traffic destination:
  - Destination address
  - Number of unique source addresses that exchanged traffic with the destination host
  - Flow count
- **Destination Addresses Report**
  Displays the volume, rate, and percent of total inbound bytes/packets, and the flow count of each traffic destination.
- **Source Address Peer Count Report**
  Displays the following information about each traffic source:
  - Source address
  - Number of destination addresses that received traffic from the source host
  - Flow count
- **Source Address Peer Count with Destination Port Report**
  Displays the following information about each traffic source:
  - Source address
  - Destination port
  - IP protocol
  - Number of destination addresses that received traffic from the source host
  - Flow count
- **Source Addresses Report**
  Displays the volume, rate, and percent of total inbound bytes/packets, and the flow count of each source address.

Application Response Time Report Group

The **Application Response Time** reports show Cisco's Application Response Time (ART) metrics. ART metrics are available for IPFIX flows from routers that have Cisco ART metrics enabled.

Meaningful data is shown in the **Application Response Time** reports under the following conditions:

- The routers and interfaces in the report are configured to export IPFIX flow.
- The exported flow includes the appropriate fields.
  If a router is not configured to return data for the appropriate fields, a zero (0) appears in the corresponding table cells. Non-NBAR2 traffic shows a zero for all of the values except the router IP address and application name.
  The **Application** column values show the NBAR2 application name followed by the application ID. The application name is included if it is defined by the standard (not custom) NBAR2 engine. Otherwise the **Application** value may consist of only the application ID.

The **Application Response Time** reports have the following fields.

Report	Router Addr	Application	Transactions	Avg Total Transaction Time	Late Responses	Retransmissions	Client Address	Server Address	New Connections	Avg Client Network Delay	Avg Server Network Delay	Avg Response Time	Avg Application Delay
Application Metrics	Y	Y	Y	Y	Y	Y							
Client Side Metrics	Y	Y					Y	Y	Y	Y			
Server Side Metrics	Y	Y							Y		Y	Y	Y

- **Application Metrics Report**
  Displays a summary of ART metrics for IPFIX flow that includes NBAR2 data. The report table includes a row for each unique combination of the following values:
  – IP address of the router that sent the data
  – NBAR2 application name
  – Number of transactions
  – Number of late responses
  – Number of times that lost packets were retransmitted
  – Average of the total transaction time, which is calculated from the flow record data
- **Client Side Metrics Report**
  Displays client-side ART metrics for IPFIX flow that includes NBAR2 data. Each row shows the values for a unique combination of the following values:
  – IP address of the router that sent the data
  – NBAR2 application name
  – IP addresses of the client and server
  – Number of new connections
  – Average client network delay
- **Server Side Metrics Report**
  Displays server-side ART metrics for IPFIX flow that includes NBAR2 data. Each row shows the values for a unique combination of the following values:
  – IP address of the router that sent the data
  – NBAR2 application name
  – Number of new connections
  – Average server network delay
  – Average response time (includes the network time from the client to the server and the server response time)
  – Average application delay

ICMP Report Group

The **ICMP** Flow Forensics reports have the following fields.

Report	Src Addr	Dest Addr	Dest Network	# of Src Addr	# of Dest Addr	ICMP	Pkts	Flows
Bad IP Headers	Y							Y
Fragmentation Required and DF Flag Set		Y		Y				Y
ICMP Traffic Summary						Y		Y
Ping Conversation Pairs	Y	Y					Y	Y
Ping Destinations		Y					Y	Y
Ping Sources	Y						Y	Y
Traceroute Requests by Destination		Y		Y				Y
Traceroute Requests by Source	Y				Y			Y
Traceroute Requests Pairs	Y	Y						Y
TTL Expired in Transit	Y	Y						Y
Unreachable Destination by Source	Y						Y	Y
Unreachable Destination Networks			Y				Y	Y
Unreachable Destinations	Y		Y				Y	Y

- **Bad IP Headers Report**
  Displays the source address and flow count of each bad IP header--each IP header that failed the checksum.
- **Fragmentation Required and DF Flag Set Report**
  Displays the following information about packets that require fragmentation--packets that had unreachable hosts or that were flagged Fragmentation Needed and Don't Fragment:
  - Destination address
  - Number of source addresses that sent this type of data to the destination address
  - Number of flows that contained packets of this type
- **ICMP Traffic Summary Report**
  Summarizes the Internet Control Message Protocol (ICMP) types and codes that occurred. The report contains the following information:

  - ICMP description, type, and code
  - Volume of inbound packets for each ICMP
  - Flow count for each ICMP
- **Ping Conversation Pairs Report**
  Displays the following information about ping requests:
  - Source address
  - Destination address
  - Volume of inbound packets
  - Flow count
- **Ping Destinations Report**
  Displays the following information about ping request destinations:
  - Destination address that received the request
  - Volume of inbound packets
  - Flow count
- **Ping Sources Report**
  Displays the following information about ping request sources:
  - Source address that generated the request
  - Volume of inbound packets
  - Flow count
- **Traceroute Requests by Destination Report**
  Displays the following information about traceroute request destinations:
  - Destination address of the traceroute request
  - Number of source addresses that sent traceroute requests to the destination
  - Flow count
- **Traceroute Requests by Source Report**
  Displays the following information about traceroute request sources:
  - Source address of each traceroute request
  - Number of destination addresses that received traceroute requests from the source
  - Flow count
- **Traceroute Requests Pairs Report**
  Displays the following information about each traceroute address pair:
  - Source address of the traceroute request
  - Destination address of the traceroute request
  - Flow count
- **TTL Expired in Transit Report**
  Displays the following information about data that met or exceeded the Time To Live (TTL) threshold:
  - Source address
  - Destination address
  - Flow count
- **Unreachable Destination by Source Report**
  Displays the following information about sources that tried to connect with unreachable destinations:
  - Source address
  - Volume of inbound packets
  - Flow count
- **Unreachable Destination Networks Report**
  Displays the following information about unreachable destinations:

  - Destination network and subnet mask
  - Number of inbound packets
  - Flow count
- **Unreachable Destinations Report**
  Displays the following information about sources and unreachable destinations:
  - Source address
  - Destination network and subnet mask
  - Volume of inbound packets
  - Flow count

MAC Report Group

The **MAC** Flow Forensics reports have the following fields.

Report	Dest MAC	Source MAC	Bytes	Rate (Bits)	% Total (Bytes)	Flows	Pkts	Rate (Pkts)	% Total (Pkts)
Destination MAC	Y		Y	Y	Y	Y	Y	Y	Y
MAC Pairs	Y	Y	Y	Y	Y	Y	Y	Y	Y
Source MAC		Y	Y	Y	Y	Y	Y	Y	Y

- **Destination MAC Report**
  Displays the volume, rate, and percent of total inbound bytes/packets, as well as the flow count on each destination Media Access Control (MAC) address.
- **MAC Pairs Report**
  Displays the volume, rate, and percent of total inbound bytes/packets, as well as the flow count on each pair of source and destination MAC addresses.
- **Source MAC Report**
  Displays the volume, rate, and percent of total inbound bytes/packets, as well as the flow count on each source MAC address.

MPLS Reports

The **MPLS Labels** Flow Forensics report has the following fields.

Report	Router Addr	Label Addr	Label Type	Top Label	Bytes	Rate (Bits)	% Total (Bytes)	Flows	Pkts	Rate (Pkts)	% Total (Pkts)
MPLS Labels	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

MPLS Labels Report
Displays the volume, rate, and percent of total inbound bytes/packets, as well as the flow count of traffic that had a unique combination of the following values:

- Router address
- Multiprotocol Label Switching (MPLS) address
- MPLS label type
- MPLS top label

Network Reports Group

The **Network** Flow Forensics reports have the following fields.

Report	Src AS	Dest AS	Src Network	Dest Network	Src Addr	Dest Addr	ToS	Next Hop	TCP Reset Count	Bytes	Rate (Bits)	% Total (Bytes)	Flows	Pkts	Rate (Pkts)	% Total (Pkts)
Autonomous System Pairs	Y	Y								Y	Y	Y	Y	Y	Y	Y
Autonomous System Pairs (with Destination Network)	Y	Y		Y						Y	Y	Y	Y	Y	Y	Y
Destination Autonomous Systems		Y								Y	Y	Y	Y	Y	Y	Y
Destination Networks				Y						Y	Y	Y	Y	Y	Y	Y
Network Pairs			Y	Y						Y	Y	Y	Y	Y	Y	Y
Network Pairs (with ToS)			Y	Y			Y			Y	Y	Y	Y	Y	Y	Y
Next Hops								Y		Y	Y	Y	Y	Y	Y	Y
Source Autonomous Systems	Y									Y	Y	Y	Y	Y	Y	Y
Source Networks	Y									Y	Y	Y	Y	Y	Y	Y
TCP Resets					Y	Y			Y							

- **Autonomous System Pairs Report**
  Displays the volume, rate, and percent of total inbound bytes and packets. Also displays the flow count of traffic between a pair of source and destination autonomous systems.
- **Autonomous System Pairs (with Destination Network) Report**
  Displays the volume, rate, and percent of total inbound bytes and packets. Also displays the flow count of traffic that had a unique combination of the following values:
  – Source AS
  – Destination AS
  – Destination network and subnet mask
- **Destination Autonomous Systems Report**
  Displays the volume, rate, and percent of total inbound bytes and packets. Also displays the flow count of traffic on each destination autonomous system.
- **Destination Networks Report**

Displays the volume, rate, and percent of total inbound bytes/packets, as well as the flow count of traffic on each destination network and subnet.

- **Network Pairs Report**
  Displays the volume, rate, and percent of total inbound bytes and packets. Also displays the flow count of traffic on each pair of source and destination network subnets.
- **Network Pairs (with ToS) Report**
  Displays the volume, rate, and percent of total inbound bytes and packets. Also displays the flow count of traffic on each network pair that had a unique combination of the following values:
  – Source network and subnet mask
  – Destination network and subnet mask
  – ToS
- **Next Hops Report**
  Displays the volume, rate, and percent of total inbound bytes and packets. Also displays the flow count of traffic for each next-hop address.
- **Source Autonomous Systems Report**
  Displays the volume, rate, and percent of total inbound bytes and packets. Also displays the flow count of traffic on each source autonomous system.
- **Source Networks Report**
  Displays the volume, rate, and percent of total inbound bytes and packets. Also displays the flow count of traffic on each source network and subnet.
- **TCP Resets Report**
  Displays the TCP reset count of traffic on each source and destination address pair.

QoS Report Group

The **QoS** (Quality of Service) Flow Forensics reports have the following fields.

Report	DSCP	ToS	Bytes	Rate (Bits)	% Total (Bytes)	Flows	Pkts	Rate (Pkts)	% Total (Pkts)
Differentiated Services	Y		Y	Y	Y	Y	Y	Y	Y
Type of Service		Y	Y	Y	Y	Y	Y	Y	Y

- **Differentiated Services Report**
  Displays the volume, rate, and percent of total inbound bytes and packets. Also displays the traffic flow count for each DiffServ code point (DSCP) value.
- **Types of Service Report**
  Displays the volume, rate, and percent of total inbound bytes and packets. Also displays the traffic flow count for each Type of Service (ToS).

Session Report Group

The **Session** Flow Forensics reports have the following fields.

Report	Router Addr	Interface In	IP Protocol	Src Addr	Src Addr (IPv6)	Src Port	Interface Out	Dest Addr	Dest Addr (IPv6)	Dest Port	ToS	Bytes	Rate (Bits)	% Total (Bytes)	Flows	Flow Duration	Pkts	Rate (Pkts)	% Total (Pkts)	Engine	Application
Client-Server Sessions			Y					Y		Y		Y	Y	Y	Y		Y	Y	Y		
Conversation Sessions	Y	Y	Y	Y				Y		Y	Y	Y	Y	Y	Y	Y			Y		
Conversation Sessions (NBAR2)	Y	Y	Y	Y		Y		Y		Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Conversations			Y	Y		Y		Y		Y		Y	Y	Y	Y		Y	Y	Y		
Conversations (IPv6)			Y		Y	Y			Y	Y		Y	Y	Y	Y		Y	Y	Y		
Conversations (with Interfaces)	Y	Y	Y	Y		Y	Y	Y		Y		Y	Y	Y	Y		Y	Y	Y		
Destination Applications			Y							Y		Y	Y	Y	Y		Y	Y	Y		
Destination Endpoints			Y					Y		Y		Y	Y	Y			Y	Y	Y		
Protocols			Y									Y	Y	Y	Y		Y	Y	Y		
Server-Client Sessions			Y	Y		Y		Y				Y	Y	Y	Y		Y	Y	Y		
Source Applications			Y			Y						Y	Y	Y	Y		Y	Y	Y		
Source Endpoints			Y	Y		Y						Y	Y	Y	Y		Y	Y	Y		

- **Client-Server Sessions**
  Displays the volume, rate, and percent of total inbound bytes and packets. Also displays the flow count of each session that had a unique combination of the following values:
  – Source and destination address
  – Destination port
  – IP protocol
- **Conversation Sessions Report**
  Displays the volume, rate, and percent of total inbound bytes and packets. Also displays the flow count and cumulative flow duration of each conversation session that had a unique combination of the following values:

  - Router address
  - Inbound interface
  - Source address and port
  - Destination address and port
  - ToS
  - IP protocol
- **Conversation Sessions (NBAR2) Report**
  Displays conversation session traffic and identifies any NBAR2 (Next Generation Network-Based Application Recognition) data that is included. The report table includes a row for each unique combination of the following values:
  - IP address of the router that sent the data
  - Name of the interface that received the data
  - IP protocol for the data
  - IP addresses and ports of the source host and destination host in the conversation
  - Type of service
  - Volume, rate and percentage of total traffic that the data represents, shown bytes (or megabytes, kilobytes) and packets
  - Flow count
  - Total duration of the flows in the row
  - NBAR2 Engine name and ID
    Standard NBAR2 data has "layer7 (13)" in the Engine column.
  - NBAR2 Application name and ID

  If a router is not configured to return NBAR2 data, a zero (0) appears in its rows under the **Engine** and **Application** columns. If other columns contain a zero for the router rows, the router may not be configured to return the fields for those values.
  The **Application** column values show the NBAR2 application name followed by the application ID. The application name is included if it is defined by the standard (not custom) NBAR2 engine. Otherwise the **Application** value may consist of only the application ID.
- **Conversations Report**
  Displays the volume, rate, and percent of total inbound bytes and packets. Also displays the flow count of traffic for each IPv4 address pair that had a unique combination of the following values:
  - Source address and port
  - Destination address and port
  - IP protocol
- **Conversations (IPv6) Report**
  Displays the volume, rate, and percent of total inbound bytes and packets. Also displays the flow count of traffic for each IPv6 address pair that had a unique combination of the following values:
  - Source address and port
  - Destination address and port
  - IP protocol
- **Conversations (with Interfaces)**
  Displays the volume, rate, and percent of total inbound bytes and packets. Also displays the flow count of each conversation that had a unique combination of the following values:
  - Router address
  - Inbound and outbound interface
  - Source address and port
  - Destination address and port
  - IP protocol
- **Destination Applications Report**

Displays the volume, rate, and percent of total inbound bytes and packets. Also displays the flow count of traffic that had a unique combination of the following values:
  – Destination port
  – IP protocol
- **Destination Endpoints Report**
  Displays the volume, rate, and percent of total inbound bytes and packets. Also displays the flow count of traffic that had a unique combination of the following values:
  – Destination address
  – Destination port
  – IP protocol
- **Protocols Report**
  Displays the volume, rate, and percent of total inbound bytes and packets. Also displays the flow count of traffic that had a unique IP protocol.
- **Server-Client Sessions Report**
  Displays the volume, rate, and percent of total inbound bytes and packets. Also displays the flow count of each traffic session that had a unique combination of the following values:
  – Source address and port
  – Destination address and port
  – IP protocol
- **Source Applications Report**
  Displays the volume, rate, and percent of total inbound bytes and packets. Also displays the flow count of each traffic session that had a unique combination of the following values:
  – Source port
  – IP protocol
- **Source Endpoints Report**
  Displays the volume, rate, and percent of total inbound bytes and packets. Also displays the flow count of each traffic session that had a unique combination of the following values:
  – Source address
  – Source port
  – IP protocol

TCP Reports

The **TCP Flags** Flow Forensics report has the following fields.

Report	TcpFlags	Bytes	Rate (Bits)	% Total (Bytes)	Flows	Pkts	Rate (Pkts)	% Total (Pkts)
TCP Flags	Y	Y	Y	Y	Y	Y	Y	Y

- **TCP Flags Report**
  Displays volume, rate, and percent of total inbound bytes/packets, as well as the flow count of traffic for each TCP flag.

VLAN Report Group

The **VLAN** Flow Forensics reports have the following fields.

Report	Destination VLAN	Source VLAN	Bytes	Rate (Bits)	% Total (Bytes)	Flows	Pkts	Rate (Pkts)	% Total (Pkts)
Destination VLANs	Y		Y	Y	Y	Y	Y	Y	Y
Source VLANs		Y	Y	Y	Y	Y	Y	Y	Y
VLAN Pairs	Y	Y	Y	Y	Y	Y	Y	Y	Y

- **Destination VLANs Report**
  Displays the volume, rate, and percent of total inbound bytes/packets, as well as the flow count of traffic on each destination VLAN.
- **Source VLANs Report**
  Displays the volume, rate, and percent of total inbound bytes/packets, as well as the flow count of traffic on each source VLAN.
- **VLAN Pairs Report**
  Displays the volume, rate, and percent of total bytes in/packets in, as well as the flow count of traffic on each source and destination VLAN pair.

WAAS Segment Report Group

WAAS Segment Report

Displays a report that identifies the WAAS (Wide Area Application Services) segment number and pass-through reason. Having the pass-through reason helps you determine why flow is not optimized.

The report shows meaningful data if the reporting routers have WAAS configured and if Cisco Performance Agent monitors the WAAS traffic. Other traffic does not return any meaningful values except for the router IP address.

The **Application** column values show the NBAR2 application name followed by the application ID. The application name is included if it is defined by the standard (not custom) NBAR2 engine. Otherwise the **Application** value may consist of only the application ID.

The **WAAS Segment** Flow Forensics report has the following fields.

Report	Router Addr	Application	Client Address	Server Address	WAAS Segment	WAAS Passthrough Reason
WAAS Segment	Y	Y	Y	Y	Y	Y

The report table includes a row for each unique combination of the following values:

- IP address of the router that sent the data
- Application name (NBAR2 application name and ID)
- IP addresses of the client and server
- WAAS segment: Type of WAAS data source
- WAAS passthrough reason

Create or View a Flow Forensics Report

Open the **Flow Forensics** page and create Flow Forensics reports to see details about raw data flows for troubleshooting. Use the saved report definitions to generate an updated report at any time. You can also use a report definition as a basis for another Flow Forensics report.

As the number of reports grows, use the folder management system to keep the reports organized and accessible.

Create a Flow Forensics Report

Complete the following steps to create a Flow Forensics report.

Follow these steps:

1. Select **Flow Forensics** from the NFA console menu.
2. Click **Create New Report**.
   The **Report Settings** page opens and displays options for the default report type, **Conversation Sessions**.
3. (*Optional*) Change the report type:
   – Click **[change]** next to the **Report Types** label to select a different report type. The **Report Types** dialog opens.
   – Click the heading for the report group that interests you. The report list expands for the selected group.
   – Select the report type from the expanded list. You return to the **Report Settings** page, which shows a name and label for the new report type.
4. (*Optional*) Identify the report and set the folder for saving it by entering values in the following fields:
   – **Name**: Add a name for the report.
   – **Description**: Add a description for the report.
   – **Folder**: Select a parent folder for the report.
5. (*Optional*) Change the filters that are used in the report definition:**Example:** To exclude the FTP protocol, complete the following steps:
   – **Remove Filters**: Click the **X** next to the filter name in the **Add Filters** list to delete a filter from the report definition.
   – **Add Filters**: Use the **Add Filters** options to filter the displayed data.
   a. Select **RA: Protocol** filter.
   b. Select **NotEqual**.
   c. Click **Index**.
   d. Select the ftp protocol.

   You can get flow forensic report from either specific harvester or from all harvester, when you select the following ROUTER filter types.

   - Router Address
   - Router Address and Interface In
   - Router Address and Interface Out
   - Router Address and Interface In or Out

   When you generate report based on one of these filters, NFA prompts you to enter the router IP. Enter the router IP address. NFA displays a drop down with list of associated harvesters. Based on your selection, the NFA displays report from All Harvesters (Default) or from selected harvester.

   NOTE

   - Harvesters Drop-down is displayed only if the specified IP is available on any of the harvesters.
   - Harvesters Drop-down is displayed even if the router belongs to only one harvester.
6. (*Optional*) Change the timespan for collecting report data: Select **Start Date** and **End Date** values.
7. Run or save the report:

- **Save**: Save the report so it is available for later use, but is not executed immediately.The report is saved and the **Saved** icon is added to the **Report Settings** page.
- **Run**: Save, queue, and run the report.The **Report Queued** message appears. When the report finishes execution, the message closes and the report appears.

Click **Back to Folders** to return to the Flow Forensic **Reports** folders before execution is complete.

Flow Forensics Report Filters

The following list describes the available filters. Certain filters might not be available for all reports.

- **RA: Protocol**: Filters for the actual protocol, including Application mapping rules
- **RA: Interface**: Filters for the actual interface, excluding aggregate and custom virtual interfaces
- **RA Type-of-service**: Filters for the DX NetOps type of service
- **Destination Address**: Filters for the IP address of one or multiple destination hosts
- **Destination Autonomous System**: Filters for the autonomous number for the destination network
- **Destination MAC Address**: Filters for the destination MAC address
- **Destination Mask**: Filters for the IP mask of a destination network
- **Destination Port**: Filters for the destination port number (0 through 65,535)
- **Destination VLAN**: Filters for the destination address of the VLAN
- **Flow Count**: Filters for the number of flows
- **Flow Duration**: Filters for the duration of the flows
- **ICMP**: Filters for information about ICMP
- **MPLS Top Label**: Filters for the top label of MPLS
- **MPLS Top Label IP Address**: Filters for the IP address of the MPLS top label
- **MPLS Top Label Type**: Filters for the type of MPLS top label type
- **Next Hop**: Filters for the IP address of next destination hop
- **Percent of Total Traffic for Bytes In**: Filters for the percentage of bytes in total traffic
- **Percent of Total Traffic for Packets In**: Filters for the percentage if packets in total traffic
- **Protocol**: Filters for the actual IP Protocol Number (6=TCP,17=UDP)
- **Protocol and Destination Port**: Filters for the actual IP Protocol Number (6=TCP,17=UDP) and the destination port number (0 through 65,535)
- **Protocol and Source or Destination Port**: Filters for the actual IP Protocol Number (6=TCP,17=UDP) and the source or destination port number (0 through 65,535)
- **Protocol and Source Port**: Filters for the actual IP Protocol Number (6=TCP,17=UDP) and the IP address of a source host
- **Router Address**: Filters for the IP address of the router
- **Router Address and Interface In**: Filters for the router address and router interface (in) index number
- **Router Address and Interface In or Out**: Filters for the router address and router interface (in or out) index number
- **Router Address and Interface Out**: Filters for the router address and router interface (out) index number
- **Source Address**: Filters for the IP address of one or multiple source hosts
- **Source Autonomous System**: Filters for the autonomous number for the source network
- **Source MAC Address**: Filters for the source MAC address
- **Source Mask**: Filters for the IP mask of a source network
- **Source or Destination Address**: Filters for the IP address of a source host or a destination host
- **Source or Destination MAC Address**: Filters for the source or destination of the MAC address
- **Source or Destination Mask**: Filters for the IP mask of a source network or a destination network
- **Source or Destination Port**: Filters for the source port number or the destination port number
- **Source or Destination VLAN**: Filters for the source or destination of the VLAN
- **Source Port**: Filters for the source port number
- **Source VLAN**: Filters for the source VLAN
- **TCP Flags**: Filters for the Transmission Control Protocol (TCP) flags
- **Type-of-service**: Filters for the IP type of service number (0 through 255)
- **Volume of Bytes In**: Filters for the volume of bytes in
- **Volume of Packets In**: Filters for the volume of packets in

View a Flow Forensics Report

Complete the following steps to view an existing Flow Forensics report.

Follow these steps:

1. Select **Flow Forensics** from the NFA console menu.
2. Click the parent folder that contains the report.
   The list of reports in the folder opens.
3. Click the check box next to the report that you want to view.
4. Click **Run**.
   A verification message opens.
5. Click **OK**.
6. Refresh the view until the report status is **Complete**.
   The report results are displayed.

Analysis Reports

You can troubleshoot problems as they occur by using Analysis reports to identify issues before users in your environment are adversely affected.

An Analysis report is designed to compare collected network data to a threshold so you can identify potential bottlenecks, anomalies, and viruses. Analysis reports help you identify potential problems before they become serious issues. You can schedule these reports to run regularly, which means you can continually analyze your network traffic for potential issues.

To open the Analysis page, click **Analysis** in the NFA console menu.

The **Analysis** page includes the following options:

- **New**
  Create an Analysis report.
- **Run**
  Execute one or more reports at the same time.
- **Move to Folder**
  Transfer one or more reports to a different directory.
- **Cancel**
  Immediately stop the execution of one or more reports that are running.

When you create **Analysis** reports, you specify protocol, ToS, host, and conversation filters to use. The valid filters and combinations of filters are:

- No filters
- Protocol filters
- ToS filters
- Protocol and ToS filters
- Protocol and host filters
- Protocol and conversation filters
- ToS and conversation filters
- ToS and host filters
- Conversation filters
- Host filters

The troubleshooting capabilities of DX NetOps are not limited to the features discussed in these topics.

Scheduled Reports

You can schedule an analysis report over a particular duration of time. These duration reports are the only reports which can be scheduled to repeat. You can email a report on a scheduled basis, but the same report will be emailed every time unless someone manually runs the report for a different time period.

A scheduled report begins the next scheduled report at the `scheduled_next` value in the database. This value is visible as the **Next run** in the **Reporting Period** when editing an Analysis Report. When the report starts this value is set as the **End Time** of the report.

The `scheduled_last` value in the database is updated with this value. This time is the last time the report, which is currently running, was scheduled to run.

Using the report time zone, this time is converted to Unix Time and all further calculations are done in Unix Time. A scheduled report displays the results in GMT regardless of the time zone used to specify the start time for the report.

If a time filter is specified it alters the data which is displayed but not the duration of the report. The time filter allows the user to create a report that shows data only for a time period of interest. That portion of the reporting period which falls outside the time period of interest will have no data displayed.

A scheduled report can be saved or saved and queued. If the report is saved, it first runs at the next scheduled time for the report. If the report is saved and queued, the first run starts immediately and covers the duration specified in the report, with the end time the point in time when the report was queued, not the scheduled time of the report.

Adding information about the duration, time filters, interfaces, protocols, etc. into the description of the report will help a person reviewing the report understand the data being presented.

Create, Change, or View an Analysis Report

No Analysis reports are installed by default. A report folder named **Analyses** exists by default, which you cannot delete. You can store your saved Analysis reports in this folder or you can create other folders to organize your Analysis reports.

Create an Analysis Report

Follow these steps:

1. Select **Create New Report**.
   The **Analysis Wizard** opens and displays options for creating an analysis definition or modifying a copy of an existing analysis definition.
2. Select one of the options and click **Next**.
   – **Create a new analysis**
   Define an entirely new **Analysis** report.
   – **Copy an existing analysis**
   Select an existing report to copy and use as a basis for the new report.
   If you select **Create a new analysis**, the **Select Interfaces** page of the Analysis wizard opens.
3. Select interfaces or interface groups for the report.
   a. Click one of the following options:
   – • **Add Interface Filter**
   Select one or more individual interfaces from the **Interface Index**.
   a. • **Add Interface Group Filter**
   Select one or more interface groups from the **Interface Group Selection** list.
   The selected interfaces are added to the interface list.
   b. Accept the default value or set the **Inclusion** value for each interface or interface group:
   • **Include** sets the program to use report data from the interface or group.
   • **Exclude** sets the program to eliminate data from the interface or group.

c. Click **Next**.
The **Specify Filters & Threshold** page opens.

4. Specify the settings on the **Specify Filters & Threshold** page:
   a. (Optional) Specify filters for gathering or excluding report data, then set the Inclusion value for each filter to **Include** or **Exclude**:
      - **Add Protocol Filter**
        Select individual protocols from the **Protocol Index**.
      - **Add Protocol Group Filter**
        Select protocol groups from the **Protocol Group Index**.
      - **Add ToS Filter**
        Select individual ToS values from the **ToS Index**.
      - **Add ToS Group Filter**
        Select ToS groups from the **ToS Group Index**.
      - **Add Host Filter**
        Specify a host IP address and mask.
      - **Add Conversation Filter**
        Identify the IP addresses and mask for each party in the conversation pair.
   b. Accept the default value or set the Inclusion value for each filter you specified:
      - **Include**
        For each filter listed, use only the data of the listed type. For example, use data from the listed protocol group, but not from other protocol groups.
      - **Exclude**
        For each filter listed, do not use the data of the listed type. For example, do not use data from the listed protocol group, but do use data from other protocol groups.
   c. Set the **Threshold Settings** values to specify the threshold that is used for the Analysis report.
   d. Click **Next**.

   For example, to report on overutilized interfaces you might specify a threshold to examine total traffic that goes above 70 percent utilization.
5. Click **Next**.
   The **Specify Schedule** page opens.
6. Select the type of reporting period from the **Period** list on the **Specify Schedule** page:
   – **Duration**
     Limits the reporting period to an amount of time, ending at the time the report runs. Enter the number of time units in the **Last** box. Select a unit of time from the list (days, weeks, months, or years).
     You can set up a schedule for a **Duration** report or you can run the report on demand.
   – **Start and end**
     a. Use one of the following methods to specify a **Start** date and **End** date:
        Select the calendar icons and click dates to specify the **Start** and **End** of the report period.
        Select hour and time values from the lists. Hour values are expressed in 24-hour format.
     b. Enter the number of time units in the **Resolution** box. Select a value from the list for **Start** and **End**.
        A Start-and-End report runs on demand. You cannot set up a schedule for a Start-and-End report.
7. Accept the default **Resolution** setting on the **Specify Schedule** page or enter the number of time units in the **Resolution** box. Select a unit of time from the list (minutes, hours, days, weeks, months, or years).
8. (Optional) Select a time filter from the list on the **Specify Schedule** page, if your Administrator has created a time filter that is appropriate for your report.

   TIP
   The time filter information does not display on the report as part of the date and time. Give the report a name and description that reflects the selected time filter.

9. (Optional) Select the **Schedule** check box on the **Specify Schedule** page and specify the following options:
   a. **Schedule**: Select the type from the **Schedule** list.
      - **Daily**: Select the day or days of the week, time of day, and time zone for report generation.
      - **Weekly**: Select the day of the week, time of day, and time zone for report generation.
      - **Monthly**: Select either the date or the week in the month and day of the week. Select the time of day, and time zone for report generation.
      - **Quarterly**: Select a month that ends the first reporting quarter, time of day, and time zone.
      - **Yearly**: Select a month that ends the first reporting year, time of day, and time zone.
   b. (Optional) **Email Results To**: Enter the email addresses of all the report recipients in the format name@domain. Separate multiple addresses with a comma or semi-colon.

      NOTE
      The options in the **Recurrence** section are available only if you select **duration** as the reporting period type.
10. Click **Next**.
    The **Enter Name** page opens.
11. Identify the report and its location:
    – **Folder**
      Accept the default folder or select a different folder to contain the new report.
    – **Name**
      Give the new report a name that will appear in the **Reports** list.
    – **Description**
      (Optional) Add a description to help identify the report. For example, you may to use the description to identify scheduled reports and to indicate distinguishing features of Duration reports. Include any time filter information, for ease of identification.

    Click **Next**. The **Summary & Submit** page opens.
12. Review the information in the **Report Definition Summary**.
    – **Save**
      Save the report definition and return to the **Analysis** page. The report will run on the schedule selected.
    – **Save and Queue Report**
      Queue the report to run and return to the **Analysis** page. The report will run now for the data available for the duration selected.
    – **Back**
      Return to previous pages in the wizard to redefine the report.

Edit an Analysis Report

When you view an Analysis report, you may want to review the **Report Definition Summary** to ensure that the report definitions are correct. This summary provides access to the Analysis wizard pages in case you want to change any settings.

Follow these steps:

1. In the **Report Settings** section at the top of the report page, click **Edit**.
   The **Report Definition Summary** is displayed.
2. Review the current settings for the Analysis report.
3. Click the name of a category to open the associated page for making changes.
4. When you have made all the desired changes, click **Queue Report** to regenerate the report with the new settings. You return to the **Reports** list, which shows the modified report with a status of **Queued**. When the report has been regenerated and is ready to be viewed, the status displays as **Complete**.
   A scheduled report will be queued and will be generated at the next scheduled runtime.

View an Analysis Report

When you have defined the Analysis reports that you need, you are ready to generate a new report and to view it in the NFA console. You can run and view the following types of reports:

- Unscheduled duration reports on demand.
- The most recent versions of scheduled duration reports that have been generated automatically.
- The most recent versions of start-and-end reports.

Follow these steps:

1. Select **Analysis** from the NFA console menu if the **Analysis** page is not already open.
   The **Analysis** page contains two panes. The left pane lists the folders that are used to store saved report definitions. The right pane lists the reports in the currently selected folder.
2. Click the name of the folder that contains the report you want to view.
3. Click the report name in the **Reports** list.
   The report runs. The results are displayed when report generation is complete.

You can click an interface name to display more information about that interface. For example, you may be able to view a calendar chart that shows the time and duration of a violation.

Site to Site Reports

Site to Site reports enable you to view volumes of data between two or more sites. Sites can be defined as a collection of subnets that can also be discontiguous.

Use Site to Site reports to compare bytes in, rate in, bytes out, and rate out between pairs of sites. You can export the displayed data to a file in comma-separated value (CSV) format.

The reports can be configured for:

- Sites involved in the report (including creation of site definitions)
- Report schedule (scheduled or on demand)
- Time period
- Selecting data granularity (1-minute or 15-minute data)

You can use saved report definitions to generate an updated report at any time. As the number of reports grows, use the folder management system to keep the reports organized and accessible.

Scheduled Reports

You can schedule a Site to Site report over a particular duration of time. These duration reports are the only reports which can be scheduled to repeat. You can email a report on a scheduled basis, but the same report is emailed every time unless someone manually runs the report for a different time period.

A scheduled report begins the next scheduled report at the `scheduled_next` value in the database. This value is visible as the **Next run** in the **Reporting Period** when editing a Site to Site Report. When the report starts this value is set as the **End Time** of the report.

The `scheduled_last` value in the database is updated with this value. This time is the last time the report, which is currently running, was scheduled to run.

Using the report time zone, this time is converted to Unix Time and all further calculations are done in Unix Time. A scheduled report displays the results in GMT regardless of the time zone used to specify the start time for the report.

If a time filter is specified it will alter the data which is displayed but not the duration of the report. The time filter allows the user to create a report with shows data only for a time period of interest. That portion of the reporting period which falls outside the time period of interest has no data displayed.

A scheduled report can be saved or saved and queued. If the report is saved, it first runs at the next scheduled time for the report. If the report is saved and queued, the first run starts immediately and covers the duration specified in the report, with the end time the point in time when the report was queued, not the scheduled time of the report.

Adding information about the duration, time filters, interfaces, protocols, etc. into the description of the report will be descriptive text to help a person reviewing the report understand the data being presented.

Create or View a Site to Site Report

When you click the **Site to Site** tab, the **Saved Report Folders** pane displays on the left and any existing reports display on the right.

You can perform the following tasks in the **Saved Report Folders** pane: create folders, rename folders, and delete custom folders. You cannot delete the default **Site to Site Reports** folder. Create additional folders to provide more extended organization for your Site to Site reports.

The right pane displays the Site to Site report definitions in the currently selected folder. Use the links at the top and bottom of the pane to delete or run selected reports, as well as to create or move reports.

Create a Site to Site Report

Use the Site to Site wizard to create a site to site report step by step. The wizard guides you to select many options, such as the sites involved in the report, the report schedule, the data resolution (1-minute or 15-minute), and the reporting period.

Follow these steps:

1. Select **Site to Site** from the NFA console menu.
2. Click **New**.
   The **Site to Site Report Wizard** opens and shows the options **Create a new Site to Site report** and **Copy an existing report**.
3. Click **Create a new Site to Site report** and click **Next**.
   Note: Alternatively, you can select the **Copy an existing report** option and modify an existing report.
   The **Select Sites** page opens.
4. Select at least two sites for the report.
   You can either click **Select Sites** to select an existing site or sites, or create a new site definition.
   To create a new site definition:
   a. Click **Create Site**.
   b. Click **Save and Add New Site** to create a new site definition without adding it to the report, or **Save and Add Sites to Report** to add a new definition to the report.
   c. Enter
      - **Site Name**

        NOTE
        Site names are unique within a tenant.
      - **Site Description** (optional)
   d. Click **Add Network.**
      - **Network Name**
      - **Network Subnet**
   e. (Optional) Click **Add Network** to add additional networks to the site definition.
   f. Click **Save and Add Sites to Report**.
   The selected sites are added to the site list.
5. Click **Next**. The **Specify Schedule** page opens.

6. Select the type of reporting period from the **Period** list in the **Reporting Period** area:
   - **duration**: Limits the reporting period to an amount of time, ending at the time the report runs. Enter the number of time units in the **Last** box.
     Select a unit of time from the list (**days**, **weeks**, **months**, or **years**).
     You can set up a schedule for a duration report or you can run the report on demand.
   - **start & end**: Specify a **Start** date and **End** date either by using the calendar icons or by selecting hour and time values from the lists. Hour values are expressed in 24-hour format.
7. Select the data resolution (1-minute or 15-minute) in the **Resolution** box.
8. (Optional) Select the **Schedule** check box in the **Recurrence** area and specify the following options:
   - **Schedule**: Select the type from the Schedule list.
     - **Daily**: Select the day or days of the week, time of day, and time zone for report generation.
     - **Weekly**: Select the day of the week, time of day, and time zone for report generation.
     - **Monthly**: Select either the date or the week in the month and day of the week. Select the time of day, and time zone for report generation.
     - **Quarterly**: Select a month that ends the first reporting quarter, time of day, and time zone.
     - **Yearly**: Select a month that ends the first reporting year, time of day, and time zone.

     NOTE
     The options in the **Recurrence** section are available only if you select **duration** as the reporting period type.
9. (Optional) Enter the email addresses of all the report recipients in the format *name@domain*. Separate multiple addresses with a comma or semi-colon.
10. Click **Next**. The **Enter Name** page opens.
11. Identify the report and its location:
    - **Folder**: Accept the default folder or select a different folder to contain the new report.
    - **Name**: Give the new report a name, which appears in the **Reports** list.
    - **Description**: (Optional) Add a description to help identify the report. For example, use the description to identify scheduled reports and to indicate distinguishing features of duration reports.

    Click **Next**. The **Summary & Submit** page opens.
12. Review the information in the **Report Definition Summary**.
    - **Save**: Save the report definition and return to the **Site to Site** page.
    - **Save and Queue Report**: Queue the report to run and return to the **Site to Site** page.

Define Site to Site Reporting Periods and Schedules

When you define a Site to Site report, you must specify the reporting time period and the resolution of the reporting data. You can define a specific, nonrecurring time period (for a start-and-end report) or you can choose a timespan that ends at the report runtime (for a duration report). You also have the option to set a duration report to regenerate on a recurring schedule and to have the automated reports sent out by email.

Specify a Reporting Period

The Site to Site Wizard provides a **Specify Schedule** page that defines the report time period. For a manually generated report, you can simply select a time period, resolution, and optional time filter. If you want to generate the report automatically at scheduled intervals, you can also use the **Schedule** option.

Follow these steps:

1. Select **Site to Site** from the NFA console menu if the **Site to Site** page is not already open.
2. Put the report in editable mode: Click the name of the report. If the report has been run previously, it runs again. In this case, click **Edit** in the **Report Settings** section at the top of the report page.
   The **Report Definition Summary** page opens, which contains links to the Site to Site wizard pages.
3. Click **Reporting Period** to open the **Specify Schedule** page of the Site to Site wizard.

NOTE
Site to Site reports display data using the Greenwich Mean Time (GMT) time zone. Even if a user has set the time to a specific time zone, the reports display the data using GMT.

4. Select the type of reporting period from the **Period** list on the **Specify Schedule** page:
   – Duration: Include data from the block of time that immediately precedes the report runtime. Enter a number of days, weeks, months, or years.
   You can set up a schedule for a duration report or you can run the report on demand. To run a scheduled report on demand, you can make a copy of the report and can disable the schedule in the copy.
   – Start and end: Include data from a specific, nonrecurring timespan. Use the calendars to specify a Start date and End date or select hour and time values from the lists.
   Hour values are expressed in 24-hour format.
   A start-and-end report runs on demand. You cannot set up a schedule for a start-and-end report.
5. Set the resolution (granularity for data collection) on the **Specify Schedule** page: Accept the default setting or select a value in the **Resolution** box.
6. (Optional) Select a time filter from the list on the **Specify Schedule** page, if your Administrator has created an appropriate time filter.

   TIP
   The time filter information does not display on the report as part of the date and time. Give the report a name and description that reflects the selected time filter.
7. Click **Save Changes**.
   Your changes are saved and you return to the **Report Definition Summary**.
8. (Optional) Return to the report list by clicking one of these buttons:
   – **Return to Listing**: Return to the **Site to Site** page without queuing the report to run.
   – **Queue Report**: Queue the report to run and return to the Site to Site page.

Specify Schedules for Auto-Generated Reports

Use the **Schedule** option to set the Site to Site report to regenerate at specified times.

For example, suppose that you want to check network traffic during the monthly backups that occur on the last Sunday of each month. You schedule a report to be regenerated on the last Sunday of every month. Suppose that operating system updates occur on the 15th of every month. To check the network traffic during those updates, you schedule a report to be regenerated on the 15th of each month.

NOTE
The options in the **Recurrence** section are available only if you select ‘duration’ as the reporting period type.

Follow these steps:

1. Select **Site to Site** from the NFA console menu if the **Site to Site** page is not already open.
2. Put the report in editable mode: Click the name of the report. If the report has been run previously, it runs again. In this case, click **Edit** in the **Report Settings** section at the top of the report page.
   The **Report Definition Summary** page opens, which contains links to the Site to Site wizard pages.
3. Click **Reporting Period** to open the **Specify Schedule** page of the Site to Site wizard.
4. Click the **Schedule** check box and choose a recurrence interval from the list:
   – **Daily**: Select the day or days of the week the report will run.
   – **Weekly**: Select the day of the week the report will run.
   – **Monthly**: Select either a date or a week and day combination to specify the one day per month the report will run.
   – **Quarterly**: Select the month that ends the first quarter in which the report will run. Starting with the specified quarter, the report will run on the last day of each quarter.
   – **Yearly**: Select the month that ends the first year the report will run. The report runs on the last day of the year.
   For all schedule interval types, select the time of day and time zone for the report to run.

5. (Optional) **Email Results To**: Enter the email addresses of anyone who should receive the report by email. Use the format *name@domain*. Separate multiple addresses with a comma or semi-colon.
6. Click **Save Changes**.
   Your changes are saved and you return to the Report Definition Summary.
7. (Optional) Return to the report list by clicking one of these buttons:
   – **Return to Listing**: Return to the **Site to Site** page without queuing the report to run.
   – **Queue Report**: Queue the report to run and return to the **Site to Site** page.

View a Site to Site Report

Complete the following steps to view an existing Site to Site report.

Follow these steps:

1. Select **Site to Site** from the NFA console menu if the **Site to Site** page is not already open.
2. Click the parent folder that contains the report.
   The list of reports in the folder opens.
3. Click the check box next to the report that you want to view.
4. Click **Run**.
   A verification message opens.
5. Click **OK**.
6. Refresh the view until the report status is **Complete**.
   The report results are displayed.

Manage Reports

Creating reports for your organization helps to troubleshoot various issues. Report definitions are saved so you can generate an updated report at any time or can use the report definition as a template to create another report. As the number of reports grows, use the folder management system to keep the report definitions easy to find.

The report folders are listed on the left. There are built-in folders, which you can rename, but cannot delete:

- Custom Reports
- Flow Forensics Reports
- Analyses
- Site to Site Reports

Create additional folders to provide more extended organization for your reports.

The right pane displays the names of the reports in the currently selected folder. Use the links at the top and bottom of the pane to delete or run selected reports, and to create or move report definitions.

Create a Report Folder

Create your own report folders to group reports. For example, you can use folder names to identify the purpose for a set of reports.

Follow these steps:

1. Navigate to the page for the report type: **Custom Reporting**, **Flow Forensics**, **Analysis**, or **Site to Site**.
2. Click **New** at the bottom of the **Saved Report Folders** pane.
   A pop-up dialog opens.
3. Enter a name for the new folder in the pop-up dialog.
4. Click **OK**.

The dialog closes and the new folder appears in the **Saved Report Folders** pane.

Move a Report to Another Folder

You can add folders and move report definitions among the folders to make the report definitions easier to find or to help identify them.

Follow these steps:

1. Navigate to the page for the report type: **Custom Reporting**, **Flow Forensics**, **Analysis**, or **Site to Site**.
2. In the **Reports** pane, select the check box next to all the reports that you want to move.
3. Click **Move to Folder**.
   A pop-up dialog opens.
4. Select a destination folder from the list.
5. Click **OK**.
   The dialog closes. The reports are now in their new location, and are visible when you click that folder name in the **Saved Report Folders** pane.

Rename a Report Folder

You can change the name of a report folder, including the name of the default report folder.

Follow these steps:

1. Navigate to the page for the report type: **Custom Reporting**, **Flow Forensics**, **Analysis**, or **Site to Site**.
2. In the **Reports** pane, select the check box next to the report folder that you want to rename.
3. Click **Rename**.
   A pop-up dialog opens.
4. Enter a new name for the folder in the pop-up dialog.
5. Click **OK**.
   The dialog closes. The new report folder name appears in the **Saved Report Folders** pane.

Delete Saved Report Definitions

When a saved report definition is no longer needed, you can delete it. You can delete one report definition at a time, or you can delete multiple definitions in a single folder simultaneously. Be sure that you do not delete a report definition that is useful as a template for creating other reports. Deleted report definitions cannot be restored.

Follow these steps:

1. In the **Reports** pane, select the check box next to all report definitions that you want to delete.
2. Click **Delete**. A confirmation dialog opens.
3. In the confirmation dialog, click **OK**.
   The confirmation dialog closes. The list of saved report definitions is updated.

Delete Report Folders

You can delete unneeded report folders and their contents. Be sure that you do not delete useful report definitions or folders. Deleted folders and report definitions cannot be restored.

Follow these steps:

1. In the **Saved Report Folders** pane, select the check box next to all the report folders that you want to delete.
2. Click **Delete**.
   A confirmation dialog opens. If the folders contain report definitions, the confirmation dialog reminds you of the number of report definitions that will be deleted.

3. In the confirmation dialog, click **OK**.
   The confirmation dialog closes. The list of saved report definitions is updated.

Views in Performance Center

You can view data from DX NetOps in Performance Center Console dashboards and Interface pages. Some of the views can be found in both consoles, although the view format and options may be different.

The topics in this section describe the Performance Center views that have DX NetOps data. The topics also describe the basics of working with the views.

Dashboards and Views

Dashboards are dynamic report-building pages in the Performance Center Console. Dashboards are accessible from the **Dashboards** tab (CA PC) or **Reports** tab (NPC). Each dashboard is a collection of views that present data from registered data sources on a single web page. The layout, views, time interval, and group context of each dashboard can be customized.

> **NOTE**
> Your user account role rights determine the dashboards that you can see.

Reports are static output from an on-demand selection or an exported dashboard page. Reports that you export from a dashboard create a static data set from the data and information in the associated dashboard. On-demand reports capture a data set from a single managed item or group in the Inventory. You can print reports, send them by email, or export them in CSV or PDF format. For each format, the report captures a selected data set.

Dashboards are organized in menus. *Menus* in the Performance Center Console are lists of items in the **Dashboards** tab (CA PC) or **Reports** tab (NPC). Menus group similar dashboards or report pages together. By default, Administrators and Designers can customize menus and assign them to user account roles.

Performance Center has a set of built-in dashboards and menus, which are available for use immediately after your administrator registers the product as a data source. Users who have the required role rights can customize dashboards, menus, and views to create a custom system for individual operators.

The menus and dashboards that are available to you are displayed when you hover over or click the **Dashboards** or **Reports** tab.

Performance Center Dashboards

Performance Center dashboards display views of data from registered data sources such as DX NetOps. *Views*, or *data views*, present statistical data, usually in a graph or table format. Each view represents a discrete set of collected data. Depending on your user account role rights, you can add and edit individual views or remove them from a dashboard page. In some cases, you can export the data to a file in CSV format.

View placement on dashboard pages is flexible. Users with the required role rights can customize dashboards. They can, for example, place views of application performance data beside views of volume data to help troubleshoot issues from a single page.

The predefined (factory) dashboards are organized into workflows. You can drill down from Top N views to more detailed metrics from a narrow context, such as an individual device. Workflows let you see data that may be related to the metric you are reviewing.

Administrators can create custom groups to display data for a specific set of sites, devices, or interfaces. You can apply groups to dashboards by using the group selector (the 'change' link at the top left). You can change the “context” of the dashboard to analyze data for specific groupings at the summary, device, or item level.

Views that show data for a group are CA Performance Center-generated views that contain rollups of data from data sources. Views that show data for a server or device, or detailed metrics from a narrow context often provide a drilldown path directly to the data source. The Single Sign-On feature lets you navigate seamlessly from a dashboard to a data source interface.

Types of Report Pages

Two categories of dashboards are available by default or through customization:

Summary pages provide high-level information, such as averages from groups of managed items. Summary dashboards often provide a drilldown path to more detailed, related pages from a selected context.

Context pages provide specific, focused performance or status data from a narrow context, such as a single router or server. These pages are available as drill-down links or tabs from Summary dashboards.

To drill in to a detailed view from a Summary dashboard, take one of the following steps:

- Right-click the item to select the context page that you want to see.
- Click the item to open the default context page.

> **NOTE**
> Your role rights must include the ability to Drill into Views.

Default sets of context pages are available for individual devices, interfaces, and servers. These pages include a set of customizable tabs that let you access more specific context data for a selected managed item. For example, the Router context includes tabs for **Health**, **Utilization**, and **Error** data.

Context Page Navigation

You can frequently access more information about individual managed items from dashboards. Most dashboards are composed of views of summary data, such as hourly rollups or averages from a group of items. If additional data is available from the data source, you can click linked items on the dashboard page to drill down into *context pages*.

> **NOTE**
> The role right to Drill into Views is required.

The views on context pages show filtered data from a narrow context, such as a view of data from a single managed item. Use the links to drill down into specific data and home in on the source of a performance problem.

In data views from some data sources, you can also right-click the name of an item in a table view to access a menu. For example, right-click the link that corresponds to an item name in the Inventory section. A menu lets you select a related context page, containing more granular data.

Finally, some context pages include tabs to additional pages of detailed data. Click a tab to see data that has been filtered by a selected managed item or type of item.

CA Network Flow Analysis Views in Performance Center

You can display DX NetOps data in the Performance Center Console in several ways:

CA Performance Center

Built-in CA Performance Center Dashboards with Enterprise-Wide Data:

- **Infrastructure Overview dashboard**:

  - Interfaces Over Threshold
  - Routers with the Most Flow Traffic
  - Top Enterprise Hosts by Volume
  - Top Enterprise Protocols by Volume
  - Top Flows by Interface
  - Top IP Interface Utilization (Flow)
- **Management: Management Overview dashboard**:
  - Top Flows by Interface
  - Top IP Interface Utilization (Flow)
- **Management: Network Overview dashboard**:
  - Top Enterprise Hosts by Volume
  - Top Enterprise Protocols by Volume
- **Capacity Planning: Router/Switch Capacity Watch Lists dashboard**:
  - Routers with the Most Flow Traffic

Custom CA Performance Center Dashboard Views with Interface-Specific Data: Display interface-specific DX NetOps data by adding the following views:

- Calendar Heat Chart (Flow)
- Stacked Protocol Trend
- Stacked ToS Trend
- Top Conversations (Bar)
- Top Conversations (Pie)
- Top Conversations (Table)
- Top Hosts (Bar)
- Top Hosts (Pie)
- Top Hosts (Table)
- Top Protocols (Bar)
- Top Protocols (Pie)
- Top Protocols (Table)
- ToS Summary (Pie)
- ToS Summary (Table)

Built-In CA Performance Center Interface Page Views with DX NetOps Data:

- **IP Performance tab**:
  - Stacked Protocol Trend
  - Top Conversations (Pie)
  - Top Hosts (Pie)
  - ToS Summary (Pie)
- **CBQoS tab**:
  - Stacked Protocol Trend
  - Stacked ToS Trend

CA NetQoS Performance Center

Enterprise-Wide Data on Built-in Dashboards in the CA NetQoS Performance Center Console:

- **Enterprise Dashboard**:

  - Interfaces Over Threshold
  - Top Enterprise Hosts by Volume
  - Top Enterprise Protocols by Volume
- **Traffic Analysis**:
  - Interfaces Over Threshold
  - Top Enterprise Hosts by Volume
  - Top Enterprise Protocols by Volume
  - Top IP Interface Utilization (Flow)
- **Network Overview**:
  - Top Enterprise Hosts by Volume
  - Top Enterprise Protocols by Volume
- **Routers/Switches Overview**:
  - Interfaces Over Threshold

Custom Dashboard Views with DX NetOps Data

Display DX NetOps data on custom report pages by adding the following views:

- Interfaces Over Threshold
- Multi-Interface Stacked Protocol Trends
- Multi-Interface Stacked ToS Trends
- Routers With the Most Flow Traffic
- Top Enterprise Hosts By Volume
- Top Enterprise Protocols By Volume
- Top Flows by Interface
- Top IP Interface Utilization (Flow)

Interface Page Views with DX NetOps Data in the CA NetQoS Performance Center Console:

- **Interface Capacity tab**:
  - Stacked Protocol Trend
  - Top Conversations (Pie)
  - Top Hosts (Pie)
- **Interface QoS tab**:
  - Stacked Protocol Trend
  - Stacked ToS Trend
  - Top Conversations (Pie Chart)
  - Top Hosts (Pie Chart)
  - Top Protocols (Pie Chart)
  - ToS Summary (Table)

Custom Interface Tab with DX NetOps Data Views

Display DX NetOps data on custom **Interface** pages by adding the following views:

- Calendar Chart (Flow)
- Top Conversations (Bar Chart)
- Top Conversations (Pie Chart)
- Top Conversations (Table)
- Top Hosts (Bar Chart)
- Top Hosts (Pie Chart)
- Top Hosts (Table)
- Top Protocols (Bar Chart)
- Top Protocols (Pie Chart)
- Top Protocols (Table)
- ToS Summary (Pie Chart)
- ToS Summary (Table)
- Stacked Protocol Trend
- Stacked ToS Trend

Enterprise-Level Views

You can view enterprise-wide data from DX NetOps in several Performance Center dashboard views, which are described in the topics that follow. The *top* interfaces, hosts, protocols, or ToS are the ones that have the highest traffic volume during the reporting period.

Top Enterprise Hosts by Volume

The **Top Enterprise Hosts by Volume** view in the Performance Center Console shows the enterprise hosts that have the highest traffic volume, as reported by DX NetOps.

The view shows a bar for each of a maximum of 10 hosts that have the highest traffic volume. The bar chart includes the following information:

- **Host**
  Identifies the host server by its name and IP address (Y-Axis). If an administrator has defined an alias for the device, the alias is displayed. Otherwise, the discovered device name is displayed.
- **Volume**
  Measures the total amount of data sent to or from the host, expressed in a scale that is appropriate for the highest-volume host (X-Axis).

Performance Center views show the data from the time range that is defined for the page.

Opening the View

To see this view in the Performance Center Console, go to one of the following locations:

- (CA PC) **Infrastructure Overview**, **Network Overview**, or **Summary** context custom dashboard
- (NPC) **Enterprise**, **Traffic Analysis**, **Network Overview**, and custom dashboards

Available Actions

You can perform several actions in this view, including the following:

- Change the view name by editing the view settings.
- Display details in a Tooltip by holding your cursor over a bar.
- Click a name or bar to open related views in the NFA console.

Find the Comparable View in the NFA Console

The **Top Enterprise Hosts by Volume** view is similar to the **Top Hosts** view on the **Enterprise Overview** page in the NFA console.

Top Enterprise Protocols by Volume

The **Top Enterprise Protocols by Volume** view in the Performance Center Console shows the protocols with the highest volume of network traffic across the enterprise.

The view includes the following information for a maximum of 10 protocols that are associated with the highest traffic during the reporting period:

- **Protocol**
  Identifies the protocol by its keyword (Y-Axis).
- **Volume**
  Measures the total amount of data associated with the protocol expressed in a scale that is appropriate for the highest-volume protocol (X-Axis).

Performance Center views show the data from the time range that is defined for the page.

Opening the View

To see this view in the Performance Center Console, go to one of the following locations:

- (CA PC) **Infrastructure Overview** or **Network Overview** dashboard; **Summary** context view in a custom dashboard
- (NPC) **Enterprise**, **Traffic Analysis**, **Network Overview**, and custom dashboards

Available Actions

You can perform several actions in this view, including the following ones:

- Change the view name in the view settings.
- Display details in a Tooltip by holding your cursor over a bar.
- Click a name or bar to open related views in the NFA console.

Find the Comparable View in the NFA Console

The **Top Enterprise Protocols by Volume** view in the Performance Center Console is similar to the **Top Protocols** view on the **Enterprise Overview** page in the NFA console.

Top IP Interface Utilization (Flow)

The **Top IP Interface Utilization (Flow)** views in the Performance Center Console show the high-utilization interfaces from across the enterprise.

The view includes the following information for a maximum of 10 top interfaces during the reporting period:

- **Name**
  Identifies the interface by its device name/interface name (Y-Axis).
- **Percent (Utilization)**
  Measures the percentage of interface capacity that was used (X-Axis). The view shows the utilization of either inbound or outbound capacity.

Performance Center views show the data from the time range that is defined for the page.

Opening the View

To see this view in the Performance Center Console, go to one of the following locations:

- (CA PC) **Infrastructure Overview** and **Management Overview** dashboards; **Summary-type** view in a custom dashboard
- (NPC) **Traffic Analysis** and custom dashboards

Available Actions

You can perform several actions in this view, including the following ones:

- Change the data direction, view name, and context (the interfaces that are used) by editing the view settings.
- (NPC) Change the utilization thresholds.
- Display details in a Tooltip by holding your cursor over a bar.
- Click a name or bar to open related information on the Interface context pages.

Find the Comparable View in the NFA Console

The **Top IP Interface Utilization (Flow)** view in the Performance Center Console is similar to the **Interface Utilization** view on the **Enterprise Overview** page in the NFA console.

Top Flows by Volume

The **Top Flows by Volume** views in the Performance Center Console show the interfaces across the enterprise that have the highest volume of inbound or outbound traffic.

The view shows the following information for a maximum of 10 top interfaces:

- **Name**
  Identifies the interface by its device name (such as its router name), followed by a colon (:) and the interface name (Y-Axis).
- **Volume**
  Measures the volume of flow data on the interface (X-Axis) expressed in a scale that is appropriate for the highest-volume interface.

Performance Center views show the data from the time range that is defined for the page.

Opening the View

To see this view in the Performance Center Console, go to one of the following locations:

- (CA PC) **Infrastructure Overview** and **Management Overview** dashboards; **Summary** context view in a custom dashboard
- (NPC) Custom dashboard

Available Actions

You can perform several actions in this view, including the following ones:

- Change the data direction or the view name by editing the view settings.
- Display details in a Tooltip by holding your cursor over a bar. The Tooltip identifies the interface position among the top 10, with Interface 0 as the one with the highest traffic volume.
- Click a name or bar to open related information on the Interface context pages.

Find Flow Volume Data in the NFA Console

To see the flow volume of multiple top interfaces in the NFA console, create and run a Custom report. For example, you can view flow volume for the top interfaces in summary pie charts, summary tables, trend charts, and stacked trend charts.

To see the flow volume of a single top interface in the NFA console, drill into details from the **Enterprise Overview** page:

1. Click an interface name or bar in one of the **Top Interfaces** views on the **Enterprise Overview** page.
2. Select **Flows** from the list labeled **For this interface, show me** on the **Interface** page that opens.
3. Click the gray bar on the left edge of the page to change the presentation mode.
4. Click **Volume** in the **Presentation** menu that opens.

The **Flows** views display a trend chart of inbound flow volume and outbound flow volume.
To jump to the Performance Center **Interface Pages** data for the selected interface, click the arrow next to the **Flows** title and select **CA PC/NPC Interface Performance**.

Interfaces Over Threshold

The **Interfaces Over Threshold** view in the Performance Center Console lists the most heavily used interfaces throughout the enterprise. A table summary shows the interfaces with utilization that exceeds the configured thresholds.

The **Interfaces Over Threshold** view shows the interfaces whose traffic exceeded the configured thresholds during the reporting period. The view includes the following information for up to ten top interfaces:

- **Status**
  Identifies the interface status as Critical (Red - Meets or exceeds the user-defined Critical threshold) or Warning (Orange - Meets or exceeds the user-defined Warning threshold).
- **Interface Name**
  Identifies the interface by its name. (Depending on the application setting for the name format, the name may be prefixed by the device name.)
- **Traffic Direction**
  Shows whether the data was inbound or outbound on the interface.
- **Speed**
  (CA PC) Records the data speed that is defined for the interface.
- **Average Utilization**
  Measures the average percentage of interface capacity that was used.
- **Percent Time Critical**
  Shows the percentage of the reporting period the interface met or exceeded the Critical threshold.
- **Percent Time Warning**
  Shows the percentage of the reporting period the interface met or exceeded the Warning threshold.

Performance Center views show the data from the time range that is defined for the page.

Opening the View

To see this view in the Performance Center Console, go to one of the following locations:

- (CA PC) **Infrastructure Overview** dashboard; **Summary** context view in a custom dashboard
- (NPC) **Enterprise**, **Traffic Analysis**, **Routers/Switches Overview**, or custom dashboard

Available Actions

You can perform several actions in this view, including the following ones:

- Change the thresholds, view name, and utilization settings as described in this topic.
- (CA PC) Change the columns that are shown in the table: Click near a column border, click **Columns**, then choose the columns to display.
- Click an interface name to open the Interface context pages. You can review details or open additional views of interface data.

How to Change the View Settings

Follow these steps:

1. Open the dialog for editing the view:
   – (CA PC) Click the **Edit** icon
   
   in the view title bar and click **Edit**.
   – (NPC) Click the arrow next to the title name and select **Edit** from the menu.

The dialog opens.

2. (Optional) Edit the text in the **Title** field to change the name in the view title bar.
3. (Optional) Edit the thresholds by changing any of the following values in the **Interfaces Over Threshold Settings** section:
   - **Critical - % Utilization**: Specify the utilization percentage for flagging interfaces with a status of Critical, the highest level of concern. If the utilization for an interface has met or exceeded this percentage, it is marked with a red (Critical) status symbol.
   - **Warning - % Utilization**: Specify the utilization percentage for flagging interfaces with a status of Warning. If the utilization for an interface has met or exceeded this percentage, but has not met the Critical threshold, the interface is marked with an orange (Warning) status symbol.
   - **Affected % of reporting period**: Specify the percentage of the reporting period that a utilization percentage must be violated in order for the threshold to be met.
     For example, if the **Affected % of reporting period** value is 25, the threshold is met for the interfaces that have a utilization level at or above the threshold level during 25% of the reporting period. With the default reporting period of 24 hours, the list includes interfaces at or above the threshold value for six hours or more during the previous 24 hours.
4. (Optional) (NPC) Define a new context to filter the interfaces that can appear in the view: Select **Filter by value** and select a context type and setting in the **Select Context** dialog.
   Interfaces that are not in the selected group do not appear in the view, even if they violate a threshold. If you select a group, the defined context appears under the view title.
5. (Optional) Specify which users are affected by the settings: Select a value from the **Apply Changes** list:
   - **For All Tenant Users**: Saves the changes so that they are only available to users associated with your tenant (possibly the Default Tenant).
   - **My User Account**: Saves the changes to your user account as a default for this view.
   - **My Current Session**: Reverts the changes when you log out.
6. Click **Save** to save your changes, **Cancel** to exit without saving changes, or **Use Defaults** to restore the default values.
   The dialog closes and the view reflects your changes.

Find the Comparable View in the NFA Console

The **Interfaces Over Threshold** view in the Performance Center Console is similar to the **Interface Utilization** view on the **Enterprise Overview** page in the NFA console.

Routers with the Most Flow Traffic

The **Routers with the Most Flow Traffic** view in the Performance Center Console displays the routers in your network that have the highest traffic. Traffic use is measured for both inbound and outbound traffic during the reporting period, as reported by DX NetOps.

The view includes the following information for a maximum of 10 routers:

- **Name**
  Consists of the router IP address and device name (Y-Axis). If an administrator defined an alias for the device item, the alias is displayed. Otherwise, the discovered device name is displayed.
- **Volume**
  Measures the total amount of traffic for the router expressed in megabytes, for example (X-Axis).

Performance Center views show the data from the time range that is defined for the page.

Opening the View

To see this view in the Performance Center Console, go to one of the following locations:

- (CA PC) **Infrastructure Overview** and **Router/Switch Capacity Watch Lists** dashboards; Summary-type view in a custom dashboard
- (NPC) Custom dashboard

Available Actions

You can perform several actions in this view, including the following ones:

- Change the view name by editing the view settings.
- (NPC) Change the context (the routers that can be used in the view).
- Display details in a Tooltip by holding your cursor over a bar.
- Click a router bar to view details in the Performance Center **Router** pages.

Calendar Chart (Flow)

The **Calendar Heat Chart** (Flow) view maps the utilization percentage of the selected interface over time.

This view makes it easy to detect recurring data patterns. Finding a pattern can help you identify the source of high traffic rates and potential performance issues. You might discover that the high traffic rates you thought were intermittent actually follow a pattern. The view can show the hour of each day when utilization is the highest, for example.

Each color represents a severity range that is calculated as a percentage of total capacity. High utilization is shown in orange and red. Low utilization is shown in green and blue.

The view includes the following information:

- **Identifier**
  Consists of the router name, interface name, and interface description (under the view title). The interface description consists of the ifDescr value by default, so it may be slightly different than the interface description that is shown in the NFA console.
  (NPC) The identifier line also includes the interface speed.
- **Month, Date, and Day of the Week**
  Denote the day that the traffic occurred (X-Axis columns).
- **Hour**
  Denotes the hour of the day that the traffic occurred (Y-Axis).

Opening the View

To see the **Calendar Heat Chart** view in the Performance Center Console, add it to a custom dashboard.

Opening the View

To see this view in the Performance Center Console, go to one of the following locations:

- (CA PC) Custom dashboard
- (NPC) **Interface** pages (with an interface selected): **Custom** tab

Available Actions

You can perform several actions in this view, including the following ones:

- Change the data direction and view name as described in this topic.
- (CA PC) Display details in a Tooltip by holding your cursor over a cell.
- (CA PC) Click **Show All** and choose a pattern-matching filter. For example, select **Busy Hour** to show only the data for the busiest hour of each day.

How to Change the View Settings

1. Open the dialog for editing the view:

– 
(CA PC) Click the **Edit** icon in the view title bar and click **Edit**.
– (NPC) Click the arrow next to the title name and select **Edit** from the menu.
The dialog opens.
2. (Optional) Edit any of the following settings in the **Calendar Heat Chart (Flow) Settings** section:
   – **Title**: Change the name that appears in the view title bar.
   – (CA PC) **Time Display Format**: Select the time format for the chart, either 12 hours or 24 hours.
   – (CA PC) **Zone Start**: Set the starting value of each heat zone. The defaults are based on IT industry standards for performance. For example, the default Red Zone Start value is 70 percent utilization.
   Defaults: Green Zone Start = 0, Yellow Zone Start = 50, Orange Zone Start = 60, Red Zone Start = 70.
   – (CA PC) **Business Week Start**: Select the day that starts the business week.
   Default: Monday.
   – (CA PC) **Direction Settings**: Select the direction of traffic on the selected interface to include in the report:
     • **Out**: Outbound on the interface.
     • **In**: Inbound on the interface.
     • **Total**: Combination of inbound and outbound traffic.
3. (Optional) (CA PC) Change the context for the view data: Select a different interface from the **Context Settings** table.
4. (Optional) Specify which users are affected by the setting changes: Select a value from the **Apply Changes** list:
   – **Default for All Users**: Saves the changes to all user accounts as a default for this view.
   – **For All Tenant Users**: Saves the changes so that they are only available to users associated with your tenant.
   – **My User Account**: Saves the changes to your user account as a default for this view.
   – **My Current Session**: Reverts the changes when you log out.
5. Click **Save** to save your changes.
   The settings dialog closes. The view refreshes to reflect your updates.

Find the Comparable View in the NFA Console

To display Calendar Chart data for an interface in the NFA console, select an interface on the **Interface** page and select the following options:

- **Report type**: **Utilization**
- **Presentation** menu option: **Direction In** or **Direction Out**

To display Flow Forensics-level detail, click the **Flow Forensics** link and run a Flow Forensics report.

Interface: Stacked Trend Charts

The **Stacked Trend** views show the top protocol or ToS values that are used for traffic on the currently selected interface. The views are described in the topics that follow.

Stacked Protocol Trend

The **Stacked Protocol Trend** views in the Performance Center Console show the protocols that are used the most heavily for traffic on the selected interface. The views also show when the traffic occurred. A timeline of rates is included for each listed ToS value. You can configure the view to display rate, utilization, or volume information.

The views include the following information:

- **Identifier**
  Identifies the interface that is used for the report. The identifier string consists of the router name, interface name, and interface description (under the view title).

(NPC) The identifier line also includes the interface speed.

- **Protocol Bands**
  Show the data rate, the data volume, or the interface capacity utilization for each top protocol that is associated with traffic on the interface.
- **Time (All Views)**
  Point in time during data transmission--expressed in hours and minutes (X-Axis).
- **Measurement Setting:**
- **Rate:** Data rate at each point in time expressed in kilobits per second, for example (Y-Axis). The rate is calculated by dividing the data volume by the elapsed transmission time.
- **Bytes** (Volume): Data volume at each point in time expressed in kilobytes, for example (Y-Axis).
- **Percent** (Utilization): Percentage of the total interface capacity that the protocol uses (Y-Axis). The utilization percentage is calculated by dividing the data rate by the data speed.

Depending on the data direction, the view shows inbound, outbound, or total data on the interface.

- **Legend**
  Identifies the protocol for each color band by protocol keyword and tcp/udp port (bottom of the view).

Performance Center views show the data from the time range that is defined for the page.

Opening the Views

To see these views in the Performance Center Console, go to one of the following locations:

- (CA PC) Interface Pages (with an interface selected): Custom dashboard; **IP Performance** and **CBQoS** tabs
- (NPC) Interface Pages (with an interface selected): **Interface Capacity**, **Interface QoS**, and custom tabs

> **NOTE**
> You can add Multi-Interface Stacked Protocol Trend views to a custom dashboard or to a custom tab in the Interface pages in the CA NetQoS Performance Center Console. This view consists of a group of interface-specific stacked protocol trend charts

Available Actions

You can perform several actions in this view, including the following ones:

- Change the traffic direction, the type of measurement (**Rate**, **Volume**, or **Utilization**), and the view name by editing the view settings. If the view is on a custom interface context dashboard in the CA Performance Center Console, you can change the interface.
- (CA PC) Zoom in to narrow the time frame.
- (CA PC) Display only the data for a single protocol: Right-click a protocol in the legend at the bottom of the view and click **Focus**. This menu is available for a view that has multiple protocols. (This option is active when the legend contains multiple protocols.)
- (CA PC) Hide data for one of multiple protocols: Right-click a protocol in the legend at the bottom of the view and click Hide.
- (CA PC) Position your cursor over legend items to display explanatory Tooltips.
- Jump to details on an NFA console Interface page by double-clicking a protocol in the legend.
- (NPC) Jump to details on the corresponding Interface page by double-clicking a protocol band in the view. To choose a destination tab on the Interface page, right-click the protocol band and select a tab from the menu.

Find Protocol Trend Data in the NFA Console

You can display protocol volume in the NFA console in trend charts or stacked trend charts for a selected interface:

- *Overview* -- Report type: Overview. Presentation menu options: Mixed Chart; Volume.
  Views: Stacked Protocol Trend (In and Out) for the Top N Protocols, plus other overview views.
- *Top N Protocols, Stacked Trends* -- Report type: Protocols. Filter: Top N Protocols. Presentation menu options: Stacked Trend Chart; Volume.

Views: Stacked Trend for the Top N Protocols (In, Out, and Total).

- *Top N Protocols, Trends* -- Report type: Protocols. Filter: Top N ToS. Presentation menu options: Trend Chart; Volume. Views: Trend (In, Out, and Total) for each of the Top N Protocols.
- *Single Protocol* -- Report type: Protocols. Filter: Single protocol.
  Views: (Depending on the selected report subtype): trends, stacked trends, trend summaries, and multi-trend summaries for protocols, protocol hosts, and protocols in conversations.

To display Flow Forensics-level detail, click the **Flow Forensics** link and run a Flow Forensics report.

Stacked ToS Trend

The **Stacked ToS Trend** views show the interface traffic for the top ToS, including the time the traffic occurred.

A timeline of rates is included for each ToS value. You can configure the view to display rate, utilization, or volume information.

The view includes the following information:

Identifier

Identifies the interface that is used for the report. The identifier string consists of the router name, interface name, and interface description (under the view title).
(NPC) The identifier line also includes the interface speed.

ToS Bands

Show the data rate, data volume, or interface capacity utilization for each top ToS that is associated with traffic on the interface.

Time

Point in time during data transmission expressed in hours and minutes (X-Axis).

Measurement Setting:

- **Rate:** Data transfer rate at each point in time expressed in kilobits per second or a rate that is appropriate for the highest-volume ToS (Y-Axis). The rate is calculated by dividing the data volume by the elapsed transmission time.
- **Bytes** (Volume): Data volume at each point in time expressed in a scale that is appropriate for the highest-volume ToS (Y-Axis).
- **Percent** (Utilization): Percentage of the total interface capacity that the ToS traffic uses (Y-Axis). The utilization percentage is calculated by dividing the data rate by the data speed.

Depending on the data direction, the view shows inbound, outbound, or total data on the interface.

- **Legend**
  Identifies the ToS for each color band by ToS number and label (bottom of the view).

Performance Center views show the data from the time range that is defined for the page.

Opening the View

To see this view in the Performance Center Console, go to one of the following locations:

- (CA PC) Custom dashboard; **Interface** pages (with an interface selected): **CBQoS** tab
- (NPC) **Interface** pages (with an interface selected): **Interface QoS** and custom tabs

Note: You can add Multi-Interface Stacked ToS Trend views to a custom dashboard or to a custom tab in the Interface pages of the CA NetQoS Performance Center Console. This view consists of a group of interface-specific stacked ToS trend charts.

Available Actions

You can perform several actions in this view, including the following ones:

- Change the traffic direction (In, Out, or Total), the type of measurement (Rate, Volume, or Utilization), and the view name by editing the view settings. If the view is on a custom interface context dashboard in the CA Performance Center Console, you can change the interface.
- (CA PC) Zoom in to narrow the time frame.
- (CA PC) Display only the data for a single ToS: Right-click a ToS in the legend at the bottom of the view and click **Focus**. This menu is available for a view that has multiple ToS values. (This option is active when the view contains multiple ToS values.)
- (CA PC) Hide data for one of multiple ToS values: Right-click a ToS in the legend at the bottom of the view and click **Hide**.
- (CA PC) Position your cursor over legend items to display explanatory Tooltips.
- Jump to details on an NFA console **Interface** page by double-clicking a ToS value in the legend.

Find ToS Trend Data in the NFA Console

You can display ToS volume in trend charts or stacked trend charts in the NFA console for a selected interface:

- *Overview* -- Report type: Overview. Presentation menu options: Mixed Chart or Mixed Trend; Volume.
  Views: Stacked ToS Trend (In and Out) for the Top N ToS, plus other overview views.
- *Top N ToS, Stacked Trends* -- Report type: ToS. Filter: Top N ToS. Presentation menu options: Stacked Trend Chart; Volume.
  Views: Stacked Trend for the Top N ToS (In, Out, and Total).
- *Top N ToS, Trends* -- Report type: ToS. Filter: Top N ToS. Presentation menu options: Trend Chart; Volume.
  Views: Trend (In, Out, and Total) for each of the Top N ToS.
- *Single ToS, Stacked Trends/Trends* -- Report type: ToS. Filter: Single ToS value. Presentation menu options: Mixed Trend; Volume.
  Views: Trend (In and Out with baselines), Stacked ToS Trend (In and Out).
- *Single ToS, Trends* -- Report type: ToS. Filter: Single ToS value. Presentation menu options: Mixed Chart; Volume.
  Views: Stacked ToS Trend (In and Out).
- *Conversation* -- Report type: Conversations. Filter: Single conversation source and destination. Report subtype: Protocols. Presentation menu options: Volume.
  Views: Conversation Trend (maximum of 7 views for different timespans).

To display Flow Forensics-level detail, click the **Flow Forensics** link and run a Flow Forensics report.

Interface: ToS Summaries

The **ToS Summary** views show the Type of Service (ToS) values for traffic on the selected interface. The views are described in the topics that follow.

ToS Summary (Pie)

The **ToS Summary (Pie)** view shows an overview of the Type of Service (ToS) values for traffic on the selected interface.

The view includes a pie chart and table of information about the high-volume ToS values in use on the selected interface. The table includes the following information by default:

- **Identifier**
  Identifies the interface that is used for the report. The identifier string consists of the router name, interface name, and interface description (under the view title).
  (NPC) The identifier line also includes the interface speed.
- **Type of Service**

Name of the ToS values associated with high-volume traffic, identified by number and label.

- **Total**
  Shows the total data volume for the reporting period.
- **Percent**
  (CA PC) Lists the percentage of the total data volume for the Top N ToS.

Performance Center views show the data from the time range that is defined for the page.

Opening the Views

To see these views in the Performance Center Console, go to one of the following locations:

- (CA PC) **Interface** pages (with an interface selected): **IP Performance** tab; **Custom** dashboard
- (NPC) **Interface** pages (with an interface selected): **Custom** tab

Available Actions

You can perform several actions in this view:

- Change the traffic direction and view name by editing the view settings. If the view is on a custom interface context dashboard in the CA Performance Center Console, you can change the interface.
- (CA PC) Change the type of measurement.
- Jump to details on an NFA console **Interface** page by double-clicking a ToS name.

Find ToS Summary Pie Charts in the NFA Console

You can display pie charts of ToS summary data in the NFA console for a selected interface:

- *Overview* -- Report type: Overview. Presentation menu option: Pie Chart.
  View: ToS Summary (In and Out) for the Top N ToS.
- *Top N ToS Summary* -- Report type: ToS. Filter: Top N ToS. Presentation menu option: Pie Chart.
  View: ToS Summary (In, Out, and Total) for the Top N ToS.
- *Single ToS Summaries* -- Report type: ToS. Filter: Single ToS. Report subtype: Overview. Presentation menu option: Pie Chart.
  Views: ToS Protocol Summary (In and Out) for the single ToS; ToS Hosts Summary (From and To) for the single ToS; ToS Conversations Summary (Total) for the single ToS.

  NOTE
  You can view additional versions of the summary pie charts by selecting **Protocols**, **Hosts**, or **Conversations** as the report subtype.

To display Flow Forensics-level detail, click the **Flow Forensics** link and run a Flow Forensics report.

ToS Summary (Table)

The **ToS Summary (Table)** views show rate, volume, or utilization for the top ToS values of the traffic on a particular interface. You can use this information to compare traffic for each of the top ToS values.

An interface identification string is shown under the view title. The table contains a row for each ToS with the Type of Service identifier (EF/AF, DSCP, and ToS values) and the following rate, volume, or utilization information:

- **Rate**:
  - (CA PC/NPC) Average rate of total, inbound, and outbound data for each ToS (**Average Total**, **Average Out**, and **Average In**)
  - (CA PC) Maximum rate of data that is outbound or inbound on the interface for each ToS (**Maximum Out** and **Maximum In**)

The rate is calculated by dividing the data volume by the elapsed transmission time.

- **Volume**: Volume of outbound, inbound, and all data for each ToS (Out, In, and Total), expressed in a scale that is appropriate for the highest-volume ToS.
- **Utilization**:
  - (CA PC/NPC) Average utilization of total, outbound, and inbound data that each ToS consumes (**Average Total**, **Average In**, and **Average Out**)
  - (CA PC) Maximum percentage of interface capacity that the outbound or inbound utilizes for each ToS (**Maximum Out** and **Maximum In**)

  The utilization percentage is calculated by dividing the data rate by the data speed.

Performance Center views show the data from the time range that is defined for the page.

Opening the View

To see this view in the Performance Center Console, go to one of the following locations:

- (CA PC) **Custom** dashboard
- (NPC) **Interface** pages (with an interface selected): **Interface QoS** or custom tab

Available Actions

You can perform several actions in this view, including the following ones:

- Change the type of measurement (Rate, Volume, or Utilization) and the view name by editing the view settings.
- (CA PC) Change the interface.
- Re-sort the table data by clicking a column heading. Click again to toggle between descending and ascending order.
- Change the **Max Per Page** value to show more or fewer items on each table page.
- (CA PC) Change the columns that are shown in the table: Click near a column border, click **Columns**, then choose the columns to display.
- Click a **Type of Service** link to display more information about the ToS on Interface report pages in the NFA console.

Find ToS Summary Tables in the NFA Console

You can display ToS summary tables in the NFA console for a selected interface:

- *Top N ToS Summary* -- Report type: ToS. Filter: Top N ToS. Presentation menu options: Summary Table; Volume.
  View: ToS Summary Table for the Top N ToS.
- *Protocol Summary for a Single ToS* -- Report type: ToS. Filter: Single ToS. Report subtype: Protocols. Subtype filter: Top N Protocols. Presentation menu options: Summary Table; Volume.
  Views: ToS Protocol Summary Table for the single ToS.
- *Host Summary for a Single ToS* -- Report type: ToS. Filter: Single ToS. Report subtype: Hosts. Subtype filter: Top N Hosts. Presentation menu options: Summary Table; Volume.
  Views: ToS Hosts Summary Table for the single ToS.
- *Conversation Summary for a Single ToS* -- Report type: ToS. Filter: Single ToS. Report subtype: Conversations. Subtype filter: Top N Conversations. Presentation menu options: Summary Table; Volume.
  Views: ToS Conversations Summary Table for the single ToS.

To display Flow Forensics-level detail, click the **Flow Forensics** link and run a Flow Forensics report.

Interface: Top Conversations

The **Top Conversations** views show the conversations that generate the highest traffic on the currently selected interface. The views are described in the topics that follow.

Top Conversations (Bar)

The **Top Conversations (Bar)** views show the conversations that have the highest traffic on the selected interface. A bar graph shows the volume for each conversation.

For example, use conversation information to determine the IP addresses of high-volume hosts. Contact the host owners or users to investigate the nature and purpose of the traffic.

You can view the conversations for incoming data, outgoing data, or all data.

The view includes a bar for each top conversation on the selected interface. A maximum of 10 conversations are shown. The view includes the following information:

- **Identifier**
  Identifies the interface that is used for the report. The identifier string consists of the router name, interface name, and interface description (under the view title).
  (NPC) The identifier line also includes the interface speed.
- **Conversation Pair**
  Identifies the conversation source and destination servers by their names (the fully qualified DNS names, if they are available), followed by the IP addresses (Y-Axis).
- **Volume**
  Measures the total amount of data that was exchanged in the conversation expressed in a scale that is appropriate for the highest-volume conversation (X-Axis).

Performance Center views show the data from the time range that is defined for the page.

Opening the View

To see this view in the Performance Center Console, go to one of the following locations:

- (CA PC) Custom dashboard
- (NPC) **Interface** pages (with an interface selected): **Interface Capacity** or custom tab

Available Actions

You can perform several actions in this view, including the following ones:

- Change the view name by editing the view settings.
- (CA PC) Change the traffic direction and the interface.
- Display details in a Tooltip by holding your cursor over a bar.
- Jump to conversation details on an NFA console **Interface** report page by clicking a bar or name.

Find Conversation Data in the NFA Console

You can display conversation volume trend charts in the NFA console for any interface you have selected:

- *Overview Multi-Trend* -- Report type: Overview. Presentation menu options: Mixed Trend; Volume.
  View: Conversations Multi Trend Summary (Total) for the Top N Conversations, plus other views.
- *Top N Conversations Trend* -- Report type: Conversations. Filter: Top N Conversations. Presentation menu options: Trend Chart; Volume.
  View: Conversations Trend for the Top N Conversations.
- *Conversations for a Single Protocol* -- Report type: Protocols. Filter: Single protocol. Report subtype: Conversations. Conversation Filter: Top N Conversations. Presentation menu option: Trend Chart.
  View: Protocol Conversations Summary (Total) for a single protocol.
- *Conversations for a Single ToS* -- Report type: ToS. Filter: Single ToS. Report subtype Conversations. Conversation Filter: Top N Conversations. Presentation menu options: Trend Chart; Volume.
  View: ToS Trend view for each conversation that uses the single ToS.

Note: To see trend charts for a single conversation, click **Top N Conversations** and select a single conversation as the filter.

To display Flow Forensics-level detail, click the **Flow Forensics** link and run a Flow Forensics report.

Top Conversations (Pie)

The **Top Conversations (Pie)** view includes a pie chart of the conversations that account for the most traffic on the selected interface.

The view includes a pie chart and table of information about the high-volume conversations on the selected interface. A text string near the top of the view identifies the interface whose data is displayed. The table includes the following information by default:

- **Identifier**
  Identifies the interface that is used for the report. The identifier string consists of the router name, interface name, and interface description (under the view title).
  (NPC) The identifier line also includes the interface speed.
- **Source - Destination Name**
  Identifies the host servers that initiated and received the conversation data by their fully qualified DNS names (if available) and IP addresses.
- **Total**
  Shows the total amount of data in the conversation expressed in a scale that is appropriate for the highest-volume conversation.
- **Percent**
  (CA PC) Records how much the conversation consumes out of the total traffic that is displayed.

Performance Center views show the data from the time range that is defined for the page.

Opening the Views

To see these views in the Performance Center Console, go to one of the following locations:

- (CA PC) Custom dashboard; **Interface** pages (with an interface selected): **IP Performance** tab
- (NPC) **Interface** pages (with an interface selected): **Interface QoS** or custom tab

Available Actions

You can perform several actions in this view, including the following ones:

- Change the view name by editing the view settings.
- (CA PC) Change the traffic direction and the type of measurement. If the view is on a custom interface context dashboard in the CA Performance Center Console, you can change the interface.
- (CA PC) Change the columns that are shown in the table: Click near a column border, click **Columns**, then choose the columns to display.
- Jump to details on an NFA console **Interface** page by clicking a link.

Find Conversation Pie Charts in the NFA Console

You can display conversation pie charts in the NFA console for any interface you have selected:

- *Overview* -- Report type: Overview. Presentation menu option: Pie Chart.
  View: Conversations Summary (Total) for the Top N Conversations, plus other overview views.
- *Top N Conversations* -- Report type: Conversations. Filter: Top N Conversations. Presentation menu option: Pie Chart.
  View: Conversations Summary (Total) for the Top N Conversations.
- *Conversations for a Single Protocol* -- Report type: Protocols. Filter: Single protocol. Report subtype: Conversations or Overview. Conversations Filter: Top N Conversations. Presentation menu option: Pie Chart or Mixed Chart.
  View: Protocol Conversations Summary (Total) for a single protocol.
- *Conversations for a Single ToS* -- Report type: ToS. Filter: Single ToS. Report subtype: Conversations. Conversation Filter: Top N Conversations. Presentation menu option: Pie Chart.
  View: ToS Conversations Summary (Total) for a single ToS.

To display Flow Forensics-level detail, click the **Flow Forensics** link and run a Flow Forensics report.

Top Conversations (Table)

The **Top Conversations (Table)** views show data for the top highest-volume conversations on a particular interface. The maximum number of top conversations shown is ten.

The table contains a row for each conversation with the source and destination - The fully qualified DNS host name (if available) and IP address of the servers that initiated and received the conversation data. The table also contains the following rate, volume, or utilization information:

- Rate:
  - (CA PC/NPC) For each conversation, the average rate of total data (Average Total), data that goes to the destination host (Average To), and data that comes from the source host (Average From).
  - (CA PC) For each conversation, maximum rate of data that comes from the source host (Maximum From) and goes to the destination host (Maximum To).

  The rate is calculated by dividing the data volume by the elapsed transmission time.
- Volume: For each conversation, total amount of data (Total), data that comes from the source host (From), and data that goes to the destination host (To), expressed in a scale that is appropriate for the highest-volume conversation.
- Utilization:
  - (CA PC/NPC) For each conversation, average utilization by data that comes from the source host (Average From), data that goes to the destination host (Average To), and total data (Average Total).
  - (CA PC) For each conversation, maximum percentage of interface capacity that is used by the data that comes from the source host (Maximum From) or that goes to the destination host (Maximum To).

  The utilization percentage is calculated by dividing the data rate by the data speed.

Performance Center views show the data from the time range that is defined for the page.

Opening the View

To see this view in the Performance Center Console, go to one of the following locations:

- (CA PC) Custom dashboard
- (NPC) **Interface** pages (with an interface selected): Custom tab

Available Actions

You can perform several actions in this view, including the following ones:

- Change the type of measurement (Rate, Volume, or Utilization) and the view name by editing the view settings.
- (CA PC) Change the interface.
- (CA PC) Change the columns that are shown in the table: Click near a column border, click **Columns**, then choose the columns to display.
- Click one of the links to jump to a pre-filtered Interface page report in the NFA console.

Find Conversation Tables in the NFA Console

You can display tables with conversation volumes in the NFA console for a selected interface:

- *Top N Conversations* -- Report type: Conversations. Filter: Top N Conversations. Presentation menu option: Summary Table; Volume.
  View: Conversation Summary Table for the Top N Conversations.
- *Conversations for a Single Protocol* -- Report type: Protocols. Filter: Single protocol. Report subtype: Conversations. Subtype filter: Top N Conversations. Presentation menu option: Summary Table; Volume.
  View: Protocol Conversation Summary Table for a single protocol.
- *Conversations for a Single ToS* -- Report type: ToS. Filter: Single ToS. Report subtype: Conversations. Subtype filter: Top N Conversations. Presentation menu options: Summary Table; Volume.

View: ToS Conversations Summary Table for a single ToS.

To display Flow Forensics-level detail, click the **Flow Forensics** link and run a Flow Forensics report.

Interface: Top Hosts

The **Top Hosts** views show the hosts that generate the highest traffic on the currently selected interface. The views are described in the topics that follow.

Top Hosts (Bar)

The **Top Hosts (Bar)** views show the top high-volume hosts for a particular interface. You can use this view to determine the IP addresses of hosts that are responsible for high volumes of network traffic. You can then contact the owner or user of each host to investigate the nature and purpose of the traffic.

You can view the hosts for incoming flows, outgoing flows, or all flows.

The bar chart includes the following information for a maximum of 10 hosts:

- **Identifier**
  Identifies the interface that is used for the report. The identifier string consists of the router name, interface name, and interface description (under the view title).
  (NPC) The identifier line also includes the interface speed.
- **Host Name**
  Identifies the host server by its fully qualified DNS name (if available) and IP address (Y-Axis).
- **Volume**
  Measures the total amount of data for the host on the interface, expressed in a scale that is appropriate for the highest-volume host (X-Axis).

Performance Center views show the data from the time range that is defined for the page.

Opening the View

To see this view in the Performance Center Console, go to one of the following locations:

- (CA PC) Custom dashboard
- (NPC) **Interface** pages (with an interface selected): **Interface Capacity** or custom tab

Available Actions

You can perform several actions in this view, including the following ones:

- Change the traffic direction and the view name by editing the view settings.
- (CA PC) Change the interface.
- Display details in a Tooltip by holding your cursor over a bar.
- Jump to details for a specific host on an NFA console Interface page by clicking a bar.

Find Host Trend Views in the NFA Console

The **Enterprise Overview** page in the NFA console displays traffic volume for the top hosts in a bar chart.

You also can display host volume in trend charts for a selected interface:

- *Overview* -- Report type: Overview. Presentation menu options: Mixed Trend; Volume.
  View: Hosts Multi Trend Summary (From and To) for the Top N Hosts, plus other overview views.
- *Top N Host Trend* -- Report type: Hosts. Filter: Top N Hosts. Presentation menu options: Trend Chart; Volume.
  View: Host Trend for each of the Top N Hosts.

Note: To see trend charts for a single host, click **Top N Hosts** and select a host as the filter.

To display Flow Forensics-level detail, click the **Flow Forensics** link and run a Flow Forensics report.

Top Hosts (Pie)

The **Top Hosts (Pie)** views show the hosts that account for the highest volumes of network traffic on the selected interface.

The table includes the following information by default:

- **Identifier**
  Identifies the interface that is used for the report. The identifier string consists of the router name, interface name, and interface description (under the view title).
  (NPC) The identifier line also includes the interface speed.
- **Host Name**
  Identifies the host server by its fully qualified DNS name (if available) and IP address.
- **Total**
  Records the total amount of data for the host on the interface, expressed in a scale that is appropriate for the highest-volume host.
- **Percent**
  (CA PC) Records how much the host consumes out of the total traffic that is displayed.

Performance Center views show the data from the time range that is defined for the page.

Opening the Views

To see these views in the Performance Center Console, go to one of the following locations:

- (CA PC) Custom dashboard; **Interface** pages (with an interface selected): **IP Performance** tab
- (NPC) **Interface** pages (with an interface selected): **Interface QoS** or custom tab

Available Actions

You can perform several actions in this view, including the following ones:

- Change the traffic direction and the view name by editing the view settings. If the view is on a custom interface context dashboard in the CA Performance Center Console, you can change the interface.
- (CA PC) Change the columns that are shown in the table: Click near a column border, click **Columns**, then choose the columns to display.
- Jump to details on an Interface report page in the NFA console by clicking a host link in the view.

Find Host Pie Charts in the NFA Console

You can display pie charts with host volumes in the NFA console for a selected interface:

- *Overview* -- Report type: Overview. Presentation menu option: Pie Chart.
  View: Host Summary (From and To) for the Top N Hosts, plus other overview views.
- *Top N Hosts Summary* -- Report type: Hosts. Filter: Top N Hosts. Presentation menu option: Pie Chart.
  View: Host Summary (From, To, and Total) for the Top N Hosts.
- *Hosts for a Single Protocol* -- Report type: Protocols. Filter: Single protocol. Report subtype: Overview. Presentation menu option: Pie Chart.
  View: Protocol Hosts Summary (From, To, and Total) for the single protocol.

To display Flow Forensics-level detail, click the **Flow Forensics** link and run a Flow Forensics report.

Top Hosts (Table)

The **Top Hosts (Table)** views show rate, volume, or utilization for the hosts who exchange the highest volume of data on a particular interface.

The view contains an interface identification string and a table. The table contains a row for each host with the fully qualified DNS host name (if available) and IP address, as well as the following rate, volume, or utilization information (by default):

- Rate:
  - (CA PC/NPC) For each host, the average rate of total data (Average Total), data that goes to the host (Average To), and data that comes from the host (Average From).
  - (CA PC) For each host, maximum rate of data that comes from the host (Maximum From) and goes to the host (Maximum To).

  The rate is calculated by dividing the data volume by the elapsed transmission time.
- Volume: For each host, total amount of data (Total), data from the host (From), and data to the host (To), expressed in a scale that is appropriate for the highest-volume host.
- Utilization:
  - (CA PC/NPC) For each host, average utilization by data that comes from the host (Average From), data that goes to the host (Average To), and total data (Average Total).
  - (CA PC) For each host, maximum percentage of interface capacity that is used by the data from the host (Maximum From) or that goes to the host (Maximum To).

  The utilization percentage is calculated by dividing the data rate by the data speed.

Performance Center views show the data from the time range that is defined for the page.

Opening the View

To see this view in the Performance Center Console, go to one of the following locations:

- (CA PC) Custom dashboard
- (NPC) **Interface** pages (with an interface selected): Custom tab

Available Actions

You can perform several actions in this view, including the following ones:

- Change the type of measurement and the view name by editing the view settings.
- (CA PC) Change the interface.
- (CA PC) Change the columns that are shown in the table: Click near a column border, click **Columns**, then choose the columns to display.
- Click a name to jump to a pre-filtered Interface page report in the NFA console.

Find Host Tables in the NFA Console

You can display tables with host volumes in the NFA console for a selected interface:

- *Top N Hosts Summary* -- Report type: Hosts. Filter: Top N Hosts. Presentation menu option: Summary Table; Volume.
  View: Host Summary Table for the Top N Hosts.
- *Hosts for a Single Protocol* -- Report type: Protocols. Filter: Single protocol. Report subtype: Overview. Presentation menu option: Summary Table; Volume.
  View: Protocol Host Summary Table for the single protocol.
- *Hosts for a Single ToS* -- Report type: ToS. Filter: Single ToS. Report subtype: Hosts. Subtype filter: Top N Hosts. Presentation menu options: Summary Table; Volume.
  View: ToS Hosts Summary Table for the single ToS.

To display Flow Forensics-level detail, click the **Flow Forensics** link and run a Flow Forensics report.

Interface: Top Protocols

The **Top Protocols** views show the protocols associated with the highest traffic on the currently selected interface. The views are described in the topics that follow.

Top Protocols (Bar)

The **Top Protocols (Bar)** views show the top high-volume IP protocols for traffic on a particular interface. A bar chart shows which protocols account for the most traffic on the selected interface.

This view gives you an overall picture of how much data is associated with particular protocols--and, therefore, with applications--on the interface. The view also lets you determine whether the application protocols are related to business-critical processes, or are related to low-priority or non-business related processes such as unauthorized web use.

You can view protocol traffic for incoming flows, outgoing flows, or all flows.

The bar chart includes the following information for a maximum of 10 protocols:

- **Identifier**
  Identifies the interface that is used for the report. The identifier string consists of the router name, interface name, and interface description (under the view title).
  (NPC) The identifier line also includes the interface speed.
- **Protocol**
  Identifies the protocol by its descriptor (Y-Axis).
- **Volume**
  Measures the total amount of protocol data expressed in a scale that is appropriate for the highest-volume protocol (X-Axis).

Performance Center views show the data from the time range that is defined for the page.

Opening the View

To see this view in the Performance Center Console, go to one of the following locations:

- (CA PC) Custom dashboard
- (NPC) **Interface** pages (with an interface selected): Custom tab

Available Actions

You can perform several actions in this view, including the following ones:

- Change the traffic direction and the view name by editing the view settings.
- (CA PC) Change the interface.
- Display details in a Tooltip by holding your cursor over a bar.
- Jump to details for a specific protocol on an NFA console Interface page by clicking a bar or name.

Find the Comparable View in the NFA Console

The **Top Protocols** bar charts in the Performance Center Console are similar to the **Top Protocol** view on the **Enterprise Overview** page in the NFA console.

Top Protocols (Pie)

The **Top Protocols (Pie)** views show the protocols that are associated with the highest traffic volumes on the selected interface.

The table includes the following information by default:

- **Identifier**

Identifies the interface that is used for the report. The identifier string consists of the router name, interface name, and interface description (under the view title).
(NPC) The identifier line also includes the interface speed.

- **Protocol Name**
  Identifies the protocol by its keyword and TCP/UDP port assignment.
- **Total**
  Records the total volume of network traffic on the interface that is associated with the protocol
- **Percent**
  (CA PC) Records how much the protocol consumes out of the total traffic that is displayed.

Performance Center views show the data from the time range that is defined for the page.

Opening the Views

To see these views in the Performance Center Console, go to one of the following locations:

- (CA PC) Custom dashboard
- (NPC) Interface Pages (with an interface selected): **Interface QoS** or custom tab

Available Actions

You can perform several actions in this view, including the following ones:

- Change the traffic direction and the view name by editing the view settings.
- (CA PC) Change the interface.
- (CA PC) Change the columns that are shown in the table: Click near a column border, click **Columns**, then choose the columns to display.
- Jump to details on an NFA console **Interface** page by clicking a protocol name.

Find Protocol Pie Charts in the NFA Console

You can display pie charts with protocol traffic volumes in the NFA console for a selected interface:

- *Overview* -- Report type: Overview. Presentation menu option: Pie Chart.
  View: Protocol Summary (In and Out) for the Top N Protocols, plus other overview views.
- *Top N Protocol Summaries* -- Report type: Protocols. Filter: Top N Protocols. Presentation menu option: Pie Chart.
  View: Protocol Summary (In, Out, and Total) for the Top N Protocols.
- *Hosts or Conversations for Single Protocol* -- Report type: Protocols. Filter: Single protocol. Presentation menu option: Pie Chart.
  Views: Protocol Hosts Summary (From and To) for the single protocol; Protocol Conversations Summary (Total) for the single protocol.

To display Flow Forensics-level detail, click the **Flow Forensics** link and run a Flow Forensics report.

Top Protocols (Table)

The **Top Protocols (Table)** views are tables that show the rate, volume, or utilization for the highest-volume protocol traffic on a particular interface. You can use this information to compare the data volume or utilization for particular protocols, for example.

The view contains an interface identification string and table. The table has a row for each protocol with the protocol name (keyword and TCP/UDP port assignment) and the following rate, volume, or utilization information (by default):

- Rate:
  - (CA PC/NPC) Average rate of total (**Average Total**), inbound (**Average In**), and outbound data (**Average Out**) for each protocol.
  - (CA PC) Maximum rate of data that is outbound, inbound, or both outbound and inbound (**Maximum Out**, **Maximum In**, and **Maximum Total**) on the interface for each protocol

The rate is calculated by dividing the data volume by the elapsed transmission time.

- Volume: Number of bytes/megabytes of outbound data (**Out**), inbound data (**To**), and all data (**Total**) for each protocol.
- Utilization:
  - (CA PC/NPC) Average utilization by inbound data (**Average In**), outbound data (**Average Out**), and total data (**Average Total**) for each protocol.
  - (CA PC) Maximum percentage of interface capacity that the outbound (**Maximum Out**) or inbound protocol data utilizes (**Maximum In**)

  The utilization percentage is calculated by dividing the data rate by the data speed.

Performance Center views show the data from the time range that is defined for the page.

Opening the View

To see this view in the Performance Center Console, go to one of the following locations:

- (CA PC) Custom dashboard
- (NPC) Interface pages (with an interface selected): Custom tab

Available Actions

You can perform several actions in this view, including the following:

- Change the type of measurement (Rate, Volume, or Utilization) and the view name by editing the view settings.
- (CA PC) Change the interface.
- Re-sort the table data by clicking a column heading. Click again to toggle between descending and ascending order.
- Change the **Max Per Page** value to show more or fewer items on each table page.
- (CA PC) Change the columns that are shown in the table: Click near a column border, click **Columns**, then choose the columns to display.
- Click a name to jump to a pre-filtered **Interfaces** report in DX NetOps.

Find Protocol Tables in the NFA Console

You can use these ways to display tables of protocol volume data in the NFA console for a selected interface:

- *Top N Protocols* -- Report type: Protocols. Filter: Top N Protocols. Presentation menu options: Summary Table; Volume.
  View: Protocol Summary Table for the Top N Protocols, plus other overview views.
- *Protocols for a Single Host* -- Report type: Hosts. Filter: Single host. Report subtype: Protocols. Presentation menu options: Summary Table; Volume.
  View: Host Protocol Summary Table for the single host.
- *Protocols for a Single Conversation* -- Report type: Conversations. Filter: Single conversation. Report subtype: Protocols. Presentation menu options: Summary Table; Volume.
  View: Conversation Protocol Summary Table for the single conversation.

To display Flow Forensics-level detail, click the **Flow Forensics** link and run a Flow Forensics report.

Customizing Dashboards and Views

The icons at the top of each dashboard or view page give you access to time frame settings and other options for all of the views. The icons in the title bar of each view let you modify view settings, export views, and access Help.

You can select predefined or custom time frames for dashboards. The time frame setting affects all of the views on the page.

View Options in CA Performance Center

This article describes some of the view options in the CA Performance Center Console.

Many views offer a search feature and other settings that you can change to modify the view. In addition to filtering and time frame options, the following options are available for most views:

- Editing view settings, such as changing the view title or data direction.
- Seeing more data by selecting another "page" of a table view.
- Increasing or decreasing the number of items that are shown per "page".
- Collapsing the view so that the data is hidden.
- Changing the managed item context for the data shown in the view.

> **NOTE**
> Users with the 'Save Changes to Shared Views' role right can save view modifications to their own user account. The changes persist after logout. However, other users cannot see changes to views.

Other view options are specific to the selected view. The available options depend on the view format and data source.

Trend View Options

The trend views that are available in context pages let you quickly and easily change the trend lines that are displayed on the graph. The following options also apply to multitrend views:

- Right-click a metric in the chart legend and select **Hide** to remove it from the view.
- Exclude all other metrics by right-clicking a metric in the legend and selecting **Focus**.
- Narrow the focus to a precise time frame using the zoom feature.

Trend views also include an option to add a "goal line" as a visual indication of performance levels or thresholds. You can supply any value or label for the goal line, and you can show or hide the goal line for a selected trend view.

Table View Options

In table views, you can drill down to detailed data for individual items. Use the page feature to see metrics from a longer list of items. Increase the **Max Per Page** value to increase the size of the view and the number of table rows per page.

You can sort table data columns by selected metrics and also select columns to include. Click a table column to sort. A white arrow on the column lets you access a menu of table column options. Select **Columns** to enable and disable the metrics that were enabled for the table by default.

Browser View Options

The *browser view* is a unique view type that lets you add a URL to a selected report page. You can use this view to compare external factors alongside your network performance views. Also, the browser view lets you update internal and external data dynamically. The URL must be for a web page that supports embedded iframes.

Multiple external factors can affect the performance of your network and servers, such as world events and adverse weather conditions. The ability to view a weather map and news headlines alongside performance data views on a single report page can help you better understand patterns in network performance.

Device Admin Option

When a view does not display CA Infrastructure Manager Data Aggregator data, this option lets you drill down directly to the CA Infrastructure Manager Data Aggregator **Admin** page to troubleshoot monitored devices and items.

View Options in CA NetQoS Performance Center

This topic describes some of the view options in the CA NetQoS Performance Center Console.

Many views offer a search feature and other settings that you can change to modify the view. In addition to filtering and time frame options, the following options are available for most views:

- Editing view settings , such as changing the view title or data direction.
  
- Increasing or decreasing the number of items on a table "page" by using the **Max Per Page** option.
- Changing the context, so that the view data is restricted to a particular group.

> **NOTE**
> Users with the 'Persist Shared View Edits' role right can save view modifications to their own user account, but other users cannot see the changes. The changes persist after logout.

Many view options are specific to the selected view. The available options depend on the view format and data source.

Table View Options

In table views, you can drill down to detailed data for individual items. Use the page feature to see metrics from a longer list of items. Increase the **Max Per Page** value to increase the size of the view and the number of table rows per page.

You can click a table column to sort the table data by the values in the column.

Browser View Options

The *browser view* is a unique view type that lets you add a URL to a selected report page. You can use this view to compare external factors alongside your network performance views. Also, the browser view lets you update internal and external data dynamically. The URL must be for a web page that supports embedded iframes.

Multiple external factors can affect the performance of your network and servers, such as world events and adverse weather conditions. The ability to view a weather map and news headlines alongside performance data views on a single report page can help you better understand patterns in network performance.

View In Option

Some **Interface** page views allow you go directly to the data source to see detailed data. Click the arrow and select the **View in** option.

The option opens the view in the NFA console, for example.

Change the View Settings

You can change several settings for an enterprise-wide view or an interface-specific view. The available settings depend on the view.

For example, some views have settings for data direction--so you can display inbound data, outbound data, or all data. Other views have settings for the type of measurement--so you can set the view to display data rates, data volumes, or interface capacity utilization.

Follow these steps:

1. Open the dashboard that contains the view that you want to modify.

2. (Optional) Change the time frame, if necessary.
3. Open the dialog for editing the view settings:
   – (CA PC)
   
   Click the **Edit** icon in the view title bar and select **Edit** from the menu.
   – (NPC)
   
   Click the **View** icon next to the view name and select **Edit** from the menu.
4. (Optional) Edit the **Title** to change the name in the view title bar.
5. (Optional) If the view has **Measurement Settings**, select the type of measurement to show in the report:
   – **Rate**: Rate of traffic, expressed in Mbps.
   – **Bytes**: Volume of traffic.
   – **Utilization**: Percentage of total capacity used by the traffic.
6. (Optional) If the view has **Direction** settings, select the direction of the data on the selected interface:
   – **Out**: Outbound traffic on the interface.
   – **In**: Inbound traffic on the interface.
   – **From**: Traffic on the interface that comes from the host (**Host** view) or the source host (**Conversation** view).
   – **To**: Traffic on the interface that goes to the host (**Host** view) or the destination host (**Conversation** view).
   – **Total**: All traffic on the interface.
7. (Optional) (CA PC) For an enterprise-wide view or interface-specific view, change the interface for the view: Select a different interface from the **Context Settings** table.
   For example, to restrict the interface set to a domain, select the domain from the **IP Domains** list.
   Note: You can change the context for a view on some custom dashboards in the CA Performance Center Console. You cannot change the context for built-in views or for interface context pages.
   To re-sort the table, click a column heading. Click the arrow to sort in the opposite direction.
   (CA PC) To change the columns that are included in the table, display the **Columns** list: Click near the right edge of a column heading, then click the white arrow and select **Columns**.
8. (Optional) Specify which users are affected by the setting changes: Select a value from the **Apply Changes** list:
   – **For All Tenant Users**: (CA PC) Saves the changes so that they apply to all users.
   – **My User Account**: Saves the changes to your user account as the default setting for this view.
   – **My Current Session**: Reverts the changes when you log out.
   Note: The availability of these options depends on your user account role rights.
9. Click **Save** (CA PC) or **OK** (NPC) to save your changes.
   The settings dialog closes. The view reflects your changes.

> **NOTE**
> You can also change the context for a dashboard in the CA Performance Center Console. In this case the context change applies the selected group or managed item as a filter to all views on the page.

Change the Context for a View

You can change the context for some individual views on a dashboard. Change the context of an enterprise-wide view to show data from a different set of managed items. Change the context of an interface-specific view in the CA Performance Center Console to show data from a different interface.

Changing the view context can be useful for troubleshooting. You can compare data from different locations, for example.

Follow these steps:

1. Open the dashboard that contains the view that you want to change.

2. (Optional) Change the time frame, if necessary.
3. Open the dialog for editing the view settings:
   – (CA PC)
   
   Click the **Edit** icon in the view title bar and select **Edit** from the menu.
   – (NPC)
   
   Click the **View** icon next to the view name and select **Edit** from the menu.
4. Take one of the following steps:
   – (CA PC) Enterprise view: Click to expand folders in the **Groups** filter tree, and select the group whose data you want to see in the view.
   – (CA PC) Interface view: Locate the interface whose data you want to see in the view, and click the link in the table.
   – (NPC) Click the **Filter by** value and select a group in the **Select Context** dialog.
   If the settings dialog does not contain a **Filter by** option, you cannot change the context for the view.
   The context types that are available depend on the type of view.
5. (Optional) Change the view **Title** to reflect the new context.
6. Select the scope of your changes from the **Apply Changes** drop-down. Select one of the following options:
   – (CA PC) **For All Tenant Users**: Saves the changes so that they are only available to users associated with your tenant (possibly the Default Tenant).
   – **My User Account**: Saves the changes to your user account as a default for this view.
   – **My Current Session**: Reverts the changes when you log out.
   Note: The availability of these options depends on your user account role rights.
7. Click **Save** (CA PC) or **OK** (NPC).
   The view is updated to show data from the new context.

You can also change the context for a custom dashboard, which applies the selected group or managed item as a filter to all views on the page.

Set a Custom Time Frame

You can select a different time frame for the data shown in the current dashboard or view page. You can select the day, the start time, and the end time using the time period selectors.

Follow these steps:

1. Navigate to a dashboard or view page.
2. Click the date links in the upper-left corner of the dashboard page to open the calendar panes.
3. Select the beginning day and ending day of the new time period on the calendar panes.
4. Click the hours or minutes links to specify the beginning and ending times of the new time period.
5. Click **Set**.
   The custom time frame is applied to all of the views on the dashboard or page.

Zoom In to Narrow the Time Frame

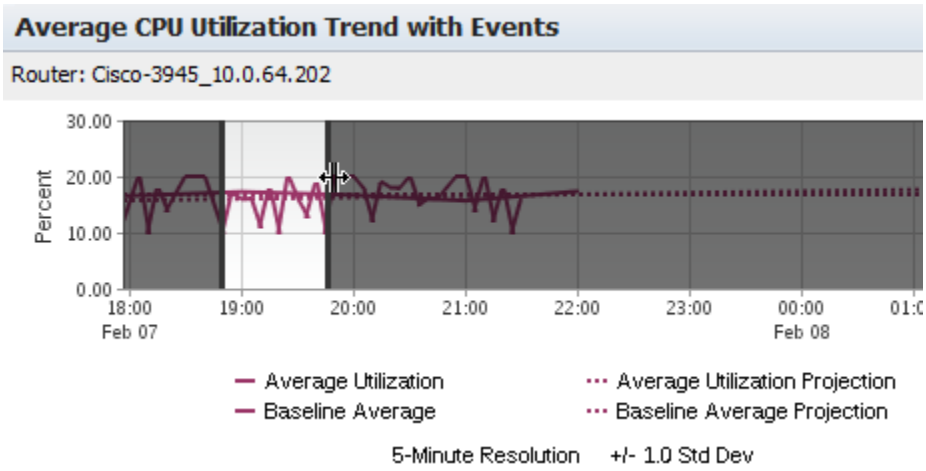
This topic describes how to zoom in to narrow the time frame for a trend chart in the CA Performance Center Console. This option is not applicable to trend charts in the CA NetQoS Performance Center Console.

You can look more closely at the data points from a small range by using the zoom feature. The ability to "zoom in" on a time frame is available for views that contain trend (line) charts. The feature is not available for bar charts, tables, or gauges.

Follow these steps:

1. Navigate to a dashboard page.
2. (Optional) Change the time frame, if necessary.
3. Select a view that contains a line chart.

   NOTE
   You cannot zoom in on a bar chart, table, or gauge.
4. Click and drag, using the mouse to select an area of the chart.



Select an area that spans at least 30 minutes. Black lines appear to indicate a valid selection.

When you release the mouse button, the custom time period you selected is applied to the current view.

- (Optional) Click **Undo**, just below the view, to return to the previous time frame.
  The view is refreshed. The previous time period is now applied to the view.
- (Optional) Click **Apply to Dashboard**.
  The dashboard page is refreshed. The new time period is now applied to all views on the current dashboard page.

Custom Dashboards

Custom dashboards are useful for displaying data from a particular item or group of items. With a custom dashboard, you can select the item context for individual views and can make other modifications to meet the requirements of a selected operator.

Custom dashboards are often used on a temporary basis to troubleshoot an issue. However, they are also deployed on a long-term basis to monitor categories of items. For example, an operator who is responsible for a region requires a dashboard that shows only items in that region. Or an operator might require a dashboard to monitor all ESX servers.

To create a custom dashboard quickly, you can edit an existing dashboard and save it with a new title. Your user account must have the Edit Dashboards role right.

If you have the necessary role rights, you can create custom dashboards. You can select views for the dashboard and set their location on the page. You also can select the menus that list the dashboard so that it can be shared with other operators.

NOTE
Role rights are slightly different in CA NetQoS Performance Center. For more information, see the *CA NetQoS Performance Center* documentation.

Edit a Dashboard

This topic describes how to customize dashboard pages in the CA Performance Center Console.

You can customize dashboard pages if your user account has the 'Administer Shared Dashboards' or the 'Create a Dashboard' role right. You can add or remove data views, rearrange views, or select a different context filter for a dashboard. You can then export the new dashboard as a report.

Follow these steps:

1. Log in as a user with the required administrative role rights.
2. Use the **Dashboards** tab to access the dashboard that you want to edit.
3. Click the **More** menu, and select **Edit Dashboard**.
   The **Edit Dashboard Layout** page opens.
4. Change the following menu and dashboard options, as needed:
   - **Menu for Dashboard**
     Is the menu where you want the dashboard to appear. The default is the menu that you used to open this dashboard page.
   - **Menu Item**
     Is the name of the dashboard as you want it to appear in the menu.
   - **Dashboard Title**
     Is the name that you want to appear at the top of the new dashboard.
5. Select a layout template for the dashboard from the **Layout** buttons.
6. Remove unwanted views from the dashboard page if desired. In the **Layout** pane, click:
   - **Clear** to remove all views from the page.
   - An **[X]** to remove an individual view from the page.
     Note: By default, the context is **Summary**. With the *Summary context* setting, the available views display summary data for the current group context of the dashboard. The Summary setting does not require you to select a specific group or item. Summary views dynamically update the context when you change the context of the page.
7. (Optional) Apply a group or context filter to the views. You can select a group, device, or interface by taking the following steps:
   a. Click **Select Context**.
   b. Select a **Context Type**, such as a type of managed item. Select **Group** to see the Groups tree.
      By default, the list is filtered to show only items and item types to which you have access. For example, if you are not monitoring any servers, the **Context Type** list does not include the Servers option. Select **Show All Context Types** to see all context options.
   c. Select a specific context item or a group context.
   d. Click **OK** to save the new context filter.
      The views that are available to be added to the page are shown in categorized lists. The lists are filtered by the selected group or item context.
      All registered data sources are represented.
8. Click to expand the categories of views. Check the **Display All Views** option only if you want to see views from data sources that you have not registered.
9. Select a view, drag it to the **Layout** pane, and drop it where you want it to appear.
   Note: The maximum number of views per dashboard is 25.
10. (Optional) Use the editing shortcut buttons to create a copy of the view or access view settings, such as the Metric Family.
    For example, click the **Copy** icon to place a copy of the view just below the original view in the layout.
    
    You can then alter view settings so that two similar views display different data.
11. (Optional) Click **Revert** to discard your changes.

The layout returns to the settings that you last saved. Or, if you have not customized the dashboard, it returns to the predefined settings.

12. Click **Save**.
    The dashboard page refreshes to reflect your changes. The changes persist across login sessions.

Change the Context for a Dashboard

You can customize a dashboard by selecting a different data context for the data. The default group setting for the views that are shown on all dashboards is **All Groups**. When you select another group for a standard dashboard, you apply a new filter to all views on the page. From a context page, such as details about a single router, you can select another managed item as the view context.

You can also view dashboards in multiple windows and apply a different data context to each dashboard.

Follow these steps:

1. Navigate to the dashboard that you want to modify.
2. (Optional) Change the time frame, if necessary.
3. Click the **[change]** link above the time period selectors.
   A dialog opens with filtering options.
4. Click to select another managed item. Or expand nodes in the **Groups** tree to select a group context.
   Data from the new item or group appears in the view.
5. Click **OK**.
   All views on the page are refreshed to reflect the new data context. The change applies until you log out. To change the context so that the change persists across login sessions, edit the dashboard.
6. (Optional) Open another browser instance, log in, and open the same dashboard.
   You can now compare the same views with two different item context settings.

Change the Time Frame for a Dashboard

You can change the time frame for a dashboard or Interface page that you are viewing. Change the time frame to see data from an earlier time of day or from another date, for example

Changing the time frame is useful for troubleshooting performance issues. For example, if data from the past day contains an anomaly, you can change the time frame to show data from the last seven days. The time frame helps you determine whether the issue occurs repeatedly.

When you change the time frame for a dashboard or Interface page, it is applied to all views on the page, and to all dashboards in that window. You can view dashboards in multiple windows and can apply a different time frame to each dashboard, however.

Follow these steps:

1. Select a dashboard from the Dashboards tab or navigate to an Interface page.
2. (Optional) Click to select any of the following time and date options at the top of the page:
   - **Time period drop-down list**
     Lets you select a predefined time frame for the data.
     Default: Last Hour.
   - **Back button**
     Shifts the time frame for the data back by one increment of the present interval (such as Last Day or Last Hour).
   - **Date and Calendar drop-down lists**

Let you select a start and end date for the data from a calendar view.
- **Time of Day drop-down lists**
  Let you select a start and end time from a list of 15-minute time intervals in the 24-hour format.
- **Forward button**
  Shifts the time frame for the data forward by one increment of the present interval (such as Last Day or Last Hour).

3. (Optional) Define a custom time frame:
   a. Complete one or more of the following steps
      - Click the start date and select a new start date from the calendar that opens.
      - Click the end date and select a new end date from the calendar that opens.
      - Click the start hour or minute and select a new hour or minute from the drop-down menu.
      - Click the end hour or minute and select a new hour or minute from the drop-down menu.
   b. Click **Set**.
      The data in the views reflects the new time frame.
4. (Optional) Scroll backward or forward in time. Use the Back and Forward buttons on either side of the timestamp to shift the time frame by one increment of the present interval.
   If you are viewing data for the last day, click the left arrow to scroll back in time by one day. Or click Latest to see the most recently collected data.

Build a Custom View for a Single Interface

You can create custom views that contain DX NetOps data for a single interface. You can add these interface views to a custom dashboard in the Performance Center Console.

Several interface views are available in addition to the views that are displayed by default on the interface pages.

Follow these steps:

1. Log in as a user with the required administrative role rights.
2. Navigate to the dashboard page that will contain the new custom view.
3. Click the **More** menu, and select **Edit Dashboard** (CA PC) or **Edit Report** (NPC).
   The page for editing the dashboard opens.
4. Open the dialog for selecting the context:
   - (CA PC) Click the **Select Context** link in the **Views** pane.
   - (NPC) Click the **Filter by** value in the left pane.

   The dialog for selecting the context opens.
5. Select **Interface** from the **Context Type** list (CA PC) or **Type** list (NPC).
   A table of interfaces opens, which shows the available interfaces or interfaces and routers (CA PC).
6. Locate and select an interface.
7. (CA PC) Click **OK**.
   You return to the **Edit Dashboard Layout** page.
8. Expand the view category:
   - (CA PC) **IP Flow**
   - (NPC) **Interface**
9. Drag one or more of the interface views into the layout.
   A dashboard can contain a maximum of 25 views. If you include multiple copies of the same view type, you can edit each view on the custom dashboard to show content from a different interface.
10. Click **Save**.
    The dashboard page refreshes.

Sharing Data with Other Users

Multiple options let you share dashboards and views with coworkers. You can export a dashboard to a static report in PDF format. You can print reports or send them by email. You can set up a schedule to send a report automatically on a regular basis.

You can also export individual views. You can publish views on a web page, such as an intranet site. Or you can export data from a view to a file in CSV format. For all data-export options, certain user account role rights are required.

Print a Report

If your user account has the required role right, you can export the current dashboard contents as a printed report. The **Print** feature first displays the current dashboard page in PDF format.

Follow these steps:

1. Navigate to the dashboard that you want to export as a report.
2. (Optional) Change the time frame.
3. Click the **Print** link on the toolbar.
   The report is exported as a PDF. Typically, it is displayed in a separate browser window.
   The data uses the current dashboard settings.
4. (Optional) Save the PDF to the local computer using the options in your PDF viewer.
5. Click the **Print** icon in the browser toolbar.
   The report page is sent to the local default printer.

Send a Report by Email

You can export the current dashboard or **Interface** page as a PDF report attached to an email message. The Email feature lets you specify the email address of the recipient and the Subject line of the email message.

Sending reports as email attachments requires an administrator to specify an SMTP server. In CA Performance Center, your user account must also have the required 'Send Reports by Email' role.

Follow these steps:

1. Open the dashboard or Interface page that you want to send in an email message.
2. (Optional) Change the time frame, if necessary.
3. Click the **Email** icon on the toolbar.
4. Supply information for the following fields:
   - **Send To**
     Specifies the email addresses where the report should be sent. Use the standard format:

     ```
     <name>@<domain>
     ```

     NOTE
     Use commas or semicolons to separate multiple addresses. Or you can enter an email alias that includes multiple recipients.
5. Select **Send Now** to send the email message immediately.
   Or select **Send on a Schedule** to create a schedule to send the email message on a regular basis.
   If the dashboard contains at least one multiview, a check box labeled **Display maximum results for multiple-chart views** appears. Select this option if you want to include more charts from the multiview than can be displayed on the current page of results.
6. Click **OK**.

The server generates a PDF from the current dashboard and sends the report as an attachment to an email message.

Set Up a Recurring Email Schedule

Each dashboard and Interface page contains options to export and send data in reports.

You can send a report by email immediately, or you can create a schedule for recurring emailed reports. For example, you can email interface utilization reports each week to coworkers in the IT department for capacity planning.

> **NOTE**
> The administrator must specify an email server to enable this feature. In CA Performance Center, your user account must have the 'Send Reports by Email' role.

Follow these steps:

1. Log in to the Performance Center Console.
2. Navigate to the dashboard or Interface report page that interests you.
3. Click **Email**.
   The **Email Dashboard** dialog opens.
4. Supply information in the following fields:
   - **Send To**
     Specifies the email addresses where the report should be sent. Use the standard format:

     ```
     <name>@<domain>
     ```

   - **Subject**
     Appears in the email Subject line; describes the emailed report.
     Example: The dashboard title and any components whose data is included in the report.
   - **Message**
     (Optional) Is a message to accompany the emailed report.
5. Select one of the following **Scheduling Options**:
   - **Send Now**
     Sends the email message immediately. Scheduling is not enabled.
   - **Send Daily**
     Sends the email message once or more per day. If you select this option, select the day or days of the week when the report is sent.
     Default: Send the emailed report every weekday (Monday - Friday) at 0:30 hours in the time zone of the logged-in user. The data in the report reflects the previous 24 hours.
   - **Send Weekly**
     Sends the email message once per week. If enabled, lets you select the day of the week to send the report.
     By default, the weekly schedule sends the emailed report every Sunday at 01:00 in your time zone.
     Default: The data in the report reflects the previous seven days (Saturday - Sunday).
     - **Week Ends on**
       Determines the day when the week ends. The start of the week is automatically adjusted to include seven days.
   - **Send Monthly**
     Sends the email message once per month. The report is sent on the first Sunday of each month at 01:00 in the time zone of the Management Console. The data in the report reflects the previous 30 days.
     - **Send Email at**
       Determines the time of day when the message is sent. The start of the month is automatically adjusted to include 30 days.
   - **Send Quarterly**

Sends the email message once per quarter. The report is sent on the first Sunday of each quarter at 01:00 in the time zone of the Performance Center Console. The data in the report reflects the previous three months.
    - **First Quarter Ends in**
      Determines the month when the quarter ends. The start of the quarter is automatically adjusted to include three months. All other quarters are also adjusted to proceed from the first quarter.
  - **Send Yearly**
    Sends the email message once per calendar year. Sends the report on the last day of the month you select for the 'Year ends in' parameter. The data in the report reflects the previous 12 months.
    - **Year Ends in**
      Determines the month when the year ends. The start of the year is automatically adjusted to include 365 days.
  - **Send email at [time of day]**
    Sends the email message at a time you select.
6. Click **Save** to save the schedule.
   The report is saved as a PDF file and attached to an email message. It is sent immediately or according to the schedule you selected.

Manage Email Schedules

Users with the required role rights can set up schedules to send reports by email on a recurring basis. Selected dashboard or Interface page data is exported in report format and sent to designated users according to a regular schedule.

Users who have the role right to schedule emails can also manage email schedules for other users.

Follow these steps:

1. Log in to the Performance Center Console as a user with the appropriate role right. In CA Performance Center, you need the 'Send Reports by Email' role right.
2. Open the list of scheduled emails:
   - (CA PC) Select **Admin**, **System Settings**, and click **Scheduled Emails**.
   - (NPC) Select **Admin**, **User Settings**, and click **Scheduled Emails**.

   The list of scheduled emails opens.

   NOTE
   Tenant administrators only see the items that are associated with their tenant.
3. Select the email schedule that you want to change, and click **Edit**.
   The dialog for editing the email schedule opens.
4. View or change the settings for email schedules.
5. Click **Save**. The list of scheduled emails reflects your changes.

Generate a URL for a View

You can export a view and share it with coworkers who do not have access to dashboards. Performance Center can generate a special uniform resource locator (URL) to recreate a selected data view on demand. The URL lets you add the view to a web page or intranet site. The Generate URL feature lets you involve others in capacity planning and infrastructure upgrade decisions, for example. This feature also lets you share status information.

A security token is included with each URL. This token is based on the user who is logged in at the time of URL generation. Consequently, any user who can access the exported view can see the same data that the original user who exported the URL could see. Note, however, that the token applies only to the initial view. If the user who is accessing the exported view attempts to drill in, he or she is asked to authenticate. The drilldown only succeeds after successful authentication, and then only for a user account with the Drill into Views role right. Finally, an option is included to let the token (and thus, the view) expire after a selected time period.

Follow these steps:

1. Log in as a user with the appropriate role right: 'Generate URLs from Views' (CA PC) or 'Allow User to Generate URLs' (NPC).
2. Navigate to the dashboard or Interface report page that contains the view that you want to generate as a URL.
3. Open the **Generate URL** dialog:
   - (CA PC)
     
     Click the **Edit** icon on the view, and select **Generate URL**.
   - (NPC)
     
     Click the **Edit** icon on the view, and select **Generate URL**.

   The **Generate URL** dialog opens. The URL is displayed in the **URL** field.
4. Enable or disable the following required parameters for the exported view:
   - **View Container**
     Displays the chart or graph with a surrounding container. The container includes the title of the view in a title bar and a black outline around the chart or graph.
     Default: Enabled
   - **Copyright**
     (CA PC) If enabled, shows the copyright information for the web page in the view.
   - **Drill Down**
     Enables users to drill down from the view into the underlying data source for more detailed data. These users must have a minimal product privilege to the data source and the 'Drill into Data Sources' role right to use this feature.
     Default: Enabled.
5. Select from the following time frame options:
   - **Time Options**
     (CA PC) Let you change the time frame for the data in the exported view. Supply a custom time frame in the Start Time and End Time fields, or select a Time Range from the drop-down list.
   - **Time Span**
     (NPC) Let you select a time frame for the data in the exported view.
   - **Token Expiration Options**
     Control view expiration. The default, 'Never' expires, lets the exported view display indefinitely.
     If you want the view to expire, select a timeout period from the **Token Expiration** list. The URL includes an encrypted token that causes the view to expire after the specified timeout period.
     The token does not enable the user who interacts with the generated view to drill down for more data.
6. (Optional) Click **Preview** (CA PC) or **View Preview** (NPC) to see how the view looks with the options you selected.
7. Copy the URL displayed at the top of the page to the Clipboard.
8. Paste the URL to the destination for displaying the view.
9. Click **OK**.
10. The **Generate URL** window closes.

Organizing Dashboards in Menus

Dashboards are organized into menus that have a central troubleshooting or monitoring purpose. You see a list of available dashboards and menus when you hover on the **Dashboards** tab (CA PC) or **Reports** tab (NPC).

Users with the required administrative role rights can reorganize menus. They also can create custom menus that contain built-in or custom dashboards. The users can associate the new menus with user account roles. When product operators log in, the dashboards they need for their daily tasks are organized in a meaningful way.

Administrators can remove a dashboard from any menu and add it to a shared menu.

View a List of Menus

This topic describes the menus in the CA Performance Center Console.

The **Manage Menus** page contains a list of currently defined menus. Before you add custom menus, only predefined menus are included in the list. The user account role determines the menus that each user can access.

Custom menus are defined for each tenant. Only the factory menus are shared among tenants. The global administrator sees a list of menus not explicitly associated with a tenant.

Follow these steps:

1. Log in as a user with the required administrative role rights.
2. Select **Admin**, **User Settings**, and click **Menus**.
   The **Manage Menus** page opens.
   The page displays the current list of menus. The following menus are provided with CA Performance Center and appear by default in the **Menu** List:
   - **Infrastructure Health**
     Contains summary and overview dashboards with at-a-glance views of system and device health and performance, events, and threshold compliance.
   - **Application Health**
     Contains overviews and detailed analysis of application performance. Also contains related dashboards, such as performance by protocol and server performance.
   - **Capacity Planning**
     Contains dashboards that are related to projections, thresholds, and recent changes to systems or devices.
   - **Management**
     Contains at-a-glance scorecards and overview dashboards, as well as high-level summary and comparison dashboards.
   - **Operations Displays**
     Contains high-level overview dashboards appropriate for display in the Operations Center and for use by Network Operators.

   To perform any action on this page, select a menu, and then click a button.
3. If any dashboards have been customized, the following additional menu appears:
   - **My Dashboards**
     Contains frequently used dashboards for an individual user account. Any dashboards that this user modified become available in this menu.

     NOTE
     Users with the required role right can edit the **My Dashboards** menu for a user account by proxying that user account.

Custom Menus

Administrators and designers can create custom menus for the **Dashboards** tab (CA PC) or **Reports** tab (NPC). Custom menus let you determine which dashboards are available to each user account. For example, an operator might log in and see three or four menus of dashboards. The menus can be configured sot that the operators see only the data that they require.

If your user account has the necessary role right, you also can create custom dashboards to populate a custom menu.

Custom dashboards that are in a user's **My Dashboards** (CA PC) or **My Reports** (NPC) menu are not visible to other users. Users can therefore copy a dashboard from a factory menu to their **My Dashboards** or **My Reports** menu and customize it.

A custom menu is not available to any users until the administrator edits a role to include it. The role must, in turn, be assigned to a user account.

Add a Menu

Custom menus let you organize dashboards and make them available to selected roles. Administrators and designers can create custom menus and can select dashboards for each menu.

A custom menu is not available to any users until the administrator edits a role to include it. The role must, in turn, be assigned to a user account.

Follow these steps:

1. Log in as a user with the required administrative role rights.
2. Select **Admin**, **Menus**.
   The current list of menus opens.
3. Click **New**.
   The **Add Menu** page opens.
4. Supply values in the following fields:
   - **Name**
     Is a name for the menu. This name appears when you click the **Dashboards** tab.
   - **Description**
     (Optional) Describes the menu to help other operators identify it.
5. Select a dashboard to include from the **Available Dashboards** list (CA PC) or **Available Reports** list (NPC).
6. Click the right arrow.
   The dashboard moves to the **Selected Dashboards** list (CA PC) or **Selected Reports** list (NPC).
   Use **Shift** + **Click** or **Ctrl** + **Click** to select multiple dashboards.
   (CA PC) Use the up and down arrows to change the order of the dashboards in the menu.
   Note: A maximum of 20 dashboards can be assigned to a single menu. An error message appears if you try to add more than 20 dashboards.
7. Click **Save** to save the menu and close the **Add Menu** page. Click **Save & Add Another** to save the menu and add another menu.

Edit a Menu

Administrators and designers can edit menus to meet the changing needs and new job responsibilities of operators. Administrators can edit custom or built-in menus by adding, removing, and reordering dashboards.

Follow these steps:

1. Log in as a user with the required administrative role rights.
2. Select **Admin**, **Menus**.
   The current list of menus opens.
3. Select the menu that you want to modify.
4. Click **Edit**.
5. Modify the menu settings as required.
   For example, to remove a dashboard from the menu:
   - Select the menu in the **Selected** list.
   - Use the arrow button to move it to the **Available** list.

     NOTE
     A maximum of 20 dashboards can be assigned to a single menu. An error message appears if you try to add more than 20 dashboards.

6. Click **Save**.

NFA QueryBuilder

This feature is available from NFA 10.0.3.

The ODataAPI is a flexible tool that lets users extract data from the DX NetOps database. The ODataAPI enables integration between DX NetOps data and external applications.

The ODataAPI is a public API that uses the QueryBuilder GUI. The QueryBuilder is a guided URL builder that lets you create custom Query URLs to extract and explore flow data. The URLs return customized data in the specified format. You can view the data in a browser or process the data in a custom web application. The ODataAPI uses the OData 4.0 industry standards.

Access the QueryBuilder

All the users who have credentials can use QueryBuilder and ODataAPI. To log in to the ODataAPI QueryBuilder, use your credentials.

To access the QueryBuilder, go to the following URL:

```
http://<nfa odata host>:<nfa odata port>/ODataQueryBuilder
```

The schema XML description provides detailed information about the items and relationships in your system. To review the schema XML description and metadata, go to the following URL:

```
http://<nfa odata host>:<nfa odata port>/odata/api/$metadata
```

More Information:

Use the ODataAPI QueryBuilder

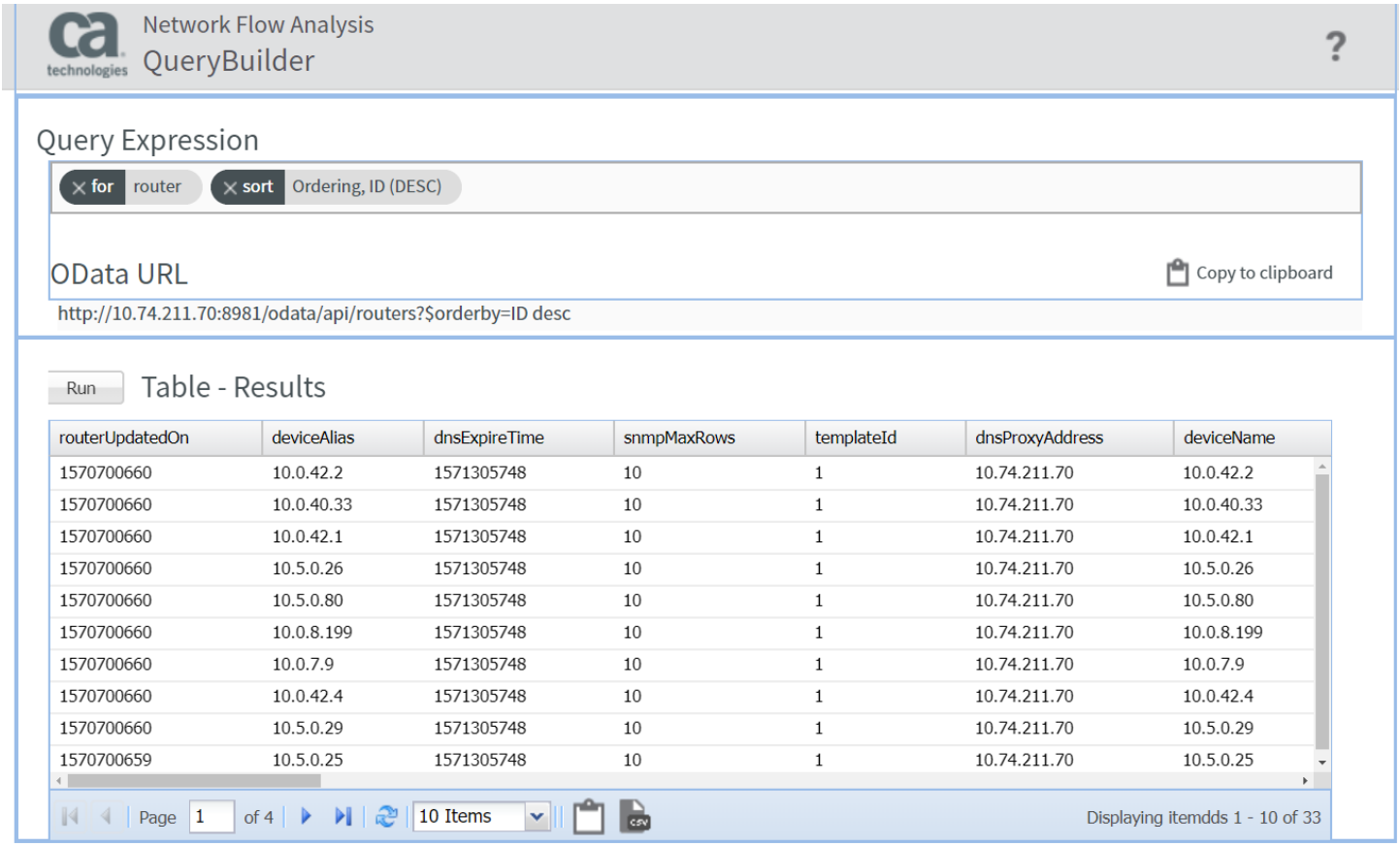
nfa1003

HID_ODataAPI_QueryBuilder

The ODataAPI QueryBuilder creates query URLs that export configuration and polled data.

NFA ODataAPI QueryBuilder User Interface

The following image shows the user interface of the NFA ODataAPI QueryBuilder.



Understanding the QueryBuilder Interface and Results

Use the **Query Expression** field to create your queries. The application creates the URL for your query and displays the URL in the **OData URL** section. The **Table - Results** section displays the results of the query. You can sort the results table in either ascending or descending order. You can select any column and customize the results table to show only the desired columns. You can also use the arrow buttons at the bottom of the **Table Results** section to view more results, if any. Click on the clipboard icon to copy the results to another application. Click on the export to CSV icon to export the results in CSV format. Click the ? (help) link, to see the NFA ODataAPI QueryBuilder documentation.

More Information

ODataAPI QueryBuilder Examples

Create an ODataAPI Query

To extract customized data sets from the data source, use the QueryBuilder to generate ODataAPI queries.

> **WARNING**
> Queries that return a large set of results can negatively affect your system. Refine your queries to return only the results that are relevant to your needs.
>
> **NOTE**
>
> The address bar of the Web browser updates to show the selected tokens in the Query Expression field. To continue editing the query later, copy and save this URL.

Follow these steps:

1. Click the Query Expression field to start a query.

2. Select the **for,** an option that represents what you want in the data set.
3. Add the **select**, **expand**, and **filter** tokens to define the output data.
4. Add more tokens to refine the results.
   The QueryBuilder creates the OData URL.
5. **Run the Odata Query -** To run the query in the QueryBuilder browser window, click **Run**.
6. **Copy OData URL to Clipboard** - After you build the query using the available expressions, select the **Odata URL** and click **Copy to clipboard**, to run the query from a Web browser, REST tool, or custom application.
   The report displays the result of the query in a table.

NFA QueryBuilder ODataAPI Controls

The ODataAPI QueryBuilder uses tokens that represent logical elements in the ODataAPI query syntax. Tokens appear when you click the Query Expression field. A token lets you select the type of available property for the selected entity. Selecting token updates the Query URL.

Use the following tokens to create and export Query URLs:

- **for**
  This token determines the type of entity that the query retrieves data about. This token appears only once in each query.
  Most options for this token include selections that set up an automatic filter. For example, when you select interface in the token, you see expand token option. When you select routers, expand token option is not available.
- **expand**
  The expand token indicates the related entities and stream values that must be represented inline. The CA NFA OData service must return the specified content and may return additional information.
- **navigation**
  The navigation token is similar to the CA Network Flow Analysis console user interface. The navigation property name is optionally followed by a /$count path segment, and optionally a parenthesized set of expand options like, filtering, sorting, selecting, paging, or expanding the related entities.

  NOTE
  The QueryBuilder currently supports navigation up to two levels.
- **select**
  This token determines the item properties to include in the data set. Property order is not supported in the results.
- **filter**

  This token adds custom filters that are based on logical functions using the AND and OR operations. Select whether the filter is case-sensitive for all logical functions. When OData evaluates a filter expression, the 'Any' operator is used to determine whether the Boolean expression is True or False for a collection of items. The following example returns True:

  ```
  odata/api/interfaces?$filter=((ID eq 10) and (contains(RouterAddress,'10.0.0.19'))
  ```
- **custom parameter**
  This token adds custom parameters in OData syntax to the query.
- **sort**

  This token controls the sorting of the query output. For example, the following query sorts the interface entity by ID:

  ```
  /interfaces?$orderby=ID
  ```
- **limit (top)**
  This token specifies the maximum number of rows or expanded rows in the query output.
  - **Maximum number of rows:** The number of rows in the Results table
  - **Number of rows to skip:** The number of leading rows to omit from the query output
- **format**

This token determines the format of the returned data set. The ODataAPI supports the HTML format.

HTML Table
If the query does not include this token, HTML Table is the default format.

NOTE
The HTML table format is supported only in the ODataAPI QueryBuilder. Direct ODataAPI queries support HTML and JSON.

Export the Query Results

You can download the query result in the CSV format.

For example: Using the following steps, you can download the report of the selected properties of the interfaces.

Follow these steps:

1. Select the following query options, in the Query Expression field.
   a. **for: interfaces**
   b. **select: AgentType,Description,Enabled,HarvesterAddress,ID,IfIndex**
2. Select **Run** to execute the query.
3. Select the  butt to export the query results in CSV format.

The report is saved in CSV format to your default download directory.

Configure ODataAPI Defaults and Limits

To perform operations that affect the system during off-hours, override the QueryBuilder properties. You can override several parameters in the QueryBuilder. The default parameter values that are defined in the OData limiters configuration file protect the system from ODataAPI queries that negatively affect the overall system performance. These parameters either limit the returned set of results or define the timeout threshold for potentially large operations. You can customize and override the default parameter values, which vary according to the scale and capability of the system.

For more information on the default limits and to override the default limiters, see theLimiterspage.

ODataAPI QueryBuilder Examples

nfa1003

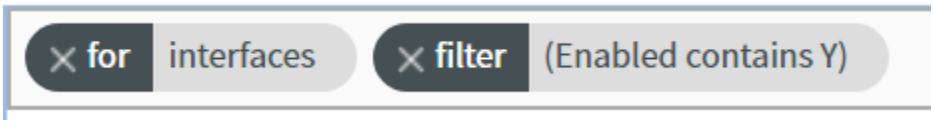
The following examples highlight the flexibility of the ODataAPI. Use these examples as a model to create your own ODataAPI queries:

Extract the Entity Information

The following examples show how to extract lists of entities with specific details.

Show all the Flow Enabled Interfaces

You want to get all the interfaces that are in the enabled state using this query.



To get this list, use the following tokens to build the query:

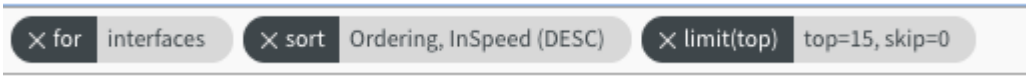
- **for: interfaces**
- **filter: Enabled contains Y**

ODataAPI URL:

```
http://<nfa odata host>:<nfa odata port>/odata/api/interfaces?$filter=contains(Enabled,'Y')
```

Show the Top Interfaces With High Inspeed

You want to get the list of interfaces sorted in the descending order of the Inspeed.



To get this list, use the following tokens to build the query:

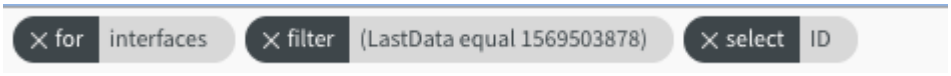
- **for: interfaces**
- **sort: Ordering, InSpeed (DESC)**
- **limit: Top=15, Skip=0**

ODataAPI URL:

```
http://<nfa odata host>:<nfa odata port>/odata/api/interfaces?$orderby=InSpeed desc&$top=15
```

Show 'Id' of Interfaces That Has Not Received Data After Specific Timestamp

You want a list of all interfaces that have not received data after a specific time.



To get this list, use the following tokens to build the query:

- **for: interfaces**
- **filter: LastData = 1547805618**
- **select: ID**

ODataAPI URL:

```
http://<nfa odata host>:<nfa odata port>/odata/api/interfaces?$filter = LastData eq 1547805618&$select =ID
```

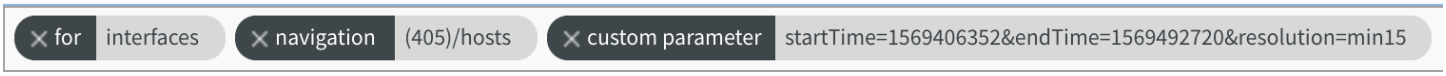
> **NOTE**
> Similarly, you can view the information for available interfaces, domains, templates and so on.

Extract the Interface Information Using Navigation

The following examples show how to extract interface information using the navigation tokens.

Show 15 Min Aggregated Information Of An Interface Within Specified Time

You want to know the fifteen minutes' aggregate information of the interface with ID 405 between 1569406352 and 1569492720.



To get this list, use the following tokens to build the query:

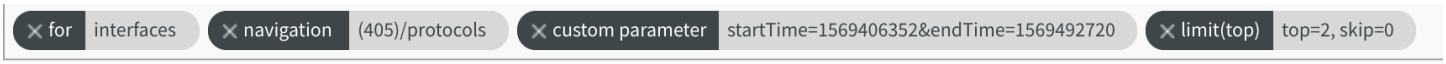
- **for: interface**
- **navigation: Enter interface Id = 405, Select Entity Name = hosts**
- **customparameter: startTime=1569406352&endTime=1569492720&resolution=min15**

ODataAPI URL:

```
http://<nfa odata host>:<nfa odata port>/odata/api/interfaces(405)/hosts?
&startTime=1569406352&endTime=1569492720&resolution=min15
```

Show The Top Two Protocols Information Of An Interface During The Specified Time

You want to see the top two protocol information of the interface with ID 405 between 1569406352 and 1569492720.



To get this list, use the following tokens to build the query:

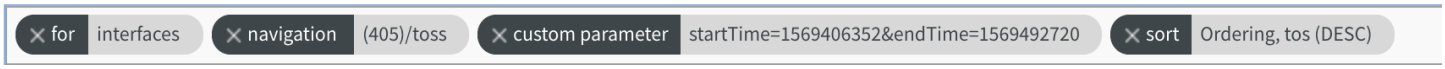
- **for: interface**
- **navigation: Enter interface Id = 405, Select Entity Name = hosts**
- **customparameter: startTime=1569406352&endTime=1569492720**
- **limit: Maximum Number of rows = 2, Number of rows to skip = 0**

ODataAPI URL:

```
http://<nfa odata host>:<nfa odata port>/odata/api/interfaces(405)/hosts?
&startTime=1569406352&endTime=156949272$top=2
```

Show The TOS Information Of An Interface During The Specified Time And Order By Most Recent To Older

You want to see the top two protocol information of the interface with ID 405 between 1569406352 and 1569492720.



To get this list, use the following tokens to build the query:

- **for: interface**
- **navigation: Enter interface Id = 405, Select Entity Name = hosts**
- **customparameter: startTime=1569406352&endTime=1569492720**
- **sort: Ordering, tos (DESC)**

ODataAPI URL:

```
http://<nfa odata host>:<nfa odata port>/odata/api/interfaces(405)/toss?
&startTime=1569406352&endTime=1569492720&$orderby=tos desc
```

Extract the Interface Information Using Expand

The following examples show how to extract the interface information using the expand token.

Show the Protocol Details of a Specific Interface

You want to know the protocol details of the interface-id 405.

To get this list, use the following tokens to build the query:

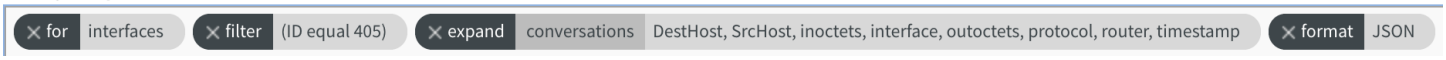
- **for: interface**
- **filter: ID equal 405**
- **expand: protocols, select=inoctets,outoctets,protocol**

ODataAPI URL:

```
http://<nfa odata host>:<nfa odata port>/odata/api/interfaces?
$expand=protocols($select=inoctets,outoctets,protocol)&$filter=((ID eq 405))
```

Show the Conversation Details of an Interface in JSON Format

You want to view the conversation details of the interface with id 405 in JSON format.



To get this data, use the following tokens to build the query:

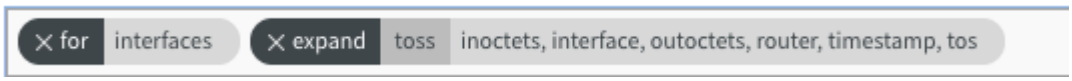
- **for: interface**
- **filter: ID equal 405**
- **expand: conversations($select=DestHost,SrcHost)**
- **format: JSON**

ODataAPI URL:

```
http://<nfa odata host>:<nfa odata port>/odata/api/interfaces?$expand=conversations&$filter=((ID eq 405))&
$format=json
```

Show the TOSS Information for all Interfaces

You want to view the conversation details of the interface with id 405 in JSON format.



To get this data, use the following tokens to build the query:

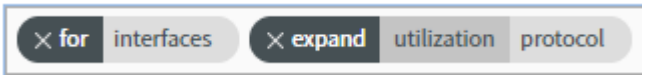
- **for: interface**
- **expand: toss($select=inoctets,interface)**

ODataAPI URL:

```
http://<nfa odata host>:<nfa odata port>/odata/api/interfaces?$expand=toss($select=inoctets,interface)
```

Get the Utilization Information for a Particular Entity

You want to view the utilization details of a protocol



To get this data, use the following tokens to build the query:

- **for: interface**
- **expand: utilization/ protocol**

ODataAPI URL:

```
http://<nfa odata host>:<nfa odata port>/odata/api/interfaces?$expand=utilization&$apply=groupby((utilization/
protocol),aggregate(Utilization))&resolution=MIN1&timeout=120
```

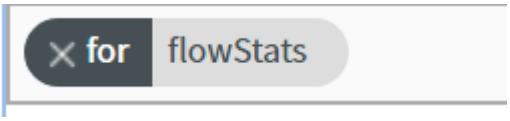
> **NOTE**
> The utilization report is shown only in JSON format.

Extract Settings and Health Information

The following examples show how to extract general application settings and health information.

View the Overall Flow Statistics

You want to view the overall flow statistics.



To get this data, use the following tokens to build the query:

- **for: flowStats**

ODataAPI URL:

```
http://<nfa odata host>:<nfa odata port>/odata/api/flowStats?
```

View the WatchDog Settings

You want to view the watchDog settings.

To get this data, use the following tokens to build the query:

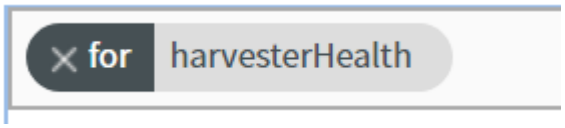
- **for: watchDogSettings**

ODataAPI URL:

```
http://<nfa odata host>:<nfa odata port>/odata/api/watchDogSettings
```

View the Health Of Harvester

You want to view the harvester health.



To get this data, use the following tokens to build the query:

- **for: harvesterHealth**

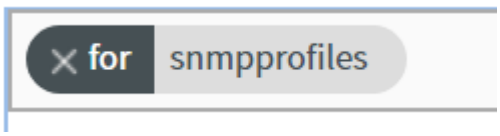
ODataAPI URL:

```
http://<nfa odata host>:<nfa odata port>/odata/api/harvesterHealth
```

> **NOTE**
> Similarly, you can view the health of the reporter.

View the Available SNMP Profiles

You want to view the list of available SNMP profiles.



To get this data, use the following tokens to build the query:

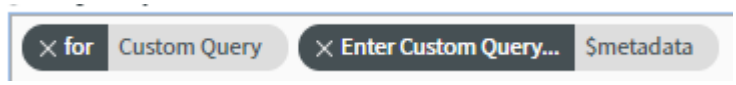
- **for: snmpprofiles**

ODataAPI URL:

```
http://<nfa odata host>:<nfa odata port>/odata/api/snmpprofiles?
```

Running a Custom Query

You want to run a custom query to view the metadata.



To get this data, use the following tokens to build the query:

- **for: Custom Query**
- **Enter a Custom Query: $metadata**

ODataAPI URL:

```
http://<nfa odata host>:<nfa odata port>/odata/api/$metadata
```

> **NOTE**
> You can run any custom query by providing only actual path information.

Integrating

DX NetOps (CA NFA) can be integrated with CA Unified Infrastructure Management (CA UIM) to display flow information.

The views displayed are:

- **Stacked Protocol Trend - In**
- **Stacked Protocol Trend - Out**
- **Top Hosts**
- **Top Conversations**

The **Advanced** tab is available when you select an individual interface in USM. The **Advanced** tab displays graphs for SNMPcollector probe metrics, as well as the following CA NFA views:

- **Stacked ToS Trend In**
- **Stacked ToS Trend Out**
- **Top Hosts per ToS**
- **Top Conversations per ToS**

Drilling down from within any of these views opens DX NetOps. Users can navigate back to the UIM console by clicking the portal link in the NFA console.

The CA Unified Infrastructure Management documentation can be accessed here, or you can use the field below to search the CA Unified Infrastructure Management documentation from this site:

Search this CA UIM keyword UIM847Search this CA UIM keyword

CA Digital Operational Integration

> **NOTE**
>
> : CA Digital Operations Intelligence 1.3 is an early access release and is available on request.

CA Network Flow Analysis (CA NFA) integrates with CA Digital Operational Intelligence. The OI Connector collects polling metrics from CA NFA and events, and inventory from CA PM and transmits them to the analytics platform. You install the connector to enable the integration with CA Digital Operational Intelligence.

Configuring the OI Connector also enables communication between CA NFA and the Operations Intelligence solution in one easy step.

By default, the OI Connector collects events every one minute with a 5-minute delay. Every minute, the OI Connector collects events that occurred in the 60-second interval that occurred 5 minutes before the collection started.

> **NOTE**
>
> **More Information:** Compatibility Matrix

Integrate CA NFA and CA Digital Operational Intelligence

To collect polling metrics and events from CA NFA, integrate CA NFA and CA DIgital Operational Intelligence. For more information about integrating CA Digital Operational Intelligence, see CA NFA - CA Digital Operational Intelligence Integration.

View CA NFA Metrics on CA Digital Operational Intelligence

You can view metrics information using Alarms Analytics and Performance Analytics. For more information, see Alarm Analytics and Performance Analytics.

View CA NFA Dashboard on CA Digital Operational Intelligence

You can create, view the CA NFA dashboard using Data Studio and Self Service Dashboards. For more information, see Unified Dashboards.

CA UIM Integration

DX NetOps (NFA) integration with CA Unified Infrastructure Management (CA UIM) requires the following:

- Integration is enabled by the nfa_inventory probe
  This is a local or remote probe used to send DX NetOps inventory to UDM for reconciliation with snmpcollector.
- The nq_services probe, which makes services internally available to DX NetOps and other NetQoS productsThis allows these products to interrogate CA UIM for data.
- DX NetOps inventory information is synced to UIM where it is correlated with inventory from other sources
- SNMP profiles are synced from CA UIM to DX NetOps
- In CA UIM, DX NetOps data displays along with SNMP data

NOTE

More Information: Compatibility Matrix

The steps required to accomplish the integration are:

1. Deployment
2. Configuration

Deployment

To deploy the necessary components to integrate DX NetOps with CA Unified Infrastructure Management:

1. Download CA UIM 8.4 or later.
2. Download CA NFA.
3. Install CA NFA.
4. Install CA UIM 8.4 (or later).
5. From the CA UIM Admin Console, download the nfa_inventory probe package into your archive.
6. From the CA UIM Admin Console, deploy the nfa_inventory probe to a robot either installed on the NFA console or remote to the NFA console.

   NOTE
   We recommend that you install a robot on the NFA console server, but it is not required.
7. Deploy the nq_services probe on the same robot as the trellis probe.
   After deploying the nq_services probe, restart trellis.
8. Deploy snmpcollector to a hub in the same domain as the NFA robot.

Configuration

1. Configure routers to send NetFlow to DX NetOps.
2. Using the CA UIM Admin Console, configure the nfa_inventory probe to point at the NFA console (by specifying the IP Address).
3. On a periodic basis, the nfa_inventory probe queries NFA for inventory information (routers, interfaces, data access and cross-launch URLs).

   NOTE
   If the nfa_inventory probe is on a non-wasp hub, the
   `/ump_common/nfa_inventory`
   key must be added to the
   `wasp config`
   using Raw Configure in the Admin Console.

   The value of the
   `nfa_inventory`
   key should be the bus address (/*domain*/*hub*/*robot*/nfa_inventory) for the nfa_inventory probe.
4. Configure snmpcollector so that all snmpcollector instances and DX NetOps have the same origin.

System Data Flow

1. The nfa_inventory probe calls DX NetOps and gets current routers and interfaces, creates a CTD and puts it on the bus.
2. The nfa_inventory probe calls DX NetOps to get the data and cross-launch URLs (which for UIM 8.4 and later are stored locally, and made available via the `get_info()` call).
3. The DiscoveryServer consumes the CTD while additionally doing correlation with any routers or interfaces found by other probes (for example, snmpcollector).
4. The DiscoveryServer inserts IPDevices and NetworkInterfaces into UDM.

User Data Flow

1. In USM, the user selects a router of interest, and then selects an interface.
2. USM calls the MetricsAPI, with the master interface ID, and timeframe.
3. The MetricsAPI calls UDM to get the local interface ID and the IP Address of the NFA console.
4. The MetricsAPI calls the data web service on DX NetOps, transforms the data into the format expected by USM, and returns that data to be displayed.

   NOTE

   More information:

   - nfa_inventory AC Configuration

Managing

These articles describe administration tasks that you perform in the NFA console.

Note: The term *Performance Center* refers to CA Performance Center and CA NetQoS Performance Center collectively. Program-specific pages names or functions may be identified by the full program name or acronym, which is *CA PC* for CA Performance Center and *CA NPC* for CA NetQoS Performance Center.

Administration Page Options

The DX NetOps **Administration** page lets you review, administer, and customize the network data view. When you open this page, the **System Status** is displayed. The **System Status** uses the following status icons:

- *Green* check mark: Component is running without any errors.
- *Red* exclamation mark: Component is running but has errors.

Click a red exclamation mark to view the associated error report. The error report includes the IP address, the type of component, and more information about the error. No error messages are displayed for a green check mark.

NOTE

- To perform administration tasks, log in as a user who has Administrator rights.
- A few options take you to a page in the Console for the Performance Center version that you use with DX NetOps. If the product is currently registered as a data source for CA Performance Center or CA NetQoS Performance Center, the following options are enabled:
  - CA PC Groups or NPC Groups
  - SNMP Profiles

A menu of administrative options is on the left side of the page.

Interfaces

Physical & Virtual

View the status of the interfaces on the **Active Interfaces** page. You can also edit, delete, or merge interfaces or create and edit custom virtual interfaces (CVIs).

Aggregations

Aggregate interfaces on the **Interface Aggregations** page. Interface aggregations let you view and report on interfaces as a unified group.

Enable Interfaces

View the router list and status of routers on the **Available Interfaces** page. Enable, disable, or delete interfaces on this page.

Templates

Use the **Interface Templates** page to view, add, edit, and delete the templates that determine the way interface names and descriptions are displayed.

Alerts

Traps

Review, add, edit, and delete traps on the **Trap Configuration** page.

Anomaly Detector

Display the Anomaly Detector window to perform administrative functions for CA Anomaly Detector. These functions have been added to the NFA console to accommodate users in a deployment that includes CA Performance Center.

The **Anomaly Detector** window gives you access to the following tabs:

- **View Monitored Products**: Add products for CA Anomaly Detector to monitor and review the list of monitored products.
  The other page functions are enabled as soon as you add an instance of DX NetOps to be monitored.
- **View Collection Sources**: View, enable, and disable the collection sources that the CA Anomaly Detector monitors.
- **View Sensors**: View and edit the default configuration settings for the sensors.
- **View Alert Targets**: Configure alert targets for snmp_traps and syslogging.

Define an Application

Application Definitions

Create and edit rules for application mapping, port priorities, or Reserved Seating on the **Application Definitions** page.

Protocol Names

Use the **Protocol Configuration** page to edit protocol names and descriptions. In a deployment that has multiple domains, protocol names are domain-specific.

Protocol Groups

Review, add, delete, or edit protocol groups on the **Protocol Group Configuration** page.

ToS Names

Use the **ToS Configuration** page to edit a ToS label (description) in the context of a particular domain. You also can add, remove, or edit groups of ToS values. In a deployment that has multiple domains, ToS names are domain-specific.

ToS Groups

Review, add, delete, or edit ToS groups on the **ToS Group Configuration** page.

AS Names

Use the **AS Names** page to search and edit Autonomous System names in the context of a particular domain. In a deployment that has multiple domains, AS names are domain-specific.

Reporting

Reporting Periods

Use the **Reporting Periods Configuration** page to add, edit, and delete reporting periods. Operators can use the reporting periods to limit the time frames for the data in Interface reports.

Time Filters

View, add, edit, and delete time filters on the **Time Filter Configuration** page.

Scheduled Emails

Use the **Scheduled Emails** page to review the reports that are scheduled for email delivery. You can change the destination address, Subject line, accompanying message text, and schedule options.

Addresses

Use the **Address-Hostname Configuration** page to specify a mask and options for resolving IP addresses to DNS names. You can list, edit, delete, and expire IP addresses. Expiring IP addresses schedules them to be refreshed. In a deployment that has multiple domains, address configuration is domain-specific.

Sites

Use the **Sites** page to add, edit, and delete site definitions.

Groups

Open the specific group-type page to review, add, edit, and remove groups.

- **Custom Report Groups**
- **Flow Forensics Report Groups**
- **Analysis Report Groups**
- **Site to Site Report Groups**
- **Site Groups**
- **Interface Groups**

Groups (CA PC or CA NPC)

Open the **Manage Groups** page in the Performance Center Console. Review system groups and add, remove, or edit custom groups and site groups. This option is enabled if DX NetOps is registered as a data source for Performance Center.

Authentication

Users

Open the page for managing user accounts. Review the user accounts and their settings. Add, remove, and edit user accounts.

This sends you to the Performance Center Console if the product is registered as a data source for CA PC or CA NPC.

Roles

Open the page to manage roles. Review the existing role names and the capabilities and menus that are available for each role. Add, delete, and edit roles.

This sends you to the Performance Center Console if the product is registered as a data source for CA PC or CA NPC.

Permissions

Open the **View Permissions** page to define sets of included groups. Then, if you have User product privileges, and that permission set, you are able to view those reports.

SNMP Profiles

Open the **SNMP Profiles** page to view, add, edit, delete, and re-order the SNMP profiles that Harvesters use for polling. In a multi-tenant environment, a Harvester uses the SNMP profiles that are assigned to its tenant. This sends you to the Performance Center Console if the product is registered as a data source for CA PC or CA NPC.

System

Harvester

Harvester is a software which parses data for processing.

Application Settings

Edit a wide range of application settings on the **Application Settings** page.

Health

Flow Statistics

The **Flow Statistics** determines the overall load for each harvester. The flow statistics allows you to make decisions about where new flows should be sent, whether rebalancing is needed. You can view the flow rates on harvesters for the last day, last week, or last 30 days.

Harvester

The harvester health shows the different metrics of the network that you monitor.

System Status

View the overall status of the DX NetOps components on the **System Status** page. Click a warning icon to display a list of the problem reports for the component.

Watchdog Settings

View and edit the Watchdog configuration settings on the **Watchdog Settings** page.

> **NOTE**
>
> **More information:**
>
> - View Flow Statistics
> - View System Status
> - How to Monitor the Components

About

The About page displays the following details about your DX NetOps installation:

- Version
- Date and Time of installation
- Link to view the version history
- Licensed Devices in Use
- License Utilization
- Link to product documentation

Manage Sites

Sites in DX NetOps are defined and managed from the **Administration** page, in the **Reporting, Sites** menu.

Sites can be defined as a collection of subnets that can also be discontiguous.

Follow these steps:

1. Select **Administration** from the NFA console menu.
   The **Administration** page opens.
2. Select **Reporting, Sites** from the menu on the left side of the page.
   The **Sites** page opens and displays a list of the current site definitions.
3. Select one of the following actions:

Create a Site Definition

1. Click **Save and Add New Site**.
   Fields and options for adding a site are displayed.
   The default **Site Name** is "`NewSite`*`nn`* ", where *`nn`* is the next number in the sequence. This can (and should) be changed.
2. Specify the following values:
   - **Site Name**: The default site name is shown. You can edit it as desired.

     NOTE
     Site names are unique within a tenant.
   - **Site Description**: (*Optional*) Additional notes to help identify the site. The description appears in the **Site to Site Report Wizard**.
3. Click **Add Network**.
4. Enter the **Network Name** and **Network Subnet**.
5. Repeat steps 3 and 4 to add more networks to this site definition.
6. Click **Save Changes to Site**.

Change a Site Definition

1. Select the site from the list.
2. Change any of the fields that are displayed.
   You can add networks to the site definition by clicking **Add Network**.
3. Click **Save Changes to Site**.

Delete a Site Definition

1. Select the site from the list.
2. Click **Delete Site**.

Managing Users, Groups, Roles, and Permissions

The DX NetOps (CA NFA) Administrator may set up accounts for operators to access various reports and functions, add those user accounts to the system, and assign them roles and product privileges.

You can create user accounts, groups, roles, and permissions in the NFA console (without using CA PC or CA NPC).

Manage User Accounts

DX NetOps (CA NFA) includes one automatically maintained user account, `admin` . This user account has full administrative and reporting privileges, which means that anyone who logs in using this account can create and run all available reports and also perform all tasks available from the **Administration** page, including creating user accounts and roles.

If the role for your user account is Administrator (like `admin`), you can create more user accounts, and new roles for those accounts.

The role for a user account must exist before you create the user account that requires it.

Creating a User Account

Follow these steps:

1. On the **Administration** page, select **Authentication, Users** from the menu on the left side of the page.
2. Click **New**.
3. In the **New User** page, enter the following information:

User Property	Description
Name	A name for the user.
Description	A description of the user.
Email Address	The email address of the user.
Authentication Type	The type of authentication to use for this user. Specify either **Product** or **LDAP** to determine where the user password is stored. To use Lightweight Directory Access Protocol (LDAP), you must install and use the *Single Sign-On Configuration Tool*.
Password	A password for this user.
Confirm Password	Enter the password again to confirm it.
Time Zone	Select the time zone. If you have users in different time zones, you can assign their local time zones to them. CA NFA displays reports in their time zone. These user settings are acquired from CA NPC or CA PC. CA UIM uses the CA NFA settings. The default time zone is the UTC time zone.
Product Privilege	Determines access to the **Administration** page. Select **None**, **Administrator**, **Power User**, or **User**. • **Administrator**: Allows access to all functionality on the **Administration** page • **Power User**: Allows access only to the reports on the **Administration** page • **User**: (Default) No access to the **Administration** page
Role	A role for the user. The default role is **Network Administrator**.
Permission Group	A permission set for the user. The default permission set is **All Groups**.
Account Status	The status of this user account. Select **Enabled** or **Disabled**.

Changing a User Account

Not all user information can be edited. You can change the email address, password, or time zone for the user.

Follow these steps:

1. Select the user account from the **Users** page.
2. Click **Edit**.
3. Change the information as necessary.
4. Click **Save**.

Deleting a User Account

Follow these steps:

1. Select the user account from the **Users** page.
2. If the user account can be deleted, the **Delete** button is active. Click **Delete**.
3. Confirm the deletion by clicking **Delete**.

Manage Groups

If you are not connected with a Performance Center, group types in DX NetOps are on the **Administration** page, in the **Reporting**, **Groups** menu:

- **Custom Report Groups**
- **Flow Forensics Report Groups**
- **Analysis Report Groups**
- **Site to Site Report Groups**
- **Site Groups**
- **Interface Groups**

The *Report Groups* allow you to control which groups of users have access to reports.

Site Groups allow you to create groups of sites, which can then be used to control who has access to those sites.

Custom *Interface Groups* can help users get the best results from custom reports and analyses. For example, interface groups can help operators set up custom reports that are based on geographic location, interface speed, T1 sites, or load balancing. You can create and manage interface groups in the NFA console, or in Performance Center as soon as you register the product as a data source.

You must have Administrator product privileges to add, edit, or delete a group.

Create a Group

1. Select the group type from the **Groups** menu.
2. In the **View** page for the group type, click **New**.
3. Enter a **Name** and **Description**.
4. Click **Edit** to select the items to include in the group.
   a. Click the items you want to add to the group, then click **Add**.
      Use **Shift + Click** or **Ctrl + Click** to select multiple items from the list.
      You can also use the **Filter <*group_type*> List** field to display only the items that match the pattern you enter. Click **Apply** after you type the pattern.
   b. Click **OK** when all the items you want have been added to the group.
5. Click **Save**.

Change a Group

1. Select the group type from the **Groups** menu.
2. Select the **Group Name** to edit.
3. Click **Edit**.
   If the group is not editable, the header displays the group title followed by **(Non-Editable)**. Click **OK** or **Cancel** to return to the group list.
4. Edit the **Name** or **Description** as needed.

5. Click **Edit** to change the items to include in the group.
   a. Select the items to add or remove.
      Use **Shift + Click** or **Ctrl + Click** to select multiple items from the list.
      You can also use the **Filter <*group_type*> List** field to display only the items that match the pattern you enter.
      Click **Apply** after you type the pattern.
   b. Click **OK**.
6. Click **Save**.

Delete a Group

1. Select the group type from the **Groups** menu.
2. Select the **Group Name** to delete.
3. Click **Delete**.
4. If the group is editable, you are asked to confirm the deletion. Click **Delete**.

Manage Roles

DX NetOps includes a Network Administrator role. You can create new roles in DX NetOps. For example, you might create a role that enables users to run existing reports without the ability to create and save new reports.

The role must exist before you create the user account that requires it.

Creating a Role

1. On the **Administration** page, select **Authentication, Roles** from the menu on the left side of the page.
2. Click **New**.
3. Enter the following information:

Name	Enter a name for the role.
Description	Enter a description of the role.
Enable Role	Leave this selected if you want this role to have permission to run reports.

4. Click **Edit** to select the access rights for this role. The **Select Access Rights** page displays.
5. In the **AVAILABLE RIGHTS** pane, select the rights for this role. Click the blue arrow to move the rights to the **SELECTED RIGHTS** pane.
   a. Use **Shift + Click** or **Ctrl + Click** to select multiple rights from the list.
   b. Click **OK** when all the rights you want have been added to the role.
6. Click **Save**. The new role is created with the selected rights.

Changing a Role

1. Select the role to change.
2. Click **Edit**.
3. Make changes as necessary.
4. Click **Save**.

Deleting a Role

1. Select the role to delete.
2. Click **Delete**.

3. Verify your selection.
4. Click **Yes**.

Manage Permissions

The default permissions in DX NetOps are:

Permission Name	Description
All Groups	Includes every group and item type defined within DX NetOps
None	This group contains no groups or items

Follow these steps:

1. Select **Authentication, Permissions** from the menu on the left side of the page.
   The list of currently configured permissions displays.
2. Select an action.

Creating a Permission

1. In the **View Permissions** page, click **New**.
2. Enter a **Name** and **Description** for the new permission.
3. Select a **Group Type** from the list of **Permitted Groups**.
   a. To change a **Group Type** definition, select the group and click **Edit**.
   b. Select an item or items from the **AVAILABLE** list and click the right arrow to move them to the **SELECTED** list.
      Use **Shift + Click** or **Ctrl + Click** to select multiple groups from the list.
   c. Click **OK**.
4. Click **Save**. The new permission is listed in the **View Permissions** page.

Changing a Permission

1. In the **View Permissions** page, select the permission to edit and click **Edit**.
   Note: If the permission is not editable, the permission information is displayed, but cannot be changed.
2. Edit the **Name** and **Description**, if necessary.
3. Select a **Group Type** from the list of **Permitted Groups**.
   a. To change a **Group Type** definition, select the group and click **Edit**.
   b. Select an item or items from the **AVAILABLE** list and click the right arrow to move them to the **SELECTED** list.
      Use **Shift + Click** or **Ctrl + Click** to select multiple groups from the list.
   c. Click **OK**.
4. Click **Save**.

Deleting a Permission

1. In the **View Permissions** page, select the permission to delete and click **Delete**.
   Note: If the permission is not editable, the permission information is displayed, but cannot be deleted.
2. Click **Delete**.
3. You are prompted to confirm the deletion. Click **Delete**.

Working with Interfaces and Routers

You use several different pages to perform the tasks for interfaces and routers:

Active Interfaces Page

The **Active Interfaces** page shows interfaces, routers, custom virtual interfaces (CVIs), and interface aggregations. The page shows only the routers and interfaces that have contributed to the collected data at some time. You can create and delete CVIs, merge interfaces, and edit properties. You also can delete routers and interfaces without deleting them from the system entirely, for example, to address capacity issues.

Select **Interfaces, Physical & Virtual** from the menu on the left side of the **Administration** page, then use the **Active Interfaces** page to perform the following main tasks:

- View details.
- Edit details.

  NOTE
  Interface names may not contain the characters < or >. Names with these characters are detected as a cross-site scripting attack from the UI.
- Delete interfaces or routers from the page.
- Create custom virtual interfaces (CVIs).
- Merge interfaces.

Available Interfaces Page

The **Available Interfaces** page shows information about all of the interfaces and routers, including the ones that have not contributed to the collected data at any time. You can enable or disable interfaces, delete routers and their interfaces from the system, and perform some polling troubleshooting.

Select **Interfaces, Enable Interfaces** from the menu on the left side of the **Administration** page, then use the **Available Interfaces** page to perform the following tasks:

- Check polling and view details about routers and interfaces.
- Enable or disable interfaces.
- Delete end-of-life routers from the system.

Interface Templates Page

You can create a custom interface template to change the way interface names and descriptions appear in several reports.

Select **Interfaces, Templates** from the menu on the left side of the **Administration** page, then use the **Interface Templates** page to perform the following tasks:

- Create and apply a template
- Edit a template

Application Settings Page

You can use a setting on the **Application Settings** page to control whether device names are included in interface names.

Select **System, Application Settings**, then use the **Application Settings** page to perform the following task:

- Show or remove the device name from interface names

  NOTE
  More information:

Active Interfaces Page

Use the **Active Interfaces** page to view the routers, interfaces, and custom virtual interfaces (CVIs). The page shows only the routers and interfaces that have contributed to the collected data at some time. You can perform the following actions on the **Active Interfaces** page:

- Review the routers, interfaces, and other elements without seeing the routers and interfaces that have never contributed to the collected data.
- Create and delete CVIs.
- Merge interfaces.
- Delete routers and interfaces to reclaim capacity without deleting them from the **Available Interfaces** page.
- Edit properties, such as:
  - Router name, domain, assigned SNMP profile/version, port, and interface naming template
  - Interface name, description, speed, type, and domain
  - CVI name, description, speed, type, domain, and subnets

Follow these steps:

1. Open the **Active Interfaces** page:
   a. Select **Administration** from the NFA console menu.
   The **Administration** page opens.
   b. Select **Interfaces, Physical & Virtual** from the menu on the left side of the page.
   The **Active Interfaces** page opens.

The **Active Interfaces** page includes the following options:

- **Search**
  Use a text string to search for routers, interfaces, or other elements with a matching address or name.
- **Edit**
  Modify properties of the selected item or items.
- **Delete**
  Remove routers and interfaces from the page; for example, to address capacity issues.
- **Merge**
  Combine the data for two selected physical interfaces or CVIs.
- **Add Custom Virtual Interface**
  Create a custom virtual interface based on the selected physical interface.

Active Interfaces: Router Information

Routers are listed in tables on the **Active Interfaces** page. The router tables contain the following columns.

- Flow
  Status 
  
  Indicator to show the status of the last flow at the router:
  - Red: Any enabled interfaces have not had flow for longer than the **Interface Data Absence Limit**.
  - Yellow: Any enabled interfaces have not had flow between 30 minutes and the **Interface Data Absence Limit**.
  - Green: All enabled interfaces have had flow in the last 30 minutes.
  - Gray: The router is a retired router.
- **Router Address**
  IP address of the router
- **Router Name**

User-assigned name of the router, such as *lab1 router*

- **Template**
  Interface naming template that is assigned
- **Interfaces**
  Total number of interfaces and CVIs for the router. This count includes only the interfaces that have contributed to the collected data at some time. To review all interfaces, go to the **Available Interfaces** page.
- **Harvester**
  IP address of the Harvester that collects data from the router

Active Interfaces: Interface Information

Interfaces and custom virtual interfaces (CVIs) are listed on the **Active Interfaces** page in tables nested under their parent routers. To display the interfaces and CVIs for a router, click the arrow next to the router name. Interface tables contain the following columns.

- Traffic
  Status 
  
  Indicator to show the status of the last flow at the interface:
  - Red: Inactive interface
  - Green: Active interface
  - Gray: Inactive Interface
- **Class**
  Icon for the interface type, such as a physical interface or CVI. To display the interface type name, position the cursor over the icon.
- **Interface Name**
  User-assigned name of the interface or CVI
- **Description**
  Optional information for identification
- **If Index**
  Index value, which is assigned by the device that sends flows to the interface
- **Type**
  Connection type, such as WAN or LAN
- **In Speed**
  Inbound speed of the interface, provided the speed is known
- **Out Speed**
  Outbound speed of the interface, provided the speed is known
- **Domain**
  Domain of the interface. If the environment has only one domain, the **Domain** column is not visible.
- **Notes**
  Link to add, edit, or view notes about an interface. If the note is empty, the Notes icon is dimmed. For example, you could add information about the time zone, business unit, geographical location, alternate bandwidth, circuit ID, or history. Notes icons are dimmed until they are populated and the page is refreshed.
  To display the Notes icon, set the **Display Notes Field** value to **True** on the **Application Settings** page.

Search for a Router or Interface

Use the **Search** function on the **Active Interfaces** page to locate routers, interfaces, or CVIs. To search for aggregations, use the **Search** function on the **Interface Aggregations** page.

Follow these steps:

1. Open the **Active Interfaces** page:
   a. Select **Administration** from the NFA console menu.
      The **Administration** page opens.
   b. Select **Interfaces, Physical & Virtual** from the menu on the left side of the page.
      The **Active Interfaces** page opens.
2. Enter a text string in the **Search** field. Search for whole or partial text strings that match the addresses, names, or descriptions of routers, interfaces, or CVIs.

   NOTE
   Do not use wildcards.
3. Click **Search**.
   The list is filtered to display only the matching entries. If you search for interfaces or CVIs, the search returns a list of routers that contain matching items. When you expand router details, all items appear in the sublists.

To clear the search, click **Clear Filter**.

Edit Router and Interface Details

Edit the properties of a router or interface. For example, you can edit properties to make corrections or to supply missing information. If you delete an interface without disabling it, the interface is automatically added again when the interface begins to send flow. You cannot Edit a Retired router and Interfaces. When you edit retired and active interface or router together, you see the following error message "Cannot save properties for RETIRED routers interfaces". However the changes are saved on the Active routers and interfaces.

You can also edit properties for CVIs and aggregations. If you select multiple elements to edit, editing options are restricted to their shared properties. The tenant-domain setting for interfaces need not match the parent router settings. Changing the domain can affect access permissions of the operators and reports to the related data.

Changing the tenant of a router in CA Performance Center can affect which SNMP profiles are available for polling the router interfaces. Changing the tenant of a router in CA NetQoS Performance Center does not affect the polling of SNMP profiles, as CA NetQoS Performance Center uses the same list of SNMP profiles for all routers.

Follow these steps:

1. Open the **Active Interfaces** page:
   a. Select **Administration** from the NFA console menu.
      The **Administration** page opens.
   b. Select **Interfaces, Physical & Virtual** from the menu on the left side of the page.
      The **Active Interfaces** page opens.
2. Locate the routers, interfaces, or other elements that interest you.
3. To display the contents under a router, select the arrow next to the router name.
4. Select the check box next to the items you want to edit.
   You can edit multiple interfaces and CVIs simultaneously or can edit multiple routers simultaneously. You cannot edit a mix of routers, interfaces, and CVIs simultaneously.
5. Select **Edit**.
   The **Edit Router** or **Edit Interface** dialog opens for editing the selected item or items.
6. Modify the properties as needed, such as:
   – Router name, tenant-domain (if applicable), SNMP profile/version, port, and template
     (CA PC deployment) The options to edit the SNMP profile and port are enabled once a tenant-domain is selected.
   – Interface name, description, speed, connection type, and domain (if applicable)
7. Select **Save**.
   The dialog closes. The changes are shown on the **Active Interfaces** page.

Delete Routers from the Active Interfaces Page

You can delete a router on the **Active Interfaces** page, but leave it in place on the **Available Interfaces** page. For example, you may want to delete a router in this way to address capacity issues. Later, you can restore the router (but not its historical data) by enabling its interfaces on the **Available Interfaces** page.

Deleting a router removes its interfaces, CVIs, 15-minute (historical) data, and traps. The deletion also affects any related aggregations, views, and reports.

NOTE
If you delete the router from the **Available Interfaces** page, the system has no further record of the router.

Follow these steps:

1. Verify that the router is no longer sending flows to the product.
2. Disable the router interfaces, if they are not disabled already:
   a. Open the **Available Interfaces** page:
      - Select **Administration** from the NFA console menu.
        The **Administration** page opens.
      - Select **Interfaces, Enable Interfaces** from the menu on the left side of the page.
        The **Available Interfaces** page opens.
   b. Locate the router by using the **Search** function or by paging through the table contents.
   c. Select the check box next to the router.
      You can select and disable multiple routers simultaneously provided the selections are on the same page.
   d. Click **Disable**.
      A confirmation message opens.
   e. Click **Yes**.
      The **Enabled** status for the interfaces changes to **No**. New data from the interfaces is no longer collected or shown in reports. Data that is already collected is still available for reports.
   f. (*Optional*) Update the **Enabled** value for the router by refreshing the page; for example, by pressing F5.
3. Open the **Active Interfaces** page:
   a. Select **Administration** from the NFA console menu.
      The **Administration** page opens.
   b. Select **Interfaces, Physical & Virtual** from the menu on the left side of the page.
      The **Active Interfaces** page opens.
4. Locate the router and select its check box.
5. Click **Delete**.
   A confirmation message opens.
6. Click **Yes**.
   The following events result:
   – The confirmation message closes.
   – The router is deleted from the **Active Interfaces** page and stops consuming capacity.
   – All related interfaces, CVIs, 15-minute (historical) data, and traps are deleted.
   – The router is deleted from any related aggregations.
   – The data for the deleted interfaces no longer appears in the NFA console or in reports.

NOTE
A router that you delete from the **Active Interfaces** page reappears if the router interfaces are enabled on the **Available Interfaces** page and the interfaces begin to send flow again.

Delete Interfaces

If you delete an interface on the **Active Interfaces** page, its historical data, related CVIs, and traps are deleted. The deletion also affects any aggregations, views, and reports that previously included the interface.

Follow these steps:

1. Disable the interface, if it is not disabled already:
   a. Open the **Available Interfaces** page:
      - Select **Administration** from the NFA console menu.
        The **Administration** page opens.
      - Select **Interfaces, Enable Interfaces** from the menu on the left side of the page.
        The **Available Interfaces** page opens.
   b. Locate the interface, using the **Search** function or by expanding the contents of the routers.
   c. Select the check box next to the interface.
      You can select and disable multiple interfaces simultaneously, including interfaces with different parent routers. All of your selections must be on the same display page.
   d. Click **Disable**.
      The **Enabled** status is changed to **No** for the interface. New data from the interface is no longer collected or shown in reports, but data that is already collected is still available for reports.
2. Open the **Active Interfaces** page:
   a. Select **Administration** from the NFA console menu.
      The **Administration** page opens.
   b. Select **Interfaces, Physical & Virtual** from the menu on the left side of the page.
      The **Active Interfaces** page opens.
3. Locate the interface and select its check box.
4. Click **Delete**.
   A confirmation message warns you about the results of the deletion.
5. Click **Yes**.
   The following events result:
   – The confirmation message closes.
   – The interface is deleted from the **Active Interfaces** page, but not from the **Available Interfaces** page.
   – All related historical data is purged permanently.
   – Data for the deleted interface no longer appears in NFA console views and reports.
   – All related CVIs and traps are deleted.
   – The interface is deleted from any related aggregations.
   – The interface stops consuming capacity on the server that stored its data.

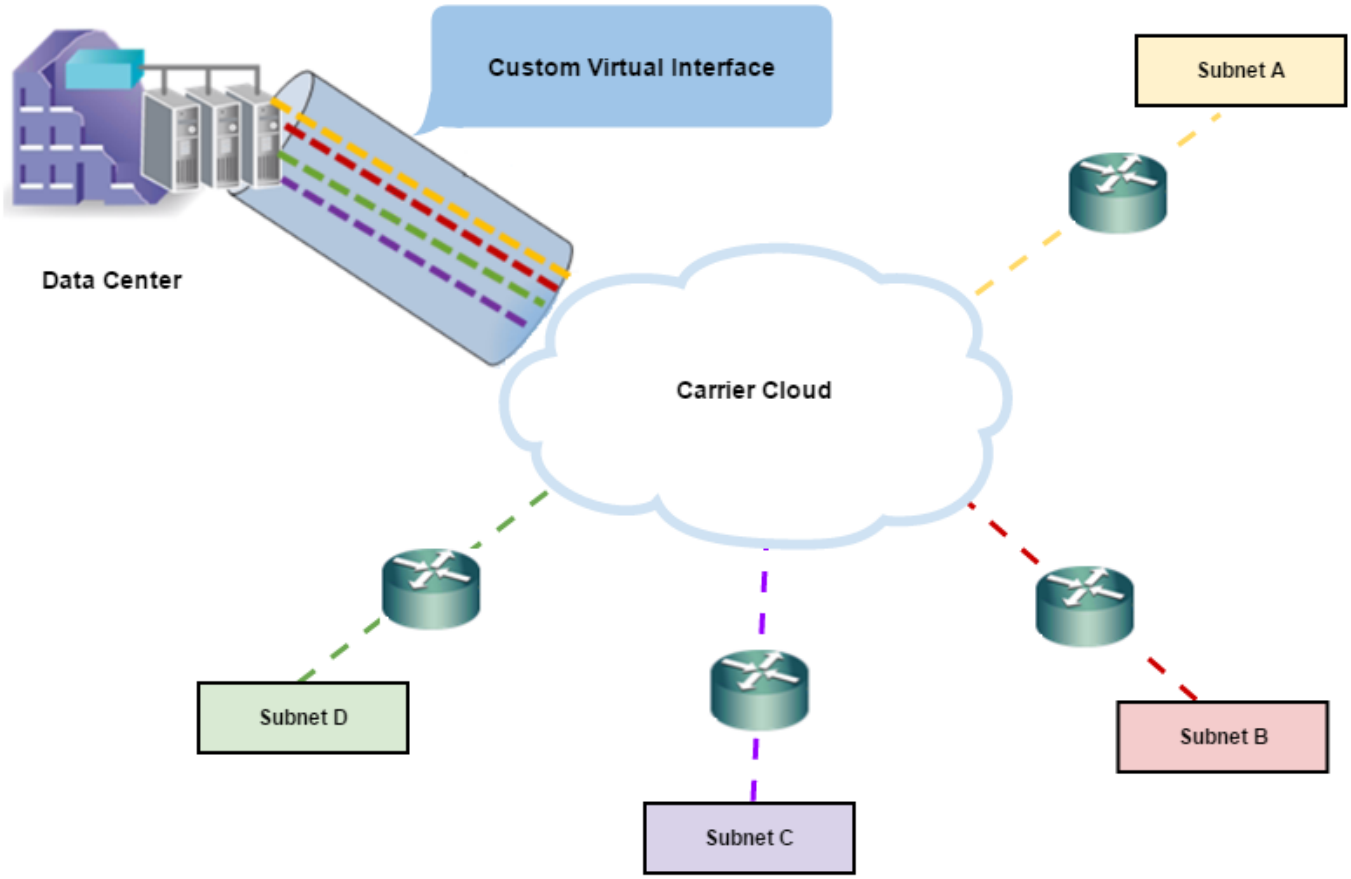
Custom Virtual Interfaces

You can create custom virtual interfaces (CVIs) to report on subnet traffic. Create CVIs for a network that is designed in the following way:

- Traffic from a data center is transferred to subnets through a multiprotocol label switching (MPLS) carrier cloud, and
- Flow is enabled on the routers in the data center rather than on the routers on the edge of the cloud.

Without virtual interfaces, you have limited visibility into which subnets actually generate the network traffic for the carrier cloud. Virtual interfaces help ensure that you can collect detailed data about the subnet traffic in this type of configuration.

Figure 99: Custom Virtual Interface



Defining a custom virtual interface (CVI) separates traffic for an interface from other traffic. The CVI in the diagram separates specific traffic from other traffic traveling to and from the data center into the cloud.

How to Configure Custom Virtual Interfaces

Create custom virtual interfaces (CVIs) to separate traffic on a particular interface and subnet from other traffic on the interface.

Create a Custom Virtual Interface

Follow these steps:

1. Open the **Active Interfaces** page:
   a. Select **Administration** from the NFA console menu.
   The **Administration** page opens.
   b. Select **Interfaces, Physical & Virtual** from the menu on the left side of the page.

The **Active Interfaces** page opens.

2. Expand the interface list for the parent router by clicking the arrow next to the router name.
3. Select the check box next to the name of a single interface.
4. Click **Add Custom Virtual Interface**.
   The **Add Custom Virtual Interface** dialog opens.
5. Enter values for the following fields:
   - **Interface Name**: Replace the default value in the **Interface Name** field with a meaningful name for the interface list. **Default:** Parent interface name, which you must replace.
   - **Description**: (*Optional*) Enter a text string to help identify the interface.
   - **In Speed** and **Out Speed**: (*Optional*) Identify the speed of data that is inbound to the parent interface and outbound from the parent interface.
   - **Type**: (*Optional*) Select the interface type from the list.
   - **Domain**: Select a tenant-domain combination from the list or accept the default setting.
     Changing the domain can affect which operators and reports have access to the data. The **Domain** option is visible only in an environment that contains multiple domains.
   - **Subnet**: Enter a subnet and mask identification for each subnet filter you want to use for this CVI, then click **Add**. Use the format:
     `<subnet IP address/subnet mask>`
     The CVI must contain at least one subnet filter.
6. Click **Save**.
   The **Custom Virtual Interface** is automatically deployed within one minute. The Class icon for the new CVI distinguishes it from the physical interfaces.

WARNING
If you delete a parent CVI, all the children CVIs are deleted automatically.

Custom Virtual Interface Priorities

CVI priorities are set according to how narrowly you specify the subnet mask. Subnet masks that are most specific have the highest priority. A CVI that has a single-node subnet mask (such as 192.168.20.2/32) has higher priority than a CVI that has a multiple-node subnet mask (such as 192.0.0.0/8). Establishing priorities in this way helps ensure that CVI traffic remains separate from other traffic, especially for a host on a large subnet.

NOTE
Reports for traffic between two CVIs with the same priority can be inconsistent.

Examples of Custom Virtual Interface Definitions

If you define the following CVIs:

- CVI-A: 192.168.0.0/16
- CVI-B: 192.168.100.0/24
- CVI-C: 192.168.100.123/32

The following reporting is enabled:

- CVI-A reports traffic that involves subnet 192.168.*x.x* except traffic that involves subnet 192.168.100.*x.*
- CVI-B reports traffic that involves subnet 192.168.100.*x* except traffic that involves host 192.168.100.123.
- CVI-C reports traffic that involves host 192.168.100.123.

If you add 192.168.100.124/32 as CVI-D, either CVI-C or CVI-D reports the traffic between 192.168.100.123 and 192.168.100.124. Only one of the CVIs reports the traffic.

If you add 192.168.200.0/24 as CVI-E, either CVI-B or CVI-E reports the traffic between 192.168.100.*x* and 192.168.200.*x*. Only one of the CVIs reports the traffic.

Merge Interfaces

You can merge interfaces so they are shown as one interface in reports. You may want to do this whenever major logical changes to a device cause a new interface to be created in the system.

Consider the following guidelines when you merge interfaces:

- You can merge two interfaces, which can be on different routers.
- The data collection period for the interfaces can contain gaps.
- If you merge interfaces that have overlapping time frames, the overlapping data is discarded. Precedence is given to the newer interface (the interface that has the later start date).
  For example, suppose you merge Interface A and B. Interface A has collected data that starts at 1:00 P.M. and ends at 5:00 P.M. on the same day. The data collected for Interface B starts at 3:00 P.M. The merged data consists of Interface A data from 1:00 to 3:00 P.M. and Interface B data from 3:00 P.M. onward.
- You cannot merge interfaces that started collecting data at the same time.

Example

For example, suppose that your 512-Kbps link that has been running for a year is upgraded to a T1 link (a 1.54-Mbps link). Using the new T1 link causes the interface `ifindex` to change. For example, the previous `ifIndex` of 5 changes to 13--the next available `ifindex` for the T1 link. Other settings are changed or created, such as the `ifdescr` and `ifAlias` settings.

These changes cause the program to see the interface as a new interface. You enable the new interface so the program can collect its data.

At this point, the history of the interface is divided. To unify the history, you merge the two versions of the interface. After the merge, the history includes the data that was collected previously from the interface on the slower link and the data from the interface on the new link. The data is combined end-to-end with no overlaps or duplication.

Steps for Merging Interfaces

Follow these steps:

1. Open the **Active Interfaces** page:
   a. Select **Administration** from the NFA console menu.
      The **Administration** page opens.
   b. Select **Interfaces, Physical & Virtual** from the menu on the left side of the page.
      The **Active Interfaces** page opens.
2. Expand the interface list for a router by clicking the arrow next to the router.
   The view expands to show a list of the interfaces for the selected router.
3. Select the check box next to two interfaces that you want to merge.
   The **Merge** button at the top of the page is activated and no longer appears dimmed.
   Note: The **Merge** button is enabled only when you have selected two interfaces.
4. Click **Merge**.
   The **Merge Interface Confirmation** dialog opens.
5. Verify that the information in the confirmation dialog is correct:
   – Verify that the interface shown as the source interface (as defined in the fields under the **Copy data from** label) is the interface from which you want to copy data.
   – Verify that the interface shown as the destination interface (as defined in the fields under the **Copy data to** label) is the interface to which you want to copy data.
   – Select the **Delete source interface after merging data** check box to delete the source interface automatically after its data is copied.
6. Click **Save**.
   The selected interfaces are merged according to the values in the **Merge Interface Confirmation** dialog box.

Customize the Active Interfaces Page

Customize the appearance of the **Active Interfaces** page as needed to make it easier to find information. You can sort tables, display details for routers, and change the maximum number of interfaces and CVIs that appear in detail pages under each router.

Sort Table Data

To sort the data, click a menu field. For example, to view all the routers with a red status indicator click **Flow Status**. The results are sorted by the Flow Status timestamp.

Expand the Details for Routers

To expand the details for one of the routers, click the arrow on the left side of the router row. A list of interfaces is shown below the router line.

> **NOTE**
> The number of details that are visible in the expanded view is limited to the **Max per Page** value for the selected router.

To display or hide details for all the routers, interfaces, or CVIs, click **Expand All** or **Collapse All** in the upper-right corner.

Change the Number of Visible Details

To change the number of items that are displayed, select a different maximum value from the **Max per Page** drop-down list. You can change the **Max per Page** value for the router list. You also can set a **Max per Page** value for the sublist of interfaces and CVIs under each router.

Available Interfaces Page

The **Available Interfaces** page displays information about routers and interfaces. The list of routers and interfaces include the interfaces that have never been enabled and have never been the source of collected data. You can enable or disable interfaces, delete end-of-life routers and their interfaces, and perform some polling troubleshooting.

Router Tasks

Review the displayed data, including the total number of interfaces, number of enabled interfaces, and the SNMP profile that was used.

- Perform a test poll with the current SNMP profile (Test).
- Look for an SNMP profile (Discover).
- Refresh the polling and interface information for the poller (Refresh).
- Enable or disable all of the interfaces for a router.
- Delete an end-of-life router and its interfaces from the system.

Interface Tasks

Review the displayed data, including the most recent time that flow was received and the enabled or disabled status.

- Enable or disable interfaces individually.

Follow these steps:

1. Open the **Available Interfaces** page:
   a. Select **Administration** from the NFA console menu.
      The **Administration** page opens.

b. Select **Interfaces, Enable Interfaces** from the menu on the left side of the page.
The **Available Interfaces** page opens.

2. Use the information and options in the following ways, for example:
   – Enable interfaces that you want to begin using.
   – Disable interfaces that you are not using.
   – Delete routers that you will not use again.
   – Review the router and interface information.

Available Interfaces: Router Information

The **Available Interfaces** page includes the following options and information for routers.

- Test

  Attempts to poll the interfaces for the router by using the settings that are displayed. You can use this option in troubleshooting to test SNMP connectivity. If the SNMP Profile value is missing, the test fails.
  – Success: If the test succeeds, click **Refresh**.
  – Failure: If the test fails, you can click the **Discover** option to try to find an SNMP profile that works. Alternatively, you can assign a different SNMP profile on the **Active Interfaces** page and click **Test** again.
- Discover

  Looks for an SNMP profile to use for polling the router's interfaces. You can use the **Discover** option if the SNMP Profile value is missing and you do not know which profile to use.
  – Success: If discovery succeeds click **Refresh**, then click **Test**. If the profile is different from the current one, the **SNMP Profile** value is updated.
  – Failure: Any SNMP Profile value that was displayed before discovery is removed. Discovery can fail for several reasons. For example, a valid SNMP profile may not be available, access to the profile may be blocked, or the router may be offline.
- Refresh

  Sends updated polling and interface information to the poller. Click the **Refresh** option after a successful **Discover** or **Test** operation.
- **Enable and Disable**
  Control whether the interface (or router and its interfaces) is allowed to send flow to DX NetOps.
- **Delete**
  Removes the router and its interfaces from the system.

  NOTE

  The Test, Discover and Refresh buttons are disabled for retired routers.

Standard Page Options

- **Search**: Search for routers or interfaces by address or name.
- **Max per Page**: Change the number of items that are displayed.
- Expand or collapse router contents.
- Sort contents by column.

Router Columns

- **Flow Status**

  Indicator to show the status of the last flow:

- – Red: Any enabled interfaces have not had flow for longer than the **Interface Data Absence Limit**.
  – Yellow: Any enabled interfaces have not had flow between 30 minutes and the **Interface Data Absence Limit**.
  – Green: All enabled interfaces have had flow in the last 30 minutes.
  – Gray: The router is a retired router.
- **Router Address**IP address of the router.
- **Router Name**Name of the router.
- **SNMP Profile**Name of the SNMP profile that was used in the most recent successful polling attempt. If this profile is not successful in the next polling attempt, the poller uses the next available profile until polling succeeds. In this case, the SNMP Profile value is updated. If polling fails for all of the profiles, the SNMP Profile value is blank.

  NOTE

  SNMP profile is shown blank for retired routers.
- **Total Interfaces**Number of interfaces for the router.
- **Enabled Interfaces**Number of enabled interfaces for the router.
- **Harvester**IP address of the parent Harvester for the router.

Available Interfaces: Interface Information

The **Available Interfaces** page includes the following options and information for interfaces.

- **Enable and Disable**
  Control whether the interface (or the router and its interfaces) is allowed to send flow to the product.
- **Standard Page Options**
  – **Search**: Search for routers or interfaces by address or name.
  – **Max per Page**: Change the number of items that are displayed.
  – Expand or collapse router contents.
  – Sort contents by column.

Interface Columns

The ifName, ifAlias, Port Name, and vrfName values are present only if the parent router is configured to provide this information to the poller. This information comes from the interfaces database. If you make changes to the corresponding properties on the **Active Interfaces** page, your changes are not shown here.

- **Enabled**Setting to let the interface send flow to the product (**Yes**) or not send flow (**No**).
- **License**
  Status that shows whether the interface has sent flow to the product (**Yes**) or has never sent flow (**No**). If the value is **No**, the system has no records for the interface.
- **ifIndex**
  Identifying index value that is automatically assigned to the interface
- **ifName**
  Interface name
- **ifAlias**
  Interface alias
- **Port Name**

Interface port name
- **vrfName**
  Virtual routing and forwarding name
- **Speed**
  Maximum data transmission speed for the interface
- **Last Flow**
  Most recent date and time that flow was processed. If flow is being collected currently, the **Last Flow** value is updated every 15 minutes.

Enable or Disable Interfaces

Interfaces are enabled automatically on the **Application Settings** page by default. You may want to enable interfaces manually in some cases, such as in the following example scenarios:

- Interfaces have been disabled temporarily and you want to re-enable them.
- The program is not configured to enable newly discovered interfaces automatically (the **Auto-Enable Interfaces** option is set to **False** in the **Application Settings** page).

Enabling an interface causes the following events to occur immediately:

- The interface can be used as a filter in Flow Forensics reports.
  The data that is available in Flow Forensics reports includes data that was collected while the interface was disabled.
- Other types of interface data begin to be collected and stored, such as data for hosts, conversations, ToS, and protocols. Once the initial period of data collection is complete, the additional interface data can be displayed in drilldown interface reports, Analysis reports, and Custom reports.

NOTE

- The interfaces for retired routers are set to disabled.
- Enable/ Disable action on interfaces of retired routers does not change anything.

Follow these steps:

1. Open the **Available Interfaces** page:
   a. Select **Administration** from the NFA console menu.
      The **Administration** page opens.
   b. Select **Interfaces, Enable Interfaces** from the menu on the left side of the page.
      The **Available Interfaces** page opens.
2. Click the arrow next to the router that contains the interfaces.
   The view expands to show the interface list. The **Enabled** status column shows which interfaces are enabled.
3. Select the check box next to one or more interfaces.
4. Click one of the following options:
   – **Enable** (enable data collection for the interface)
   – **Disable** (prevent the collection of data for the interface)

Data collection for the selected interfaces is enabled or disabled immediately.

Delete Routers from the System

If you delete a router on the **Available Interfaces** page, the router is deleted from the system entirely. The deletion removes the router, its configuration information, 15-minute (historical) data, interfaces, CVIs, and traps. The deletion also affects any aggregations, views, and reports that previously included the interfaces. Deleting an Active router deletes all the routers and interfaces including retired routers and interfaces. Deleting a Retired router deletes its data.

NOTE
If the interfaces from the deleted router begin to send flow again, a new router appears on the **Available Interfaces** page. If the program is configured to enable new interfaces automatically, the new router and interfaces also appear on the **Active Interfaces** page. The new router inherits the current tenant-domain setting of its parent Harvester. Configuration settings for the previous router are lost. If your deployment includes CA Performance Center, the router polls by using the SNMP profiles that are assigned to the Harvester tenant.

Follow these steps:

1. Verify that the router is no longer sending flows to DX NetOps.
2. Open the **Available Interfaces** page:
   a. Select **Administration** from the NFA console menu.
      The **Administration** page opens.
   b. Select **Interfaces, Enable Interfaces** from the menu on the left side of the page.
      The **Available Interfaces** page opens.
3. Locate the router and select its check box.
4. Click **Delete**.
   A confirmation message opens.
5. Click **Yes**.
   The following events result:
   – The confirmation message closes.
   – The router is deleted from the system, the **Available Interfaces** page, and the **Active Interfaces** page.
   – The router configuration information is deleted from the NFA console server.
   – All related interfaces, CVIs, 15-minute (historical) data, and traps are deleted.
   – The router is deleted from any related aggregations.
   – The data for the deleted interfaces no longer appears in the NFA console or in reports.

Define Interface Name Templates

To change the rules that determine how interface names and descriptions appear in the NFA console, use any or all of the following procedures:

- Create an interface template and make it the active template for DX NetOps.
- Edit the current interface template.
- Refine the interface naming convention for specific NFA console views by using the **Application Settings** page.

You can create multiple interface templates, but only the currently selected interface template determines the names and descriptions of interfaces in the NFA console, its views, and its printed reports.

(CA PC only) The currently selected interface template also affects interface names and descriptions in some CA Performance Center views of DX NetOps data. The following locations in CA Performance Center may display different interface names and descriptions: **Interface Pages** (**Details** tab), **Inventory** pages, and **Trend** views. To customize interface descriptions in CA Performance Center, apply **Interface Description Overrides** to specific domains.

Create or Change a Custom Interface Template

Create a custom interface template to change the way all interface names and descriptions appear in the NFA console.

Follow these steps:

1. Log in as a user who has administrator privileges for DX NetOps.
2. Display the **Interface Templates** page in the NFA console:
   a. Select **Administration** from the NFA console menu.

The **Administration** page opens.

b. Select **Interfaces, Templates** from the menu on the left side of the page.
The **Interface Templates** page opens.

3. Select an action:

Create an Interface Template

1. Click **Add**. The **Interface Templates** page displays options for adding an interface template.
2. Specify the interface template settings:
   - **Name**: Identifying text string that appears in the template list.
   - **Interface Name**: Properties, text, or properties and text that comprise the name of each interface in the NFA console.
   - **Interface Description**: Properties, text, or properties and text that appear as the description of each interface in the NFA console.

   Use properties from the following list for both settings:
   - *[DeviceAlias]*: Device (router) name that is displayed in the NFA console
   - *[DeviceName]*: DNS name or IP address of the device (router)
   - *[ifDescr]*: Interface description, which is taken from the original `ifDescr` value in the SNMP `ifEntry` table unless the interface description has been customized. If the interface description has been customized on the **Active Interfaces** page, the customized value is used.
   - *[ifAlias]*: Interface alias
   - *[ifName]*: Name of the interface, which is taken from the original `ifName` value in the SNMP `ifEntry` table value unless the interface name has been customized. If the interface name has been customized on the **Active Interfaces** page, the customized value is used.
   - *[portName]*: Port name, which may be the port number
   - *[ifIndex]:* Unique numerical identifier for the interface as defined in the SNMP `ifEntry` table
   - *[ifType]:* Interface type as defined in the `ifType` field of the SNMP `ifEntry` table
3. Click **Submit**.
   The new template is created and is added to the template list. The extra options are removed from the **Interface Templates** page.
4. (*Optional*) Apply the template:
   a. Select **Interfaces**, **Physical & Virtual** from the menu on the left side of the **Administration** page.
   b. Select one or more routers, then click **Edit**.
   c. Change the template to your new template.
   d. Click **Save**.
   The template is applied to the NFA console almost immediately. The Performance Center views that use the template reflect the changes at the next synchronization, which occurs within 5 minutes.
5. (*Optional*) Review the results of changing the template. For example, review the new labels on **Interface** pages and on the **Enterprise Overview** page.

Conventions for Interface Templates

Interface template settings consist of properties, plain text, or properties and plain text. The following conventions apply to interface templates:

- Enclose properties in square brackets.
- Separate multiple properties with a pipe symbol. The first property that returns a value is displayed. Specify enough properties to allow for any interfaces that have missing property definitions.
- To display plain text in the interface names or descriptions, include the text without enclosing it in brackets.

Change an Interface Template

Edit an interface template to change the way all interface names and descriptions appear in the NFA console when the template is selected.

Follow these steps:

1. Select the template from the list at the top of the page.
2. Click **Edit**.
   A confirmation message opens.
3. Click **OK**.
   The contents of the **Interface Name** and **Interface Description** fields become editable.
4. Specify the interface template settings:
   - **Interface Name**: Text and/or properties that comprise the names of interfaces in the NFA console.
   - **Interface Description**: Text and/or properties that appear as the description of interfaces in the NFA console.
5. Click **Update**.
   The contents of the fields are updated and are no longer editable.
   The interface names and definitions are displayed in the NFA console almost immediately. The CA Performance Center views that use the template reflect the changes at the next synchronization, which occurs automatically within 5 minutes.
6. (*Optional*) Review the results of changing the template. For example, review the new labels on **Interface** pages and on the **Enterprise Overview** page.

Change the Application Setting for Interface Names

You can use a setting on the **Application Settings** page to change the naming convention for interfaces. The default setting adds the device name in front of the interface name. This setting affects the way interface names appear in some NFA console report views, such as the **Enterprise Overview** views, **Interface** pages, and **Custom Report Interface Summaries**.

The interface names that result can include unwanted duplications. If a device and an interface are both named Device1, for example, by default the interface name is shown as Device1::Device1. The following example illustrates this repetition.

Status	Interface ▲
	Device1::Device1 - 0
	Device2::Device2 - 0

To eliminate the duplication, edit the **Show Device Name** setting on the **Application Settings** page.

Follow these steps:

1. Display the **Application Settings** page:
   a. Select **Administration** from the NFA console menu.
      The **Administration** page opens.
   b. Select **System, Application Settings** from the menu on the left side of the page.
      The **Application Settings** page opens.
2. Set the **Show Device Name** value to **False**.
3. Click **Save**.
   With the **Show Device Name** value set to **False**, device names in reports and views are not prepended to interface names.

Working with Interface Aggregations

You can create and manage interface aggregations in the NFA console. An interface aggregation combines traffic from two or more interfaces so the traffic is reported together. For example, suppose you have two load-balanced circuits and you want to report on their interfaces as a single unit. Aggregations let you report on interfaces as a single unit without setting up a custom report for that purpose. All aggregations appear in the **Interface Index** under the label *Aggregations*.

> **NOTE**
> You can aggregate interfaces only when their data is collected by the same Harvester. Do not attempt to aggregate interfaces from multiple Harvesters.

Create, Change, or Delete Interface Aggregations

Interface aggregations combine traffic from two or more interfaces so the traffic is reported together.

For example, suppose that you create aggregations for each of several geographical regions. Operators use the aggregations in reports to review and compare the totals for specific regions. Before you created the aggregations, operators had the following problems:

- Data was scattered throughout reports.
- Values were not totaled by region, so quick overviews and comparisons were difficult to make.
- Operators spent too much time designing specialized reports to collect the data and create the totals they needed.

Follow these steps:

1. Display the **Interface Aggregations** page:
   a. Select **Administration** from the DX NetOps console menu.
      The **Administration** page opens.
   b. Select **Aggregations** from the **Interfaces** menu.
      The **Interface Aggregations** page opens and shows the list of current interface aggregations.
2. Select an action:

Create an Aggregation

1. Click **New**.
   The page changes to show options for adding an aggregation.
2. Select interfaces to add to the aggregation by using any of the following methods:
   – *Select routers:* Select the check boxes next to one or more router names. You can select one router, multiple routers, or a combination of routers and individual interfaces from any routers that are not selected. When a router has been selected, you cannot select any of its individual interfaces.
   – *Select interfaces:* Click **Expand All** or click the arrow icon next to a router name, then select the interfaces from the list. To select all the interfaces for a router, select the check box in the heading row.
   – *Filter the list to find routers or interfaces:* Filter your view by entering a text string in the **Search** box, then clicking **Search**. You can search for a full or partial name or address of a router or interface. The router/interface list is filtered to display matching routers or routers that contain matching interfaces.
   – *Filter the list to check your selections:* Click **Show Selected** to show only the routers that are selected or that contain selected interfaces. You can combine this function with a search string and with **Expand All** to locate and review your selections quickly.
3. Specify the following values:

- **Routers and/or Interfaces**: Select one or more of the routers or interfaces that are listed at the bottom of the page. Your selections are added to the aggregation.
- **Aggregation Name**: Specify a name for the aggregation.
- **Description**: (*Optional*) Add a notation to help identify the aggregation.
- **In Speed**: (*Optional*) Specify the inbound speed of the selected interfaces.
- **Out Speed**: (*Optional*) Specify the outbound speed of the selected interfaces.
  If you do not specify the **In Speed** and **Out Speed**, both values are set to 0 by default. Inaccurate speeds cause some report results to be inaccurate, such as utilization percentages.
- **Type**: (*Optional*) Select the mode of interface connection from the **Type** list, such as WAN or Ethernet.
  If you do not specify the interface type, the type is set to **Unknown** by default.

4. Click **Save**.
   The aggregation is automatically deployed within one minute of creation.

Change an Aggregation

1. Click the check box next to the aggregation that you want to edit, then click **Edit**.
   The page changes to the **Edit Aggregation** view, which includes editable options.
2. Make any changes that are needed, such as adding or removing interfaces or changing the aggregation name, description, in/out speed, or type.
3. Click **Save**.
   Your changes are saved and you return to the list of aggregations.

Delete an Aggregation

1. Select the check box next to each aggregation that you want to delete.
2. Click **Delete**.
3. Click **Yes** in the confirmation message box that opens.
   All of the selected aggregations are deleted automatically.

Working with Harvesters

Use the **Harvester** page to view, add, delete, and edit details about the Harvesters that supply flows to the NFA console.

Follow these steps:

1. Open the **Harvester** page:
   a. Select **Administration** from the NFA console menu.
      The **Administration** page opens.
   b. Select **System, Harvester** from the menu on the left side of the page.
      The **Harvester** page opens and displays a table of information about the current Harvesters.
2. Select an action to perform on the Harvester:

Add a Harvester

Make sure the Harvesters you add have not been deleted from the **Harvester** page previously. To add a Harvester instance successfully in DX NetOps after deleting it, the Harvester installation server must be re-imaged and the Harvester software must be re-installed.

Follow these steps:

1. Click **Add**.
   The **Add Harvester** dialog opens.
2. Enter the following information:

- **IP Address**
  Address of the Harvester server.
- **Description**
  Identifying text about the Harvester, which appears in the **Harvester** page table.
- **Domain**
  Parent tenant and domain combination for the Harvester and for any routers and interfaces that begin to supply data to the Harvester.
  Changing this setting affects the tenant-domain association for new routers that begin exporting flow data and any new interfaces that begin generating flow.
  In a multi-tenant environment, the Harvester tenant affects which SNMP profiles are available for routers to poll interfaces.
  The domain affects which operators and reports have access to the data from routers and interfaces.
  This option is visible only in an environment that contains multiple domains.

3. Click **Save**.
   The new Harvester is added and appears in the Harvester list, provided that the IP address passes the connection test. If the test connection to the web service fails, an error message opens.
   The usual process is to add one or more Harvesters, then configure the router interfaces to export flow to the Harvesters. If you configure the routers to export flow to the Harvesters first, the DX NetOps console immediately begins to collect data from the new Harvester. In this case, the domain for the routers is set at the time you add the parent Harvester.

Edit the Details for Harvesters on the Harvester Page

You can edit the IP address, description, and tenant-domain setting.

Follow these steps:

1. Click **Edit** on the Harvester row that you want to edit.
   The **Edit Harvester** dialog opens.
2. Change any of the following settings:
   - **IP Address**
     Address of the Harvester
   - **Description**
     (*Optional*) Additional information to help identify the Harvester
   - **Domain**
     The tenant-domain association of the Harvester in a multi-domain deployment.
     Changing this setting affects the tenant-domain for any new routers that begin exporting flow data. Existing routers and interfaces retain their previous tenant-domain associations.
     The domain affects which operators and reports have access to the data from routers and interfaces.
     Changing the tenant of a router in CA Performance Center can affect which SNMP profiles are available for polling. This is not applicable to CA NetQoS Performance Center, which uses the same list of SNMP profiles for all routers.
     Default: Default Tenant\Default Domain

   If no custom IP domains have been created, the **Harvester** table includes only the **IP Address** and **Description** columns.
3. Click **Save**.
   Your changes are saved immediately and appear in the **Harvester** table.

Delete a Harvester

Once you delete a Harvester, you cannot recover any of the data that the Harvester collected previously.

Follow these steps:

1. Click **Delete** in the row for the Harvester that you want to delete.

A confirmation message opens.

2. Click **Yes** to confirm that you want to delete the Harvester.
   The Harvester is deleted and is not listed in the **Harvester** table. The NFA console no longer collects data from the routers that are associated with the deleted Harvester. The data that was collected from those routers previously is not available in reports.

Creating Names and Groups for Protocols, ToS, and AS Data

You can create and manage protocol groups, ToS labels, ToS groups, and customized autonomous system (AS) names from the **Administration** page. You use several different pages to perform the functions.

Protocol Group Configuration Page

You use the **Protocol Group Configuration** page to view the existing protocol groups and their contents. You can add, edit, and delete custom protocol groups.

On the **Administration** page, select **Define an Application, Protocol Groups** from the menu on the left side of the page. Use the **Protocol Group Configuration** page to perform the following main tasks:

- Create a shell protocol group.
- Configure the protocol group.
- Review and edit the group.

ToS Configuration Page

You use the **ToS Configuration** page to view and edit the labels and descriptions for ToS values.

On the **Administration** page, select **Define an Application, ToS Names** from the menu on the left side of the page. Use the **ToS Configuration** page to perform the following task:

- Label ToS values.

ToS Group Configuration Page

You use the **ToS Group Configuration** page to view the existing ToS groups and their contents. You can add, edit, and delete custom ToS groups.

On the **Administration** page, select **Define an Application, ToS Groups** from the menu on the left side of the page. Use the **ToS Group Configuration** page to perform the following main tasks:

- Create a shell ToS group.
- Configure the ToS group.
- Edit ToS groups.
- Delete ToS groups.

AS Names Page

You use the **AS Names** page to view and edit AS names.

On the **Administration** page, select **Define an Application, AS Names** from the menu on the left side of the page. Use the **AS Names** page to perform the following main tasks:

- Review AS names.
- Edit AS names.

Create Protocol Groups

Protocol groups can be used to filter data in Custom and Analysis reports. As an administrator, you can set up custom protocol groups that contain the protocols for particular types of network traffic.

For example, suppose that operators want to report on network traffic for different types of applications, such as email, videoconferencing, VoIP, and streaming media. No default protocol groups are defined for this purpose, so you create a custom protocol group for each of the application types. Each protocol group includes all the protocol values that are used in your enterprise for the target applications.

Operators can use a custom protocol group without having a thorough knowledge of the way each protocol is used and without adding individual protocol filters to report definitions.

Create a Shell Protocol Group

Create a shell protocol group that you can configure to filter reports for a particular type of network traffic.

Follow these steps:

1. Log in as a user who has administrator privileges for DX NetOps.
2. Display the **Protocol Group Configuration** page:
   a. Select **Administration** from the NFA console menu.
      The **Administration** page opens.
   b. Select **Define an Application, Protocol Groups** from the menu on the left side of the page.
      The **Protocol Group Configuration** page opens and displays information about the currently selected protocol group.
3. Click **Add**.
   The options for identifying a new protocol group are displayed on the **Protocol Group Configuration** page.
4. Enter values in the **Group Name** and **Description** boxes.
   The name and description appear in the following locations in DX NetOps:
   – List of protocol groups on the **Protocol Group Configuration** page
   – **Protocol Group Index** that an operator can display from the report wizard for defining an Analysis or Custom report

   A new shell protocol group is created. The new protocol group name is added to the list of protocol groups on the **Protocol Group Configuration** page.

Configure the Shell Protocol Group

Configure the shell protocol group to represent a particular type of network traffic.

Follow these steps:

1. Select the protocol group from the **Select a Protocol Group** list on the **Protocol Group Configuration** page.
2. Click **List**.
   Configuration options are added to the **Protocol Group Configuration** page.
3. (*Optional*) Review and correct the **Domain** setting if necessary.
   The **Domain** setting is displayed only in an environment that has multiple domains.
   The contents of the selected protocol group are displayed with the protocol names for the selected tenant-domain combination. If an Administrator defines domain-specific protocol names, selecting that domain displays the appropriate protocol names.
   The **Domain** setting does not restrict access to the protocol group or to reports that use the protocol group as a filter.
4. Select the protocols for the group:
   a. Click **Add/Remove**.
      A dialog opens, which shows a list of available protocols and a list of any protocols that are included in the list currently.

If the **Add/Remove** link is not visible, click **List**.

b. Select the protocols in the top pane that you want to add to the group.
To select multiple protocols to add simultaneously, use the **Shift** or **Control** key.
To filter the protocol list, enter a search string in the **Filter the Protocol List** field box, then click **Apply**. For example, to show only UDP protocols, enter **udp**.

c. Click **Add**.
The selected protocols are added to the protocol list.

d. Select any protocols in the bottom pane that you want to remove from the group.

e. Click **Remove**.
The selected protocols are removed from the protocol list.

f. Click **Done** when you finish configuring the contents of the protocol group.
To review a list of protocol groups that contain a particular protocol, select the protocol and click **Jump to Protocol**. The **Protocol Configuration** page opens and displays all of the protocol lists that contain the selected protocol.

The protocol group is configured with your settings. Operators can select the configured protocol group as a filter for an Analysis or Custom report.

NOTE
Operators can define reports only if their user settings are set up to enable this capability.

Change a Shell Protocol Group

After you create a protocol group and operators use it in reports, you may want to make changes. For example, you may want to rename the protocol group to reflect the way operators use it. You also may want to change the list of protocols in the group.

Follow these steps:

1. Select the protocol group from the **Select a Protocol Group** list on the **Protocol Group Configuration** page.
2. (*Optional*) Edit the name or description of the protocol group:
   a. Click **Edit**.
   The **Edit a Protocol Group** page opens.
   b. Change the values for the **Group Name** and **Description** fields to help identify the purpose of the protocol group more clearly.
   c. Click **Submit**.
   The name and description are updated in the **Protocol Group Configuration** page list and in the **Protocol Group Index**.
3. (*Optional*) Review and revise the list of protocols that are included:
   a. Click **List**.
   A table is added to the bottom of the page, which lists the protocols.
   b. Select the appropriate domain from the **Domain** list, if custom protocol names have been defined.
   The **Domain** setting is displayed only in an environment that has multiple domains.
   c. Click **Add/Remove** and change the contents of the protocol group as needed
   d. Click **Done** when you finish configuring the contents of the protocol group.

   The protocol group is configured with your settings. Operators can select the reconfigured protocol group as a filter for an Analysis or Custom report.

Label ToS Values

You can create labels (or descriptions) for ToS values to ensure that operators know which service or application the ToS values represent. If you do not label ToS values, the numeric ToS value is displayed by default.

NOTE

- The ToS labels that you create affect all uses of the ToS in the assigned domain, but do not affect ToS labels in other domains. ToS labels are domain-specific in a deployment that has multiple domains.
- You can display custom ToS names in trap alerts (in a single-domain environment).

Follow these steps:

1. Open the **ToS Configuration** page:
   a. Select **Administration** from the DX NetOps console menu.
      The **Administration** page opens.
   b. Select **Define an Application, ToS Names** from the menu on the left side of the page.
      The **ToS Configuration** page opens and displays information about the selected ToS and domain.
2. (*Multiple-domain environment*) Select the domain that contains the ToS values that you want to edit.
3. Select a ToS value from the list, then click **Edit**.
   The **Description** becomes editable.
4. Enter the new description for the ToS value in the **Description** field and click **Save**.
   Your change is saved and the **Description** and **Value** fields for the selected ToS are updated.

You can also use the **ToS Configuration** page to add or delete ToS values to ToS groups.

1. Select the domain and the ToS value.
2. Click the **Add/Remove** link at the top of the ToS groups list.
3. The **ToS Group Index** dialog opens, which you use to make changes for user-configured groups. (You cannot change the built-in **All ToS** group.)

Create and Manage ToS Groups

An administrator can create ToS groups that act as filters for report data. The administrator sets up each ToS group to contain the ToS values that characterize a particular type of network traffic. Operators can use the ToS group as a filter in reports instead of adding filters for each ToS value.

For example, suppose that operators want to report on network traffic for the Low Drop Assured Forwarding group. The Low Drop Assured Forwarding group is a group of ToS values that are assigned to applications that should have priority over less critical traffic. The administrator creates a ToS group named Low Drop, which includes all the ToS values for AF11, AF21, AF31, and AF41. Operators can create a report about the application traffic by using the new Low Drop ToS group.

NOTE
In an environment that includes domains, ToS groups are available in all domains as filters for Analysis reports and Custom reports. The ToS labels that administrators see displayed for the contents of a ToS group are domain-specific, however.

Create, Change, or Delete a ToS Group

Create ToS groups to help users get optimal results from Analysis and Custom reports quickly.

Follow these steps:

1. Log in as a user who has administrator privileges for DX NetOps.
   a. Select **Administration** from the console menu.
      The **Administration** page opens.
   b. Select **Define an Application, ToS Groups** from the menu on the left side of the page.

The **ToS Group Configuration** page opens and displays information specific to the selected ToS group and domain (if applicable).

2. Select an action:

Create a ToS Group

1. Click **Add**.
   The **ToS Group Configuration** page displays options for identifying a new shell ToS group.
2. Enter values in the **Name** and **Description** boxes.
   The name and description appear in the following locations:
   – List of ToS groups on the **ToS Group Configuration** page
   – **ToS Group Index** that a user can display from the report wizard for defining an Analysis report or Custom report
3. Click **Add**.
   A new shell ToS group is created. The new ToS group name appears in the list of ToS groups on the **ToS Group Configuration** page.

Add ToS Values to a ToS Group

1. Select the ToS group from the **Select a ToS group** list on the **ToS Group Configuration** page.
   a. (*Optional*) Select a tenant-domain combination from the **Domain** list, provided that you have multiple domains and domain-specific ToS labels.
      The **Domain** setting is displayed only in a multi-domain environment.
      By selecting a tenant-domain combination, you display any available domain-specific ToS labels, which help you identify the ToS values to use. The ToS group will not be restricted to the selected domain: ToS groups are available in all domains.
2. Click **Add/Remove**.
   The **ToS Index** dialog opens. The ToS list displays the ToS index values and the ToS labels that have been defined. In a multi-domain environment, the ToS labels are displayed for the currently selected domain.
3. Select the ToS values to add by clicking their check boxes.
4. Scroll to the bottom of the list and click **Save**.
   The ToS values are added to the ToS group.
   An operator can select the ToS group from the **ToS Group Index** when defining an Analysis or Custom report.

   NOTE
   To define reports, the operator must have the necessary role and permission group settings.

Change the Contents of a ToS Group

1. Select the ToS group to edit from the **Select a ToS group** list.
   The **ToS Group Configuration** page opens and displays information for the currently selected ToS group and domain (if applicable).

   NOTE
   You cannot edit built-in groups, such as the **All ToS** group in the **Default Domain**.
2. (Multi-domain environment only) Select a tenant-domain combination from the **Domain** list, provided that you have multiple domains and ToS labels. In a multi-domain environment, ToS labels are domain-specific.
   The **Domain** setting is displayed only if multiple domains exist.
3. Click the **Add/Remove** link above the list of ToS values that are included in the group.
   The **ToS Index** dialog opens. The ToS list displays the ToS labels that have been defined.
4. Select or clear check boxes to include only the ToS values you want.
5. Click **Save** when your changes are complete. The list of ToS values is updated.

Delete a ToS Group

1. Select the ToS group to delete from the **Select a ToS group** list.
   Select a configurable ToS group. For example, you cannot delete the **All ToS** group.

   NOTE
   The **Domain** setting, if any, has no effect on the list of available ToS groups. If you delete a ToS group, it is deleted for all domains.
2. Click **Delete**.
   A confirmation message opens.
3. Click **OK**.
   The selected ToS group is deleted. The list of ToS values is updated.

Customize AS Names

Interface reports that show data about Autonomous System (AS) traffic typically label the AS traffic by name and number. Administrators can customize AS names to make the AS references in reports shorter or more descriptive.

For example, suppose that operators frequently view reports about network traffic for AS 4000000. The label with the default AS name is "UUNET-CANADA - Progressive Communications Services, Inc. d/b/a Acme Business (4000000)." The administrator can customize the AS name so that the report label is "Acme (4000000)."

In a multi-domain environment, AS names are domain-specific. In this type of environment, the AS names that appear in reports are taken from the domain of the interface for the report.

Prerequisites

These instructions assume that you:

- Installed the DX NetOps software on the servers that collect and process the data.
- Configured your NetFlow or NetFlow-compliant flow to support AS reporting.
  To view meaningful AS data in reports, you must enable it: NetFlow does not export full AS information by default. For information about enabling AS data, refer to the Knowledge Base article TEC562036, *Viewing AS Numbers in Reports*, at CA Support (http://ca.com/support).
  AS data is shown as AS 0 in reports when any of the following conditions apply:
  - Flow is not configured to support AS reporting.
  - The data source route is unknown.
  - The data originates from within the local system.

Review Autonomous System Names

Review the AS references in the interface reports for your enterprise and locate any AS numbers in use that have long or unclear AS references.

The interface reports that contain AS references are:

- Single interface AS Next Hop summary table
- Top N AS summary tables, trend charts, and pie charts

Follow these steps:

1. Collect information about which AS names are used in interface reports for your enterprise.
   The list of all AS numbers and names is extensive. You will not want to customize every AS name in the list.
   For example, compile a list of commonly used AS names by examining AS reports or by asking operators which AS numbers they track.

2. Display the **AS Names** page:
   a. Log in as a user who has administrator privileges for DX NetOps.
   b. Select **Administration** from the NFA console menu.
      The **Administration** page opens.
   c. Select **Define an Application, AS Names** from the menu on the left side of the page.
      The **AS Names** page opens and displays information about the first page of AS values.
3. Locate each AS with a name that you want to customize:
   – To change the number of rows on each page, use the **Max per Page** option.
   – To view a different page, use the navigation elements at the bottom of the page.
   – To search, enter one or more text strings in the **Search** box, then click **Search**.
     Rules for matches:
     • Searches are case-insensitive.
     • All of the specified search strings must be found in either the **AS Number** or **Description** column.
     • Enclose the search string in double quotation marks if you want order and completeness to be limiting factors. For example, matching results for *internet back* without quotation marks include *internet global backbone* and *Backbone Internet Service*.
4. Review the AS names that appear in the **Description** column.
   Once you locate a default AS name that requires customization, you are ready to edit the AS name.

Edit Autonomous System Names

Edit AS names as needed to make interface report labels more user-friendly.

Follow these steps:

1. (*Optional*) Review and correct the **Domain** setting if necessary.
   In a multi-domain environment, AS names are domain-specific. In this case, changes to AS names affect reports about interfaces in the selected domain.
2. Select the row for the AS number that you want to edit.
3. Click **Edit**.
   The **Edit AS Number Description** dialog opens.
4. Enter the custom name (description) in the **New Name** box.
   The **Old Name** value is the official (base) name for the AS, which cannot be edited.
5. Click **Save**.
   The name that you specified appears in the **Description** column for the selected AS number. All custom AS names are shown in bold.
   The updated AS names also appear in labels and other references in the following DX NetOps Interface views:
   – Single interface AS Next Hop summary table
   – Top N AS summary tables, trend charts, and pie charts
6. Repeat these steps for each AS that you want to customize.

   NOTE
   To restore the AS name to its official value, click **Reset** on the appropriate row, then **OK** in the confirmation box that opens.

Report Customizations

You can make additional customizations to ensure that operators get the best results from reports.

Once the product is registered as a data source, you use the Performance Center Console to administrate users, roles, SNMP profiles, and many types of groups.

This section describes how to perform the following customization tasks in the NFA console:

Create, Change, or Delete Time Filters

Use time filters to help users create reports that contain streamlined data. For example, suppose that your users need some reports that describe network traffic during business hours. You create a time filter for Monday through Friday from 8:00 A.M. to 5:00 P.M. You may also want to create time filters for the time frames of specific operations that occur in your environment, such as automated backups.

You can also use the time filters that you create to configure traps on the **Trap Configuration** page.

The custom time filters are available to users in **Time Filter** option lists in the following locations:

- **Specify Schedule** page of the **Custom Report** wizard or **Analysis** wizard.
- Options that operators display by clicking the time period setting in a drilldown interface report of any of the following types: Overview, Protocols, ToS, Hosts, Conversations, Flows, Utilization, and AS Numbers.

Follow these steps:

1. Select **Administration** from the NFA console menu.
   The **Administration** page opens.
2. Select **Reporting, Time Filters** from the menu on the left side of the page.
   The **Time Filter Configuration** page opens and displays a list of the currently configured time filters.
3. Select one of the following actions:

Create a Time Filter

1. Click **Add**.
   The **Time Filter Configuration** page displays options for adding a time filter.
2. Enter values for the following options:
   - **Time Filter Name**: Define the time filter identifier, which appears in the following lists:
     - **Time Filter Configuration** page list (for administrators)
     - Time Filter options that are available on the **Specify Schedule** page of the **Custom Report** wizard
     - Time Filter options that are available when a user or administrator clicks the time period setting in a drilldown interface report of any of the following types: Overview, Protocols, ToS, Hosts, Conversations, Flows, Utilization, and AS Numbers
   - **Description**: (*Optional*) Add information to identify the time filter, which appears in the list of time filters on the **Time Filter Configuration** page.
   - **on**: Accept the default settings (Monday through Friday) or select other days for collecting report data.
   - **Start Time** and **End Time**: Accept the default settings for the start and end of the daily time span or select custom settings, using the 24-hour clock system.
     For example, to restrict the reporting period to business hours, select 08:00 as the start time and 17:00 as the end time. To set up a filter for backups that run from 11 P.M. Tuesday to 3:00 A.M. Wednesday, select Tuesday, then select 23:00 and 03:00.

     NOTE
     Time filters are applied using the time zone that was set for the report when the report was created. Mailed reports use the time zone set in the report.
3. Save the time filter by clicking one of the following buttons:
   - **Submit, finished**: Save the time filter and return to the time filter list.
   - **Submit, add another**: Save the time filter and keep the page open so you can configure another time filter.

Edit a Time Filter

1. Select a time filter from the list of available filters.

2. Change any of the options that are displayed.
3. Click **Submit**.

Delete a Time Filter

1. Select a time filter from the list of available filters.
2. Click **Delete**.
3. Confirm the deletion when prompted.

Create, Change, or Delete Reporting Periods

Create reporting periods to get the best results from reports. For example, suppose that your users typically analyze network traffic that occurs over two weeks. To make it easier to display interface data over this timespan, you create a two-week reporting period.

Custom reporting periods are available as **Time Period** options when a user clicks the time period setting in a drilldown interface report of any of the following types: Overview, Protocols, ToS, Hosts, Conversations, Flows, Utilization, and AS Numbers.

Follow these steps:

1. Select **Administration** from the NFA console menu.
   The **Administration** page opens.
2. Select **Reporting, Reporting Periods** from the menu on the left side of the page.
   The **Reporting Periods Configuration** page opens and displays a list of the current reporting periods, including the built-in and customized reporting periods.
3. Perform one of the following actions:

Create a Reporting Period

1. Click **Add**.
   Fields and options for adding a reporting period are displayed.
2. Specify the following values:
   - **Reporting Period**: Identifier for the reporting period, which appears in the following lists of available reporting periods:
     - **Reporting Periods Configuration** page list (visible to administrators)
     - **Time Period** options available when a user or Administrator clicks the time period setting in a drilldown interface report of any of the following types: Overview, Protocols, ToS, Hosts, Conversations, Flows, Utilization, and AS Numbers.
   - **Duration**: Number of time units and type of time units: years, months, weeks, days, or hours.
   - **Description**: (Optional) Additional notes to help identify the reporting period. The description appears in the **Reporting Periods Configuration** page list and is visible only to administrators.
3. Save the reporting period by clicking one of the following buttons:
   - **Submit, finished**: Save the current reporting period and return to the list of available reporting periods.
   - **Submit, add another**: Save the current reporting period and keep the Add options open so you can continue to configure additional reporting periods.

Change a Reporting Period

1. Select the reporting period from the list.
2. Change any of the options that are displayed.

3. Click **Submit**.

Delete a Reporting Period

1. Select the reporting period from the list.
2. Click **Delete**.
3. Confirm the deletion when prompted.

Set Up Application Mapping

Create application mapping rules to identify traffic in reports. Rules can identify traffic types such as a ToS, host, subnet, or NBAR2 application.

You can use application mapping to combine, differentiate, or more clearly identify traffic in reports:

- Differentiate Traffic: Separate larger traffic blocks by re-mapping traffic sub-types to separate destination ports.
  For example, suppose reports show a large block of FTP traffic on TCP port 20. You want to track the FTP traffic from your internal FTP server separately from internet traffic. To accomplish this, you create a Host application mapping rule, which you name *Internal FTP Traffic.* The Host value of the rule matches the IP address of the internal FTP server. The Port value is 20. You specify 65000 as the Destination Port; a port that does not currently receive any traffic.
  Reports now show traffic from the FTP server on TCP port 65000 with the label *Internal FTP Traffic.* Other TCP port 20 traffic is still labeled *FTP.*
- Combine Traffic: Report traffic of different types as a single unit by re-mapping them to a single destination port.
  For example, suppose your enterprise mail systems use the IMAP and POP protocols. The IMAP mail uses TCP port 443 and the POP mail uses TCP ports 109 and 100. You want reports to show the combined mail traffic, so you create application mapping rules that re-map each type of mail traffic to port 3100. The traffic is combined in reports and is labeled with the rule name, *Mail*. Even though you created several rules, the program uses the same name for all of the rules that map traffic to port 3100.
- Identify Traffic: Use application mapping to re-label traffic without combining or separating it.

Application mapping affects the following reports:

- **Enterprise Overview** page: **Top Protocols** report
- **Interface** page: All reports that show protocol data
- **Custom Reporting** page and **Analysis** page: **Protocol Index**, which you use to select protocols to filter a new or edited report.

Notes:

- Application mapping does not affect Flow Forensics reports.
  To continue the example for differentiating types of FTP traffic, the **Flow Forensics Session Protocols** reports show the FTP traffic as it was before you mapped the FTP sub-category.
  The Flow Forensics reports that display NBAR2 data show the official application name and ID regardless of any application mapping rules that you have.
- Reports show mapping results within the time frame that the rules are in effect. If you create rules today, reports of last week's data do not show the effects of the rules. If you create, delete, or edit application mapping rules within a report time frame, the report may show a dramatic change at the time you made the rule changes.

You can perform the following application mapping tasks:

- Use Administration functions in the NFA console to set up or modify application mapping rules to aggregate or segregate data:

  - Create an All (ToS) application mapping rule
  - Create a Host application mapping rule
  - Create a Subnet application mapping rule
  - Create an NBAR2 application mapping rule
  - Edit application mapping rules
- Configure global settings to support application mapping rules in reports
- Understand how application mapping rule priorities work
- Perform batch Import operations by using a `.csv` file:
  - Import the default NBAR2 application mapping rules
  - Import custom application mapping rules
  - Import application mapping rule updates
  - Review errors for failed rule imports

Application Mapping Priorities

If incoming flow matches the combination of criteria that are specified in an application mapping rule, the flow is mapped to the rule destination port. If rules conflict, the following priorities apply. A rule takes precedence when its criteria are ranked higher than the criteria of another rule.

- *Priority 1:* Host rule with Host, Protocol, Port, and ToS specified
- *Priority 2:* Host rule with Host, Protocol, and Port specified, but with the ToS set to ALL
- *Priority 3:* All (ToS) rule with ToS specified
- *Priority 4:* Host rule with the Host specified, but with the Protocol and ToS set to ALL
- *Priority 5:* Subnet rule
- *Priority 6:* NBAR2 rule

Configure Global Settings for Application Mapping

Review the global settings that affect application mappings. Make any adjustments that are needed to customize application mapping behavior. The global settings that affect application mapping include the following options:

- *TCP Rebase Port:* Target port for redirected TCP traffic. If an application mapping rule sends TCP traffic to a port that already has native (non-mapped) traffic, the native traffic is redirected to the TCP rebase port.
- *ToS Mask:* Setting to determine whether all ToS bits are used for ToS values.
- *UDP Rebase Port:* Target port for redirected UDP traffic. The UDP rebase port receives redirected native UDP traffic in the way the TCP rebase port receives redirected native TCP traffic.
- *Preserve ToS Map Proto:* Setting to determine whether protocol values are reported in addition to ToS values.

Follow these steps:

1. Open the **Application Definitions** page:
   a. Select **Administration** from the NFA console menu.
      The **Administration** page opens.
   b. Select **Define an Application**, **Application Definitions** from the menu on the left side of the page.
      The **Application Definitions** page opens.
2. Click **Additional Settings**.
   The **Application Settings** page opens.
3. Review and correct the global settings that affect application mapping:
   - **TCP Rebase Port**: Target port for redirecting native TCP traffic, which is used if an application mapping rule sends TCP traffic to a port that already receives native (non-mapped) traffic.

For example, suppose that you create an application mapping rule that has port 655 as the destination port. You set the TCP Rebase Port value to 630. If port 655 is receiving native (non-mapped) traffic, the native traffic is redirected to port 630.

NOTE
The rebase port is used to avoid merging mapped traffic with other types of traffic. If application mapping rules are configured with unused destination ports, the rebase port is not used: Reports do not show traffic on the rebase port. If your reports show traffic on the rebase port, you may want to specify a new destination port for the mapped traffic. Examine the rebase port traffic to determine which rules are involved.

- **ToS Mask**: Limits the 8-bit ToS values reported for flow. The default value for this setting is 255, which enables all ToS bits.
- **UDP Rebase Port**: Redirects native UDP traffic in the same way the TCP Rebase Port setting redirects native TCP traffic.
- **Preserve ToS Map Proto**: Setting to determine whether protocol values are reported in addition to ToS values. **Y**(es) retains the protocols that are used and reports ToS data separately. **N**(o) shows you the ToS values for traffic, but you do not see which protocols are involved. The default setting is **Y**.

4. Click **Save**.
   Your changes are saved. The **Application Settings** page remains open.

Create an All (ToS) Application Mapping Rule

Create an *All application mapping rule* to combine, separate, or more clearly identify traffic by its ToS (Type of Service) value.

Follow these steps:

1. Open the **Application Definitions** page:
   a. Select **Administration** from the NFA console menu.
      The **Administration** page opens.
   b. Select **Define an Application**, **Application Definitions** from the menu on the left side of the page.
      The **Application Definitions** page opens.
2. Verify that **Application Mapping** is the selected value for **Rules**.
3. Click **Add Rule**.
   The **Add Application Mapping** dialog opens.
4. Select **All** from the list of rule types at the top of the dialog.
   The **Add Application Mapping** dialog switches to **All (ToS)** rule mode.
5. Specify the setting values:
   - **ToS**: Type of Service (ToS) to use as a filter for the collected data.
     ALL is the ToS value that is shown initially when you open the dialog. If left unchanged, this setting would map traffic for all ToS values to the destination port.
   - **Destination Port**: Target port that collects the mapped data
     If you specify a destination port that is already used by other rules, the traffic for the related rules will be combined.
   - (*Optional*) Click **Check** to run a general check to detect whether the specified port is already receiving data. The check fails if the port is receiving native data--that is, data that is not mapped by application mapping rules. If any native data is on the specified destination port, that data is redirected to the rebase port.
   - **Name**: Identifier for the rule as it is listed on the **Application Definitions** page
     The rule name also is the label for the mapped traffic in certain reports. If other rules map traffic to the same destination port that you specify for this rule, specify the name that you want to use for the combined traffic.
   - **Description**: (Optional) Additional descriptive text to identify the rule type and its use, which is displayed only on the **Application Definitions** page
6. Click **Save**.

The dialog closes. The new rule is added to the **Application Mapping** rule list. If any other rules map traffic to the same port and you specified a new rule name, the other rule names are updated.

7. (*Optional*) Run reports to verify that the traffic on the designated destination port fits the rule.
8. (*Optional*) Review the effects of the new or changed application mapping rules on reports, then consider renaming the rule to label the mapped traffic more clearly in reports.

Create a Host Application Mapping Rule

Create a Host application mapping rule to combine, separate, or identify traffic based on its source host. For example, a Host application mapping rule can report the total traffic from a particular server or from an application on the server.

Follow these steps:

1. Open the **Application Definitions** page:
   a. Select **Administration** from the NFA console menu.
      The **Administration** page opens.
   b. Select **Define an Application, Application Definitions** from the menu on the left side of the page.
      The **Application Definitions** page opens.
2. Verify that **Application Mapping** is the selected value for **Rules**.
3. Click **Add Rule**.
   The **Add Application Mapping** dialog opens.
4. Select **Host** from the list of rule types at the top of the dialog.
   The **Add Application Mapping** dialog switches to **Host rule** mode.
5. Specify values for the following settings:
   – **Host**: IP address of the server whose network traffic you are mapping
   – **ToS**: Type of Service (ToS) to use as a filter for the collected data. To match all ToS values, accept the default value of **ALL** or leave the ToS box blank.
   – **Protocol:** Protocol of the data that is affected by the rule, either TCP or UDP
   – **Port**: Port to use for collecting data. To match all ports, accept the default value of **ALL** or leave the Port box blank.
   – **Destination Port**: Target port that collects the mapped data
     If you specify a destination port that is already used by other rules, the traffic for the related rules will be combined.
     (*Optional*) Click **Check** to run a general check to detect whether the specified port is already receiving data. The check fails if the port is receiving native data; that is, data that is not mapped by application mapping rules. If any native data is on the specified destination port, that data is redirected to the rebase port.
   – **Name**: Identifier for the rule as it is listed on the **Application Definitions** page
     The rule name also is the label for the mapped traffic in certain reports. If other rules map traffic to the same destination port that you specify for this rule, specify the name that you want to use for the combined traffic.
   – **Description**: (*Optional*) Additional descriptive text to identify the rule type and its use, which is displayed only on the **Application Definitions** page
6. Click **Save**.
   The dialog closes. The new rule is added to the **Application Mapping** rule list. If any other rules map traffic to the same port and you specified a new rule name, the other rule names are updated.
7. (*Optional*) Run reports to verify that the traffic on the designated destination port fits the rule.
8. (*Optional*) Review the effects of the new or changed application mapping rules on reports, then consider renaming the rule to label the mapped traffic more clearly in reports.

Create a Subnet Application Mapping Rule

Create a subnet application mapping rule to combine, separate, or more clearly identify traffic that originates from a particular subnet and mask. For example, a subnet rule can enable reports to show the total traffic for an application.

DX NetOps applies subnet application mapping rules by first finding the most specific subnet match, then looking at all port ranges defined for that subnet. If no matches are found, either by subnet or port range within the most specific matching subnet, any rules defined for the 0.0.0.0/0 subnet are evaluated. If overlapping subnets are defined (other than 0.0.0.0/0), only port ranges for the most specific subnet match are considered.

Follow these steps:

1. Open the **Application Definitions** page:
   a. Select **Administration** from the NFA console menu.
      The **Administration** page opens.
   b. Select **Define an Application**, **Application Definitions** from the menu on the left side of the page.
      The **Application Definitions** page opens.
2. Verify that **Application Mapping** is the selected value for **Rules**.
3. Click **Add Rule**.
   The **Add Application Mapping** dialog opens.
4. Select **Subnet** as the rule type at the top of the dialog. (**Subnet** is selected by default.)
   The **Add Application Mapping** dialog displays the options for a subnet application mapping rule.
5. Specify values for the following settings:
   – **Subnet**: IP address of the data source, expressed in dotted decimal format. To specify a subnet that matches all addresses, use 0.0.0.0/0 as the subnet and mask.
   – **Mask**: Mask to apply to the subnet.
   – **Protocol**: Protocol of the data that is affected by the rule, either TCP or UDP
   – **Start Port**: Beginning of the port range for collected data, expressed in Base 10 decimal format. The start port is included in the port range. The maximum port value that is allowed is 65535.
   – **End Port**: Last port in the range to use for collecting data. The end port is included in the port range.
   – **Destination Port**: Target port that collects the mapped data.
   If you specify a destination port that is already used by other rules, the traffic for the related rules are combined.
   (*Optional*) Click **Check** to run a general check to detect whether the specified port is already receiving data. The check fails if the port is receiving native data; that is, data that is not mapped by application mapping rules. If any native data is on the specified destination port, that data is redirected to the rebase port.
   – **Name**: Identifier for the rule as it is listed on the **Application Definitions** page
   The rule name also is the label for the mapped traffic in certain reports. If other rules map traffic to the same destination port that you specify for this rule, specify the name that you want to use for the combined traffic.
   – **Description**: (*Optional*) Additional descriptive text to identify the rule type and its use, which is displayed only on the **Application Definitions** page
6. Click **Save**.
   The dialog closes. The new rule is added to the **Application Mapping** rule list. If any other rules map traffic to the same port and you specified a new rule name, the other rule names are updated.
7. (*Optional*) Run reports to verify that the traffic on the designated destination port fits the rule.
8. (*Optional*) Review the effects of the new or changed application mapping rules on reports, then consider renaming the rule to label the mapped traffic more clearly in reports.

Create an NBAR2 Application Mapping Rule

Create NBAR2 (Next Generation Network-Based Application Recognition) application mapping rules to identify NBAR2 application traffic in reports. NBAR2 rules can identify traffic for individual applications, combine traffic for multiple applications, or separate NBAR2 traffic from other traffic.

If multiple rules map traffic to the same destination port, the program gives the rules the same name - the name you specified most recently. The rule name is the label for the NBAR2 traffic in reports.

You can create NBAR2 application mapping rules individually on the **Applications Definitions** page. You also can batch import NBAR2 application rules by using the command line.

Notes:

- To display NBAR2 data in reports, your routers must be configured to return IPFIX flows that include the appropriate NBAR2 fields
- Application mapping supports rules for the applications that are identified by the standard Cisco NBAR2 engine (NBAR2 engine 13), but not for applications that are identified by custom NBAR2 engines.

Follow these steps:

1. Open the **Application Definitions** page:
   a. Select **Administration** from the NFA console menu. The **Administration** page opens.
   b. Select **Define an Application, Application Definitions** from the menu on the left side of the page. The **Application Definitions** page opens.
2. Verify that **Application Mapping** is the selected value for **Rules**.
3. Click **Add Rule**. The **Add Application Mapping** dialog opens.
4. Select **NBAR2** from the list of rule types. The dialog switches to NBAR2 rule mode.
5. Specify values for the following settings:
   – **NBAR2 Application ID**: The application ID that is defined by the standard NBAR2 engine.
   Specify the application ID correctly or the rule does not function as expected. The application IDs are included in the `nbar2.csv` file:

   ```
   install_path\reporter\racmd\nbar2.csv
   ```

   – **Destination Port**: Target port that collects the mapped data
   If you specify a destination port that is already used by other rules, the traffic for the related rules will be combined. (*Optional*) Click **Check** to run a general check to detect whether the specified port is already receiving data. The check fails if the port is receiving native data--that is, data that is not mapped by application mapping rules. If any native data is on the specified destination port, that data is redirected to the rebase port.
   – **Name**: Identifier for the rule as it is listed on the **Application Definitions** page
   The rule name also is the label for the mapped traffic in certain reports. If other rules map traffic to the same destination port that you specify for this rule, specify the name that you want to use for the combined traffic.
   – **Description**: (*Optional*) Additional descriptive text to identify the rule type and its use, which is displayed only on the **Application Definitions** page.

   NOTE
   The **NBAR2 Engine ID** value is pre-populated and cannot be edited. The value is 13, the standard NBAR2 engine.
6. Click **Save**.
   The dialog closes. The new rule is added to the **Application Mapping** rule list. If any other rules map traffic to the same port and you specified a new rule name, the other rule names are updated.
7. (*Optional*) Run reports to verify that the traffic on the designated destination port fits the rule.
8. (*Optional*) Review the effects of the new or changed application mapping rules on reports, then consider renaming the rule to label the mapped traffic more clearly in reports.

Change or Delete Application Mapping Rules

Edit an Application Mapping rule when you want to change the specified source port or port range, destination port, protocol, host, ToS, subnet, mask, rule name, rule description, or rule type.

Follow these steps:

1. Open the **Application Definitions** page:
   a. Select **Administration** from the NFA console menu.
      The **Administration** page opens.
   b. Select **Define an Application, Application Definitions** from the menu on the left side of the page.
      The **Application Definitions page** opens.
2. Select an action:

Change an Application Mapping Rule

1. Click the check box next to the rule.
2. Click **Edit**.
   The **Edit Application Mapping** dialog opens.
3. Make the needed changes.
4. Click **Save**.
   If the changes are successful, the dialog closes. The list of rules is updated to reflect your changes. If multiple rules map traffic to the same destination port and you change one of the rule names, the other names are updated to match. This name is used to label the combined traffic in reports.

Delete an Application Mapping Rule

You can delete one or more rules.

1. Click the check box next to the rule.
2. Click **Delete**.

Import Application Mapping Rules

You can create or update application mapping rules by importing a `.csv` file that is formatted for the rule type. You can perform several types of application mapping rule imports:

Importing rules has the following effects and characteristics:

- After the import, the new or updated rules are shown in the **Application Definitions** page list.
- Reports show mapped traffic with labels that match the new rule names.
- Perform this type of import operation locally. You cannot perform the import remotely.
- You can work on the application mapping rules in a Microsoft Excel spreadsheet, then export the rules to `.csv` format.
- If you import the default NBAR2 rule set, each NBAR2 application is mapped to a separate port by default. The default port range is port 65001 and above.

NOTE

More information:

- Troubleshoot Importing Application Mapping Rules

Import the Default NBAR2 Application Mapping Rules

You can add a complete set of default NBAR2 application mapping rules in a batch import operation. You perform this task on the command line by using the `nbar2.csv` file that comes with the product.

NOTE
The NBAR2 application mapping rules that you import represent the set of NBAR2 application definitions compatible with NBAR2 Engine 13.

Follow these steps:

1. Log in to the NFA console server or stand-alone server as a user who is a member of the Administrators group.
2. Open a command prompt.
3. Navigate to the directory that contains the `nbar2.csv` file:

   ```
   cd install_path\reporter\racmd
   ```

   where:
   - *install_path* is the product installation path. The default path is `C:\CA\NFA`.
   - `racmd` is the directory that contains the `nbar2.csv` file. The file is written to this directory when you install the product.
4. Enter the following command:

   ```
   racmd -import nbar2.csv
   ```

   where:
   - `nbar2.csv` is the name of the application mapping rule file that you want to import. The command string is based on the expectation that the `racmd` command and `.csv` file are in their default location. If you have moved the `.csv` file, include the fully qualified path (the path and file name).
     If any errors occur during the import, error messages are shown. If no message is returned, the import succeeded with no problems.
5. (*Optional*) Verify that the rules are listed on the **Applications Definitions** page:
   a. Select **Administration** from the NFA console menu.
      The **Administration** page opens.
   b. Select **Define an Application, Application Definitions** from the menu on the left side of the page.
      The **Application Definitions** page opens. The new rules are shown in the **Application Definitions** page list.
6. (*Optional*) Verify that the mapped NBAR2 application traffic is labeled appropriately in the following locations:
   - **Enterprise Overview** page: **Top Protocols** report
   - **Interface** page: All reports that show protocol data
   - **Custom Reporting** page and **Analysis** page: **Protocol Index**, which you use to select protocols to filter a new or edited report
7. (*Optional*) Combine traffic reporting for selected applications.
   a. Identify a set of applications that you want to see reported as combined traffic.
   b. Edit the corresponding application rules to send the data to a single destination port.
      By default, each NBAR2 application is mapped to a separate port. The default ports are above 65000.
   c. Use an appropriate name for the rules: Use a name that reflects the type of applications that are included. Reports use this name to label the combined traffic.
      If you update a rule name, all rules that use that destination port are renamed to match the new definition. The rule name and label is set when you edit the final rule name.

Import Custom Application Mapping Rules

You can create application mapping rules of a single type by importing a properly formatted `.csv` file. Example `.csv` files are provided, which show the required fields to include in the import file for each rule type.

Follow these steps:

1. Log in to the NFA console server or stand-alone server as a user who is a member of the Administrators group.
2. Open a command prompt.
3. Open the example import file for your rule type:

- All (ToS) rule - `tos.csv`
- Host rule with a specified protocol - `server-protocol.csv`
- Host rule without a specified protocol - `server.csv`
- Subnet rule - `subnet.csv`
- NBAR2 rule - `nbar2.csv`

4. Follow the format in the example file for your rule type:
   - The first row in the file is the column name row, which identifies the fields. Leave the first row exactly as it is shown in the example file. Do not change the spelling or order of the column name row.
   - Add a row below the top row for each rule that you want to import.
   - Enter values for each required field.
     Separate the field values with commas. Do not include a comma inside any value string. A comma signals the import utility to go the next field.
     All fields are required except for the desc (Description) field. To specify a blank desc value, enter only a comma (with no space).
   - The import file columns correspond to the following columns on the **Application Definitions** page:
     - name = **Name**: Rule name
     - desc = (*Optional*) **Description**
     - protocolName - **Protocol**
       To specify all protocols, enter the value -1.
     - tos = **ToS** (Type of Service)
       To specify all ToS, enter the value -1.
     - ip = **IP/Subnet**: Host IP address
     - mask = IP/Subnet: Subnet addition to the host IP address
     - newPort = **Destination Port**
     - beginPort = **Start Port**: Starting port in a port range or port number for a server-protocol rule
     - endPort = **End Port**: Ending port in a port range
     - applicationID = **NBAR2 Application ID**

     Some fields apply only to particular rule types, as shown in the table at the end of these steps.
5. Go to the directory that contains the `.csv` file. The following command shows the default location:

   ```
   cd install_path\reporter\racmd
   ```

   where:
   - *install_path* is the product installation path. The product installation path is `C:\CA\NFA` by default.
   - `racmd` is the directory that contains the `.csv` import file. The file is written to this directory when you install the product.
6. Perform the import by entering the following command:

   ```
   racmd -import nbar2.csv
   ```

   where:
   - `nbar2.csv` is the name of the application mapping rule file that you want to import. The command string is based on the expectation that the racmd command and .csv file are in their default location. If you have moved the .csv file, include the fully qualified path (the path and file name).
     If any errors occur during the import, error messages are shown. If no message is returned, the import succeeded with no problems.
7. (*Optional*) Verify that the rules are listed on the **Applications Definitions** page:
   a. Select **Administration** from the NFA console menu.

The **Administration** page opens.

b. Select **Application Definitions** in the **Define an Application** menu.
The **Application Definitions** page opens.

8. (*Optional*) Verify that the application traffic is labeled appropriately in the following locations:
   - **Enterprise Overview** page: **Top Protocols** report
   - **Interface** page: All reports that show protocol data
   - **Custom Reporting** page and **Analysis** page: **Protocol Index**, which you use to select protocols to filter a new or edited report
9. (*Optional*) Combine or separate traffic for selected applications.

The following table lists the fields that apply to each rule type. List the fields in the order that is shown in the import file example, not the order shown in the table.

Rule Type	name	desc	protocolName	tos	ip	mask	newPort	beginPort	endPort	applicationID
All - tos	Y	Y		Y			Y			
Host - server	Y	Y			Y		Y			
Host - server-protocol	Y	Y	Y	Y	Y		Y	Y		
NBAR2 - nbar2	Y	Y					Y			Y
Subnet - subnet	Y	Y	Y		Y	Y	Y	Y	Y	

Import Application Mapping Rule Updates

You can create application mapping rules of a single type by importing a properly formatted `.csv` file. Example `.csv` files are provided, which show the fields to include in each type of import file.

Follow these steps:

1. Log in to the NFA console server or stand-alone server as a user who is a member of the Administrators group.
2. Get the rule ID for the existing rules that you want to update.
   a. Open a command prompt and go to the directory that contains the `.csv` file. The following command shows the default location:

      ```
      cd install_path\reporter\racmd
      ```

      where:
      - *`install_path`* is the product installation path. The product installation path is `C:\CA\NFA` by default.
      - `racmd` is the directory that contains the `.csv` import file. The file is written to this directory when you install the product.

   b. Export the rule definitions by entering the following command:

      ```
      racmd -export csv
      ```

      The export file is named `getapplicationmapping_<timestamp>.csv`. The file is located in the current directory.

The command returns the status message: `Creating csv file .` When the operation is complete, the command prompt reappears.

NOTE
Rule IDs are specific to the current stand-alone or NFA console system.

3. Open the export file in a spreadsheet or text editor.
   The export file contains a line for each current application mapping rule, which begins with the rule ID. The line also includes extra information that you can ignore.
4. Locate and make a note of the ID for each rule that you want to edit.
5. Prepare the import file:
   a. Open a copy of the example import file for your rule type:
      - All (ToS) rule - `tos.csv`
      - Host rule with a specified protocol - `server-protocol.csv`
      - Host rule without a specified protocol - `server.csv`
      - Subnet rule - `subnet.csv`
      - NBAR2 rule - `nbar2.csv`
   b. Add the appID column and rule values, as shown in the following examples:
      Example: First two lines in an import file to add NBAR2 rules

      ```
      name,desc,newPort,applicationid

      youtube,Youtube video streaming,65035,82
      ```

      where:
      - name = Rule name
      - desc = Rule description
      - newPort = Destination port
      - applicationID = NBAR2 application ID

      Example: First two lines in an import file to update NBAR2 rules

      ```
      appID,name,desc,newPort,applicationid

      35,YouTube,Youtube video streaming,65035,82
      ```

      where:
      - applicationID = Rule ID

      NOTE
      If you update NBAR2 application mapping rules, do not change the NBAR2 application ID value. If you change this value, the rule will not function as expected.
   c. Other than adding the appID column and values, follow the import mapping guidelines for custom Application Mapping Rules.
   d. Save the import file.
      We recommend that you save the import file to the same directory that contains the import command: *install_path*`\reporter\racmd .`
6. Go to the directory that contains the .csv file:

   ```
   cd install_path\reporter\racmd
   ```

   where:

- *install_path* is the product installation path. The product installation path is `C:\CA\NFA` by default.
- `racmd` is the directory that contains the `.csv` import file. The file is written to this directory when you install the product.

7. Enter the following command:

```
racmd -import nbar2_updated.csv
```

where:

- `nbar2_updated.csv` is the name of the update file that you just created. If you have moved the `.csv` file, include the fully qualified path (the path and file name).

If any errors occur during the import, error messages are shown. If no message is returned, the import succeeded with no problems.

8. (*Optional*) Verify that the updated rules are listed correctly on the **Applications Definitions** page:
   a. Select **Administration** from the NFA console menu.
      The **Administration** page opens.
   b. Select **Define an Application, Application Definitions** from the menu on the left side of the page.
      The **Application Definitions** page opens.
9. (*Optional*) Verify that the application traffic is labeled correctly in the following locations:
   - **Enterprise Overview** page: **Top Protocols** report
   - **Interface** page: All reports that show protocol data
   - **Custom Reporting** page and **Analysis** page: **Protocol Index**, which you use to select protocols to filter a new or edited report

Create, Change, or Delete Reserved Seating Rules

You can create Reserved Seating rules to help ensure that reports include the port and protocol combinations that interest you, regardless of traffic volume or rates. The rules create ‘reserved seats’ for the ports that are used by those protocols so the data is sure to be included in reports.

For example, during an application rollout you want to watch the traffic for a particular application, but the **Top N Protocols** reports for interfaces do not show the traffic for the application. The protocol that the application uses is not included in the Top N Protocol group; the group of protocols with the highest traffic volume or utilization rate. You create a Reserved Seating rule to collect data for the specific protocol and port that the application uses. The protocol now is included in the **Top N Protocols** reports.

NOTE

More information:

- Data Collection

Follow these steps:

1. Open the **Application Definitions** page:
   a. Select **Administration** from the NFA console menu.
      The **Administration** page opens.
   b. Select **Define an Application**, **Application Definitions** from the menu on the left side of the page.
      The **Application Definitions** page opens.
2. Select **Reserved Seating** from the **Rules** list.
   The **Application Definitions** page switches to **Reserved Seating** mode and displays a list of the current **Reserved Seating** rules.
3. Select one of the following actions:

Add a Rule

1. Click **Add Rule**.
2. Specify ports as follows:
   - **Protocol**: Protocol of the data that is affected by the rule, either TCP or UDP
   - **Port**: Target port for the Reserved Seating rule. Enter the port number in the **Port** box, a value from 0 through 65535, expressed in Base 10 decimal format. If you do not enter a value, port 0 is assigned. The port and protocol combination must be unique; that is, it cannot match any other Reserved Seating rule.
     Data of the specified protocol type is reported for this port regardless of traffic rate or volume.
   - **Description**: (*Optional*) Identifying text for the Reserved Seating rule. The description appears in the list of **Reserved Seating** rules on the **Application Definitions** page.
3. Click **Save**.
   If you entered a valid port and protocol combination and you have not yet reached the maximum number of rules, the dialog closes. The new rule appears in the list of **Reserved Seating** rules.
4. Repeat this process for each Reserved Seating rule you want to add.
   You can specify a maximum of 50 Reserved Seating rules.

Change a Rule

1. Click the check box next to the rule you want to edit, then click **Edit**.
   The **Edit Reserved Seating** dialog opens.
2. Make the needed changes in the **Protocol**, **Port**, and **Description** values.
3. Click **Save**.
   If the changes are successful, the **Edit Reserved Seating** dialog closes.

Delete a Rule

1. Click the check box next to one or more rules you want to delete. To select the check boxes for all the rules, click the check box in the heading row.
2. Click **Delete**.
   A confirmation message opens.
3. Click **Yes**.
   The list of **Reserved Seating** rules is updated to reflect your deletions.

Work with Port Priorities

By default, DX NetOps (CA NFA) defines the server port and protocol as the lower number in the flow record.

Source port: 80

Destination port: 8000

In this case, CA NFA determines that the lower port is 80, and therefore http.

When the server port (tcp / udp) is a high number, the port priority can be used.

For example:

Server port: 8888

Client port: 6000

By default, CA NFA would use the lower port (6000) as the server port. The Port Priority functionality tells it to use 8888 as the server port when it is present in the data.

Create Port Priority Rules

Create Port Priority rules to help ensure that the correct protocols are identified for each range of ports.

Follow these steps:

1. Open the **Application Definitions** page:
   a. Select **Administration** from the NFA console menu. The **Administration** page opens.
   b. Select **Define and Application, Application Definitions** from the menu on the left side of the page. The **Application Definitions** page opens.
2. Select **Port Priority** from the **Rules** list.
   The **Application Definitions** page switches to **Port Priority** mode and displays a list of the current Port Priority rules.
3. Click **Add Rule**.
   The **Add Port Priority** dialog opens.
4. Specify ports as follows:
   – **Protocol**: Protocol of the data that is affected by the rule, either **TCP** or **UDP**
   – **Start Port**: Target starting port for the Port Priority rule. Enter the port number in the **Start Port** box, a value from 0 through 65535, expressed in Base 10 decimal format. If you do not enter a value, port 0 is assigned.
   – **End Port**: Target ending port for the Port Priority rule. Enter the port number in the **End Port** box, a value from 0 through 65535, expressed in Base 10 decimal format. If you do not enter a value, port 0 is assigned.
   Note: The start port, end port, and protocol combination must be unique; that is, it cannot match any other **Port Priority** rule.
   – **Description**: (*Optional*) Identifying text for the Port Priority rule. The description appears in the list of **Port Priority** rules on the **Application Definitions** page.
5. Click **Save**.
   If you entered a valid start port, end port, and protocol combination and you have not yet reached the maximum number of rules, the dialog closes. The new rule appears in the list of Port Priority rules.
6. Repeat this process for each Port Priority rule you want to add.
   You can specify a maximum of 50 rules.

Maintenance and Data Collection

Maintain DX NetOps to ensure that all features are performing well. You will want to perform some maintenance tasks regularly. For example, when a user sets up a trap, the Administrator must deploy that trap.

> **NOTE**
> To complete the tasks that are described in this section, the role for your DX NetOps user account must be either Administrator or Power User.

View Flow Statistics

View the **Flow Statistics** page at any time to view the flow rates on a per harvester basis.

Flow Rate by Harvester

An interactive time series graph at the top of the screen that contains flow rates for each harvester on a single graph. This view defaults to showing data over the last 24 hours, with selectable time periods. You can hover over a point on the graph to see the Harvester name, data point value, date, and time.

The interactive legend (to the right of the graph) allows you to control which Harvesters are included in the graph. Click the checkmark next to the Harvester name to add or remove it from the graph. Click a Harvester name to display only that Harvester in the graph.

The graph types are:

- **area**
- **bar**
- **line**
- **scatter**

Within each graph type, you can define the form (not all forms are available for all graph types):

- **stack**
- **stream**
- **percent**
- **value**

Within each form, you can define the line smoothing method (not all methods are available for all forms):

- **cardinal**
- **linear**
- **step**

The range slider (under the graph) allows you to zoom in to the data that is shown in the graph.

Time Period

The time period is defined as the time specified by the **From** and **To** date and time. Selecting the time period allows you to compress or expand the amount of data shown in the graph. Different time periods use different data intervals:

Time Period	Data interval
24 hours or less	5 Minute Data
24 hours to 1 week	15 Minute Data
1 to 2 weeks	30 Minute Data
Greater than 2 weeks	60 Minute Data

Only one month of data (rolling month) is kept in the database.

Update

Clicking **Update** reloads the graph with the most current data based on the time period. Because it updates the data, it resets the Harvester selections and closes any tables opened from the **Flow Statistics by Harvester** table.

Flow Statistics by Harvester

A table view that contains individual harvesters with all of the processing metrics above (dropped flows, dropped packets, map failures, etc.). This view defaults to showing data over the defined time period.

- - **IP**
    The IP address of the Harvester.
  - **Description**
    The name of the Harvester (the name used in the interactive legend).
  - **Routers**
    The number of routers on the Harvester, in the defined time period.
  - **Interfaces**
    The number of interfaces on the Harvester, in the defined time period.
  - **Min Flow Rate**
    The minimum flow rate for the Harvester, in the defined time period.
  - **Max Flow Rate**

The maximum flow rate for the Harvester, in the defined time period.

- **Avg Flow Rate**
  The average flow rate for the Harvester, in the defined time period.
- **95th Pct Flow Rate**
  The 95th percentile flow rate for the Harvester, in the defined time period.
- **Discarded Packets**
  The number of packets that were discarded on the Harvester, in the defined time period.
  Discarded packets are packets that were successfully received, but for some reason are not valid for further processing. The next three columns reflect reasons that packets might have been discarded.
  - **No Flows Found**
    The number of NetFlow FlowSets that had either no content or invalid content for the Harvester, in the defined time period.
  - **Header Failures**
    The number of NetFlow header parse failures for the Harvester, in the defined time period.
  - **Router Reboots**The number of times the Harvester detected that a router has been rebooted, in the defined time period.This could indicate device or NetFlow configuration issues, or normal behavior if planned maintenance is being performed.
- **Discarded Flows**
  The number of flows that were discarded on the Harvester, in the defined time period.
  Discarded flows are flows that were successfully received and parsed, but were disqualified from further processing based on the state of the interfaces reported in the flow. The next column reflects reasons that flows might have been discarded.
  - **Map Failures**
    The number of interfaces that are either new to the Harvester or are disabled on the Harvester, in the defined time period.
    For new interfaces, a poll is scheduled to discover the type and speed of the interface, and to potentially remap any pre-existing interfaces on the same router.

Follow these steps:

1. Display the **Flow Statistics** page:
   a. Select **Administration** from the NFA console menu.
      The **System Status** page opens.
   b. Select **Health, Flow Statistics** from the menu on the left side of the page.The **Flow Statistics** page opens.
2. The **Flow Rate by Harvester** chart defaults to **area**, **stack**, **cardinal**, with a time period of **Daily**. Select the graph, form, and smoothing method.
3. Specify the **Time Period**.
4. Click **Update**. The graph and table update with the latest information available for the requested time period.
5. Click a row in the **Flow Statistics by Harvester** table to display the **Router Information** table for that Harvester. Click a row in the **Router Information** table to display the **Interface Information** table for that router.

View System Status

View the **System Status** page at any time to see the overall status of DX NetOps components.

Follow these steps:

1. Display the **System Status** page:
   - Select **Administration** from the NFA console menu.
     or
   - Select **Health, System Status** from the menu on the left side of the **Administration** page.

The **System Status** page opens and shows a quick overview of the status of DX NetOps components. Status symbols identify any components that have generated warnings. The number of warnings appears in parentheses after the component label.

2. Click any component that has a warning symbol to see details.
   A table of warnings appears.
3. Review any warnings that you find for each component.

Configure Application Settings

You can configure a wide range of settings on the **Application Settings** page.

Follow these steps:

1. Select **Administration** from the NFA console menu.
   The **System Status** page opens.
2. Select **System, Application Settings** from the menu on the left side of the page.
   The **Application Settings** page opens.
3. (*Optional*) Change any of the following settings as needed, then click **Save**.
   - **Interface Data Absence Limit**
     Specifies how long the program waits before it flags an interface as inactive, starting from the **Last Flow** value on the **Available Interfaces** page. When the limit is reached, the interface status changes in the following locations:
     - **Interface Index**: Active column value changes to **No**.
     - **Active Interfaces** page: **Traffic Status** value changes to **Inactive/Red**.

     Default: 4 Hours
   - **TCP Rebase Port**
     Specifies the target port for TCP traffic that is redirected by an application mapping rule. TCP traffic that you do not want to go to a target port goes to the TCP Rebase Port instead. Other settings that affect application mapping behavior are **UDP Rebase Port**, **ToS Mask**, and **Preserve ToS Map Proto**.
     Default: 9000
   - **ToS Mask**
     Specifies the number of bits that application mapping rules use for matching ToS values. The default value of 255 sets the program to look for matches throughout all ToS values. Other settings that affect application mapping behavior are **TCP Rebase Port**, **UDP Rebase Port**, and **Preserve ToS Map Proto**.
     Default: 255
   - **UDP Rebase Port**
     Specifies the target port for UDP traffic that is redirected by an application mapping rule. UDP traffic that you do not want to go to a target port goes to the UDP Rebase Port instead. Other settings that affect application mapping behavior are **TCP Rebase Port**, **ToS Mask**, and **Preserve ToS Map Proto**.
     Default: 8000
   - **Auto-Enable Interfaces**
     Specifies whether newly discovered interfaces are enabled automatically (True) or are disabled (False). If you want to control which interfaces are reported and consume licenses, set the value to False. In this case you enable the interfaces manually on the **Available Interfaces** page. This setting affects the **Enabled** status for new interfaces. Interfaces that have already been discovered are not affected by changes to this setting.
     Default: True
   - **DNS Domains**
     Removes the specified suffixes from host names in NFA console views and reports. If you include `.my_company.com`, for example, this suffix is not shown in the host names that appear in any views or reports. To specify multiple entries, separate the entries with commas and without intervening spaces.
     Default: <no default>
   - **Show Trendline Zeroes**
     Specifies whether trendline reports show data with interconnecting lines.

If the value is True, trend lines connect data points in reports such as the **Multi Trend Summary** and **Stacked Trend** on the **Interface** pages. Fill pattern is shown beneath the lines.
If the value is False, the reports show only the data points that are real. The trend line ends at the last data point and starts at the next data point. The reports show gaps wherever data points are missing. The reports do not have boundary lines for fill pattern, so the fill pattern is missing.
Default: False

– **From Address**
Specifies the email address of the DX NetOps Administrator, which is used as the **From** value when reports are emailed. If this setting is not configured properly, users cannot send scheduled or on-the-fly reports. These functions also require a properly configured **SMTP Server** value.
Default: <no default>

– **SMTP Server**
Specifies the IP address of the SMTP mail server that is used for emailing reports. If this setting is not configured properly, users cannot send scheduled or on-the-fly reports. These functions also require a properly configured **From Address** value.
Default: <no default>

– **Licensed Devices**
Records the total number of licenses that you purchased from CA. This value is used to calculate the percentage of licenses in use, which is displayed on the **About** page. The **License Utilization** percentage is accurate only if the **Licensed Devices** value is accurate.
Default: 50

– **Preserve ToS Map Proto**
Specifies whether protocol traffic for ToS-based application-mapped data is combined (N) or is shown as separate data streams that are labeled with the original protocol designators (Y).
For example, suppose the value is Y and you map TCP, UDP, and some other IP protocol traffic to one port. To continue the example, suppose you drill in to a link in the **Enterprise Overview: Top Host** view for a host that has the mapped data. In this case the **Stacked Protocol Trend** and **Protocol Trend** views show and label the protocol traffic separately, whether the protocol traffic is for TCP, UDP, or some other IP protocol.
  - **Stacked Protocol Trend** and **Protocol Trend** views show protocol traffic that meets the following conditions: (1) The traffic passes the minimum threshold and (2) The protocol volume is high enough to place it in the Top N group.

If the **Preserve ToS Map Proto** value is N and the **Stacked Protocol Trend** views show related protocol traffic, all of the protocols for the mapped traffic are combined in a single traffic stream that has a TCP label.
Other settings that affect application mapping are **TCP Rebase Port**, **UDP Rebase Port**, and **ToS Mask**.
Default: Y

– **Pump Broadcast/Multicast**
Specifies whether interface views and reports include (True) or hide (False) broadcast/multicast traffic.
Default: True

– **Reporter IP**
Specifies the IP address of the NFA console.
Default:
  - Stand-alone deployment: Loopback IP address of the stand-alone server
  - Distributed 2-tier deployment: Loopback IP address of the NFA console

– **Report Service Polling Delay**
Specifies the number of seconds between checks to see if reports have finished running. The status **Complete** is displayed when both of the following conditions are met:
  - The Report Service check confirms that the report is finished.
  - You refresh the report list on the **Custom Reporting** page, **Analysis** page, or **Flow Forensics** page.

Default: 15

– **Router Domains**

Removes the specified suffixes from router names that appear in the NFA console views and reports. To specify multiple entries, separate the entries with commas and without intervening spaces.
Default: <no default>

- **Show Aggregations**
  Specifies whether to include interface aggregations in **Enterprise Overview** page views. If the value is True, interface aggregations are included in the views. To be included in **Enterprise Overview** page views, the aggregations must have enough traffic to pass the minimum thresholds and to rank in the Top N group.
  Default: False
- **Show Device Name**
  Specifies whether the interface name format starts with the device name (True) or omits it (False). This setting affects the interface names that appear in views and reports, such as the **Enterprise Overview** views, **Interface** page reports, and **Custom Report Interface Summaries**.
  Default: True
- **Display Notes Field**
  Displays (True) or hides (False) the **Notes** icon for interface rows on the **Active Interfaces** page. If the **Notes** icon is visible, you can click it to add, edit, or view additional information about an interface.
  Default: False
- **Trap Destination**
  IP address or DNS name of the target server for sending the traps that are shown as events in the Performance Center console:
  - (CA PC) **Events Display** page
  - (CA NPC) **Event List** page

  Traps can be displayed as events only when this setting is configured correctly. Set the **Trap Destination** value to match the IP address of one of the following servers:
  - (CA PC) NFA console or stand-alone server that is registered as a data source for Performance Center
  - (CA NPC) Event Manager server

  Default: IP address of the NFA console (distributed deployment) or stand-alone server (stand-alone deployment)

How to Monitor the Components

Watchdog Services let you monitor DX NetOps components. The Watchdog Services poll each server in your DX NetOps configuration once every two hours to determine the status of all components. You can establish thresholds, an email address for receiving messages, and other settings for the Watchdog Services to ensure that you are notified of issues with the components as soon as possible.

Edit Watchdog Service Settings

Edit the Watchdog settings to change configuration values such as thresholds, trap settings, polling settings, notification address, and community strings.

Follow these steps:

1. Display the **Watchdog Settings** page:
   a. Select **Administration** from the NFA console menu.
      The **Administration** page opens.
   b. Select **Health, Watchdog Settings** from the menu on the left side of the page.
      The **Watchdog Settings** page opens and displays the current settings.
2. Edit the **Watchdog Service** settings:
   - **Community String**
     SNMP string that the Watchdog Services use to verify the identity of components in a distributed deployment. The community string is used for gathering information from Harvesters. Use the same community name throughout the DX NetOps deployment:

- **Watchdog Settings** page
- SNMP service on each Windows server
- `snmpd.conf` file on each Linux server

Default: public

– **CPU Threshold**
Threshold for CPU utilization. You are notified by email when the CPU threshold on a server is exceeded on any server and an SNMP trap notification is generated, provided that the address and string are set.
Default: 80 percent CPU utilization

– **Disk Threshold**
Threshold for disk utilization. If the disk threshold on a server is exceeded, you are notified by email and an SNMP trap notification is generated, provided that the address and string are set.
Default: 80 percent disk utilization

– **Email Address**
Destination email address to use for email notifications when thresholds are exceeded. To notify multiple recipients, separate the addresses with commas. The **Email Address** setting has no default value.
Default: (none)

– **Memory Threshold**
Threshold for memory utilization. You are notified by email when the memory threshold on a server is exceeded, provided that the address and string are set.
Default: 80 percent memory utilization

– **SNMP Retries**
Number of times the program attempts to poll an SNMP device. A high number of SNMP Retries can affect performance, depending on your network configuration.
Default: 2

– **SNMP Timeout**
Number of seconds before an SNMP poll times out.
Default: 5

– **System Check Interval**
Number of minutes between Watchdog system checks.
Default: 60

– **Trap Community String**
SNMP string to use for sending traps to a third-party trap receiver. Use one of the community names that the trap receiver is configured to accept.
Default: public

– **Trap Destination**
IP address of the server that receives SNMP traps from the Watchdog Services. The traps are generated when thresholds for DX NetOps component performance are violated.
Default: (none)

3. Click **Save** when you finish editing the settings.

Work with Traps

Create traps to notify network management when certain events occur. Administrators can create traps and must deploy any traps that users create.

DX NetOps traps can be integrated with external fault management packages.

NOTE
More information:

Create, Change, or Delete Traps

You can create traps to notify network management when certain events occur.

TIP

You can display custom ToS and Protocol names in trap alerts with interface groups using the `trapsHaveUniqueDomain` parameter. This should only be used in a single-domain environment.

1. On the NFA console, in the `reporter.parameter_descriptions` table, set the `trapsHaveUniqueDomain` parameter to `True` .
2. Restart the NFA console services.

This parameter only affects new traps; it does not have any effect on existing traps.

Follow these steps:

1. Display the **Add a Trap Configuration** page:
   a. Select **Administration** from the NFA console menu.
   The **Administration** page opens.
   b. Select **Alerts, Traps** from the menu on the left side of the page.
   The **Trap Configuration** page opens and displays a list of the current traps.
2. Select an action:

Create a Trap

1. Click **Add**.
   The **Add a Trap Definition** page opens.
2. Define the trap description and the interfaces that the trap monitors:
   – **Description**: Identification for the new trap. Include information about the protocol and threshold, such as "Microsoft SQL Monitor utilization above 65%".
   – **Select Interface** or **Select Interface Group**: Link to an index of interfaces or interface groups, which you use to select which interfaces that the trap monitors.

   NOTE

   - If the interface speed is unknown or 0, any attempt to set a trap is logged. The trap is not sent to the Harvester and therefore no alerts on this interface are generated. This interface is not included in any statistics calculations.
   - If the associated interface or interface group is deleted after you deploy the trap, an "Unknown interface group error" occurs. To eliminate this error, associate the trap with a different interface or interface group.
   - If using interface groups, custom ToS or Protocol names do not apply unless you have set the `trapsHaveUniqueDomain` parameter.
3. (*Optional*) Create a Utilization-based trap by providing the following information:
   – **Thresholds: Utilization**: Setting to make the trap threshold type utilization. Use the context-sensitive fields that are added for this trap type.

- **Utilization: In** field: Percentage of utilization that acts as the inbound threshold value
- **Utilization: In: minutes** list: Number of minutes of inbound threshold violation that generates a trap
- **Utilization: Out** field: Percentage of utilization that acts as the outbound threshold value
- **Utilization: Out: minutes** list: Number of minutes of outbound threshold violation that generates a trap

– **Protocol**: Protocol for the monitored traffic. Select the protocol in the **Protocol Index** dialog. The trap applies to the selected interfaces and protocol.
– **ToS**: (*Optional*) Type of Service (ToS) of the monitored traffic The trap applies to the selected interfaces, protocol, and ToS. To monitor all ToS traffic, select **All**.
– **Time Filter**: (*Optional*) Time period for monitoring if you choose to limit monitoring in this way. Select one of the available time filters from the list. The **Time Filter** list includes all the time filters that the Administrator has created.

NOTE
A time filter applied to a trap uses the time zone of the Harvester.

4. (*Optional*) Create a Rate-based trap by providing the following information:
For example, suppose that you create a Rate-based trap and establish thresholds for both the inbound and outbound traffic. The trap is triggered when the threshold is met during the defined period.
   – **Thresholds: Rate**: Setting to base the trap threshold on data transfer rate. Use the context-sensitive fields that are added for this trap type.
     - **Rate: In**: Inbound data transfer speed that acts as a threshold value. Define the number of bits, kilobits, megabits, or gigabits per second, according to the unit of measurement selected.
     - **Rate: Units of Measure**: Unit of measurement for data transfer. Select bits per second (bps), kilobits per second (kbps), megabits per second (Mbps), or gigabits per second (Gbps).
     - **Rate: In: minutes**: Number of minutes of inbound threshold violation that generates a trap
     - **Rate: Out**: Threshold for outbound data transfer. Enter the number of bits, kilobits, megabits, or gigabits per second, according to the unit of measure for the inbound threshold.
     - **Rate: Out: minutes**: Number of minutes of outbound threshold violation that generates a trap
   – **Protocol**: Protocol for the monitored traffic. Select the protocol in the **Protocol Index** dialog. The trap applies to the selected interfaces and protocol.
   – **ToS**: (*Optional*) Type of Service (ToS) of the monitored traffic The trap applies to the selected interfaces, protocol, and ToS. To monitor all ToS traffic, select **All**.
   – **Time Filter**: (*Optional*) Time period for monitoring if you choose to limit monitoring in this way. Select one of the available time filters from the list. The **Time Filter** list includes all the time filters that the Administrator has created.

   NOTE
   A time filter applied to a trap uses the time zone of the Harvester.
5. (*Optional*) Create a Volume-based trap by providing the following information:
   – **Thresholds: Volume**: Setting to base the trap threshold on data volume. Use the context-sensitive fields that are added for this trap type.
     - **Volume: In**: Inbound data volume that acts as a threshold. Define the number of bytes, kilobytes, megabytes, gigabytes, or terabytes, according to the unit of measurement selected.
     - **Volume: Units of Measure**: Unit of measurement for data transfer. Select bytes, kilobytes (kB), megabytes (MB), gigabytes (GB), or terabytes (TB).
     - **Volume: Out**: Inbound data volume that acts as a threshold. Enter the number of bytes, kilobytes, megabytes, gigabytes, or terabytes, according to the unit of measure for the inbound threshold.
     - **Volume: Out: minutes**: Number of minutes of outbound threshold violation that generates a trap.
   – **Protocol**: Protocol for the monitored traffic. Select the protocol in the **Protocol Index** dialog. The trap applies to the selected interfaces and protocol.
   – **ToS**: (*Optional*) Type of Service (ToS) of the monitored traffic The trap applies to the selected interfaces, protocol, and ToS. To monitor all ToS traffic, select **All**.
   – **Time Filter**: (*Optional*) Time period for monitoring if you choose to limit monitoring in this way. Select one of the available time filters from the list. The **Time Filter** list includes all the time filters that the Administrator has created.

NOTE
A time filter applied to a trap uses the time zone of the Harvester.

6. (*Optional*) Create a Flow-based trap by providing the following information:
   – **Thresholds: Flows**: Setting to base the trap threshold on the number of flows. Use the context-sensitive fields that are added for this trap type.
     • **Flows**: Total: Number of inbound flows that acts as a threshold value. Define the number by flows per minute, thousands of flows per minute, or millions of flows per minute, according to the unit of measurement selected.
     • **Flows: Units of Measure**: Unit of measurement for data transfer. Select flows/minute, thousands of flows/minute, or millions of flows/minute.
     • **Flows: minutes**: Number of minutes of threshold violation that generates a trap
   – **Time Filter**: (*Optional*) Time period for monitoring if you choose to limit monitoring in this way. Select one of the available time filters from the list. The **Time Filter** list includes all the time filters that the Administrator has created.

     NOTE
     A time filter applied to a trap uses the time zone of the Harvester.
7. Click **Submit** when the selections are complete.
   The trap is deployed.

   NOTE
   Traps are sent to the destination that is defined in the application settings.

Change a Trap Definition

1. Click the trap description in the **Trap Configuration** page trap list.
   The **Trap Configuration** page changes to edit mode.
2. Make updates in the **Trap Configuration** page.
3. Click **Submit**. The trap is updated.

Delete a Trap Definition

1. Click the trap description in the **Trap Configuration** page trap list.
   The **Trap Configuration** page changes to edit mode.
2. Click the **Delete** button that appears in edit mode.
3. Verify the deletion when prompted.

Configure Trap Destinations

Configure destinations for the traps that you and operators create.

- Set the trap destination in the **Application Settings** page to display traps as events in the Performance Center Console.
- If you also want to generate Watchdog traps, configure the trap destination on the **Watchdog Settings** page in the NFA console.

Follow these steps to enable traps to be displayed in the Performance Center Console:

1. Display the **Application Settings** page:
   a. Select **Administration** from the NFA console menu.
      The **Administration** page opens.
   b. Select **System, Application Settings** from the menu on the left side of the page.
      The **Application Settings** opens.

2. Locate the **Trap Destination** field.
3. Verify that the **Trap Destination** field value matches the appropriate IP address:
   – (CA PC) The NFA console or stand-alone server that is registered as a data source
   – (CA NPC) The Event Manager server

   To check this value in CA Performance Center, make sure the **Trap Destination** IP address matches the **Host Name** value in CA Performance Center. If the host IP address is not included in the data source name, check the **Host Name** value by completing the following steps:
   a. Select **Admin**, **Data Sources** in the Performance Center Console.
      The **Manage Data Sources** page opens.
   b. Right-click the DX NetOps data source and select **Edit** from the menu.
      The **Edit Data Source** dialog opens.
   c. Check the **Host Name** value.

Follow these steps to set the destination for Watchdog traps:

1. Display the **Watchdog Settings** page:
   a. Select **Administration** from the NFA console menu.
      The **Administration** page opens.
   b. Select **Health, Watchdog Settings** from the menu on the left side of the page.
      The **Watchdog Settings** page opens and displays the current settings.
2. Verify that the Watchdog settings are correct, including the **Trap Destination**:
   Enter the IP address of the server that hosts the trap receiver. The Watchdog Services send SNMP to the **Trap Destination** address.
3. Click **Save** when you finish editing the settings.
   The settings are saved. Messages are sent out for any new traps that are generated.

Optional Tasks:

- Verify that the events are displayed on the **Events Display** page (CA PC) or the **Event List** page (CA NPC). If the page does not show the events as expected, verify that the following conditions are met:
  – The logs show that events have been generated and forwarded to the Event Manager.
  – The Event Manager host name is resolvable by the DNS server for DX NetOps.
  – The **Trap Destination** value on the **Application Settings** page in the NFA console matches the IP address of the NFA console or stand-alone server that is registered as a data source.
- Configure traps for external programs.

Set Up Traps for External Fault Management Programs

You can integrate traps that DX NetOps generates with other network management programs. The traps provide the following information:

- Interface that is affected by the threshold
- Protocol that is affected by the threshold
- Direction of traffic that is affected by the threshold (in, out, or both)
- Threshold that was crossed (for example, 1.23 Mb/s)
- Actual observed traffic that triggered the trap (for example, 1.30 Mb/s)
- Whether the notification indicates a newly observed threshold violation or a cleared threshold violation. For a cleared threshold violation, the reason for clearance is included. The reasons are:
  – Rate fell below the threshold
  – No data was observed for the last time period
  – A time filter prevented data from being considered

Configure Traps for External Programs

Configure DX NetOps traps to collate events with external programs and to supply useful data to the external programs.

Follow these steps:

1. Access the MIB file.
   The MIB contains information about the contents of the traps that DX NetOps sends. The MIB file is located in the following directory: *install_path*\reporter\MIB .
2. Compile the MIB into the network management program, and configure the program to match DX NetOps traps logically.
   A trap is sent when the program observes a new threshold violation and again when the condition is cleared. Configure the network management program to collate these two events, so you clear the warning condition when the program sends the clear condition trap. To accomplish the collation configure the program to look at the following values in the MIB:
   - NetQoSTrafficFlowEntry
   - NetQoSTrafficFlowDataEventStart , which signifies that the trap is new
   - NetQoSTrafficFlowDataEventStop , which signifies that the trap has been cleared

   For more information about defining rules for intelligent trap handling, refer to the documentation for the network management program.
3. Create and deploy DX NetOps traps, if you have not already done so.
4. Verify that the trap destination contains the IP address of the server that hosts the fault management program.
   Traps are sent to the destination established on the **Application Settings** page.

Troubleshoot Issues with Integrated Traps

Review trap settings to correct problems with the DX NetOps trap data that external fault management programs receive.

Follow these steps:

1. Verify that the trap destination is set to the IP address of the host server for the fault management program.
2. Verify that the traps are configured properly.

If you have checked these settings and you still do not receive traps as expected, contact CA Support for help.

Backing Up and Restoring Data

Use the information here as a reference to manually backup and restore the DX NetOps databases.

You can back up some files automatically, by adding the appropriate directories to your regular backup routine:

- Customized configuration files: Custom locations (Any server)
- Historical (15-Minute) data on a stand-alone server or a two-tier distributed deployment: *install_path*\Netflow\datafiles\ReaperArchive15 (Stand-alone or Harvester server)
- High-resolution (1 minute) data: *install_path*\Netflow\datafiles\ReaperArchive (Stand-alone or Harvester server)
- Flow Forensics (raw) data: *install_path*\Netflow\datafiles\HarvesterArchive (Stand-alone or Harvester server)

WARNING

- Run backups concurrently. If you restore data from backups that have different timestamps, problems can result. Ensure that your backed-up data files are timestamped with the same hour. On two-tier systems, backups should be done as simultaneously as possible across systems.
- Store backups to a remote location to guard against the possibility of a hardware or operating system failure on the main server. For example, back up the databases to an administrative share or mapped network drive.
- When a database drop occurs, all procedures in the `mysql.proc` table for that dropped database are dropped.

Before you back up large databases, consider how much disk space the backups will require and the lifespan of the files you are backing up. You probably will not want to back up the raw flow data (`.NFA` files) or back up the entire `datafiles` directory.

Databases to Back Up

DX NetOps uses several databases to store configuration data, reporting data, 15-minute (historical) data, and high resolution (1-minute) data. This section describes databases to consider for backups and gives general recommendations for backup frequency. Data is used differently in different environments, however, so your priorities for safeguarding against data loss could vary.

By excluding some directories, you can safely backup your data without stopping services or collecting data. Always exclude any `*Work` directories in the `datafiles` directory.

All Servers

- Customized configuration files: Back up any other customized configuration files--files that you customized or that were customized by CA Support.
  Location: Files with a `.config` , `.conf` , or `.ini` extension that are located anywhere in the DX NetOps installation path.
  Recommendation: Daily backup

Single Sign-On Program

Back up any customized Single Sign-On configuration settings on the SSO server, which might be the Performance Center server. If you lose customized Single Sign-On settings, you might not be able to log in.

- Single Sign-On (SSO) configuration files, as described in the Single Sign-On article in the CA Performance Management documentation.

NFA Console (Distributed Deployment)

- Reporter database: Contents include the previous 24 hours of Enterprise Overview data, settings, reports, inventory, and synchronization information.
  Recommendation: Weekly backup

Harvester (Distributed Deployment)

- Historical (15-minute) database (archive15 database on a two-tier distributed deployment): The contents of these datafiles include the 15-minute data that is stored for the reporting routers and interfaces.
  Location: *install_path*`\Netflow\datafiles\ReaperArchive15`
  Exclude `*.tmp`
  Recommendation: Weekly backup
- Harvester configuration database: The harvester configuration data is essential to perform the relational mapping that provides access to the 15-minute data. It provides information about devices and interfaces to enable polling, such as persistent IDs for interfaces.

Recommendation: Daily backup

- (*Optional*) Flow Forensics data (HarvesterArchive database): Many administrators do not back up Flow Forensics data because of its short storage life; a maximum of 24 hours. When you decide whether to back up this database, consider the value that this single day of data has in your particular circumstances.
  Location: *install_path*\Netflow\datafiles\HarvesterArchive Exclude *.tmp
- (*Optional*) High-resolution (1-minute) database (archive database): Many administrators do not back up 1-minute data because its high volume requires long backup times.
  The 1-minute data is stored for one month by default.
  Location: *install_path*\Netflow\datafiles\ReaperArchive
  Exclude *.tmp
  Recommendation: Daily backup
- Customized Data Retention configuration files: If you have customized any data retention settings, back up the data retention configuration database.
  Recommendation: Daily backup

  NOTE
  It is unusual to customize data retention settings, except with the assistance of CA Support. Changes to data retention settings can cause problems as the demands on drive space rise.

Stand-Alone Server

- Reporter database: Contents include the previous 24 hours of Enterprise Overview data, settings, reports, inventory, and synchronization information.
  Recommendation: Weekly backup
- Historical (15-minute) data (archive15 database): The contents of this database include the 15-minute data that is stored for the reporting routers and interfaces.
  Location: *install_path*\Netflow\datafiles\ReaperArchive15
  Exclude *.tmp
  Recommendation: Weekly backup
- Harvester configuration database: The harvester configuration data is essential to perform the relational mapping that provides access to the 15-minute data. It provides information about devices and interfaces to enable polling, such as persistent IDs for interfaces.
  Recommendation: Daily backup
- (Optional) Flow Forensics data (HarvesterArchive database): : Many administrators do not back up Flow Forensics data because of its short storage life; a maximum of 24 hours. When you decide whether to back up this database, consider the value that this single day of data has in your particular circumstances.
  Location: *install_path*\Netflow\datafiles\HarvesterArchive Exclude *.tmp
- (Optional) High-resolution (1-minute) data files (archive database): Many administrators do not back up 1-minute data because its high volume requires long backup times.
  The 1-minute data is stored for one month by default.
  Location: *install_path*\Netflow\datafiles\ReaperArchive
  Exclude *.tmp
  Recommendation: Daily backup
- Customized Data Retention database: If you have customized any data retention settings, back up the data retention configuration database.

  NOTE
  It is unusual to customize data retention settings, except with the assistance of CA Support. Changes to data retention settings can cause problems as the demands on drive space rise.

  Recommendation: Daily backup

Back Up the Data

Follow these steps:

1. Determine which DX NetOps databases and files to back up.
   The databases and files to back up for DX NetOps are listed in the following table.

Database	Stand-Alone Server	Harvester Servers (Distributed)	NFA Console Server (Distributed)
reporter	Important	N/A	Important
harvester	Important	Important	N/A
archive15	Recommended	Recommended	N/A
Customized Files	Important	Important	Important
data_retention	Important if customized	Important if customized	N/A
HarvesterArchive	Optional, rarely backed up	Optional, rarely backed up	N/A
archive	Recommended	Recommended	N/A

2. Copy each of the target directories or files to a remote location.
   - Customized configuration files: Various locations
   - HarvesterArchive database: *install_path*\Netflow\datafiles\HarvesterArchive
   - Archive database: *install_path*\Netflow\datafiles\ReaperArchive
   - Archive15 database: *install_path*\Netflow\datafiles\ReaperArchive15
   - Single Sign-On (SSO) configuration files: Back up the following files and directories on the SSO server, which might be the Performance Center server:
     - *install_path*\jre\lib\security directory
     - *install_path*\Portal\Jetty\etc\keystore file
     - *install_path*\Portal\SSO\start.ini file
     - *install_path*\Portal\SSO\etc directory
     - *install_path*\Portal\SSO\conf\wrapper.conf file
     - *install_path*\Portal\SSO\webapps\sso\configuration directory
3. Back up the following databases to a remote location, using `mysqldump` . Back up the `reporter` database last, regardless of the deployment architecture.
   - Customized data_retention database (Stand-alone or Harvester server): `data_retention`
   - harvester database (Stand-alone or Harvester server): `harvester`
   - reporter database (Stand-alone or NFA console): `reporter`

     ```
     mysqldump --routines --events -u root <dbname> --skip-lock-tables > dbbackupname.sql
     ```

     For example, on the NFA console:

     ```
     mysqldump --routines --events -u root reporter --skip-lock-tables > reporterbackup.sql mysqldump --
     routines --events -u root mysql proc > proc.sql
     ```

     On the Harvesters:

     ```
     mysqldump --routines --events -u root harvester --skip-lock-tables > harvesterbackup.sql
     mysqldump --routines --events -u root data_retention --skip-lock-tables > data_retentionbackup.sql
      mysqldump --routines --events -u root mysql proc > proc.sql
     ```
4. (Optional) Verify that the `mysqldump` was successful by checking that the size of the backup is over 1 KB.

   NOTE
   More information:

   See the mysqldump documentation.

Restore the Backups

These steps are for restoring data on a Windows server, but the database path is the same for a Linux server.

Follow these steps:

1. Log in with administrator rights.
2. Restore each of the target directories or files from its remote location to its original location:
   - Customized configuration files: Various locations (Any server)
   - HarvesterArchive database: *install_path*\Netflow\datafiles\HarvesterArchive (Stand-alone or Harvester server)
   - Archive database: *install_path*\Netflow\datafiles\ReaperArchive (Stand-alone or Harvester server)
   - Archive15 database: *install_path*\Netflow\datafiles\ReaperArchive15 (Stand-alone or Harvester server)
3. Restore each of the databases. Restore the
   `reporter`
   database first, regardless of the deployment architecture.
   a. reporter database: `reporter` (Stand-alone or NFA console)
   b. Customized data_retention database: `data_retention` (Stand-alone or Harvester server)
   c. harvester database: `harvester` (Stand-alone or Harvester server)

   TIP

   For best results, restore to a clean installation.

```
mysql –e “drop database <dbname>;”
mysql -e "create database <dbname>;"
mysql -u root <dbname> < dbbackupname.sql
mysql –u root mysql < proc.sql
```

Recommendations for Preserving Data Integrity

To ensure data integrity and to prevent the corruption of the DX NetOps databases, implement the following recommendations on the servers or hardware systems you use for running DX NetOps components:

- Exclude the following directories from real-time antivirus scans:
  - `C:\Windows\Temp`
  - *install_path* and all its subdirectories.
- Do not implement drive space compression.
- Do not install any third-party software, except for the following types of software: antivirus, system management, and time synchronization.
- Install important Microsoft updates. Use discretion when you decide whether to install optional updates.
- Defragment the hard disk drive infrequently. Frequent defragmentation is not necessary: DX NetOps components typically write sequentially to their database so data fragmentation is minimized.

Manage Address-Hostname

IP addresses can be automatically assigned by Dynamic Host Configuration Protocol (DHCP), and addresses can become outdated. You can list, edit, delete or expire one or multiple IP addresses. You can expire stale (outdated) IP addresses to schedule them for a refresh. You also can change the frequency of DNS name updates when you edit an address. You can configure the frequency at which the address-hostname association must be checked. By default, the resolution frequency is three months. The NFA maintains only the latest resolved hostname against the IP address.

The program checks the DNS server and refreshes the DNS name for each expired address.

Follow these steps:

1. Display the **Address-Hostname Configuration** page:
   a. Select **Administration** from the NFA console menu. The **Administration** page opens.
   b. Select **Addresses** from the **Reporting** menu. The **Address-Hostname Configuration** page opens.
2. (Multi-domain environment only) Select the tenant-domain combination that contains the address you want to refresh.
3. **List IP Addresses:**
   – Enter an IP address or type * to list all hosts.
   – Click **List** to view IP addresses.
4. **Hostname Resolution**:
   a. Select an IP address and click **Edit**.
   b. Change the **Hostname** on the **Address-Hostname Configuration** page, if required.
   c. Select one of the following **Hostname Resolution** method that suits your strategy.
      - Do not perform DNS name resolution
      - Automatically resolve, never expire
      - Automatically resolve, resolve again every <custom_duration>
   d. Click **Submit** to save the changes.
5. **Delete an IP Address:**
   a. Select the IP address that you want to change.
   b. Click **Delete**.
   c. Confirm deletion when prompted.
6. **Expire an IP Address:**
   a. Select the IP address that you want to expire.
   b. Click **Expire**.
   c. Confirm expiration when prompted. The selected addresses are scheduled to have their DNS resolution refreshed. The refresh process is typically run within five minutes.

More Information:

IP Address and Hostname Historical Data API

IP Address and Hostname Historical Data

Data Collection

DX NetOps supports stand-alone and two-tier architecture distributed deployments.

The product components work together to collect, process, and store flow data; display the data in reports; and generate traps, events, and scheduled reports.

Two-Tier Architecture

A two-tier architecture deployment consists of the NFA console and one or more Harvesters.

The deployment footprint is reduced, so you have fewer component types to upgrade and maintain. Data handling requires fewer steps and files are not transferred as frequently.

A stand-alone deployment uses the two-tier architecture: the stand-alone server hosts the NFA console and Harvester software.

Data Collection in a Two-Tier Deployment

The NFA console and Harvester components work together in the following way on a 2-tier architecture deployment:

Harvester

Distills the raw flows from the routers, parses the data, and compiles historical (15-minute) data and 1-minute data. The Harvester stores the following data:

- Raw flow data
- 1-minute resolution data
- Historical (15-minute) data

> **NOTE**
> It is important for the Harvester server to have adequate storage space.

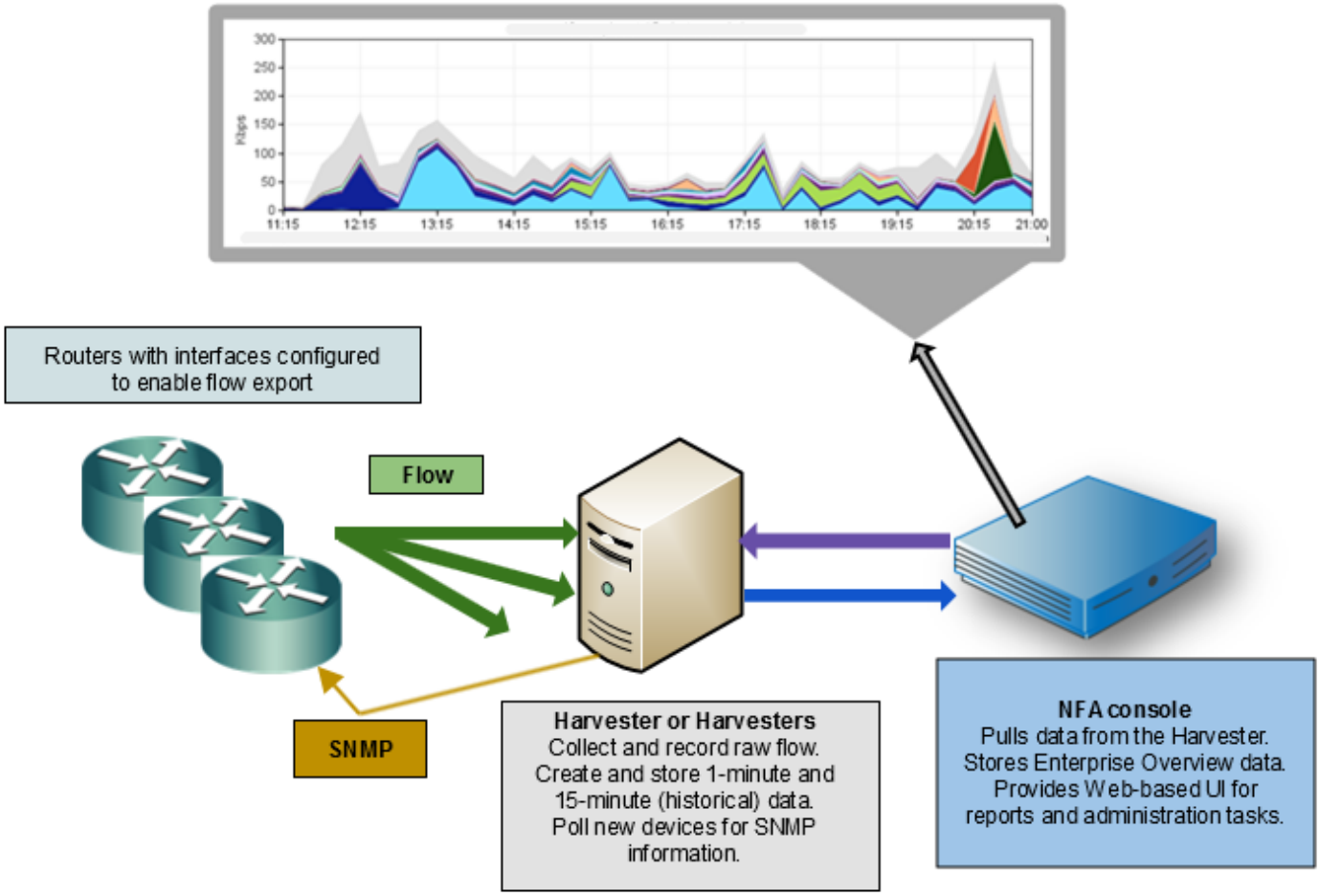
NFA console

Gathers data from the Harvester as needed and displays report data in the web interface. Stores Enterprise Overview data.

Supplies a web-based user interface for administrative tasks and reports and stores the Enterprise Overview data. Reports consist of the following data:

- Enterprise Overview data that is stored locally
- 1-minute data and 15-minute data that is stored on the Harvester

Figure 100: Data Collection in a Two-Tier Deployment



Enterprise Overview Data

The Enterprise Overview data are collected and displayed on the **Enterprise Overview** page. For each report, a maximum of 12 interfaces, protocols, or hosts is included from across the enterprise. Enterprise Overview views are based on the previous 24-hours of data collection. The data are displayed in the following reports:

- **Interface Utilization**
  Selected by: User-defined thresholds for interface capacity utilization.
- **Top Interfaces - In** (inbound traffic) and **Top Interfaces - Out** (outbound traffic)
  Selected by: Amount of interface capacity that was utilized during thereport period. The utilization calculation is based on data volume and interface speeds, The interface speeds are derived from SNMP polling.
- **Top Protocols**
  Selected by: Volume of traffic that used the protocols.
  - **Top Interfaces for Protocol**
    Selected by: Interface traffic volume for the protocol that you clicked in the **Top Protocols** view.
  - **Top Hosts for Protocol**
    Selected by: Host traffic volume for the protocol that you clicked in the **Top Protocols** view.
- **Top Hosts**

Selected by: Volume of total traffic that the hosts generated.

- **Top Interfaces for Host**
  Selected by: Interface traffic volume for the host that you clicked in the **Top Hosts** view.
- **Top Protocols for Host**
  Selected by: Protocol traffic volume for the host that you clicked in the **Top Hosts** view.

Data Lifespan: Enterprise Overview data is retained for 1 month by default.

Storage Location: The data is stored on the NFA console.

Disable Top Protocols and Top Hosts

You can disable **Top Protocols** and **Top Hosts** by changing the default value of **DisableEOVData.**

Follow these steps:

1. Login to Harvester Database:

```
mysql -P3308 harvester
```

2. Execute the following query to change the value of **DisableEOVData**.

```
update parameter_descriptions set defaultvalue = 'Y' where parameter =
 'DisableEOVData';
```

3. Restart CA NFA Repear service.

15-Minute Data

The following types of data are collected and stored as 15-minute resolution data by default.

Overall Traffic

- Total IP, TCP, and UDP traffic that travels in and out of each interface
- Flows and bytes of flow traffic that travels in and out of each interface
- Total flow traffic from each router

While 15-minute data reflect totals for each interface, Enterprise Overview data reflects totals for all interfaces. For example, Enterprise Overview data includes the Top N protocols for all interfaces. The 15-minute data includes the Top N protocols for each interface.

Hosts
Traffic to and from the top 50 IP hosts

Conversations
Traffic for the top 50 IP conversations

Protocols

- *Overall:* Top 100 protocols in and out of each interface
  Reserved Seating rules let you include specific protocols in the top protocols group regardless of their traffic volume.
- *Protocol-Hosts:* Traffic to and from the top 10 hosts for the top 20 protocols
- *Protocol-Conversations:* Traffic for the top 10 conversations for the top 20 protocols

ToS

- *Overall:* Total IP traffic in and out of each interface for each ToS value
- *ToS-Protocols:* Traffic for the top 20 protocols for the top 5 non-zero ToS values
- *ToS-Hosts:* Traffic for the top 10 hosts for the top 5 non-zero ToS values
- *ToS-Conversations:* Traffic for the top 10 conversations for the top 5 non-zero ToS values

AS

- Top 100 source and destination AS values
- Top 20 next hop addresses for each AS value

AS traffic is stored for 100 days by default.

Data Lifespan: Each data type has a default maximum storage period, as shown in the following list:

- Overall traffic: 372 days or 12.4 months
- Protocols (Top 100): 372 days or 12.4 months
- Hosts, Protocol-Hosts, and ToS-Hosts: 67 days
- Conversations (Top 50), Protocol-Conversations, and ToS-Conversations: 67 days
- ToS and ToS-Protocols: 372 days or 12.4 months
- AS: 100 days

Minimum Thresholds: To be included in a 15-minute data report, data traffic must meet or exceed the default minimum thresholds for the 15-minute period:

- Protocols: 50 KB
- Hosts: 100 KB
- Conversations: 100 KB
- ToS: No minimum
- AS: No minimum

Storage Location: Historic (15-minute) data is stored on the Harvester in a two-tier deployment.

Reports: 15-minute data is often used for examining trends and performing capacity analysis. This data appears in the following reports:

- Interface reports that show more than 2 hours of data
- Custom reports
- Analysis reports
- Performance Center views and reports that show more than 2 hours of DX NetOps data

1-Minute Data

The following types of data are stored as 1-minute resolution data by default.

Hosts
Traffic to and from the top 300 IP hosts

Conversations
Traffic to and from the top 300 IP conversations

Protocols

- *Overall:* Top 150 protocols for traffic that travels in and out of each interface
- *Hosts:* Traffic to and from the top 25 hosts that use the top 25 protocols
- *Conversations:* Traffic for the top 25 conversations that use the top 25 protocols

ToS

- *Protocols:* Traffic for the top 25 protocols that use the top 5 ToS values
- *Hosts:* Traffic for the top 25 hosts that use the top 5 ToS values
- *Conversations:* Traffic for the top 25 conversations that use the top 5 ToS values

Data Lifespan: The default maximum storage period for 1-minute data is 1 month.

Storage Location: The 1-minute data is stored on the Harvester in two-tier deployments.

Reports: This type of data is often used for troubleshooting and granular analysis. The data appears in the following reports, provided that the reports are configured to show time ranges of 2 hours or less:

- Interface views and reports
- Performance Center views and reports that show less than 2 hours of DX NetOps data

Raw Data

Raw data is used for even more granular analysis in **Flow Forensics** reports. Raw data is stored on the Harvester for a maximum of 24 hours by default.

IP Address and Hostname Historical Data

The NFA maintained only the latest resolved hostname against the IP address previously. From NFA 10.0.3, we maintain the historical data of the IP address and hostname association.

The following are some benefits of such reverse DNS resolution:

- Identify the domain name of the originator
- Identify the name of the internet service provider assigned to a particular IP address
- Associate host with the IP address
- Whitelist the hosts
- Trace header field for SMTP e-mail, tracking web sites users
- Use it as an anti-spam technique

> **NOTE**
>
> **More Information:**
>
> For more information on how to view the IP Address and Hostname historical data, see IP Address and Hostname API.
>
> Also, see the Manage Address-Hostname page,

Data Retention

The Data Retention Service attempts to prune historical data from DX NetOps when the percentage of free disk space crosses the percentage threshold specified by the `freeSpacePercentage` attribute in the `parameter_descriptions` table in the harvester database. The default value for this percentage threshold is 10%. This means that when there is less than 10% available disk space, the data retention service begins removing data, starting with the oldest data first.

The pruning service runs checks every minute to determine whether there is a need to start cleaning out data (to prevent disk space from being fully consumed). If there is enough disk space, the pruning only occurs once each day to reduce the data to the “max” threshold in the `data_retention.datastores` table. So for some time periods of the day, you can see more than 24 hours’ worth of `.nfa` files.

When data retention determines it is time to prune, the service deals with three main categories of data to prune:

1. NetFlow Archive Data – “Raw” NetFlow data, including NetFlow template data (`.nfa` and `.nta` files). Typically keep 24 hours of this data.

2. Reaper Archive Data – 1-minute summarized data（archive database). Typically keep 30 days of this data.
3. Historical Archive Data – 15-minute summarized data (archive15 database). Typically keep 1 year of this data.

With each pass of pruning, three attempts are made to free disk space, each with a slightly different method:

1. Prune to Max – this is pruning the data to our default thresholds based on type of data (24 hours for raw NetFlow, 30 days for 1-minute data, 1 year for 15-minute data). If pruning to this level results in enough free space, no other pruning is performed.
2. Prune to Min – this will attempt to prune data to meet the minimum free space threshold (10% is the default). Any data required to meet this threshold will be deleted, starting with the oldest data first. If pruning to this level results in enough free space, no other pruning is performed. This level of pruning is also incremental, which means that as soon as the required free space is available, pruning stops until it is needed again.
3. Prune All Data – this will attempt a more aggressive form of the minimum free space pruning. This can result in pruning all data of a given type and is only used as a last resort

The data retention service logs its various pruning activities. If it is unable to free enough disk space to bring free space back above the threshold, it logs as such and tries again at its next scheduled interval.

If you find that you are unable to keep the amount of historical data you need, you should strongly consider either getting a larger disk, or redistributing some of the NetFlow traffic your harvester is receiving to another harvester.

NOTE

More information:

- Service Management

Reference

This section contains the following articles:

CA Network Flow Analysis Service Management

To start, stop, or check the status of services, you can use the Windows Services window or the Linux Service Configuration window.

The DX NetOps and CA Anomaly Detector services are described briefly in the following table. The **Stand-Alone** column shows which services reside on a two-tier stand-alone server. The other server columns show where the services reside in a two-tier distributed deployment. The **Anomaly Detector** services run on the installation server for that program, which may be the stand-alone, NFA console, or separate CA Anomaly Detector server.

Service	Stand-Alone	Harvester	Console	Anomaly Detector
CA NFA Collection and Poller Webservices (Linux: nfa_collpollws) Provides web service interfaces for the NFA console to communicate with the Harvester and Poller (stand-alone and Harvester servers)	Yes	Yes	N/A	N/A

CA NFA Data Retention (Linux: nfa_dataretention) Manages trimming to enforce volume and date limits for retaining data files. 2-tier: 15-minute, 1-minute, and Flow Forensics data (Stand-alone/Harvester)	Yes	Yes	N/A	N/A
CA NFA DNS/SNMP Proxies (Linux: nfa_proxies) Handles SNMP and DNS requests from the Poller. Uses port 8081 by default.	Yes	Yes	N/A	N/A
CA NFA File Server (Linux: nfa_filewebservice) 2-tier: In a distributed deployment, hosts the web service on the Harvesters to handle file requests from the NFA console Pump service.	Yes	Yes	Yes	N/A
CA NFA Harvester (Linux: nfa_harvester) Runs the Harvester/ collector process.	Yes	Yes	N/A	N/A
CA NFA Host Resolver Service Performs hostname lookup for any host that has anomalies detected by CA Anomaly Detector.	Yes	N/A	N/A	Yes
CA NFA Hunter Tracker Service Starts and stops the AnomalyDetector process for CA Anomaly Detector.	Yes	N/A	N/A	Yes
CA NFA Poller (Linux: nfa_poller) Initiates and handles SNMP requests to the devices that export flow to the product.	Yes	Yes	N/A	N/A
CA NFA Pump General file transfer service that is used between the Harvesters and the NFA console.	N/A	Yes	Yes	N/A

CA NFA Reaper (Linux: nfa_reaper) Runs the Reaper process, which processes incoming files and writes out files for 1-minute, 15-minute, and some Enterprise Overview data.	Yes	Yes	N/A	N/A
CA NFA RibSource Provides a static interface--a RIB or Report Information Base interface--to provide the DX NetOps data that appears in Performance Center views.	Yes	N/A	Yes	Yes
CA Performance Center SSO Runs the Single Sign-On authentication software, so users can navigate between the DX NetOps console and Performance Center without signing on again.	Yes	N/A	Yes	N/A
CA MySql (Linux: mysql) Runs the standard MySQL instance using port 3308 and stores MySQL configuration data.	Yes	Yes	Yes	Yes
NetQoS NQMySql (Linux: nfa_mysqlCSE) Runs the Custom Storage Engines and provides an interface to run queries against the Flow Forensics data.	Yes	Yes	N/A	N/A
NetQoS Reporter Manager Service Runs maintenance threads in the background to handle the interoperation of the components.	Yes	N/A	Yes	N/A
NetQoS Reporter/ Analyzer General Services Maintain log retention in the reporter/Logs directory and handles data retention for Enterprise Overview data.	Yes	N/A	Yes	N/A

NetQoS Reporter/ Analyzer Pump Service Collecting and processing Enterprise Overview data from Harvesters.	Yes	N/A	Yes	N/A
NetQoS Reporter/ Analyzer Query Services Act as the interface for Custom reports and Analyses.	Yes	N/A	Yes	N/A
NetQoS ReporterAnalyzer Report Service Executes Custom reports, Analyses, Flow Forensics, and Site to Site reports.	Yes	N/A	Yes	N/A
NetQoS Reporter/ Analyzer Watchdog Polls components for status information, in order to provide administrators with warnings. Also runs simple database integrity checks.	Yes	N/A	Yes	N/A

Service Logs

The service logs are described in the following table. The initial part of each path is the DX NetOps installation path, such as the default locations (`C:\CA\NFA` on a Windows server or `/opt/CA/NFA` on a Linux Harvester server). The table lists the service log file names and locations on Windows and Linux servers, along with any available configuration file to control the log level setting.

Service / Log Information	Stand-Alone	Harvester	Console
CA NFA Collection and Poller Webservices (Linux: nfa_collpollws) Log: `\Netflow\Logs\collpollws-wrapper.log`	Yes	Yes	N/A
CA NFA Data Retention (Linux: nfa_dataretention) Log: `\Netflow\Logs\dataretention-wrapper.log`	Yes	Yes	N/A

CA NFA DNS/SNMP Proxies (Linux: nfa_proxies) Log: `\Netflow\Logs\proxies-wrapper.log`	Yes	Yes	N/A
CA NFA File Server (Linux: nfa_filewebservice) Log: `\Netflow\Logs\fileserver-wrapper.log`	Yes	Yes	N/A
CA NFA Harvester (Linux: nfa_harvester) Log: `\Netflow\Logs\harvester-wrapper.log`	Yes	Yes	N/A
CA NFA Poller (Linux: nfa_poller) Log: `\Netflow\Logs\poller-wrapper.log`	Yes	Yes	N/A
CA NFA Reaper (Linux: nfa_reaper) Log: `\Netflow\Logs\RealtimeReaperErrors<yyyy-mm-dd>.log`	N/A	Yes	N/A
CA NFA RibSource Log: `\Reporter\RIB\NFA\webapps\NFARS\WEB-INF\logs\application.log`	Yes	N/A	Yes

CA Performance Center SSO
Logs:
`\Portal\SSO\logs\SSOService.log`
`\Portal\SSO\logs\wrapper-<yyyy-mm-dd>.log`
	Yes	N/A	Yes

DNS Proxy Web Service
Log:
`\ProxyServices\Logs\DnsProxyWSLog<yyyy-mm-dd>.log`
Log level setting (NFA console only):
`\ProxyServices\Web.config`
`<add key="LogLevel" value="6"/>`
Default value: 4
	Yes	N/A	Yes

NetQoS MySql (Linux: mysql) Log: `\MySql\data\<server_name>.err`	Yes	Yes	Yes
NetQoS NQMySql (Linux: nfa_mysqlCSE) Log: `\Netflow\Logs\oursql_error.log`	N/A	Yes	N/A
NetQoS Reporter Manager Service Log: `\Reporter\Logs\Manager Service Log<yyyy-mm-dd>.log` Log level setting: `\Reporter\ReporterAnalyzer.ManagerService\bin\ReporterManagerService.exe.config` `<setProperties Severity="6" />` Default value: 4	Yes	N/A	Yes
NetQoS Reporter Manager Service thread logs in the `\Reporter\Logs\` directory: MigrationService: `MigrationLog<yyyy-mm-dd>.log` PollerSyncService: `CollectorSyncServiceLog<yyyy-mm-dd>.log` System Maintenance Service: `ManagerService_MaintenanceLog<yyyy-mm-dd>.log`	Yes	N/A	N/A
NetQoS Reporter/Analyzer General Services Log: `\Reporter\Logs\nqservErrors<yyyy-mm-dd>.log`	Yes	N/A	Yes

NetQoS Reporter/Analyzer Pump Service Log: \Reporter\Logs \PumpLog<yyyy-mm-dd>.log Log level setting: \Reporter \NetQoS.ReporterAnalyzer.PumpService \bin\ NetQoS.ReporterAnalyzer.PumpService.exe.config <setProperties Severity="6"/> Default value: 4	Yes	N/A	Yes
NetQoS Reporter/Analyzer Query Services ('Reporting services') Log: \Reporter\Logs \nqreporterErrors<yyyy-mm-dd>.log Automatic Groups task log: \Reporter\Logs \ManagerService_AutomaticGroupsLog<yyyy-mm-dd>.log DNS task log: \Reporter\Logs \ManagerService_DnsLog<yyyy-mm-dd>.log Maintenance tasks log: \Reporter\Logs \ManagerService_MaintenanceLog<yyyy-mm-dd>.log	Yes	N/A	Yes
NetQoS Reporter/Analyzer Report Service Log (Flow Forensics log): \Reporter\Logs \ReportServiceLog<yyyy-mm-dd>.log Log level setting: \Reporter \NetQoS.ReporterAnalyzer.ReportService \ bin \NetQoS.ReporterAnalyzer.ReportService.exe.config <setProperties Severity="6" /> Default value: 4 To log each poll in addition to regular event logging: <add key="VerboseLogging" value="True"/>	Yes	N/A	Yes

NetQoS Reporter/Analyzer Watchdog
Log:
`\Reporter\Logs\WatchdogLog<yyyy-mm-dd>.log`
Log level setting:
`\Reporter\NetQoS.ReporterAnalyzer.WatchdogService\bin\WatchdogService.exe.config`
`<setProperties Severity="6"/>`
Default value: 4
	Yes	N/A	Yes

Multiple processes Log: `\Reporter\Logs\ConsoleErrorsLog<yyyy-mm-dd>.log`	Yes	N/A	N/A
ProductSyncWS Web Service Log: `\Reporter\Logs\ProductSyncWSLog<yyyy-mm-dd>.log`	Yes	N/A	Yes
15-minute Custom Storage Engine (mysqld) Log: `\Netflow\Logs\rpr15.log`	Yes	Yes	N/A

Flow Cloner Configuration Files

You can codify the `FlowCloneDef.ini` to suit your requirement. This section provides the following information:

NOTE

More Information: Introduction to Flow Cloner

Examples of Flow Cloner Configuration Files

The following examples show the contents of a modified `FlowCloneDef.ini` file for a single-NIC server and for a multiple-NIC server environment.

Example: Default Input NIC and Port

In this example, the Flow Cloner uses the first available NIC to send the cloned data to hosts 192.100.0.100 and 192.160.0.100. The destination hosts listen for the data on UDP port 9995.

The example has a line for each host that receives the cloned data. These lines begin with the */dest ip=* token, followed by the host IP address. A semicolon precedes the comments at the end of the line.

```
/use defaults; take default setup

/dest ip=192.100.0.100; send cloned packets to ddev1
```

```
/dest ip=192.160.0.100; send cloned packets to gdev3
```

Example: Custom Input NIC and Port

In this example, the Flow Cloner uses a specified NIC (190.0.0.100) to send the cloned data to the hosts (10.0.0.000, 192.100.0.100, and 192.160.0.100). The Harvesters listen for the original incoming data on UDP port 9994.

> **NOTE**
> If you specify a custom */port* value, it must match the harvester listening port and the destination port that you used to configure the flow. The port values must match or no flow data is cloned.

```
/listen ip=190.0.0.100; Use 190.0.0.100 to send the cloned packets /
port=9994; Listen on port 9994 instead of the default port 9995

/dest ip=10.0.0.000 ; send cloned packets to 10.0.0.000

/dest ip=192.100.0.100; send cloned packets to ddev1

/dest ip=192.160.0.100; send cloned packets to gdev3
```

Example: Custom Input NIC and Destination Port

In this example, the Flow Cloner uses a specified NIC (190.0.0.100) to send the cloned data to host 10.0.0.000. The Harvesters that receive the cloned data listen on UDP port 9996.

```
/listen ip=190.0.0.100; Use 190.0.0.100 to send the cloned packets /
dest port=9996; Listen for the cloned data on port 9996

/dest ip=10.0.0.000 ; send cloned packets to 10.0.0.000
```

Conventions

Observe the following conventions for `FlowCloneDef.ini` files:

- Header line -- Begin the `FlowCloneDef.ini` file with a single line, which contains all of the header token-value entries.
- Destination host lines -- Follow the header line with a line for each destination host. Use the following format /dest ip=<X.X.X.X>, where <X.X.X.X> is the IP address of the destination host. You can put the destination host lines in any order, but the dest ip= lines must follow the header line.
- Tokens -- Precede all values in the file with a token that identifies which type of value is defined. Precede each token with a **|** or */* character to signal that a token follows.
  Tokens are ignored if they are:
  - Unsupported
  - Entered incorrectly
  - Have no specified values

  The */use defaults* token is ignored if the header line also contains one or more token-value entries.
- Commenting Out Content:

- Entire line -- Enter a semicolon (;) at the beginning of the line. The entire line is ignored.
- Content from a particular point to the start of the next token -- Enter a semicolon (;) at the start of the content to ignore.
- Everything from a particular point to the end of the line -- Enter two semicolons (;;) at the start of the content to ignore.

Characteristics of Cloned Packets

Cloned IP packets have the following characteristics:

- ToS value of cloned packets = 0 (zero), regardless of the original packet value. This characterizes the ToS value of the cloned packets that contain the flows. The ToS values of the traffic that the flows describe are not affected.
- Sender port = 10,000, regardless of the sender port of the original packet
- Time To Live (TTL) value = 255, regardless of the TTL value of the original packet
- Fragment ID - Arbitrary number

Network Flow Analysis OData API

DX NetOps (NFA) offers APIs that enables you to extract data from the DX NetOps or perform Administration operations.It enables integration between DX NetOps and other external applications.

NOTE

The Network Flow Analysis OData API currently supports only English Locale.

The DX NetOps OData API uses the OData v4.0 industry standard. For more information on documentation of this standard, see OData v4.0 web page.The Open Data Protocol (OData) enables the creation of REST-based data services, which allows resources to be defined in a data model and be identified by using the Uniform Resource Locators (URLs).

NFA Entities / Operations	Read	Add	Single Edit	Bulk Edit	Single Delete	Bulk Delete	Merge	Enable/ Disable
Routers	Yes	No	Yes	Yes	Yes	Yes	No	Yes
Interfaces	Yes	No	Yes	Yes	Yes	Yes	Yes	No
Interface Aggregations	Yes	Yes	Yes	No	Yes	Yes	No	No
Interface Template	Yes	Yes	Yes	No	Yes	No	No	No
Available Interface	Yes	No	No	No	Yes	No	No	Yes
Custom Virtual Interface	Yes	Yes	Yes	No	Yes	Yes	No	No
Harvester	Yes	Yes	Yes	No	Yes	No	No	No
Application Settings	Yes	No	Yes	No	No	No	No	No
Reporter Health	Yes	No	No	No	No	No	No	No
Harvester Health	Yes	No	No	No	No	No	No	No
Flow Statistics	Yes	No	No	No	No	No	No	No

Watchdog Settings	Yes	No	Yes	No	No	No	No	No
Application Mapping	Yes	Yes	Yes	No	Yes	Yes	No	No
Port Priority Rules	Yes	Yes	Yes	No	Yes	Yes	No	No
Reserved Seating	Yes	Yes	Yes	No	Yes	Yes	No	No
AS Names	Yes	No	Yes	No	No	No	No	No

NOTE

CA NFA lets you perform only **Read** operation on the Reporter Health, Harvester Health, Flow Statistics, Domain, SNMP Profiles, and Reporter entities.

API Details

The CA NFA OData API URL consists of three parts, which are Service root URL, Resource path and Query options.

Syntax:

```
<Service Root URL>/<Resource Path>?<Query Options>
```

Where,

- **Service Root URL**
  Identifies the root of an CA NFA OData service. A GET request to this URL returns the format-specific service document. The service root URL must terminate in a forward slash.

  NOTE

  The CA NFA OData API uses 8981 port number.

  To change the default port, update the value of the **ReporterAnalyzer.ODataServicePort** in the ReporterAnalyzer.ini file and restart the CA NFA OData service.

  If NFA is integrated with CAPC, the new port value is synced in the next poll cycle of CAPC or you may perform a manual resync with NFA data source.
- **Resource Path**
  Resources exposed by a CA NFA OData service is addressed by a corresponding resource path URL, which allows client to interact with a resource aspect.
- **Query Options**
  The query options part of an OData URL specifies three types of information, which are system query options, custom query options, and parameter aliases.
  The system query options are query string parameters that control the amount and order of the data returned for the resource identified by the URL. The names of all system query options are optionally prefixed with a dollar ($) character. The custom query options provide an extensible mechanism for service-specific information to be placed in a URL query string.

Example:

```
http://127.0.0.1:8981/path/SampleService.svc/Categories(1)/Products?$top=2&
$orderby=Name
```

Authentication

All users who have DX NetOps credentials can use the CA NFA OData API. To log in to the CA NFA OData API, use your DX NetOps credentials. The CA NFA OData API uses Single Sign-On (SSO) for credential authentication.

You can perform HTTP Basic authentication to log in to CA NFA OData API, which is built in the HTTP protocol. The client sends a HTTP request with the authorization header that contains the basic word followed by a space and a `base64-encoded` string username:password. For example, to authorize as ***demo/p@55w0rd,*** the client must send

```
Authorization:
Basic ZGVtbzpwQDU1dzByZA==
```

where **ZGVtbzpwQDU1dzByZA==** is the base64 encoded string of ***demo:p@55w0rd.***

About

The API for the About page displays the following details about your DX NetOps installation:

- Version history
- Licensed Devices in Use
- License Utilization

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/ProductInfo()
```

Method

GET

Sample Request

```
http://127.0.0.1:8981/odata/api/ProductInfo()
```

Sample Response


```
{
 "@odata.context": "$metadata#com.ca.nfa.odata.productInfo",
 "NetworkFlowAnalysisVersionHistory": [
 {
 "Version": "9.3.3.0",
 "Description": "Network Flow Analysis 9.3 (build 0)",
 "InstallDate": "10/06/2017 11:28:17 AM",
 "Update": "3"
 },
 {
 "Version": "10.0.0",
 "Description": "CA Network Flow Analysis 10.0 (build 0)",
 "InstallDate": "02/02/2019 08:44:45 PM",
 "Update": 0
 }
```

```
 ],
 "LicensedDevicesInUse": 75,
 "LicenseUtilization": "150.0%"
}
```

Metadata

The structure of CA NFA OData API service is declared in Metadata as defined by OData specification. The Metadata defines the contract such that users knows which requests can be executed, displays the structure of the result and how a service can be navigated.

CA NFA OData services expose their entity model at the metadata URL, formed by appending `$metadata` to the service root URL.

Request URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/$metadata
```

You can request a particular format of metadata either by using the Accept header or by using the `$format` system query option.

For Example:

```
http://host/service/$metadata?$format=json
```

Sample Request

```
http://127.0.0.1:8981/odata/api/$metadata
```

Sample Response

```
This XML file does not appear to have any style information associated with it. The
document tree is shown below.
<edmx:Edmx xmlns:edmx="http://docs.oasis-open.org/odata/ns/edmx" Version="4.0">
<edmx:DataServices>
<Schema xmlns="http://docs.oasis-open.org/odata/ns/edm" Namespace="com.ca.nfa.odata">
<EntityType Name="asNums">
<Key>
<PropertyRef Name="asnum"/>
</Key>
<Property Name="router" Type="Edm.String"/>
<Property Name="interface" Type="Edm.Int64"/>
<Property Name="timestamp" Type="Edm.Int64"/>
<Property Name="asnum" Type="Edm.Int64"/>
<Property Name="asdir" Type="Edm.Int64"/>
<Property Name="inoctets" Type="Edm.Double"/>
<Property Name="outoctets" Type="Edm.Double"/>
<NavigationProperty Name="asNextHop" Type="Collection(com.ca.nfa.odata.asNextHop)"/>
</EntityType>
```

```
<EntityType Name="asNames">
<Key>
<PropertyRef Name="ASNumber"/>
</Key>
<Property Name="ASNumber" Type="Edm.Int64" Nullable="false"/>
<Property Name="Description" Type="Edm.String"/>
<Property Name="isBaseDescription" Type="Edm.String"/>
<Property Name="DomainId" Type="Edm.Int64"/>
</EntityType>
<EntityType Name="harvesterHealth">
<Key>
<PropertyRef Name="HarvesterId"/>
</Key>
<Property Name="HarvesterId" Type="Edm.Int32" Nullable="false"/>
<Property Name="HarvesterAddress" Type="Edm.String" Nullable="false"/>
<Property Name="SNMPServiceStatus" Type="Edm.String"/>
<Property Name="DBTables" Type="Edm.Int32"/>
<Property Name="UpdateTime" Type="Edm.Int32"/>
<Property Name="LastHarvester" Type="Edm.Int32"/>
<Property Name="LastBilling" Type="Edm.Int32"/>
<Property Name="DBRepairs" Type="Edm.Int32"/>
<Property Name="DBFailures" Type="Edm.Int32"/>
<Property Name="PollerDBTables" Type="Edm.Int32"/>
<Property Name="PollerDBRepairs" Type="Edm.Int32"/>
<Property Name="PollerDBFailures" Type="Edm.Int32"/>
<Property Name="DataRetentionDBTables" Type="Edm.Int32"/>
<Property Name="DataRetentionDBRepairs" Type="Edm.Int32"/>
<Property Name="DataRetentionDBFailures" Type="Edm.Int32"/>
<Property Name="HarvestStatus" Type="Edm.String"/>
<Property Name="WebServerStatus" Type="Edm.String"/>
<Property Name="CollectorPollerWSStatus" Type="Edm.String"/>
<Property Name="ProxyServiceStatus" Type="Edm.String"/>
<Property Name="DataRetentionServiceStatus" Type="Edm.String"/>
<Property Name="PollerServiceStatus" Type="Edm.String"/>
<Property Name="DBConnectStatus" Type="Edm.String"/>
<Property Name="PollerDBConnectStatus" Type="Edm.String"/>
<Property Name="DataRetentionDBConnectStatus" Type="Edm.String"/>
<Property Name="MemoryUtilization" Type="Edm.String"/>
</EntityType>
<EntityType Name="conversations">
<Key>
<PropertyRef Name="SrcHost"/>
<PropertyRef Name="DestHost"/>
</Key>
<Property Name="router" Type="Edm.String"/>
<Property Name="interface" Type="Edm.Int64"/>
```

```
<Property Name="timestamp" Type="Edm.Int64"/>
<Property Name="protocol" Type="Edm.Int64"/>
<Property Name="SrcHost" Type="Edm.String"/>
<Property Name="DestHost" Type="Edm.String"/>
<Property Name="inoctets" Type="Edm.Double"/>
<Property Name="outoctets" Type="Edm.Double"/>
<NavigationProperty Name="protocols" Type="Collection(com.ca.nfa.odata.protocols)"
 Partner="conversations"/>
</EntityType>
<EntityType Name="reservedSeatingRules">
<Key>
<PropertyRef Name="ID"/>
</Key>
<Property Name="ID" Type="Edm.Int64" Nullable="false"/>
<Property Name="protocol" Type="Edm.Int64"/>
<Property Name="port" Type="Edm.Int64"/>
<Property Name="description" Type="Edm.String"/>
</EntityType>
<EntityType Name="harvesters">
<Key>
<PropertyRef Name="HarvesterId"/>
</Key>
<Property Name="HarvesterId" Type="Edm.Int64" Nullable="false"/>
<Property Name="HarvesterAddress" Type="Edm.String"/>
<Property Name="Description" Type="Edm.String"/>
<Property Name="ManagementServerPort" Type="Edm.Int32"/>
<Property Name="FlowEnabledLastModified" Type="Edm.Int32"/>
<Property Name="FlowEnabledLastDeployed" Type="Edm.Int32"/>
<Property Name="LastPushCheckpoint" Type="Edm.Int32"/>
<Property Name="LastPullCheckpoint" Type="Edm.Int32"/>
<Property Name="ApplicationMappingLastDeployed" Type="Edm.Int32"/>
<Property Name="PortDefinitionsLastDeployed" Type="Edm.Int32"/>
<Property Name="ReservedSeatingLastDeployed" Type="Edm.Int32"/>
<Property Name="TrapDefinitionsLastDeployed" Type="Edm.Int32"/>
<Property Name="DomainId" Type="Edm.Int64"/>
<Property Name="DomainName" Type="Edm.String"/>
<Property Name="DomainDescription" Type="Edm.String"/>
<Property Name="TenantId" Type="Edm.Int32"/>
<Property Name="TenantName" Type="Edm.String"/>
<NavigationProperty Name="harvesterHealth"
 Type="Collection(com.ca.nfa.odata.harvesterHealth)"/>
<NavigationProperty Name="flowStats" Type="Collection(com.ca.nfa.odata.flowStats)"/>
</EntityType>
<EntityType Name="addressesHostNames">
<Key>
<PropertyRef Name="Address1"/>
```

```
<PropertyRef Name="HostIp1"/>
<PropertyRef Name="Valid_From"/>
</Key>
<Property Name="Address" Type="Edm.String"/>
<Property Name="Name" Type="Edm.String"/>
<Property Name="AutoResolve" Type="Edm.String"/>
<Property Name="TTL" Type="Edm.String"/>
<Property Name="LastAccessed" Type="Edm.Int64"/>
<Property Name="ExpireTime" Type="Edm.Int64"/>
<Property Name="DomainId" Type="Edm.Int64"/>
<Property Name="HostIp" Type="Edm.String"/>
<Property Name="HostName" Type="Edm.String"/>
<Property Name="Valid_From" Type="Edm.Int64" Nullable="false"/>
<Property Name="Valid_To" Type="Edm.Int64"/>
</EntityType>
<EntityType Name="interfaceTemplate">
<Key>
<PropertyRef Name="templateId"/>
</Key>
<Property Name="templateId" Type="Edm.Int32" Nullable="false"/>
<Property Name="name" Type="Edm.String"/>
<Property Name="nameTemplate" Type="Edm.String"/>
<Property Name="descriptionTemplate" Type="Edm.String"/>
</EntityType>
<EntityType Name="reporterHealth">
<Key>
<PropertyRef Name="UpdateTime"/>
</Key>
<Property Name="UpdateTime" Type="Edm.Int64" Nullable="false"/>
<Property Name="SNMPStatus" Type="Edm.String"/>
<Property Name="MemoryUtilization" Type="Edm.Double"/>
<Property Name="DBConnectStatus" Type="Edm.String"/>
<Property Name="GeneralServiceStatus" Type="Edm.String"/>
<Property Name="PumpServiceStatus" Type="Edm.String"/>
<Property Name="QueryServiceStatus" Type="Edm.String"/>
<Property Name="ReportServiceStatus" Type="Edm.String"/>
<Property Name="RibServiceStatus" Type="Edm.String"/>
<Property Name="SSOServiceStatus" Type="Edm.String"/>
</EntityType>
<EntityType Name="protocols">
<Key>
<PropertyRef Name="protocol"/>
</Key>
<Property Name="router" Type="Edm.String"/>
<Property Name="interface" Type="Edm.Int64" Nullable="false"/>
<Property Name="timestamp" Type="Edm.Int64" Nullable="false"/>
```

```
<Property Name="protocol" Type="Edm.Int64" Nullable="false"/>
<Property Name="inoctets" Type="Edm.Double"/>
<Property Name="outoctets" Type="Edm.Double"/>
<NavigationProperty Name="hosts" Type="Collection(com.ca.nfa.odata.hosts)"
 Partner="protocols"/>
<NavigationProperty Name="conversations"
 Type="Collection(com.ca.nfa.odata.conversations)" Nullable="false" Partner="protocols"/
>
</EntityType>
<EntityType Name="router">
<Key>
<PropertyRef Name="ID"/>
</Key>
<Property Name="ID" Type="Edm.Int32" Nullable="false"/>
<Property Name="routerAddress" Type="Edm.String"/>
<Property Name="sysDescr" Type="Edm.String"/>
<Property Name="sysName" Type="Edm.String"/>
<Property Name="deviceName" Type="Edm.String"/>
<Property Name="deviceAlias" Type="Edm.String"/>
<Property Name="sysUptime" Type="Edm.Int32"/>
<Property Name="lastData" Type="Edm.Int32"/>
<Property Name="lastReboot" Type="Edm.Int32"/>
<Property Name="lastRefresh" Type="Edm.Int32"/>
<Property Name="lastDiscovery" Type="Edm.Int32"/>
<Property Name="lastHarvesterUpdate" Type="Edm.Int32"/>
<Property Name="firstPollError" Type="Edm.Int32"/>
<Property Name="nextPollRetry" Type="Edm.Int32"/>
<Property Name="harvesterID" Type="Edm.Int32"/>
<Property Name="profileId" Type="Edm.Int32"/>
<Property Name="snmpVersion" Type="Edm.Int32"/>
<Property Name="snmpPort" Type="Edm.Int32"/>
<Property Name="snmpTimeout" Type="Edm.Int32"/>
<Property Name="snmpRetry" Type="Edm.Int32"/>
<Property Name="snmpMaxRows" Type="Edm.Int32"/>
<Property Name="ifNumber" Type="Edm.Int32"/>
<Property Name="interfaceCount" Type="Edm.Int32"/>
<Property Name="agentCount" Type="Edm.Int32"/>
<Property Name="dnsLastLookupTime" Type="Edm.Int32"/>
<Property Name="dnsExpireTime" Type="Edm.Int32"/>
<Property Name="syncUpdateTime" Type="Edm.Int32"/>
<Property Name="routerName" Type="Edm.String"/>
<Property Name="routerUpdatedOn" Type="Edm.Int32"/>
<Property Name="templateId" Type="Edm.Int32"/>
<Property Name="snmpProxyAddress" Type="Edm.String"/>
<Property Name="dnsProxyAddress" Type="Edm.String"/>
<Property Name="tenantId" Type="Edm.Int32"/>
```

```
<Property Name="domainID" Type="Edm.Int32"/>
<Property Name="flowStatus" Type="Edm.String"/>
<Property Name="lifecyclestate" Type="Edm.String"/>
<Property Name="profileName" Type="Edm.String"/>
<Property Name="enabledInterfacesCount" Type="Edm.Int32"/>
<Property Name="harvesterAddress" Type="Edm.String"/>
</EntityType>
<EntityType Name="applicationSettings">
<Key>
<PropertyRef Name="Parameter"/>
</Key>
<Property Name="Parameter" Type="Edm.String" Nullable="false"/>
<Property Name="Label" Type="Edm.String"/>
<Property Name="Value" Type="Edm.String"/>
</EntityType>
<EntityType Name="asNextHop">
<Key>
<PropertyRef Name="asnum"/>
</Key>
<Property Name="router" Type="Edm.String"/>
<Property Name="interface" Type="Edm.Int64"/>
<Property Name="timestamp" Type="Edm.Int64"/>
<Property Name="asnum" Type="Edm.Int64"/>
<Property Name="nexthop" Type="Edm.String"/>
<Property Name="asdir" Type="Edm.Int64"/>
<Property Name="inoctets" Type="Edm.Double"/>
<Property Name="outoctets" Type="Edm.Double"/>
</EntityType>
<EntityType Name="interfaces">
<Key>
<PropertyRef Name="ID"/>
</Key>
<Property Name="ID" Type="Edm.Int64" Nullable="false"/>
<Property Name="AgentType" Type="Edm.String"/>
<Property Name="RouterAddress" Type="Edm.String"/>
<Property Name="RouterId" Type="Edm.Int64"/>
<Property Name="Description" Type="Edm.String"/>
<Property Name="InSpeed" Type="Edm.Int64"/>
<Property Name="OutSpeed" Type="Edm.Int64"/>
<Property Name="IfIndex" Type="Edm.Int64"/>
<Property Name="PersistentIfIndex" Type="Edm.Int64"/>
<Property Name="IfType" Type="Edm.String"/>
<Property Name="UpdatedOn" Type="Edm.Int32"/>
<Property Name="Enabled" Type="Edm.String"/>
<Property Name="LastData" Type="Edm.Int32"/>
<Property Name="HarvesterAddress" Type="Edm.String"/>
```

```
<Property Name="Name" Type="Edm.String"/>
<Property Name="TrafficStatus" Type="Edm.String"/>
<Property Name="DomainId" Type="Edm.Int32"/>
<Property Name="DomainName" Type="Edm.String"/>
<Property Name="Subnet" Type="Collection(Edm.String)"/>
<Property Name="ComponentAgentIds" Type="Collection(Edm.Int64)"/>
<NavigationProperty Name="hosts" Type="Collection(com.ca.nfa.odata.hosts)"
 Nullable="false"/>
<NavigationProperty Name="utilization" Type="Collection(com.ca.nfa.odata.utilization)"
 Nullable="false"/>
<NavigationProperty Name="protocols" Type="Collection(com.ca.nfa.odata.protocols)"
 Nullable="false"/>
<NavigationProperty Name="toss" Type="Collection(com.ca.nfa.odata.toss)"
 Nullable="false"/>
<NavigationProperty Name="conversations"
 Type="Collection(com.ca.nfa.odata.conversations)" Nullable="false"/>
<NavigationProperty Name="asNums" Type="Collection(com.ca.nfa.odata.asNums)"
 Nullable="false"/>
<NavigationProperty Name="asNextHop" Type="Collection(com.ca.nfa.odata.asNextHop)"
 Nullable="false"/>
</EntityType>
<EntityType Name="snmpprofiles">
<Key>
<PropertyRef Name="id"/>
</Key>
<Property Name="id" Type="Edm.Int32" Nullable="false"/>
<Property Name="itemId" Type="Edm.Int32"/>
<Property Name="name" Type="Edm.String"/>
<Property Name="version" Type="Edm.Int32"/>
<Property Name="port" Type="Edm.Int32"/>
<Property Name="updatedOn" Type="Edm.Int32"/>
<Property Name="modifiedOn" Type="Edm.Int32"/>
<Property Name="tenantId" Type="Edm.Int32"/>
</EntityType>
<EntityType Name="predefinedApplicationMappings">
<Key>
<PropertyRef Name="ID"/>
</Key>
<Property Name="ID" Type="Edm.Int64" Nullable="false"/>
<Property Name="Name" Type="Edm.String"/>
<Property Name="Description" Type="Edm.String"/>
<Property Name="newPort" Type="Edm.Int32"/>
<Property Name="portType" Type="Edm.String"/>
</EntityType>
<EntityType Name="hosts">
<Key>
```

```
<PropertyRef Name="host"/>
</Key>
<Property Name="router" Type="Edm.String"/>
<Property Name="interface" Type="Edm.Int64"/>
<Property Name="timestamp" Type="Edm.Int64"/>
<Property Name="protocol" Type="Edm.Int64"/>
<Property Name="host" Type="Edm.String"/>
<Property Name="inoctets" Type="Edm.Double"/>
<Property Name="outoctets" Type="Edm.Double"/>
<NavigationProperty Name="protocols" Type="Collection(com.ca.nfa.odata.protocols)"
 Partner="hosts"/>
</EntityType>
<EntityType Name="portPriorityRules">
<Key>
<PropertyRef Name="ID"/>
</Key>
<Property Name="ID" Type="Edm.Int64" Nullable="false"/>
<Property Name="portType" Type="Edm.String"/>
<Property Name="startPort" Type="Edm.Int64"/>
<Property Name="endPort" Type="Edm.Int64"/>
<Property Name="Description" Type="Edm.String"/>
</EntityType>
<EntityType Name="domains">
<Key>
<PropertyRef Name="ID"/>
</Key>
<Property Name="ID" Type="Edm.Int32" Nullable="false"/>
<Property Name="ItemID" Type="Edm.Int32"/>
<Property Name="Name" Type="Edm.String"/>
<Property Name="Description" Type="Edm.String"/>
<Property Name="TenantId" Type="Edm.Int32"/>
<Property Name="GroupId" Type="Edm.Int32"/>
<Property Name="UpdatedOn" Type="Edm.Int32"/>
<Property Name="TenantName" Type="Edm.String"/>
</EntityType>
<EntityType Name="utilization">
<Key>
<PropertyRef Name="protocol"/>
</Key>
<Property Name="router" Type="Edm.String"/>
<Property Name="interface" Type="Edm.Int64" Nullable="false"/>
<Property Name="timestamp" Type="Edm.Int64" Nullable="false"/>
<Property Name="protocol" Type="Edm.Int64" Nullable="false"/>
<Property Name="inoctets" Type="Edm.Double"/>
<Property Name="outoctets" Type="Edm.Double"/>
</EntityType>
```

```
<EntityType Name="reporter">
<Key>
<PropertyRef Name="UpdateTime"/>
</Key>
<Property Name="UpdateTime" Type="Edm.Int64" Nullable="false"/>
<NavigationProperty Name="reporterHealth"
 Type="Collection(com.ca.nfa.odata.reporterHealth)"/>
</EntityType>
<EntityType Name="toss">
<Key>
<PropertyRef Name="tos"/>
</Key>
<Property Name="router" Type="Edm.String"/>
<Property Name="interface" Type="Edm.Int64" Nullable="false"/>
<Property Name="timestamp" Type="Edm.Int64" Nullable="false"/>
<Property Name="tos" Type="Edm.Int64"/>
<Property Name="inoctets" Type="Edm.Double"/>
<Property Name="outoctets" Type="Edm.Double"/>
<NavigationProperty Name="protocols" Type="Collection(com.ca.nfa.odata.protocols)"/>
<NavigationProperty Name="conversations"
 Type="Collection(com.ca.nfa.odata.conversations)"/>
<NavigationProperty Name="hosts" Type="Collection(com.ca.nfa.odata.hosts)"/>
</EntityType>
<EntityType Name="availableInterfaces">
<Key>
<PropertyRef Name="ID"/>
</Key>
<Property Name="ID" Type="Edm.Int64" Nullable="false"/>
<Property Name="routerId" Type="Edm.Int32"/>
<Property Name="name" Type="Edm.String"/>
<Property Name="description" Type="Edm.String"/>
<Property Name="speed" Type="Edm.Int64"/>
<Property Name="ifMapId" Type="Edm.Int32"/>
<Property Name="ifIndex" Type="Edm.Int64"/>
<Property Name="ifDescr" Type="Edm.String"/>
<Property Name="ifType" Type="Edm.Int32"/>
<Property Name="ifName" Type="Edm.String"/>
<Property Name="portName" Type="Edm.String"/>
<Property Name="vrfName" Type="Edm.String"/>
<Property Name="ifAlias" Type="Edm.String"/>
<Property Name="UpdatedOn" Type="Edm.Int32"/>
<Property Name="enabled" Type="Edm.String"/>
<Property Name="LastFlow" Type="Edm.Int32"/>
<Property Name="IpAddr" Type="Edm.String"/>
</EntityType>
<EntityType Name="watchDogSettings">
```

```
<Key>
<PropertyRef Name="Parameter"/>
</Key>
<Property Name="Parameter" Type="Edm.String" Nullable="false"/>
<Property Name="Label" Type="Edm.String"/>
<Property Name="Value" Type="Edm.String"/>
</EntityType>
<EntityType Name="applicationMappings">
<Key>
<PropertyRef Name="ID"/>
</Key>
<Property Name="ID" Type="Edm.Int64" Nullable="false"/>
<Property Name="Description" Type="Edm.String"/>
<Property Name="protocol" Type="Edm.Int64"/>
<Property Name="tos" Type="Edm.Int64"/>
<Property Name="ip" Type="Edm.String"/>
<Property Name="mask" Type="Edm.Int64"/>
<Property Name="beginPort" Type="Edm.Int64"/>
<Property Name="endPort" Type="Edm.Int64"/>
<Property Name="newPort" Type="Edm.Int64"/>
<Property Name="nbar2EngineId" Type="Edm.Int64"/>
<Property Name="nbar2ApplicationId" Type="Edm.Int64"/>
<Property Name="ruleType" Type="Edm.String"/>
<Property Name="Name" Type="Edm.String"/>
</EntityType>
<EntityType Name="flowStats">
<Key>
<PropertyRef Name="HarvesterId"/>
</Key>
<Property Name="HarvesterId" Type="Edm.Int32" Nullable="false"/>
<Property Name="Created" Type="Edm.String"/>
<Property Name="PointTime" Type="Edm.Int32"/>
<Property Name="FlowRate" Type="Edm.Int32"/>
<Property Name="DroppedFlows" Type="Edm.Int32"/>
<Property Name="DroppedPackets" Type="Edm.Int32"/>
<Property Name="MapFailures" Type="Edm.Int32"/>
<Property Name="NoFlowsFound" Type="Edm.Int32"/>
<Property Name="HeaderFailures" Type="Edm.Int32"/>
<Property Name="RouterReboots" Type="Edm.Int32"/>
</EntityType>
<ComplexType Name="limiters">
<Property Name="defaultExpandTopLimit" Type="Edm.Int32"/>
<Property Name="expandMaxTopLimit" Type="Edm.Int32"/>
<Property Name="requestTimeout" Type="Edm.Int32"/>
<Property Name="maxRequestTimeout" Type="Edm.Int64"/>
<Property Name="requestThreadPoolCoreSize" Type="Edm.Int32"/>
```

```
<Property Name="requestThreadPoolMaxSize" Type="Edm.Int32"/>
<Property Name="defaultQueryTimeout" Type="Edm.Int32"/>
<Property Name="maxQueryTimeout" Type="Edm.Int32"/>
<Property Name="defaultPageSize" Type="Edm.Int32"/>
<Property Name="maxPageSize" Type="Edm.Int32"/>
<Property Name="enableMultiTenancy" Type="Edm.Boolean"/>
</ComplexType>
<ComplexType Name="revisionHistory">
<Property Name="Version" Type="Edm.String"/>
<Property Name="Description" Type="Edm.String"/>
<Property Name="InstallDate" Type="Edm.String"/>
<Property Name="Update" Type="Edm.String"/>
</ComplexType>
<ComplexType Name="refreshInterface">
<Property Name="routerAddress" Type="Edm.String"/>
<Property Name="persistentId" Type="Edm.Int64"/>
<Property Name="ifIndex" Type="Edm.Int64"/>
<Property Name="snmpDataIsValid" Type="Edm.Boolean"/>
<Property Name="ifSpeed" Type="Edm.Int64"/>
<Property Name="inSpeed" Type="Edm.Int64"/>
<Property Name="outSpeed" Type="Edm.Int64"/>
<Property Name="ifType" Type="Edm.Int64"/>
<Property Name="updatedOn" Type="Edm.Int64"/>
<Property Name="ifName" Type="Edm.String"/>
<Property Name="ifAlias" Type="Edm.String"/>
<Property Name="ifDescr" Type="Edm.String"/>
<Property Name="ifIpAddr" Type="Edm.String"/>
<Property Name="portName" Type="Edm.String"/>
<Property Name="vrfName" Type="Edm.String"/>
</ComplexType>
<ComplexType Name="systemStatusList">
<Property Name="SystemType" Type="Edm.String"/>
<Property Name="LastContact" Type="Edm.Int64"/>
<Property Name="MemoryUtilization" Type="Edm.Double"/>
<Property Name="CPUUtilization" Type="Edm.Double"/>
<Property Name="Status" Type="Edm.String"/>
<Property Name="Address" Type="Edm.String"/>
<Property Name="ActiveInterfaces" Type="Edm.Int32"/>
<Property Name="TenantID" Type="Edm.String"/>
<Property Name="DomainID" Type="Edm.String"/>
<Property Name="TenantItemID" Type="Edm.String"/>
<Property Name="DomainItemID" Type="Edm.String"/>
<Property Name="TenantName" Type="Edm.String"/>
<Property Name="DomainName" Type="Edm.String"/>
<Property Name="StartTime" Type="Edm.Int64"/>
</ComplexType>
```

```
<ComplexType Name="refreshRouter">
<Property Name="address" Type="Edm.String"/>
<Property Name="sysName" Type="Edm.String"/>
<Property Name="sysDescr" Type="Edm.String"/>
<Property Name="sysUpTime" Type="Edm.Int64"/>
<Property Name="ifNumber" Type="Edm.Int64"/>
<Property Name="updatedOn" Type="Edm.Int64"/>
</ComplexType>
<ComplexType Name="systemStatus">
<Property Name="SystemStatusList" Type="Collection(com.ca.nfa.odata.systemStatusList)"/>
</ComplexType>
<ComplexType Name="portTrafficStatus">
<Property Name="PortName" Type="Edm.String"/>
</ComplexType>
<ComplexType Name="attribute">
<Property Name="name" Type="Edm.String"/>
<Property Name="value" Type="Edm.String"/>
</ComplexType>
<ComplexType Name="csnmprefresh">
<Property Name="status" Type="Edm.String"/>
<Property Name="profileName" Type="Edm.String"/>
<Property Name="refreshRouter" Type="Collection(com.ca.nfa.odata.refreshRouter)"/>
<Property Name="refreshInterface" Type="Collection(com.ca.nfa.odata.refreshInterface)"/>
</ComplexType>
<ComplexType Name="productInfo">
<Property Name="VersionHistory" Type="Collection(com.ca.nfa.odata.revisionHistory)"/>
<Property Name="LicensedDevicesInUse" Type="Edm.Int32"/>
<Property Name="LicenseUtilization" Type="Edm.String"/>
</ComplexType>
<Action Name="enableInterfaces" IsBound="true">
<Parameter Name="ParamInterface"
 Type="Collection(com.ca.nfa.odata.availableInterfaces)"/>
<Parameter Name="InterfaceIds" Type="Collection(Edm.Int64)" Nullable="false"/>
<ReturnType Type="Collection(com.ca.nfa.odata.availableInterfaces)"/>
</Action>
<Action Name="createInterfaceAggregation" IsBound="true">
<Parameter Name="ParamInterface" Type="Collection(com.ca.nfa.odata.interfaces)"/>
<Parameter Name="InterfaceIds" Type="Collection(Edm.Int64)" Nullable="false"/>
<Parameter Name="Name" Type="Edm.String" Nullable="false"/>
<Parameter Name="Description" Type="Edm.String"/>
<Parameter Name="IfType" Type="Edm.String" Nullable="false"/>
<Parameter Name="InSpeed" Type="Edm.Int64" Nullable="false"/>
<Parameter Name="OutSpeed" Type="Edm.Int64" Nullable="false"/>
<ReturnType Type="Collection(com.ca.nfa.odata.interfaces)"/>
</Action>
<Action Name="disableInterfaces" IsBound="true">
```

```
<Parameter Name="ParamInterface"
 Type="Collection(com.ca.nfa.odata.availableInterfaces)"/>
<Parameter Name="InterfaceIds" Type="Collection(Edm.Int64)" Nullable="false"/>
<ReturnType Type="Collection(com.ca.nfa.odata.availableInterfaces)"/>
</Action>
<Action Name="disableInterface" IsBound="true">
<Parameter Name="ParamInterface" Type="com.ca.nfa.odata.availableInterfaces"/>
<ReturnType Type="com.ca.nfa.odata.availableInterfaces"/>
</Action>
<Action Name="mergeInterfaces" IsBound="true">
<Parameter Name="ParamInterface" Type="Collection(com.ca.nfa.odata.interfaces)"/>
<Parameter Name="interface1" Type="Edm.Int64" Nullable="false"/>
<Parameter Name="interface2" Type="Edm.Int64" Nullable="false"/>
<Parameter Name="deleteSource" Type="Edm.Boolean" Nullable="false"/>
<Parameter Name="isOverLap" Type="Edm.Boolean" Nullable="false"/>
<ReturnType Type="Collection(com.ca.nfa.odata.interfaces)"/>
</Action>
<Action Name="disableRouter" IsBound="true">
<Parameter Name="ParamRouter" Type="com.ca.nfa.odata.router"/>
<ReturnType Type="com.ca.nfa.odata.router"/>
</Action>
<Action Name="editInterfaces" IsBound="true">
<Parameter Name="ParamInterface" Type="Collection(com.ca.nfa.odata.interfaces)"/>
<Parameter Name="InterfaceIds" Type="Collection(Edm.Int64)" Nullable="false"/>
<Parameter Name="IfType" Type="Edm.String"/>
<Parameter Name="InSpeed" Type="Edm.Int64"/>
<Parameter Name="OutSpeed" Type="Edm.Int64"/>
<Parameter Name="DomainId" Type="Edm.Int32"/>
<ReturnType Type="Collection(com.ca.nfa.odata.interfaces)"/>
</Action>
<Action Name="deleteInterfaces" IsBound="true">
<Parameter Name="ParamInterface" Type="Collection(com.ca.nfa.odata.interfaces)"/>
<Parameter Name="InterfaceIds" Type="Collection(Edm.Int64)" Nullable="false"/>
<Parameter Name="Schedule" Type="Edm.Boolean"/>
<ReturnType Type="Collection(com.ca.nfa.odata.interfaces)"/>
</Action>
<Action Name="editRouters" IsBound="true">
<Parameter Name="ParamRouter" Type="Collection(com.ca.nfa.odata.router)"/>
<Parameter Name="RouterIds" Type="Collection(Edm.Int64)" Nullable="false"/>
<Parameter Name="profileId" Type="Edm.Int32"/>
<Parameter Name="snmpVersion" Type="Edm.Int32"/>
<Parameter Name="snmpPort" Type="Edm.Int32"/>
<Parameter Name="templateId" Type="Edm.Int32"/>
<Parameter Name="domainID" Type="Edm.Int32"/>
<ReturnType Type="Collection(com.ca.nfa.odata.router)"/>
</Action>
```

```
<Action Name="enableInterface" IsBound="true">
<Parameter Name="ParamInterface" Type="com.ca.nfa.odata.availableInterfaces"/>
<ReturnType Type="com.ca.nfa.odata.availableInterfaces"/>
</Action>
<Action Name="deleteAvailableInterface" IsBound="true">
<Parameter Name="ParamInterface" Type="com.ca.nfa.odata.availableInterfaces"/>
<ReturnType Type="com.ca.nfa.odata.availableInterfaces"/>
</Action>
<Action Name="enableRouters" IsBound="true">
<Parameter Name="ParamRouter" Type="Collection(com.ca.nfa.odata.router)"/>
<Parameter Name="RouterIds" Type="Collection(Edm.Int64)" Nullable="false"/>
<ReturnType Type="Collection(com.ca.nfa.odata.router)"/>
</Action>
<Action Name="createCustomVirtualInterface" IsBound="true">
<Parameter Name="ParamInterface" Type="Collection(com.ca.nfa.odata.interfaces)"/>
<Parameter Name="InterfaceId" Type="Edm.Int64" Nullable="false"/>
<Parameter Name="Name" Type="Edm.String" Nullable="false"/>
<Parameter Name="Description" Type="Edm.String"/>
<Parameter Name="IfType" Type="Edm.String" Nullable="false"/>
<Parameter Name="InSpeed" Type="Edm.Int64" Nullable="false"/>
<Parameter Name="OutSpeed" Type="Edm.Int64" Nullable="false"/>
<Parameter Name="Subnet" Type="Collection(Edm.String)" Nullable="false"/>
<ReturnType Type="Collection(com.ca.nfa.odata.interfaces)"/>
</Action>
<Action Name="snmprefresh" IsBound="true">
<Parameter Name="ParamRouter" Type="com.ca.nfa.odata.router"/>
<ReturnType Type="com.ca.nfa.odata.csnmprefresh"/>
</Action>
<Action Name="deletePortPriorityRules" IsBound="true">
<Parameter Name="ParamInterface" Type="Collection(com.ca.nfa.odata.portPriorityRules)"/>
<Parameter Name="RuleIds" Type="Collection(Edm.Int64)" Nullable="false"/>
<ReturnType Type="Collection(com.ca.nfa.odata.portPriorityRules)"/>
</Action>
<Action Name="reset" IsBound="true">
<Parameter Name="ParamAS" Type="com.ca.nfa.odata.asNames"/>
<ReturnType Type="com.ca.nfa.odata.asNames"/>
</Action>
<Action Name="enableRouter" IsBound="true">
<Parameter Name="ParamRouter" Type="com.ca.nfa.odata.router"/>
<ReturnType Type="com.ca.nfa.odata.router"/>
</Action>
<Action Name="disableRouters" IsBound="true">
<Parameter Name="ParamRouter" Type="Collection(com.ca.nfa.odata.router)"/>
<Parameter Name="RouterIds" Type="Collection(Edm.Int64)" Nullable="false"/>
<ReturnType Type="Collection(com.ca.nfa.odata.router)"/>
</Action>
```

```
<Action Name="removeApplicationMappings" IsBound="true">
<Parameter Name="ParamApplicationMapping"
 Type="Collection(com.ca.nfa.odata.applicationMappings)"/>
<Parameter Name="ApplicationMappingIds" Type="Collection(Edm.Int64)" Nullable="false"/>
<ReturnType Type="Collection(com.ca.nfa.odata.applicationMappings)"/>
</Action>
<Action Name="deleteRouters" IsBound="true">
<Parameter Name="ParamRouter" Type="Collection(com.ca.nfa.odata.router)"/>
<Parameter Name="RouterIds" Type="Collection(Edm.Int64)" Nullable="false"/>
<Parameter Name="Schedule" Type="Edm.Boolean"/>
<ReturnType Type="Collection(com.ca.nfa.odata.router)"/>
</Action>
<Action Name="deleteReservedSeatingRules" IsBound="true">
<Parameter Name="ParamInterface"
 Type="Collection(com.ca.nfa.odata.reservedSeatingRules)"/>
<Parameter Name="RuleIds" Type="Collection(Edm.Int64)" Nullable="false"/>
<ReturnType Type="Collection(com.ca.nfa.odata.reservedSeatingRules)"/>
</Action>
<Function Name="ProductInfo">
<ReturnType Type="com.ca.nfa.odata.productInfo"/>
</Function>
<Function Name="SystemDefaults">
<ReturnType Type="com.ca.nfa.odata.limiters"/>
</Function>
<Function Name="PortTrafficStatus" IsBound="true">
<Parameter Name="ParamApp" Type="Collection(com.ca.nfa.odata.applicationMappings)"/>
<Parameter Name="port" Type="Edm.Int64" Nullable="false"/>
<ReturnType Type="com.ca.nfa.odata.portTrafficStatus"/>
</Function>
<EntityContainer Name="default">
<EntitySet Name="asNums" EntityType="com.ca.nfa.odata.asNums">
<NavigationPropertyBinding Path="asNextHop" Target="asNextHop"/>
</EntitySet>
<EntitySet Name="asNames" EntityType="com.ca.nfa.odata.asNames"/>
<EntitySet Name="harvesterHealth" EntityType="com.ca.nfa.odata.harvesterHealth"/>
<EntitySet Name="conversations" EntityType="com.ca.nfa.odata.conversations">
<NavigationPropertyBinding Path="protocols" Target="protocols"/>
</EntitySet>
<EntitySet Name="reservedSeatingRules"
 EntityType="com.ca.nfa.odata.reservedSeatingRules"/>
<EntitySet Name="harvesters" EntityType="com.ca.nfa.odata.harvesters">
<NavigationPropertyBinding Path="harvesterHealth" Target="harvesterHealth"/>
<NavigationPropertyBinding Path="flowStats" Target="flowStats"/>
</EntitySet>
<EntitySet Name="addressesHostNames" EntityType="com.ca.nfa.odata.addressesHostNames"/>
<EntitySet Name="interfaceTemplate" EntityType="com.ca.nfa.odata.interfaceTemplate"/>
```

```
<EntitySet Name="reporterHealth" EntityType="com.ca.nfa.odata.reporterHealth"/>
<EntitySet Name="protocols" EntityType="com.ca.nfa.odata.protocols">
<NavigationPropertyBinding Path="hosts" Target="hosts"/>
<NavigationPropertyBinding Path="conversations" Target="conversations"/>
</EntitySet>
<EntitySet Name="routers" EntityType="com.ca.nfa.odata.router"/>
<EntitySet Name="applicationSettings" EntityType="com.ca.nfa.odata.applicationSettings"/
>
<EntitySet Name="asNextHop" EntityType="com.ca.nfa.odata.asNextHop"/>
<EntitySet Name="interfaces" EntityType="com.ca.nfa.odata.interfaces">
<NavigationPropertyBinding Path="hosts" Target="hosts"/>
<NavigationPropertyBinding Path="utilization" Target="utilization"/>
<NavigationPropertyBinding Path="conversations" Target="conversations"/>
<NavigationPropertyBinding Path="protocols" Target="protocols"/>
<NavigationPropertyBinding Path="toss" Target="toss"/>
<NavigationPropertyBinding Path="asNums" Target="asNums"/>
<NavigationPropertyBinding Path="asNextHop" Target="asNextHop"/>
</EntitySet>
<EntitySet Name="snmpprofiles" EntityType="com.ca.nfa.odata.snmpprofiles"/>
<EntitySet Name="predefinedApplicationMappings"
 EntityType="com.ca.nfa.odata.predefinedApplicationMappings"/>
<EntitySet Name="hosts" EntityType="com.ca.nfa.odata.hosts">
<NavigationPropertyBinding Path="protocols" Target="protocols"/>
</EntitySet>
<EntitySet Name="portPriorityRules" EntityType="com.ca.nfa.odata.portPriorityRules"/>
<EntitySet Name="domains" EntityType="com.ca.nfa.odata.domains"/>
<EntitySet Name="utilization" EntityType="com.ca.nfa.odata.utilization"/>
<EntitySet Name="reporter" EntityType="com.ca.nfa.odata.reporter">
<NavigationPropertyBinding Path="reporterHealth" Target="reporterHealth"/>
</EntitySet>
<EntitySet Name="toss" EntityType="com.ca.nfa.odata.toss">
<NavigationPropertyBinding Path="hosts" Target="hosts"/>
<NavigationPropertyBinding Path="protocols" Target="protocols"/>
<NavigationPropertyBinding Path="conversations" Target="conversations"/>
</EntitySet>
<EntitySet Name="availableInterfaces" EntityType="com.ca.nfa.odata.availableInterfaces"/
>
<EntitySet Name="watchDogSettings" EntityType="com.ca.nfa.odata.watchDogSettings"/>
<EntitySet Name="applicationMappings" EntityType="com.ca.nfa.odata.applicationMappings"/
>
<EntitySet Name="flowStats" EntityType="com.ca.nfa.odata.flowStats"/>
<FunctionImport Name="ProductInfo" Function="com.ca.nfa.odata.ProductInfo"
 IncludeInServiceDocument="true"/>
<FunctionImport Name="SystemDefaults" Function="com.ca.nfa.odata.SystemDefaults"
 IncludeInServiceDocument="true"/>
</EntityContainer>
```

```
</Schema>
</edmx:DataServices>
</edmx:Edmx>
```

Entity API

The CA NFA OData API can be used for processing different service requests like, you can request the service for a list of entities, a single entity or for a selected property.You can define the `content-type` to retrieve the response in different format.

> **NOTE**
>
> : CA NFA OData API supports the content-type only in XML or JSON format. By default, the content type is JSON.

- **Example of content-type in JSON format**

```
http://127.0.0.1:8981/odata/api/routers?$format=application/
json;odata.metadata=minimal
```

- **Example of content-type in XML format**

```
http://127.0.0.1:8981/odata/api/routers?$format=application/xml
```

Entity Collection

This API displays collection of entities.

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/<EntitySet Name>
```

Method

GET

Parameters

The following table includes the mandatory parameter.

Parameters	Description
EntitySet Name	Indicates the type of entity set. For more information on list of valid EntitySet Name, see Metadata .

Sample Request

This following request illustrates entity collection API for Routers Entity type.

```
http://127.0.0.1:8981/odata/api/routers
```

Sample Response

The following response displays all records for Routers entity set.

```
{
  "@odata.context": "$metadata#Collection(com.ca.nfa.odata.router)",
  "value": [
    {
      "routerUpdatedOn": 1599147491,
      "deviceAlias": "10.0.9.11",
      "dnsExpireTime": 1599752503,
      "snmpMaxRows": 10,
      "templateId": 1,
      "dnsProxyAddress": "10.74.241.203",
      "deviceName": "10.0.9.11",
      "domainID": 1,
      "lifecyclestate": "ACTIVE",
      "lastHarvesterUpdate": 0,
      "sysDescr": "10.0.9.11",
      "sysUptime": 0,
      "snmpProxyAddress": "10.74.241.203",
      "snmpRetry": 3,
      "ifNumber": 0,
      "harvesterID": 2,
      "lastReboot": 1599562571,
      "routerName": null,
      "firstPollError": 1599562606,
      "sysName": "",
      "interfaceCount": 2,
      "snmpTimeout": 3,
      "ID": 1500001,
      "lastData": 0,
      "lastRefresh": 0,
      "snmpVersion": 2,
      "routerAddress": "10.0.9.11",
      "lastDiscovery": 0,
      "dnsLastLookupTime": 1599147703,
      "nextPollRetry": 0,
      "syncUpdateTime": 0,
      "profileId": 0,
      "tenantId": 8,
      "snmpPort": 161,
      "agentCount": 2,
      "flowStatus": "Red",
      "enabledInterfacesCount": 2,
      "profileName": null,
      "harvesterAddress": "10.74.241.203"
    },
    {
      "routerUpdatedOn": 1599147492,
      "deviceAlias": "10.0.9.14",
      "dnsExpireTime": 1599752503,
      "snmpMaxRows": 10,
      "templateId": 1,
      "dnsProxyAddress": "10.74.241.203",
      "deviceName": "10.0.9.14",
      "domainID": 1,
```

```
      "lifecyclestate": "ACTIVE",
      "lastHarvesterUpdate": 0,
      "sysDescr": "10.0.9.14",
      "sysUptime": 0,
      "snmpProxyAddress": "10.74.241.203",
      "snmpRetry": 3,
      "ifNumber": 0,
      "harvesterID": 2,
      "lastReboot": 1599562571,
      "routerName": null,
      "firstPollError": 1599562606,
      "sysName": "",
      "interfaceCount": 2,
      "snmpTimeout": 3,
      "ID": 1500002,
      "lastData": 0,
      "lastRefresh": 0,
      "snmpVersion": 2,
      "routerAddress": "10.0.9.14",
      "lastDiscovery": 0,
      "dnsLastLookupTime": 1599147703,
      "nextPollRetry": 0,
      "syncUpdateTime": 0,
      "profileId": 0,
      "tenantId": 8,
      "snmpPort": 161,
      "agentCount": 2,
      "flowStatus": "Red",
      "enabledInterfacesCount": 2,
      "profileName": null,
      "harvesterAddress": "10.74.241.203"
    }
  ]
}
```


Single Entity

This API displays the details of a specified entity, entity type with the property reference.

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/<EntitySet Name>(<PropertyRef Name value>)
```

Method

GET

Parameters

The following table includes the mandatory parameters.

Parameters	Description
EntitySet Name	Indicates the type of the entity set. For more information about list of valid EntitySet Name, see Metadata.
PropertyRef Name value	Indicates the value of the PropertyRef Name. For more information about list of valid PropertyRef Name, see Metadata.

Sample Request

This sample request displays the details of the router id 1500002.

```
http://127.0.0.1:8981/odata/api/routers(1500002)
```

Sample Response


```
{
"@odata.context": "$metadata#routers",
"ID": 1500002,
"routerAddress": "127.0.0.1",
"sysDescr": "127.0.0.1",
"sysName": "",
"deviceName": "127.0.0.1",
"deviceAlias": "127.0.0.1",
"sysUptime": 0,
"lastData": 0,
"lastReboot": 0,
"lastRefresh": 0,
"lastDiscovery": 0,
"lastHarvesterUpdate": 0,
"firstPollError": 0,
"nextPollRetry": 0,
"harvesterID": 2,
"profileId": 0,
"snmpVersion": 2,
"snmpPort": 161,
"snmpTimeout": 3,
"snmpRetry": 3,
"snmpMaxRows": 10,
"ifNumber": 0,
"interfaceCount": 10,
"agentCount": 10,
"dnsLastLookupTime": 1535344115,
"dnsExpireTime": 1535948915,
"syncUpdateTime": 0,
```

```
"routerName": null,
"routerUpdatedOn": 1526876265,
"templateId": 1,
"snmpProxyAddress": "127.0.0.1",
"dnsProxyAddress": "127.0.0.1",
"tenantId": 8,
"domainID": 1
}
```

Entity Property

This API displays the value of a specified property for a given entity type.

Resource URL

```
http://<nfa odata host>:<nfa odata port>/odata/api/<EntitySet Name>(<PropertyRef Name
 value>)/<Property Name>
```

Method

GET

Parameters

The following table includes the mandatory parameter.

Parameters	Description
EntitySet Name	Indicates the type of the entity set. For more information about list of valid EntitySet Name, see Metadata.
PropertyRef Name value	Indicates the value of the Property reference name. For more information about list of valid PropertyRef Name, see Metadata.
Property Name	Indicates the type of property. For more information about list of valid Property Name, see Metadata.

Sample Request

This sample request displays the value of the UpdatedOn property, of the interface id 291.

```
http://127.0.0.1:8981/odata/api/interfaces(4)/UpdatedOn
```

Sample Response

```
{
 "@odata.context": "../$metadata#interfaces/UpdatedOn",
 "value": 1539615618
```

```
}
```

Limiters

You can use the limiters to restrict the returned data or to control the behavior of API. The following limiters are available in the CA NFA API.

Limiter	Value	Description
defaultExpandTopLimit	10	Displays the top 10 values for a expanded NavigationProperty. To override this limiter, see the Override System Default Limiters section.
expandMaxTopLimit	2000	Indicates that the maximum top limit that can be expanded for a NavigationProperty is 2000.
requestTimeout	30	Indicates that the default request time out of an API is 30, beyond which the request displays the following error message "Request has been accepted. Request timed out due to no response from the server". To override this limiter, see the Override System Default Limiters section.
maxRequestTimeout	120	Indicates that the maximum request time-out of an API is 120 seconds.
requestThreadPoolCoreSize	10	Indicates that the default request thread pool core size is 10 seconds.
requestThreadPoolMaxSize	50	Indicates that the maximum request thread pool core size is 50 seconds.
defaultQueryTimeout	10	Indicates that the default single query time-out value is 10 seconds, beyond which the query timed-out error appears. To override this limiter, see the Override System Default Limiters section.
maxQueryTimeout	60	Indicates that the maximum query time-out value is 60 seconds.
defaultPageSize	10	Indicates that the default number of data that will be visible in a page is 10. To override this limiter, see the Override System Default Limiters section.
maxPageSize	1000	Indicates that the maximum number of data that will be visible in a page is 1000.
enableMultiTenancy	false	Indicates whether you can enable or disable MultiTenancy.

You can perform the following operations using Limiters:

Read System Default Limiters

The API displays the list of system default limiters and its values.

Resource URL

http://<nfa odata host>:<nfa odata port>/odata/api/SystemDefaults()

Method

GET

Sample Request

```
http://127.0.0.1:8981/odata/api/SystemDefaults()
```

Sample Response

```
{
 "@odata.context": "$metadata#com.ca.nfa.odata.limiters",
 "defaultExpandTopLimit": 10,
 "expandMaxTopLimit": 10,
 "requestTimeout": 30,
 "maxRequestTimeout": 120,
 "requestThreadPoolCoreSize": 10,
 "requestThreadPoolMaxSize": 50,
 "defaultQueryTimeout": 10,
 "maxQueryTimeout": 60,
 "defaultPageSize": 10,
 "maxPageSize": 1000
"enableMultiTenancy": false
}
```

Update System Default Limiters

You can update the system default limiter values at the application level for all APIs. Locate the com.ca.nfa.odata.ODataLimiters.cfg file, and update the system default limiter values as required.

> **NOTE**
>
> The updated values are visible in the system without restarting. Once the values are updated in the com.ca.nfa.odata.ODataLimiters.cfg file, it is considered as the new default values for all APIs.

Override System Default Limiters

You can override the following limiters:

- **defaultExpandTopLimit**
  You can override this parameter by using the $top system query parameter in $expand.
  Resource URIL
  *http://<nfa odata host>:<nfa odata port>*odata/api/*<EntityType Name>*
  $expand=protocols($top=*<Enter value lesser than expandMaxTopLimit>*)
  Sample Request
  This example overrides the defaultExpandTopLimit to 100.

```
http://127.0.0.1:8981/odata/api/interfaces?$expand=protocols($top=100)
```

- **requestTimeout**
  You can override this parameter by using the custom query parameter, timeout.
  Resource URL
  *http://<nfa odata host>:<nfa odata port>*odata/api/*<EntityType Name>*?timeout=*<Enter value lesser than maxRequestTimeout>*
  Sample Request
  This example overrides the requestTimeout to 40.
  ```
  http://127.0.0.1:8981/odata/api/interfaces?timeout=40
  ```

- defaultQueryTimeoutYou can override this parameter by using the custom query parameter, queryTimeOut.
  Resource URL
  *http://<nfa odata host>:<nfa odata port>*odata/api/*<EntityType Name>*?queryTimeOut=*<Enter value lesser than maxQueryTimeout>*
  Sample Request
  This example overrides the defaultQueryTimeout to 100 for an Interface entity.
  ```
  http://127.0.0.1:8981/odata/api/interfaces?queryTimeOut=100
  ```
- **defaultPageSize**
  You can override this parameter by entering the required value for the maxpagesize parameter in HTTP header.**Sample HTTP Header**
  This example overrides the defaultPageSize to 100.
  ```
  Prefer:odata.maxpagesize=100
  ```

Query Options

CA NFA OData service supports querying collections of entities, complex type instances, and primitive values. You can apply query option to any selected property by appending a semi-colon to the list of query options. The query options are enclosed within parentheses.

The various categories of query options available in DX NetOps APIs are:

Custom Query Options

Custom query option provides additional information to the CA NFA OData service by including them in the URL query string. Custom query options must not begin with a $ or @ character.

The list of supported custom query parameters are:

Custom Query Parameter	Description
startTime	UNIX timestamp that indicates the beginning time for NFA metric data. If you do not specify the start time, the start time is one hour before the specified end time.
endTime	UNIX timestamp that indicates the end time for NFA metric data. If you do not specify the end time, the end time is one hour from the specified start time.
resolution	Specifies the target data. Valid values are, **min1** or **min15**, which indicates 1 minute or 15 minute respectively.

queryTimeOut	Overrides the defaultQueryTimeout system default limiter for query execution that is present in CA NFA OData limiters. The values are entered in seconds.
timeout	Overrides the requestTimeout system default limiter for HTTP Request that is present in CA NFA OData limiters. The values are entered in seconds.

NOTE

: We recommend you to enter the startTime and endTime. But, If the startTime and endTime are not specified, by default, the UNIX timestamp is last one hour when the query was hit.

Request URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/<Entity Type>(<PropertyRef Name value>)/<NavigationProperty>?<Custom Query parameter>=<Value of Custom Query parameter>
```

Method

GET

Parameters

Parameters	Description
Entity Type	Indicates the entity type. For more information about the list of valid Entity Type, see Metadata.
PropertyRef Name value	Indicates the value of the PropertyRef Name. For more information about the list of PropertyRef Name, see Metadata.
Custom Query parameter	Indicates the type of Custom Query parameter. The list of supported custom query parameters are: • startTime • endTime • resolution • queryTimeOut • timeout

Examples

The list of examples for Custom Query Options are as follows:

Example for startTime

Sample Request

```
http://127.0.0.1:8981/odata/api/interfaces(3)/hosts?startTime=1541894400
```

Sample Response

```
{

"@odata.context": "$metadata#hosts",
 "value": [
 {
 "router": "127.0.0.1",
 "interface": 2,
 "timestamp": 1541894400,
 "protocol": 0,
 "host": "127.0.0.1",
 "inoctets": 1650000,
 "outoctets": 1650000
 },
 {
 "router": "127.0.0.1",
 "interface": 2,
 "timestamp": 1541894400,
 "protocol": 0,
 "host": "127.0.0.1",
 "inoctets": 1650000,
 "outoctets": 1650000
 },
 {
 "router": "127.0.0.1",
 "interface": 2,
 "timestamp": 1541894400,
 "protocol": 393239,
 "host": "127.0.0.1",
 "inoctets": 1650000,
 "outoctets": 1650000
 },
 {
 "router": "127.0.0.1",
 "interface": 2,
 "timestamp": 1541894400,
 "protocol": 393239,
 "host": "127.0.0.1",
 "inoctets": 1650000,
 "outoctets": 1650000
 },
 {
 "router": "127.0.0.1",
 "interface": 2,
 "timestamp": 1541895300,
 "protocol": 0,
 "host": "127.0.0.1",
 "inoctets": 1650000,
```

```
"outoctets": 1650000
},
{
"router": "127.0.0.1",
"interface": 2,
"timestamp": 1541895300,
"protocol": 0,
"host": "127.0.0.1",
"inoctets": 1650000,
"outoctets": 1650000
},
{
"router": "127.0.0.1",
"interface": 2,
"timestamp": 1541895300,
"protocol": 393239,
"host": "127.0.0.1",
"inoctets": 1650000,
"outoctets": 1650000
},
{
"router": "127.0.0.1",
"interface": 2,
"timestamp": 1541895300,
"protocol": 393239,
"host": "127.0.0.1",
"inoctets": 1650000,
"outoctets": 1650000
},
{
"router": "127.0.0.1",
"interface": 2,
"timestamp": 1541896200,
"protocol": 0,
"host": "127.0.0.1",
"inoctets": 1650000,
"outoctets": 1650000
},
{
"router": "127.0.0.1",
"interface": 2,
"timestamp": 1541896200,
"protocol": 0,
"host": "127.0.0.1",
"inoctets": 1650000,
"outoctets": 1650000
```

```
 }
 ],
 "@odata.nextLink": "http://127.0.0.1:8981/odata/api/interfaces(3)/hosts?
startTime=1541894400&$skiptoken=10"
}
```

Example for endTime

Sample Request

```
http://127.0.0.1:8981/odata/api/interfaces(3)/hosts?endTime=1541808000
```

Sample Response

```
{
 "@odata.context": "$metadata#hosts",
 "value": [
 {
 "router": "127.0.0.1",
 "interface": 2,
 "timestamp": 1541804400,
 "protocol": 0,
 "host": "127.0.0.1",
 "inoctets": 1650000,
 "outoctets": 1650000
 },
 {
 "router": "127.0.0.1",
 "interface": 2,
 "timestamp": 1541804400,
 "protocol": 0,
 "host": "127.0.0.1",
 "inoctets": 1650000,
 "outoctets": 1650000
 },
 {
 "router": "127.0.0.1",
 "interface": 2,
 "timestamp": 1541804400,
 "protocol": 393239,
 "host": "127.0.0.1",
```

```
"inoctets": 1650000,
"outoctets": 1650000
},
{
"router": "127.0.0.1",
"interface": 2,
"timestamp": 1541804400,
"protocol": 393239,
"host": "127.0.0.1",
"inoctets": 1650000,
"outoctets": 1650000
},
{
"router": "127.0.0.1",
"interface": 2,
"timestamp": 1541805300,
"protocol": 0,
"host": "127.0.0.1",
"inoctets": 1650000,
"outoctets": 1650000
},
{
"router": "127.0.0.1",
"interface": 2,
"timestamp": 1541805300,
"protocol": 0,
"host": "127.0.0.1",
"inoctets": 1650000,
"outoctets": 1650000
},
{
"router": "127.0.0.1",
"interface": 2,
"timestamp": 1541805300,
"protocol": 393239,
"host": "127.0.0.1",
"inoctets": 1650000,
"outoctets": 1650000
},
{
"router": "127.0.0.1",
"interface": 2,
"timestamp": 1541805300,
"protocol": 393239,
"host": "127.0.0.1",
"inoctets": 1650000,
```

```
 "outoctets": 1650000
 },
 {
 "router": "127.0.0.1",
 "interface": 2,
 "timestamp": 1541806200,
 "protocol": 0,
 "host": "127.0.0.1",
 "inoctets": 1650000,
 "outoctets": 1650000
 },
 {
 "router": "127.0.0.1",
 "interface": 2,
 "timestamp": 1541806200,
 "protocol": 0,
 "host": "127.0.0.1",
 "inoctets": 1650000,
 "outoctets": 1650000
 }
 ],
 "@odata.nextLink": "http://127.0.0.1:8981/odata/api/interfaces(3)/hosts?
endTime=1541808000&$skiptoken=10"
}
```


Example for startTime and endTime

Sample Request

```
http://127.0.0.1:8981/odata/api/interfaces(10)/hosts?
startTime=1541894400&endTime=1541808000
```

Example without startTime and endTime

Sample Request

```
http://127.0.0.1:8981/odata/api/interfaces(10)/hosts?resolution=min15
```

Example for resolution

Sample Request

```
http://127.0.0.1:8981/odata/api/routers(1500001)/deviceName?resolution=min2
```

Sample Response


```
{
 "@odata.context": "../$metadata#routers/deviceName",
```

```
 "value": "127.0.0.1"
}
```


Example for queryTimeOut

Sample Request

```
http://127.0.0.1:8981/odata/api/interfaces(291)/hosts?queryTimeOut=
```

Example for timeout

Sample Request

```
http://127.0.0.1:8981/odata/api/interfaces(291)/hosts?timeout
```

Sample Request

```
http://127.0.0.1:8981/odata/api/interfaces(291)/hosts?
&startTime=1526891820&endTime=1526896858&resolution=min15
```

Sample Response


```
{
 "@odata.context": "$metadata#hosts",
 "value": [
 {
 "router": "127.0.0.1",
 "interface": 9,
 "timestamp": 1526892300,
 "protocol": 0,
 "host": "127.0.0.1",
 "inoctets": 1200000,
 "outoctets": 12000000
 },
 {
 "router": "127.0.0.1",
 "interface": 9,
 "timestamp": 1526892300,
 "protocol": 0,
 "host": "127.0.0.1",
 "inoctets": 12000000,
 "outoctets": 1200000
 },
 {
 "router": "127.0.0.1",
 "interface": 9,
 "timestamp": 1526892300,
 "protocol": 0,
 "host": "127.0.0.1",
 "inoctets": 795000,
```

```
  "outoctets": 7950000
  },
  {
  "router": "127.0.0.1",
  "interface": 9,
  "timestamp": 1526892300,
  "protocol": 0,
  "host": "127.0.0.1",
  "inoctets": 7950000,
  "outoctets": 795000
  }
  ]
}
```

System Query Options

DX NetOps OData API uses System Query Options to control the limit, and order of the data returned for a resource. System Query Options can be specified in the query string parameters for the resource that is identified by the URL. The names of all system query options are optionally prefixed with a dollar (`$`) character.

The following rules apply for GET, PATCH, and PUT requests:

- Resource paths indicating a single entity, a complex type instance, a collection of entities, or a collection of complex type instances allow `$expand` and `$select` .
- Resource paths indicating a collection, allows `$filter` , `$count, $orderby, $skip` , and `$top` .
- Resource paths ending in `/$count` allows `$filter` .
- Resource paths not ending in `/$count` or allows `$format` .

For, a URL with POST request, the return type determines the applicable system query options that a service may support. The rules are same as GET request.

POST requests to an entity set follow the same rules as GET requests that return a single entity.

> **NOTE**
>
> : Do not apply system query option to a DELETE request.
>
> CA NFA OData API rejects the request that contains any unsupported system query options.

An OData service may support some or all of the system query options defined. If a data service does not support a system query option, it must reject any request that contains the unsupported option.

The same system query option must not be specified more than once for any resource, even if it is prefixed with a $.

The list of valid system query options are:

Select

The `$select` system query option enables you to request the CA NFA OData service to return a specific property. The service returns a specified content with any expanded available navigation or stream properties. Also, it may return additional information.
The value of the $select query option is a comma-separated list of properties, qualified action names, and qualified function names. You can also request all declared properties and dynamic structural properties using a star (*).
When a $select query option is not specified, the service may return full set of properties or default set of properties.

NOTE

The default set of properties must include all key properties.

Resource URL

```
http://<nfa odata host>:<nfa odata port>/odata/api/<EntityType Name>?$select=<Property Name>
```

TIP

To display multiple properties for a given entity, separate each property name by a comma.

Method

GET

Parameters

The following table includes the mandatory parameter.

Parameters	Description
EntityType Name	Indicates the type of entity. For more information about list of valid EntityType Name, see Metadata.
Property Name	Indicates the property type of an Entity. For more information about list of Property Name, see Metadata.

Sample Request

This sample request displays all the IfType and ID of Interface entity.

```
http:/127.0.0.1:8981/odata/api/interfaces?$select=IfType,ID
```

Sample Response

```
{
 "@odata.context": "$metadata#interfaces(ID,IfType)",
 "value": [
 {
 "ID": 1,
 "IfType": "WAN"
 },
 {
 "ID": 2,
 "IfType": "WAN"
 },
 {
 "ID": 3,
 "IfType": "WAN"
 },
 {
 "ID": 4,
 "IfType": "WAN"
 }
```

```
]
}
```

Sample Request

This sample request displays all properties of Interface entity.

```
http://127.0.0.1:8981//odata/interfaces?$select=*
```

Sample Response

```
{
 "@odata.context": "$metadata#interfaces(*)",
 "value": [
 {
 "ID": 283,
 "AgentType": "Physical",
 "RouterAddress": "127.0.0.1",
 "Description": "",
 "InSpeed": 0,
 "OutSpeed": 0,
 "IfIndex": 1,
 "PersistentIfIndex": 1,
 "IfType": "WAN",
 "UpdatedOn": 1526877024,
 "Enabled": "N",
 "LastData": 1527153321,
 "HarvesterAddress": "127.0.0.1"
 },
 {
 "ID": 284,
 "AgentType": "Physical",
 "RouterAddress": "127.0.0.11",
 "Description": "",
 "InSpeed": 0,
 "OutSpeed": 0,
 "IfIndex": 2,
 "PersistentIfIndex": 2,
 "IfType": "WAN",
 "UpdatedOn": 1526877024,
 "Enabled": "N",
 "LastData": 1527153321,
 "HarvesterAddress": "127.0.0.1"
 },
 {
 "ID": 287,
 "AgentType": "Physical",
 "RouterAddress": "127.0.0.1",
 "Description": "",
```

```
  "InSpeed": 0,
  "OutSpeed": 0,
  "IfIndex": 3,
  "PersistentIfIndex": 5,
  "IfType": "WAN",
  "UpdatedOn": 1526877024,
  "Enabled": "N",
  "LastData": 1527153321,
  "HarvesterAddress": "127.0.0.1"
  },
  {
  "ID": 291,
  "AgentType": "Physical",
  "RouterAddress": "127.0.0.1",
  "Description": "",
  "InSpeed": 0,
  "OutSpeed": 0,
  "IfIndex": 4,
  "PersistentIfIndex": 9,
  "IfType": "WAN",
  "UpdatedOn": 1526877024,
  "Enabled": "N",
  "LastData": 1527153321,
  "HarvesterAddress": "127.0.0.1"
  }
  ]
 }
```


Filters

The $filter system query option allows clients to filter a collection of resources that are addressed by a request URL. The expression that is specified with $filter is evaluated for each resource in the collection.

The items for which the expression is set are included in the response and the responses are omitted when:

- The resources for which the expression is set to false or to null.
- The reference properties are unavailable due to permissions.

Note: The operators **Equal (eq)** and **Not Equal (ne)** support only **numeric** values. When you need to filter based on **string** values use the **contains** operator.

The list of valid filter operators are as follows:

Comparison Operator

Operator	Description	Example
eq	Equal	InSpeed eq 100
ne	Not equal	InSpeed ne 100
gt	Greater than	InSpeed gt 100
ge	Greater than or equal	InSpeed ge 100

lt	Less than	InSpeed lt 100
le	Less than or equal	InSpeed le 100

Logical operators

Operator	Description	Example
and	Logical AND	InSpeed gt 100 and OutSpeed lt 1000
or	Logical OR	InSpeed gt 100 or OutSpeed lt 1000
not	Logical negation	not InSpeed gt 100

Arithmetic Operators

Operator	Description	Example
add	Addition	Price add 5 gt 10
sub	Subtraction	Price sub 5 gt 10
mul	Multiplication	Price mul 2 gt 2000
div	Division	Price div 2 gt 4
mod	Modulo	Price mod 2 eq 0

Grouping Operators

Operator	Description	Example
( )	Precedence grouping	(InSpeed lt 100 InSpeed le 100) ge 100

Arithmetic Operators

Operator	Example
contains	contains(CompanyName,'freds')
endswith	endswith(deviceName,'1')
startswith	startswith(AgentType,'Alfr')
length	length(PersistentIfIndex) eq 19
indexof	indexof(CompanyName,'lfreds') eq 1
substring	substring(CompanyName,1) eq 'lfreds Futterkiste'
tolower	tolower(AgentType) eq 'virtual'
toupper	toupper(AgentType) eq 'VIRTUAL'
concat	concat('AgentType', RouterAddress) eq 'AgentType, RouterAddress'

Math Function

Operator	Example
round	round(Freight) eq 32
floor	floor(Freight) eq 32
ceiling	ceiling(Freight) eq 33

Resource URL

```
http://<nfa odata host>:<nfa odata port>/odata/api/<EntityType Name>?$filter=<Property
 Name><space><Filter Operator><space><Property Name>(<ProperyRef Name value>)
```

Method

GET

Parameters

The following table includes the mandatory parameter.

Parameters	Description
EntityType Name	Specifies the name of the entity type.
Property Name	Indicates the property type of an Entity.
Filter Operator.	Specifies the valid list of filter operator.
Property Name	Indicates the Property type.
ProperyRef Name value	Specifies the value of property reference.

Sample Request

This example uses the eq filter option to display the router Id 150002**.**

```
http://127.0.0.1:8981/odata/api/routers?$filter=ID eq (1500002)
```

Sample Response


```
 {
"@odata.context": "$metadata#routers",
"value": [
{
"ID": 1500002,
"routerAddress": "127.0.0.1",
"sysDescr": "127.0.0.1",
"sysName": "",
"deviceName": "127.0.0.1",
"deviceAlias": "127.0.0.1",
"sysUptime": 0,
"lastData": 0,
"lastReboot": 0,
"lastRefresh": 0,
"lastDiscovery": 0,
"lastHarvesterUpdate": 0,
"firstPollError": 0,
"nextPollRetry": 0,
"harvesterID": 2,
"profileId": 0,
"snmpVersion": 2,
"snmpPort": 161,
"snmpTimeout": 3,
"snmpRetry": 3,
"snmpMaxRows": 10,
"ifNumber": 0,
"interfaceCount": 10,
```

```
 "agentCount": 10,
 "dnsLastLookupTime": 1535948950,
 "dnsExpireTime": 1536553750,
 "syncUpdateTime": 0,
 "routerName": null,
 "routerUpdatedOn": 1526876265,
 "templateId": 1,
 "snmpProxyAddress": "127.0.0.1",
 "dnsProxyAddress": "127.0.0.1",
 "tenantId": 8,
 "domainID": 1
 }
 ]
}
```


Expansion

The expand system query option indicates the related entities and stream values that must be represented inline. The CA NFA OData service must return the specified content, and may return additional information.

Simple Expansion

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/<EntityType Name>(<PropertyRef Name Value>)?$expand=<EntityType Name>
```

Method

GET

Parameters

The following table includes the mandatory parameter.

Parameters	Description
EntityType Name	Indicates the type of entity. For more information about list of valid EntityType Name, see Metadata.
PropertyRef Name	Indicates the property type of an entity. For more information about list of valid EntityType Name, see Metadata.

Sample Request

This sample API expands the details of host entity for interface ID 3.

```
http://127.0.0.1:8981/odata/api/interfaces(109)?$expand=hosts
```

Sample Response

```
{
"@odata.context": "$metadata#interfaces(hosts())",
 "ID": 109,
 "AgentType": "Physical",
 "RouterAddress": "127.0.0.1",
 "IfIndex": 1,
 "PersistentIfIndex": 1,
 "RouterId": "1500001",
 "RouterName": "127.0.0.1",
 "Name": "Interface 1",
 "Description": "",
 "IfType": "Token Ring",
 "InSpeed": 101,
 "OutSpeed": 121,
 "Linked": "N",
 "LastData": 1553473820,
 "UpdatedOn": 1553389258,
 "HarvesterAddress": "127.0.0.1",
 "DomainId": "1",
 "Enabled": "Y",
 "hosts": [
 {
 "protocol": 0,
 "inoctets": 1500000,
 "host": "127.0.0.1",
 "outoctets": 150000,
 "interface": 1,
 "router": "127.0.0.1",
 "timestamp": 1553442300
 },
 {
 "protocol": 0,
 "inoctets": 150000,
 "host": "127.0.0.1",
 "outoctets": 1500000,
 "interface": 1,
 "router": "127.0.0.1",
 "timestamp": 1553442300
 },
 {
 "protocol": 65536,
 "inoctets": 1500000,
 "host": "127.0.0.1",
 "outoctets": 150000,
 "interface": 1,
 "router": "10.0.9.13",
```

```
 "timestamp": 1553442300
 }
 ]
}
```

You can also use expand query with:

- System query options like select, top, skip, count, and filter.
- Custom query options like startTime, endTime, resolution, queryTimeOut, and timeout.

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/<EntityType Name>(<PropertyRef Name
 Value>)?$expand=<EntityType Name>&<list of other system or custom query operator>
```

Method

GET

Parameters

The following table includes the mandatory parameter.

Parameters	Description
EntityType Name	Indicates the type of entity. For more information about list of valid EntityType Name, see Metadata.
PropertyRef Name	Indicates the Property Reference name of an entity. For more information about list of valid PropertyRef Name, see Metadata.

Sample Request

```
http://127.0.0.1:8981/odata/api/interfaces(110)?$expand=conversations($top=8;$skip=2;
$count=true;$filter=protocol eq
 0),protocols($top=2)&startTime=1526891820&endTime=1553445000
```

Sample Response

```
{
 "@odata.context": "$metadata#interfaces(protocols(),conversations())",
 "ID": 110,
 "AgentType": "Physical",
 "RouterAddress": "127.0.0.1",
 "IfIndex": 2,
 "PersistentIfIndex": 2,
 "RouterId": "1500001",
```

```
"RouterName": "127.0.0.1",
"Name": "Interface 2",
"Description": "",
"IfType": "Token Ring",
"InSpeed": 101,
"OutSpeed": 121,
"Linked": "N",
"LastData": 1553476521,
"UpdatedOn": 1553389258,
"HarvesterAddress": "127.0.0.1",
"DomainId": "1",
"Enabled": "Y",
"protocols": [
{
"protocol": 0,
"inoctets": 800000,
"outoctets": 520000,
"interface": 2,
"router": "127.0.0.1",
"timestamp": 1553389200
},
{
"protocol": 4,
"inoctets": 800000,
"outoctets": 520000,
"interface": 2,
"router": "127.0.0.1",
"timestamp": 1553389200
}
],
"conversations@odata.count": 6,
"conversations": [
{
"protocol": 0,
"inoctets": 150000,
"outoctets": 1500000,
"SrcHost": "1.1.1.10",
"interface": 2,
"DestHost": "127.0.0.1",
"router": "127.0.0.1",
"timestamp": 1553390100
},
{
"protocol": 0,
"inoctets": 0,
"outoctets": 675000,
```

```
"SrcHost": "127.0.0.1",
"interface": 2,
"DestHost": "255.255.255.255",
"router": "127.0.0.1",
"timestamp": 1553390100
},
{
"protocol": 0,
"inoctets": 0,
"outoctets": 150000,
"SrcHost": "127.0.0.1",
"interface": 2,
"DestHost": "255.255.255.255",
"router": "127.0.0.1",
"timestamp": 1553390100
},
{
"protocol": 0,
"inoctets": 150000,
"outoctets": 1500000,
"SrcHost": "1.1.1.10",
"interface": 2,
"DestHost": "127.0.0.1",
"router": "127.0.0.1",
"timestamp": 1553391000
},
{
"protocol": 0,
"inoctets": 0,
"outoctets": 675000,
"SrcHost": "127.0.0.1",
"interface": 2,
"DestHost": "255.255.255.255",
"router": "127.0.0.1",
"timestamp": 1553391000
},
{
"protocol": 0,
"inoctets": 0,
"outoctets": 150000,
"SrcHost": "127.0.0.1",
"interface": 2,
"DestHost": "255.255.255.255",
"router": "127.0.0.1",
"timestamp": 1553391000
}
```

```
 ]
}
```

Expansion with Utilization

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/interfaces?$expand=utilization&
$apply=groupby((utilization/protocol),aggregate(Utilization))&startTime=<Start
 time>&endTime=<end time>&resolution< MIN1 or Min15>
```

Method

GET

Parameters

The following table includes the optional parameter.

Parameters	Description
startTime	UNIX timestamp that indicates the beginning time for NFA metric data. If you do not specify the start time, the start time is one hour before the specified end time.
endTime	UNIX timestamp that indicates the end time for NFA metric data. If you do not specify the end time, the end time is one hour from the specified start time.
resolution	Specifies the target data. Valid values are, **min1** or **min15**, which indicates 1 minute or 15 minute respectively. By default, the value is 15 minutes.

Sample request

```
http://127.0.0.1:8981/odata/api/interfaces?$expand=utilization&
$apply=groupby((utilization/
protocol),aggregate(Utilization))&startTime=1541480880&endTime=1541480980&resolution=MIN1
```

Sample Response

This sample response expands the utilization entity type of an Interface entity, and then groups the utilization data based on the protocol, start time, end time.

```
{
"@odata.context": "$metadata#interfaces(utilization())",
"value": [
{
"ID": 229,
"AgentType": "Physical",
```

```
"RouterAddress": "127.0.0.1",
"IfIndex": 8,
"PersistentIfIndex": 10,
"RouterId": "1500001",
"RouterName": "127.0.0.1",
"Name": "Interface 8",
"Description": "",
"IfType": "WAN",
"InSpeed": 100,
"OutSpeed": 200,
"Linked": "N",
"LastData": 1543301125,
"UpdatedOn": 1539340223,
"HarvesterAddress": "127.0.0.1",
"DomainId": "1",
"Enabled": "Y",
"utilization": []
},
{
"ID": 228,
"AgentType": "Physical",
"RouterAddress": "127.0.0.1",
"IfIndex": 7,
"PersistentIfIndex": 9,
"RouterId": "1500001",
"RouterName": "127.0.0.1",
"Name": "Interface 7",
"Description": "",
"IfType": "WAN",
"InSpeed": 100,
"OutSpeed": 200,
"Linked": "N",
"LastData": 1543301125,
"UpdatedOn": 1539340223,
"HarvesterAddress": "127.0.0.1",
"DomainId": "1",
"Enabled": "Y",
"utilization": [
{
"protocol": 1769472,
"UtcTimeStamp": "1541480400",
"Total": "6.52",
"In": "9.78",
"Out": "4.89",
"interface": 9,
"router": "127.0.0.1"
```

```
},
{
"protocol": 1769472,
"UtcTimeStamp": "1541484000",
"Total": "6.52",
"In": "9.78",
"Out": "4.89",
"interface": 9,
"router": "127.0.0.1"
}
]
},
{
"ID": 223,
"AgentType": "Physical",
"RouterAddress": "127.0.0.1",
"IfIndex": 3,
"PersistentIfIndex": 4,
"RouterId": "1500001",
"RouterName": "127.0.0.1",
"Name": "Interface 3",
"Description": "",
"IfType": "WAN",
"InSpeed": 100,
"OutSpeed": 200,
"Linked": "N",
"LastData": 1543301125,
"UpdatedOn": 1539340223,
"HarvesterAddress": "127.0.0.1",
"DomainId": "1",
"Enabled": "Y",
"utilization": [
{
"protocol": 0,
"UtcTimeStamp": "1541480400",
"Total": "57.68",
"In": "85.91",
"Out": "43.57",
"interface": 4,
"router": "127.0.0.1"
},
{
"protocol": 0,
"UtcTimeStamp": "1541484000",
"Total": "57.68",
"In": "85.91",
```

```
"Out": "43.57",
"interface": 4,
"router": "127.0.0.1"
},
{
"protocol": 393236,
"UtcTimeStamp": "1541480400",
"Total": "3.26",
"In": "4.89",
"Out": "2.44",
"interface": 4,
"router": "127.0.0.1"
},
{
"protocol": 393236,
"UtcTimeStamp": "1541484000",
"Total": "3.26",
"In": "4.89",
"Out": "2.44",
"interface": 4,
"router": "127.0.0.1"
},
{
"protocol": 393239,
"UtcTimeStamp": "1541480400",
"Total": "3.75",
"In": "5.62",
"Out": "2.81",
"interface": 4,
"router": "127.0.0.1"
},
{
"protocol": 393239,
"UtcTimeStamp": "1541484000",
"Total": "3.75",
"In": "5.62",
"Out": "2.81",
"interface": 4,
"router": "127.0.0.1"
},
{
"protocol": 393241,
"UtcTimeStamp": "1541480400",
"Total": "7.41",
"In": "11.11",
"Out": "5.56",
```

```
"interface": 4,
"router": "127.0.0.1"
},
{
"protocol": 393241,
"UtcTimeStamp": "1541484000",
"Total": "7.41",
"In": "11.11",
"Out": "5.56",
"interface": 4,
"router": "127.0.0.1"
},
{
"protocol": 393296,
"UtcTimeStamp": "1541480400",
"Total": "13.04",
"In": "19.56",
"Out": "9.78",
"interface": 4,
"router": "127.0.0.1"
},
{
"protocol": 393296,
"UtcTimeStamp": "1541484000",
"Total": "13.04",
"In": "19.56",
"Out": "9.78",
"interface": 4,
"router": "127.0.0.1"
},
{
"protocol": 1114165,
"UtcTimeStamp": "1541480400",
"Total": "8.64",
"In": "12.96",
"Out": "6.48",
"interface": 4,
"router": "127.0.0.1"
},
{
"protocol": 1114165,
"UtcTimeStamp": "1541484000",
"Total": "8.64",
"In": "12.96",
"Out": "6.48",
"interface": 4,
```

```
"router": "127.0.0.1"
}
]
},
{
"ID": 221,
"AgentType": "Physical",
"RouterAddress": "127.0.0.1",
"IfIndex": 5,
"PersistentIfIndex": 2,
"RouterId": "1500001",
"RouterName": "127.0.0.1",
"Name": "Interface 5",
"Description": "",
"IfType": "WAN",
"InSpeed": 100,
"OutSpeed": 200,
"Linked": "N",
"LastData": 1543301125,
"UpdatedOn": 1539340223,
"HarvesterAddress": "127.0.0.1",
"DomainId": "1",
"Enabled": "Y",
"utilization": [
{
"protocol": 1114142,
"UtcTimeStamp": "1541480400",
"Total": "4.89",
"In": "7.33",
"Out": "3.67",
"interface": 2,
"router": "127.0.0.1"
},
{
"protocol": 1114142,
"UtcTimeStamp": "1541484000",
"Total": "4.89",
"In": "7.33",
"Out": "3.67",
"interface": 2,
"router": "127.0.0.1"
}
]
},
```

```
"@odata.nextLink": "http://localhost:8981/odata/api/interfaces?$expand=utilization&
$apply=groupby%28%28utilization/protocol%29,aggregate%28Utilization
%29%29&startTime=1541480880&endTime=1541480980&resolution=MIN1&$skiptoken=10"
}
```

Navigation

The navigation property is similar to the DX NetOps console user interface. The navigation property name is optionally followed by a `/$count` path segment, and optionally a parenthesized set of expand options like, filtering, sorting, selecting, paging, or expanding the related entities.

The lists of options available in Navigation are:

Simple Navigation

NOTE

The action that you perform on NFA API is the same as in NFA console. The list of possible scenarios for Navigation in NFA console are as shown in the table:

For this Interface (entity name), display	Navigation
the interface selected from the previous column>, show me	Hosts (protocols)
Conversations	

Protocols (toss)

Hosts

Conversations

Protocols (hosts)

Protocols Conversations (conversations)

asNextHop

Resource URL

```
http://<nfa odata host>:<nfa odata port>/odata/api/<EntityType Name>(<PropertyRef
Name>)/<EntityType Name>(<PropertyRef Name>)
```

Method

GET

Parameters

The following table includes the mandatory parameter.

Parameters	Description
EntityType Name	Indicates the name of the entity type. For more information about the list of valid EntityType Name, see Metadata.
PropertyRef Name	Indicates the property reference name of the entity. For more information about the list of valid PropertyRef Name, see Metadata.

Sample Request

This example illustrates navigation from Interfaces to Hosts.

```
http://127.0.0.1:8981/odata/api/interfaces(9)/hosts?
&startTime=1526891820&endTime=1526896858&resolution=min15
```

Sample Response


```
{
"@odata.context": "$metadata#hosts",
"value": [
{
"router": "127.0.0.1",
"interface": 9,
"timestamp": 1526892300,
"protocol": 0,
"host": "127.0.0.1",
"inoctets": 1200000,
"outoctets": 12000000
},
{
"router": "127.0.0.1",
"interface": 9,
"timestamp": 1526892300,
"protocol": 0,
"host": "127.0.0.1",
"inoctets": 12000000,
"outoctets": 1200000
},
{
"router": "127.0.0.1",
"interface": 9,
```

```
 "timestamp": 1526892300,
 "protocol": 0,
 "host": "127.0.0.1",
 "inoctets": 795000,
 "outoctets": 7950000
 },
 {
 "router": "127.0.0.1",
 "interface": 9,
 "timestamp": 1526892300,
 "protocol": 0,
 "host": "127.0.0.1",
 "inoctets": 7950000,
 "outoctets": 795000
 }
 ]
}
```


Navigation with Host Filter

Sample Request

```
 http://127.0.0.1:8981/odata/api/interfaces(79)/protocols(0)/hosts?
$top=10&amp;startTime=1536754500 &amp;endTime=1537289100 &amp;
$filter=contains(host,'8.8.8.80')
```

Navigation with Conversation Filter

Sample Request

```
 http://127.0.0.1:8981/odata/api/interfaces(79)/conversations?startTime=1536754500
 &amp;endTime=1537289100 &amp;$filter=contains(DestHost,'80.80.80.80' )
```

Navigation With Conversation to Protocols

Sample Request

```
 http://127.0.0.1:8981/odata/api/interfaces(79)/
conversations(SrcHost='127.0.0.1',DestHost='127.0.0.1')/protocols?$top=10&amp;
$count=true&amp;startTime=1536754500 &amp;endTime=1537289100
```

Navigation with asNum

Sample Request

```
 http://127.0.0.1:8981/odata/api/interfaces(79)/asNums?
&startTime=1536754500&endTime=1537289100&resolution=min15
```

Navigation with asNextHop

Sample Request

```
 http://127.0.0.1:8981/odata/api/interfaces(79)/asNextHop?
&startTime=1536754500&endTime=1537289100&resolution=min15&
$filter=contains(nexthop,'172.168.0.1')
```

Navigation with asNum to asNextHop

Sample Request

```
 http://127.0.0.1:8981/odata/api/interfaces(79)/asNums(0)/asNextHop?
&startTime=1536754500&endTime=1537289100&resolution=min15
```

Orderby

The $orderby system query option specifies the order in which items are returned from the CA NFA OData service. The value of the $orderby system query option contains a comma-separated list of expressions whose result are used to sort the items. The expression can include a suffix 'asc' for ascending or 'desc' for descending, separated from the property name by one or more spaces. If asc or desc is not specified, the service must order by the specified property in ascending order. You can order the entities based on different harvesters. You can also use the order by with the Expand operation.

Resource URL

```
http://<nfa odata host>:<nfa odata port>/odata/api/<EntityType Name>?$orderby=<Property
 Name 1>, <Property Name 2>
```

Method

GET

Parameters

The following table includes the mandatory parameter.

Parameters	Description
EntityType Name	Indicates the type of entity. For more information about list of valid EntityType Name, see Metadata.
Property Name 1, Property Name 2	Indicates the property of a given Entity type. For more information about list of valid Property Name, see Metadata.

Sample Request

This example displays list of `availableInterfaces` entity data `orderby`

```
ifIndex
```

property.

```
http://127.0.0.1:8981/odata/api/availableInterfaces?$orderby=ifIndex
```

Sample Response


```
{
 "@odata.context": "$metadata#availableInterfaces",
 "value": [
 {
 "ID": 1,
 "routerId": 1500001,
 "name": "10.0.9.20",
 "description": "10.0.9.20",
 "speed": 0,
 "ifMapId": 1,
 "ifIndex": 1,
 "ifDescr": "Interface 1",
 "ifType": 1,
 "ifName": "",
 "portName": "",
 "vrfName": "",
 "ifAlias": "",
 "UpdatedOn": 1542306911,
 "enabled": "N",
 "LastFlow": 1541996116,
 "IpAddr": null
 },
 {
 "ID": 7,
 "routerId": 1500002,
 "name": "10.5.0.29",
 "description": "10.5.0.29",
 "speed": 0,
 "ifMapId": 1,
 "ifIndex": 5,
 "ifDescr": "Interface 5",
 "ifType": 1,
 "ifName": "",
 "portName": "",
 "vrfName": "",
 "ifAlias": "",
 "UpdatedOn": 1542306913,
 "enabled": "N",
 "LastFlow": 1541996116,
 "IpAddr": null
```

```
 },
 {
 "ID": 24,
 "routerId": 1500003,
 "name": "10.0.42.4",
 "description": "10.0.42.4",
 "speed": 0,
 "ifMapId": 1,
 "ifIndex": 31,
 "ifDescr": "Interface 31",
 "ifType": 1,
 "ifName": "",
 "portName": "",
 "vrfName": "",
 "ifAlias": "",
 "UpdatedOn": 1542306912,
 "enabled": "N",
 "LastFlow": 1541996116,
 "IpAddr": null
 }
 ],
}
```


Top, Skip and Count

The `$top` system query option indicates the top number of items in the queried collection to be included in the result.

The `$skip` system query option indicates the number of items in the queried collection that are to be skipped and not included in the result. You can request a particular page of an item by combining `$top` and `$skip`.

The $count system query option indicates the clients to display a count of the matching resources in the response. The `$count` query option has a Boolean value of `true` or `false`.

- `true`: Indicates the client to display a count of the matching resources.
- `false`: Indicates the client not to display a count of the matching resources.

Resource URL

```
http://<nfa odata host>:<nfa odata port>/odata/api/<EntityType Name>?$top=<value>&
$skip=<value>&$count=<value>
```

Method

GET

Parameters

The following table includes the mandatory parameter.

Parameters	Description
EntityType Name	Indicates the type of entity. For more information about list of valid EntityType Name, see Metadata.

Sample Request

This sample API performs the following operations:

- Displays top 3 values of router entity.
- Skips the first value in router entity.
- Displays the total count of matching resources.

```
http://127.0.0.1:8981/odata/api/routers?$top=3&$skip=1&$count=true
```

Sample Response


```
{
 "@odata.context": "$metadata#routers",
 "@odata.count": 3,
 "value": [
 {
 "ID": 1500002,
 "routerAddress": "127.0.0.1",
 "sysDescr": "127.0.0.1",
 "sysName": "",
 "deviceName": "127.0.0.1",
 "deviceAlias": "127.0.0.1",
 "sysUptime": 0,
 "lastData": 0,
 "lastReboot": 1535031377,
 "lastRefresh": 0,
 "lastDiscovery": 0,
 "lastHarvesterUpdate": 0,
 "firstPollError": 1535031440,
 "nextPollRetry": 0,
 "harvesterID": 2,
 "profileId": 0,
 "snmpVersion": 2,
 "snmpPort": 161,
 "snmpTimeout": 3,
 "snmpRetry": 3,
 "snmpMaxRows": 10,
 "ifNumber": 0,
 "interfaceCount": 10,
```

```
"agentCount": 10,
"dnsLastLookupTime": 1535373746,
"dnsExpireTime": 1535978546,
"syncUpdateTime": 0,
"routerName": null,
"routerUpdatedOn": 1533559103,
"templateId": 1,
"snmpProxyAddress": "127.0.0.1",
"dnsProxyAddress": "127.0.0.1",
"tenantId": 8,
"domainID": 1
},
{
"ID": 1500003,
"routerAddress": "127.0.0.1",
"sysDescr": "127.0.0.1",
"sysName": "",
"deviceName": "127.0.0.1",
"deviceAlias": "127.0.0.1",
"sysUptime": 0,
"lastData": 0,
"lastReboot": 1535031377,
"lastRefresh": 0,
"lastDiscovery": 0,
"lastHarvesterUpdate": 0,
"firstPollError": 1535031440,
"nextPollRetry": 0,
"harvesterID": 2,
"profileId": 0,
"snmpVersion": 2,
"snmpPort": 161,
"snmpTimeout": 3,
"snmpRetry": 3,
"snmpMaxRows": 10,
"ifNumber": 0,
"interfaceCount": 10,
"agentCount": 10,
"dnsLastLookupTime": 1535373746,
"dnsExpireTime": 1535978546,
"syncUpdateTime": 0,
"routerName": null,
"routerUpdatedOn": 1533559103,
"templateId": 1,
"snmpProxyAddress": "127.0.0.1",
"dnsProxyAddress": "127.0.0.1",
"tenantId": 8,
```

```
 "domainID": 1
 },
 {
 "ID": 1500004,
 "routerAddress": "127.0.0.1",
 "sysDescr": "127.0.0.1",
 "sysName": "",
 "deviceName": "127.0.0.1",
 "deviceAlias": "127.0.0.1",
 "sysUptime": 0,
 "lastData": 0,
 "lastReboot": 1535031377,
 "lastRefresh": 0,
 "lastDiscovery": 0,
 "lastHarvesterUpdate": 0,
 "firstPollError": 1535031440,
 "nextPollRetry": 0,
 "harvesterID": 2,
 "profileId": 0,
 "snmpVersion": 2,
 "snmpPort": 161,
 "snmpTimeout": 3,
 "snmpRetry": 3,
 "snmpMaxRows": 10,
 "ifNumber": 0,
 "interfaceCount": 10,
 "agentCount": 10,
 "dnsLastLookupTime": 1535373746,
 "dnsExpireTime": 1535978546,
 "syncUpdateTime": 0,
 "routerName": null,
 "routerUpdatedOn": 1533559103,
 "templateId": 1,
 "snmpProxyAddress": "127.0.0.1",
 "dnsProxyAddress": "127.0.0.1",
 "tenantId": 8,
 "domainID": 1
 }
 ]
}
```


Pagination

The pagination option lets you view the records in chunks on multiple pages by limiting number of records on each page. By default, this API display 10 records per page. If there are more than ten records for an entity, then it displays the first ten records in the first page followed by a link to the next set of pages.

NOTE

: You can increase the number of records to be displayed per page by setting the required value for the `maxpagesize` parameter in HTTP Header. The maximum limit to display the records per page is 2000. For more information about overriding maxpagesize, see Limiters.

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/<EntitySet Name>
```

Method

GET

Parameters

The following table includes the mandatory parameter.

Parameters	Description
EntitySet Name	Indicates the type of entity set. For more information about list of valid EntitySet Name, see Metadata.

Sample Request

The following request displays the ten records in application settings entity.

```
http://127.0.0.1:8981/odata/api/applicationSettings
```

Sample Response

```
{
"@odata.context": "$metadata#applicationSettings",
"value": [
{
"Parameter": "agentAWOLLimit",
"Label": "Interface Data Absence Limit",
"Value": "44h"
},
{
"Parameter": "AppMap_TCPRebasePort",
"Label": "TCP Rebase Port",
"Value": "8000"
},
{
"Parameter": "AppMap_TOSMask",
"Label": "ToS Mask",
"Value": "254"
```

```
},
{
"Parameter": "AppMap_UDPRebasePort",
"Label": "UDP Rebase Port",
"Value": "7000"
},
{
"Parameter": "autoEnableInterfaces",
"Label": "Auto-Enable Interfaces",
"Value": "True"
},
{
"Parameter": "defaultTZ",
"Label": "Default Time Zone",
"Value": "GMT"
},
{
"Parameter": "DNSDomains",
"Label": "DNS Domains",
"Value": "aa-"
},
{
"Parameter": "dropToZero",
"Label": "Show Trendline Zeroes",
"Value": "True"
},
{
"Parameter": "emailFromAddress",
"Label": "From Address",
"Value": "<no default>"
},
{
"Parameter": "emailSMTPServer",
"Label": "SMTP Server",
"Value": "aaa"
}
],
"@odata.nextLink": "http://127.0.0.1:8981/odata/api/applicationSettings?$skiptoken=10"
}
```


Pagination with `$top` System Query Option

Pagination can be combined with the `$top` system query option using the GET method. The `$top`

query is applied first and then the records are paginated.

Sample Request

In this example, the `$top` is set to 23 and the `maxpagesize` parameter is not overridden. The response displays ten records each on the first page, and the second page, and three records in the third page.

```
http://127.0.0.1:8981/odata/api/interfaces?$top=23
```

> **NOTE**
>
> : Pagination displays 10 data per page followed by a link to the next set of pages. By default, expansion displays only 10 records. You can increase the display limit by using the `$top` system query parameter in `$expand`. The maximum limit of expand is 2000 record per page, beyond which the data is chunked. To retrieve larger data sets, use Navigation.
>
> For more information about using Navigation, see Navigation.

Drill Down Operations

The CA NFA drill down feature provides the following benefits to a Network Administrator:

Benefits of Drill Down:

- Enhanced visibility to information from NFA
- Ability to generate reports on the fly
- Pinpoint the root cause of issues that help reduce outages
- Improved network data analysis

The CA NFA OData API provides the following drill-down options:

- Expansion
- Navigation

Entity Operations

The CA NFA OData API lets you perform the administrative operations on following CA NFA entities:

> **NOTE**
>
> **More Information:**
>
> Administration Page Options

Routers and Interfaces

In CA NFA OData API, the `availableInterfaces` entity refers to interfaces that you can view on the **Available Interfaces** page, in the CA NFA Console. The `availableInterfaces` entity lists the interfaces that are not enabled and do not have source of collected data.
In CA NFA OData API, the `Interfaces` entity refers to the interfaces that you can view on the **Active Interfaces** page, in the CA NFA Console. The `Interfaces` entity lists only the interfaces that have contributed to collected data at some time and custom virtual interfaces.
The action of an entity in NFA API is same as action in the CA NFA Console. For more information on Administration page, see Administration Page.

Routers

This section explains the following supported operations on routers:

Edit Router - Single

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/routers(PropertyRef Name value)
```

Method

PATCH

Payload

```
{
"snmpVersion": <snmp version value>,
"snmpPort": <snmp portvalue>,
"routerName": "<Router name>",
"templateId": <template ID value>,
"domainID": <domain ID value>,
"profileId": <profile ID value>
}
```

NOTE

- The valid snmpVersion values are 1, 2 or 3.
- Enter the template Id greater than or equal to 1.

Sample Request

```
http://127.0.0.1:8981/odata/api/routers(1500004)
```

Sample Payload

```
{
"snmpVersion": 2,
"snmpPort": 123,
"routerName": "127.0.0.1",
"templateId": 5,
"domainID": 1,
"profileId": 5
}
```

Sample Response

There is no response for this API.

NOTE
Editing the domain ID of a router updates the associated tenant ID.

Edit Routers - Bulk

This API enables you to update the values of the multiple entities having similar properties, at once. You can edit the following properties using this API.

- SNMP Profile
- Domain

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/routers/com.ca.nfa.odata.editRouters
```

Method

POST

Payload

```
{
"RouterIds": [<Enter the list of Router_IDs separated by a comma>],
"snmpVersion": <snmp version value>,
"snmpPort" : <snmp port value>,
"templateId" : <template id value>,
"profileId" : <profile id value>
"domainID" : <domain id value>
}
```

NOTE

The valid snmpVersion values are 1, 2 or 3.

Sample Request

```
http://127.0.0.1:8981/odata/api/routers/com.ca.nfa.odata.editRouters
```

Sample Payload

```
{
 "RouterIds": [1500012,1500013],
"snmpVersion": 3,
"snmpPort": 1654,
"templateId": 1,
"profileId": 4,
"domainID" :1
}
```

Sample Response

```
{
  "@odata.context": "$metadata#Collection(com.ca.nfa.odata.router)",
  "value": [
    {
      "routerUpdatedOn": 1599147491,
      "deviceAlias": "10.0.9.11",
      "dnsExpireTime": 1599752503,
      "snmpMaxRows": 10,
      "templateId": 1,
      "dnsProxyAddress": "10.74.241.203",
      "deviceName": "10.0.9.11",
      "domainID": 1,
      "lifecyclestate": "ACTIVE",
```

```
  "lastHarvesterUpdate": 0,
  "sysDescr": "10.0.9.11",
  "sysUptime": 0,
  "snmpProxyAddress": "10.74.241.203",
  "snmpRetry": 3,
  "ifNumber": 0,
  "harvesterID": 2,
  "lastReboot": 1599562571,
  "routerName": null,
  "firstPollError": 1599562606,
  "sysName": "",
  "interfaceCount": 2,
  "snmpTimeout": 3,
  "ID": 1500001,
  "lastData": 0,
  "lastRefresh": 0,
  "snmpVersion": 2,
  "routerAddress": "10.0.9.11",
  "lastDiscovery": 0,
  "dnsLastLookupTime": 1599147703,
  "nextPollRetry": 0,
  "syncUpdateTime": 0,
  "profileId": 0,
  "tenantId": 8,
  "snmpPort": 161,
  "agentCount": 2,
  "flowStatus": "Red",
  "enabledInterfacesCount": 2,
  "profileName": null,
  "harvesterAddress": "127.0.0.1"
},
{
  "routerUpdatedOn": 1599147492,
  "deviceAlias": "10.0.9.14",
  "dnsExpireTime": 1599752503,
  "snmpMaxRows": 10,
  "templateId": 1,
  "dnsProxyAddress": "10.74.241.203",
  "deviceName": "10.0.9.14",
  "domainID": 1,
  "lifecyclestate": "ACTIVE",
  "lastHarvesterUpdate": 0,
  "sysDescr": "10.0.9.14",
  "sysUptime": 0,
  "snmpProxyAddress": "10.74.241.203",
  "snmpRetry": 3,
```

```
      "ifNumber": 0,
      "harvesterID": 2,
      "lastReboot": 1599562571,
      "routerName": null,
      "firstPollError": 1599562606,
      "sysName": "",
      "interfaceCount": 2,
      "snmpTimeout": 3,
      "ID": 1500002,
      "lastData": 0,
      "lastRefresh": 0,
      "snmpVersion": 2,
      "routerAddress": "10.0.9.14",
      "lastDiscovery": 0,
      "dnsLastLookupTime": 1599147703,
      "nextPollRetry": 0,
      "syncUpdateTime": 0,
      "profileId": 0,
      "tenantId": 8,
      "snmpPort": 161,
      "agentCount": 2,
      "flowStatus": "Red",
      "enabledInterfacesCount": 2,
      "profileName": null,
      "harvesterAddress": "127.0.0.1"
    }
  ]
}
```


Enable Router - Single

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/routers(<routerID Value>)/
com.ca.nfa.odata.enableRouter
```

Method

POST

Sample Request

```
http://127.0.0.1:8981/odata/api/routers(1500002)/com.ca.nfa.odata.enableRouter
```

Sample Response


```
{
"@odata.context": "$metadata#com.ca.nfa.odata.router",
"ID": 1500002,
```

```
"routerAddress": "127.0.0.1",
"sysDescr": "127.0.0.1",
"sysName": "",
"deviceName": "127.0.0.1",
"deviceAlias": "127.0.0.1",
"sysUptime": 0,
"lastData": 0,
"lastReboot": 0,
"lastRefresh": 0,
"lastDiscovery": 1552995238,
"lastHarvesterUpdate": 0,
"firstPollError": 0,
"nextPollRetry": 0,
"harvesterID": 2,
"profileId": 4,
"snmpVersion": 2,
"snmpPort": 161,
"snmpTimeout": 3,
"snmpRetry": 3,
"snmpMaxRows": 10,
"ifNumber": 0,
"interfaceCount": 10,
"agentCount": 10,
"dnsLastLookupTime": 1556194180,
"dnsExpireTime": 1556798980,
"syncUpdateTime": 0,
"routerName": null,
"routerUpdatedOn": 1552995238,
"templateId": 1,
"snmpProxyAddress": "127.0.0.1",
"dnsProxyAddress": "127.0.0.1",
"tenantId": 8,
"domainID": 1
}
```


Enable Router - Bulk

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/routers/
com.ca.nfa.odata.enableRouters
```

Method

POST

Payload

```
"RouterIds":[<List of Router_ids separated by a comma>
```

Sample Request

```
http://127.0.0.1:8981/odata/api/routers/com.ca.nfa.odata.enableRouters
```

Sample Payload

```
"RouterIds":[1554235706, 1554840506, 1553025868]
```

Sample Response


```
{
"@odata.context": "$metadata#Collection(com.ca.nfa.odata.router)",
"value": [
{
"ID": 1500001,
"routerAddress": "127.0.0.1",
"sysDescr": "127.0.0.1",
"sysName": "",
"deviceName": "127.0.0.1",
"deviceAlias": "127.0.0.1",
"sysUptime": 0,
"lastData": 0,
"lastReboot": 0,
"lastRefresh": 0,
"lastDiscovery": 0,
"lastHarvesterUpdate": 0,
"firstPollError": 0,
"nextPollRetry": 0,
"harvesterID": 1,
"profileId": 0,
"snmpVersion": 2,
"snmpPort": 161,
"snmpTimeout": 3,
"snmpRetry": 3,
"snmpMaxRows": 10,
"ifNumber": 0,
"interfaceCount": 10,
"agentCount": 10,
"dnsLastLookupTime": 1554235706,
"dnsExpireTime": 1554840506,
"syncUpdateTime": 0,
"routerName": null,
"routerUpdatedOn": 1553025868,
"templateId": 1,
"snmpProxyAddress": "127.0.0.1",
"dnsProxyAddress": "127.0.0.1",
"tenantId": 8,
"domainID": 1
}
]
}
```

Disable Router - Single

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/routers(<routerID value>)/com.ca.nfa.odata.disableRouter
```

Method

POST

Sample Request

```
http://127.0.0.1:8981/odata/api/routers(1500002)/com.ca.nfa.odata.disableRouter
```

Sample Response


```
{
"@odata.context": "$metadata#com.ca.nfa.odata.router",
"ID": 1500002,
"routerAddress": "127.0.0.1",
"sysDescr": "127.0.0.1",
"sysName": "",
"deviceName": "127.0.0.1",
"deviceAlias": "127.0.0.1",
"sysUptime": 0,
"lastData": 0,
"lastReboot": 0,
"lastRefresh": 0,
"lastDiscovery": 1552995238,
"lastHarvesterUpdate": 0,
"firstPollError": 0,
"nextPollRetry": 0,
"harvesterID": 2,
"profileId": 4,
"snmpVersion": 2,
"snmpPort": 161,
"snmpTimeout": 3,
"snmpRetry": 3,
"snmpMaxRows": 10,
"ifNumber": 0,
"interfaceCount": 10,
"agentCount": 10,
"dnsLastLookupTime": 1556194180,
"dnsExpireTime": 1556798980,
"syncUpdateTime": 0,
"routerName": null,
"routerUpdatedOn": 1552995238,
"templateId": 1,
"snmpProxyAddress": "127.0.0.1",
"dnsProxyAddress": "127.0.0.1",
```

```
"tenantId": 8,
"domainID": 1
}
```

Disable Router - Bulk

Sample URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/routers/
com.ca.nfa.odata.disableRouters
```

Method

POST

Payload

```
"RouterIds":[<List of router_ids separated by a comma>]
```

Sample Request

```
http://127.0.0.1:8981/odata/api/routers/com.ca.nfa.odata.disableRouters
```

Sample Payload

```
"RouterIds":[150032,123894,222891,100034]
```

Sample Response

```
{
"@odata.context": "$metadata#Collection(com.ca.nfa.odata.router)",
"value": [
{
"ID": 1500001,
"routerAddress": "127.0.0.1",
"sysDescr": "127.0.0.1",
"sysName": "",
"deviceName": "127.0.0.1",
"deviceAlias": "127.0.0.1",
"sysUptime": 0,
"lastData": 0,
"lastReboot": 0,
"lastRefresh": 0,
"lastDiscovery": 0,
"lastHarvesterUpdate": 0,
"firstPollError": 0,
"nextPollRetry": 0,
"harvesterID": 1,
"profileId": 0,
"snmpVersion": 2,
"snmpPort": 161,
"snmpTimeout": 3,
"snmpRetry": 3,
"snmpMaxRows": 10,
```

```
"ifNumber": 0,
"interfaceCount": 10,
"agentCount": 10,
"dnsLastLookupTime": 1554235706,
"dnsExpireTime": 1554840506,
"syncUpdateTime": 0,
"routerName": null,
"routerUpdatedOn": 1553025868,
"templateId": 1,
"snmpProxyAddress": "127.0.0.1",
"dnsProxyAddress": "127.0.0.1",
"tenantId": 8,
"domainID": 1
}
]
}
```


Delete Router - Single

This API allows you to delete a single router.

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/routers(<router_id>)
```

Method

DELETE

HTTP Headers

```
Content-Type:application/json; odata.metadata=minimal
```

Parameters

The following table includes the mandatory parameters.

Parameters	Description
router_id	Indicates the router id that you wish to delete from a router entity set.

Sample Request

The following request illustrates the delete action for the router id 1500002.

```
http://127.0.0.1:8981/odata/api/routers(1500002)
```

Sample Response

There is no response for this API.

Delete Router - Bulk

The following code is a reference metadata section for deleting bulk router action.

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/routers/
com.ca.nfa.odata.deleteRouters
```

Method

POST

Payload

```
"RouterIds":[1500008,1500009]
```

Sample Request

The following request illustrates the delete action for bulk delete router entity.

```
http://127.0.0.1:8080/odata/api/routers/com.ca.nfa.odata.deleteRouters
```

Sample Payload

```
"RouterIds":[150032,123894,222891,100034]
```

Sample Response


```
{
"@odata.context": "$metadata#Collection(com.ca.nfa.odata.router)",
"value": [
{
"ID": 1500008,
"routerAddress": "127.0.0.1",
"sysDescr": "127.0.0.1",
"sysName": "",
"deviceName": "127.0.0.1",
"deviceAlias": "127.0.0.1",
"sysUptime": 0,
"lastData": 0,
"lastReboot": 0,
"lastRefresh": 0,
"lastDiscovery": 0,
"lastHarvesterUpdate": 0,
"firstPollError": 0,
"nextPollRetry": 0,
"harvesterID": 1,
"profileId": 4,
"snmpVersion": 1,
"snmpPort": 36863,
"snmpTimeout": 3,
"snmpRetry": 3,
"snmpMaxRows": 10,
"ifNumber": 0,
"interfaceCount": 2,
"agentCount": 0,
```

```
"dnsLastLookupTime": 1556273867,
"dnsExpireTime": 1556878667,
"syncUpdateTime": 0,
"routerName": null,
"routerUpdatedOn": 1556273741,
"templateId": 1,
"snmpProxyAddress": "127.0.0.1",
"dnsProxyAddress": "127.0.0.1",
"tenantId": 8,
"domainID": 1
},
{
"ID": 1500009,
"routerAddress": "127.0.0.1",
"sysDescr": "127.0.0.1",
"sysName": "",
"deviceName": "127.0.0.1",
"deviceAlias": "127.0.0.1",
"sysUptime": 0,
"lastData": 0,
"lastReboot": 0,
"lastRefresh": 0,
"lastDiscovery": 0,
"lastHarvesterUpdate": 0,
"firstPollError": 0,
"nextPollRetry": 0,
"harvesterID": 1,
"profileId": 4,
"snmpVersion": 1,
"snmpPort": 36863,
"snmpTimeout": 3,
"snmpRetry": 3,
"snmpMaxRows": 10,
"ifNumber": 0,
"interfaceCount": 10,
"agentCount": 8,
"dnsLastLookupTime": 1556273867,
"dnsExpireTime": 1556878667,
"syncUpdateTime": 0,
"routerName": null,
"routerUpdatedOn": 1556273741,
"templateId": 1,
"snmpProxyAddress": "127.0.0.1",
"dnsProxyAddress": "127.0.0.1",
"tenantId": 8,
"domainID": 1
```

```
}
]
}
```

Interfaces

This section explains the following supported operations on interfaces:

Edit Interfaces - Single

To edit a single entity, use HTTP PATCH Method to send the request.

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/interfaces(PropertyRef Name value)
```

Method

PATCH

Payload

```
{
"Description": <Value>,
"InSpeed" : <Value>,
"OutSpeed" : <Value>,
"IfType" : <Value>,
"Name" : <Value>,
"DomainId" : <Value>
}
```

Sample Request

```
http://127.0.0.1:8981/odata/api/interfaces(10)
```

Sample Payload

```
{
"Description":"interfaceupdate",
"Name":"Interface1111",
"InSpeed":100,
"OutSpeed":101,
"IfType":"WAN",
"DomainId":1
}
```

Sample Response

There is no response for this API.

Edit Interfaces - Bulk

This CA NFA OData API enables you to update values of the multiple entities having similar properties at once.

> **NOTE**
>
> This Bulk Edit API is applicable only for few properties of Interfaces.

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/interfaces/
com.ca.nfa.odata.editInterfaces
```

Method

POST

Payload

```
{
"InterfaceIds":[<List of Interface_ID s separated by a comma>],
"InSpeed" : <value>,
"OutSpeed": <value>,
"IfType" : <value>,
"DomainId": <value>
}
```

Sample Request

```
http://127.0.0.1:8981/odata/interfaces/com.ca.nfa.odata.editInterfaces
```

Sample Payload

```
{
"InterfaceIds":[943,944],
"InSpeed":100,
"OutSpeed":100,
"IfType":"WAN",
"DomainId":3
}
```

Sample Response


```
{
    "@odata.context": "$metadata#Collection(com.ca.nfa.odata.interfaces)",
    "value": [
        {
            "ID": 943,
            "AgentType": "Physical",
            "RouterAddress": "127.0.0.1",
            "RouterId": 1500016,
            "Description": "interface Description",
            "InSpeed": 10000000,
            "OutSpeed": 1000000,
            "IfIndex": 4,
            "PersistentIfIndex": 5,
            "IfType": "WAN",
            "UpdatedOn": 1602590969,
            "Enabled": "N",
            "LastData": 1602673218,
            "HarvesterAddress": "10.80.89.28",
```

```
            "Name": "Interface 1",
            "TrafficStatus": null,
            "DomainId": 1,
            "DomainName": null,
            "Subnet": [],
            "ComponentAgentIds": []
        },
        {
            "ID": 944,
            "AgentType": "Physical",
            "RouterAddress": "127.0.0.1",
            "RouterId": 1500016,
            "Description": "",
            "InSpeed": 127,
            "OutSpeed": 149,
            "IfIndex": 6,
            "PersistentIfIndex": 8,
            "IfType": "WAN ATM AAL5",
            "UpdatedOn": 1602590969,
            "Enabled": "N",
            "LastData": 1602673218,
            "HarvesterAddress": "127.0.0.1",
            "Name": "Interface 6",
            "TrafficStatus": null,
            "DomainId": 10,
            "DomainName": null,
            "Subnet": [],
            "ComponentAgentIds": []
        }
    ]
}
```


Merge Interfaces

The CA NFA OData API lets you merge two interfaces, similar to Merging interfaces in DX NetOps console.

Resource URI

```
 http://<nfa odata host>:<nfa odata port>/odata/api/interfaces/
com.ca.nfa.odata.mergeInterfaces
```

Method

POST

Payload


```
{
```

```
"interface1" : <Interface_ID>,
 "interface2" : <Interface_ID>,
 "deleteSource" : <true/false>,
"isOverLap" : <true/false>
}
```

Parameters

The following are the parameters in payload:

Parameters	Description
interface1	Indicates the Interface id for the first interface as set in the interface table.
interface2	Indicates the Interface id for the second interface as set in the interface table.
deleteSource	Indicates if the source must be deleted after merging. Valid values are: • **true** : Indicates that the source must be deleted. • **false:** Indicates that the source must not be deleted.
isOverLap	This parameter indicates if overlapped data must be merged. Valid values are: • **true:** Indicates that the overlapped data must be merged. • **false** : Indicates that the overlapped data must not be merged and displays the user an appropriate message to proceed.

NOTE
Either of the Interface1 or Interface2 can be source, while the other will be destination.

Sample Request

```
http://10.13.91.413:8981/odata/api/interfaces/com.ca.nfa.odata.mergeInterfaces
```

Sample Payload

```
 {
"interface1" : 46,
"interface2" : 31,
"isOverLap" : true,
"deleteSource":false
}
```

Sample Response

```
 {
"@odata.context": "$metadata#Collection(com.ca.nfa.odata.interfaces)",
"value": [
{
"ID": 31,
"AgentType": "Physical",
"RouterAddress": "10.0.42.2",
```

```
"Description": "edge network",
"InSpeed": 1000000000,
"OutSpeed": 1000000000,
"IfIndex": 4,
"PersistentIfIndex": 7,
"IfType": "LAN-ET",
"UpdatedOn": 1551158203,
"Enabled": "Y",
"LastData": 1551344433,
"HarvesterAddress": "10.238.80.99",
"Name": "Fa1/1"
},
{
"ID": 46,
"AgentType": "Physical",
"RouterAddress": "10.0.42.3",
"Description": "",
"InSpeed": 1000000000,
"OutSpeed": 1000000000,
"IfIndex": 4,
"PersistentIfIndex": 6,
"IfType": "LAN-ET",
"UpdatedOn": 1551158203,
"Enabled": "Y",
"LastData": 1551189619,
"HarvesterAddress": "10.238.80.99",
"Name": "Fa1/1"
}
]
}
```


Delete Interface - Single

This API allows you to delete a single interface.

NOTE

The API to delete single interface and interface aggregation are the same

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/interfaces(<interface_id>)
```

Method

DELETE

Parameters

The following table includes the mandatory parameters.

Parameters	Description
interface_id	Indicates the interface id that you wish to delete from Interfaces entity set.

Sample Request

The following request illustrates the delete action for the interface id 3051.

```
http://127.00.1:8981/odata/api/interfaces(3051)
```

Sample Response

There is no response for this API.

Delete Interfaces - Bulk

The API allows you to delete multiple interface aggregations in bulk.

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/interfaces/
com.ca.nfa.odata.deleteInterfaces
```

Method

POST

Payload

```
{
"InterfaceIds":[<Enter the list of Interface_ids separated by a comma>]
}
```

Sample Request

The following request illustrates the bulk delete action for interfaces entity.

```
http://127.00.1:8981/odata/api/interfaces/com.ca.nfa.odata.deleteInterfaces
```

Sample Payload

```
{
"InterfaceIds" : [123,402]
}
```

Sample Response

```
{
"@odata.context": "$metadata#Collection(com.ca.nfa.odata.interfaces)",
"value": [
{
```

```
"ID": 123,
"AgentType": "Physical",
"RouterAddress": "127.0.0.1",
"Description": "Device209",
"InSpeed": 1000000000,
"OutSpeed": 1000000000,
"IfIndex": 1,
"PersistentIfIndex": 1,
"IfType": "LAN-ET",
"UpdatedOn": 1544529750,
"Enabled": "Y",
"LastData": 1546937182,
"HarvesterAddress": "127.0.0.1",
"Name": "Device209"
},
{
"ID": 402,
"AgentType": "Physical",
"RouterAddress": "127.0.0.1",
"Description": "127.0.0.1",
"InSpeed": 0,
"OutSpeed": 0,
"IfIndex": 7,
"PersistentIfIndex": 4,
"IfType": "WAN",
"UpdatedOn": 1544598925,
"Enabled": "Y",
"LastData": 1545139819,
"HarvesterAddress": "127.0.0.1",
"Name": "127.0.0.1"
}
]
}
```


Available Interface

This section explains the following supported operations on available interface:

Enable Available Interface - Single

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/availableInterfaces(<PropertyRef Name value>)/com.ca.nfa.odata.enableInterface
```

Method

POST

Parameters

Parameter	Description
PropertyRef Name	Indicate the interface ID of the availableInterfaces entity type. For more information about list of availableInterfaces, see Metadata.

Sample Request

```
http://127.00.1:8981/odata/api/availableInterfaces(11)/com.ca.nfa.odata.enableInterface
```

Sample Response

```
{
"@odata.context": "$metadata#com.ca.nfa.odata.availableInterfaces",
"ID": 11,
"routerId": 1500002,
"name": "Interface 1",
"description": "127.0.0.1",
"speed": 0,
"ifMapId": 10,
"ifIndex": 9,
"ifDescr": "Interface 1",
"ifType": 1,
"ifName": "",
"portName": "",
"vrfName": "",
"ifAlias": "",
"UpdatedOn": 1542306913,
"enabled": "true",
"LastFlow": 1541996116,
"IpAddr": null
}
```

Enable Available Interfaces - Bulk

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/
availableInterfaces/com.ca.nfa.odata.enableInterfaces
```

Method

POST

Payload

```
{
  "InterfaceIds":[<List of Interface_ids separated by a comma>]
}
```

Sample Request

```
http://127.0.0.1:8981/odata/api/availableInterfaces/com.ca.nfa.odata.enableInterfaces
```

Sample Payload

```
{
"InterfaceIds":[1,3,10]
}
```

Sample Response


```
{
"@odata.context": "$metadata#Collection(com.ca.nfa.odata.availableInterfaces)",
"value": [
{
"ID": 1,
"routerId": 1500001,
"name": "127.0.0.1",
"description": "127.0.0.1",
"speed": 0,
"ifMapId": 1,
"ifIndex": 1,
"ifDescr": "Interface 1",
"ifType": 1,
"ifName": "",
"portName": "",
"vrfName": "",
"ifAlias": "",
"UpdatedOn": 1542306911,
"enabled": "true",
"LastFlow": 1541996116,
"IpAddr": null
},
{
"ID": 3,
```

```
"routerId": 1500002,
"name": "127.0.0.1",
"description": "127.0.0.1",
"speed": 0,
"ifMapId": 3,
"ifIndex": 1,
"ifDescr": "Interface 1",
"ifType": 1,
"ifName": "",
"portName": "",
"vrfName": "",
"ifAlias": "",
"UpdatedOn": 1542306912,
"enabled": "true",
"LastFlow": 1541996116,
"IpAddr": null
},
{
"ID": 10,
"routerId": 1500002,
"name": "127.0.0.1",
"description": "127.0.0.1",
"speed": 0,
"ifMapId": 9,
"ifIndex": 8,
"ifDescr": "Interface 8",
"ifType": 1,
"ifName": "",
"portName": "",
"vrfName": "",
"ifAlias": "",
"UpdatedOn": 1542306913,
"enabled": "true",
"LastFlow": 1541996116,
"IpAddr": null
}
]
}
```


Disable Available Interface - Single

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/availableInterfaces(<InterfaceId
 value>)/com.ca.nfa.odata.disableInterface
```

Method

POST

Sample Request

```
http://127.0.0.1:8981/odata/api/
availableInterfaces(43)/com.ca.nfa.odata.disableInterface
```

Sample Response


```
{

"@odata.context": "$metadata#com.ca.nfa.odata.availableInterfaces",
"ID": 43,
"routerId": 1500006,
"name": "Interface 1",
"description": "",
"speed": 0,
"ifMapId": 1,
"ifIndex": 1,
"ifDescr": "Interface 1",
"ifType": 1,
"ifName": "",
"portName": "",
"vrfName": "",
"ifAlias": "",
"UpdatedOn": 1551354673,
"enabled": "false",
"LastFlow": 1556632817,
"IpAddr": null
}
```


Disable Available Interfaces - Bulk

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/
availableInterfaces/com.ca.nfa.odata.disableInterfaces
```

Method

POST

Payload

```
{
  "InterfaceIds":[<List of Interface_ids separated by a comma>]
}
```

Sample Request

```
http://127.0.0.1:8981/odata/api/availableInterfaces/com.ca.nfa.odata.disableInterfaces
```

Sample Payload

```
{
"InterfaceIds" : [483,484,485]
}
```

Sample Response


```
{

"@odata.context": "$metadata#Collection(com.ca.nfa.odata.availableInterfaces)",
"value": [
{
"ID": 483,
"routerId": 1500037,
"name": "Gi4/29",
"description": "Rac. non utilise",
"speed": 1000000000,
"ifMapId": 37,
"ifIndex": 37,
"ifDescr": "GigabitEthernet4/29",
"ifType": 6,
"ifName": "Gi4/29",
"portName": "",
"vrfName": "",
"ifAlias": "Rac. non utilise",
"UpdatedOn": 1553360767,
"enabled": "false",
"LastFlow": 0,
"IpAddr": null
```

```
},
{
"ID": 484,
"routerId": 1500037,
"name": "Gi4/30",
"description": "Rac. non utilise",
"speed": 1000000000,
"ifMapId": 38,
"ifIndex": 38,
"ifDescr": "GigabitEthernet4/30",
"ifType": 6,
"ifName": "Gi4/30",
"portName": "",
"vrfName": "",
"ifAlias": "Rac. non utilise",
"UpdatedOn": 1553360767,
"enabled": "false",
"LastFlow": 0,
"IpAddr": null
},
{
"ID": 485,
"routerId": 1500037,
"name": "Gi4/31",
"description": "Rac. non utilise",
"speed": 1000000000,
"ifMapId": 39,
"ifIndex": 39,
"ifDescr": "GigabitEthernet4/31",
"ifType": 6,
"ifName": "Gi4/31",
"portName": "",
"vrfName": "",
"ifAlias": "Rac. non utilise",
"UpdatedOn": 1553360767,
"enabled": "false",
"LastFlow": 0,
"IpAddr": null
}
]
}
```


Delete Available Interface

From DX NetOps 10.0.1, you can delete an interface from the available interfaces.

NOTE

When you delete an interface that is reachable through the SNMP, the interface is added back after the next SNMP poll.

Resource URI:

```
http://<nfa odata host>:<nfa odata port>/odata/api/availableInterfaces(<Enter the
 Available_Interface_ID to be deleted>)/com.ca.nfa.odata.deleteAvailableInterface
```

Method

POST

Sample Request

```
http://127.0.0.1:8981/odata/api/availableInterfaces(442)/
com.ca.nfa.odata.deleteAvailableInterface
```

Sample Response

```
{
"@odata.context": "$metadata#com.ca.nfa.odata.availableInterfaces",
 "ID": 442,
 "routerId": 1500023,
 "name": "Interface 7",
 "description": "",
 "speed": 0,
 "ifMapId": 4,
 "ifIndex": 7,
 "ifDescr": "Interface 7",
 "ifType": 1
 "ifName": "",
 "portName": "",
 "vrfName": "",
 "ifAlias": "",
 "UpdatedOn": 1548839196,
 "enabled": "Y",
 "LastFlow": 1548927084,
 "IpAddr": null
}
```

Interface Template

This section explains the following supported operations on interface template:

Add Interface Template

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/interfaceTemplate
```

Method

POST

Payload

```
{
"name": "<Enter the Interface template Name>",
"nameTemplate": "<Enter valid Interface name>",
"descriptionTemplate": "<Enter Valid Interface Description>"
}
```

NOTE

: The list of valid nameTemplate and descriptionTemplate values are:

- **[DeviceAlias]**: Indicates the Device alternate name.
- **[DeviceName]:** Indicates the Device Name.
- **[ifType]**: Indicates the If Type.
- **[ifIndex]**: Indicates the If Index.
- **[portName]**: Indicates the Port Name
- **[ifName]**: Indicates the If Name.
- **[ifAlias]**: Indicates the If Alternname Name
- **[ifDescr]**: Indicates the If Descriptions

You can also enter multiple nameTemplate and descriptionTemplate values by separating them with a | symbol.

Example 1: Add single value

Sample Request

This example illustrates single value for name, nameTemplate and descriptionTemplate.

```
http://127.0.0.1:8981/odata/api/interfaceTemplate
```

Sample Payload

```
{
"name": "test sample 3",
"nameTemplate": "[ifIndex]",
"descriptionTemplate": "[ifIndex]"
}
```

Sample Response

A new templateId is created for the sample payload parameter values.

```
{
"@odata.context": "$metadata#interfaceTemplate",
"templateId": 19,
```

```
"name": "test sample 3",
"nameTemplate": "[ifIndex]",
"descriptionTemplate": "[ifIndex]"
}
```

Example 2: Add multiple value

Sample Request

This example illustrates multiple values for nameTemplate and descriptionTemplate.

```
http://127.0.0.1:8981/odata/api/interfaceTemplate
```

Sample Payload

```
{
"name": "test sample 4",
"nameTemplate": "[ifType|ifIndex]",
"descriptionTemplate": "[ifType|ifIndex]"
}
```

Sample Response

A new template id is created for the given sample payload parameter values.

```
{
"@odata.context": "$metadata#interfaceTemplate",
"templateId": 20,
"name": "test sample 4",
"nameTemplate": "[ifType|ifIndex]",
"descriptionTemplate": "[ifType|ifIndex]"
}
```

Edit Interface Template

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/interfaceTemplate(<Enter the
 template_id to be updated>)
```

Method

PATCH

Payload

> **NOTE**
>
> : You can edit only the nameTemplate and descriptionTemplate properties for the **global-default** Interface Template.

The properties that you can edit for an Interface Template, other than **global-default** are:

```
{
"name": "<Enter the Interface template Name>",
"nameTemplate": "<Enter a valid Interface name>",
"descriptionTemplate": "<Enter a Valid Interface Description>"
```

```
}
```

NOTE

: The list of valid nameTemplate and descriptionTemplate values are:

- DeviceAlias
- DeviceName
- ifDescr
- ifAlias
- ifName
- portName
- ifIndex
- ifType

Sample Request

```
http://127.0.0.1:8981/odata/api/interfaceTemplate(1234)
```

Sample Payload

In this sample, the properties that you can edit for a Interface Template other than **global-default** are:

```
{
"name": "test123376",
"nameTemplate": "[ifmapid]",
"descriptionTemplate": "[ifmapid]"
}
```

Sample Response

There is no response for this API.

Delete Interface Template

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/interfaceTemplate(<Enter the
 template_id to be deleted>)
```

Method

DELETE

NOTE

: You cannot delete any property for the **global-default** Interface Template.

Sample Request

```
http://127.0.0.1:8981/odata/api/interfaceTemplate(123)
```

Sample Payload

```
{
"interfaceTemplate":[123]
}
```

Sample Response

There is no response for this API.

Interface Aggregations

This section explains the following supported operations on Interface Aggregations

Add Interface Aggregation

Resource URI

```
 http://<nfa odata host>:<nfa odata port>/odata/api/interfaces/
com.ca.nfa.odata.createInterfaceAggregation
```

Method

POST

Payload

```
{
"InterfaceIds" : [<Enter a valid ID>],
"Name" : "<Name of the Interface Aggregation>",
"Description" : "<Description of the interface Aggregation>",
"IfType" : "<Interface Aggregation Type>",
"InSpeed" : <Inspeed value> ,
"OutSpeed" : <Outspeed Value>
}
```

> **NOTE**
>
> You can also enter multiple InterfaceIds values by separating them with a comma (,) within the square brackets .

Example 1: Add single InterfaceId

Sample Request

This example illustrates single value for InterfaceIds,.

```
 http://127.0.0.1:8981/odata/api/interfaces/com.ca.nfa.odata.createInterfaceAggregation
```

Method

POST

Sample Payload

```
 {
"InterfaceIds" : [446],
"Name" : "test1",
"Description" : "test",
"IfType" : "WAN",
"InSpeed" : 10 ,
"OutSpeed" : 11
}
```

Example 2: Add multiple InterfaceId

Sample Request

This example illustrates multiple values for `InterfaceIds` .

```
http://127.0.0.1:8981/odata/api/interfaces/com.ca.nfa.odata.createInterfaceAggregation
```

Sample Payload

```
 {
"InterfaceIds" : [447,673,224],
"Name" : "test2",
"Description" : "test2",
"IfType" : "WAN",
"InSpeed" : 10 ,
"OutSpeed" : 11
}
```

Sample Response

```
 {
"@odata.context": "$metadata#Collection(com.ca.nfa.odata.interfaces)",
"value": [
{
"ID": 255,
"AgentType": "Aggregate",
"RouterAddress": "127.0.0.1",
"Description": "Description interface Aggregation2",
"InSpeed": 100000,
"OutSpeed": 1000,
"IfIndex": 255,
"PersistentIfIndex": 255,
"IfType": "WAN",
"UpdatedOn": 1554274019,
"Enabled": "Y",
"LastData": 0,
"HarvesterAddress": "127.0.0.1",
"Name": "Interface Aggregation10"
}
]
}
```

Get Interface Aggregations

URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/interfaces?
$filter=contains(AgentType, 'Aggregate')
```

Method

GET

Sample Request

```
http://127.0.0.1:8981/odata/api/interfaces?$filter=contains(AgentType, 'Aggregate')
```

Sample Response


```
 {
    "@odata.context": "http://localhost:8981/odata/api/
$metadata#interfaces(ID,AgentType,RouterAddress,Description,InSpeed,OutSpeed,IfIndex,Persistent
    "value": [
        {
            "InSpeed": 1,
            "IfType": "WAN",
            "Description": "",
            "PersistentIfIndex": 188,
            "Enabled": "Y",
            "RouterAddress": "0.0.0.1",
            "Name": "test",
            "AgentType": "Aggregate",
            "DomainId": 1,
            "LastData": 0,
            "OutSpeed": 1,
            "IfIndex": 188,
            "UpdatedOn": 1601359065,
            "ID": 188,
            "HarvesterAddress": "10.74.241.110",
            "TrafficStatus": "Red",
            "DomainName": "Default Domain",
            "Subnet": [],
            "ComponentAgentIds": [
                153,
                154
            ]
        }
    ]
}
```


NOTE

"ComponentAgentIds" property with the child interface ids is added to the Interface Aggregations.

Get Single Interface Aggregation

URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/interfaces(<interfaceaggregationid>)
```

Method

GET

Sample Request

```
http://127.0.0.1:8981/odata/api/interfaces(188)
```

Sample Response


```
{
    "@odata.context": "http://localhost:8981/odata/api/
$metadata#interfaces(ID,AgentType,RouterAddress,Description,InSpeed,OutSpeed,IfIndex,Persistent
$entity",
    "InSpeed": 1,
    "IfType": "WAN",
    "Description": "",
    "PersistentIfIndex": 188,
    "Enabled": "Y",
    "RouterAddress": "0.0.0.1",
    "Name": "test",
    "AgentType": "Aggregate",
    "DomainId": 1,
    "LastData": 0,
    "OutSpeed": 1,
    "IfIndex": 188,
    "UpdatedOn": 1601359065,
    "ID": 188,
    "HarvesterAddress": "10.74.241.110",
    "TrafficStatus": "Red",
    "DomainName": "Default Domain",
    "Subnet": [],
    "ComponentAgentIds": [
        153,
        154
    ]
}
```


Edit Interface Aggregation

This API is enhanced to edit the ComponentAgentIds for the single interface aggregation entity, this feature is not supported for the bulk interface edit.

URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/interfaces(<interfaceaggregationid>)
```

Method

PATCH

Sample Request

```
http://127.0.0.1:8981/odata/api/interfaces(188)
```

Sample Payload

```
{
    "Description": "aggregate interface update",
    "Name": "Interface aggregate",
    "InSpeed": 2000000,
    "OutSpeed": 20000000,
    "IfType": "WAN",
    "DomainId":1,
     "ComponentAgentIds": [
                153
           ]
}
```

Sample Response

There is no response for this API.

Delete Interface Aggregation

Interface aggregations can be deleted using the existing delete interfaces APIs. See the following APIs for more details:

Delete Interface - Single

Delete Interfaces - Bulk

Custom Virtual Interface

This section explains the following supported operations on Custom Virtual Interface

Add Custom Virtual Interface

Resource URI

```
 http://<nfa odata host>:<nfa odata port>/odata/api/interfaces/
com.ca.nfa.odata.createCustomVirtualInterface
```

Method

POST

Payload

```
{
"InterfaceId" : <Enter a valid ID>,
"Name" : "<Name of the Interface Aggregation>",
"Description" : "<Description of the interface Aggregation>",
"IfType" : "<Custom Virtual Interface Type>",
"InSpeed" : <Inspeed value> ,
"OutSpeed" : <Outspeed Value>,
"Subnet" :<Subnet value>
}
```

Example 1: Add Single Subnet

Sample Request

Create a Custom Virtual Interface with single Subnet:

```
  http://127.0.0.1:8981/odata/api/interfaces/
com.ca.nfa.odata.createCustomVirtualInterface
```

Method

POST

Sample Payload

```
 {
"InterfaceId" : 11,
"Name" : "testCVI",
"Description" : "CVI",
"IfType" : "WAN",
"InSpeed" : 10 ,
"OutSpeed" : 11 ,
"Subnet" :["10.0.0.4/22"]
}
```

Sample Response:

```
 {
"@odata.context": "$metadata#Collection(com.ca.nfa.odata.interfaces)",
"value": [
{
"ID": 537,
"AgentType": "Virtual",
"RouterAddress": "127.0.0.1",
"Description": "CVI",
"InSpeed": 10,
"OutSpeed": 11,
"IfIndex": 0,
"PersistentIfIndex": 2147483729,
"IfType": "WAN",
"UpdatedOn": 1546247778,
"Enabled": "Y",
"LastData": 0,
"HarvesterAddress": "127.0.0.1",
"Name": "testCVI"
}
]
}
```

Example 2: Add Multiple Subnet

Sample Request

Create a Custom Virtual Interface with multiple Subnets:

```
 http://127.0.0.1:8981/odata/api/interfaces/
com.ca.nfa.odata.createCustomVirtualInterface
```

Method

POST

Sample Payload

```
 {
"InterfaceId" : 12,
"Name" : "testCVI1",
"Description" : "CVI",
"IfType" : "WAN",
"InSpeed" : 10 ,
"OutSpeed" : 11 ,
"Subnet" :[
"10.0.0.4/22",
"10.0.0.9/3",
"10.0.0.9/22"
]
}
```

Sample Response:


```
{
    "@odata.context": "$metadata#Collection(com.ca.nfa.odata.interfaces)",
    "value": [
        {
            "ID": 11,
            "AgentType": "Virtual",
            "RouterAddress": "127.0.0.1",
            "RouterId": null,
            "Description": "CVI",
            "InSpeed": 1391671,
            "OutSpeed": 14393,
            "IfIndex": 0,
            "PersistentIfIndex": 2147483653,
            "IfType": "WAN ATM",
            "UpdatedOn": 1603098194,
            "Enabled": "N",
            "LastData": 0,
            "HarvesterAddress": "127.0.0.1",
            "Name": "testCVI",
            "TrafficStatus": null,
            "DomainId": null,
            "DomainName": null,
```

```
            "Subnet": [],
            "ComponentAgentIds": []
        }
    ]
}
```

Get Custom Virtual Interface

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/interfaces?$filter=
 contains(AgentType,'Virtual')
```

Sample Request

```
http://127.0.0.1:8981/odata/api/interfaces?$filter= contains(AgentType,'Virtual')
```

Method

GET

Sample Response:

```
{
"@odata.context": "http://localhost:8981/odata/api/
$metadata#interfaces(ID,AgentType,RouterAddress,Description,InSpeed,OutSpeed,IfIndex,Persistent
"value": [
  {
"InSpeed": 2000000,
"IfType": "WAN",
"Description": "interface updatemani",
"PersistentIfIndex": 2147483650,
"Enabled": "Y",
"RouterAddress": "10.0.40.33",
"Name": "Interface -edit apimani",
"AgentType": "Virtual",
"DomainId": 1,
"LastData": 0,
"OutSpeed": 20000000,
"IfIndex": 0,
"UpdatedOn": 1600937363,
"ID": 185,
"HarvesterAddress": "10.74.241.110",
"TrafficStatus": "Red",
"DomainName": "Default Domain",
"Subnet": [
  "1.1.1.0/24"
],
"ComponentAgentIds": [],
}
],
```

```
}
```

Edit Custom Virtual Interface

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/
interfaces(<CustomVirtualInterface_ID>)
```

Sample Request

```
http://127.0.0.1:8981/odata/api/interfaces(185)
```

Method

PATCH

Sample Payload

```
{
"Name": "Interface-edit2",
"Subnet": [
                                "1.1.1.0/24",
                                "10.10.10.10/24"
                ]
}
```

Sample Response

There is no response for this API.

Delete Custom Virtual Interface

Custom Virtual Interfaces can be deleted using the existing delete interfaces APIs. See the following APIs for more details:

Delete Interface - Single

Delete Interfaces - Bulk

System

The following options are available in System:

Harvester

Harvester supports the following options:

> **NOTE**
>
> **More Information:**
>
> Administration Page Options

Read All Harvesters

This API displays all Harvester records:

Resource URI

http://<*nfa odata host*>:<*nfa odata port*>/odata/api/harvesters

Method

GET

Sample Request

```
http://127.0.0.1:8981/odata/api/harvesters
```

Sample Response


```
{
"@odata.context": "$metadata#harvesters",
"value": [
{
"HarvesterId": 2,
"LastPullCheckpoint": 1545212329,
"PortDefinitionsLastDeployed": 0,
"HarvesterAddress": "127.0.0.1",
"DomainId": 3,
"FlowEnabledLastModified": 0,
"ManagementServerPort": 8080,
"DomainName": "Domain A",
"DomainDescription": "Domain A for FellsCargo",
"FlowEnabledLastDeployed": 0,
"LastPushCheckpoint": 1545212344,
"ApplicationMappingLastDeployed": 0,
"ReservedSeatingLastDeployed": 0,
"TrapDefinitionsLastDeployed": 1541748300,
"TenantId": 9,
"Description": "Single box configuration"
},
{
"HarvesterId": 5,
"LastPullCheckpoint": 0,
"PortDefinitionsLastDeployed": 0,
"HarvesterAddress": "127.0.0.1",
"DomainId": 1,
"FlowEnabledLastModified": 0,
"ManagementServerPort": 8080,
"DomainName": "Default Domain",
"DomainDescription": "The default domain for devices, interfaces, interface addresses
and networks.",
"FlowEnabledLastDeployed": 0,
"LastPushCheckpoint": 0,
"ApplicationMappingLastDeployed": 0,
"ReservedSeatingLastDeployed": 0,
```

```
 "TrapDefinitionsLastDeployed": 0,
 "TenantId": 8,
 "Description": "Test Harvesters2"
 },
]
 }
```

Read Single Harvester

The following API displays the details of the specified harvester:

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/harvesters(HarvesterId=<Value of the harvester ID>)
```

Method

GET

Sample Request

```
http://127.0.0.1:8981/odata/api/harvesters(HarvesterId=2)
```

Sample Response

```
{
"@odata.context": "$metadata#harvesters",
"HarvesterId": 2,
"HarvesterAddress": "127.0.0.1",
"Description": "Single box configuration",
"ManagementServerPort": 8080,
"FlowEnabledLastModified": 0,
"FlowEnabledLastDeployed": 0,
"LastPushCheckpoint": 1545212832,
"LastPullCheckpoint": 1545212820,
"ApplicationMappingLastDeployed": 0,
"PortDefinitionsLastDeployed": 0,
"ReservedSeatingLastDeployed": 0,
"TrapDefinitionsLastDeployed": 1541748300,
"DomainId": 3,
"DomainName": "Domain A",
"DomainDescription": "Domain A for FellsCargo",
"TenantId": 9
}
```

Add Harvester

NOTE
Adding/Editing any non-harvester IP will give timeout error as response.

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/harvesters
```

Method

POST

Payload

```
{
"HarvesterAddress": "<Enter the harvester address that you wish to add >",
"Description":"<Enter the Description>",
"DomainId":<Enter the Domain Id>
}
```

Sample Request

```
http://127.0.0.1:8981/odata/api/harvesters
```

Sample Payload

```
{
"HarvesterAddress": "0.0.0.7",
"Description":"Testing Purpose",
"DomainId": 1
}
```

Sample Response

```
{
"@odata.context": "$metadata#harvesters",
"HarvesterId": 9,
"HarvesterAddress": "0.0.0.7",
"Description": "Testing Purpose",
"ManagementServerPort": null,
"FlowEnabledLastModified": null,
"FlowEnabledLastDeployed": null,
"LastPushCheckpoint": null,
"LastPullCheckpoint": null,
"ApplicationMappingLastDeployed": null,
"PortDefinitionsLastDeployed": null,
"ReservedSeatingLastDeployed": null,
"TrapDefinitionsLastDeployed": null,
"DomainId": 1,
"DomainName": null,
```

```
 "DomainDescription": null,
 "TenantId": null
}
```

Edit Harvester

NOTE

Harvester does not allow bulk edit option, hence you must edit one harvester at a time.

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/harvesters(HarvesterId=<Value of the harvester ID>)
```

Method

PATCH

Payload

```
{
"Description":"<Enter the Description>",
"DomainId":<Enter the Domain Id>
}
```

Sample Request

```
http://127.0.0.1:8981/odata/api/harvesters(HarvesterId=33)
```

Sample Payload

```
{
"Description":"Single box configuration Modified",
"DomainId": 3
}
```

Sample Response

There is no response for this API.

Delete Harvester

NOTE

You can delete only one harvester at a time.

Request URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/harvesters(HarvesterId=<Value of the harvester ID>)
```

Method

DELETE

Sample Request

```
http://127.0.0.1:8981/odata/api/harvesters(HarvesterId=33)
```

Sample Response

There is no response for this API.

Application Settings

The following are the options available in application setting:

More Information: View System Status

Read All Application Setting

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/applicationSettings
```

Method

GET

Sample Request

```
http://127.0.0.1:8981/odata/api/applicationSettings
```

Sample Response


```
{
 "@odata.context": "$metadata#applicationSettings",
 "value": [
 {
 "Parameter": "agentAWOLLimit",
 "Label": "Interface Data Absence Limit",
 "Value": "4h"
 },
 {
 "Parameter": "AppMap_TCPRebasePort",
 "Label": "TCP Rebase Port",
 "Value": "9000"
 },
 {
 "Parameter": "AppMap_TOSMask",
 "Label": "ToS Mask",
 "Value": "255"
```

```
 },
 {
 "Parameter": "AppMap_UDPRebasePort",
 "Label": "UDP Rebase Port",
 "Value": "8000"
 },
 {
 "Parameter": "autoEnableInterfaces",
 "Label": "Auto-Enable Interfaces",
 "Value": "True"
 },
 {
 "Parameter": "defaultTZ",
 "Label": "Default Time Zone",
 "Value": "GMT"
 },
 {
 "Parameter": "DNSDomains",
 "Label": "DNS Domains",
 "Value": ""
 },
 {
 "Parameter": "dropToZero",
 "Label": "Show Trendline Zeroes",
 "Value": "False"
 },
 {
 "Parameter": "emailFromAddress",
 "Label": "From Address",
 "Value": "<no default>"
 },
 {
 "Parameter": "emailSMTPServer",
 "Label": "SMTP Server",
 "Value": "<no default>"
 }
 ],
 "@odata.nextLink": "http://localhost:8981/odata/api/applicationSettings?$skiptoken=10"
}
```


Read a Specific Application Setting

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/applicationSettings('<Application
 Setting's Parameter name>')
```

Method

GET

Sample Request

```
http://127.0.0.1:8981/odata/api/applicationSettings('AppMap_TCPRebasePort')
```

Sample Response

```
{
 "@odata.context": "$metadata#applicationSettings",
 "Parameter": "AppMap_TCPRebasePort",
 "Label": "TCP Rebase Port",
 "Value": "9000"
}
```

Edit Application Setting

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/applicationSettings('<Application
 Setting's Parameter name>')
```

Method

PATCH

Payload

```
{

   "Value": "<Enter the new value>"

}
```

Sample Request

```
http://127.0.0.1:8981/odata/api/applicationSettings('showNotes')
```

Sample Payload

```
{

   "Value": "False"
```

```
}
```

Sample Response

There is no response for this API.

Health

Using the DX NetOps you can find the health of the network that you monitor. The following options are available in Health:

Reporter Health

The Reporter Health displays the important information related to reporter console. It tell about the DB connection status and related metrics like DB tables, DB repairs, and DB failures. It also tells about the pump service status, RIB service status, SSO status, SNMP Status, Memory Utilization, and report status.

Display Reporter Health

The following API displays reporter health data.

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/reporterHealth
```

Method

GET

Sample Request

```
http://127.0.0.1:8981/odata/api/reporterHealth
```

Sample Response

```
{
 "@odata.context": "$metadata#reporterHealth",
 "value": [
 {
 "UpdateTime": 1547195785,
 "SNMPStatus": "Failed",
 "MemoryUtilization": 0,
 "DBConnectStatus": "Successful",
 "GeneralServiceStatus": "Unknown",
 "PumpServiceStatus": "Unknown",
 "QueryServiceStatus": "Unknown",
 "ReportServiceStatus": "Unknown",
```

```
 "RibServiceStatus": "Unknown",
 "SSOServiceStatus": "Unknown"
 }
 ]
}
```

Harvester Health

The harvester health shows the different metrics of the network that you monitor. Th following options are available in Harvester Health:

Display All Harvester Health

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/harvesterHealth
```

Method

GET

Sample Request

```
http://127.0.0.1:8981/odata/api/harvesterHealth
```

Sample Response

```
{
 "@odata.context": "$metadata#harvesterHealth",
 "value": [
 {
 "PollerDBRepairs": 0,
 "CollectorPollerWSStatus": "Unknown",
 "DBConnectStatus": "Successful",
 "DataRetentionDBConnectStatus": "Successful",
 "DataRetentionDBFailures": 0,
 "DBTables": 38,
 "LastBilling": 0,
 "ProxyServiceStatus": "Unknown",
 "PollerDBTables": 38,
 "DBRepairs": 0,
 "PollerServiceStatus": "Unknown",
 "PollerDBConnectStatus": "Successful",
 "HarvestStatus": "Unknown",
 "SNMPServiceStatus": "Failed",
 "DataRetentionDBTables": 3,
```

```
 "DataRetentionServiceStatus": "Unknown",
 "DataRetentionDBRepairs": 0,
 "LastHarvester": 0,
 "HarvesterAddress": "127.0.0.1",
 "WebServerStatus": "Unknown",
 "PollerDBFailures": 0,
 "UpdateTime": 1556711685,
 "MemoryUtilization": "0.0",
 "HarvesterId": 0,
 "DBFailures": 0
 }
 ]
}
```


Display Specific Harvester Health

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/harvesterHealth(HarvesterId=<Value of the Harvester ID>)
```

Method

GET

Sample Request

```
http://127.0.0.1:8981/odata/api/harvesterHealth(HarvesterId=2)
```

Sample Response


```
{
 "@odata.context": "$metadata#harvesterHealth",
 "HarvesterId": 2,
 "HarvesterAddress": "127.0.0.1",
 "SNMPServiceStatus": "Failed",
 "DBTables": 37,
 "UpdateTime": 1545213117,
 "LastHarvester": 0,
 "LastBilling": 0,
 "DBRepairs": 0,
 "DBFailures": 0,
 "PollerDBTables": 37,
 "PollerDBRepairs": 0,
 "PollerDBFailures": 0,
 "DataRetentionDBTables": 3,
```

```
 "DataRetentionDBRepairs": 0,
 "DataRetentionDBFailures": 0,
 "HarvestStatus": "Unknown",
 "WebServerStatus": "Unknown",
 "CollectorPollerWSStatus": "Unknown",
 "ProxyServiceStatus": "Unknown",
 "DataRetentionServiceStatus": "Unknown",
 "PollerServiceStatus": "Unknown",
 "DBConnectStatus": "Successful",
 "PollerDBConnectStatus": "Successful",
 "DataRetentionDBConnectStatus": "Successful",
 "MemoryUtilization": "0.0"
}
```


Display Harvester Health Associated with All Harvester

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/harvesters?$expand=harvesterHealth
```

Method

GET

Sample Request

```
http://127.0.0.1:8981/odata/api/harvesters?$expand=harvesterHealth
```

Sample Response


```
{
 "@odata.context": "$metadata#harvesters(harvesterHealth())",
 "value": [
 {
 "HarvesterId": 2,
 "LastPullCheckpoint": 1545213745,
 "PortDefinitionsLastDeployed": 0,
 "HarvesterAddress": "127.0.0.1",
 "DomainId": 3,
 "FlowEnabledLastModified": 0,
 "ManagementServerPort": 8080,
 "DomainName": "Domain A",
 "DomainDescription": "Domain A for FellsCargo",
 "FlowEnabledLastDeployed": 0,
 "LastPushCheckpoint": 1545213749,
 "ApplicationMappingLastDeployed": 0,
```

```
 "ReservedSeatingLastDeployed": 0,
 "TrapDefinitionsLastDeployed": 1541748300,
 "TenantId": 9,
 "Description": "Single box configuration Modified",
 "harvesterHealth": [
 {
 "PollerDBRepairs": 0,
 "CollectorPollerWSStatus": "Unknown",
 "DBConnectStatus": "Successful",
 "DataRetentionDBConnectStatus": "Successful",
 "DataRetentionDBFailures": 0,
 "DBTables": 37,
 "LastBilling": 0,
 "ProxyServiceStatus": "Unknown",
 "PollerDBTables": 37,
 "DBRepairs": 0,
 "PollerServiceStatus": "Unknown",
 "PollerDBConnectStatus": "Successful",
 "HarvestStatus": "Unknown",
 "SNMPServiceStatus": "Failed",
 "DataRetentionDBTables": 3,
 "DataRetentionServiceStatus": "Unknown",
 "DataRetentionDBRepairs": 0,
 "LastHarvester": 0,
 "HarvesterAddress": "127.0.0.1",
 "WebServerStatus": "Unknown",
 "PollerDBFailures": 0,
 "UpdateTime": 1545213117,
 "MemoryUtilization": "0.0",
 "HarvesterId": 2,
 "DBFailures": 0
 }
 ]
 },

{
 "HarvesterId": 9,
 "LastPullCheckpoint": 0,
 "PortDefinitionsLastDeployed": 0,
 "HarvesterAddress": "127.0.0.1",
 "DomainId": 1,
 "FlowEnabledLastModified": 0,
 "ManagementServerPort": 8080,
 "DomainName": "Default Domain",
 "DomainDescription": "The default domain for devices, interfaces, interface addresses
 and networks.",
 "FlowEnabledLastDeployed": 0,
```

```
 "LastPushCheckpoint": 0,
 "ApplicationMappingLastDeployed": 0,
 "ReservedSeatingLastDeployed": 0,
 "TrapDefinitionsLastDeployed": 0,
 "TenantId": 8,
 "Description": "Testing Purpose",
 "harvesterHealth": []
 }
 ]
}
```


Display Harvester Health Associated with a Specific Harvester

Resource URI

```
http://<nfa odata host>:<nfa odata port>odata/api/harvesters(HarvesterId=<Harvester ID
 value>)?$expand=harvesterHealth
```

Method

GET

Sample Request

```
http://127.0.0.1:8981/odata/api/harvesters(HarvesterId=2)?$expand=harvesterHealth
```

Sample Response


```
{

"@odata.context": "$metadata#harvesters(harvesterHealth())",
 "HarvesterId": 2,
 "HarvesterAddress": "127.0.0.1",
 "Description": "Single box configuration Modified",
 "ManagementServerPort": 8080,
 "FlowEnabledLastModified": 0,
 "FlowEnabledLastDeployed": 0,
 "LastPushCheckpoint": 1545214055,
 "LastPullCheckpoint": 1545214053,
 "ApplicationMappingLastDeployed": 0,
 "PortDefinitionsLastDeployed": 0,
 "ReservedSeatingLastDeployed": 0,
 "TrapDefinitionsLastDeployed": 1541748300,
 "DomainId": 3,
 "DomainName": "Domain A",
 "DomainDescription": "Domain A for FellsCargo",
```

```
 "TenantId": 9,
 "harvesterHealth": [
 {
 "HarvesterId": 2,
 "HarvesterAddress": "127.0.0.1",
 "SNMPServiceStatus": "Failed",
 "DBTables": 37,
 "UpdateTime": 1545213117,
 "LastHarvester": 0,
 "LastBilling": 0,
 "DBRepairs": 0,
 "DBFailures": 0,
 "PollerDBTables": 37,
 "PollerDBRepairs": 0,
 "PollerDBFailures": 0,
 "DataRetentionDBTables": 3,
 "DataRetentionDBRepairs": 0,
 "DataRetentionDBFailures": 0,
 "HarvestStatus": "Unknown",
 "WebServerStatus": "Unknown",
 "CollectorPollerWSStatus": "Unknown",
 "ProxyServiceStatus": "Unknown",
 "DataRetentionServiceStatus": "Unknown",
 "PollerServiceStatus": "Unknown",
 "DBConnectStatus": "Successful",
 "PollerDBConnectStatus": "Successful",
 "DataRetentionDBConnectStatus": "Successful",
 "MemoryUtilization": "0.0"
 }
 ]
}
```


http://10.13.91.413:8981/odata/api/harvesters(HarvesterId=2)?$expand=harvesterHealth

Watchdog Settings

The following options are available in Watchdog Settings:

> **NOTE**
>
> **More Information:**
>
> How to Monitor the Components

Read All Watchdog data

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/watchDogSettings
```

Method

GET

Sample Request

```
http://127.0.0.1:8981/odata/api/watchDogSettings
```

Sample Response

```
{
 "@odata.context": "$metadata#watchDogSettings",
 "value": [
 {
 "Parameter": "watchdogCommunity",
 "Label": "Community String",
 "Value": "public"
 },
 {
 "Parameter": "watchdogCPUThreshold",
 "Label": "CPU Threshold",
 "Value": "80"
 },
 {
 "Parameter": "watchdogDiskThreshold",
 "Label": "Disk Threshold",
 "Value": "80"
 },
 {
 "Parameter": "watchdogEmailDest",
 "Label": "Email Address",
 "Value": "<no default>"
 },
 {
 "Parameter": "watchdogMemoryThreshold",
 "Label": "Memory Threshold",
 "Value": "80"
 },
 {
 "Parameter": "watchdogSnmpRetry",
 "Label": "SNMP Retries",
 "Value": "2"
 },
 {
```

```
 "Parameter": "watchdogSnmpTimeout",
 "Label": "SNMP Timeout",
 "Value": "5"
 },
 {
 "Parameter": "watchdogSystemCheck",
 "Label": "System Check Interval",
 "Value": "60"
 },
 {
 "Parameter": "watchdogTrapCommunity",
 "Label": "Trap Community String",
 "Value": "public"
 },
 {
 "Parameter": "watchdogTrapDestination",
 "Label": "Trap Destination",
 "Value": "10.10.10.101"
 }
 ]
}
```


Read a Specific Watchdog

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/watchDogSettings('<Watchdog setting's
 Parameter value>')
```

Method

GET

Sample Request

```
http://127.0.0.1:8981/odata/api/watchDogSettings('watchdogCommunity')
```

Sample Response


```
{
 "@odata.context": "$metadata#watchDogSettings",
 "Parameter": "watchdogCommunity",
 "Label": "Community String",
 "Value": "public"
}
```

Edit Watchdog

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/watchDogSettings('<Watchdog setting's
 Parameter value>')
```

Method

PATCH

Payload

```
{
"Value": "<Enter the new value of the Watchdog setting's Parameter>"
}
```

Sample Request

```
http://127.0.0.1:8981/odata/api/watchDogSettings('watchdogTrapDestination')
```

Sample Payload

```
{
"Value": "127.0.0.1"
}
```

Sample Response

There is no response for this API.

Flow Statistics

Flow Statistics determines the overall load for each harvester. The flow statistics allows you to make decisions about where new flows should be sent, whether rebalancing is needed. You can view the flow rates on harvesters for the last day, last week, or last 30 days.

NOTE

More Information:

View Flow Statistics

The following options available in Flow Statistics:

Display All Flow Statistic

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/flowStats
```

Method

GET

Sample Request

```
http://127.0.0.1:8981/odata/api/flowStats
```

Sample Response


```
{
 "@odata.context": "$metadata#flowStats",
 "value": [
 {
 "RouterReboots": 0,
 "DroppedPackets": 0,
 "MapFailures": 0,
 "HeaderFailures": 0,
 "HarvesterId": 2,
 "FlowRate": 3222,
 "Created": "2018-11-15 00:02:35.0",
 "PointTime": 1542240155,
 "DroppedFlows": 0,
 "NoFlowsFound": 0
 },
 {
 "RouterReboots": 0,
 "DroppedPackets": 0,
 "MapFailures": 0,
 "HeaderFailures": 0,
 "HarvesterId": 2,
 "FlowRate": 3222,
 "Created": "2018-11-15 00:07:35.0",
 "PointTime": 1542240455,
 "DroppedFlows": 0,
 "NoFlowsFound": 0
 }

]

}
```

Display Flow Statistic Associated for a Specific Harvester

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/harvesters(HarvesterId=<Harvester Id Value>)?$expand=flowStats($count=true)
```

Method

GET

Sample Request

```
http://127.0.0.1:8981/odata/api/harvesters(HarvesterId=2)?$expand=flowStats($count=true)
```

Sample Response


```
{
 "@odata.context": "$metadata#harvesters(flowStats())",
 "HarvesterId": 2,
 "HarvesterAddress": "127.0.0.1",
 "Description": "Single box configuration Modified",
 "ManagementServerPort": 8080,
 "FlowEnabledLastModified": 0,
 "FlowEnabledLastDeployed": 0,
 "LastPushCheckpoint": 1545213322,
 "LastPullCheckpoint": 1545213313,
 "ApplicationMappingLastDeployed": 0,
 "PortDefinitionsLastDeployed": 0,
 "ReservedSeatingLastDeployed": 0,
 "TrapDefinitionsLastDeployed": 1541748300,
 "DomainId": 3,
 "DomainName": "Domain A",
 "DomainDescription": "Domain A for FellsCargo",
 "TenantId": 9,
 "flowStats@odata.count": 10,
 "flowStats": [
 {
 "HarvesterId": 2,
 "Created": "2018-11-15 00:02:35.0",
 "PointTime": 1542240155,
 "FlowRate": 3222,
 "DroppedFlows": 0,
 "DroppedPackets": 0,
 "MapFailures": 0,
```

```
"NoFlowsFound": 0,
"HeaderFailures": 0,
"RouterReboots": 0
},
{
"HarvesterId": 2,
"Created": "2018-11-15 00:07:35.0",
"PointTime": 1542240455,
"FlowRate": 3222,
"DroppedFlows": 0,
"DroppedPackets": 0,
"MapFailures": 0,
"NoFlowsFound": 0,
"HeaderFailures": 0,
"RouterReboots": 0
},
{
"HarvesterId": 2,
"Created": "2018-11-15 00:12:35.0",
"PointTime": 1542240755,
"FlowRate": 3222,
"DroppedFlows": 0,
"DroppedPackets": 0,
"MapFailures": 0,
"NoFlowsFound": 0,
"HeaderFailures": 0,
"RouterReboots": 0
},
{
"HarvesterId": 2,
"Created": "2018-11-15 00:17:35.0",
"PointTime": 1542241055,
"FlowRate": 3222,
"DroppedFlows": 0,
"DroppedPackets": 0,
"MapFailures": 0,
"NoFlowsFound": 0,
"HeaderFailures": 0,
"RouterReboots": 0
},
{
"HarvesterId": 2,
"Created": "2018-11-15 00:22:35.0",
"PointTime": 1542241355,
"FlowRate": 3222,
"DroppedFlows": 0,
```

```
 "DroppedPackets": 0,
 "MapFailures": 0,
 "NoFlowsFound": 0,
 "HeaderFailures": 0,
 "RouterReboots": 0
 },
 {
 "HarvesterId": 2,
 "Created": "2018-11-15 00:27:35.0",
 "PointTime": 1542241655,
 "FlowRate": 3222,
 "DroppedFlows": 0,
 "DroppedPackets": 0,
 "MapFailures": 0,
 "NoFlowsFound": 0,
 "HeaderFailures": 0,
 "RouterReboots": 0
 }

]

}
```


Display Flow Statistics Associated with All Harvester

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/harvesters?
$expand=flowStats($count=true)
```

Method

GET

Sample Request

```
http://127.0.0.1:8981/odata/api/harvesters?$expand=flowStats($count=true)
```

Sample Response


```
{
 "@odata.context": "$metadata#harvesters(flowStats())",
 "value": [
 {
```

```
"HarvesterId": 2,
"LastPullCheckpoint": 1545214484,
"PortDefinitionsLastDeployed": 0,
"HarvesterAddress": "127.0.0.1",
"DomainId": 3,
"FlowEnabledLastModified": 0,
"ManagementServerPort": 8080,
"DomainName": "Domain A",
"DomainDescription": "Domain A for FellsCargo",
"FlowEnabledLastDeployed": 0,
"LastPushCheckpoint": 1545214483,
"ApplicationMappingLastDeployed": 0,
"ReservedSeatingLastDeployed": 0,
"TrapDefinitionsLastDeployed": 1541748300,
"TenantId": 9,
"Description": "Single box configuration Modified",
"flowStats@odata.count": 10,
"flowStats": [
{
"RouterReboots": 0,
"DroppedPackets": 0,
"MapFailures": 0,
"HeaderFailures": 0,
"HarvesterId": 2,
"FlowRate": 3222,
"Created": "2018-11-15 00:02:35.0",
"PointTime": 1542240155,
"DroppedFlows": 0,
"NoFlowsFound": 0
},
{
"RouterReboots": 0,
"DroppedPackets": 0,
"MapFailures": 0,
"HeaderFailures": 0,
"HarvesterId": 2,
"FlowRate": 3222,
"Created": "2018-11-15 00:07:35.0",
"PointTime": 1542240455,
"DroppedFlows": 0,
"NoFlowsFound": 0
},
{
"RouterReboots": 0,
"DroppedPackets": 0,
"MapFailures": 0,
```

```
 "HeaderFailures": 0,
 "HarvesterId": 2,
 "FlowRate": 3222,
 "Created": "2018-11-15 00:12:35.0",
 "PointTime": 1542240755,
 "DroppedFlows": 0,
 "NoFlowsFound": 0
 },
 {
 "RouterReboots": 0,
 "DroppedPackets": 0,
 "MapFailures": 0,
 "HeaderFailures": 0,
 "HarvesterId": 2,
 "FlowRate": 3222,
 "Created": "2018-11-15 00:17:35.0",
 "PointTime": 1542241055,
 "DroppedFlows": 0,
 "NoFlowsFound": 0
 }
 ]
 },
 {
 "HarvesterId": 5,
 "LastPullCheckpoint": 0,
 "PortDefinitionsLastDeployed": 0,
 "HarvesterAddress": "127.0.0.1",
 "DomainId": 1,
 "FlowEnabledLastModified": 0,
 "ManagementServerPort": 8080,
 "DomainName": "Default Domain",
 "DomainDescription": "The default domain for devices, interfaces, interface addresses
 and networks.",
 "FlowEnabledLastDeployed": 0,
 "LastPushCheckpoint": 0,
 "ApplicationMappingLastDeployed": 0,
 "ReservedSeatingLastDeployed": 0,
 "TrapDefinitionsLastDeployed": 0,
 "TenantId": 8,
 "Description": "Test Harvesters2",
 "flowStats@odata.count": 0,
 "flowStats": []
 }
 ]
}
```

Display Flow Statistic and Harvester Health associated to a Specific Harvester

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/harvesters(HarvesterId=<Harvester Id Value>)?$expand=flowStats($count=true),harvesterHealth
```

Method

GET

Sample Request

```
http://127.0.0.1:8981/odata/api/harvesters(HarvesterId=2)?$expand=flowStats($count=true),harvesterHealth
```

Sample Response


```
{
 "@odata.context": "$metadata#harvesters(harvesterHealth(),flowStats())",
 "HarvesterId": 2,
 "HarvesterAddress": "127.0.0.1",
 "Description": "Single box configuration Modified",
 "ManagementServerPort": 8080,
 "FlowEnabledLastModified": 0,
 "FlowEnabledLastDeployed": 0,
 "LastPushCheckpoint": 1545213383,
 "LastPullCheckpoint": 1545213375,
 "ApplicationMappingLastDeployed": 0,
 "PortDefinitionsLastDeployed": 0,
 "ReservedSeatingLastDeployed": 0,
 "TrapDefinitionsLastDeployed": 1541748300,
 "DomainId": 3,
 "DomainName": "Domain A",
 "DomainDescription": "Domain A for FellsCargo",
 "TenantId": 9,
 "harvesterHealth": [
 {
 "HarvesterId": 2,
 "HarvesterAddress": "127.0.0.1",
 "SNMPServiceStatus": "Failed",
 "DBTables": 37,
 "UpdateTime": 1545213117,
 "LastHarvester": 0,
 "LastBilling": 0,
 "DBRepairs": 0,
 "DBFailures": 0,
```

```
"PollerDBTables": 37,
"PollerDBRepairs": 0,
"PollerDBFailures": 0,
"DataRetentionDBTables": 3,
"DataRetentionDBRepairs": 0,
"DataRetentionDBFailures": 0,
"HarvestStatus": "Unknown",
"WebServerStatus": "Unknown",
"CollectorPollerWSStatus": "Unknown",
"ProxyServiceStatus": "Unknown",
"DataRetentionServiceStatus": "Unknown",
"PollerServiceStatus": "Unknown",
"DBConnectStatus": "Successful",
"PollerDBConnectStatus": "Successful",
"DataRetentionDBConnectStatus": "Successful",
"MemoryUtilization": "0.0"
}
],
"flowStats@odata.count": 10,
"flowStats": [
{
"HarvesterId": 2,
"Created": "2018-11-15 00:02:35.0",
"PointTime": 1542240155,
"FlowRate": 3222,
"DroppedFlows": 0,
"DroppedPackets": 0,
"MapFailures": 0,
"NoFlowsFound": 0,
"HeaderFailures": 0,
"RouterReboots": 0
},
{
"HarvesterId": 2,
"Created": "2018-11-15 00:07:35.0",
"PointTime": 1542240455,
"FlowRate": 3222,
"DroppedFlows": 0,
"DroppedPackets": 0,
"MapFailures": 0,
"NoFlowsFound": 0,
"HeaderFailures": 0,
"RouterReboots": 0
},
{
"HarvesterId": 2,
```

```
 "Created": "2018-11-15 00:12:35.0",
 "PointTime": 1542240755,
 "FlowRate": 3222,
 "DroppedFlows": 0,
 "DroppedPackets": 0,
 "MapFailures": 0,
 "NoFlowsFound": 0,
 "HeaderFailures": 0,
 "RouterReboots": 0
 }
 ]
}
```


Application Mapping

DX NetOps 10.0.1 supports the application mapping API.

Using the application mapping, you can discover and map all your entities and their inter-dependencies. DX NetOps supports the ToS, host, and subnet applications. The Application Mapping API lets you manage the entities in the following ways:

More Information: Set Up Application Mapping

Application Mappings

Read Single Entity

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/applicationMappings(applicationMappings_Id)
```

Method

GET

Sample Request

```
http://127.0.0.1:8981/odata/api/applicationMappings(108)
```

Sample Response


```
{ "@odata.context": "$metadata#applicationMappings",
"protocol": -1,
"description": "11",
"nbar2EngineId": -1,
"endPort": -1,
"nbar2ApplicationId": -1,
"Name": "test",
"ip": "127.0.0.1",
"tos": 12,
"newPort": 11,
"mask": 0,
```

```
"ruleType": "Tos",
"ID": 108,
"beginPort": -1
}
```


Read Bulk Entity

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/applicationMappings
```

Method

GET

Sample Request

```
http://127.0.0.1:8981/odata/api/applicationMappings
```

Sample Response


```
{ "@odata.context": "http://localhost:8981/odata/api/$metadata#applicationMappings",
"value":
[
{
 "protocol": -1,
 "description": "11",
 "nbar2EngineId": -1,
 "endPort": -1,
 "nbar2ApplicationId": -1,
 "Name": "test",
 "ip": "127.0.0.1",
 "tos": 12,
 "newPort": 11,
 "mask": 0,
 "ruleType": "Tos",
 "ID": 108,
 "beginPort": -1 },
 { "protocol": -1,
 "description": "",
 "nbar2EngineId": -1,
 "endPort": -1,
 "nbar2ApplicationId": -1,
 "Name": "test",
 "ip": "127.0.0.1",
 "tos": 1,
 "newPort": 11,
 "mask": 0,
 "ruleType": "Tos",
 "ID": 109,
 "beginPort": -1
```

```
}
]
}
```

Create ToS (Type of Service)

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/applicationMappings
```

Method

POST

Payload

```
{
"ruleType" : "<Enter the rule type>",
"Name" : "<Enter the application mapping name>",
"newPort" : "<Enter a value between 0 and 65535>",
"tos" : "<Enter the ToS>"
}
```

Sample Request

```
http://127.0.0.1:8981/odata/api/applicationMappings
```

Sample Payload

```
{
"ruleType" : "All",
"Name" : "Tes1t",
"newPort" :23457,
"tos" : 66
}
```

Sample Response

```
{
 "@odata.context": "$metadata#applicationMappings",
 "ID": 150,
 "description": null,
 "protocol": null,
 "tos": 66,
 "ip": null,
 "mask": null,
 "beginPort": null,
 "endPort": null,
 "newPort": 23457,
 "nbar2EngineId": null,
 "nbar2ApplicationId": null,
 "ruleType": "All",
 "Name": "Tes1t"
```

```
}
```

Create Host

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/applicationMappings
```

Method

POST

Example 1

Payload

```
{
"ruleType" : "<Enter the rule type>",
"Name" : "<Enter the application mapping name>",
"newPort" : "<Enter a value between 0 and 65535>",
"beginPort" : "<Enter a value between -1 and 255>>",
"tos" : "<Enter a value between -1 and 255>",
"protocol" : "<Enter-1>",
"ip" : "<Enter the Host IP address>"
}
```

Sample Request

```
http://127.0.0.1:8981/odata/api/applicationMappings
```

Sample Payload

```
{
 "ruleType" : "Host",
 "Name" : "Test22",
 "newPort" :123,
 "beginPort" :-1,
 "tos" : -1,
 "protocol" : -1,
 "ip" : "127.0.0.1"
}
```

Sample Response

```
{
 "@odata.context": "$metadata#applicationMappings",
 "ID": 152,
 "description": null,
 "protocol": -1,
 "tos": -1,
 "ip": "127.0.0.1",
 "mask": null,
 "beginPort": -1,
 "endPort": null,
```

```
 "newPort": 123,
 "nbar2EngineId": null,
 "nbar2ApplicationId": null,
 "ruleType": "Host",
 "Name": "Test22"
}
```

Example 2

Payload

```
{
"ruleType" : "<Enter the rule type>",
"Name" : "<Enter the rule name>",
"newPort" : "<Enter a value between 0 and 65535>",
"beginPort" : "<Enter a value between 0 and 65535>",
"tos" : "<Enter a value between -1 and 255>",
"protocol" : "<Enter 6 or 17>",
"ip" : "<Enter the Host IP address>"
}
```

Sample Request

```
http://127.0.0.1:8981/odata/api/applicationMappings
```

Sample Payload

```
{
"ruleType" : "Host",
"Name" : "Test211",
"newPort" :888,
"beginPort" :211,
"tos" : -1,
"protocol": 6,
"ip": "127.0.0.1"
}
```

Sample Response

```
{
"ID": 155,
"description": null,
"protocol": 6,
"tos": -1,
"ip": "127.0.0.1",
"mask": null,
"beginPort": 211,
"endPort": null,
"newPort": 888,
"nbar2EngineId": null,
"nbar2ApplicationId": null,
"ruleType": "Host",
```

```
"Name": "Test211"
}
```

Create Subnet

For Subnet app mapping of type TCP, protocol value must be 6 and for UDP protocol value must be 17.

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/applicationMappings
```

Method

POST

Payload

```
{
"ruleType" : "<Enter the rule type>",
"Name" : "<Enter the application mapping name>",
"newPort" : "<Enter a value between 0 and 65535>",
"beginPort" : "<Enter a value between 0 and 65535>",
"endPort" : "<Enter a value between 0 and 65535>",
"protocol" : "<Enter 6 or 17>",
"ip" : "<Enter the Host IP address>"
"mask": "<Enter a value between 0 and 32>"
}
```

Sample Request

```
http://127.0.0.1:8981/odata/api/applicationMappings
```

Sample Payload

```
{
"ruleType" : "Subnet",
"Name" : "Test4",
"newPort" :23411,
"beginPort" :255,
"endPort" : 255,
"protocol": 6,
"ip": "127.0.0.1",
"mask": 22
}
```

Sample Response

```
{
 "@odata.context": "$metadata#applicationMappings",
 "ID": 156,
 "description": null,
 "protocol": 6,
 "tos": null,
 "ip": "127.0.0.1",
```

```
 "mask": 22,
 "beginPort": 255,
 "endPort": 255,
 "newPort": 23411,
 "nbar2EngineId": null,
 "nbar2ApplicationId": null,
 "ruleType": "Subnet",
 "Name": "Test4"
}
```

Edit ToS (Type of Service)

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/
applicationMappings(applicationMappings_id)
```

Method

PATCH

Payload

```
{
"ruleType" : "<Enter the rule type>",
"Name" : "<Enter the application mapping name>",
"newPort" : "<Enter a value between 0 and 65535>",
"tos" : "<Enter the ToS>"
}
```

Sample Request

```
{
"ruleType" : "All",
"Name" : "Tes1t",
"newPort" :23457,
"tos" : 66
}
```

Sample Response

There is no response for this API.

Edit Host

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/
applicationMappings(applicationMapping_id)
```

Method

PATCH

Example 1

Payload

```
{
"ruleType" : "<Enter the rule type>",
"Name" : "<Enter the application mapping name>",
"newPort" : "<Enter a value between 0 and 65535>",
"beginPort" : "<Enter a value between -1 and 255>>",
"tos" : "<Enter a value between -1 and 255>",
"protocol" : "<Enter-1>",
"ip" : "<Enter the Host IP address>"
}
```

Sample Request

```
{
"ruleType" : "Host",
"Name" : "Test22",
"newPort" :23457,
"beginPort" :-1,
"tos" : -1,
"protocol" : -1,
"ip" : "127.0.0.1"
}
```

Sample Response

There is no response for this API.

Example 2

Payload

```
{
"ruleType" : "<Enter the rule type>",
"Name" : "<Enter the rule name>",
"newPort" : "<Enter a value between 0 and 65535>",
"beginPort" : "<Enter a value between 0 and 65535>",
"tos" : "<Enter a value between -1 and 255>",
"protocol" : "<Enter 6 or 17>",
"ip" : "<Enter the Host IP address>"
}
```

Sample Request

```
{
"ruleType" : "Host",
"Name" : "Test2",
"newPort" :23457,
"beginPort" :255,
"tos" : -1,
"protocol": 6,
"ip": "127.0.0.1"
```

```
}
```

Sample Response

There is no response for this API.

Edit Subnet

For the Subnet app mapping of type TCP, the protocol value must be 6 and for UDP protocol value must be 17.

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/
applicationMappings(applicationMapping_id)
```

Method

PATCH

Payload

```
{
"ruleType" : "<Enter the rule type>",
"Name" : "<Enter the application mapping name>",
"newPort" : "<Enter a value between 0 and 65535>",
"beginPort" : "<Enter a value between 0 and 65535>",
"endPort" : "<Enter a value between 0 and 65535>",
"protocol" : "<Enter 6 or 17>",
"ip" : "<Enter the Host IP address>"
"mask": "<Enter a value between 0 and 32>"
}
```

Sample Request

```
{
"ruleType" : "Subnet",
"Name" : "Test4",
"newPort" :23411,
"beginPort" :255,
"endPort" : 255,
"protocol": 6,
"ip": "127.0.0.1",
"mask": 22
}
```

Sample Response

There is no response to this API.

Delete Single Entity

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/
applicationMappings(applicationMappings_id)
```

Method

DELETE

Payload

```
{
"ApplicationMapping_Id": [<Enter id of the application mapping that should be deleted>]
}
```

Sample Request

```
http://127.0.0.1:8981/odata/api/applicationMappings(108)
```

Sample Response

There is no response to this API.

Delete Bulk Entities

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/applicationMappings/
com.ca.nfa.odata.removeApplicationMappings
```

Method

POST

Payload

```
{
ApplicationMappingIds: [<Enter the list of ApplicationMapping_Ids separated by a comma>]
}
```

Sample Request

The following request illustrates the bulk delete of application mapping entities.

```
http://127.0.0.1:8080/odata/api/routers/com.ca.nfa.odata.removeApplicationMappings
```

Sample Payload

```
{ "ApplicationMappingIds":[4, 5] }
```

Sample Response

```
{
 "@odata.context": "$metadata#Collection(com.ca.nfa.odata.applicationMappings)",
 "value": [
 {
 "ID": 4,
 "description": "",
 "protocol": -1,
 "tos": 123,
 "ip": "127.0.0.1",
 "mask": 0,
 "beginPort": -1,
 "endPort": -1,
 "newPort": 125,
```

```
 "nbar2EngineId": -1,
 "nbar2ApplicationId": -1,
 "ruleType": "All",
 "Name": "locus-map"
 },
 {
 "ID": 5,
 "description": "",
 "protocol": -1,
 "tos": 124,
 "ip": "127.0.0.1",
 "mask": 0,
 "beginPort": -1,
 "endPort": -1,
 "newPort": 125,
 "nbar2EngineId": -1,
 "nbar2ApplicationId": -1,
 "ruleType": "All",
 "Name": "ftp"
}
 ]
}
```


Predefined Application Mappings

To get all predefined Application Mappings, use predefined application mappings API.

Read Single Entity

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/predefinedApplicationMappings
```

Method

GET

Payload

None

Sample Request

```
http://127.0.0.1:8981/odata/api/applicationMappings
```

Sample Response


```
{
    "@odata.context": "http://127.0.0.1:8981/odata/api/$metadata#predefinedApplicationMappings",
    "value": [
        {
            "newPort": 0,
            "Description": "TCP Fragments",
            "Name": "fragment",
            "ID": 630,
```

```
        "portType": "tcp"
    },
    {
        "newPort": 1,
        "Description": "TCP Port Service Multiplexer Port",
        "Name": "tcpmux",
        "ID": 68,
        "portType": "tcp"
    },
    {
        "newPort": 2,
        "Description": "Remote Job Entry Protocol",
        "Name": "rje",
        "ID": 69,
        "portType": "tcp"
    },
    {
        "newPort": 3,
        "Description": "Compression Process",
        "Name": "compressnet",
        "ID": 634,
        "portType": "tcp"
    },
    {
        "newPort": 5,
        "Description": "Remote Job Entry",
        "Name": "rje",
        "ID": 636,
        "portType": "tcp"
    },
    {
        "newPort": 7,
        "Description": "Echo Protocol",
        "Name": "echo",
        "ID": 70,
        "portType": "tcp"
    },
    {
        "newPort": 9,
        "Description": "Discard Protocol",
        "Name": "discard",
        "ID": 72,
        "portType": "tcp"
    },
    {
        "newPort": 11,
        "Description": "Active Users Protocol",
        "Name": "systat",
        "ID": 74,
        "portType": "tcp"
    },
    {
        "newPort": 13,
```

```
            "Description": "Daytime Protocol",
            "Name": "daytime",
            "ID": 76,
            "portType": "tcp"
        },
        {
            "newPort": 17,
            "Description": "Quote of the Day Protocol",
            "Name": "qotd",
            "ID": 78,
            "portType": "tcp"
        }
    ],
    "@odata.nextLink": "http://127.0.0.1:8981/odata/api/predefinedApplicationMappings?$skiptoken=10"
}
```


Create NBAR2 Application Mapping Rule

Resource URI

`http://<nfa odata host>:<nfa odata port>/odata/api/applicationMappings`

Method

POST

Payload

```
{
  "ruleType": "<Enter the rule type>",
  "Name": "<Enter the application mapping name>",
  "Description": "<Enter the mapping description>",
  "nbar2EngineId": <Enter the NBAR2 Engine ID>,
  "nbar2ApplicationId": <Enter the NBAR2 Application ID>,
  "newPort": <Enter a value between 0 and 65535>
}
```

Sample Request

`http://127.0.0.1:8981/odata/api/applicationMappings`

Sample Payload

```
{
  "ruleType": "NBAR2",
  "Name": "NBAR2_NBAR2",
  "Description": "NBAR2_NBAR2",
  "nbar2EngineId": 13,
  "nbar2ApplicationId": 664421,
  "newPort": 32324
}
```

Sample Response


```
{
"@odata.context": "$metadata#applicationMappings",
"ID": 1,
"Description": "NBAR2_NBAR2",
```

```
"protocol": null,
"tos": null,
"ip": null,
"mask": null,
"beginPort": null,
"endPort": null,
"newPort": 32324,
"nbar2EngineId": 13,
"nbar2ApplicationId": 664421,
"ruleType": "NBAR2",
"Name": "NBAR2_NBAR2"
}
```

Edit NBAR2 Application Mapping Rule

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/
applicationMappings(NBar2Application_ID)
```

Method

PATCH

Payload

```
{
  "ruleType": "<Enter the rule type>",
  "Name": "<Enter the application mapping name>",
  "Description": "<Enter the mapping description>",
  "nbar2ApplicationId": <Enter the NBAR2 Application ID>
}
```

Sample Request

```
http://127.0.0.1:8981/odata/api/applicationMappings(1)
```

Sample Payload

```
{
  "ruleType": "NBAR2",
  "Name": "NBAR2_NBAR2_EDIT",
  "Description": "NBAR2_NBAR2_EDIT",
  "nbar2ApplicationId": 664422
}
```

Sample Response

There is no response for this API.

Get Single Port Traffic Status

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/applicationMappings/
com.ca.nfa.odata.PortTrafficStatus(port=<port number>)
```

Method

GET

Sample Request

```
http://127.0.0.1:8981/odata/api/applicationMappings/
com.ca.nfa.odata.PortTrafficStatus(port=161)
```

Sample Response

```
{
"@odata.context": "http://127.0.0.1:8981/odata/api/
$metadata#com.ca.nfa.odata.portTrafficStatus",
"PortName": "snmp"
}
```

Port Priority Rules

You can set port priority values based on the actual networking requirements.

This section explains the following supported operations on port priority rules using the API:

More Information: Work with Port Priorities

Get Port Priority Rules

This API allows you to get the list of port priority rules.

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/portPriorityRules
```

Method

GET

Sample Request

```
http://127.0.0.1:8981/odata/api/portPriorityRules
```

Sample Response

```
{
    "@odata.context": "http://localhost:8981/odata/api/
$metadata#portPriorityRules(ID,portType,startPort,endPort,Description)",
    "value": [
        {
            "startPort": 1080,
            "portType": "TCP",
            "Description": "Socks",
            "ID": 1,
            "endPort": 1080
        },
        {
            "startPort": 1352,
            "portType": "TCP",
            "Description": "Lotus Notes",
            "ID": 2,
            "endPort": 1352
        },
        {
            "startPort": 1352,
            "portType": "UDP",
```

```
        "Description": "Lotus Notes",
        "ID": 3,
        "endPort": 1352
    },
    {
        "startPort": 1433,
        "portType": "TCP",
        "Description": "Microsoft SQL Server DBMS Server",
        "ID": 4,
        "endPort": 1433
    },
    {
        "startPort": 1434,
        "portType": "UDP",
        "Description": "Microsoft SQL Server DBMS Monitor",
        "ID": 5,
        "endPort": 1434
    },
    {
        "startPort": 1494,
        "portType": "UDP",
        "Description": "ICA",
        "ID": 6,
        "endPort": 1494
    },
    {
        "startPort": 1494,
        "portType": "TCP",
        "Description": "ICA",
        "ID": 7,
        "endPort": 1494
    },
    {
        "startPort": 1521,
        "portType": "TCP",
        "Description": "Oracle",
        "ID": 8,
        "endPort": 1521
    },
    {
        "startPort": 2000,
        "portType": "TCP",
        "Description": "SCCP (CM Encore)",
        "ID": 9,
        "endPort": 2002
    },
    {
        "startPort": 2049,
        "portType": "TCP",
        "Description": "NFS",
        "ID": 10,
        "endPort": 2049
    }
```

```
    ],
"@odata.nextLink": http://localhost:8981/odata/api/portPriorityRules?timeout=120&$skiptoken=10
```

Get Single Port Priority Rule

This API allows you to get a single port priority rule.

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/portPriorityRules(portPriorityRuleID)
```

Method

GET

Sample Request

```
http://127.0.0.1:8981/odata/api/portPriorityRules(168)
```

Sample Response

```
{
    "@odata.context": "$metadata#portPriorityRules",
    "ID": 168,
    "portType": "UDP",
    "startPort": 1080,
    "endPort": 1080,
    "Description": "Socks"
}
```

Create Port Priority Rules

This API allows you to create a port priority rule.

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/portPriorityRules
```

Method

POST

Payload

```
{
    "startPort": <start port value>,
    "portType": "<port type value>",
    "endPort": <end port value>,
    "Description": "<description details>"
}
```

Sample Request

```
http://127.0.0.1:8981/odata/api/portPriorityRules
```

Sample Payload

```
{
    "startPort": 1080,
    "portType": "UDP",
    "endPort": 1080,
    "Description": "Socks"
}
```

Sample Response

```
{
    "@odata.context": "$metadata#portPriorityRules",
    "ID": 168,
    "portType": "UDP",
    "startPort": 1080,
    "endPort": 1080,
    "Description": "Socks"
}
```

Update Port Priority Rules

This API allows you to update a single port priority rule. This API does not support bulk edits.

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/portPriorityRules(portPriorityRuleID)
```

Method

PATCH

Payload

```
{
    "startPort": <start port value>,
    "portType": "<port type value>",
    "endPort": <end port value>,
    "Description": "<description details>"
}
```

Sample Request

```
http://127.0.0.1:8981/odata/api/portPriorityRules(168)
```

Sample Payload

```
{
    "startPort": 1080,
    "portType": "UDP",
    "endPort": 1080,
    "Description": "Socks_Change"
}
```

Sample Response

There is no response for this API.

Delete Port Priority Rules

This API allows you to delete a single port priority rule.

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/portPriorityRules(portPriorityRuleID)
```

Method

DELETE

Sample Request

The following request illustrates the delete action for the port priority rule id 168.

```
http://127.0.0.1:8981/odata/api/portPriorityRules(168)
```

Sample Response

There is no response for this API.

Bulk Delete Port Priority Rules

The following code is a reference metadata section for deleting bulk port priority rules.

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/portPriorityRules/com.ca.nfa.deletePortPriorityRules
```

Method

POST

Payload

```
 "RuleIds":[<List of rule_ids separated by a comma>]
```

Sample Request

The following request illustrates the delete action for bulk delete port priority rules.

```
http://127.0.0.1:8981/odata/api/portPriorityRules/com.ca.nfa.deletePortPriorityRules
```

Sample Payload

```
{
    "RuleIds": [106,107]
}
```

Sample Response

```
{
    "@odata.context": "$metadata#Collection(com.ca.nfa.odata.portPriorityRules)",
    "value": [
        {
            "ID": 106,
            "portType": "TCP",
            "startPort": 1000,
            "endPort": 1050,
            "Description": "Uday3"
        },
        {
            "ID": 107,
            "portType": "TCP",
            "startPort": 1000,
            "endPort": 1010,
            "Description": ""
        }
    ]
}
```

Validation Checks

Following data validations must be considered while entering the values.

- Combination of PortType, StartPort, and EndPort should be unique.
- Values for PortType should be either ‘TCP’ or ‘UDP’.
- Values for ports should be from 0 through 65535.
- Length of description should be fewer than 120 characters.

Reserved Seating Rules

You can create Reserved Seating rules to help ensure that reports include the port and protocol combinations that interest you, regardless of traffic volume or rates.

This section explains the following supported operations on reserved seating rules using the API:

More Information: Create, Change, or Delete Reserved Seating Rules

Get Reserved Seating Rules

This API allows you to get the list of reserved seating rules.

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/reservedSeatingRules
```

Method

GET

Sample Request

```
http://127.0.0.1:8981/odata/api/reservedSeatingRules
```

Sample Response

```
{
    "@odata.context": "http://localhost:8981/odata/api/
$metadata#reservedSeatingRules(ID,protocol,port,description)",
    "value": [
        {
            "protocol": 6,
            "port": 4444,
            "description": "Reserved Seating Test",
            "ID": 3
        },
        {
            "protocol": 17,
            "port": 4444,
            "description": "Reserved Seating Test",
            "ID": 4
        },
        {
            "protocol": 6,
            "port": 8080,
            "description": "Reserved Seating Test 8080",
            "ID": 5
        },
        {
            "protocol": 17,
            "port": 8080,
```

```
            "description": "Reserved Seating Test 8080 UDP",
            "ID": 6
        }
    ]
}
```

Get Single Reserved Seating Rule

This API allows you to get a single reserved seating rule.

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/reservedSeatingRules(reservedSeatingRuleID)
```

Method

GET

Sample Request

```
http://127.0.0.1:8981/odata/api/reservedSeatingRules(4)
```

Sample Response

```
{
    "@odata.context": "http://localhost:8981/odata/api/
$metadata#reservedSeatingRules(ID,protocol,port,description)/$entity",
    "protocol": 17,
    "port": 4444,
    "description": "Reserved Seating Test",
    "ID": 4
}
```

Create Reserved Seating Rules

This API allows you to create a reserved seating rule.

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/reservedSeatingRules
```

Method

POST

Payload

```
{
    "protocol": <protocol value>,
    "port": <port value>,,
    "description": "<description details>"
}
```

Sample Request

```
http://127.0.0.1:8981/odata/api/reservedSeatingRules
```

Sample Payload

```
{
    "protocol": 17,
    "port": 8080,
    "description": "Reserved Seating Test 8080 UDP"
```

```
}
```

Sample Response

```
{
    "@odata.context": "http://localhost:8981/odata/api/
$metadata#reservedSeatingRules(ID,protocol,port,description)/$entity",
    "protocol": 17,
    "port": 8080,
    "description": "Reserved Seating Test 8080 UDP",
    "ID": 6
}
```

Update Reserved Seating Rules

This API allows you to update a single reserved seating rule. This API does not support bulk edits.

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/reservedSeatingRules(reservedSeatingRuleID)
```

Method

PATCH

Payload

```
{
    "protocol": <protocol value>,
    "port": <port value>,,
    "description": "<description details>"
}
```

Sample Request

```
http://127.0.0.1:8981/odata/api/reservedSeatingRules(5)
```

Sample Payload

```
{
    "protocol": 17,
    "port": 8080,
    "description": "Reserved Seating Test 8080 UDP Update"
}
```

Sample Response

There is no response for this API.

Delete Reserved Seating Rules

This API allows you to delete a single reserved seating rule.

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/reservedSeatingRules(reservedSeatingRuleID)
```

Method

DELETE

Sample Request

The following request illustrates the delete action for the reserved seating rule id 5.

```
http://127.0.0.1:8981/odata/api/reservedSeatingRules(5)
```

Sample Response

There is no response for this API.

Bulk Delete Reserved Seating Rules

The following code is a reference metadata section for deleting bulk reserved seating rules.

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/reservedSeatingRules/
com.ca.nfa.odata.deleteReservedSeatingRules
```

Method

POST

Payload

```
 "RuleIds":[<List of rule_ids separated by a comma>]
```

Sample Request

The following request illustrates the delete action for bulk delete reserved seating rules.

```
http://127.0.0.1:8981/odata/api/reservedSeatingRules/com.ca.nfa.odata.deleteReservedSeatingRules
```

Sample Payload

```
{
    "RuleIds": [4,5]
}
```

Sample Response

```
{
    "@odata.context": "$metadata#Collection(com.ca.nfa.odata.reservedSeatingRules)",
    "value": [
        {
            "ID": 4,
            "protocol": 17,
            "port": 4444,
            "description": "Reserved Seating Test"
        },
        {
            "ID": 5,
            "protocol": 6,
            "port": 8080,
            "description": "Reserved Seating Test 8080"
        }
    ]
}
```

Validation Checks

Following data validations must be considered while entering the values.

- Combination of protocol and port should be unique.
- Values for protocol should be either 6 for TCP and 17 or UDP.
- Values for port should be from 0 through 65535.
- Length of description should be fewer than 100 characters.

AS Names

Interface reports that show data about Autonomous System (AS) traffic typically label the AS traffic by name and number. Administrators can customize AS names to make the AS references in reports shorter or more descriptive.This section explains the following supported operations on AS Names using the API:

More Information: Customize AS Names

Get AS Names

This API allows you to get the list of AS Names.

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/asNames
```

Method

GET

Sample Request

```
http://127.0.0.1:8981/odata/api/asNames
```

Sample Response


```
{
    "@odata.context": "http://localhost:8981/odata/api/
$metadata#asNames(ASNumber,Description,isBaseDescription)",
    "value": [
        {
            "isBaseDescription": "true",
            "Description": "LVLT-1 - Level 3 Communications, Inc.",
            "ASNumber": 1
        },
        {
            "isBaseDescription": "true",
            "Description": "DCN-AS - University of Delaware",
            "ASNumber": 2
        },
        {
            "isBaseDescription": "true",
            "Description": "MIT-GATEWAYS - Massachusetts Institute of Technology",
            "ASNumber": 3
        },
        {
            "isBaseDescription": "true",
            "Description": "ISI-AS - University of Southern California",
            "ASNumber": 4
        },
        {
```

```
            "isBaseDescription": "true",
            "Description": "SYMBOLICS - Symbolics, Inc.",
            "ASNumber": 5
        },
        {
            "isBaseDescription": "true",
            "Description": "BULL-NETWORK for further information please visit http://www.bull.com",
            "ASNumber": 6
        },
        {
            "isBaseDescription": "true",
            "Description": "UK Defence Research Agency",
            "ASNumber": 7
        },
        {
            "isBaseDescription": "true",
            "Description": "RICE-AS - Rice University",
            "ASNumber": 8
        },
        {
            "isBaseDescription": "true",
            "Description": "CMU-ROUTER - Carnegie Mellon University",
            "ASNumber": 9
        },
        {
            "isBaseDescription": "true",
            "Description": "CSNET-EXT-AS - CSNET Coordination and Information Center (CSNET-CIC)",
            "ASNumber": 10
        }
    ],
    "@odata.nextLink": "http://localhost:8981/odata/api/asNames?$skiptoken=10"
}
```


NOTE

If you do not provide domainId in the request, default domainId 1 is considered and AS Names for that domain are returned.

To get the AS Names for a different domain other than the default domain, add the domainId parameter in the request.

URI: http://127.0.0.1:8981/odata/api/asNames?domainId=2

Method: GET

Get Single AS Name

This API allows you to get a single AS Name.

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/asNames(asNameID)
```

Method

GET

Sample Request

```
http://127.0.0.1:8981/odata/api/asNames(1)
```

Sample Response

```
{
    "@odata.context": "http://localhost:8981/odata/api/
$metadata#asNames(ASNumber,Description,isBaseDescription)/$entity",
    "isBaseDescription": "true",
    "Description": "LVLT-1 - Level 3 Communications, Inc.",
    "ASNumber": 1
}
```

NOTE

If you do not provide domainId in the request, default domainId 1 is considered and AS Names for that domain are returned.

To get the AS Names for a different domain other than the default domain, add the domainId parameter in the request.

Sample Request: http://127.0.0.1:8981/odata/api/asNames(1)?domainId=2

Edit AS Names

This API allows you to update a single AS name. This API does not support bulk edits.

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/asNames(asNameID)
```

Method

PATCH

Payload

```
{
    "Description": "<description to be updated",
    "DomainId": <domainId>
}
```

Sample Request

```
http://127.0.0.1:8981/odata/api/asNames(1)
```

Sample Payload

```
{
    "Description": "new updated as description",
    "DomainId": 1
}
```

Sample Response

There is no response for this API.

NOTE

- DomainId must be included in the payload to get the AS Names of the specified domain.
- Description is the only field that is editable and must be included in the payload to edit.

Reset AS Name

This API allows you to reset AS Name.

Resource URI

```
http://<nfa odata host>:<nfa odata port>/odata/api/asNames(asNameID)/com.ca.nfa.odata.reset
```

Method

POST

Sample Request

```
http://127.0.0.1:8981/odata/api/asNames(1)/com.ca.nfa.odata.reset
```

Sample Response

```
{
    "@odata.context": "$metadata#com.ca.nfa.odata.asNames",
    "ASNumber": 1,
    "Description": "LVLT-1 - Level 3 Communications, Inc.",
    "isBaseDescription": "true"
}
```

NOTE

This API resets the description of AS Name that is associated to the specified AS for the default domainId 1.

If you want to reset the same AS for a different domain, add the domainId in the request.

Sample Request: http://127.0.0.1:8981/odata/api/asNames(1)/com.ca.nfa.odata.reset?domainId=2

IP Address and Hostname

Network Flow Analysis 10.0.3 supports the IP Address and Hostname API.

NOTE

More Information:

Manage Address-Hostname

IP Address and Hostname Historical Data

The API is used to fetch the data for hostnames associated with IP addresses.

Resource URI

```
http://{nfa odata host>:<nfa odata port>/odata/api/addressesHostNames
```

Method

GET

Sample Request

```
http://127.0.0.1:8981/odata/api/addressesHostNames
```

Payload

Not Applicable

Sample Payload

Not Applicable

Response

```
  {
"@odata.context": "http://127.0.0.1:8981/odata/api/
$metadata#addressesHostNames(Address,Name,AutoResolve,TTL,LastAccessed,ExpireTime,DomainId,HostIp,HostName,Valid_From,V
"value": [
{
"Valid_From": 1566966841,
"Address": "127.0.0.1",
"AutoResolve": "Y",
"TTL": "1n",
"ExpireTime": 1569559006,
"Valid_To": 0,
"LastAccessed": 1566966841,
"Name": null,
"HostName": null,
"HostIp": "127.0.0.1",
"DomainId": 1
},
{
"Valid_From": 1566966841,
"Address": "127.0.0.1",
"AutoResolve": "Y",
"TTL": "1n",
"ExpireTime": 1569559090,
"Valid_To": 1566966855,
"LastAccessed": 1566966841,
"Name": "test1",
"HostName": null,
"HostIp": "127.0.0.1",
"DomainId": 1
},
{
"Valid_From": 1566966855,
"Address": "127.0.0.1",
"AutoResolve": "Y",
"TTL": "1n",
"ExpireTime": 1569559090,
"Valid_To": 1566966942,
"LastAccessed": 1566966841,
"Name": "test1",
"HostName": "dns.broadcom.net",
"HostIp": "127.0.0.1",
"DomainId": 1
},
{
"Valid_From": 1566966942,
"Address": "127.0.0.1",
"AutoResolve": "Y",
"TTL": "1n",
"ExpireTime": 1569559090,
"Valid_To": 1566967090,
"LastAccessed": 1566966841,
"Name": "test1",
"HostName": "test",
```

```
"HostIp": "127.0.0.1",
"DomainId": 1
},
{
"Valid_From": 1566967090,
"Address": "127.0.0.1",
"AutoResolve": "Y",
"TTL": "1n",
"ExpireTime": 1569559090,
"Valid_To": 0,
"LastAccessed": 1566966841,
"Name": "test1",
"HostName": "test1",
"HostIp": "127.0.0.1",
"DomainId": 1
},
{
"Valid_From": 1566982491,
"Address": "127.0.0.1",
"AutoResolve": "Y",
"TTL": "3n",
"ExpireTime": 1574762357,
"Valid_To": 0,
"LastAccessed": 1566982491,
"Name": null,
"HostName": null,
"HostIp": "127.0.0.1",
"DomainId": 1
},
{
"Valid_From": 1566982491,
"Address": "127.0.0.1",
"AutoResolve": "Y",
"TTL": "3n",
"ExpireTime": 1574762357,
"Valid_To": 1566982757,
"LastAccessed": 1566982491,
"Name": "cst242.ftc.avagotech.net",
"HostName": null,
"HostIp": "127.0.0.1",
"DomainId": 1
},
{
"Valid_From": 1566982757,
"Address": "127.0.0.1",
"AutoResolve": "Y",
"TTL": "3n",
"ExpireTime": 1574762357,
"Valid_To": 0,
"LastAccessed": 1566982491,
"Name": "cst242.ftc.avagotech.net",
"HostName": "cst242.ftc.avagotech.net",
"HostIp": "127.0.0.1",
```

```
"DomainId": 1
},
{
"Valid_From": 1566982491,
"Address": "127.0.0.1",
"AutoResolve": "Y",
"TTL": "3n",
"ExpireTime": 1574762357,
"Valid_To": 1566982757,
"LastAccessed": 1566982491,
"Name": "ld01.ftc.avagotech.net",
"HostName": null,
"HostIp": "127.0.0.1",
"DomainId": 1
},
{
"Valid_From": 1566982757,
"Address": "127.0.0.1",
"AutoResolve": "Y",
"TTL": "3n",
"ExpireTime": 1574762357,
"Valid_To": 0,
"LastAccessed": 1566982491,
"Name": "ld01.ftc.avagotech.net",
"HostName": "ld01.ftc.avagotech.net",
"HostIp": "127.0.0.1",
"DomainId": 1
}
],
"@odata.nextLink": "http://127.0.0.1:8981/odata/api/addressesHostNames?$skiptoken=10"
}
```


SNMP router Refresh

Using ODATA API, you can now perform SNMP router refresh.

The new SNMP refresh ODATA API perform the below functionalities.

1. Discovers the SNMP profile that is needed for given router. If not assigned already, assign the SNMP profile to the router.
2. Runs the refresh functionality which is similar to console refresh functionality and reports the status as Success or Fail.
3. Updates router information if any router configuration changes like sysName, sysDesc.
4. Adds the newly added interfaces for a given router.
5. Updates the interfaces information for a given router.

> **NOTE**
> While performing the refresh, you must pass the routerId in the request.

Below are few example scenarios:

If the Router does not exist:

Method

POST

Sample Request:

```
http://<nfa odata host>:<nfa odata port>/odata/api/routers(routerID)/
com.ca.nfa.snmprefresh
```

Sample Response:

```
{
    "error": {
        "code": "404",
        "message": "No Data exists for this key"
    }
 }
```

If any SNMP Router:

Method

POST

Sample Request:

```
http://<nfa odata host>:<nfa odata port>/odata/api/routers(routerID)/
com.ca.nfa.snmprefresh
```

Sample Response:

```
{
    "@odata.context": "$metadata#com.ca.nfa.odata.csnmprefresh",
    "status": "Refresh successful",
    "profileName": "public",
    "refreshRouter": [
        {
            "address": "10.84.200.33",
            "sysName": "Sim34608:nhplt0058.nwie.net",
            "sysDescr": "Linux nhplt0058.nwie.net 3.10.0-1062.1.2.el7.x86_64 #1 SMP Mon
 Sep 16 14:19:51 EDT 2019 x86_64",
            "sysUpTime": 0,
            "ifNumber": 0,
            "updatedOn": 1590127948
        }
    ],
    "refreshInterface": [
        {
            "routerAddress": "10.84.200.33",
            "persistentId": 295,
            "ifIndex": 300000,
            "snmpDataIsValid": false,
            "ifSpeed": 0,
            "inSpeed": 0,
            "outSpeed": 0,
```

```
            "ifType": 0,
            "updatedOn": 1589527628,
            "ifName": "",
            "ifAlias": "",
            "ifDescr": "",
            "ifIpAddr": "",
            "portName": "",
            "vrfName": ""
        },
{
----if any more routers
}

 ]
}
```

If any Non-SNMP Router:

Method

POST

Sample Request:

```
http://<nfa odata host>:<nfa odata port>/odata/api/routers(routerID)/
com.ca.nfa.snmprefresh
```

Sample Response:

```
{
    "@odata.context": "$metadata#com.ca.nfa.odata.csnmprefresh",
    "status": "Refresh Failed",
    "profileName": "Profile not found",
    "refreshRouter": [],
    "refreshInterface": []
}
```

If any Retired Router:

Method

POST

Sample Request:

```
http://<nfa odata host>:<nfa odata port>/odata/api/routers(routerID)/
com.ca.nfa.snmprefresh
```

Sample Response:

```
{
    "error": {
        "code": "404",
        "message": "No Data exists for this key"
    }
```

```
}
```

Troubleshooting

This section contains the following articles:

Installation Troubleshooting

This section provides some troubleshooting tips for problems that are revealed by prerequisite tests. Prerequisite tests can generate warnings or failure notices. If you receive a warning, you can correct the problem immediately or after the installation or upgrade software runs. Failures must be corrected before you can continue. Most of the troubleshooting topics are for prerequisite failures.

> **NOTE**
> Many prerequisite tests rely on general indicators to identify problem areas. Passing a prerequisite test is not a guarantee that everything is configured properly. It is important to meet all of the server requirements, verify that supported versions of the required software are installed and complete all of the configuration tasks.

The following prerequisite tests are run:

Test	Description	Warning or Failure	Server
SC.exe	Verifies that SC.exe is installed on the server.	Failure	Windows servers
Browser	Checks the Registry for a browser. Verifies that a supported browser version is installed.	Warning	Stand-alone Distributed: NFA console
DEP	Verifies that the `winmgt` service is running.	Warning	Stand-alone Distributed: NFA console, Harvester (Windows)
FIPS Algorithm Policy	Verifies that the FIPS Algorithm policy is not enabled.	Verify automatic fix or Failure	Stand-alone Distributed: NFA console
IIS Installed	Verifies that the `wcsvc` service is running.	Warning	Stand-alone Distributed: NFA console
IIS Version	Checks the Registry for IIS version 7.0, 7.5, 8.0, or 8.5.	Warning	Stand-alone Distributed: NFA console
Unsupported Windows Dot Net detected	Checks for .NET 4.6.2 installation.	Failure	Windows servers
NPC Installation Detected	Checks to ensure that NPC is not installed on the server.	Failure	Stand-alone Distributed: NFA console
SNMP	Verifies that the snmp service is running and the process ID is present.	Warning	Stand-alone Distributed: NFA console, all Harvesters
Windows 2012 Detected	Verifies that the server is running Windows Server 2012 for new installs.	Failure	Windows servers

Unsupported Red Hat System Detected	Checks that the Harvester server is running Red Hat Enterprise Linux 6.7 or 6.8 for new installs.	Failure	Distributed: All Linux Harvester servers

NOTE

More information:

- Configure DEP
- Install .NET Framework
- Install IIS, ASP, COM+, and SNMP
- Configure SNMP on Windows servers
- Configure SNMP on Linux servers?

FIPS Algorithm Policy Is Enabled

Valid on Console only

When I click **Next** in the License Agreement screen in the installation or upgrade program for the NFA console, a **Pre-requisite Check Warning** message opens, which includes the following text:

"The FipsAlgorithmPolicy registry key for this system is set to enabled. If the following key is enabled, Windows will not allow certain algorithms to run..."

The error message opens because a system check found the FipsAlgorithmPolicy key in the Windows Registry, which indicates that the Federal Information Processing Standard (FIPS) 140 cryptographic standard is enabled. While this policy is enabled, the server can run only the cryptographic algorithms that have been submitted to and approved by the National Institute of Standards and Technology (NIST).

This restriction can cause problems connecting to databases through Open Database Connectivity (ODBC). Problems with DX NetOps connectivity may result.

To disable the FipsAlgorithmPolicy Registry key, click **OK** in the **Pre-requisite Check Warning** message. The FIPS algorithm policy is disabled and does not restrict database connections.

NPC Installation Detected

Valid on Console

If you attempt to launch the installation or upgrade program on a server that has NetQoS Performance Center installed, an error message opens, and the installation is canceled.

DX NetOps cannot be installed on the same server as CA NetQoS Performance Center. NPC must be completely removed before you proceed with the DX NetOps installation or upgrade.

SC.exe Is Not Installed

Valid on Console or Harvester

When I click **Next** in the **License Agreement** screen of the installation or upgrade program, an error message opens, which begins with the following text:

```
sc.exe is not installed. The installer was unable to find "sc.exe" in the System32 folder.
```

A system check did not find the Service Control command (the `sc.exe` file) in the `Windows/System32` directory. The Service Control command is used for communicating with the Service Controller during command line operations. If the file is missing, the installation or upgrade program exits.

The `sc.exe` file is included with the Windows Server software by default. To correct the problem, restore the missing `sc.exe` from your Windows Server installation software, Windows Resource Kit, or other resource.

Troubleshoot SNMP Issues

This section discusses troubleshooting issues with SNMP:

SNMP Is Not Enabled

Symptom:

When I click **Next** in the **License Agreement** screen of the installation or upgrade program, an SNMP warning message opens. The message reads:

```
Pre-requisite Check Warning
The following issues were found:
SNMP is not enabled.
While not required before installation, some functionality may not work correctly if these are not addressed.
```

Solution:

The SNMP warning message opens because the prerequisite check does not find that the snmpd daemon is running. You can correct the problem when the warning appears or you can proceed with the installation or upgrade. In any case, DX NetOps does not run properly until you configure SNMP and make sure that the snmpd daemon is running.

Use the following procedure to check the SNMP status on a Linux server.

Follow these steps:

1. Enter the status command in a terminal window:

   ```
   service snmpd status
   ```

   The command returns the process ID of the snmpd daemon. If the return text does not list a process ID for the snmpd daemon, it is not running.

Profile not Found

An error message **Profile not found** appears when I click **Discover** router on Enable Interface Page.

Symptom:

SNMP v3 profile does not get discovered for a router.

Solution:

SNMP profile Username must be unique for a NFAConsole. Select a unique user name for the SNMP profile of the router.

Windows Server 2012 R2 or 2016 Required for New Installs

A new installation of or upgrade to DX NetOps 9.5.0 can only be done on Windows Server 2012 R2 or 2016 systems. It cannot be done on Windows Server 2008 R2 systems.

If you attempt to do an installation or upgrade, a dialog box displays in the installer informing you that you are using an unsupported operating system and the installation quits.

```
Unsupported Windows system detected
```

```
It appears that you are not running a Windows 2012 system. Currently, the only supported
 versions include Windows 2012.

Please contact CA support for assistance in upgrading your system and refer to your
 installation guide for more details

regarding system requirements for performing upgrades. This installer will now exit.

OS Detected: [USER'S OS LISTED HERE]
```

This is true for the following installed components:

- NFA console
- Harvester
- Anomaly Detector
- Flow Cloner

Red Hat Enterprise Linux 6.x or 7.x Required for New Installs

A new installation of the DX NetOps 9.3.8 Harvester can only be done on RHEL 6.7, 6.8 or 7.3 systems. It cannot be done on RHEL 5.6 systems.

If you attempt to do a clean install or an upgrade of an existing installation, a dialog box displays in the installer informing you that you are using an unsupported operating system and the installation quits.

```
Unsupported Red Hat system detected

It appears that you are not running a Red Hat Enterprise Linux

Server 6.7+ system. Currently, the only supported versions include

Red Hat Enterprise Linux Server 6.7 and above. Please contact CA

Support for assistance in upgrading your system and refer to your

installation guide for more details regarding system requirements

for performing upgrades. This installer will now exit.

OS Detected: <User's OS version>
```

This is true for the following installed components:

- Linux Harvester

MySQL-Generated Errors During Installation or Upgrade

The MySQL database installation or restore errors include:

- During MySQL database backup:

  - MySQL Database Backup Failed
    An error occurred while attempting to create database backups needed for the MySQL upgrade. See `C:\CA\NFA\MySql_Backups\Backup_Log.log` for more information, or contact CA Support for help in upgrading your product.
    The upgrade is cancelled.
- During MySQL database restore:
  - MySQL Database Restore Failed
    An error occurred while attempting to restore the database backups created for the MySQL upgrade. See `C:\CA\NFA\MySql_Backups\Backup_Log.log` for more information, or contact CA Support for help in upgrading your product.

Troubleshoot Data Collection

As flow starts going to the routers, then to DX NetOps, the flow data is processed and stored on the DX NetOps component servers. If data does not appear in reports as expected, a core troubleshooting approach is to check whether the data files are present on the servers.

This article shows administrators how to check the file systems on DX NetOps servers to verify that data files are being collected.

Prerequisites

This procedure assumes that you:

- Installed the DX NetOps software on the servers that collect and process the data.
- Configured your NetFlow or NetFlow-compliant flow to be exported to DX NetOps.

Once you set up and configure DX NetOps to receive flow, verify that the appropriate data files are collected.

Verify Data Collection in a Stand-Alone Configuration

Ensure that the appropriate data files are collected on a stand-alone system by verifying that valid `.nfa` files are present.

If the `.nfa` files are not present, wait for an hour and check again.

Follow these steps:

Verify that valid NetFlow archive (`.nfa`) files are present:

1. Log in as a user who has administrator privileges for DX NetOps.
2. Locate the following directory:
   `<install_path>\Netflow\datafiles\HarvesterArchive`
3. Verify that the `HarvesterArchive` directory contains valid NetFlow archive files, `.nfa` files that are larger than 0 KB in size.
4. (Recommended) If you do not find valid `.nfa` files, complete the following troubleshooting tasks:
   - Verify that the CA NFA Harvester service is running. If the service is not running, start it.
   - If the Harvester is not receiving data, verify that it listens for data on the same port that was used to configure flow export from the Harvester interfaces. If the ports do not match, modify the `NetflowPacketListenPort` value in the Harvester database `harvester.parameter_descriptions` table.
   - Review the log file:
     `<install_path>\NetFlow\Logs\harvester-wrapper.log`
     If you consult with CA Support for a resolution, supply this log or refer to the errors in it.
     To change the logging level, change `enableDebugLogging=false` to `enableDebugLogging=true` in
     `<install_path>\Netflow\bin\harvester.properties`

Verify Data Collection in a Two-Tier Distributed Deployment

Ensure that the appropriate data files are collected in a two-tier distributed deployment by verifying that Harvester servers have valid `.nfa` files.

If the `.nfa` files are not present, wait for an hour and check again.

Follow these steps:

Verify that valid NetFlow archive (`.nfa`) files are present on the Harvester server:

1. Log in as a user who has administrator privileges for DX NetOps.
2. Locate the following directory on the Harvester server:
   `<install_path>\Netflow\datafiles\HarvesterArchive`
3. Verify that the `HarvesterArchive` directory contains valid NetFlow archive files (`.nfa` files that are larger than 0 KB in size).
4. (Recommended) If you do not find valid `.nfa` files, complete the following troubleshooting tasks:
   - Verify that the CA NFA Harvester service is running. If the service is not running, start it.
   - Review the log file:
     `<install_path>\NetFlow\Logs\harvester-wrapper.log`
     If you consult with CA Support for a resolution, supply this log or refer to the errors in it.
   - To change the logging level, change `enableDebugLogging=false` to `enableDebugLogging=true` in
     `<install_path>\Netflow\bin\harvester.properties`
5. Repeat these steps for any additional Harvester servers.

Troubleshoot Harvester Connection Errors

This section shows CA Network Flow Analysis administrators how to troubleshoot certain Harvester connection error messages.

Custom Report Fails to Connect to Database

Symptom:

Status is "Error".

Status Message is: "Failed to complete baseline calculation: The connection to the database has been lost."

Solution:

1. Verify network connections to all Harvester servers.
2. Ensure that the Harvester service is running on all Harvester servers.
3. Ensure that the NetQoS Reporter/Analyzer Query Services service is running on the NFA Console.
4. Select the custom report, then select **Run**.

Troubleshoot a WSDL Access Error

Symptom:

When I try to add a Harvester, I receive an error that says the WSDL could not be accessed during discovery. The error text is:

"An error occurred during the discovery process. The following information may be helpful. Failed to access the WSDL at: http://localhost:8081/snmp?wsdl. It failed with: Got Server returned HTTP response code: 403 for URL: http://localhost:8081/snmp?wsdl while opening stream from http://localhost:8081/snmp?wsdl"

Solution:

Test polling failed due to a conflict on port 8081, the port that is used by default for the CA NFA DNS/SNMP Proxies service. Port 8081 may be in use by some other program, such as an anti-virus program.

To resolve the problem, change the port setting for the other program or for the CA NFA DNS/SNMP Proxies service. To change the CA NFA DNS/SNMP Proxies service port setting, complete the following steps.

Follow these steps:

1. Log in to the Harvester server with an account that has administrator privileges.
2. Stop the following services on the Harvester server.
   - CA NFA Collection and Poller Webservices (nfa_collpollws on Linux Harvesters)
   - CA NFA DNS/SNMP Proxies (nfa_proxies on Linux Harvesters)
   - CA NFA Poller (nfa_poller on Linux Harvesters)
3. Navigatetomysqlata command line prompt and enter the following commands or input statements:

   ```
   use Harvester;

   select * From settings;
   ```

   The `SnmpProxyAddress` and `SnmpProxyPort` key names and current values are shown.
4. Change the `SnmpProxyPort` key setting, for example, by entering the following command or input statement:

   ```
   Update settings set value = '8089' where name='SnmpProxyPort';
   ```

   In this example, the port for the CA NFA DNS/SNMP Proxies service is changed to 8089.
   Do not change the setting for the `SnmpProxyAddress` key. The only supported value for this key is "localhost."
5. Navigate to *$INSTALL_DIR\NFA\REPORTER\NetQoS.ReporterAnalyzer.ManagerService\bin* folder.
6. Edit the **ReporterManagerService.exe.config** file and update the port number from 8081 to 8089 in the following tag.

   ```
   <add key="DnsProxyClient.JavaDnsProxyUrl" value="http://{0}:8081/dns"/
   >
   to

   <add key="DnsProxyClient.JavaDnsProxyUrl" value="http://{0}:8089/dns"/>
   ```

7. Restart the following services on the Harvester server.
   - CA NFA Collection and Poller Webservices (nfa_collpollws on Linux Harvesters)
   - CA NFA DNS/SNMP Proxies (nfa_proxies on Linux Harvesters)
   - CA NFA Poller (nfa_poller on Linux Harvesters)

   You are now ready to add the Harvester in the NFA console Harvester window.

Troubleshoot a TCP 10060 Error

Symptom:

When I try to add a Harvester, I receive a connection error that is similar to the following one:

"Could not connect to http://<address>:8066/collector. TCP error code 10060: A connection attempt failed because the connected party did not properly respond after a period of time, or established connection failed because connected host has failed to respond <address>:8066."

Solution:

Verify that the following conditions are met:

- You correctly entered the IP address of the Harvester or stand-alone server.
- The DX NetOps services are running on the Harvester or stand-alone server, including the CA NFA Collection and Poller Webservices (nfa_collpollws on Linux Harvesters).
- If the Harvester is installed on a different server than the NFA console, you confirmed that the Harvester server is running. You also confirmed that the Harvester is configured as described in the DX NetOps installation instructions.

Troubleshoot Importing Application Mapping Rules

These are some of the error messages that may be displayed when importing rules for application mapping. When importing is complete, the command returns error messages for any failed imports, then lists the fields for the failed rules.

If the rule import file contains column definition errors, the import operation fails. If the file contains errors in rule definitions, the import operation skips each faulty rule and continues with the next rule.

Notes about the **Description** field:

- All of the rule fields must be populated with values except the optional **Description** (desc) field. The rule line must include the comma for the desc field, however.
- If **Description** field values contain internal commas, the rule import fails. The import utility does not support commas inside **Description** values.

A column named 'XXX' already belongs to this DataTable

The import file contains a duplicate of the column named in the error message. The column name is shown in place of 'XXX'.
Troubleshooting: Delete the duplicated column.

An invalid ApplicationMapping

One of the following problems occurred while you were updating existing rules:

- The import file specifies the `appID` of the original rule incorrectly.
- The specified `appID` corresponds to a rule that has been deleted.

Troubleshooting: Verify that the `appID` is entered correctly and that its rule still exists. You can export the current rules to a `.csv` file. The export file includes the `appID` s for all of the current rules.

Application Mapping object is not valid: An existing record is already in use

Both of the following conditions were met:

- You are using the file format for importing a new rule.
- One of the rule definitions in the import file has the same field values as an existing rule.

Troubleshooting: If this error occurs rarely, you may want to edit the rules individually on the **Application Definitions** page. You can update a group of existing rules by using the `racmd` command.

Application Mapping object is not valid: An invalid protocol was entered

The import file may have a misspelled column name or may include an invalid column. This error may also occur if a rule has an invalid value for a required field (a field other than **desc**).
Troubleshooting: Verify that the following elements are correct in the import file:

- – Field values: Rule definitions have supported field values. The values are entered correctly.
  – Format: The correct columns are included in the file. The column names are spelled correctly.

Application Mapping object is not valid: IP cannot be blank

The error may occur if either of the following conditions is met:

- – An NBAR2 rule is missing the NBAR2 application ID value.
  – A Host (server) rule is missing the server IP address value.

Troubleshooting: Add the missing value to the rule definition.

Application Mapping object is not valid: Mask must be between 0 and 32

The Subnet rule has an unsupported Mask value.
Troubleshooting: Specify a Mask value between 0 and 32.

Application Mapping object is not valid: ToS must be between 0 and 255

The All (ToS) rule has an unsupported ToS value.
Troubleshooting: Specify a ToS value between 0 (all ToS) and 255.

Error: Missing parameter name from -params list

The rule has no specified value for one of the required fields.
Troubleshooting: Verify that all of the required field values are included. All field values are required except for **desc**.
To display help for the command, enter:

```
setapplicationmapping
```

Troubleshoot CA Unified Infrastructure Management - CA Network Flow Analysis Integration

The starting point for any troubleshooting should be the CA Unified Infrastructure Management (CA UIM) server.

- Look in *install_path* `/probes/network/nfa_inventory`
  – `nfa_inventory.txt` is the log file for the probe
  – `discovery.json` is a CTD graph dump of the query results from the discovery server for the SNMP profile sync
  – If debug logging is enabled, you'll also see `graphDump.txt` and `internalGraph.json` files
  – Creating an empty directory here named `debug` will get you more CTD traces
- On the NFA console, look in `ConsoleErrorsLog*.log` and `WebServicesQueryLog*.log` for exceptions.
  – When CA UIM is registered as a portal, the **portalUrl** parameter will be set in the system_settings table.
- SNMP credentials synced from UIM will have a prefix of "UIM-" and a suffix of the ID assigned to it. For example, "UIM-public-1".

Users and Groups

The "Manage Reports" permission no longer functions as a pseudo-super user. This permission now allows users to create, edit, and run reports within the scope that their group membership (permission sets) defines.

Check User Mapping

The concept of an Account doesn't exist in DX NetOps, but there are Group and Permission Set concepts (Permission Sets being a specialization of a group). The mapping is:

- CA UIM Account maps to DX NetOps Permission Set
- CA UIM Origin maps to DX NetOps Group
- CA UIM ACL maps to DX NetOps Roles/Rights
- CA UIM User/Contact maps to DX NetOps User

This means:

- For each CA UIM Account, CA NFA creates a Permission Set.
- Each CA NFA Permission Set has access to Interface Groups that correspond to each CA UIM Origin in the CA UIM Account.
- For each CA UIM ACL, a CA NFA Role is created with rights that correspond to the ACL permissions specified in CA UIM.

  NOTE
  CA UIM adds the CA NFA rights, prefixed with NFA, to facilitate this mapping.
- For a CA UIM User/Contact, a CA NFA User is created with access to the CA NFA Permission Set corresponding to the CA UIM Account, and having the CA NFA Role created corresponding to the ACLs.

Troubleshoot Administration Links Failing in the CA NFA Console

Problem:

The following error occurs when clicking **SNMP Profiles** in the **Administration** page:

```
Microsoft VBScript runtime error '800a01fa'
Class not defined: 'SettingsDB'
/nqsrv_nav.asp, line 111
```

A browser on a remote client machine may show an **HTTP 500 Internal Server Error**.

Cause:

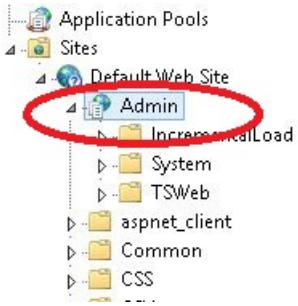
CA NFA was installed without all of the prerequisites. The problem persists even after uninstalling CA NFA, installing the prerequisites, and reinstalling CA NFA.

Solution:

The installer may have created an **Admin** application pool in IIS. The uninstaller is unable to remove the **Admin** application pool (which will have a blue globe icon), so future installs will install into the existing application pool.

Follow these steps:

1. Start IIS Manager.
2. Expand **Default Web Site**.
3. Check whether there is a blue globe icon next to **Admin**. This means that **Admin** is listed as an application.



4. Right-click **Admin** (with the globe icon) and select **Remove**.
5. Run the following from a command prompt:

   ```
   IISRESET
   ```
6. Log in to CA NFA and verify that the links on the **Administration** page now work properly.

Troubleshoot Migrating NBAR2 Application Mappings

If you are migrating a CA NFA installation where NBAR2 application mappings are used, the following log file is created by the installer. It contains details about the import of new NBAR2 application mappings:

```
CA\NFA\REPORTER\racmd\Install_RaCmd_Import_NBAR2.log
```

The following errors may be present in the log file:

- `Application Mapping object is not valid: An existing record is already in use`
  - Indicates that a duplicate entry was not imported. It is normal for these errors to be present.
- `Port XXXXX is already used by an existing application mapping. Using port YYYYY instead.`
  - Indicates that an existing application mapping (of any type) is already using the port listed in `nbar2.csv` , and the mapping was successfully imported using a different port.
- `Port XXXXX is already used by an existing application mapping.`
  - Indicates that a suitable unused port was not found, and the mapping was not imported.

A list of mappings that failed to import (including duplicates) can be found at the end of the log file.

Troubleshoot Missing Data in Reports That Have a Time Filter Applied

Problem

I do not see data in reports that have a Time Filter applied.
The `\Reporter\logs\WebServiceQuery*log` contains errors like:

```
archive@10.0.0.1:3307
Unknown or incorrect time zone: 'UTC'
set time_zone = ?TimeZone;
(0.0000 ms)
at NetQoS.Data.MySqlDataComponentBase.OpenConnection(MySqlConnection connection)
at NetQoS.Data.MySqlDataComponentBase.ExecuteDataTable(MySqlConnection connection, String sql, DataTable
 table, DataParameter[] dataParameters)
at NetQoS.Data.MySqlDataComponentBase.ExecuteDataTable(MySqlConnection connection, String sql, DataParameter[]
 dataParameters)
at NetQoS.Data.MySqlDataComponentBase.ExecuteDataTable(String sql, DataParameter[] dataParameters)
at NetQoS.ReporterAnalyzer.Data.Archive.ArchiveDc.TopTypesOfService(String sourceTable, DateTime utcStartTime,
 DateTime utcEndTime, String timeZone, List`1 whereClauseExpressions, DataDirection dataDirection, Nullable`1
 resolution, UInt32 topTypesOfServiceCount)
```

Cause

This will happen if the installation or upgrade of your Harvesters does not install all values into the `time_zone*` tables on the 3307 instance of MySql.

Resolution

> **NOTE**
> Troubleshoot MySQL Time Zone Tables

Troubleshoot Interface Names

Problem:

An error message similar to the following occurs in the DX NetOps UI:

```
Server Error in '/RA' Application

A potentially dangerous Request.Form value was detected from the client

(ctl00_ContentPlaceHolder1_ManageGroups1_ctl01__dualList_left_hf="...IRCUIT ID
 <XXXXXX350nullCA...").

Description: Request Validation has detected a potentially dangerous client input value,
 and processing of the request has been aborted.

This value may indicate an attempt to compromise the security of your application, such
 as a cross-site scripting attack.
```

Solution:

Rename interfaces so that they do not contain the characters < or >.

> **NOTE**
> **More information:**
>
> Working with Interfaces and Routers

Troubleshoot MySQL Time Zone Tables

After the installation or upgrade of your Harvesters, you might need to install all values into the `time_zone*` tables on the 3307 instance of MySQL.

Follow these steps:

1. Download the `3307_time_zone_tables.sql` file from the ftp link below to your Harvesters:
   ftp://ftp.ca.com/pub/netqos/product_patches/NFA/9.3.3/
   Rename the downloaded file to remove the `.txt` from the end of the filename.
2. Open a command prompt and run the following command from the directory where you saved the `.sql` file:
   `mysql -P3307 mysql < 3307_time_zone_tables.sql`
3. Restart the `Netqos NQMySql` service on the Harvesters.
4. On the NFA Console, recycle all of the `CA MySql` services.

CA NFA and CA Anomaly Detector Connection Issue

When you change the default password for the CA NFA MySQL users using the NFA MySQL User Password Change Utility, you must change the password for the CA NFA MySQL users in the CA Anomaly Detector database.

Follow these steps:

1. Log in to the NetQos MySQL using nsas user.
2. Open the `parameter_descriptions`

table.

3. Set the property value for the following fields with the same password you set for the CA NFA MySQL users:

```
netflow_reporter_db_conn_string
harvester_db_conn_string
```

4. Save and close the MySQL.

CA Anomaly Detector

Your network is one of your most critical resources--and one of your most vulnerable assets. Security threats affect network performance, and the reverse is also true. Whenever the network team makes even a small change, such as adding a new user, security and performance can be compromised.

CA Anomaly Detector gives you visibility into highly variable server and client traffic patterns. The program continually monitors baselines and rapidly analyzes the multiple flow patterns that may indicate misconfiguration, malicious attacks, or poor application delivery.

CA Anomaly Detector provides insight into customary network performance, traffic composition, and traffic volumes. The product alerts you when it detects certain types of data or patterns of anomalous behavior. Unlike other performance analysis products that are limited to one or two common traffic patterns, CA Anomaly Detector alerts you to anomalous behavior by using the powerful infrastructure that is already provided through the following allied products:

- NetFlow data distillation by DX NetOps,
- SNMP collection by CA NetVoyant,
- TCP application performance from CA Application Delivery Analysis, and
- Voice and video performance from CA Unified Communications Monitor.

By leveraging the data that your CA product suite collects, CA Anomaly Detector performs behavior analysis and security monitoring with minimal configuration and with no need for ongoing data entry.

Get Started with CA Anomaly Detector

To get started with CA Anomaly Detector, complete the following tasks:

1. Verify that the installation server meets the following requirements and recommendations:
   - Upgrades: Verify that the existing software is supported for upgrade
   - Operating system
   - (*Recommended*) Hardware specifications.
   - Installation and upgrade prerequisites.
2. Verify that a supported version of Performance Center is installed in your environment: CA Performance Center supported version, or CA NetQoS Performance Center 6.2.
3. Install or upgrade to the latest version of CA Anomaly Detector and DX NetOps.
4. Register CA Anomaly Detector as a data source for Performance Center.
5. (*CA PC deployment*) Enable CA Anomaly Detector configuration functions in the NFA console.
6. Add DX NetOps and any other products that you want CA Anomaly Detector to monitor.
7. Enable the monitored products so their data is accessible to CA Anomaly Detector.
8. Configure the firewalls.
9. (*Optional*) Review the default sensor configurations and make any necessary changes.
10. (*Optional*) Configure targets for alerts, to send Syslog or SNMP trap messages when alerts are triggered.

Features and Benefits

CA Anomaly Detector goes beyond intrusion detection and other more static security tools to take a broader view of the network. The program can monitor your entire network from end to end. Instead of painstakingly applying a fixed set of rules to traffic, CA Anomaly Detector uses a set of dynamic algorithms to create and continually modify a unique profile of the network. The program uses this profile in combination with efficient mathematical analysis to determine whether network traffic is anomalous.

In addition to detecting suspicious or damaged packets, CA Anomaly Detector identifies abnormally high flow and volume sources that can indicate a variety of issues. The program easily scales to create integrated monitoring and reporting across your enterprise. You receive alerts about potential problems, such as:

- Infected hosts
- Victims of infected hosts
- Unauthorized application servers
- Misconfigured servers

Operating in real time, the program identifies fan-out, SYN-only, and ICMP flood traffic that usually indicates a spreading virus, worm, or port-scanning activity. The program also alerts you to:

- Null routing and TTL-expired traffic--helping you identify poorly configured ACLs or routing loops
- Large ICMP or DNS packets that may indicate tunneling activities
- Sources of fragmented packets that double-load network devices and that can ultimately result in retransmission of TCP traffic. These symptoms can signal a frag attack. Knowledge about such sources enables you to make configuration changes that can improve network or application performance.

The program reports only the essential data you need to secure your system and stop intrusions, other security issues, and performance problems. Report views are shown in the Performance Center console, where they contribute to an enterprise-wide perspective on network performance and health.

CA Anomaly Detector provides the following benefits:

- Trending, with per-host breakdown of anomaly sources for timely, precise troubleshooting
- Enterprise-wide correlation of anomalous behavior, broken out per host so you get a full perspective of how key servers behave
- Identification of attacks before symptoms appear so you can prevent of downtime; isolate viruses quickly, and resolve problems
- Accurate and complete data, collected by leveraging existing flow collection infrastructure for easy installation and configuration
- Lightweight reporting of essential data, giving you quick access to crucial information for identifying anomaly causes
- Integration with the following related products for enterprise-wide reporting on network health and application performance from a single console:
  - CA NetQoS Performance Center or CA Performance Center
  - DX NetOps
  - CA Application Delivery Analysis
  - CA NetVoyant
  - CA Unified Communications Monitor

Probability Thresholds

CA Anomaly Detector uses a sophisticated mechanism to help avoid false positives, minimizing the number of alerts that do not correspond to true anomalies. The program uses probability threshold settings that you can customize to control the sensitivity of alert triggering. The thresholds are called probability thresholds because they are keyed to the probability that an actual anomaly has been detected.

In addition to a probability component, the threshold mechanism also relies on the following factors:

- A unique network profile that is based on statistics and the observation of typical operations
- Configurable alert levels, which are a function of the unique profile
- Statistical analysis to determine whether observed network behavior is anomalous or is within the normal range

To determine whether current data is anomalous, the detection process takes all previous data into account to create a statistics-based network profile. Using the profile as a reference, the anomaly detection process estimates and prioritizes any potentially anomalous network activity, based on percentiles and calculates the probability that the observed behavior is anomalous. The entire system is dynamic: It is updated each time it runs to ensure reliability and accuracy.

Correlated Anomalies

Correlated anomalies reduce alarm overload and help cull out false positives so you can focus on the events that are most likely to be issues. CA Anomaly Detector provides an **Enterprise-Wide Correlated Anomalies** view that highlights correlated anomalies. You can navigate from this view to more detailed information while you investigate. A correlated anomaly meets the following minimum requirements:

- Contains three or more anomaly instances
- Has an anomaly index of 2.0 or more
- Originates from a single device

An anomaly index of 2.0 or more indicates the presence of two or more primary anomalies or one primary anomaly and two or more secondary anomalies.

This cross-data source, temporal clustering provides actionable workflows that support a fast, proactive response to issues.

Data Collection in CA Anomaly Detector

Data is monitored in DX NetOps databases by querying flow data directly from the Harvesters. If you add other products for monitoring, data is gathered for analysis from their databases. You can configure CA Anomaly Detector to monitor data from the following products:

- (Required) DX NetOps
- (Optional) CA Application Delivery Analysis
- (Optional) CA NetVoyant
- (Optional) CA Unified Communications Monitor

The anomalies that are detected are shown on the Anomaly Detector page views, which are accessible in the Performance Center Console.

CA Anomaly Detector Scalability

CA Anomaly Detector is highly scalable. The program does not retain a large amount of data. It collects and analyzes only the results of data analysis from the data sources that it is configured to analyze.

The MySQL database does not overload easily. For example, after several weeks of running CA Anomaly Detector at a large enterprise, the database growth rate was equivalent to 4 GB per year. The typical growth rate in testing is 160,000 database rows per day.

Although you can run CA Anomaly Detector with most Harvesters, some limitations apply. In our testing, the following configurations were the maximum configurations that had acceptable performance:

- Linux-based Harvester that runs a maximum of 2 million flows per minute
- Windows-based Harvester that runs a maximum of 4 million flows per minute and does not run the Flow Cloner
- Windows-based Harvester that runs a maximum of 2 million flows per minute and runs the Flow Cloner

We support using a single instance of CA Anomaly Detector to monitor the following data sources:

- DX NetOps: Single instance (*Required*)
- CA NetVoyant: Two instances (*Optional*)
- CA Application Delivery Analysis: Two instances (Optional)
- CA Unified Communications Monitor: Two instances (*Optional*)

Installation and Upgrade

This section describes the installation and configuration procedures that you perform specifically for CA Anomaly Detector.

Co-Installation

You can install CA Anomaly Detector on the same server that hosts DX NetOps software or on a separate server. However, even though installing it on the NFA console is supported, it is highly recommended to have CA Anomaly Detector on a separate server.

WARNING
Installing CA Anomaly Detector on the same server as DX NetOps is only recommended for short trials.

The following table shows the supported configurations for co-installing CA Anomaly Detector with related software.

- With DX NetOps: You can install CA Anomaly Detector on a server that already hosts a stand-alone DX NetOps deployment or the NFA console in a distributed deployment.
- With other related software: You cannot install CA Anomaly Detector on a server that hosts any of the following software:
  - CA Performance Center (CA PC)
  - CA NetQoS Performance Center (CA NPC)
  - CA Application Delivery Analysis (CA ADA)
  - CA Unified Communications Monitor (CA UC Monitor)
  - CA NetVoyant

Components Installed on a Single Server	Supported?
CA Anomaly Detector (AD) with no other related software	Yes
NFA stand-alone (Harvester + NFA console software) + AD	Yes, not recommended
NFA console + AD	Yes, not recommended
CA NPC + the current release of DX NetOps or AD	No
CA PC + AD or any DX NetOps component	No
CA ADA, CA UC Monitor, or CA NetVoyant + AD	No

Prerequisites

Before you install or upgrade CA Anomaly Detector, perform the following tasks:

- If you are co-installing CA Anomaly Detector on a server with DX NetOps, complete the DX NetOps installation or upgrade before you install CA Anomaly Detector.

You can install CA Anomaly Detector on the server that hosts a stand-alone DX NetOps deployment or the NFA console server in a distributed deployment.

- (Recommended) Review the topology of your network. For example, create a diagram of your network and its connections to ensure that you monitor the correct devices for the traffic that interests you.
- (Recommended) Collect information about the Performance Center deployment, including the locations and IP addresses for all related components.
- Verify that you have access to the servers that host Performance Center and the products you want to monitor. Make sure that all of the products are configured and functioning properly, including flow configuration on the routers.
- Prepare the server as described for DX NetOps.
- Stop other programs from running during the installation or upgrade.
- Restart all servers to ensure that all operating system patches are applied.
- Ensure that no one else is logged in to the server during the installation or upgrade.
- Download the installation and upgrade files.
- Check the `nsas` database.
- Stop the services.
- Back up the database and restart the services.

Download the Installation and Upgrade File

Copy the file to the installation server whether you plan to install the software locally or remotely. This ensures that you have access to the file.

1. Get the file for installing or upgrading the components:
   a. Log in to support.ca.com.
   b. Navigate to the **Download Center**: For example, select **Download Center** from the **Support** menu in the left pane.
   c. Select the following navigation options:
      - Select a Product: Select "DX NetOps - MULTI-PLATFORM" to display the links for the NFA console, Harvester (Windows), Harvester (Linux), and CA Anomaly Detector installation and upgrade ISO files.
      - Select a Release: Select **10.0**
      - Select a Gen Level: Select **1**
   d. Download the ISO file from the **Product Components** list that is displayed.

      TIP
      An ISO file is an archive file that contains the contents of an optical disk. Each one of the available ISO files contains the files for installing or upgrading the component named in the file link.
2. Perform one of the following tasks:
   – Burn the ISO file to a CD-ROM or DVD.
   – Extract the contents of the ISO file by using an ISO image software application. Many free ISO image applications are available.
3. Extract the appropriate file to the installation server:
   – `ADSetupx.x.x.exe`

You can install or upgrade the software locally or remotely.

Install CA Anomaly Detector

Complete the steps in this topic to install CA Anomaly Detector on a Windows server or virtual machine.

If the CA Anomaly Detector and any related software components are co-installed, install the software in the order described in Co-Installation. The installation steps are written with the expectation that you have already installed any related products that will be co-located with CA Anomaly Detector.

NOTE
More information:

- Make sure that the server has the necessary components installed before installing CA Anomaly Detector. See CA Anomaly Detector Scalability.
- Download the installation file. See Download the Installation and Upgrade File.

Follow these steps:

1. Log in to the CA Anomaly Detector server as a user who is a member of the Administrators group.
2. Verify that the ISO file is downloaded. The ISO file contains the `ADSetupx.x.x.exe` file, the installation or upgrade program for CA Anomaly Detector.
   Copy the `ADSetupx.x.x.exe` file to the CA installation server, even if you plan to install the software remotely.
3. If the server will host any DX NetOps software, install or upgrade the other software before you proceed.
4. Double-click the `ADSetupx.x.x.exe` file in Windows Explorer.
   The installation program opens.
5. Verify that **English** is selected, then click **OK**.
   The **Welcome** screen opens.
6. Click **Next**.
   The **Pre-Installation Summary** screen opens.
7. Review the information, then click **Install**.
   The **Choose Install Folder** screen opens.
8. (Optional) Specify a custom installation location:
   a. Click **Choose** in the **Choose Install Folder** screen to change the installation location.
      Use the same installation folder for CA Anomaly Detector and the NFA console or stand-alone installation. The default location is `C:\CA\NFA` . We recommend that you install CA Anomaly Detector on a non-system drive.
   b. Click **Next** when the installation path setting is correct.
      The **Installing AD** screen opens. When the installation is complete, the **Install Complete** screen opens and verifies that the installation was successful.
9. (Optional) If errors occurred during the installation, see the following log for details:

   install_path`\AD_Install_<timestamp>`
10. Click **Done** in the **Install Complete** screen.
    The installation program closes.
11. (Optional) Verify that the services are running:
    a. Click **Start**, **Programs**, **Administrative Tools**, **Services**.
    b. Verify that the AD-related services have the status **Started**:
       - CA NFA Host Resolver Service
       - CA NFA Hunter Tracker Service
       - CA MySql

Upgrade CA Anomaly Detector

Complete these steps to upgrade CA Anomaly Detector on a Windows server or virtual machine.

The DX NetOps and CA Anomaly Detector must be on the same release level. Upgrade any previous version of CA Anomaly Detector that is currently installed.

If CA Anomaly Detector is co-installed on a server with DX NetOps (CA NFA), complete the CA NFA installation or upgrade before you install or upgrade CA Anomaly Detector. The upgrade steps are written with the expectation that you have already upgraded any related products that will be co-located with CA Anomaly Detector.

Upgrade Prerequisites

Check the nsas Database

We recommend that you check the `nsas` database before upgrading. You can use the `mysqlcheck` command to verify that the database tables are set up properly. The check can correct some types of problems and can help you avoid an upgrade failure. You can run `mysqlcheck` without stopping the MySQL service.

If the CA Anomaly Detector software is on an upgraded NFA console server or stand-alone server, you may have already checked the `nsas` database as part of the earlier upgrade.

Follow these steps:

1. Log in to the installation server as a user with administrator privileges.
2. Check the nsas database: Enter one of the following `mysqlcheck` commands at a command or shell prompt:
   - To check the tables in all of the applicable databases on the server (`reporter`, `harvester`, `data-retention`, and `nsas` databases):
     ```
     mysqlcheck --all-databases
     ```
   - To check all of the tables in a single database:
     ```
     mysqlcheck --databases db_name
     ```
     where *db_name* = Name of the database to check
   - Example:
     ```
     mysqlcheck --databases nsas
     ```

   You do not need to specify the path to the database. The `mysqlcheck` command finds any or all databases that use the default port (port 3308).
   The command checks each table, attempts to repair any problems, then analyzes and optimizes the table. The return text lists the database tables that were checked and reports the status for each one.
   If the table passed the check, "OK" follows the table name. If a warning is returned and is followed by "OK," the problem was resolved. If unresolved errors occur, contact CA Support.

Stop the Services

To prepare for backing up the database, stop the services on the installation server for CA Anomaly Detector.

Follow these steps:

1. Log in as a user who has administrator privileges for the product.
2. Open the **Services** window: Click **Start, Control Panel, Administrative Tools, Services**.
3. If CA Anomaly Detector is installed on the CA Network Flow Analysis stand-alone server, stop the CA NFA Harvester service and wait 15 minutes.
   Data file processing completes.
4. Stop the remaining services, as listed in the following table.

Service	Stand-Alone NFA with AD	NFA Console with AD	Anomaly Detector (not co-installed)
CA NFA Collection and Poller Webservices	Yes	N/A	N/A
CA NFA Data Retention	Yes	N/A	N/A
CA NFA DNS/SNMP Proxies	Yes	N/A	N/A
CA NFA File Server	Yes	Yes	N/A
CA NFA Harvester	Yes	N/A	N/A

CA NFA Host Resolver Service	Yes (if AD is co-installed)	Yes (if AD is co-installed)	Yes
CA NFA Hunter Tracker Service	Yes (if AD is co-installed)	Yes (if AD is co-installed)	Yes
CA NFA Poller	Yes	N/A	N/A
CA NFA Reaper	Yes	N/A	N/A
CA NFA RibSource	Yes	Yes	Yes
NetQoS MySql	Yes	Yes	Yes
NetQoS NQMySql	Yes	N/A	N/A
NetQoS Reporter Manager Service	Yes	Yes	N/A
NetQoS Reporter/Analyzer General Services	Yes	Yes	N/A
NetQoS Reporter/Analyzer Pump Service	Yes	Yes	N/A
NetQoS Reporter/Analyzer Query Services	Yes	Yes	N/A
NetQoS Reporter/Analyzer Watchdog	Yes	Yes	N/A
NetQoS ReporterAnalyzer Report Service	Yes	Yes	N/A

5. If CA Anomaly Detector is installed on the DX NetOps stand-alone server or NFA console server, check the following directory:

   install_path\Netflow\datafiles\HarvesterWork

   When the `HarvesterWork` directory is empty, you can back up the database.
6. The services are restarted automatically during the upgrade process.

Back Up the Databases and Restart the Services

Before you upgrade, back up the `nsas` database.

Important: Run backups concurrently. If you restore data from backups that have different timestamps, problems can result. Ensure that your backed-up data files are timestamped with the same hour.

Store backups to a remote location to guard against the possibility of a hardware or operating system failure on the main server. For example, back up the databases to an administrative share or mapped network drive.

Follow these steps:

1. Log in as a user who has administrator privileges for the product.
2. Connect to the server:
   Open a Remote Desktop session.
   Initiate a Terminal Services or VNC session to the installation server.
3. Copy each target directory and file to a remote location.
4. Restart the services.

NOTE

More information:

- Compatibility Matrix
- Windows Servers

Upgrade CA Anomaly Detector

Follow these steps:

1. Log in as a user who has administrator privileges for DX NetOps.
2. Verify that the ISO file is downloaded. The ISO file contains `ADSetupx.x.x.exe` , the installation or upgrade program for CA Anomaly Detector.
   Copy the `ADSetupx.x.x.exe` file to the CA Anomaly Detector installation server, even if you plan to install the software remotely.
3. If the server will host any DX NetOps software, install or upgrade the other software before you proceed.
4. Double-click the `ADSetupx.x.x.exe` file in Windows Explorer.
   The installation program opens.
5. Verify that **English** is selected, then click **OK**.
   The **Prior Installation Detected** message opens.
6. Review the information, then click **OK**.
   If the existing software version is not supported for upgrade, upgrade to the supported version.
   If the existing version is supported for upgrade, the **Welcome** screen opens.
7. Click **Next**.
   The **Pre-Installation Summary** screen opens.
8. Review the information, then click **Install**.
   The **Installing AD** screen opens. When the upgrade is complete, the **Install Complete** screen opens.
9. (Optional) If errors occurred during the upgrade, see the following logs for details:
   - General installation log: *install_path*\AD_Install_*<timestamp>*
   - Upgrade log: *install_path*\migrator\migrator.log
10. Click **Done** in the **Install Complete** screen.
    The program closes.
11. (Optional) Verify that the services are running:
    a. Click **Start**, **Programs**, **Administrative Tools**, **Services**.
    b. Verify that the AD-related services have the status **Started**:
       - CA NFA Host Resolver Service
       - CA NFA Hunter Tracker Service
       - NetQoS Mysql
12. Verify that the software is upgraded to the correct version: To display the revision history and verify the current software version, complete the following steps:
    a. Open a Command Prompt window
    b. Start MySQL by entering the following command:
       ```
       mysql nsas
       ```
    c. Display the revision history by entering the following command:
       ```
       select * from revision_history
       ```
13. Complete any post-upgrade configuration tasks.

Configure CA Anomaly Detector

The tasks described here assume that you have completed the steps for pre-installation and installation or upgrade of CA Anomaly Detector.

Be prepared to configure the firewalls to allow communication between CA Anomaly Detector and other components.

Enable TLS for CA Anomaly Detector Data

The Anomaly Detector data can be secured using Transport Layer Security (TLS). The general procedure is:

1. Create a certificate that acts as a Certificate Authority (*CA*) certificate.
2. Use that certificate to generate a new certificate signed by the *CA* for the Anomaly Detector.

Follow these steps:

On the NFA console, use `openssl` to generate a new certificate signed by the Certificate Authority (*CA*) for the Anomaly Detector:

1. Set the `OPENSSL_CONF` environment variable for the `openssl` configuration file.
   ```
   set OPENSSL_CONF=install_path\Tools\openssl\bin\openssl.cfg
   ```
2. Use `openssl` to generate a self-signed certificate, to act as the certificate authority (*CA*) certificate. The `openssl` command can be found on the NFA Console, in the *install_path*`\tools\openssl\bin` directory.
   a. Generate the private key:
      ```
      openssl genrsa -des3 -out nfa-ad-key.pem 2048
      ```
   b. Generate a Certificate Signing Request:
      ```
      openssl req -new -key nfa-ad-key.pem -out nfa-ad.csr
      ```
   c. Remove the passphrase from private key:
      ```
      copy nfa-ad-key.pem nfa-ad-key.pem.orig
      openssl rsa -in nfa-ad-key.pem.orig -out nfa-ad-key.pem
      ```
   d. Generate a certificate signed by the console acting as a *CA*:
      ```
      openssl x509 -req -days 1825 -in nfa-ad.csr -CA nfa-ca-cert.pem -CAkey nfa-ca-key.pem -set_serial
       <unique_num> -out nfa-ad-cert.pem
      ```

On the Anomaly Detector server:

1. Create a directory named *install_path*`\certs` for storing certificates.
2. Copy the *CA* certificate file and the AD certificate files to the `certs` directory created in step 1.
3. Edit *install_path*`\MySql\my.ini` and uncomment these lines:
   ```
   require_secure_transport=true
   tls_version=TLSv1,TLSv1.1,TLSv1.2
   ssl-ca=install_path/certs/nfa-ca-cert.pem
   ssl-cert=install_path/certs/nfa-ad-cert.pem
   ssl-key=install_path/certs/nfa-ad-key.pem
   ```
   Edit all occurrences of these lines. Use appropriate names for the `*.pem` files, and restrict the versions of TLS by removing those you do not wish to allow.
4. Reboot the server to restart all services.

Register CA Anomaly Detector

To display CA Anomaly Detector data in views, you must add CA Anomaly Detector as a data source for Performance Center.

Follow these steps:

1. Display the **Data Source List** page:
   a. Log in to the Performance Center console as a user with administrator privileges, if you are not already logged in.
   b. Select **Admin**, **Data Sources**.

The page for managing data sources opens: **Manage Data Sources** (CA PC) or **Data Source List** (CA NPC).

2. Click **Add** (CA PC) or **New** (CA NPC).
   The dialog for adding a data source opens.
3. Specify values for the following options:
   - **Source Type**: Select **Anomaly Detector** from the list.
     All related products and selected third-party integrations are listed in the **Source Type** list regardless of whether they are installed in your environment. If a data source type is not listed, check to see if you have already registered the maximum number of data sources for that type.
   - **Status**: Select **Enabled**.
   - **Host Name**: Enter the IP address or name of the server that hosts the CA Anomaly Detector software.
   - **Protocol**: Select the protocol to use for contacting CA Anomaly Detector.
     Select https if your network uses SSL for communications. If you select the https option, verify that you have configured the system correctly for https access. For more information, see the Single Sign-on article in the CA Performance Management documentation.
   - **Port**: Enter the port for CA Anomaly Detector.
     The port you specify depends on the protocol you select. The recommended port for the http protocol is 80. The recommended port for the https protocol is 443. If you plan to use SSL for communication with Performance Center, see the Single Sign-on article in the CA Performance Management documentation.
   - **Web Console**: Confirm whether the **Web Console** address is the same as the **Host Name**. If it is different, click the **Same as above** check box to provide the **Web Console** information.
     The default setting is for the **Web Console** address to be the same as the **Host Name**.
   - **Web Console: Same as Data Source**: Confirm whether the same IP address, port, and protocol are used for the **Web Console** as for the host server. If any of the values are different, clear the **Same as above** check box to provide the **Web Console** information.
   - **Display Name**: (Optional) Enter a user-friendly name for the data source. This name is shown on the **Admin** menu under **Product Settings**.
     If you do not specify a name, a default name is generated automatically. The default name consists of the data source type combined with the host name. For example, you can specify `AnomalyDetector_London` as the **Display Name** in place of a default name like `AnomalyDetector@xxx.x.x.xx`.
   - **Enabled**: Select the **Enabled** check box.
4. Click **Test** to verify that the connection to CA Anomaly Detector works.
   If the connection fails, verify that the server name or IP address is accurate for the source type. If the connection fails again, contact CA Support.
5. Click **Save** when the settings are complete and the connection test has succeeded.
   You return to the main page, which shows CA Anomaly Detector in the list of data sources.

Enable Configuration in the NFA Console

If your deployment includes CA Performance Center, you use the NFA console to perform administration tasks for CA Anomaly Detector. Skip this task if your deployment includes CA NetQoS Performance Center.

Follow these steps:

1. Log in to the NFA console as a user with administrator privileges, if you are not already logged in.
2. Click **Administration**. The **Administration** page opens.
3. Click **Alerts, Anomaly Detector** from the menu on the left side of the page. The **Anomaly Detector** page opens with the **View Monitored Products** page displayed.
4. Click **Add/Edit Anomaly Detector**. The **Add/Edit Anomaly Detector** dialog opens.
5. Select values for the following options:
   - **Protocol**: Select the protocol to use for contacting CA Anomaly Detector.

Select https if your network uses SSL for communications. If you select the https option, verify that you have configured the system correctly for https access. For more information, see the Single Sign-On article in the CA Performance Management documentation.

- **IP Address**: Enter the IP address of the server that hosts the CA Anomaly Detector software that you want to administer.

6. Click **Save**.
   You return to the **Anomaly Detector** page. The configuration options are now enabled, so you can add products for CA Anomaly Detector to monitor.

Add Products to Monitor

The next step is to add DX NetOps (CA NFA) as a monitored product for CA Anomaly Detector. You also have the option to add more products that CA Anomaly Detector can monitor for anomalies. This task is described in the following topics:

CA Performance Center

You must add an instance of DX NetOps for CA Anomaly Detector to monitor. You also have the option to add any other products that you want to monitor for anomalies. This topic describes how to complete this task in a CA Performance Center deployment.

Follow these steps:

1. Log in to the CA Performance Center console as a user with administrator privileges, if you are not already logged in.
2. Click **Admin**, **Data Source Settings**: *Name of the CA Anomaly Detector instance that you want to configure*.
   The **Anomaly Detector** page opens in the NFA console.
3. Click **Add** at the bottom of the **Anomaly Detector** page (**View Monitored Products** tab).
   The **Add Monitored Product** dialog opens.
4. Add a CA NFA instance as a monitored product: Specify values for the following options:
   - **Type**
     Select the product name from the list: DX NetOps.
     If DX NetOps is not listed, verify that you have not already registered an instance as a monitored product. You can register only one instance to a single instance of CA Anomaly Detector.
   - **IP Address**
     Enter the IP address of the CA NFA server.
5. Click **Save**.
   The **Anomaly Detector** page shows the NFA console that you added. All of the currently configured Harvesters are added as potential data collectors on the **Collection Sources** tab, but the Harvesters are not enabled at this point.
6. (Optional) Add another monitored product: Repeat the previous two steps until you have all of the monitored products that you need.
   - If you add CA NetVoyant Enterprise, enter the IP address of the individual pollers instead of the NetVoyant Master Console.
   - You can monitor multiple instances of some products, as listed in Software Compatibility.

   The list of monitored products page reflects your changes. CA Anomaly Detector contacts the product to add the collection sources to the **View Collection Sources** tab.

CA NetQoS Performance Center

You must add an instance of DX NetOps for CA Anomaly Detector to monitor. You also have the option to add any other products that you want to monitor for anomalies. This topic describes how to complete this task in a CA NetQoS Performance Center deployment.

Follow these steps:

1. Log in to the CA NetQoS Performance Center console as a user with administrator privileges, if you are not already logged in.
2. Click **Admin**, **Product Settings**: Name of the CA Anomaly Detector instance.
   The **Monitored Products** page opens with the **View Monitored Products** tab displayed.
3. Click **New**.
   The **New Monitored Product** page opens.
4. Add DX NetOps as a monitored product: Specify values for the following options:
   - **Type**
     Select the product name from the list: DX NetOps.
     If DX NetOps is not listed, verify that you have not already registered an instance as a monitored product. You can register only one instance to a single instance of CA Anomaly Detector.
   - **IP Address**
     Enter the IP address of the CA NFA server.
5. Click **Save**.
   The **Monitored Products** page now shows the instance of DX NetOps that you added. All of the currently configured Harvesters are added as potential data collectors on the **Collection Sources** tab, but the Harvesters are not enabled at this point.
6. (*Optional*) Add another monitored product: Repeat the previous two steps until you have all of the monitored products you need.
   - If you add CA NetVoyant Enterprise, enter the IP address of the individual pollers instead of the CA NetVoyant Master Console.
   - You can monitor multiple instances of some products.

   The list of monitored products page is updated to show your changes. CA Anomaly Detector contacts the product to add the collection sources to the **View Collection Sources** tab.

Enable the Monitored Products' Collection Sources

After you add products for CA Anomaly Detector to monitor, enable the products so that monitoring can begin.

This task is described in the following sections:

When you add DX NetOps as a monitored product, the Harvesters that are configured with the DX NetOps instance are added automatically as possible data collection sources. You can enable as many or as few Harvesters as you need. Special considerations apply when you enable Harvesters, as described in *Best Practices: Slow Harvester Rollout*.

The product databases are the collection sources for the CA Application Delivery Analysis, CA NetVoyant, and CA Unified Communications Monitor products.

CA Performance Center

This topic describes how to enable the monitored products for CA Anomaly Detector in a deployment that includes CA Performance Center. As soon as the products are enabled, CA Anomaly Detector can begin monitoring their data for anomalies.

Follow these steps:

1. Log in to the NFA console as a user with administrator privileges, if you are not already logged in.
2. Display the list of collection sources for CA Anomaly Detector:
   a. Select **Administration**.
      The **Administration** page opens.
   b. Select **Alerts, Anomaly Detector** from the menu on the left side of the page.

The **Anomaly Detector** page opens.

c. Click the **View Collection Sources** tab.
The collection sources are displayed in a table.

3. Select the name of each collection source that you want to enable.

NOTE
Special considerations apply when you enable Harvesters, as described in *Best Practices: Slow Harvester Rollout*.

4. Click **Enable**.
CA Anomaly Detector begins to monitor the selected collection sources. The source shows as **Enabled** in the **State** column.

CA NetQoS Performance Center

This section describes how to enable the monitored products for CA Anomaly Detector in a deployment that includes CA NetQoS Performance Center. As soon as the products are enabled, CA Anomaly Detector can begin monitoring their data for anomalies.

Follow these steps:

1. Log in to the CA NetQoS Performance Center console as a user with administrator privileges, if you are not already logged in.
2. Select **Admin**, **Product Sources**: Name of the Anomaly Detector instance.
The **Monitored Products** page opens with the **View Monitored Products** page displayed.
3. Click the **View Collection Sources** tab.
The collection sources are displayed in a table.
4. Select a collection source that you want to enable.

NOTE
Special considerations apply when you enable Harvesters, as described in *Best Practices: Slow Harvester Rollout*.

5. Click **Edit**.
The **Edit Collection Source** dialog opens.
6. Select the **State** check box to enable the collection source.
7. Click **Save**.
You return to the list of collection sources. The **State** value is updated to reflect your change. CA Anomaly Detector begins to monitor the collection source.

Best Practices: Slow Harvester Rollout

When you enable a Harvester for monitoring, performance demands for the Harvester increase. CA Anomaly Detector queries the Harvesters for flow data every 15 minutes. The Harvesters in turn search the flow archive, a cache of raw flow data, for the requested data. Searches can include millions of flows.

We strongly recommend that you enable Harvesters gradually. For example, enable two or three Harvesters, wait 15 minutes, then check the query time for the Harvesters on the **View Collection Sources** page. Query times should be less than 15 minutes.

To check Harvester errors in the NFA console, open the NFA console, click **Administration**, and click the **Harvester** icon.

The program may take as long as 15 minutes to query newly added Harvesters. The current reporting interval must be completed before the program can query the newly added Harvesters.

You can add a maximum of 10 Harvesters to a single instance of CA Anomaly Detector.

Configure Firewalls

Modify your firewall as needed to enable CA Anomaly Detector and the Harvesters to communicate. In addition, review your Access Control Lists (ACLs) and make any changes that are needed.

The following ports are used by default for communication:

- CA Anomaly Detector server to the Harvester ports:
  - TCP 3307 (MySQL server)
  - UDP 161 (Watchdog services)
- CA Anomaly Detector server to other servers that host data sources (NFA console, CA Application Delivery Analysis, CA NetVoyant, and CA Unified Communications Monitor):
  - TCP 3308 (MySQL server)
- CA Anomaly Detector server ports (outbound):
  - UDP 514 (Syslog alerting)
  - UDP 162 (SNMP alerting)
- Performance Center server to the CA Anomaly Detector server:
  - TCP/HTTP 80, if http is used for communication
  - TCP/HTTPS 443, if https is used for communication
- CA Performance Center to the CA Anomaly Detector server:
  - TCP 8681, which CA PC uses to request data for the views

Sensor Configuration

Most sensors are enabled by default, even sensors that do not have a collection source. The inactive sensors have no effect on data analysis until the appropriate data source is added. If you disable inactive sensors, remember to manually enable them again when you add the corresponding collection source.

Probability Thresholds

The probability threshold is a sensitivity setting that increases or decreases the likelihood that the sensor will send out alerts when it detects a monitored behavior.

The default threshold for all sensors is 90% probability. As you raise the probability threshold, fewer alerts are likely to be sent and it is more likely that real issues triggered the alerts.

Probability Threshold	Number of Anomalies Reported
Lower	More
Higher	Less

If you monitor a device that has a known issue, you may want to raise the probability threshold to prevent false alerts. A lower probability setting may be better for a critical item that is fairly stable, so that you are notified about issues early.

White List

Use the White List to exclude servers from monitoring and from views when you edit a sensor configuration. The White List typically consists of hosts that are known to be safe.

Do not add a host to the White List if the host is important and the host may be vulnerable to attack. For example, do not add DNS servers to the White List even though these servers may generate spurious anomalies. As CA Anomaly Detector continues to gather data about typical behavior, it should create fewer spurious anomalies for hosts like DNS servers.

Absolute Thresholds

You have the option to set an absolute threshold so that alerts are generated when the threshold is violated, provided that alerts are enabled. Absolute thresholds are applied before probability thresholds.

The following table lists the sensors and units of measure for absolute thresholds.

Sensor	Looks For:	Unit of Measure
Buffer Misses	Pattern in buffer misses	buffer misses
Congestion Sources	Sources of overloading based on ICMP Source Quench	flows
Destination Unreachable Sources	High-volume sources of network, host, or port unreachable messages	host or port unreachable messages
Dropped Packets	Pattern in queue drops	dropped packets
Fragmented Packet Sources	Large sources of fragmented packets	fragmented packets
Frags And Loss Sources	Sources of fragmentation and packet loss	flows
High and Variable Vol-In	Highest-volume and variability destinations	bytes in
High and Variable Vol-Out	Highest-volume and variability sources	bytes out
High Flow Sources	Top sources of data flows	flows
High Packet Fan Out	Sources of the largest fan-out traffic patterns	destination hosts
Incoming Discard Rate	Pattern in incoming discards	discard rate
Incoming Error Rate	Pattern in incoming errors	error rate
Large DNS Packet Sources	Large sources of DNS packets that are larger than normal	flows
Large ICMP Packet Sources	Large sources of ICMP packets that are larger than normal	flows
Non-local Sources	Top sources of non-local traffic	non-local traffic
Packet Load	Pattern in bytes per packet to server	bytes per packet
Previously Null Routed Srcs	Hosts with high volumes of traffic that is no longer null-routed	flows
Refused Sessions	Pattern in refused sessions	percent
Retransmission Time	Pattern in retransmissions	seconds
RST-Only Sources	Highest-volume sources of RST-only flows	flows
SYN/RST-Only Packet Srcs	Highest-volume sources of SYN/RST-only flows	flows
SYN-Only Packet Sources	Highest-volume sources of SYN-only flows	flows
TTL Expired Sources	High-volume sources of TTL expired packets	flows
Voice Call DoS	DoS in voice calls	calls per minute
Voice Call Fan Out	Pattern in fan-out of voice calls	call ratio
Voice Server Distress	Call Server Distress	Volume Weighted Error Ratio

Configure Sensor Thresholds and Options

You can configure sensors to have different probability thresholds and to include the following optional features: absolute thresholds, alerts, and host exclusions.

Follow these steps:

1. Display the **Data Source List** page:
   a. Log in to the Performance Center Console as a user with administrator privileges, if you are not already logged in.
   b. Select **Admin**, **Data Sources**.
      The page for managing data sources opens: **Manage Data Sources** (CA PC) or **Data Source List** (NPC).
2. Click the name of the CA Anomaly Detector instance that you want to configure.
   The **Monitored Products** page opens.
3. Click **View Sensors**.
   The **Sensors** page opens and displays the first group of sensors. Click the page numbers at the bottom to page through the sensor list.
4. Double-click the row for the sensor that you want to edit. The **Edit Sensor** page opens.
5. Edit any of the following options that you want to change:
   – **Probability Threshold**: (*Optional*) Change the numeric value to change the probability threshold.
   The probability threshold determines the likelihood that the sensor will send out alerts when it detects the behavior it monitors.
   – **Absolute Threshold**: (*Optional*) Enter a numeric value to create an absolute threshold.
   Enter the absolute threshold value based on the unit of measure for the sensor. An absolute threshold signals an anomaly if the threshold is exceeded. The anomaly triggers an alert if you select the **Alert** option.
   Absolute thresholds are applied before probability thresholds. The program continues to calculate the percentiles for the host/sensor combination even if you set an absolute threshold. These calculations are used if the absolute threshold is removed later or if the probability threshold is exceeded first. An anomaly is created if either threshold is exceeded.
   – **Alert**: (*Optional*) Select the **Alert** check box to send alerts when the probability or absolute threshold is exceeded. This option is enabled for most sensors by default. The exceptions are sensors that are most useful when you monitor correlated anomalies (that is, when alerts are triggered for multiple types of sensors).
   – **State**: Select the **State** box to enable the sensor or clear the check box to disable the sensor.
   – **IP address**: (*Optional*) Enter a host IP address and click **Add** to add the host to the White List, so the host is excluded from anomaly reporting:
   The IP address appears in the **White List** field.
   – **White List**: (*Optional*) Highlight a host in the **White List** field and click **Remove** to delete the host from the White List.
6. Close the **Edit Sensor** page: Click **Save** or **Cancel**.
   You return to the **Sensors** page, which reflects any changes that you saved.

Configure Alert Targets

Alerts are enabled for most sensors by default. The anomalies that result are displayed in the Anomaly Detector page views. You can also receive anomaly information by configuring the alerts to trigger Syslog messages, SNMP trap messages, or both.

Syslog Alerts

Alerts are enabled for most sensors by default, but messages for the alerts are not sent to a Syslog server unless you configure the target for Syslog messages.

About Syslog Alerting

The alerting feature lets you specify parameters for alerts to be sent to a CEF-compliant Syslog server. Syslog is a standard protocol for handling log messages in a heterogeneous environment. A Syslog server that is running a syslog daemon collects log messages, and sometimes filters and processes the messages. The log messages pass to the Syslog server from devices on the network such as routers and switches.

Each syslog message corresponds to a single alert for an anomaly cluster or for a basic anomaly. An anomaly cluster alert contains details about the basic anomalies in the cluster. The details can give the appearance of multiple entries in the message.

The alert format complies with the Common Event Format (CEF) standard. The message type formats are described in the following topics:

Configure the Target for Syslog Messages

Configure Syslog alerting to identify a target Syslog server that will receive messages when sensors report a threshold violation. If Syslog alerting is not configured, the alerts that sensors generate may appear in the **Anomaly Detector** page views, but no messages are sent to report the alerts.

Prerequisites for Syslog Alerting:

- Set up a Syslog server.
- Configure a service on the Syslog server to read syslog messages from CA Anomaly Detector.

Follow these steps:

1. Display the **Data Source List** page:
   a. Log in to the Performance Center Console as a user with administrator privileges, if you are not already logged in.
   b. Select **Admin**, **Data Sources**.
      The page for managing data sources opens: **Manage Data Sources** (CA PC) or **Data Source List** (CA NPC).
2. Click the name of the CA Anomaly Detector instance that you want to configure.
   The **Monitored Products** page opens.
3. Click **View Alert Targets**.
   The **Alert Targets** page opens.
4. Double-click the **syslogging** row.
   The **Edit Alert Target** page opens.
5. Specify the **Target**: Enter the IP address or DNS hostname of the system that will receive the Syslog information.
6. Select one or both of the following options to enable the alert:
   – **Basic State**: Send alerts whenever any single sensor detects an anomaly that crosses a threshold.
   – **Cluster State**: Send alerts only when the requirements for a correlated anomaly are met.
   This is the recommended setting for first-time use of CA Anomaly Detector. If you have already disabled alerts for the sensors that are irrelevant to you, the recommended setting is **Basic State**.
7. Click **Save**.
   You return to the **Alert Targets** page, which reflects any changes you saved.

Best Practices:

Alerts are enabled for most sensors by default so that when you start using CA Anomaly Detector, you can review a wide range of anomalous behaviors with a minimum of configuration. If you use Syslog alerting and you select the **Basic State** option at this stage, you may see so many anomalies that you cannot determine which ones are significant.

If you begin by selecting the **Cluster State** option for alert targets, the anomalies you see are much more likely to be significant. You can quickly determine which sensors are useful to you. At this point you can disable alerts for the other sensors, then start using the **Basic State** option. This produces an expanded set of results for the anomaly types that interest you. The anomalies from the other sensors are eliminated.

To explore all of the potential anomaly cluster types, you may want to enable any disabled sensors. In this case, use the **Cluster State** option.

SNMP Trap Alerts

You can enable SNMP traps to be sent to a third-party network monitoring program. If you enable SNMP trap alerts, an SNMP trap is sent each time a threshold violation triggers an alert. You also can set the alert to be triggered only by a correlated anomaly.

Multiple targets for SNMP traps are not supported.

Follow these steps:

1. Configure the trap receiver to listen for traps from the CA Anomaly Detector server.
   Use the MIB file to import the CA Anomaly Detector Object IDs (OIDs) into your trap receiver. The MIB file is located in the following directory on the CA Anomaly Detector server:
   install_path\NQAD\MIB\NetQoS-AnomalyDetector-mib
   The steps for importing the OIDs and configuring the trap receiver are specific to the trap receiver.
2. Display the **Data Source List** page: Select **Admin**, **Data Sources**.
   The page for managing data sources opens: **Manage Data Sources** (CA PC) or **Data Source List** (CA NPC).
3. Click the name of the Ca Anomaly Detector instance that you want to configure. The **Monitored Products** page opens.
4. Click **View Alert Targets**. The **Alert Targets** page opens.
5. Double-click the **snmp_traps** row. The **Edit Alert Target** page opens.
6. Specify the **Target**: Enter the IP address or DNS hostname of the system that will receive the SNMP traps.
7. Select one or both of the following options to enable the alert:
   – **Basic State**: Send alerts whenever any single sensor detects an anomaly that crosses a threshold.
   – **Cluster State**: Send alerts only when the requirements for a correlated anomaly are met.
   This is the recommended setting for first-time use of CA Anomaly Detector. If you have already disabled alerts for the sensors that are irrelevant to you, the recommended setting is **Basic State**.
8. Click **Save**.
   You return to the **Alert Targets** page, which reflects any changes you saved.

Example of a Basic Anomaly Message

A Syslog Basic Anomaly message reports an alert for a single anomaly instance. The following example shows the message fields for a Basic Anomaly message:

```
02-12-2009 17:51:42 Local0.Alert 10.0.23.138 Feb 12 17:51:42 sk23-138 CEF:0|
NetQoS|AnomalyDetector|2.0.12.1|23|Frags and Loss Sources|5|
src=XXX.XX.X.XXX start=2/12/2009 5:36:00 PM msg=metric 2 anomaly probability 1%. A router close
```

The sixth CEF field, the `msg=metric` field, identifies the sensor type that detected the violation. The sensor type in the example is Frags and Loss Sources. The main body of the CA Anomaly Detector information follows `msg=` .

```
msg=metric METRICVALUE anomaly probability PROBVALUE%. OptionalROUTERINFOVALUE
```

where:

- **METRICVALUE**
  Actual value, expressed in the units of measure for the sensor. This value is an integer with a value over 0 in the example. The data type is double, and double.maxvalue is positive 1.79769313486232e308. The significance of the

integer depends on the sensor type. For example, the integer for a FanOut sensor represents the number of hosts with which the source IP communicated.

- **PROBVALUE%**
  The percentage of anomaly probability, expressed as a value between 1 and 100. In the example, the value is 1. This value is the statistical probability that a sensor has detected anomalous traffic. You can use thresholds to suppress Syslog messages when the probability is low. You can specify the threshold for each sensor independently.
- **ROUTERINFOVALUE**
  (Optional) The optional ROUTERINFOVALUE field is provided for anomalies that are based on NetFlow. The close router and interface information is derived from the router that sent the flow data.

  ```
  A router close to the issue (for further analysis or ACL) is ROUTERIPADDRESSVALUE and IN inte
  ```

  where:
  - **ROUTERIPADDRESSVALUE**
    IP address of the router, as reported by NetFlow.
  - **INIFINDEX**
    Interface (IF) index on the incoming interface of the router, as reported by NetFlow.

    NOTE
    If you want to write a parser to handle Syslog Basic Anomaly or Anomaly Cluster messages, specify a value for the
    `msg=`
    field. The other fields are in the CEF standard format, and you do not need to specify their values.

Example of an Anomaly Cluster Message

A Syslog Anomaly Cluster message reports an alert for a group of anomalies. The following example shows the message fields for an Anomaly Cluster message:

```
02-12-2009 17:51:42 Local0.Alert 10.0.23.138 Feb 12 17:51:42 sk23-138 CEF:0|
NetQoS|AnomalyDetector|9.2.0.1|2161393963:1234482300|AnomalyCluster|10|
src=xxx.212.65.43 start=2/12/2009 5:45:00 PM msg=AnomalyCluster: anomalies included Frags, Frag
interfaces close to the issue (for further analysis or ACL) are 10.00.00.100 : 12, 10.00.30.100
```

The sixth CEF field identifies the sensor types that detect the violation. The message contains multiple sensor types, so the sixth CEF field always has the value AnomalyCluster. The list of sensor types is contained in the message body. The `msg` field format is as follows:

```
msg=AnomalyCluster: anomalies included LISTOFANOMALIESVALUE anomalyScore CLUSTERSCOREVALUE max
%. OptionalROUTERINFOVALUE
```

where:

- **LISTOFANOMALIESVALUE**
  List of the anomalies in the cluster, which is comma-delimited in the following manner: fanOut, SYNOnly, and topNullRoutes.
- **CLUSTERSCOREVALUE**

Weighted severity value. The cluster score is the weighted count of anomalies. Secondary role anomalies, such as flows, volume in, and volume out, count as 0.5. All other anomalies count as 1.

- **PROBVALUE**
  Maximum anomaly probability of the anomalies in the cluster. The PROBVALUE field is similar to the field in the basic anomaly message, except that it identifies the maximum probability across all the anomalies in the cluster.
- **ROUTERINFOVALUE**
  (Optional) The optional ROUTERINFOVALUE field is provided for anomalies that are based on NetFlow. The close router and interface information is derived from the router that sent the flow data. The format for ROUTERINFOVALUE is as follows:

  ```
  Routers/
  interfaces close to the issue (for further analysis or ACL) are ROUTERandINTERFACElistVALUE
  ```

  where:
  - **ROUTERandINTERFACElistVALUE**
    List of the router IP addresses and the associated incoming IF index. For example, enter 199.30.15.30 : 1, 199.30.15.30 : 1, 199.30.15.30 : 1. The router : interface pairs follow the same order as the anomaly types that are reported in the LISTOFANOMALIESVALUE field.

Uninstall CA Anomaly Detector

CA Anomaly Detector 9.3.0 and later includes an option to uninstall the product, which you can use after an installation or upgrade.

If CA Anomaly Detector is installed on the same server as DX NetOps, uninstalling CA Anomaly Detector disables DX NetOps.

You should be able to install and uninstall the software once or twice without any problem. If you have ongoing problems, contact CA Support rather than continuing to install and uninstall the software.

As an alternative to using the Uninstaller, you can uninstall the software from the **Control Panel**. The program is listed as AD in the Windows **Add or Remove Programs** or **Programs and Features** window.

Follow these steps:

1. Log in as a user who has administrator privileges for DX NetOps.
2. Perform the following preliminary steps:
   - Open the **Services** window and verify that the NetQoS MySql service is running. If the service is not running, start it.
   - If the installation server is also used to host Performance Center, back up your data and configuration files.
   - Exit from all applications, with no exceptions.
3. If you plan to uninstall both CA Anomaly Detector and DX NetOps (CA NFA) from the same server, uninstall the CA NFA software first.
4. Start the Uninstaller for CA Anomaly Detector: In *install_path*\Uninstall\NQAD , double-click AD_Uninstaller.exe
   The **Uninstall** window opens.
5. Click **Next** if you are ready to proceed.
   When you click **Next**, the **Cancel** button is disabled. The Uninstaller has no Undo option - once you begin uninstalling the software, you cannot restore the deleted files automatically.
   The uninstall program removes the program and data files, including the following CA NFA and MySQL elements:

- Registry entries
- Shortcuts, links, and aliases
- Most files
- Some directories

As the uninstaller runs, the screen displays progress messages. When the process completes, the screen displays a list of the directories and files that were not deleted.
Note: Leave the file system undisturbed while uninstallation is in progress. Do not attempt to view the progress in Windows Explorer, for example.
Once the program finishes, the **Uninstall Complete** screen opens.

6. Click **Done** to close the program.
   Wait a few minutes to allow the helper process to finish the final cleanup.
   Some files are not deleted until this phase is finished. Once the final cleanup is finished, the Uninstaller itself is deleted.

Notes:

- The uninstallation log is at: *install_path*\AD_Uninstall_<timestamp>.txt .
- You can manually delete any remaining CA Anomaly Detector directories, files, and services. If other related software is installed on the server, do not delete the MySQL service.
- If you make an unsuccessful attempt to reinstall the software, contact CA Support.

CA Anomaly Detector Views in Performance Center

CA Anomaly Detector is a CA Infrastructure Management product that gives you visibility into the highly variable traffic patterns of servers and clients. The program continually monitors behaviors and rapidly analyzes multiple network data patterns to search for signs of misconfiguration, malicious attacks, or poor application delivery. CA Anomaly Detector can send alerts for as many as 27 types of potential anomalies from a number of data sources, depending on which data sources are the product is configured to monitor:

- Flow distillation by DX NetOps
- SNMP collection by CA NetVoyant
- TCP application performance from CA Application Delivery Analysis
- Voice and video performance from CA Unified Communications Monitor

To see the predefined CA Anomaly Detector reports, go to one of the following locations:

- (CA PC) Open the CA Performance Center Console and select **Dashboards**, **Operations Displays: Anomaly Detector**.
- (NPC) Open the CA NetQoS Performance Center Console and select **Reports**, **Operations: Anomaly Detector**.

> **NOTE**
> To view the reports, your user account must be assigned a role that gives you access to the report menu. User account settings are managed in the Performance Center Console, as described in the *Administrator* information for your Performance Center version.

You can customize the Anomaly Detector page to meet your specific needs.

Built-in CA Anomaly Detector Views on the Anomaly Detector Page:

- Anomaly Activity
- Anomaly Detector Overall Status
- Enterprise-Wide Correlated Anomalies
- Top Anomalies by Host
- Top Anomalies by Interface
- Top Enterprise-Wide Network Anomalies

Additional View that You Can Add to a Custom Page:

- Enterprise-Wide Anomalies

Display Predefined Views

The predefined CA Anomaly Detector views are displayed in the Performance Center Console. To see the views, open the **Anomaly Detector** page:

- (CA PC) Select **Dashboards**, **Operations Displays**: **Anomaly Detector** in the CA Performance Center Console.
- (NPC) Select **Reports**, **Operations**: **Anomaly Detector** in the CA NetQoS Performance Center Console.

To see the views, your user account must have a role that gives you access to the view menu. User account settings are managed in the Performance Center Console.

Customize Predefined Views

Change Reporting Time Frame

To change the reporting time frame for the views on the Anomaly Detector page, use the controls at the top of the page in one of the following ways:

- Choose a reporting time frame from the drop-down list.
- Click the scroll button on the left or right side of the date to move the reporting period forward or backward by a day.
- Click the start date or end date and select a new date from the calendar that opens.
- Click the starting hour, starting minute, ending hour, or ending minute and select a new value from the drop-down list.

For more information, see Set a Custom Time Frame.

Other View Customizations

You can also customize the view page to change which views are included, the view order, the page layout, the page title, and the label of the menu option that opens the page. To access these editing options, select **More** in the upper right corner, then select **Edit Dashboard** (CA PC) or **Edit Report** (NPC).

For more information about how to customize view pages, see the *Administrator* information for your version of Performance Center.

For more information, see:

- Change the View Settings
- Zoom In to Narrow the Time Frame

Links and Detail Pages

Links are included in some views to help kickstart anomaly troubleshooting. Links take you to pre-configured reports or to a general page of additional detail. The following views include links:

- **Enterprise-Wide Correlated Anomalies**
- **Enterprise-wide Anomalies**
- **Anomaly Drill-In**

Click one of the links for more information.

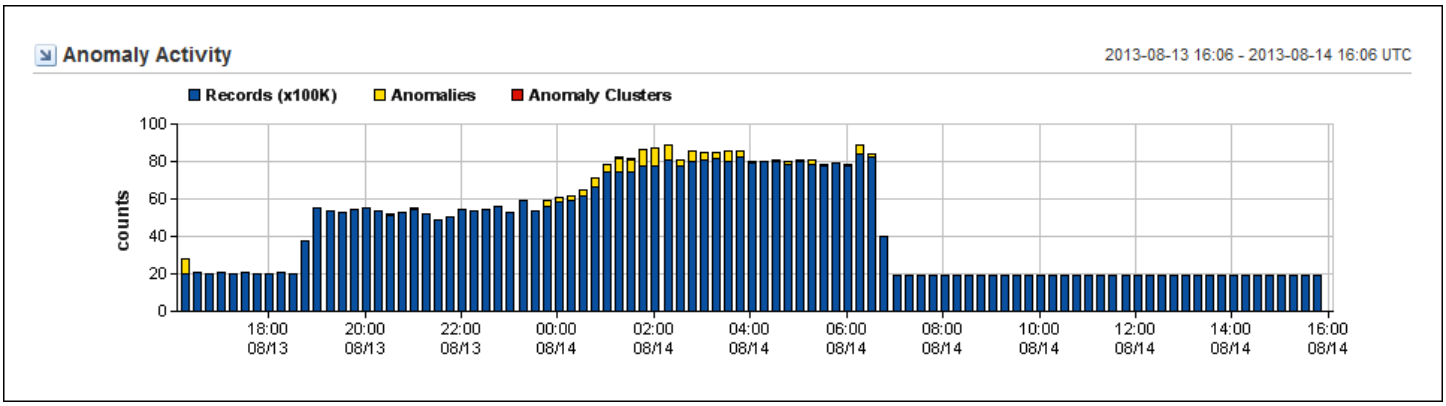
- **Date** Link

Click the *Date* link to go to the **Anomaly Trend** view. This view shows the value and probability of the anomaly over time.

- **Discovered by** Link
  Click a *Discovered By* link to view details. The type of anomaly determines the link destination:
  – DX NetOps Anomalies: **Router** or **Interface** page in the Performance Center Console
  – Anomalies from other data sources: Main page for the originating product
- **Host** Link
  Click a *Host* link to go to more granular information about the device that has the anomaly. Clicking a **Host** link can be the first step in troubleshooting the anomaly.
  Host link destinations are based on the sensor type. For many DX NetOps sensors, the **Host** link opens the page for defining a Flow Forensics report in the NFA console, which has pre-populated report filters.

Anomaly Activity

The **Anomaly Activity** view displays anomalous activity as a bar chart. This view gives you a visual overview of how many anomalies and anomaly clusters occurred in comparison to all of the records that were processed.



The example graphic shows the number of records in the hundreds of thousands (Y-Axis) over a 24-hour period. Activity is shown along the X-Axis each time the program runs (usually at 15-minute intervals).

You can edit the following view settings:

- Time frame for all views on the page.
  By changing the time frame for the page, you can discover when the issue began and look for patterns.
- View title and context.

This view is included by default on the **Anomaly Detector** page in the Performance Center Console. The example graphic shows the view as it appears in the CA NetQoS Performance Center Console.

Anomaly Detector Overall Status

The **Anomaly Detector Overall Status** table shows the number of records that were processed during the selected time frame, the number of anomalies, and the number of anomaly clusters.

You can edit the following view settings:

- Time frame for all views on the page
- View title and context

This view is included by default on the **Anomaly Detector** page in the Performance Center Console.

Top Enterprise-Wide Network Anomalies

The **Top Enterprise-Wide Network Anomalies** pie chart shows the top anomaly types for the reporting time frame.

This view shows the type of network traffic that had the highest proportion of anomalous traffic. This data may give you the first insight into poor network performance.

The legend identifies the number of instances and the colors for each anomaly type. Anomaly types are named for the corresponding sensors.

The **Top Enterprise-Wide Network Anomalies** view is most useful for tracking sudden changes in network behavior. For example, suppose that the Enterprise-Wide Network Anomalies view shows that the Large DNS Packet Sources category accounts for 25% of all potentially anomalous behavior for the past week. If the summary indicates that Large ICMP Packets account for 50% of such traffic today, you would follow up with an investigation.

You can edit the following view settings:

- Time frame for all views on the page
- View title and context

This view is included by default on the **Anomaly Detector** page in the Performance Center Console.

Top Anomalies by Host

The **Top Anomalies by Host** pie chart shows the top anomalous hosts, ranked by the number of anomalies for the reporting time frame. A maximum of 10 hosts are included. The number of instances is shown next to each pie slice.

You can edit the following view settings:

- Time frame for all views on the page
- View title and context

This view is included by default on the **Anomaly Detector** page in the Performance Center Console.

Top Anomalies by Interface

The **Top Anomalies by Interface** pie chart shows the anomalies for the top interfaces, ranked by number of anomalies. A maximum of 10 anomalies are included. The number of instances is shown next to each pie slice.

You can edit the following view settings:

- Time frame for all views on the page
- View title and context

This view is included by default on the **Anomaly Detector** page in the Performance Center Console.

Enterprise-Wide Correlated Anomalies

The **Enterprise-Wide Correlated Anomalies** table summarizes the anomalous behaviors that are most likely to be damaging to the network. This view identifies network locations that you can begin investigating if you suspect malicious activity.

Anomaly clusters are better indicators for problems than single anomalies. Many types of attacks involve multiple instances of anomalous network behavior. Instances are often clustered around a group of a few hosts at first, then the behavior spreads in a fan-out behavior. In a fan-out behavior, seemingly unrelated devices are affected and unexpected traffic is produced from multiple sources.

Correlation is performed by using an algorithm that considers the typical patterns for each type of monitored network traffic.

An anomaly is *correlated* when the following requirements are met:

- Three or more anomaly instances exist.
- Two different anomaly types are present or have an Anomaly Index above 2.0.
- One device is the source of the anomalies.

You can edit the following view settings:

- Time frame for all views on the page
- View title and context

This view is included by default on the **Anomaly Detector** page in the Performance Center Console.

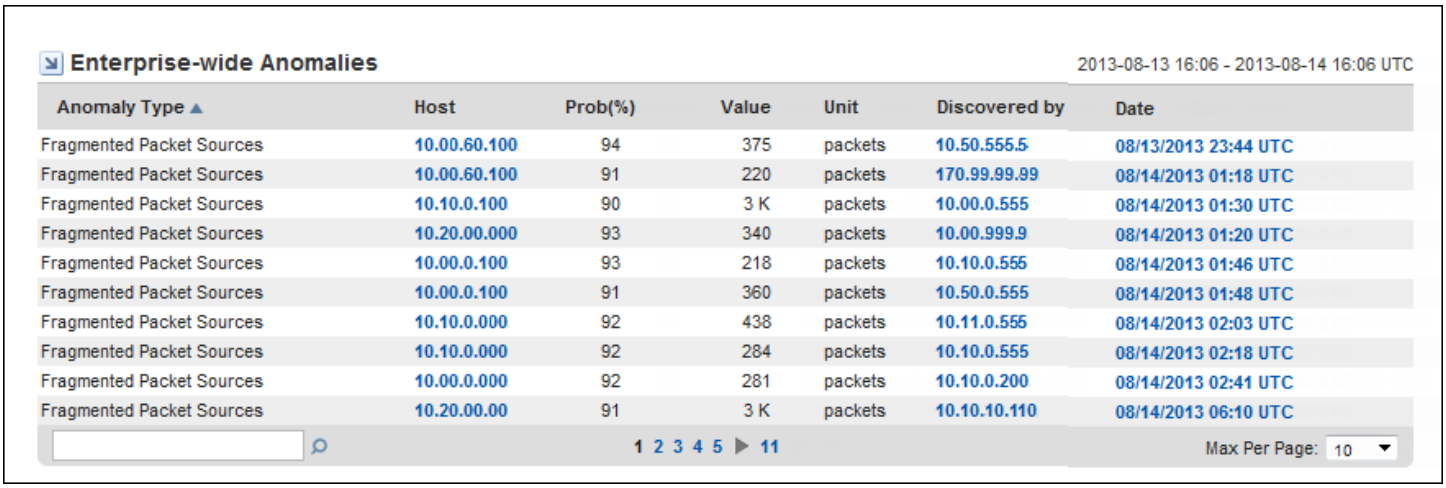
The view provides the following information about anomalous network behavior:

- **Host**
  The IP address of the host that displays the anomalous behavior. The host may be a client computer, a server, a router, or an interface. The program attempts to resolve the hostname of the IP address and displays that name in the **Host** field.
- **Anomaly Index**
  The count of the anomalies in the cluster, weighted by their role as either primary or secondary. The anomaly correlation algorithm compares each particular behavior to the typical patterns for the network traffic type. The higher the index number, the more severe the issue is.
- **Types**
  The number of different types of anomalous network behavior that occurred during the reporting period.
- **Date**
  The date and time of the first correlated anomaly on the host.
  The time may vary by up to 15 minutes from the time when the flows actually took place. Data is pulled from the Harvesters for analysis at 15-minute intervals.
- **Date Link**
  Click the *Date* link in the **Enterprise-Wide Correlated Anomalies** view to go to the **Anomaly Detector Drill-In** table.

Enterprise-Wide Anomalies

The **Enterprise-Wide Anomalies** view is a comprehensive summary of the anomalous behavior during the reporting time frame, with details about the anomaly type, location, and size.

This view is useful for beginning an investigation of problem behavior or for initiating troubleshooting procedures to stem an attack. The view provides more detailed information about the anomalies you see in other views. This view also identifies network locations to begin investigating.



Enterprise-wide Anomalies 2013-08-13 16:06 - 2013-08-14 16:06 UTC

Anomaly Type ▲	Host	Prob(%)	Value	Unit	Discovered by	Date
Fragmented Packet Sources	10.00.60.100	94	375	packets	10.50.555.5	08/13/2013 23:44 UTC
Fragmented Packet Sources	10.00.60.100	91	220	packets	170.99.99.99	08/14/2013 01:18 UTC
Fragmented Packet Sources	10.10.0.100	90	3 K	packets	10.00.0.555	08/14/2013 01:30 UTC
Fragmented Packet Sources	10.20.00.000	93	340	packets	10.00.999.9	08/14/2013 01:20 UTC
Fragmented Packet Sources	10.00.0.100	93	218	packets	10.10.0.555	08/14/2013 01:46 UTC
Fragmented Packet Sources	10.00.0.100	91	360	packets	10.50.0.555	08/14/2013 01:48 UTC
Fragmented Packet Sources	10.10.0.000	92	438	packets	10.11.0.555	08/14/2013 02:03 UTC
Fragmented Packet Sources	10.10.0.000	92	284	packets	10.10.0.555	08/14/2013 02:18 UTC
Fragmented Packet Sources	10.00.0.000	92	281	packets	10.10.0.200	08/14/2013 02:41 UTC
Fragmented Packet Sources	10.20.00.00	91	3 K	packets	10.10.10.110	08/14/2013 06:10 UTC

1 2 3 4 5 ▶ 11 Max Per Page: 10

You can edit the following view settings:

- Time frame for all views on the page
- View title and context

This view is not included by default on the **Anomaly Detector** page in the Performance Center Console. To see this view, add it to a page or to a new custom page. The graphic shows an example view in the CA NetQoS Performance Center Console.

The view provides the following information about anomalous network behavior:

- **Anomaly Type**
  The type of anomalous behavior.
- **Host**
  The name or IP address of the host on which the anomalous behavior is detected. The host may be a client system, a server, a router, or an interface. The program attempts to resolve the hostname of any IP address and displays that name in this field.
- **Probability**
  The calculated likelihood that flagged packet flows are truly anomalous.
  Probability is expressed as a percentage. For example, if the probability for an anomaly type is 91%, the packet flows that triggered the reported anomalous behavior are calculated to have a 91% probability of being truly anomalous. In this case, the packet flows have a low probability of occurring normally on this network.
- **Date**
  The date and time that the anomalous behavior is detected. The time may vary by up to 15 minutes from the time when the flows actually took place. Data is pulled from the Harvesters for analysis at 15-minute polling intervals.

Anomaly Drill-In

If you drill into an anomaly cluster from the **Enterprise-wide Correlated Anomalies** view, the **Anomaly Drill-In** table opens. For each anomaly, the table lists the probability, value, originating router and interface, and the time that the anomaly occurred. You can use the **Date** link to drill into a trend chart that shows the value and probability over time.

The **Anomaly Drill-In** view provides the following information about each anomaly:

- **Anomaly Type**
  The type of anomalous behavior.
- **Host**

The name or IP address of the host on which the anomalous behavior is detected. The host may be a client system, a server, a router, or an interface. The program attempts to resolve the hostname of any IP address and displays that name in this field.

- **Host Link**
  Click a *Host* link to go to more granular information about the device that has the anomaly. Clicking a **Host** link may be the first step in troubleshooting the anomaly.
  Host link destinations are based on the sensor type. For many DX NetOps sensors, the Host link opens the page for defining a Flow Forensics report in the NFA console, which has pre-populated report filters.
- **Prob(%)**
  The calculated likelihood that flagged packet flows are truly anomalous.
  Probability is expressed as a percentage. For example, if the probability for an anomaly type is 91%, the packet flows that triggered the reported anomalous behavior are calculated to have a 91% probability of being truly anomalous. In this case, the packet flows have a low probability of occurring normally on this network.
- **Value**
  The value that triggered the report of anomalous behavior, expressed in the units of measure shown in the Unit column. For example, the value could be the number of gigabytes of data in the anomalous flow.
- **Metric/Unit**
  The unit of measurement that is used to express the Value, such as packets, flows, or destination hosts (dest hosts).
- **Discovered by**
  The router, interface, or data source that detected the anomalous data.
- **Discovered by Link**
  Click a *Discovered By* link to view details. The link destination is determined by the type of anomaly:
  - DX NetOps Anomalies: Router or interface page in the Performance Center Console
  - Anomalies from other data sources: Main page for the originating product
- **Date**
  The date and time that the anomalous behavior is detected. The time may vary by up to 15 minutes from the time when the flows actually took place. Data is pulled from the Harvesters for analysis at 15-minute polling intervals.
- **Date Link**
  Click the *Date* link to go to the Anomaly Trend view. This view shows the value and probability of the anomaly over time.

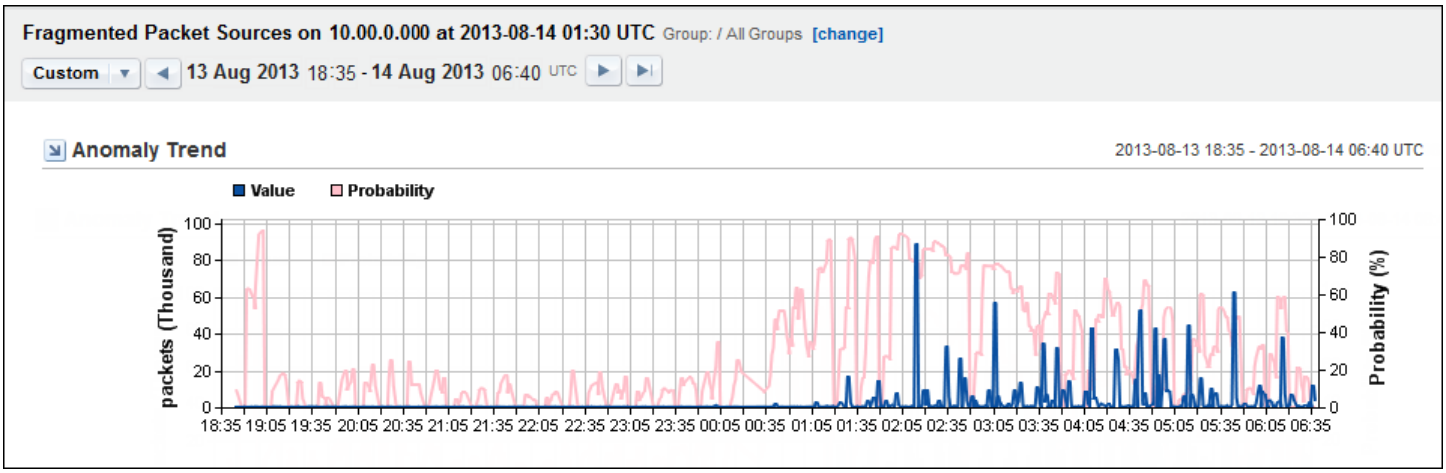
You can edit the following view settings:

- Time frame.
- View title.

> **NOTE**
> If your deployment includes CA Performance Center, you can use the **Zoom** feature to interactively limit the time frame.

Anomaly Trend

The **Anomaly Trend** view shows the value and probability of the anomaly over time. To display this trend chart, click a link in the **Date** column in either the **Anomaly Drill-in** view or the **Enterprise-Wide Anomalies** view. The following example shows an **Anomaly Trend** view in the CA NetQoS Performance Center Console.



The view shows the pattern of deviation from normal network behavior. You can see when the behavior began and how severe the behavior was. A longer term view can help to determine patterns over days, weeks, or months.

The recorded values are shown as a blue trend line on the X-Axis. The probability that the behavior is a true anomaly is shown as a pink trend line on the Y-Axis.

You can edit the following view settings:

- Time frame
- View title

Note: If your deployment includes CA Performance Center, you can use the **Zoom** feature to interactively limit the time frame.

CA Anomaly Detector Sensors and Troubleshooting

CA Anomaly Detector provides a diverse set of sensors that search for potential network anomalies in the collected data.

Most sensors are enabled by default. The sensors become active as soon as you add the supporting data source to CA Anomaly Detector. You can disable sensors or apply filtering to exclude selected devices for individual sensors.

Sensor configuration also includes a probability factor. This factor controls sensor sensitivity and frequency of alerts.

CA Network Flow Analysis Sensors

DX NetOps provides information to help you understand how application traffic affects your network performance. DX NetOps enables you to see which applications are using bandwidth, who is using the bandwidth, and when. This knowledge helps you can timely and cost-effective decisions to optimize network performance.

When you add DX NetOps as a data source for Performance Center, the Harvesters are added automatically as possible data sources for CA Anomaly Detector. Special considerations apply when you enable Harvesters as data sources.

RST-Only Sources

The RST-Only Sources sensor looks for the following types of connections:

- Hosts that send out RST TCP packets with no ACK packets to acknowledge an open connection to a server.
- Servers that receive only RST packets from certain hosts with no ACK or SYN packets.

Troubleshooting an RST-Only Sources Alert

An alert from the RST-Only Sources sensor could indicate one of the following issues:

- A server that is running out of resources for an active application.
  Investigate whether a server upgrade is needed.
- A server that has a previously active application, which has become inactive.
  Try restarting associated application services.
- Users who are connecting to the wrong server.
  Check for DNS issues and correct any issues that you find.
- A host that is the victim of port-scanning activity.
  Identify and use the firewall or ACL to block the offending host from sending data on the network while you investigate.

Fragmented Packet Sources

The Fragmented Packet Sources sensor looks for sources of packet fragmentation. Packet fragmentation may indicate poor application delivery or "frag attacks" that can circumvent Access Control Lists (ACLs) and stateless firewalls.

Troubleshooting a Fragmented Packet Sources Alert

An alert from the Fragmented Packet Sources sensor may indicate one of the following issues:

- A Layer 3 network maximum transmission unit (MTU) mismatch.
  Check the MTU settings on the affected interfaces. Interfaces on the same router can have different MTU settings. Any IP packets that exceed the configured maximum number of bytes are broken into fragments. Stateless firewalls in particular often drop packet fragments.
- A "frag attack" that is designed to exhaust network resources.
  Packets can be fragmented normally by routers if the packets exceed the MTU size on a router interface. In this case, a flag is set in the IP header to indicate that the segment is a fragment. The packets continue until the end of fragment (EOF) flag is set. In a frag attack, fragmented packets are sent in a steady stream, but a packet with the EOF flag set is never sent. The packets eventually fill the receive buffer on the target host and disable it.
  If you suspect a frag attack, identify and block the offending host from sending data on the network by using a firewall or ACL.
- Attempted reconnaissance or fingerprinting.
  Fragmented packets may indicate that an attacker is probing the host for system credentials, such as operating system level and known vulnerabilities.
  Identify and block the offending host from sending data on the network by using a firewall or ACL.

Even though a host who communicates with another host across a VPN tunnel is a legitimate source for fragmented packets, the host is still a source of increased packet load. Adjusting the MTU on the host with the VPN client can help.

CA Anomaly Detector currently cannot actually identify the sources of packet fragmentation. The software can identify only the original source of the IP packet. Any Layer 3 device along the path may be the source of the actual act of fragmenting the packet.

Large DNS Packet Sources

The Large DNS Packet Sources sensor looks for DNS requests that are larger than usual. Such packets might indicate tunneling attempts or data ex-filtration. Data ex-filtration is a network security violation in which a user discreetly attempts to send data from an internal network to an external location.

Troubleshooting a Large DNS Packet Sources Alert

If you suspect data ex-filtration, click the link for the offending host to investigate the suspected user by using a **Flow Forensics** report in the NFA console. You can also use other packet-inspection tools to identify the data that was sent.

Large ICMP Packet Sources

The Large ICMP Packet Sources sensor looks for ICMP packets that are unusually large. Large packets might indicate tunneling attempts or data ex-filtration. Data ex-filtration is a network security violation in which a user discreetly attempts to off-load data from an internal network to an external location.

Troubleshooting a Large ICMP Packet Sources Alert

If you suspect data ex-filtration, click the link for the offending host and investigate the suspected user in a **Flow Forensics** report in the NFA console. You also can use a packet-inspection tool to identify the data that was off-loaded.

TTL Expired Sources

The TTL Expired Sources sensor looks for possible router misconfiguration, such as routing loops, by monitoring for time to live (TTL) expiration messages.

Troubleshooting a TTL Expired Sources Alert

An alert from the TTL Expired Sources sensor could indicate one of the following issues:

- A network configuration issue such as a routing loop or equipment problem. A routing loop occurs when packets are routed incorrectly so that they travel in a continuous circle. The loop typically occurs when a router or line fails before routine notifications about the link failure have had time to reach the other network routers. Routing loops often appear when network changes are made.
  Try to identify and correct any network outage that becomes apparent. Try to find any "black hole" network ranges and check the associated routers.
- Heavy traceroute activity.
  Investigate any unusual traceroute activity to determine whether the activity is legitimate. For example, investigate whether the traceroute activity is performed for testing or troubleshooting.

Congestion Sources

The Congestion Sources sensor monitors ICMP Source Quench messages by looking for devices that are overwhelmed by incoming traffic.

If the packet receiver experiences congestion, it sends a source quench message to the packet sender, which causes the sender to initiate a TCP slow start. This has a significant negative impact on performance. CA Anomaly Detector reports the host that is the source of the overwhelming traffic and indicates which congested router reported the issue.

Troubleshooting a Congestion Sources Alert

Look at the load on the device that issues the source quench message. Legitimate traffic may be overloading the device.

If the host is under attack, correlated anomalies are likely to appear in the **Enterprise-Wide Correlated Anomalies** view.

Previously Null Routed Sources

The Previously Null Routed Sources sensor aggregates data about traffic from null-routed sources. The sensor reports on the hosts with the highest volume of normally-routed traffic that previously had null-routed traffic.

Troubleshooting a Previously Null Routed Sources Alert

An alert from the Previously Null-Routed Sources sensor could indicate that Access Control Lists (ACLs) are applied inconsistently across the enterprise. ACL problems could result from a security violation.

Audit all ACLs to ensure that they are properly configured and conform to established network access and usage policies. Try to determine whether the ACLs have unauthorized modifications.

It is also possible that the host is sending malicious traffic in various ways and the ACLs are catching some of that traffic. It might be worthwhile to investigate the type of traffic that the ACLs do not block. To do this, use the **Host** link on the **Anomaly Drill-in** to run a **Flow Forensics** report in the NFA console.

SYN/RST-Only Packet Sources

The SYN/RST-Only Packet Sources sensor looks for hosts that send out unusually large amounts of SYN-RST packets. This behavior could indicate a SYN or RST flood.

Troubleshooting an SYN/RST-Only Packet Sources Alert

An alert from the SYN/RST-Only Packet Sources sensor could indicate that someone is attempting to hack into the network. Specifically, SYN-RST packets are associated with attempts to bypass a firewall perimeter.

SYN/RST-only packet flows signal a connection establishment request that is immediately followed by a reset request, which is not characteristic of normal TCP behavior. This behavior typically indicates scanning activity, which might include OS fingerprinting.

SYN-Only Packet Sources

The SYN-Only Packet Sources sensor looks for:

- Hosts that send out only SYN TCP packets with no corresponding ACK packets to acknowledge an open connection to a server
- Servers that receive only SYN packets from certain hosts with no corresponding ACK or RST packets.

Requests for a server connection without establishing a valid socket are typical of a denial-of-service attack by using SYN packet reflection.

Troubleshooting a SYN-Only Packet Sources Alert

An alert from the SYN-Only Packet Sources sensor could indicate a worm infection or a SYN flood, which is a type of denial of service attack.

A SYN flood involves a series of SYN TCP packets that contain invalid source IP addresses. The target server is unable to establish a valid connection in response to the SYN request, but it still allocates the necessary resources and waits until a timeout expires for the ACK packet from the requesting host. The server is quickly brought to a standstill by trying to process invalid connection requests.

If you suspect a denial of service attack, use CA Anomaly Detector to identify each offending host, then use firewalls or ACLs to try to block the host from sending data on the network. You also can take the affected server offline.

High Packet Fan Out

The High Packet Fan Out sensor looks for packets that fan out from a single host to many hosts. Packet fanning is typical of a spreading virus or worm. A High Packet Fan Out anomaly is a primary type of anomaly, which typically occurs in conjunction with another type of anomaly.

Troubleshooting a High Packet Fan Out Alert

An alert from the High Packet Fan Out Sources sensor may indicate a spreading virus or worm infection. Some viruses or worms operate by creating "zombie" hosts that spread the infection in a fan-out pattern.

If you suspect a virus infection, use CA Anomaly Detector to identify each offending host, then use a firewall or ACL to try to block the host from sending data on the network. You also can take the affected server offline.

High and Variable Volume-Out

The High and Variable Volume-Out sensor tracks host volume and detects sudden changes in server output. These changes may indicate server misconfiguration, virus infection, or unauthorized activity, such as BitTorrent server activity. These conditions produce a secondary type of alert, which typically occurs with another type of anomaly.

Troubleshooting a High and Variable Volume-Out Alert

An alert from the High and Variable Volume-Out sensor may indicate one of the following issues:

- Server misconfiguration.
  Find the source of the anomalous traffic and investigate the activity.
- A new authorized or unauthorized application that is deployed on the network.
  Find the source and investigate. Such behavior may indicate the presence of a BitTorrent or other peer-to-peer file-sharing server.
  Peer-to-peer file-sharing or other unauthorized activity often consumes excessive network resources. This type of activity potentially can expose your enterprise to copyright-infringement legal action.

High and Variable Volume-In

The High and Variable Volume-In sensor tracks host volume and detects sudden changes in incoming volume. These changes may indicate server misconfiguration, virus infection, or unauthorized activity, such as BitTorrent server activity. The activity produces a secondary type of alert, and it typically occurs together with another type of anomaly.

Troubleshooting a High and Variable Volume-In Alert

An alert from the High and Variable Volume-In sensor may indicate one of the following issues:

- Server misconfiguration.
  Find the source of the anomalous traffic and investigate the activity.
- A new authorized or unauthorized application that is deployed on the network.
  Find the source and investigate. The behavior may indicate the presence of a BitTorrent or other peer-to-peer file sharing server.
  Peer-to-peer file-sharing or other unauthorized activity often consumes excessive network resources. This type of activity potentially can expose your enterprise to copyright-infringement legal action.

High Flow Sources

The High Flow Sources sensor tracks host flows and detects sudden changes in output that may indicate server misconfiguration, virus infection, or unauthorized activity, such as BitTorrent server activity.

Troubleshooting a High Flow Sources Alert

An alert from the High Flow Sources sensor may indicate an increase in connections. The increase is often caused by the introduction of a new authorized or unauthorized application on the network.

Find the source and investigate. The behavior may indicate the presence of a BitTorrent or other peer-to-peer file sharing server.

Peer-to-peer file-sharing or other unauthorized activity often consumes excessive network resources. This type of activity potentially can expose your enterprise to copyright-infringement legal action.

Frags and Loss Sources

The Frags and Loss Sources sensor looks for sources that cause packet loss by monitoring "Fragment reassembly time exceeded" messages. Packet fragmentation can be used to circumvent blocking rules on some firewalls.

Troubleshooting a Frags and Loss Sources Alert

Fragmentation and packet loss occurs when you deploy high bit-rate Windows Media content.

Fragmentation can cause retransmissions when some fragments fail to arrive in time so that the packet is lost. Reliable protocols such as TCP retransmit all of the fragments in order to recover from the loss of a single fragment.

Dest Unreachable Sources

The Dest (Destination) Unreachable Sources sensor looks for unusually high levels of network traffic that fails to reach the destination designated by the packet. The sensor operates by monitoring ICMP return codes, whether for a network, host, or port. This type of traffic flood consumes bandwidth on the affected link, which causes reduced performance for the link's legitimate users.

Troubleshooting a Dest Unreachable Sources Alert

An ICMP flood (also known as *Ping flood* or *Smurf attack*) is one source of a Destination Unreachable Sources alert. This type of Denial of Service attack sends large amounts of ICMP packets or over-sized ICMP packets to a system. The goal of the attack is to crash the system's TCP/IP stack so that the system stops responding to TCP/IP requests.

An ICMP attack can come in many forms. An ICMP flood is typically accomplished by broadcasting a large number of ICMP pings or UDP packets. The attack sends so much data to the system that it slows down enough to disconnect from normal business applications due to timeouts.

If you suspect a Denial of Service attack, use CA Anomaly Detector to identify each offending host, then use a firewall or ACL to try to block the host from sending data on the network. You also can set up an ACL that blocks external traffic to the affected host or you can take the affected server offline.

Non-Local Sources

The Non-Local sensor is turned off by default. Consult with your technical representative to determine whether this sensor is beneficial for your network.

The Non-Local sensor looks for traffic that does not have either a private source or destination IP address. The sensor assumes that the enterprise uses private IP addresses internally--that the majority of traffic has a private source or destination IP address.

For example, suppose a system in the network has a fake external IP address and the system sends messages to another fake external IP address. The resulting traffic can stop legitimate network traffic by tying up the routers and interfaces closest to that system.

Depending on your enterprise network configuration, legitimate traffic may flow between non-private IP addresses. If this is the case, your technical representative may be able to help you configure the sensor to exclude IP address ranges and avoid false alerts.

Troubleshooting a Non-Local Alert

Determine whether the non-private IP addresses are used for legitimate traffic. If the traffic is not legitimate, use the router and interface information that CA Anomaly Detector provides to apply ACLs to mitigate the traffic while you locate the offending source device.

CA NetVoyant Sensors

CA NetVoyant is a performance analysis and reporting software package that automates the collection, analysis, and reporting of critical device data. Using Simple Network Management Protocol (SNMP), CA NetVoyant automatically polls and correlates data from devices on your network. Data from routers, switches, servers, RMON2 probes, frame relay circuits, logical segments, and wide area links are collected and organized to give you information about your network.

CA Anomaly Detector contacts the CA NetVoyant database to gather data for analysis. The following sensors are used to detect anomalies in the data that CA NetVoyant collects that could indicate an issue with your network.

Incoming Discard Rate

The Incoming Discard Rate sensors look for patterns of incoming packets that are dropped even though they contain no errors.

Troubleshooting an Incoming Discard Rate Alert

An Incoming Discard Rate alert may indicate one of the following issues:

- Overutilization
- Denial of service
- VLAN misconfiguration

This alert may also result from inbound policing on the interface. QoS policies affect discards-in and discards-out counters, depending on whether the QoS policy is inbound or outbound.

Incoming Error Rate

The Incoming Error Rate sensor looks for patterns in incoming errors.

Troubleshooting an Incoming Error Rate Alert

Incoming Errors typically relate to CRC or other physical layer conditions and are not due to QoS, routing, or configuration issues.

Dropped Packets

The Dropped Packets sensor looks for patterns in queue drops.

Troubleshooting a Dropped Packets Alert

Dropped packets, or queue drops, may indicate buffer depletion. Input queue drops typically occur when a packet is process-switched. A process switch means that the router cannot use a preferable route-cache method, such as fast switching or Cisco Express Forwarding (CEF), to handle the forwarding decision. If input drops are still present, it implies that there is simply too much traffic. Output drops are caused by a congested interface.

Buffer Misses

The Buffer Misses sensor looks for patterns in buffer misses. Buffer failures are one of the most common reasons for packet drops.

Troubleshooting a Buffer Miss Alert

Look at the buffers on your router to determine whether they are configured properly.

CA Unified Communications Monitor Sensors

CA Unified Communications Monitor (UCM) ensures the availability and performance of Cisco Voice over IP (VoIP) systems. The program passively monitors the VoIP call setup traffic and the call quality for IP phones, call servers, and voice gateways and keeps a continuous record of VoIP call setup performance and call quality from one end of the network to the other.

CA Anomaly Detector contacts the UCM database to gather data for analysis. The following sensors are used to detect anomalies in the UCM data, which may indicate an issue with your VoIP network.

Voice Call Fan Out

The Voice Call Fan Out sensor looks for patterns in voice call fanouts. In a fanout, a single IP phone calls a large number of unique phone numbers in a short time.

Troubleshooting a Voice Call Fanout Alert

The IP address of the phone that originates the call fanout is identified in the anomaly event. Investigate the phone at this address to determine why it originates fanout calls. Use the CA Unified Communications Monitor dashboard to determine which calls were made in the fanout time frame.

Voice Call DoS

The Voice Call DoS sensor looks for a single phone that calls repeatedly over a short time. This type of attack differs from other Denial of Service (DoS) attacks because of its low volume. It only takes about 10 calls per minute to keep a phone ringing all the time. A traditional intrusion detection system might not catch the attack.

Troubleshooting a VoIP DoS Attack Alert

The anomaly event identifies the IP address of the phone that is called. Check to see who is calling the phone. Use the CA Unified Communications Monitor dashboard to determine which calls were made during the attack time frame.

Voice Server Distress

The Voice Server Distress sensor looks for calls that are not completed, which are going through a specific call server. The calls are either abandoned or they fail setup. The abandons and failures are weighted with the call volume to determine whether to flag a server. Even if the condition does not result from a security attack on the server, it is still a performance anomaly that warrants investigation.

Troubleshooting a Voice Server Distress Alert

The anomaly event identifies the IP address of a call server that has a large percentage of errored or abandoned calls. Use the CA Unified Communications Monitor dashboard to look at the details of calls that are attempted, completed, and in error for the call server during the anomaly time frame.

CA Application Delivery Analysis Sensors

CA Application Delivery Analysis provides end-to-end performance monitoring through reports and views that show historically normal performance for users and metrics that cross acceptable performance thresholds. CA Application Delivery Analysis gathers troubleshooting information and helps you determine the origin of an application, network, or server performance problem.

CA Anomaly Detector contacts the CA Application Delivery Analysis database to gather data for analysis. The following sensors are used to detect anomalies in the data collected by CA Application Delivery Analysis that might indicate an issue with your network.

Retransmission Time

The Retransmission Time sensor looks for patterns in retransmissions.

TCP retransmission behavior has a significantly negative effect on application performance and network use. Communication across the affected connection can slow to a halt until a missing segment is retransmitted successfully. Only a limited number of missing segments can remain outstanding at any time.

Troubleshooting a Retransmission Time Alert

An anomaly in retransmission delay typically is caused by several critical components:

- Congested Path: Retransmission delay can increase when interfaces between the clients and servers are congested.
- QoS: A misapplied QoS policy for a particular application might cause increased buffer or queue drops.
- Backup/Routing Issue: A change in the per hop route might cause the issue. Additional retransmission delay could result from routing changes that add congestion to a pipe that is already congested or a pipe that is smaller, such as a backup link.

Identify sources of packet loss along the traffic paths. Keep in mind that the paths going to the server may be different from the paths coming from the server. If that is the case, analyze all paths.

Refused Sessions

The Refused Sessions sensor looks for a pattern in refused sessions. Refused session counts could indicate a SYN attack or a server that cannot respond to all requests.

Troubleshooting a Refused Sessions Alert

A server tends to stop fulfilling and refusing sessions when it reaches its capacity to handle more data. Sessions could also be refused because a load balancer is sending heart beat messages to the servers.

Refused Sessions counts could also result when legitimate traffic overloads a server that has inadequate resources for additional TCP session requests.

An alert may indicate a SYN or RST flood. If this is the case, the anomaly should be visible in the **Enterprise-Wide Correlated Anomalies** view.

Packet Load

The Packet Load sensor looks for changes in the number of bytes per packet that goes to a particular server. Applications that run on servers typically have a fairly constant ratio between the number of packets they receive in requests for their service and the packet volume. This sensor looks for anomalous changes in that ratio.

Troubleshooting a Packet Load Alert

In WAN applications, you want to have as much data in each packet as the maximum transmission unit (MTU) allows. Applications that have less data in the packets have to wait longer to get data across the network. Analyze whether the application is loading the packets fully to send to remote users.

An alert may indicate varying implementation of the clients or it may indicate that unexpected clients are trying to subvert the server.

Documentation Legal Notice

at any time. This Documentation is proprietary information of Broadcom and may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of Broadcom.

If you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all Broadcom copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to Broadcom that all copies and partial copies of the Documentation have been returned to Broadcom or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION “AS IS” WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is Broadcom Inc.

Provided with “Restricted Rights.” Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.


A Broadcom
Company
technologies